

СЕКЦІЯ:

Математичне моделювання

М.А. Новотарський

Б.Б. Нестеренко

**ШТУЧНІ НЕЙРОННІ МЕРЕЖІ:
ОБЧИСЛЕННЯ**

**ШТУЧНІ НЕЙРОННІ МЕРЕЖІ :
ОБЧИСЛЕННЯ**

**Інститут математики НАН України
Київ 2004**

М а т е м а т и к а
та її застосування

Головний редактор
А.М. Самойленко

Інститут математики
Національної академії наук України, Київ

Праці інституту математики
Національної академії наук України

Том 50

УДК 681.324

М.А. Новотарський, Б.Б. Нестеренко. Штучні нейронні мережі: обчислення // Праці Інституту математики НАН України. – Т50. – Київ: Ін-т математики НАН України, 2004. – 408 с.

В монографії розглянуто основні типи архітектур штучних нейронних мереж (ШНМ). Головну увагу приділено ієрархічним, конкурентним і рекурентним нейронним структурам та принципам функціонування нейронних мереж типу ART. Описано особливості застосування методів навчання з учителем для одношарових та багатошарових ШНМ прямого поширення. Окремо розглянуто методи навчання з учителем для штучних нейронних мереж типу РБФ. Наведено основні методи оптимізації ШНМ. Розглянуто використання нейронних мереж для розв'язування задач апроксимації та задач, які виникають при обробці сигналів. Дається огляд ітераційних методів, орієнтованих на реалізацію в штучних нейронних мережах. Розглянуто різні підходи до організації комунікаційних процесів між нейронами та основні структури ШНМ, що їх реалізують. Значну увагу приділено мультисітковим ітераційним методам з метою покращення параметрів паралельних локально-асинхронних методів при реалізації їх на ШНМ.

Для наукових та інженерно-технічних працівників, що працюють у сфері застосування нейронних мереж до паралельних обчислень.

*Затверджено до друку вченою радою
Інституту математики НАН України*

ISBN 966-02-2571-7

ISBN

© М.А. Новотарський,
Б.Б. Нестеренко,
2004

ПЕРЕДМОВА

*Метелик-одноденка не знає
ні початку, ні закінчення місячного місяця;
плярмиста цикада не знає ні весни, ні осені...*
Чжуан-Цзи

Прискорений розвиток високих технологій став причиною значного зростання актуальності сфер наукової діяльності, що пов'язані з математичним моделюванням процесів та явищ. Моделювання реальних об'єктів навколишнього світу, як правило, супроводжується значними труднощами, які виникають ще на етапі постановки задачі. Ці труднощі частіше за все є наслідком недосконаlosti обчислювальних методів та засобів їх реалізації і причиною існування великої кількості нерозв'язаних проблем, що виникають при дослідженні аеродинамічних моделей, задач магнітогідродинаміки та астрофізики, моделюванні процесів фізики твердого тіла за наявності фазових переходів із врахуванням квантових ефектів, тощо. Особливо актуальними стають задачі, що мають некоректну постановку, для яких відсутні оптимальні алгоритми їх розв'язку. Власне кажучи, при розв'язуванні саме таких задач найчастіше використовують штучні нейронні мережі (ШНМ).

На сьогодні не існує загально визнаного означення штучних нейронних мереж, але важливо відрізнити їх від біологічних нейронних мереж, які є незмірно складніші від тих математичних моделей, на основі яких побудовані ШНМ. Сучасні штучні нейронні мережі складаються з великої кількості простих процесорних елементів із деякою кількістю локальної пам'яті (нейронів), об'єднаних за допомогою дискретних або неперервних комунікаційних каналів. Задачі, що розв'язують на ШНМ, підлягають декомпозиції на множину локальних задач, кожна з яких може бути розв'язаною за допомогою окремого нейрону шляхом реалізації певного алгоритму обробки локальних даних. Для штучних мереж існує також поняття навчання – спеціального процесу, що забезпечує настройку параметрів мережі з метою підвищення ефективності її функціонування. Відомо два основних підходи до навчання штучних нейронних мереж, які дістали назву навчання з учителем та навчання без учителя. Навчання із вчителем пов'язане з існуванням навчальної вибірки, тобто множини

пар, кожна з яких складається із вхідного вектора даних та ідеального вихідного вектора. Навчання з учителем має на меті досягнення такого стану нейронної мережі, при якому кожному з вхідних векторів навчальної вибірки відповідає максимально близький до ідеального вихідний вектор. Навчання без учителя є більш складним процесом, який базується на модифікації внутрішнього стану ШНМ на основі оцінки вхідних даних за допомогою певної кількості критеріїв та набутого в ході навчання досвіду.

Для розв'язування крайових задач на ШНМ застосовують як прямі, так і ітераційні методи, а також різноманітні їх комбінації (гібридні методи). Більшість із них, в основному, зводиться до розв'язання системи лінійних алгебраїчних рівнянь, які отримують в результаті дискретизації області. Слід відмітити, що використання прямих методів у даній сфері є традиційним і найчастіше полягає у відображенні одержаної системи лінійних алгебраїчних рівнянь на архітектуру типу Хопфілда. Далі процес одержання розв'язку базується на мінімізації енергетичної функції нейронної мережі.

Ще один відомий підхід полягає у використанні нейронної мережі як універсального апроксиматора функцій. Так, деякі типи сплайнів, наприклад, B_1 -сплайни, можуть бути одержані шляхом суперпозиції кусочно-лінійних активаційних функцій. Використовуючи такі сплайни як базисні функції, розв'язок системи лінійних алгебраїчних рівнянь можна отримати шляхом підбору відповідних коефіцієнтів даних сплайнів. Для реалізації цього підходу застосовують ШНМ прямого поширення з відповідними кусочно-лінійними активаційними функціями.

Якщо відмовитися від обмежень на форму активаційної функції нейрона, то можна одержати більш сучасний підхід до розв'язання крайових задач, що базується на властивості апроксимації функцій, характерній для ШНМ прямого поширення. Архітектура нейронної мережі повинна забезпечувати близький до аналітичного розв'язок, точне значення якого одержують шляхом навчання, що проводиться за критерієм мінімізації функції похибки.

Більшість крайових задач можна описати з достатньою точністю за допомогою множини локальних задач, кожна з яких описує динаміку процесів у фіксованій точці області. На використанні згаданого підходу базуються кліткові нейронні мережі, у яких кожний нейрон

об'єднаний комунікаційними каналами з фіксованим числом сусідніх нейронів.

Ітераційні методи розв'язання крайових задач на нейронних мережах є відносно молодю сферою застосування ШНМ. Основною перешкодою до їх застосування тривалий час були технологічні проблеми, пов'язані з підвищеною складністю обчислень у кожному нейроні. Сучасні досягнення високих технологій дозволили будувати нейрокомп'ютери, що налічують тисячі нейронів, кожен з яких може забезпечувати складні обчислення та має локальну оперативну пам'ять. Це спричинило розвиток нових обчислювальних методів, у тому числі ітераційних.

Нейронні ітераційні методи для розв'язування рівнянь типу $AX = F$ потребують застосування рекурентних ШНМ із лінійними активаційними функціями. Ітераційний процес задається виразом: $X(n+1) = WX(n) + C$, де X — вектор стану нейронної мережі, W — матриця синаптичних зв'язків, C — вектор констант.

Найпростіша архітектура ШНМ для реалізації ітераційного алгоритму розв'язання диференціальних рівнянь складається з трьох шарів, кожен з яких реалізує відповідний етап ітерації. Перший шар нейронів забезпечує обчислення нев'язки, значення якої використовується на другому шарі з метою визначення умов завершення алгоритму. Якщо умови завершення виконуються, то стан вихідного вектора ШНМ відповідає розв'язку рівняння. В протилежному випадку вектор нев'язки передається до третього шару для формування нового наближення вихідного вектора.

Такою реалізацією ітераційного процесу на ШНМ зовсім не вичерпуються всі можливі підходи, що впливають із розмаїття ітераційних методів. Значного підвищення ефективності ШНМ можна досягти у випадку застосування методів, які розроблялися з метою застосування їх у паралельних структурах. Особливість таких методів полягає у тому, що вони допускають наявність деякої кількості елементів обробки даних, об'єднаних між собою каналами зв'язку. Оскільки одержання кінцевого результату в даному випадку залежить від успішної роботи всіх елементів архітектури, то критерії ефективності визначаються ступенем застосування принципів асинхронності та локальності.

Принцип асинхронності визначає ступінь залежності нейрона від інших нейронів у ході обчислювального процесу. При повній

відсутності асинхронності, тобто коли обчислення є синхронними, вони можуть успішно продовжуватися тільки за умови одержання конкретних даних у задані моменти часу. Неможливість на практиці забезпечити виконання таких умов для кожного нейрона веде до катастрофічної деградації всього процесу розв'язування задачі. Виходом із даної ситуації є введення принципу асинхронності, який полягає у зменшенні критичності до „строку давності” даних або до моменту їх одержання.

У більшості випадків для обчислення значення функції в деякій фіксованій точці області необхідно використовувати поточні значення функції у всіх інших точках даної області. За умови паралельних обчислень це веде до збільшення міжнейронних комунікацій, що також призводить до деградації процесу розв'язування задачі. Застосування принципу локальності дозволяє одержати розв'язок крайової задачі, використовуючи в ході обчислень тільки дані від деякої скінченної множини сусідніх нейронів.

Природно, що застосування принципів асинхронності та локальності поряд із позитивними має також негативні риси, основною з яких є зниження швидкості збіжності ітераційного процесу. З метою подолання цього недоліку сьогодні надають перевагу застосуванню технології мультисіткових обчислень, що передбачає організацію обчислювального процесу на певній послідовності сіток дискретизації області. Перехід від однієї сітки до іншої забезпечують спеціальні оператори, що дістали назву операторів пролонгації та рестрикції. Мультисіткова технологія має широке застосування не лише в нейромережних обчисленнях, а розглядається як самостійний напрямок обчислювальної математики.

У даній роботі зроблено спробу розглянути всі згадані аспекти, що супроводжують процес організації обчислень на штучних нейронних мережах.

У розділі 1 наведено короткий огляд історії розвитку нейронних мереж та розглянуто основні принципи функціонування як біологічного нейрона, так і його моделей. Штучні нейронні мережі, що існують поряд з іншими відомими паралельними структурами, мають свою сферу традиційних застосувань: задачі апроксимації неперервних функцій та класифікаційні задачі. Кожна з цих задач, а також ряд нових застосувань ШНМ, зокрема, для паралельних обчислень, характеризується певними, орієнтованими на їх розв'язання особливостями архітектур ШНМ. В роботі розглянуто ієрархічні,

конкурентні та рекурентні нейронні структури, а також принципи функціонування нейронних мереж типу ART.

Поряд із принципами функціонування моделі нейрона як елементарного вузла ШНМ та особливостями архітектури, вирішальним фактором, що забезпечує ефективність застосування нейронних мереж, є метод навчання. Методи навчання з учителем базуються на використанні послідовності навчальних вибірок, а механізм навчання полягає в модифікації вагових коефіцієнтів нейронів. Основним критерієм оцінки ефективності навчання є цільова функція, що дозволяє оцінити, наскільки робота ШНМ відповідає бажаному результату. В розділі 2 наводяться особливості застосування методів навчання з учителем для одношарових та багатшарових ШНМ прямого поширення. Окремо розглянуті методи навчання з учителем для штучних нейронних мереж типу РБФ.

Відомо, що в результаті застосування методів навчання з учителем до нейронної мережі не завжди вдається досягти бажаних результатів. Значного покращення параметрів функціонування ШНМ досягають шляхом застосування методів, що містять оптимізацію топології ШНМ та поняття узагальнення, метою якого є забезпечення ефективного функціонування не тільки на тих даних, які використовувались у процесі навчання, а в значно більшому просторі. В розділі 3 розглянуто дилему відхилення/дисперсії та методи визначення реальної похибки. Описані основні методи оптимізації: метод попередньої зупинки, регуляризація, усереднення, методи відсікання та нарощування.

Методи навчання без учителя застосовують у випадках, коли невідома навчальна вибірка, тобто не існує зовнішнього сигналу, який інформував би нейронну мережу про те, наскільки правильною є її реакція на вхідний сигнал. Тому для формування даних методів дуже важливо знати мету навчання, що безпосередньо пов'язана зі сферою застосування ШНМ. В розділі 4 розглянуто використання ШНМ для розв'язування задач, які виникають при обробці сигналів та для традиційного застосування в задачах апроксимації і класифікації. У сфері обробки сигналів для ШНМ найбільш відомими є аналіз важливих компонентів, що застосовується для зниження розмірності елементів даних, та аналіз незалежних компонентів, основною метою якого є одержання сигналів від недоступних джерел інформації шляхом виділення їх із лінійної суміші. Задачі апроксимації представлені великою кількістю методів, більшість яких базується на правилах Хебба та Ойя. В роботі також розглянуто ряд модифікацій

цих правил, що дозволяють підвищити ефективність функціонування ШНМ. Для розв'язання задач класифікації та категоризації частіше застосовують штучні нейронні мережі типу ART. У роботі наведено методи та алгоритми функціонування мереж Fuzzy ART, S-Fuzzy ART, SART та мережі FOSART.

Ускладнення моделі нейрона та розвиток сучасних технологій спричинили створення ітераційних методів, орієнтованих на реалізацію в штучних нейронних мережах. Дані методи використовують математичні моделі, що вперше були застосовані для паралельних обчислювальних структур. Розділ 5 містить опис основних ітераційних методів згаданого типу, даний у відповідності до хронології їх розвитку. Розглянуто різні підходи до організації комунікаційних процесів між нейронами та базові структури ШНМ, що їх реалізують. Викладено основи одностадійних методів, представлених методом хаотичних ітерацій, асинхронним методом та методом асинхронних ітерацій із нерухомими точками. Ці методи взяті за основу для створення двостадійних методів, які принципово відрізняються від одностадійних тим, що містять два типи ітерацій: *локальні ітерації* та *глобальні ітерації* або *макроітерації*. Локальними називають ітерації, що виконуються над елементами блока, а макроітерації формують чергове наближення вектора ітерації. Відомими представниками методів даного типу є мультирозщеплюючі ітераційні методи, основи яких також викладені в даній роботі. Викладені основи локально-асинхронних методів, орієнтованих на реалізацію в рекурентних кліткових нейронних мережах. Асинхронність у межах даних методів розуміють як можливість обчислення кожної компоненти вектора ітерації незалежно від решти компонент шляхом дії відповідної компоненти оператора переходу. При цьому послідовність ітерацій збігається до нерухомої точки, спільної для всіх компонент оператора переходу.

Розділ 6 присвячений мультисітковим ітераційним методам, що мають на меті покращення параметрів уже відомих паралельних асинхронних методів і характеризуються підвищеною швидкістю збіжності за умови незалежності від величини кроку дискретизації. Ідея мультисіткових ітераційних методів викладена з використанням методу дворівневої кореляції, який містить типову послідовність мультисіткових обчислень. Природним узагальненням методу дворівневої кореляції є перехід від двох сіток до використання деякої послідовності сіток, що характеризуються типовим коефіцієнтом

зростання кроку дискретизації $h_i = 2h_{i-1}$, де i – поточний номер сітки. В межах даного методу описані ітераційні схеми згладжування та основні правила побудови операторів пролонгації і рестрикції, що задають порядок переходу між сітками з різними кроками дискретизації. Розглянуто умови збіжності мультисіткового ітераційного методу для лінійних крайових задач та наведено алгоритм його реалізації. Для мультисіткових методів існує фіксований порядок переходів між сітками, який задається видом циклу. Частіше застосовують так звані V -цикли або W -цикли. Ця властивість мультисіткових методів вимагає синхронізації, а тому їх застосування в асинхронних структурах є обмеженим. Ієрархічний сітковий метод об'єднує переваги звичайного мультисіткового методу з асинхронними ітераційними методами, що розглядались у попередньому розділі. При його застосуванні послідовність сіток для обчислення може вибиратися довільним чином. Мультисіткові методи у поєднанні з принципами асинхронності, які були обґрунтовані для ієрархічних методів, успішно застосовані для рекурентних нейронних структур. Такий підхід дозволяє побудувати паралельний алгоритм розв'язування крайової задачі, виходячи безпосередньо з її постановки. В роботі описані алгоритми даного типу для лінійних і нелінійних крайових задач та його модифікації для нестационарних крайових задач, розглянуто особливості застосування в структурах прямого поширення та рекурентних структурах. На завершення наведено приклади розв'язування крайових задач на рекурентних нейронних мережах.

Розділ 1**НЕЙРОННІ МЕРЕЖІ:
ОЗНАЧЕННЯ ТА ОСНОВНІ ВЛАСТИВОСТІ**

Сьогодні є безперечним значний науковий та практичний інтерес до обчислювальних структур нового типу — штучних нейронних мереж. Він спричинений низкою успішних застосувань цієї нової технології, яка дозволила розробити ефективні підходи до вирішення проблем, що вважалися складними для реалізації на традиційних комп'ютерах. На назву “нейронні мережі” зараз претендують усі обчислювальні структури, які в тій чи іншій мірі моделюють роботу мозку. Але таке моделювання, здебільшого, є дуже фрагментарним, і говорити про створення у найближчому майбутньому штучного мозку або навіть деякої його моделі, яка дублювала б роботу мозку найпримітивніших живих створінь, ще зарано. Такий висновок випливає з надзвичайної складності цього загадкового витвору природи.

При побудові моделі мозку розглядають локальні та глобальні аспекти пізнання його функціонування. Основною глобальною характеристикою, яка істотно утруднює моделювання, є надзвичайно велика кількість базових структурних елементів. Мозок людини містить близько сотні мільярдів нейронів, кожен з яких кількома тисячами зв'язків об'єднується з іншими нейронами. Використання навіть найпростішої моделі нейрона не дозволяє побудувати модель мозку, що наближалася б за своїми глобальними показниками до реального об'єкта моделювання. До локальних характеристик слід віднести власне принципи, за якими будують модель нейрона. Останнім часом нейробіологія досягла значних успіхів у вивченні нейрона як елементарної структурної одиниці мозку. Відкрито велику кількість закономірностей, що описують його функціонування та взаємодію з іншими нейронами. Однак, як і раніше,

залишаються без відповіді питання про те, яким чином реалізуються такі властиві мозку основні функції, як пам'ять та свідомість. Отже, залишається до кінця не з'ясованим зв'язок між локальними характеристиками нейрона та глобальними функціями мозку. Але саме такий зв'язок і є основою побудови штучних нейронних мереж, які моделюють функції мозку. Тому основною проблемою концептуального підходу до нейромережного моделювання є вертикальна стратифікація моделі, тобто з'ясування питання про взаємодію елементів на всіх рівнях знизу вгору. Лише шляхом вдалої координації дій великої кількості структурних елементів можливо досягти вияву якісно нової властивості всієї моделі.

Успішний розвиток теорії нейронних мереж за останнє десятиліття дозволив реалізувати ряд таких глобальних властивостей. Найвідомішими з них є навчання, узагальнення та абстрагування.

Властивість навчання проявляється у здатності нейронної мережі змінювати свою поведінку в залежності від стану навколишнього середовища. Завдяки саме цій властивості нейронні мережі привертають до себе значну увагу. Існує велике розмаїття алгоритмів навчання нейронних мереж, кожен з яких має свої сильні та слабкі сторони, але сьогодні ще не сформовано єдиної думки про те, чому можна навчити нейронну мережу і як таке навчання має проводитись.

Властивість узагальнення дає можливість нейронній мережі знижувати чутливість до незначних флуктуацій вхідних сигналів. Ця властивість дуже важлива для об'єктів, які існують у реальному середовищі. Особливістю нейромережного підходу в даному випадку є те, що узагальнення — це результат властивостей структури, а не роботи спеціальної програми на традиційному комп'ютері.

Властивість абстрагування дозволяє створювати на нейронній мережі нову сутність, виходячи з аналізу вхідної інформації. Особливо ця властивість проявляється для задач розпізнавання образів. Завдяки їй нейромережа може створювати деякий

ідеальний образ, керуючись вхідною інформацією, яка має деякі властивості цього образу.

1.1. Історичні аспекти розвитку штучних нейронних мереж

Чи не найдивовижнішою властивістю людського мозку є здатність до самоаналізу. Природа думки цікавить вчених найрізноманітніших напрямків, починаючи з духовних основ буття і закінчуючи анатомією. Але ні висновки філософів, ні результати наукових експериментів не змогли до кінця вирішити цю проблему.

Проте за останні кілька десятиліть зусилля нейробіологів та нейроанатомів увінчалися значними успіхами, головним з яких можна вважати механізм навчання нейрона. Це досягнення дозволило в 1943 році Мак-Каллоку та Піттсу в піонерській роботі [131] сформулювати основні принципи побудови моделі нейрона, яка включала в себе механізм навчання і отримала назву формального нейрона.

Наступним революційним кроком у напрямку розуміння механізму навчання стала робота Доналда Хебба [97], що вийшла в 1949 році. Базуючись на роботах Еудженіо Танці [171] та іспанського нейроанатома Роман-і-Кахаля [156], Д.Хебб сформулював основні положення теорії, яка вказує, що глобальні механізми навчання визначаються набором правил модифікації синапсів. Згідно з цією теорією, міжнейронні синаптичні зв'язки безперервно змінюються в результаті реалізації процесу навчання. Завдяки “тренуванню” відповідних синапсів у ході навчання зростає також вплив одних нейронів на інші. Внаслідок процесів виникнення нових та інтенсифікації старих синапсів виникають нейронні ансамблі, у зв'язках яких закодована корисна інформація.

Незважаючи на те, що і досі не існує обґрунтованого підтвердження цієї теорії, вона відіграла дуже важливу роль у створенні алгоритмів навчання штучних нейронних мереж.

В 1952 році опубліковано ще одну роботу, яку цитують і зараз. Це була книга Ешбі [23], в якій висловлювалась думка про те, що адаптивна поведінка живих створінь зумовлена процесами навчання.

Одержані теоретичні результати стали потужним поштовхом для дослідників штучних нейронних мереж в 50-х – 60-х роках минулого століття. На базі формального нейрона були створені перші штучні нейронні мережі [135,136,161,180] спочатку у вигляді фізичних електронних моделей, а згодом побудовані і комп'ютерні моделі у вигляді алгоритмів. Уже в 1956 році було опубліковано роботи [159,172] по створенню штучних нейронних мереж, навчання яких проводилось відповідно до теорії Хебба. У цих роботах для навчання нейронної мережі використовувались два протилежні процеси — активації та гальмування, які і визначали характер синаптичних зв'язків.

Найбільш вагомим досягненням того періоду було створення штучної нейронної мережі, що отримала назву перцептрона [161]. За допомогою цієї нейромережної структури вдалося вирішити велику кількість задач, серед яких передбачення погоди, обробка зображень та розпізнавання образів. Перші успіхи стали причиною значного інтересу до проблем моделювання штучних нейронних мереж. Оптимісти вважали, що єдиною перешкодою до створення штучного інтелекту залишаються лише технічні труднощі створення достатньо великої нейронної мережі.

У цей період також було запропоновано ряд нових активаційних функцій для моделі формального нейрона. Сигмоїдна функція, яка вперше введена в [61], дозволила значно розширити коло задач, доступних для вирішення на перцептроні. Проте невдовзі з'ясувалось, що одношаровий перцептрон не є універсальним обчислювальним середовищем. Цей факт був сформульований із математичною точністю за допомогою низки теорем [6] і дуже негативно вплинув на подальший розвиток нейронних мереж. Незважаючи на те, що на той час уже були відомі проекти багатшарових перцептронів [179], на які не

розповсюджувалась дія обмежувальних теорем, більшість проектів була згорнута. В 70-ті роки лише поодинокі ентузіасти, такі як Кохонен, Гроссберг, Андерсен, продовжували працювати в цій області [91]. Переборюючи труднощі, пов'язані з поганим фінансуванням та відмовою від публікації робіт, вчені продовжили дослідження багат шарових перцептронів. Виявилось, що обмеження, сформульовані Мінським [6], були надто песимістичними. Незабаром було знайдено просте розв'язання багатьох задач, які вважались такими, що не мають розв'язку на одношаровому перцептроні.

На початку 80-х років інтерес до штучних нейронних мереж почав стрімко зростати. Поштовхом до такого зростання були досягнення, серед яких слід згадати мережі Хопфілда [102] зі спеціально розробленим для них алгоритмом формування вагових синаптичних коефіцієнтів та карти Кохонена [118], які мають властивість самоорганізації. Але чи не найбільшим досягненням, яке викликало великий практичний інтерес, стала публікація алгоритму зворотного розповсюдження [162], що проявив себе як один із найперспективніших при навчанні багат шарових перцептронів. За всю історію розвитку штучних нейронних мереж цей алгоритм пере відкривався декілька разів [152,125], але зараз вважають, що пріоритет належить Вербосу [175], який вперше сформулював його в своїй докторській дисертації в 1974 році.

Як виявилось згодом, алгоритм зворотного розповсюдження також не вільний від проблем, основною з яких є відсутність гарантії того, що мережа може бути навчена за скінченну кількість кроків. При невдалій спробі навчання, як правило, роблять наступну спробу без впевненості в тому, що вона принесе кращі результати. Якщо результат навчання відповідає заданим умовам, то все одно це не гарантує того, що мережа навчена найкращим чином.

Однією з відомих альтернатив для багат шарових перцептронів є РБФ-мережі [46], тобто мережі, які ґрунтуються на радіальних базисних функціях. Вони мають свої специфічні

алгоритми навчання, які в окремих випадках працюють краще, ніж алгоритми зворотного розповсюдження.

Слід відмітити, що переважна більшість штучних нейронних мереж донедавна використовували модель нейрона, яка повністю відповідала або була частково модифікованою моделлю формального нейрона. Ця модель не змінювалась за останні роки настільки істотно, наскільки цього вимагали проблеми, які пов'язані з застосуванням штучних нейронних мереж. Назріла необхідність підвищення адекватності моделі, оскільки процес синаптичної передачі інформації в біологічному нейроні виходить далеко за межі тієї тривіальної схеми, що використовується у формальному нейроні.

Спроби ускладнення внутрішньої структури моделі нейрона були відомі і раніше. В літературі описані моделі, що враховували біологічно важливі параметри нейрона, такі як період рефракції чи часові затримки. Використання цих моделей здебільшого призводило тільки до погіршення загальних параметрів нейромережі. Основною причиною невдач було те, що нейронну мережу з такими нейронами намагалися використати для реалізації уже відомих методів, тому нові властивості моделі були лише баластом, а не працювали на підвищення інтенсивності обробки інформації.

Протягом останніх років значно активізувались дослідження, спрямовані на пошук більш ефективних моделей нейрона. В роботах [83,101] запропоновані моделі, принциповою відмінністю яких є залучення до процесу обробки інформації нових характеристик, що задані внутрішніми змінними. Наступним кроком у згаданому напрямку стала робота [119], в якій крім внутрішніх структур даних використовувались і внутрішні процедури, які реалізують характеристичні функції, призначені для модифікації спільних структур даних прихованого шару.

Сучасний стан даної області свідчить про те, що вона вже цілком довела свою перспективність та інтенсивно розвивається. За допомогою штучних нейронних мереж успішно реалізовано

ряд комерційних проектів, пов'язаних із розпізнаванням зображень та звукової інформації. Але для досягнення більш значних результатів потрібна розробка нових технологій та методів обробки інформації, які базувалися б на принципово нових теоретичних основах.

1.2. Нейрон та його моделювання

1.2.1. Біологічний нейрон

Нейрон, як і всі інші клітини, складається з ядра та цитоплазми (рис.1.1). Тіло клітини (сома) від навколишнього середовища відокремлене тонкою мембраною завтовшки 75 ангстрем, що складається з ліпідів (жироподібних речовин) та характеризується низькою провідністю.

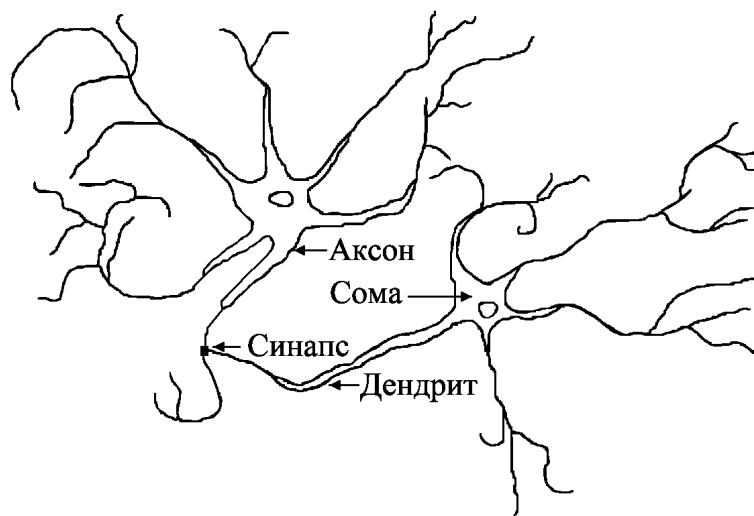


Рис.1.1. Біологічний нейрон.

У мембрану вбудовані білкові молекули, які виконують функції рецепторів та іонних каналів. Для внутрішнього складу клітини характерною є висока концентрація іонів калію та

низька концентрація іонів натрію. Таке співвідношення є результатом роботи натрій-калієвого насоса, на функціонування якого витрачається накопичена клітиною енергія. Спеціальні транспортні білкові молекули іонних каналів захоплюють іони натрію з внутрішньої сторони мембрани та переміщують на зовнішню сторону. Аналогічно іони калію захоплюються на зовнішній стороні мембрани та переміщуються на внутрішню. Результатом роботи такого механізму є різниця потенціалів (приблизно -60мВ) між внутрішньою частиною нейрона та зовнішнім середовищем.

Нейрони характеризуються також наявністю у них спеціальних відростків. По одних відростках нейрони одержують інформацію, а по других передають сигнали іншим нейронам. По деревовидних відростках (дендритах) нейрон отримує інформацію через спеціальні контакти (синапси). Передача інформації від одного нейрона до іншого відбувається шляхом розповсюдження нервового імпульсу вздовж нервового волокна-аксона. Кожен нейрон може мати велику кількість дендритів і тільки один аксон. Волокно аксона розгалужується і створює з'єднання з дендритами інших нейронів через відповідні синаптичні контакти. Під дією сигналів з інших нейронів або рецепторів мембрана аксона змінює свою провідність. Це відбувається, коли внутрішній потенціал перевищує значення близько -50мВ . Механізм розповсюдження нервового імпульсу базується на роботі потенціалозалежних іонних каналів, які створюються за допомогою складних білкових молекул. При деполяризації, тобто зменшенні мембранного потенціалу у бік зменшення від'ємного заряду в середині клітини, відкриваються потенціалозалежні натрієві канали, і натрій по хімічному і електричному градієнту входить у клітину. Тривалість цього стану складає близько 1–3 мс. Після цього натрієві канали починають закриватися і одночасно відкриваються потенціалозалежні калієві канали. Калій по хімічному градієнту виходить із клітини, внаслідок чого збільшується її від'ємний заряд. Процес деполяризації (спайк),

який супроводжується потенціалом дії близько +40мВ, розповсюджується вздовж аксона майже без затухання зі швидкістю від 100 до 1000 сантиметрів у секунду. Після проходження спайка настає період рефрактерності, який характеризується повною пасивністю нейрона протягом 200мс. Таке розповсюдження спайка вздовж аксона часто порівнюють із розповсюдженням зони горіння у бікфордовому шнурі. Відповідно до цієї аналогії зона горіння відповідає спайку, а область згорілої порохової суміші — це стан рефрактерності. Частина аксона, що лежить попереду, відповідає ще не активованим натрієвим каналам.

Як уже згадувалось, взаємодія нейронів відбувається через спеціальні контакти — синапси. Синапси бувають електричні та хімічні. Електричний синапс — це ділянка з'єднання двох нейронів, яка характеризується високою провідністю. Внаслідок цього електричний струм, що протікає в одному нейроні, може протікати і в другому. Такий зв'язок менш інформативний, ніж хімічний, але характеризується високою швидкістю взаємодії. Тому основним його призначенням є синхронізація пула нейронів. Хімічний синапс значно складніший у порівнянні з електричним і складається з пресинаптичної частини однієї клітини та постсинаптичної частини другої клітини, які розділені між собою синаптичною щілиною. Пресинаптична частина складається з декількох бутонів, якими закінчується аксон. На кожному з бутонів розташовані ділянки викиду трансмітера, який знаходиться в бульбашках. Кожна така бульбашка містить один квант трансмітера. Постсинаптична частина — це ділянка мембрани постсинаптичної клітини, яка містить білки, що є рецепторами трансмітера. Характерною особливістю пресинаптичної частини є наявність потенціалозалежних кальцієвих каналів. Спайк, досягаючи кінця аксона, відкриває кальцієві канали, в результаті чого кальцій з навколишнього середовища входить в аксон і запускає низку кальцій залежних біохімічних реакцій. Результатом цих реакцій є вилив у синаптичну частину певної кількості квантів

трансмітера з певних ділянок бутона. Кожен квант трансмітера містить білок ацетилхолін (AcChoR), який використовується нейроном власне для передачі інформації. Нікотинний рецептор ацетилхоліна — це молекула, що складається з п'яти гомологічних пептидів. Потрібно відмітити надзвичайно високу стійкість цього білка та механізму його дії в ході еволюції. На думку багатьох дослідників [1,164], основою розподіленої асоціативної пам'яті — характерної властивості біологічних нейронних мереж, є саме ці механізми.

Ще одна важлива властивість нейрона — пластичність. Під пластичністю розуміють зміни в нейронних мережах та окремих нейронах, тривалість яких значно перевищує час передачі інформації. Найважливішими формами пластичності є звикання (негативне навчання) та класичний умовний рефлекс (асоціативне навчання). Механізм звикання для нейронів-детекторів зумовлений тим, що, крім сигналів від специфічних для них подразників, вони сприймають також сигнали неспецифічного збудження від нейронів новизни. Компонента неспецифічного збудження з часом згасає, що призводить до зменшення збудження детектора. Цей ефект отримав назву часткового звикання. Він проявляється на нейронних мережах у безлічі різновидів завдяки різній спеціалізації нейронів. Пресинаптичний механізм пластичності при звиканні пов'язаний зі зниженням активності викиду трансмітера з бутонів аксона. Причиною зниження цієї активності є зменшення кількості працюючих кальцієвих каналів. В результаті зменшується кальцієвий струм, що входить у пресинаптичну частину синапса. Зменшення щільності іонів кальцію в пресинаптичній зоні призводить до того, що хімічний синапс перестає реагувати на спайки. Ще однією формою пластичності нейрона є асоціативне навчання, що проявляється через умовний рефлекс. Виникнення умовного рефлексу зумовлене встановленням зв'язку між умовним та безумовним подразником. Він базується на внутрішньоклітинному асоціативному процесі, який модифікує ділянку постсинаптичної мембрани нейрона таким чином, що

вона стає селективно більш чутливою до відповідного типу подразників. Крім прямої передачі інформації від нейрона до нейрона виявлені спеціальні нейронні механізми керування ефективністю дії синаптичних зв'язків у вигляді модулюючих нейронів, так званих “синапсів на синапсах”. До останніх відкриттів нейробіології також відносять відкриття явища “парасинаптичної передачі інформації”. У цьому випадку нейрон реалізує взаємодію з іншими нейронами шляхом послідовного викиду в міжклітинне середовище або русло крові короткої послідовності амінокислот-нейропептидів. При цьому взаємодія відбувається без участі синаптичних зв'язків. Переданий таким чином сигнал сприймається тими нейронами, які мають відповідні рецепторні білки.

1.2.2. Формальний нейрон

Історично першою публікацією, що заклала підвалини для створення штучних нейронів та нейронних мереж, вважають роботу Уоррена С. Мак-Каллока та Вальтера Пітса [131]. У цій роботі було започатковано теорію, в основі якої лежав той факт, що всі аспекти нервової діяльності можна моделювати за допомогою мережі елементів, які мають два стійкі стани. Модель такого елемента, яка показана на рис.1.2, отримала назву формального нейрона.

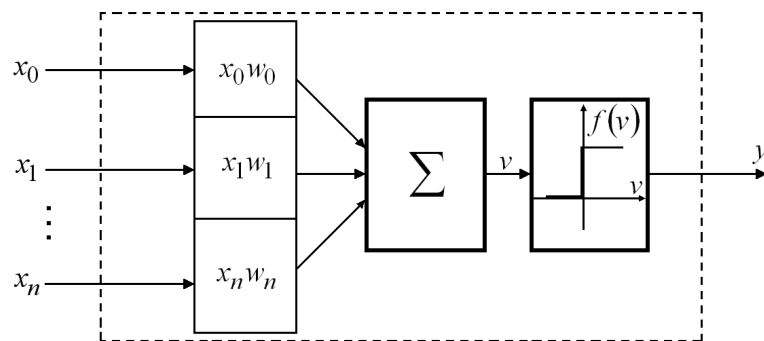


Рис.1.2. Формальний нейрон.

Моделювання функцій синапсів відбувається шляхом масштабування вхідних сигналів (x_0, x_1, \dots, x_n) за допомогою вагових коефіцієнтів (w_0, w_1, \dots, w_n) . Одержані сигнали подаються на вхід суматора, який виконує їх обробку за формулою:

$$v = \sum_{i=0}^n w_i x_i . \quad (1.1)$$

Змінна v є аргументом порогової активаційної функції:

$$y = f(v) . \quad (1.2)$$

Активаційна функція, що вперше була запропонована в [131], має вигляд:

$$y = \begin{cases} 1 & \text{при } v \leq 0, \\ 0 & \text{при } v > 0. \end{cases} \quad (1.3)$$

У сучасній літературі зустрічається велика кількість парадигм штучних нейронних мереж, елементи яких реалізують різні активаційні функції. Найбільш поширеною є сигмоїдальна функція, що може бути представлена в дискретному та аналоговому варіанті.

Прикладом найпростішої активаційної функції є дискретний сигмоїд із параметром, що має назву функції Хевісайда або тета-функції:

$$y = \begin{cases} 1 & \text{при } v \leq a, \\ 0 & \text{при } v > a. \end{cases} \quad (1.4)$$

Якщо величина збудження v нейрона з активаційною функцією Хевісайда не перевищує значення параметра a , то нейрон залишається пасивним, а при перевищенні порога видає фіксоване значення функції, що приймається за логічну одиницю.

Найпоширенішою нелінійною аналоговою активаційною функцією є сигмоїдальна логічна функція з параметрами, b c і d , яку задають виразом:

$$y = \frac{b}{c + e^{dv}}. \quad (1.5)$$

При одиничних значеннях параметрів $b=1$, $c=1$, $d=-1$ одержуємо:

$$y = \frac{1}{1 + e^{-v}}. \quad (1.6)$$

Амплітуда вихідного сигналу нейрона з даною активаційною функцією залежить від амплітуди вхідних сигналів. З рис.1.3 видно, що крутизна S-подібної функції залишається високою тільки у деякому визначеному діапазоні.

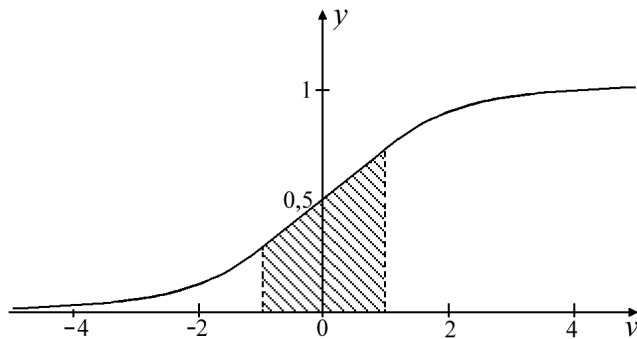


Рис.1.3. Сигмоїдальна логічна функція.

Ця властивість була вперше використана Гроссбергом у [90] для вирішення поставленої ним дилеми шумового насичення. Завдяки нелінійному коефіцієнту підсилення нейрон може надійно працювати в широкому діапазоні рівнів вхідних сигналів. Ще однією корисною властивістю аналогової сигмоїдальної активаційної функції є диференційованість на всій осі абсцис та простий вираз для похідної

$$y' = y(1 - y),$$

де $y = \frac{1}{1 + e^{-v}}$.

Подібною до функції Хевісайда є функція гіперболічного тангенса (рис.1.4).

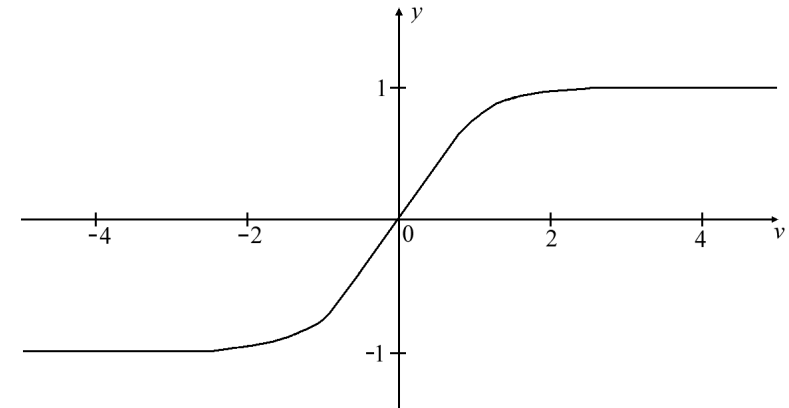


Рис.1.4. Функція гіперболічного тангенса.

Основна відмінність полягає у симетрії відносно початку координат, що дозволяє працювати у від'ємному діапазоні вхідних сигналів.

Гіперболічну функцію задано виразом, що містить коефіцієнти k, m , які призначені для регулювання параметрів вихідного сигналу нейрона.

$$y = k \operatorname{th}(mv) \quad (1.7)$$

При $k=1, m=1$ одержимо варіант активаційної функції

$$y = \operatorname{th}(v),$$

що має найбільше поширення.

Подібно до сигмоїдальної функції, в дискретному та аналоговому варіанті існує дзвонovidна активаційна функція. Дискретний варіант цієї активаційної функції з порогоми k, m при $m > k$ задано виразом (1.8), а її графічне зображення показано на рис.1.5.

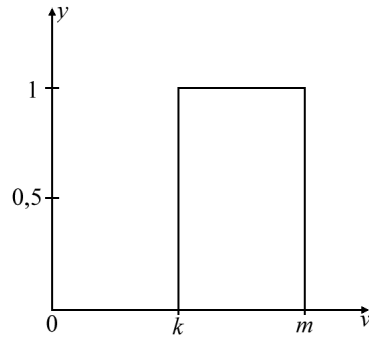


Рис.1.5. Дискретна дзвонovidна функція.

$$y = \begin{cases} 1 & \text{при } v > k, v \leq m, \\ 0 & \text{при } v \leq 0, v > m. \end{cases} \quad (1.8)$$

Аналогова дзвонovidна функція використовується у випадку аналогового моделювання нейронів із властивостями смугового фільтра. Розрізняють дзвонovidні експоненціальні функції першого та другого порядку.

Аналогову дзвонovidну функцію першого порядку у загальному випадку задають виразом, що містить параметри k, m, n, p, r :

$$y = \frac{1}{ke^{mv} + ne^{pv} + r}. \quad (1.9)$$

Задавши $k = 1, m = 1, n = 1, p = -1, r = -1$, одержимо базовий варіант дзвонovidної функції першого порядку, графік якої представлений на рис.1.6:

$$y = \frac{1}{e^v + e^{-v} - 1}. \quad (1.10)$$

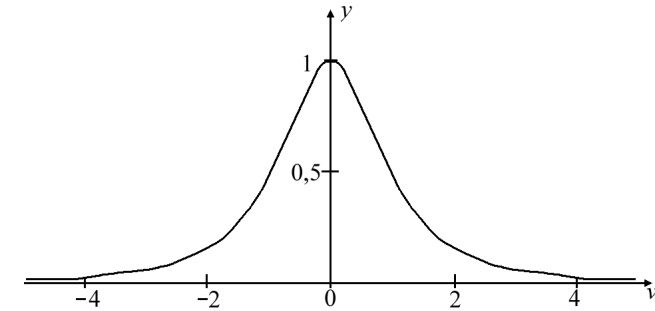


Рис.1.6. Аналогова дзвонovidна функція першого порядку.

Дзвонovidну функцію другого порядку задають виразом $Y = e^{-v^2}$. Як видно з рис.1.7, її графік нагадує дзвонovidну функцію першого порядку, але має більшу крутизну, тому може використовуватися в смугових фільтрах із жорсткішими характеристиками.

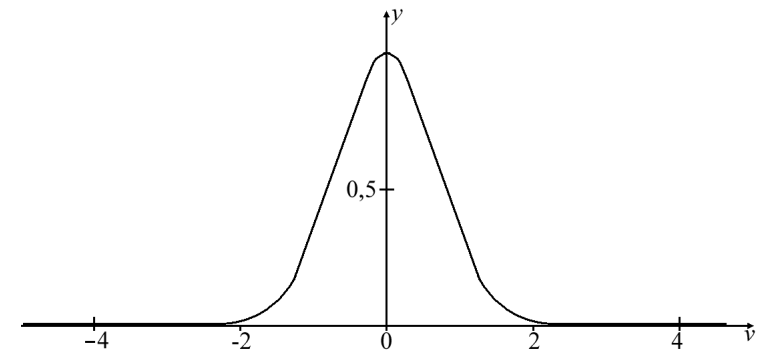


Рис.1.7. Аналогова дзвонovidна функція другого порядку.

1.2.3. Аналогова та дискретна реалізації формального нейрона

Як можна було помітити з опису активаційних функцій, модель формального нейрона допускає дискретний або аналоговий їх вид. Вибір того чи іншого підходу залежить від кола задач, для вирішення яких призначається штучна нейронна мережа. У випадку аналогового моделювання аналоговий нейрон може бути реалізований на операційному підсилювачі з суматором, який розглядається в теорії аналогових обчислювальних машин (АОМ), оскільки між ними існує пряма аналогія. На рис.1.8 зображена функціональна схема інвертуючого суматора.

Його робота описується наступними залежностями:

$$\frac{1}{R} = \frac{1}{R_{зв}} + \sum_{i=0}^n \frac{1}{R_i}, \quad U_{вих} = \sum_{i=0}^n K_i U_{вхд_i}, \quad (1.11)$$

де $K_i = \frac{R_{зв}}{R_i}$ — коефіцієнт передачі i -го вхідного сигналу.

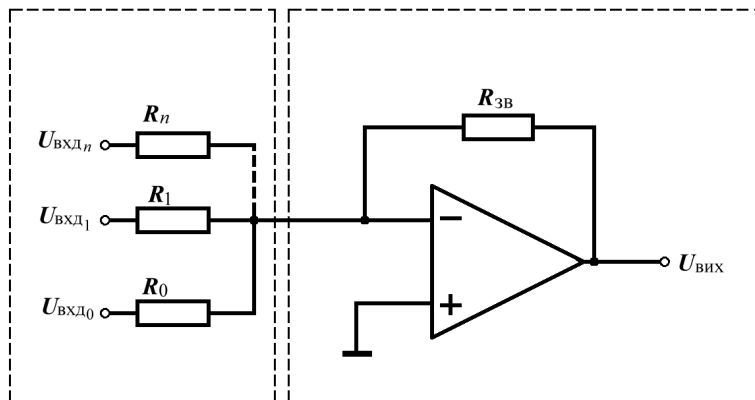


Рис.1.8 Інвертуючий суматор.

Порівнявши залежності (1.1) та (1.11), можна впевнитись у прямій аналогії функції додавання вхідних сигналів на моделі формального нейрона та операційного підсилювача за умови, що вектор вхідних сигналів (x_0, x_1, \dots, x_n) визначається вектором $(U_{вхд_0}, U_{вхд_1}, \dots, U_{вхд_n})$, а вектор вагових коефіцієнтів (w_0, w_1, \dots, w_n) відповідає вектору коефіцієнтів передачі (K_0, K_1, \dots, K_n) . Активаційну функцію (1.2) у даному випадку задають передатною функцією підсилювача. Для схеми, що зображена на рис.1.8, вона має вигляд:

$$U_{вих} = \frac{R_{зв}}{\prod_{i=0}^n R_i / \sum_{i=0}^n R_i} U_{вхд}. \quad (1.12)$$

Таким чином, проста нейронна мережа може бути реалізована на АОМ методом прямої аналогії. При цьому процес навчання нейрона зводиться до настройки параметрів суматора. Подібність також підтверджується спільними перевагами та недоліками, які характерні для АОМ та аналогової моделі нейрона.

У випадку дискретного моделювання інформаційним значенням сигналу є біт, а не значення амплітуди. Тому для підвищення рівня інформативності міжнейронного обміну можуть бути використані багаторозрядні шини передачі даних, а обробку даних виконують сигнальні процесори, що реалізуються у вигляді нейрочіпів. Перехід до дискретної моделі нейрона є також природним, оскільки дозволяє використати всі її переваги. Перш за все — це точність представлення інформації, яка визначається кількістю розрядів цифрового нейрона. Інформація у даному випадку може бути структурованою, і організація її може досягати високого ступеня складності. Виникає також можливість використання складних алгоритмів, що моделюють внутрішньоклітинну обробку інформації та функції синапсів.

На сьогодні вже існує велика кількість нейронних обчислювальних структур, які по суті є мультипроцесорними

системами [7] з можливістю паралельної обробки інформації. Це дозволяє організувати на їх основі нейрообчислювальні системи, що складаються з двох основних частин:

- керуюча ЕОМ, реалізована найчастіше на базі персонального комп'ютера;
- плата нейрообчислювача, що вставляється в слот персонального комп'ютера.

Типова структура такого нейрообчислювача показана на рис.1.9.



Рис.1.9. Узагальнена функціональна схема нейрообчислювального пристрою.

Особливістю апаратної реалізації обчислювача з можливістю паралельної обробки інформації є те, що в основі його архітектури лежить матриця сигнальних процесорів, міжпроцесорні зв'язки якої відповідають структурі модельованої нейронної мережі. Як правило, такі нейрообчислювачі мають можливість модульного нарощування кількості сигнальних процесорів з метою збільшення обчислювальних потужностей.

Оскільки взаємодія із зовнішнім середовищем відбувається через керуючий ПК, то базовою платформою для їх реалізації вибирають модулі з зовнішнім інтерфейсом ISA, PCI або VME.

Основними функціональними елементами є матриця сигнальних процесорів (МСП), оперативний запам'ятовуючий пристрій (ОЗП), постійний запам'ятовуючий пристрій (ПЗП) мікропрограм, блоки керування та вводу-виводу інформації. Для побудови нейрообчислювачів такого типу найбільш перспективним є використання сигнальних процесорів із плаваючою крапкою: ADSP2106x, TMS320C4x,8x, DSP96002. Існують також широко відомі нейрочіпи MA16 фірми Siemens, ETAN80170NX фірми Intel та ряд інших.

Описані підходи швидше можна віднести до дискретного моделювання аналогової нейронної структури, оскільки робота їх базується на оцифрованій моделі того ж таки формального нейрона. Сучасні моделі нейрона використовують переваги цифрової обробки даних, починаючи з ранніх етапів проектування [10], що дозволяє, на думку авторів, істотно розширити сферу застосування нейронних мереж шляхом адаптації відповідних алгоритмів паралельної обробки інформації.

1.3. Задачі штучних нейронних мереж

Задачі, які вирішують ШНМ, зводяться до апроксимації багатовимірних функцій, тобто побудови відображення $F: x \rightarrow y$. В залежності від вигляду активаційної функції формального нейрона дане відображення описує один з типів задач. У випадку застосування порогових активаційних функцій вихідні сигнали ШНМ мають дискретний характер, а задачі, що при цьому можуть бути розв'язні, називають задачами *класифікації*. При застосуванні формальних нейронів із сигмоїдальною або іншою неперервною активаційною функцією на ШНМ доцільно розв'язувати задачі *регресії*. Такий поділ є до деякої міри умовним, оскільки задачі класифікації можуть використовувати також неперервні функції активації, значення яких трактуються як імовірності приналежності до відповідних класів.

1.3.1. Проблема представлення неперервних функцій

Для того, щоб відповісти на питання про універсальність ШНМ, необхідно було вирішити проблему представлення функції від n змінних за допомогою операцій додавання, множення та суперпозиції функцій від m змінних при $m < n$. Ця задача впливає з так званої 13-ї проблеми Гілберта, що була відома вже на початку минулого століття. Розв'язком її можна вважати теорему Колмогорова [2,3].

Теорема Колмогорова

Кожна неперервна функція від n змінних, що задана в одиничному кубі n -вимірного простору, може бути представлена у вигляді:

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} h_q \left[\sum_{p=1}^n \varphi_q^p(x_p) \right], \quad (1.13)$$

де функції $h_q(u)$ — неперервні, а $\varphi_q^p(x_p)$ — стандартні функції, які не залежать від вибору функції f .

На жаль, практичне застосування цієї теореми стикається з проблемами, які полягають у тому, що функції $\varphi_q^p(x_p)$ не гладкі, і обчислення їх викликає значні труднощі, крім того, не існує ефективних методик підбору функцій $h_q(u)$ для даної функції f .

Згадана теорема Колмогорова була застосована Хехт-Нільсеном у 1987 році для нейронних мереж прямого поширення з повними зв'язками. В його праці [98] вказується, що довільна неперервна функція $f(x_1, x_2, \dots, x_n)$ може бути реалізована на двошаровому перцептроні за формулою (1.13), для чого необхідно задати $2n+1$ обмежених активаційних

функцій $h_q(u)$ прихованого шару і підібрати n активаційних функцій $\varphi_q^p(x_p)$ нейронів вихідного шару.

Незважаючи на складність практичного застосування, теорема Колмогорова відіграє виключно важливу роль у теорії нейронних мереж. Завдяки їй існує математично точне обґрунтування універсальності нейроподібних структур, які теоретично здатні до реалізації довільної багатовимірної функції шляхом представлення цієї функції за допомогою більш простих функцій, що можуть бути реалізованими на елементах нейронної мережі.

Для одержання практичних результатів необхідно знизити вимоги до точності представлення функції і перейти до задач апроксимації. Крім наближеного значення функції потрібно також зняти обмеження на кількість елементів нейронної мережі. Теорема, яка доводить універсальність нейронних структур як засобів апроксимації довільних функцій, була майже одночасно відкрита кількома авторами [106,63,81].

Теорема про універсальність нейронних мереж

Для кожного як завгодно малого $\varepsilon > 0$ існує таке число H та непусті множини чисел:

$$\left\{ \{w_{ij}\}_{j=1}^n \right\}_{i=1}^H, \{v_i\}_{i=1}^H \text{ і } \{u_i\}_{i=1}^H,$$

при яких функція

$$\phi(x_1, x_2, \dots, x_n) = \sum_{i=1}^H v_i \sigma(w_{i1}x_1 + w_{i2}x_2 + \dots + w_{in}x_n + u_i)$$

апроксимує визначену на обмеженій множині неперервну функцію $f(x_1, x_2, \dots, x_n)$ таким чином, що

$$|f(x_1, x_2, \dots, x_n) - \phi(x_1, x_2, \dots, x_n)| < \varepsilon.$$

Ця теорема є ключовою при опрацюванні методик формування апроксимаційних функцій. Такі методики у

більшості випадків залежать від архітектури нейронної мережі, на якій вони мають бути реалізовані, та від алгоритмів її навчання.

1.3.2. Проблема класифікації та категоризації

Використання дискретної активаційної функції призводить до поділу простору вихідних сигналів на деяку множину класів. Завданням нейронної структури у цьому випадку є встановлення однозначної відповідності між векторами вхідних сигналів і даними класами. Тип задач по встановленню такої відповідності називають задачами класифікації або категоризації.

Поняття класу не завжди чітко задане і визначається з контексту розв'язуваних проблем. Але у більшості випадків під терміном *клас* розуміють сукупність об'єктів, які згруповані за деякими ознаками та правилами. Власне кажучи, завдання класифікації полягає в застосуванні до векторів вхідних сигналів встановлених правил виявлення сукупності ознак та прийняття рішення щодо віднесення кожного конкретного вектора до певного класу.

У випадку, коли допускається динамічна зміна кількості класів та їх ознак, одержуємо задачу категоризації. Категорія, подібно до класу, є сукупність об'єктів, що згруповані за деякими ознаками. Різниця тільки у тому, що категорія, у порівнянні з класом, не має статично визначеної сукупності ознак. Ці ознаки, як і кількість самих категорій, можуть змінюватися в процесі вирішення задачі категоризації. Формування ознак відбувається шляхом проб і помилок із подальшим відкиданням неістотних параметрів і підсиленням тих, які трапляються частіше.

Якщо задача категоризації дає можливість динамічної зміни множини ознак, то правила формування цих ознак залишаються незмінними і визначаються моделлю нейрона. Тому актуальним є питання, чи може бути та чи інша нейронна структура універсальним класифікатором. Іншими словами, чи може весь

допустимий набір векторів вхідних сигналів бути однозначно розподілений між заданими класами. Відповіддю на це питання є вирішення проблеми лінійної роздільності.

Задача класифікації зводиться до розподілу векторів вхідних сигналів, що належать до n -вимірного гіперпростору, на деяку кількість класів. Це відбувається шляхом поділу гіперпростору відповідною кількістю гіперплощин. У випадку n -входового перцептрона

$$\sum_{i=1}^n w_{ik} x_i = T_k \text{ при } k = 1, 2, \dots, m,$$

де n — число входів, а m — число виходів перцептрона.

Розглянемо поняття лінійної роздільності на прикладі двовходового формального нейрона з активаційною функцією Хевісайда. У даному випадку значення аргументу v активаційної функції визначається за формулою $v = w_0 x_0 + w_1 x_1$, тому гіперплощина поділу змінюється на пряму в двовимірному просторі. На рис.1.10 графічно показані можливі варіанти лінійної роздільності для функціонально повної системи логічних функцій.

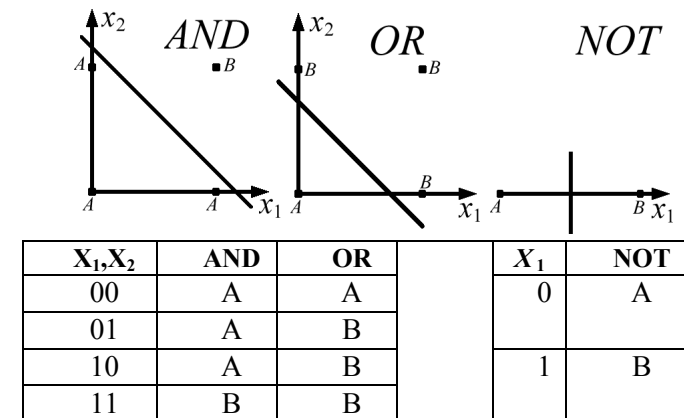


Рис.1.10. Лінійна роздільність функціонально повної системи (AND,OR,NOT).

Але не всі логічні функції лінійно роздільні. Найвідомішим представником лінійно нероздільних функцій є “XOR”. На прикладі цієї функції було вперше наочно показано факт обмеженості можливостей моделі формального нейрона.

З рис.1.11 видно, що для функції “XOR” від двох змінних неможливо провести пряму, яка б однозначно розділила на два класи множину значень цієї функції.

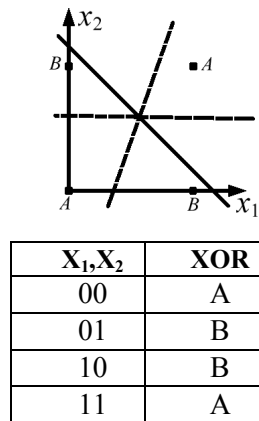


Рис.1.11. Лінійна нероздільність функції “XOR”.

Отже, для того, щоб логічна функція могла бути реалізованою на формальному нейроні, вона повинна бути лінійно роздільною. Факт означення лінійної роздільності логічної функції не є тривіальним у випадку її багатовимірності, коли число можливих варіантів достатньо велике.

Оскільки функціонально повна система логічних функцій є лінійно роздільною, то ансамбль формальних нейронів може реалізувати ДНФ довільної функції. Отже, проблема лінійної роздільності може бути подолана шляхом створення відповідних архітектур нейронних мереж.

1.4. Архітектура штучних нейронних мереж

Що ж являє собою штучна нейронна мережа (ШНМ)? На жаль, сьогодні не існує такого загально визнаного означення, яке б задовольняло всіх. Причиною є той факт, що проблемою нейронних мереж займаються спеціалісти в різних галузях науки, і взаємному розумінню заважають методологічні та термінологічні бар'єри. Якщо розглядати штучну нейронну мережу як деяке середовище для обробки інформації, тоді її можна задати шляхом визначення елементів даного середовища та правил їх взаємодії.

В цьому випадку говорять, що штучна нейронна мережа є структурою, яка складається з великої кількості процесорних елементів, кожен з яких має локальну пам'ять і може взаємодіяти з іншими процесорними елементами за допомогою комунікаційних каналів з метою передачі даних, що можуть бути інтерпретовані довільним чином. Процесорні елементи незалежно в часі обробляють локальні дані, що поступають до них через входні канали. Зміна параметрів алгоритмів такої обробки залежить тільки від характеристик даних.

Іншими словами, штучні нейронні мережі – це обчислювальні парадигми, які реалізують спрощені моделі біологічних нейронних мереж (БНМ). Під БНМ будемо розуміти локальні ансамблі нейронів, які об'єднані синаптичними зв'язками. Сукупність таких ансамблів формує мозок із його різноманітними функціональних можливостями.

Сьогодні відома велика кількість нейронних структур та їх модифікацій, що орієнтовані на вирішення конкретного типу задач. Найбільш відомі типи таких структур показані на рис.1.12.

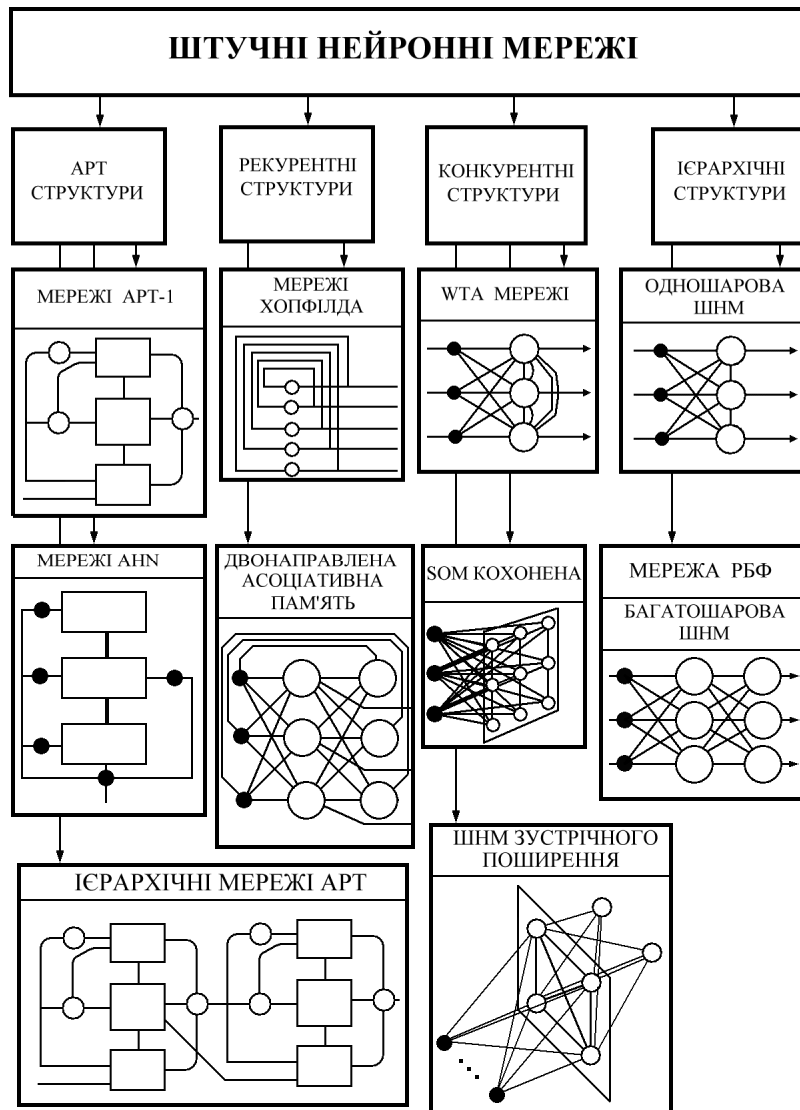


Рис.1.12. Нейронні структури.

Розглянемо детальніше основні властивості ШНМ:

- локальна обробка інформації в штучному нейроні, який є базовою структурною одиницею мережі;
- паралелізм, результатом якого є вирішення глобальної задачі шляхом представлення її у вигляді множини локальних задач, що тісно взаємодіють між собою;
- здатність до навчання, яке підвищує ефективність роботи мережі;
- здатність до розподіленого зберігання знань, які були одержані в ході навчання.

ШНМ задають у вигляді направлених графів, вершинами яких є нейрони, а ребрами позначені міжнейронні зв'язки.

Архітектури сучасних нейронних мереж найчастіше поділяють на три категорії:

- мережі з повним набором міжнейронних зв'язків;
- мережі з фіксованим індексом оточення;
- мережі з пошаровою структурою.

У ШНМ із повним набором міжнейронних зв'язків забезпечується можливість взаємодії кожного нейрона мережі з будь-яким іншим. На рис.1.13 наведений приклад повного з'єднання чотирьох нейронів.

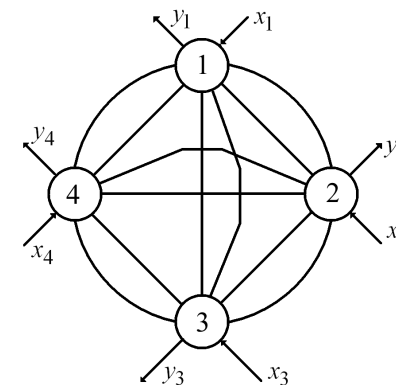


Рис.1.13. Повне з'єднання.

Структури з повним з'єднанням є узагальненими структурами, оскільки всі інші довільні об'єднання нейронів можуть розглядатися як підмножини даної структури. Тому ШНМ із повним з'єднанням є універсальним середовищем для реалізації мережних алгоритмів. Широке використання таких структур обмежується недоліком, який полягає в значному зростанні кількості міжнейронних зв'язків при збільшенні кількості нейронів.

У випадку, коли необхідно використовувати структури з великою кількістю нейронів, застосовують кліткові структури з фіксованим індексом оточення. На рис.1.14 наведений приклад структури такого типу з індексом оточення 4.

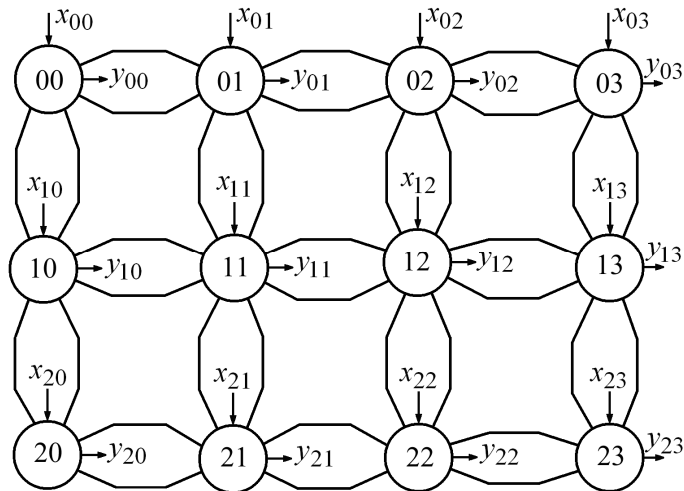


Рис.1.14. Кліткова нейронна мережа з індексом оточення 4.

Ще однією важливою характеристикою нейронних мереж із постійним індексом оточення є модульність. Потужність структури у цьому випадку може нарощуватись простим додаванням елементів без зміни ідеології алгоритму, що на ній працює.

Найбільш поширеними є структуровані за шарами нейронні мережі, які, в залежності від свого функціонального призначення, можуть містити однотипні або різнотипні нейрони. Виходячи з шарової структури ШНМ, характер міжнейронних зв'язків має свої міжшарові та внутрішньшарові особливості. На рис.1.15 показана класифікація видів міжнейронних зв'язків.

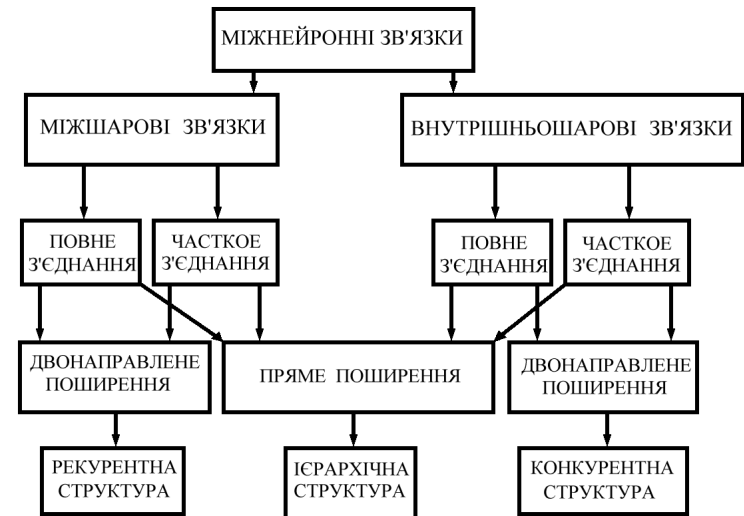


Рис.1.15. Види міжнейронних зв'язків.

Нейронні структури з повним з'єднанням можуть бути як одношаровими, так і багатшаровими. У одношаровій структурі з повним з'єднанням всі вхідні сигнали можуть поступати на всі

нейрони. Класичною структурою даного типу є перцептрон Розенблатта.

Багатошарові мережі з повним з'єднанням забезпечують можливість передачі інформації з кожного нейрона попереднього шару на будь-який нейрон наступного. Найчастіше це — багатошарові перцептрони.

У випадку прямого міжшарового поширення інформація передається однонаправлено у напрямку зростання номера шару.

Пряме поширення в межах одного шару використовують у випадку, коли група нейронів даного шару з'єднана з попереднім шаром опосередковано через виділені нейрони.

Двонаправлене поширення допускає також зворотну передачу, що дозволяє створювати алгоритми, за якими враховувався б взаємний міжнейронний обмін.

Таким чином, однонаправленість зв'язків призводить до побудови виключно ієрархічних структур, у яких обробка інформації розподіляється по рівнях. За кожний рівень ієрархічної обробки інформації відповідає свій шар нейронів. Вихідна інформація більш високого рівня попереднього шару є вхідною для нейронів наступного шару, який забезпечує глибший рівень обробки.

Двонаправлені міжшарові зв'язки необхідні для реалізації рекурентних структур, які дають можливість застосування ітераційних алгоритмів. Загальною рисою таких структур є те, що подальша передача інформації відбувається тільки у випадку завершення ітераційного процесу.

Двонаправлені зв'язки у межах одного шару використовуються для створення конкуруючих груп нейронів. При активації сигналом з попереднього шару кожен з нейронів передає сигнал активації нейронам своєї групи та сигнал гальмування всім іншим нейронам. В результаті конкурентоздатною стає та група нейронів, що одержала найбільше збудження.

1.4.1. Ієрархічні структури

1.4.1.1. Одношарові нейронні мережі прямого поширення

Першою штучною нейронною мережею, яку за сучасною термінологією відносять до одношарових, був перцептрон Розенблатта (рис.1.16).

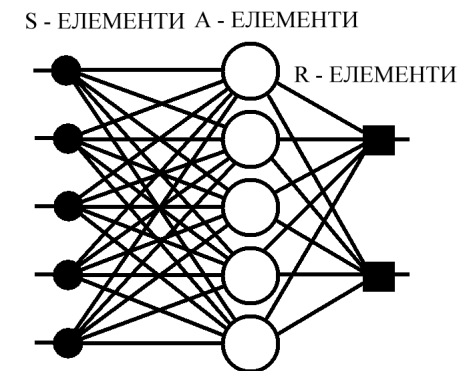


Рис.1.16. Перцептрон Розенблатта.

Оскільки перцептрон розглядався автором як модель роботи мозку, то елементи його структури відповідали елементам простої рефлекторної мережі. S-елементи моделюють роботу сенсорних клітин, які збирають інформацію про навколишнє середовище та передають її до A-елементів, які насправді і є формальними нейронами з пороговою активаційною функцією.

Для формування реакції перцептрона призначені R-елементи, що мають фіксовані вхідні ваги для визначення внеску кожного A-елемента.

Сучасним прототипом перцептрона Розенблатта є одношарова нейронна мережа прямого поширення, яку показано на рис.1.17.

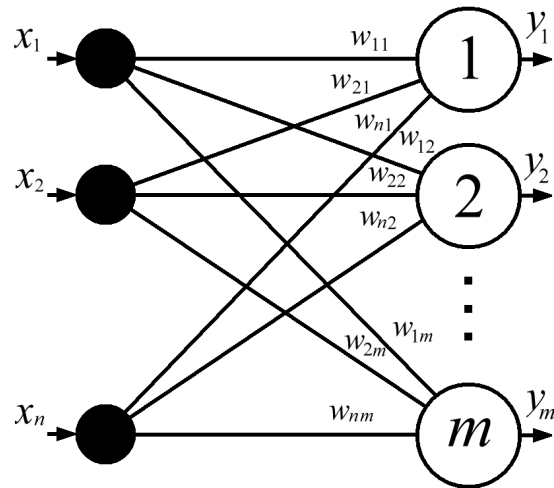


Рис.1.17. Одношарова нейронна мережа.

Вона складається з m нейронів, які одночасно здатні прийняти вхідний вектор сигналів $X = (x_1, \dots, x_i, \dots, x_n)$. Для розмноження елементів x_i цього вектора використовують спеціальні пристрої, які показані зліва від нейронів.

Ці пристрої не виконують обробки інформації, тому не вважаються шаром нейронної мережі. Згідно з моделлю формального нейрона, кожен з його вхідних сигналів множать на ваговий коефіцієнт w_{ij} , де i — поточний номер елемента вектора X , а j — поточний номер нейрона.

Всі вагові коефіцієнти одношарової нейронної мережі утворюють матрицю вагових коефіцієнтів:

$$W = \begin{pmatrix} w_{11} & \dots & w_{1j} & \dots & w_{1m} \\ \cdot & \dots & \cdot & \dots & \cdot \\ w_{i1} & \dots & w_{ij} & \dots & w_{im} \\ \cdot & \dots & \cdot & \dots & \cdot \\ w_{n1} & \cdot & w_{nj} & \cdot & w_{nm} \end{pmatrix}.$$

Тоді вектор аргументів визначається як добуток $V = XW$, а вектор вихідних сигналів є вектором значень активаційних функцій:

$$Y = F(V) = \begin{cases} f_1(v_1), \\ \dots \\ f_j(v_j), \\ \dots \\ f_m(v_m). \end{cases} \quad (1.14)$$

Згадані функції можуть бути однаковими для всіх нейронів, що складають мережу. У цьому випадку її називають *гомогенною*. Мережу, в якій вигляд активаційної функції залежить від номера нейрона, називають *гетерогенною*.

На рис.1.17 показаний загальний випадок одношарової нейронної мережі, у якій кожен нейрон приймає участь в обробці всіх елементів вхідного вектора даних. Такий підхід не завжди є економічно та технічно виправданим. Тому для вирішення конкретних задач можуть використовуватись архітектури зі структурою зв'язків, яка є підмножиною повної зв'язності.

1.4.1.2. Багатшарові нейронні мережі прямого поширення

Спроби застосувати одношарові нейронні мережі для розв'язування широкого кола задач наштовхнулися на ряд труднощів, пов'язаних з проблемою лінійної роздільності. Природним вирішенням цієї проблеми стало застосування багатшарових ШНМ, що нагадують багатшарові структури

мозку. Розглянемо ієрархічну структуру, в якій нейрони структуровані за шарами. Вона складається з m нейронів першого (прихованого) шару, які одночасно здатні прийняти вхідний вектор сигналів $X = (x_1, \dots, x_i, \dots, x_n)$, та k нейронів другого (вихідного) шару. Така структура зображена на рис.1.18 і є узагальненою структурою багатозарової нейронної мережі прямого поширення.

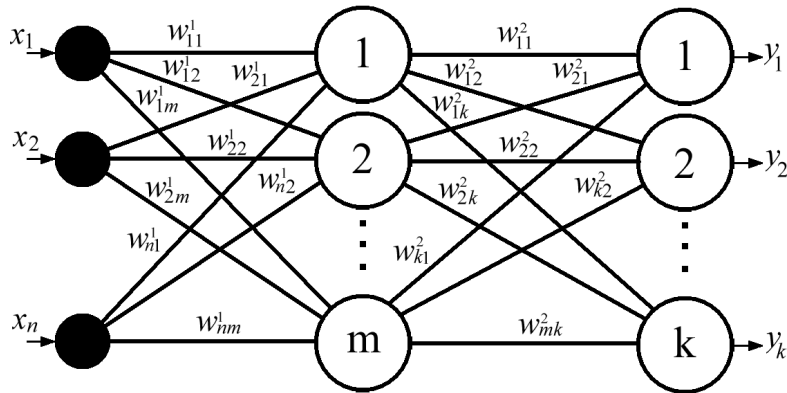


Рис.1.18. Багатозарова нейронна мережа прямого поширення.

У нейронних мережах прямого поширення синаптичні зв'язки організовані таким чином, що кожний нейрон даного рівня ієрархії сприймає інформацію тільки від деякої непустої множини нейронів, які розташовані на більш низькому рівні. Назва мереж вказує на те, що у них існує виділений напрям поширення сигналів, які рухаються, починаючи з входу, через один або декілька прихованих шарів до вихідного шару.

Легко помітити, що багатозарова нейронна мережа може бути одержана шляхом каскадного об'єднання одношарових мереж з матрицями вагових коефіцієнтів W^1, W^2, \dots, W^p , де p — кількість шарів нейронної мережі.

У випадку лінійності активаційних функцій багатозарова нейронна мережа може бути зведена до еквівалентної одношарової з матрицею вагових коефіцієнтів

$W = W^1 * W^2 * \dots * W^p$, тому формування подібних структур має сенс тільки у випадку застосування у нейронах нелінійних активаційних функцій.

1.4.1.3. РБФ-мережі

Штучні нейронні мережі, які використовують радіальні базисні функції, називають РБФ-мережами. Вони є окремим випадком двошарових нейронних мереж прямого поширення, в яких прихований шар нейронів використовує радіальні базисні функції типу гаусової як активаційні.

В просторі вхідних векторів вибирають вектор, який називають центром, і відповідно до нього задають вагові коефіцієнти прихованого шару. Аргумент активаційної функції v_j для нейрона j прихованого шару визначатиметься відстанню між вхідним вектором та вектором прихованого шару:

$$v_j = \|X - W_j\| = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2}. \quad (1.15)$$

Тоді значення вихідного сигналу цього нейрона дорівнюватиме:

$$y_i = \exp\left(-\left(\frac{v_j}{\sigma_j}\right)^2\right) = \exp\left(-\left(\frac{\|X - W_j\|}{\sigma_j}\right)^2\right). \quad (1.16)$$

Процес настройки мережі, крім вибору кількості центральних векторів та їх координат, також включає вибір параметра σ_j , який є індивідуальним для кожного нейрона і задає крутизну активаційної функції.

Отже, прихований шар формує сукупність функцій, що утворюють базисну систему, а на вихідному шарі формується лінійна комбінація цих функцій.

При розв'язанні окремих задач [138] РБФ-мережі показали високу ефективність, але необхідність використання великої кількості нейронів прихованого шару заважає використовувати їх як універсальне середовище моделювання процесів.

1.4.2. Конкурентні структури

1.4.2.1. WTA структура Ліпмана–Хеммінга

Для розв'язання задач класифікації застосовують велику кількість структур, елементами яких є зірки Гроссберга. Вхідною зіркою Гроссберга називають нейрон з вхідним вектором $X = (x_1, x_2, \dots, x_n)$ та активаційною функцією

$$Y = f\left(\sum_{i=1}^n w_i x_i\right),$$

для якого існує такий вектор вагових коефіцієнтів $W = (w_1, w_2, \dots, w_n)$, який забезпечує збудження нейрона тільки у випадку, коли на його вхід поступає потрібний вектор вхідних сигналів. Існує також вихідна зірка Гроссберга з одним входом X та вектором вихідних сигналів $Y = (y_1, y_2, \dots, y_n)$, що моделює функції командного нейрона. При надходженні сигналу на вхід, на його виході формується заданий вектор вихідних сигналів. Елементи нейронної мережі Ліпмана–Хеммінга мають властивості вхідної та вихідної зірок Гроссберга.

Структура, показана на рис.1.19, реалізує принцип “Winner Take All” (WTA). Вона містить один шар з m нейронів, кожен з яких відповідає за свій клас об'єктів, які потрібно класифікувати. У загальному випадку кожний елемент вхідного вектора зв'язаний з кожним класифікуючим нейроном. Вагові коефіцієнти утворюють матрицю:

$$W = \begin{Bmatrix} w_{01} & \dots & w_{0m} \\ \cdot & \dots & \cdot \\ w_{n1} & \dots & w_{nm} \end{Bmatrix}, \quad (1.17)$$

кожний елемент якої визначається за формулою:

$$w_{ij} = \frac{x_i^j}{\sum_{k=1}^n x_k^j}, \quad (1.18)$$

де x_i^j — i -й елемент вхідного вектора, що через відповідний зв'язок подається на j -й нейрон; w_{ij} — ваговий коефіцієнт елемента x_i^j вхідного вектора.

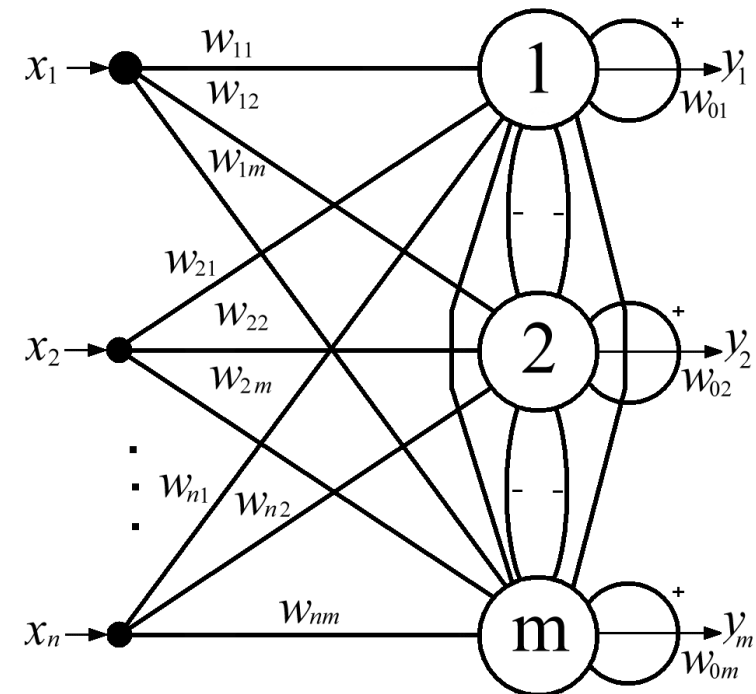


Рис.1.19. Нейронна мережа Ліпмана–Хеммінга.

Після подачі на вхід нейронної мережі вектора вхідного сигналу, вихідний сигнал кожного з нейронів буде відповідати його початковій активності, що визначається за формулою:

$$y_j(0) = f(v_j) = f\left(\frac{\sum_{i=1}^n x_i w_{ij}}{2} - \Theta\right). \quad (1.19)$$

Функція $f(v)$ є пороговою функцією:

$$y = f(v) = \begin{cases} v & \text{при } v \geq 0, \\ 0 & \text{при } v < 0. \end{cases}$$

Тому Θ визначає зсув порога, який зазвичай дорівнює нулю.

Отже, на першому етапі одержуємо множину активних елементів мережі, сформовану з тих нейронів, аргумент активізаційної функції яких перевищив поріг. Наступні етапи розв'язання задачі класифікації на даній мережі полягають у виборі з даної множини елементів нейрона з максимальною активністю. Такий вибір реалізують шляхом введення внутрішньосарових зв'язків, які виконують функції "літерального гальмування". Кожен нейрон пригнічує активність інших нейронів і пропорційно підвищує свою активність за формулою:

$$y_j(t+1) = f\left(y_j(t) - \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^m y_i(t)\right). \quad (1.20)$$

Після кількох ітерацій такого типу активним залишається лише один нейрон, який відповідає за клас, до якого належить даний вхідний вектор сигналів.

Принцип "літерального гальмування" має свої аналоги у живій природі, тому широко використовується при реалізації ШНМ. Загальним недоліком подібного типу структур є те, що кількість класів простору класифікації дорівнює кількості

елементів нейронної мережі. Тому вихід з ладу довільного нейрона призводить до втрати інформації про клас, за який відповідав даний нейрон. Така структура пам'яті називається прямою.

1.4.2.2. Нейронна мережа Кохонена

На відміну від мережі Ліпмана–Хеммінга модель нейронної мережі Кохонена характеризується структурою розподіленої пам'яті. Така структура дозволяє уникнути катастрофічної деградації у випадку відмови одного з нейронів. Ефект підвищеної живучості досягається саме завдяки розподіленій пам'яті, дія якої проявляється за рахунок того, що за класифікацію вхідного вектора відповідає не один нейрон, а кластер нейронів.

На рис.1.20 показана структура нейронної мережі Кохонена. У цій структурі кожний елемент вхідного вектора сигналів $X = (x_1, x_2, \dots, x_n)$ поступає на вхід кожного елемента двовимірної матриці нейронів. Тому множина вагових коефіцієнтів структурована у вигляді матриці

$$W = \begin{pmatrix} W^{11} & W^{12} & \dots & W^{1m} \\ W^{21} & W^{22} & \dots & W^{2m} \\ \vdots & \vdots & \dots & \vdots \\ W^{k1} & W^{k2} & \dots & W^{km} \end{pmatrix}, \quad (1.21)$$

елементами якої є вектори вагових коефіцієнтів $W^{ij} = (w_1^{ij}, w_2^{ij}, \dots, w_n^{ij})$, що масштабують відповідні елементи вхідного вектора.

Всім ваговим коефіцієнтам на початку роботи мережі задають випадкові значення. Наступний етап полягає в обчисленні

відстані $d_{ij} = \sum_{p=1}^n (x_p(t) - w_p^{ij}(t))^2$ між вектором вхідного сигналу та кожним з нейронів мережі.

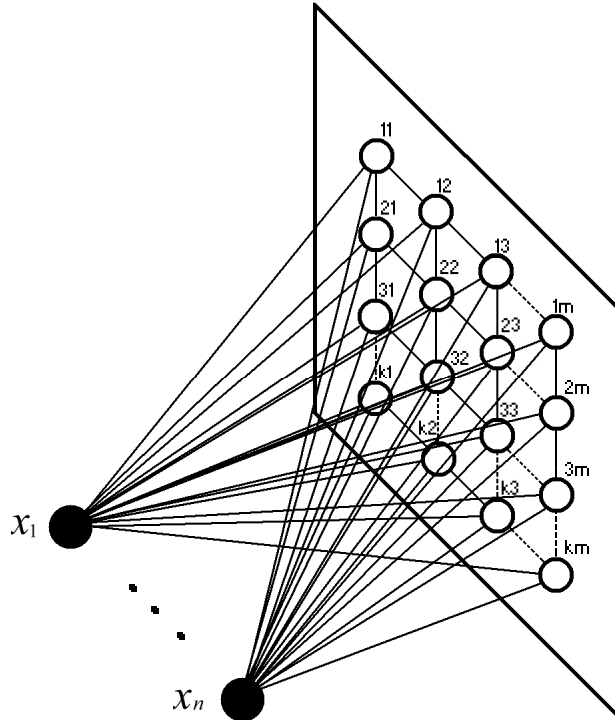


Рис.1.20. Нейронна мережа Кохонена.

Нейрон із координатами (i^*, j^*) , для якого ця відстань є мінімальною, вибирається центром кластера. Відносно нього проводиться підстроювання вагових коефіцієнтів всіх нейронів, які входять до початкового кластера, за формулою:

$$W^{ij}(t+1) = W^{ij}(t) + \eta r(l^{ij}) [X(t) - W^{ij}(t)], \quad (1.22)$$

де $0 < \eta < 1$ — коефіцієнт навчання, який змінюється обернено пропорційно до часу навчання;

$$r(l^{ij}) = \exp\left(\frac{-(l^{ij})^2}{\sigma^2}\right) — \text{одна з можливих функцій сусідства,}$$

що задає зміну вагових коефіцієнтів у залежності від відстані

$$l^{ij} = \left| \sqrt{(j^* - j)^2 + (i^* - i)^2} \right| \text{ нейрона } (i, j) \text{ до центра кластера.}$$

Конфігурацію початкового кластера найчастіше формують як просту геометричну фігуру (квадрат, прямокутник, круг), яка займає від 0,5 до 0,75 розміру нейронної мережі.

1.4.2.3. Нейронні мережі зустрічного поширення

Нейронні мережі зустрічного поширення — це ще одна відома парадигма з використанням механізму WTA. Узагальнена структура такої мережі показана на рис.1.21. Вона складається з двох шарів нейронів різного типу. Перший шар є двовимірною матрицею Кохонена, а другий шар називають шаром Гроссберга. Об'єднання цих двох шарів надає структурі додаткові властивості, які відсутні у кожного з них окремо.

Вхідний вектор сигналів $X = (x_1, x_2, \dots, x_n)$ поступає на кожний з нейронів шару Кохонена. Масштабування відбувається за допомогою матриці вагових коефіцієнтів, що містить вектори вагових коефіцієнтів. Відомі два методи функціонування шару Кохонена, що отримали назви “метод акредитації” та “метод інтерполяції”.

Метод акредитації

У випадку застосування методу акредитації шар Кохонена реалізує механізм WTA, який забезпечує активацію лише одного

нейрона-переможця. Аргумент активаційної функції v_{ij} для нейрона (i, j) шару Кохонена обчислюється за формулою:

$$v_{ij} = \sum_{k=1}^n w_k^{ij} x_k, \quad (1.23)$$

а сукупність всіх аргументів утворює матрицю

$$V = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \dots & \dots & \dots & \dots \\ v_{k1} & v_{k2} & \dots & v_{km} \end{pmatrix}. \quad (1.24)$$

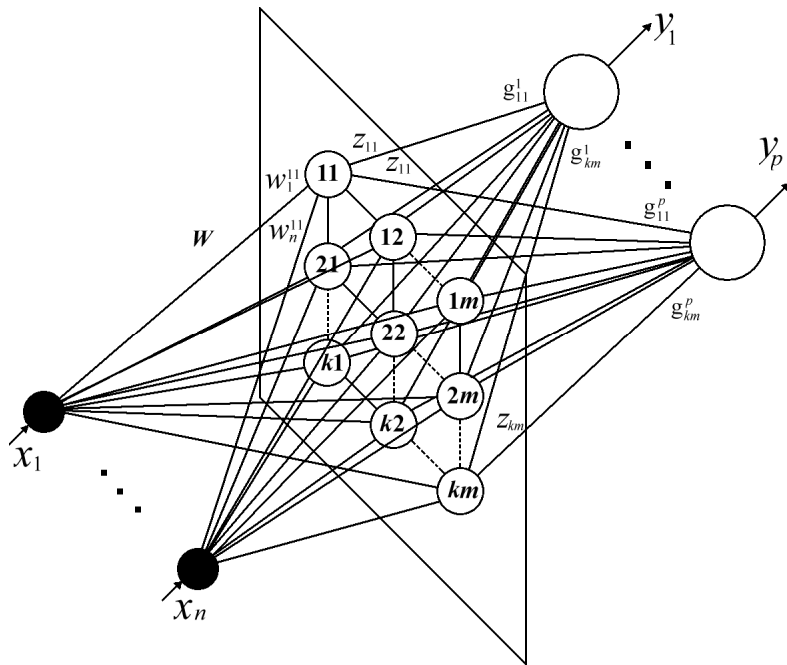


Рис.1.21. Нейронна мережа зустрічного поширення.

Вихідний сигнал z_{ij} нейрона (i, j) заданий пороговою активаційною функцією

$$z_{ij} = f_{ij}(v_{ij}) = \begin{cases} 1 & \text{при } v_{ij} > r, \\ 0 & \text{при } v_{ij} \leq r, \end{cases} \quad (1.25)$$

де r — поріг чутливості нейрона.

Механізм типу WTA у даному випадку реалізується за допомогою ітераційного процесу

$$W^{ij}(t+1) = W^{ij}(t) + \eta(X(t) - W^{ij}(t)), \quad (1.26)$$

де $0 < \eta < 1$ — коефіцієнт навчання.

Результатом його є пропорційне зменшення вагових коефіцієнтів до того моменту, поки не залишиться лише один нейрон, для якого справджується умова $v_{ij} > r$.

Кожний нейрон шару Гроссберга формує свій аргумент активаційної функції за формулою:

$$b_h = \sum_{i=1}^m \sum_{j=1}^k g_{ij}^h z_{ij}, \quad (1.27)$$

Вихідний сигнал нейрона h шару Гроссберга формується активаційною функцією

$$y_h = f_h(b_h) = b_h. \quad (1.28)$$

Оскільки в результаті роботи шару Кохонена залишається активним тільки один нейрон, то тільки один елемент z_{cd} матриці вихідних сигналів Z має значення $z_{cd} = 1$. У цьому випадку шар Гроссберга видає вихідний вектор сигналів

$$Y = (g_{cd}^1, g_{cd}^2, \dots, g_{cd}^h, \dots, g_{cd}^p).$$

Метод інтерполяції

Даний метод функціонування нейронної мережі зустрічного поширення полягає в тому, що в результаті роботи механізму WTA в шарі Кохонена залишається переможцем кластер

нейронів, який характеризувався максимальною активністю. Вихідні сигнали кластера нормалізують. Якщо в такий кластер входить підмножина нейронів, що задана матрицею розмірності (c, d) , то довільний вихідний сигнал нейрона $z_{\alpha\beta}$, який належить до даного кластера, нормалізуємо за формулою:

$$\overline{z_{\alpha\beta}} = \frac{z_{\alpha\beta}}{\sqrt{\sum_{i=1}^c \sum_{j=1}^d z_{ij}^2}}. \quad (1.29)$$

Тоді активаційна функція шару Гроссберга має вигляд:

$$y_h = \sum_{i=1}^m \sum_{j=1}^k g_{ij}^h \overline{z_{ij}}. \quad (1.30)$$

Оскільки всі нейрони шару Кохонена, крім тих, що входять до кластера-переможця, видають нульові значення вихідного сигналу, результатом роботи шару Гроссберга є значення сигналів активних нейронів, промасштабовані відповідними ваговими коефіцієнтами. Завдяки розподіленій пам'яті шару Кохонена метод інтерполяції дає можливість реалізувати більш складні алгоритми класифікації та категоризації, ніж метод акредитації.

1.4.3. Рекурентні структури

Структури мереж прямого поширення, що уже розглядалися, не мають зворотних зв'язків, тобто сигнали в них поширюються від шару до шару тільки в одному напрямку, формуючи статичний стан кожного нейрона мережі. Дещо складнішим є принцип функціонування конкурентних структур. В мережах такого типу затухаючий ітераційний процес відбувається в межах одного шару нейронів, а ітераційна формула завжди має властивість зниження рівня вихідного сигналу, що викликає затухання слабких вихідних сигналів до рівня, нижчого від порога чутливості. Таким чином реалізується стратегія

“переможець забирає все”, що зупиняє ітераційний процес у випадку перемоги одного нейрона або кластера нейронів.

Принципово новою є ситуація, коли структура нейронної мережі допускає зворотні зв'язки, тобто, коли обчислення в нейроні даного шару відбуваються з урахуванням попереднього стану цього ж шару. Мережі такого типу називають **рекурентними** та говорять про динамічний характер їх функціонування. Динаміка зміни станів сукупності взаємопов'язаних об'єктів традиційно вимагає визначення правила взаємної координації дій цих об'єктів у часі та просторі.

Коли встановлюється послідовність спрацьовування нейронів одного шару, то таку нейронну мережу називають **синхронною**. У випадку, коли час спрацьовування кожного з нейронів не регламентується, мережу називають **асинхронною**. Синхронні мережі можуть бути паралельними, послідовними та паралельно-послідовними. Для паралельних мереж визначальною є властивість одночасного спрацьовування всіх нейронів одного шару, для послідовних мереж задають послідовність спрацьовування нейронів одного шару, а для паралельно-послідовних мереж визначається послідовність спрацьовування кластерів нейронів. Про асинхронні мережі говорять, що вони є паралельними, маючи на увазі те, що такі мережі допускають одночасне функціонування всіх нейронів.

1.4.3.1. Нейронна мережа Хопфілда

Структура рекурентної нейронної мережі, яка показана на рис.1.22, вперше була запропонована в [105].

Вона складається з одного шару нейронів, виходи яких через спеціальні пристрої з'єднані зі входами всіх нейронів цього ж шару, крім тих зв'язків, що з'єднують вихід нейрона з його власним входом.

Формування аргументу активаційної функції v_j довільного нейрона j відбувається за формулою:

$$v_j = \sum_{\substack{i=1 \\ i \neq j}}^n w_{ij} y_i + x_j. \quad (1.31)$$

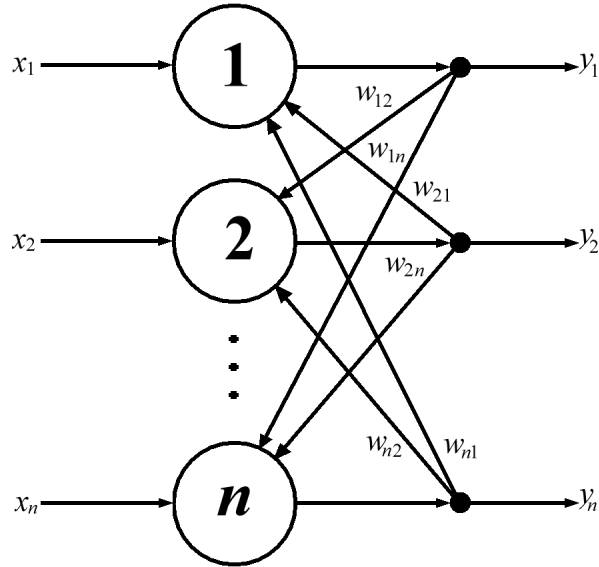


Рис.1.22. Нейронна мережа Хопфілда.

В залежності від вигляду активаційної функції розрізняють дискретну та аналогову моделі мереж Хопфілда.

1.4.3.1. 1. Бінарна модель

Бінарна модель базується на використанні порогової активаційної функції, яку задають виразом:

$$y_j = f(v_j) = \begin{cases} 1 & \text{при } v_j > T_j, \\ 0 & \text{при } v_j < T_j, \\ y_j & \text{при } v_j = T_j. \end{cases} \quad (1.32)$$

Такий вигляд активаційної функції був запропонований Хопфілдом. У сучасній літературі частіше зустрічається порогова активаційна функція зі зміною знака

$$y_j = f(v_j) = \begin{cases} 1 & \text{при } v_j > T_j, \\ -1 & \text{при } v_j < T_j, \\ j & \text{при } v_j = T_j. \end{cases} \quad (1.33)$$

В (1.32) і (1.33) через T_j позначається величина порога чутливості довільного нейрона j мережі Хопфілда.

Якщо для виходу y_j кожного нейрона поставити у відповідність двійковий розряд, то поточний стан мережі Хопфілда може бути виражений числом у двійковій системі числення, а множина допустимих сусідніх станів є множиною чисел з одиничною відстанню Хеммінга. Якщо стани нейронної мережі позначити точками з відповідними координатами у просторі та з'єднати дугами ті точки, між якими допустимий перехід, одержимо граф переходів мережі. Для мережі, що складається з трьох нейронів, такий граф має вигляд куба (рис.1.23). Мережі з кількістю нейронів більше трьох утворюють гіперкуби вищих порядків, що не мають графічної інтерпретації.

У загальному випадку множина сусідніх вершин N для довільної вершини a визначається з виразу:

$$N(a) = \left\{ a \text{ XOR } 2^{i-1} \right\}_{i=1}^n, \text{ де } a = 0, 1, \dots, 2^n - 1. \quad (1.34)$$

Функціонування мережі Хопфілда полягає в пересуванні вздовж ребер переходів, поки мережа не досягне стійкого стану, умови наявності якого вперше сформульовані в [58]. Мережа Хопфілда має стійкий стан у випадку, коли матриця вагових коефіцієнтів W симетрична і має нулі на головній діагоналі, тобто

$$w_{ij} = w_{ji}, \quad w_{ii} = 0.$$

Для пояснення факту стійкості мережі за згаданих умов введемо функцію енергії довільної пари нейронів:

$$e_{ij} = -w_{ij}y_i y_j - x_j y_j + T_j y_j.$$

Виходячи з умови симетричності $w_{ij} = w_{ji}$, $w_{ii} = 0$ матриці вагових коефіцієнтів, загальну енергію E нейронної мережі визначають за функцією Ляпунова:

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} y_i y_j - \sum_{j=1}^n x_j y_j + \sum_{j=1}^n T_j y_j. \quad (1.35)$$

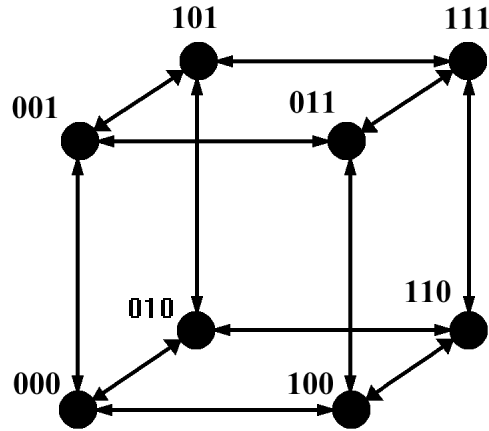


Рис. 1.23. Граф переходів.

Розглянемо стан мережі перед спрацюванням нейрона k , виділивши з загальної суми складову C , що не пов'язана зі зміною енергії:

$$C = -\frac{1}{2} \sum_{\substack{i \neq k \\ j \neq k}} w_{ij} y_i y_j + \sum_{\substack{j \neq k}} x_j y_j - \sum_{\substack{j \neq k}} T_j y_j. \quad (1.36)$$

Тоді загальна енергія мережі

$$E = C - \frac{1}{2} \sum_i w_{ik} y_i y_k - \frac{1}{2} \sum_i w_{ki} y_i y_k + x_k y_k - T_k y_k. \quad (1.37)$$

Враховуючи умови симетричності $w_{ki} = w_{ik}$, одержимо вираз:

$$E = C - \sum_i w_{ik} y_i y_k + x_k y_k - T_k y_k. \quad (1.38)$$

На момент розгляду стану мережі вихід нейрона y_k є константою, і може бути винесеним за знак суми.

$$E = C - y_k \left(\sum_i w_{ik} y_i + x_k - T_k \right). \quad (1.39)$$

Загальна енергія перед черговим спрацюванням k -го нейрона дорівнює $E = C - y_k (v_k - T_k)$, а після спрацювання — $E' = C - y'_k (v_k - T_k)$. Зміну енергії задамо виразом:

$$\Delta E = E' - E = -\Delta y_k (v_k - T_k) \quad (1.40)$$

і розглянемо такі умови:

1. $v_k > T_k$. За цієї умови:

$$\Delta y_k = y'_k - y_k = \begin{cases} 1 & \text{при } y'_k = 1, y_k = 0, \\ 0 & \text{при } y'_k = 1, y_k = 1, \end{cases}$$

отже, $\Delta E \leq 0$.

2. $v_k < T_k$. В цьому випадку:

$$\Delta y_k = y'_k - y_k = \begin{cases} -1 & \text{при } y'_k = 0, y_k = 1, \\ 0 & \text{при } y'_k = 0, y_k = 0, \end{cases}$$

отже, $\Delta E \leq 0$.

3. $v_k = T_k$. Згідно з активаційною функцією у цьому випадку

не відбувається жодних змін, тобто $y_k = y'_k$. Відповідно, $\Delta E = 0$.

Отже, загальна енергія мережі Хопфілда за умов $w_{ij} = w_{ji}$, $w_{ii} = 0$ завжди або зменшується, або залишається незмінною. Стійкий стан мережі відповідає мінімуму її енергії.

1.4.3.1.2. Асоціативна пам'ять

Поверхня енергії в просторі станів мережі має складну форму, яка характеризується великою кількістю локальних мінімумів. У випадку застосування мережі Хопфілда в ролі запам'ятовуючого середовища стійкі стани, що відповідають локальним мінімумам енергії, інтерпретують як образи пам'яті. Під час переходу мережі від одного стану до такого, що характеризується меншим рівнем енергії, завжди відбувається пошук сусіднього стану за формулою (1.34). Тому мережа завжди знаходить локальний енергетичний мінімум, який розміщується на мінімальній відстані Хеммінга від початкового стану. Нехай вектор вхідних сигналів X відповідає ідеальному образу пам'яті, тобто локальному мінімуму енергії мережі. Тоді надходження на вхід мережі вектора X_i приводить до еволюції мережі у напрямку локального мінімуму, що відповідає відновленню ідеального образу по його неповній копії.

Така пам'ять відрізняється від тієї, що використовується в традиційних обчислювальних структурах. Вона більше нагадує людську пам'ять, для якої характерний асоціативний, а не адресний принцип запам'ятовування інформації. Асоціативна пам'ять є значно стійкішою щодо різного роду помилок та шумів за рахунок того, що має здатність відновлення ушкодженого образу, на відміну від адресної пам'яті, у якій помилка адресації призводить до неправильного зчитування значних фрагментів.

Недоліком асоціативної пам'яті, яка реалізується на мережі Хопфілда, є незначний її об'єм, пропорційний кількості нейронів. У цьому випадку об'єм адресної пам'яті пропорційний 2^n , де n — кількість двійкових розрядів.

Згаданий недолік та ряд інших, які характерні для побудованої на основі класичної мережі Хопфілда асоціативної пам'яті, роблять її малопридатною для практичного використання.

1.4.3.1.3. Неперервна модель

В роботі [104] розглянуто ще одну модель з дискретною зміною часу, яка пізніше була названа неперервною моделлю Хопфілда. Принциповою відмінністю від бінарної моделі стало використання сигмоїдальної активаційної функції

$$y_j = f_j(v_j) = \frac{\beta}{1 + e^{-\alpha v_j}}, \quad (1.41)$$

а також введення показника швидкості зміни активності, що визначається за формулою:

$$\frac{\partial v_j}{\partial t} = \alpha v'_j - b v_j, \quad (1.42)$$

де a, b — константи, а v_j, v'_j — значення аргументу активаційної функції відповідно до i після спрацьовування нейрона j .

Як і в попередньому випадку, простором зміни станів мережі є гіперкуб розмірності, яка дорівнює кількості нейронів. Оскільки виходи нейронів неперервні зі зміною в діапазоні від 0 до 1, то допустимими станами мережі в даному випадку стають не тільки вершини, а й увесь внутрішній простір гіперкуба.

В [104] принципи стійкості бінарної моделі поширені також на неперервну модель. Доведено, що ця модель має стійкий стан у випадку симетричності матриці вагових коефіцієнтів та рівності нулю її головної діагоналі $w_{ij} = w_{ji}$, $w_{ii} = 0$. Від коефіцієнта крутизни активаційної функції α залежить відстань від вершини гіперкуба до точки, яка характеризується мінімумом енергії. Коли активаційна функція за формою нагадує порогову, точки стійких станів максимально наближаються до вершин гіперкуба. З точки зору стійкості бінарна модель може розглядатися як граничний стан неперервної моделі.

Більша наближеність неперервної моделі до її біологічного аналога та відносна легкість реалізації зробили цей варіант мережі Хопфілда популярним. У такому варіанті вона часто використовується в сукупності з іншими видами мереж з метою підвищення ефективності обробки інформації.

1.4.3.1.4. Комбінаторна оптимізація

Цікаве застосування неперервної моделі для розв'язування задач комбінаторної оптимізації було запропоноване в [103]. Типовим представником таких задач є задача комівояжера [13], яка полягає в тому, щоб знайти найкоротший шлях між N містами, відвідавши кожне з них тільки один раз. Розв'язок цієї задачі шукають на матриці нейронів розмірністю $N \times N$. У даній матриці кожному місту відповідає рядок, а номером стовпця задають послідовність їх відвідування.

Активаційну функцію вибирають типу (1.41) з такими коефіцієнтами α, β , які забезпечують її максимальне наближення до порогової функції (1.32). Правила роботи мережі задають шляхом конструювання відповідної функції енергії, яка повинна задовольняти таким умовам:

1. Мінімальне значення енергії E_1 для кожного стовпця матриці досягається у випадку, коли в даному стовпці залишається активним лише один нейрон:

$$E_1 = \frac{A}{2} \sum_{k=1}^N \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N y_{ki} y_{kj}. \quad (1.43)$$

2. Мінімальне значення енергії E_2 для кожного рядка матриці досягається у випадку, коли в даному рядку залишається активним лише один нейрон:

$$E_2 = \frac{B}{2} \sum_{i=1}^N \sum_{k=1}^N \sum_{\substack{m=1 \\ m \neq k}}^N y_{ki} y_{mi}. \quad (1.44)$$

3. Мінімальне значення енергії E_3 для всієї матриці досягається, коли кількість активних нейронів дорівнює N , тобто в кожному стовпці матриці та в кожному її рядку існує активний нейрон, і до того ж тільки один:

$$E_3 = \frac{C}{2} \left[\left(\sum_{k=1}^N \sum_{i=1}^N y_{ki} \right) - N \right]^2. \quad (1.45)$$

4. Енергія мережі E_4 досягає мінімального значення у випадку, коли між можливими парами нейронів (k, m) вибирається тільки така пара, яка характеризується мінімальною відстанню d_{km} :

$$E_4 = \frac{D}{2} \sum_{k=1}^N \sum_{m=1}^N \sum_{\substack{i=1 \\ m \neq k}}^N d_{km} y_{ki} (y_{m,i+1} + y_{m,i-1}). \quad (1.46)$$

Загальна енергія мережі визначається за формулою:

$$E = E_1 + E_2 + E_3 + E_4.$$

Всі ці умови впливають з умови задачі комівояжера. Виходячи з них, задають правила формування вагових коефіцієнтів, які дозволяють мінімізувати час досягнення мережею стійкого стану. В роботі [13] наведені результати ефективного розв'язання задачі комівояжера для десяти міст.

1.4.3.2. Двонаправлена асоціативна пам'ять

Основною властивістю асоціативної пам'яті, що реалізована на мережі Хопфілда, є здатність відтворювати образи шляхом їх асоціації з частково помилковим запитом. Таку мережу ще

називають *автоасоціативною*. В [120] запропоновано більш досконалу версію асоціативної пам'яті — двонаправлену асоціативну пам'ять (ДАП), що отримала назву *гетероасоціативної*. Ця структура дозволяє додатково встановлювати асоціативні зв'язки між різними образами. На рис. 1.24 показана структурна схема такої мережі.

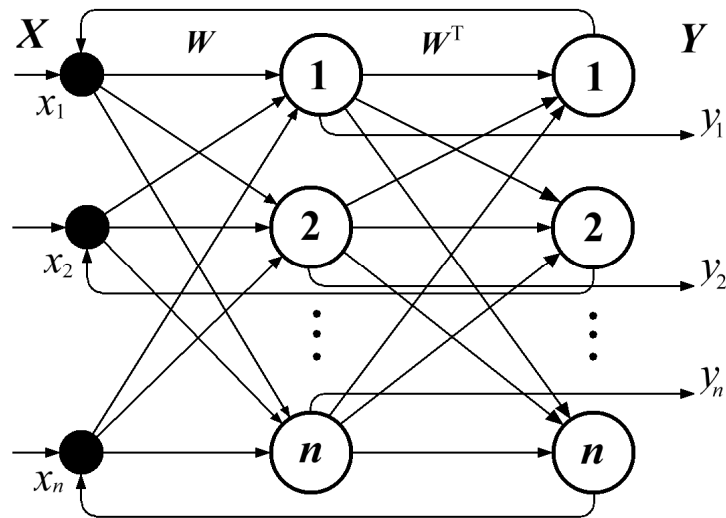


Рис. 1.24. Двонаправлена асоціативна пам'ять.

Вектор вхідних сигналів X , відповідно промасштабований елементами матриці вагових коефіцієнтів W , надходить на вхід першого шару нейронів. Результатом спрацювання цього шару є вихідний вектор сигналів Y , який надходить на вхід другого шару з транспонованою матрицею вагових коефіцієнтів W^T . До моменту його спрацювання зовнішній вхідний вектор X знімається і замінюється тим, який є результатом

роботи другого шару. Далі починається новий цикл рекурентних обчислень, і він продовжується до того часу, поки мережа не досягне стійкого стану, в якому вектори X та Y залишаються незмінними.

Активаційна функція першого шару нейронів визначається за формулою :

$$y_j = f(v_j) = f\left(\sum_{i=1}^n w_{ij}x_i\right). \quad (1.47)$$

Нейрони другого шару також працюють в традиційному режимі, формуючи вхідний вектор сигналів за формулою:

$$x_i = f(v_i^t) = f\left(\sum_{j=1}^n w_{ji}^t y_j\right), \quad (1.48)$$

де w_{ji}^t — ваговий коефіцієнт, що є елементом транспонованої матриці W^T .

Найчастіше в ролі активаційної функції використовують традиційну сигмоїдальну функцію (1.41) з великими значеннями коефіцієнта α , що наближає її до порогової функції.

За принципами роботи розрізняють синхронні та асинхронні ДАП. У синхронному режимі відбувається одночасне спрацювання всіх нейронів одного шару під дією зовнішнього синхронізуючого сигналу. Найчастіше у цьому випадку використовують порогову активаційну функцію. Для досягнення більшої подібності з біологічним прототипом застосовують асинхронний режим функціонування ДАП з сигмоїдальною активаційною функцією. В цьому режимі кожний нейрон обробляє неперервні сигнали і може бути змодельований на операційному підсилювачі з нелінійним зворотним зв'язком.

ДАП — більш стійкі структури, ніж мережі Хопфілда. Для них притаманним є прагнення до пошуку локального енергетичного мінімуму у випадку, коли матриця вагових коефіцієнтів W має довільну форму. Спільним недоліком для

цих мереж є незначна ємність пам'яті. В роботі [90] наводиться оцінка, за якою кількість образів, що можуть запам'ятовуватись мережами такого типу, пропорційна $\left(\frac{n}{2 \log_2 n}\right)$, де n — кількість нейронів.

Існує ряд підходів, які дозволяють покращити розпізнавальні властивості ДАП і, відповідно, збільшити кількість образів, що можуть викликати адекватні асоціації. До таких підходів слід віднести застосування нейронів з різними порогоми спрацьовування активаційних функцій. Мережі, що мають такі властивості, називають адаптивними ДАП. Активаційні функції нейронів адаптивної ДАП визначаються за формулою (1.32) у випадку синхронного режиму функціонування. При цьому для кожного нейрона j встановлюється свій поріг спрацьовування T_j . Асинхронний режим функціонування використовує неперервну сигмоїдальну активаційну функцію, амплітуда якої визначається параметром β_j окремо для кожного нейрона j . Активаційна функція у цьому випадку:

$$y_j = f_j(v_j) = \frac{\beta_j}{1 + e^{-\alpha v_j}}. \quad (1.49)$$

Алгоритм настройки порогів підвищує чутливість мережі у вибраному діапазоні вхідних сигналів.

Ще одним підходом такого типу є застосування конкурентних зв'язків на кожному шарі нейронів, що приводить до формування нової матриці вагових коефіцієнтів. Як і в попередньому випадку, підвищення чутливості відбувається за рахунок зниження допустимого діапазону вхідних образів.

1.4.3.3. Стохастичні рекурентні мережі

У попередніх параграфах розглядалися класичні рекурентні структури, поведінку нейронів у яких визначають певні детерміновані правила. Виходячи з цих правил, можна

визначити стан мережі в кожний фіксований момент часу. Еволюція стійких детермінованих рекурентних мереж завжди відбувається у напрямку локального мінімуму енергії. Ця властивість є корисною у випадку, коли нейромережа моделює функції асоціативної пам'яті. Коли ж ставиться задача пошуку глобального мінімуму, стійкий стан мережі у локальному мінімумі є небажаним явищем. Для розв'язування подібних задач використовують стохастичні рекурентні мережі. Принциповою відмінністю стохастичних алгоритмів є відмова від заборони переходу мережі у стан з більш високим рівнем енергії.

Модель, що описує роботу стохастичної мережі, нагадує процес застигання металу, кожен атом якого знаходиться в неупорядкованому русі і має різний запас енергії, але загальна температура прямує до нуля. Імовірність переходу між енергетичними рівнями визначається за формулою Больцмана:

$$P(E) = \frac{1}{1 + \exp\left\{-\frac{E_2 - E_1}{T}\right\}}, \quad (1.50)$$

де $P(E)$ — імовірність переходу системи зі стану з енергією E_1 у стан з енергією E_2 ; T — температура за шкалою Кельвіна.

При великих значеннях температури величина імовірності $P(E) = \frac{1}{2}$, тобто наступний стан системи, не залежить від її попереднього стану, що відповідає хаотичній поведінці. Зі зменшенням температури величина імовірності прямує до одиниці, і поведінка мережі вже в такому стані збігається з поведінкою детермінованої мережі. В [85] дано результати дослідження стохастичних мереж, що показали високу ефективність використання подібних структур для розв'язування задач пошуку глобального мінімуму функції.

Для досягнення адекватності моделі охолодження металу на мережі вводиться поняття штучної температури $T(t)$ як функції часу, яка монотонно спадає і визначається за формулою:

$$T(t) = \frac{T_0}{\log(1+t)}, \quad (1.51)$$

де T_0 — початкова штучна температура; t — час функціонування мережі.

З (1.51) випливає основний недолік всіх стохастичних структур — це мала швидкість спадання штучної температури. Отже, для досягнення глобального мінімуму мережі потрібні значні затрати часу та обчислювальних ресурсів. Після досягнення мінімуму енергії мережа продовжує коливатись у його околі, що заважає одержанню чіткого образу.

З огляду на високу адекватність біологічному прототипу, стохастичні методи є дуже перспективними і широко досліджуються не тільки для рекурентних структур, а й для структур прямого поширення.

1.4.4. Нейронні мережі типу ART

При розв'язуванні задач класифікації та категоризації однією з найактуальніших залишається проблема стабільності-пластичності. Суть її полягає в тому, що для класифікуючої структури важливо встановити факт, чи є об'єкт класифікації новою інформацією. У випадку визначення факту новизни даний образ повинен запам'ятовуватись. Цей факт встановлюється шляхом застосування критерію новизни. Якщо структура знаходить у своїй пам'яті образ, який зустрічався раніше, то запам'ятовування не відбувається, а реакція є ідентичною до реакції на попередній образ.

Стабільність полягає в тому, що набута інформація не змінюється при застосуванні процесів пластичності, які забезпечують зростання об'єму інформації про навколишнє середовище.

Відомим підходом до вирішення проблеми стабільності-пластичності є теорія адаптивного резонансу, що виникла в Центрі адаптивних систем Бостонського університету під керівництвом професора Гроссберга. На базі цієї теорії

побудовано ряд нейронних мереж, що отримали назву ART-1-мережі [53]. Такі мережі працюють тільки з бінарними вхідними даними та мають загальний недолік, який полягає у підвищеній чутливості процесу формування кластерів від порядку подання на вхід послідовності випадкових образів. Подальший розвиток мереж даного типу був спрямований на розширення можливостей представлення вхідних образів. Мережа ART-2 розроблена для пошуку регулярних елементів у аналоговому вхідному потоці даних [50]. Ця мережа має складну технічну реалізацію, що спричинено складністю обробки аналогових сигналів. Нові дослідження дозволили створити мережу IART-1 (Improved ART-1) [165], яка є менш чутливою до вхідного потоку даних. Адаптивна мережа Хеммінга (AHN) функціонально еквівалентна мережам ART-1 та IART-1, але допускає паралельну реалізацію вихідного шару (MAXNET), що дозволяє оптимізувати час, який витрачається на обробку інформації, об'єми її зберігання та параметри доступу до неї [108]. Проблеми, що виникли при створенні як аналогових, так і бінарних ART-мереж, вдалося частково подолати шляхом застосування технології нечіткої логіки. В результаті розробки цього підходу було створено ряд модифікацій мереж Fuzzy ART. Узагальнена архітектура ART-мереж, які базуються на нечіткій логіці, отримала назву Fuzzy ARTMAP [51].

1.4.4.1. Нейронна мережа типу ART-1

Мережа ART-1 [53] стала першою нейронною структурою, яка реалізує бінарну інформаційну модель адаптивного резонансу. Спрощена її структура показана на рис.1.25, з якого видно, що мережі типу ART нагадують уже відомі нам конкурентні нейронні мережі Ліпмана-Хеммінга з використанням механізму “переможець забирає все”.

Але, на відміну від цих мереж, ART-мережі є тришаровими структурами, які складаються з вхідного шару F_0 , порівняльного шару F_1 та розпізнавального шару F_2 . Між

нейронами шару F_1 та шару F_2 існують двонаправлені зв'язки. Механізм літерального гальмування також потребує внутрішніх зв'язків між нейронами шару F_2 .

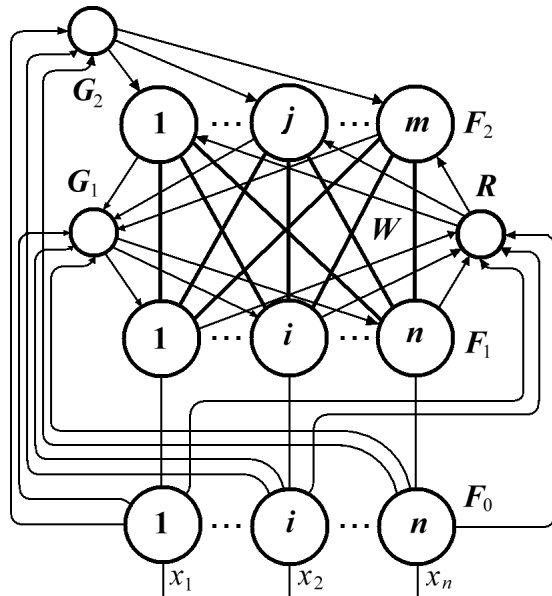


Рис.1.25. Нейронна мережа типу ART.

Особливістю мереж типу ART є наявність елементів глобального управління, які реалізують алгоритм функціонування нейронної мережі. Вузли G_1 і G_2 керують активністю відповідних шарів нейронів, а R забезпечує виконання операції початкової установки.

Введемо позначення: n — кількість нейронів порівняльного шару; m — кількість нейронів розпізнавального шару;

W — множина вагових коефіцієнтів зв'язку між нейронами шару F_1 та F_2 ; r — величина порога подібності.

Після надходження на вхід вектора сигналів X , ($x_i \in \{0,1\}$, $i = 1, 2, \dots, n$), нейронна мережа формує значення прямої активації T_j на виході шару F_2 за формулою:

$$T_j = \frac{\|w_j \cap X\|}{\beta + \|w_j\|}, \quad j = 1, \dots, m, \quad (1.52)$$

де $w_j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})$; $\|\cdot\|$ — оператор норми

$\left(\|u\| = \sum_{j=1}^m u_j \right)$; $\beta > 0$ — параметр вибору.

Після цього запускається внутрішній ітераційний процес літерального гальмування:

$$T_j(\tau+1) = f \left(T_j(\tau) - \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^m T_i(\tau) \right) \quad (1.53)$$

де τ — порядковий номер внутрішньої ітерації.

Результатом цього процесу є вибір нейрона j з максимальною активністю, і тому вектор сигналів зворотного зв'язку T_j містить тільки один відмінний від нуля елемент:

$$T_j = \max \{T_j | j = 1, 2, \dots, m\}.$$

На цьому закінчується початкова фаза аналізу вхідного вектора. Наступна робоча фаза починається з операції порівняння на першому шарі нейронів. Вектор вагових коефіцієнтів w_j нейрона j порівнюється з вектором X :

$$\frac{\|w_j \cap X\|}{\|X\|} \geq r. \quad (1.54)$$

Якщо нерівність (1.54) виконується, то цей факт свідчить про те, що знайдено прототип, який перевищує поріг подібності. Для того, щоб уточнити параметри знайденого прототипу,

виконують процедуру корекції відповідних вагових коефіцієнтів за формулою:

$$w_j(t+1) = \eta [w_j(t) \cap X] + (1-\eta)w_j(t), \quad (1.55)$$

де t — номер кроку навчання; η — коефіцієнт навчання $\eta \in [0,1]$. Якщо нерівність (1.54) не виконується, то це свідчить про відсутність відповідного прототипу для вхідного образу. У цьому випадку вузлом R активується сигнал початкової установки для вузла j , що вилучає його з подальшої роботи. Після цього цикл пошуку прототипу повторюється без участі нейрона, який був переможцем у попередньому циклі.

Якщо після гальмування всіх нейронів шару F_2 не буде знайдений образ, що відповідав би вхідному вектору, то даний вхідний вектор вважається новим. Під нього виділяється невикористаний нейрон шару F_2 з відповідною модифікацією матриці вагових коефіцієнтів W .

Як видно з опису, нейронна мережа ART-1 здатна працювати тільки з бінарними вхідними сигналами. Розширення сфери представлення вхідного образу досягається в подальших версіях ART-2 та ART-3. Особливістю роботи з аналоговими сигналами є необхідність використання більш точних засобів порівняння векторів даних, оскільки відстань між порівнюваними векторами в аналоговому просторі може бути як завгодно малою. Для підвищення розпізнавальних властивостей застосовують багатозарову архітектуру з послідовним уточненням відмінностей на кожному з шарів. Така архітектура підвищує розпізнавальні властивості ART-мереж, але залишає неподоланим основний недолік, який полягає в концентрації пам'яті про конкретний прототип лише в одному нейроні. Така властивість не відповідає біологічній пам'яті, яка розподілена між багатьма нейронами.

1.4.4.2. Нейронна мережа типу АНН

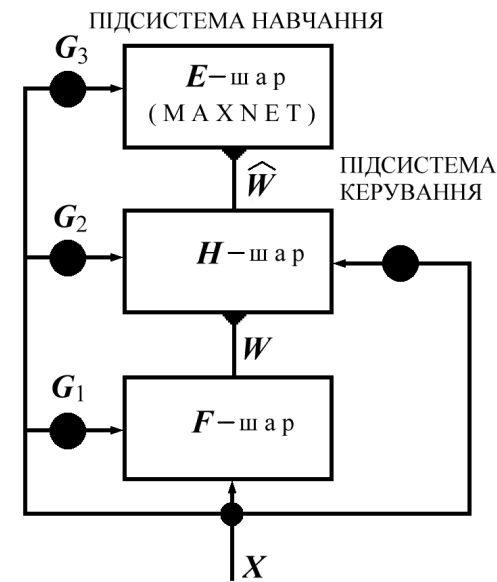


Рис.1.26. Нейронна мережа АНН. (крім прямого поширення) є застосування мережі MAXNET для реалізації стратегії „переможець забирає все” на вихідному шарі.

Структура мережі MAXNET показана на рис.1.27. Всі m нейронів утворюють повне з'єднання, включаючи замикання виходу на вхід у кожному нейроні. MAXNET не потребує алгоритму навчання, оскільки використовує фіксовані значення вагових коефіцієнтів. Функція активації для кожного з нейронів має вигляд:

$$f(a) = \begin{cases} a & \text{при } a > 0, \\ 0 & \text{при } a \leq 0. \end{cases} \quad (1.56)$$

Значення вагових коефіцієнтів задають, виходячи з виразу:

$$w_{ij} = \begin{cases} -d & \text{при } i \neq j, \\ 1 & \text{при } i = j. \end{cases} \quad (1.57)$$

Нейронна мережа АНН, зображена на рис.1.26, є функціонально еквівалентна ШНМ ART-1 [29], хоч вона і є мережею прямого поширення, тобто відмінна від мережі класичної структури, яка містить зв'язки прямого та зворотного поширення. АНН складається з підсистеми навчання та підсистеми керування.

Основною її структурною відмінністю

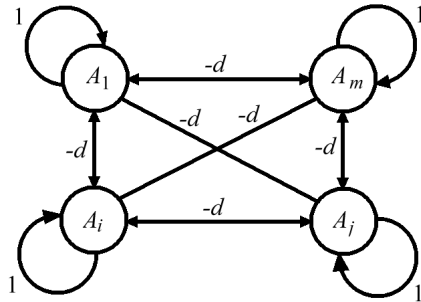


Рис.1.27. Мережа MAXNET.

З метою визначення нейрона-переможця застосовують ітераційний процес, який продовжується до виконання умови закінчення алгоритму:

$$a_j(\tau + 1) = f \left(a_j(\tau) - d \sum_{k \neq j} a_k(\tau) \right), \text{ де } j = 1, \dots, m. \quad (1.58)$$

Умовою його закінчення є нульове значення на виході всіх нейронів, крім одного.

1.4.4.3. Ієрархічні мережі типу ART

Подальший розвиток нейронних мереж типу ART привів до побудови модульних структур, кожен з модулів у яких є тривіальною мережею ART. Відомі структури з послідовним з'єднанням модулів [109,31,32], які отримали назву ієрархічних ART-мереж. В залежності від структури міжмодульних зв'язків розрізняють дві моделі ієрархічних ART-мереж: об'єднуючі (ARTO) та розділяючі (ARTR). Ієрархічні мережі реалізують процес розпізнавання вхідних образів шляхом формування ієрархії кластерів. Для ARTO-мереж величина порога подібності r обернено пропорційна до номера модуля. Тому перший модуль завжди формує потенційно велику кількість кластерів, що узагальнюються в наступних модулях. Оберненою до даної є стратегія ARTR-мереж. Для них величина порога новизни

збільшується пропорційно до номера модуля, що призводить до формування спочатку найбільш загальних кластерів з подальшим їх уточненням в наступних модулях.

Формально ієрархічні ART-мережі задають кортежем параметрів

$$I = (k, n, m, \beta, r, \eta), \quad (1.59)$$

де значення параметрів визначаються наступним чином:

k — кількість модулів;

$n = (n_1, n_2, \dots, n_k)$ — кількість нейронів у шарах F_0 та F_1 ;

$m = (m_1, m_2, \dots, m_k)$ — кількість нейронів у шарі F_2 ;

$\beta = (\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_k)$, $0 < \beta_i < 1$ — параметри вибору;

$r = (r_1, r_2, \dots, r_k)$ — пороги подібності;

$\eta = (\eta_1, \eta_2, \dots, \eta_k)$ — коефіцієнти навчання.

ARTO-мережа, що складається з k послідовно з'єднаних модулів, показана на рис.1.28.

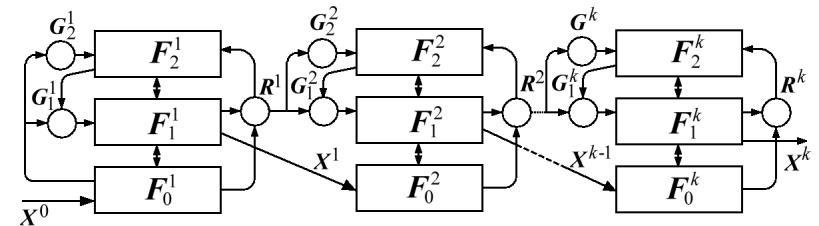


Рис.1.28. Нейронна мережа типу ARTO.

Локальні дані, виходячи з принципів роботи ART-мережі, надходять на вхід модуля з номером $1 < l \leq k$ і визначаються у відповідності до виразу:

$$X^l = X^{l-1} \cap w_{l-1,J}, \quad (1.60)$$

де $w_{l-1,J}$ — вектор вагових коефіцієнтів нейрона-переможця шару F_2^{l-1} .

В модулі l корекція знайденого прототипу відбувається згідно з формулою (1.55). Тому при $\eta = 1$ одержуємо

$$w_{l,J} = X^l \cap w_{l,J}. \quad (1.61)$$

Поширюючи залежності (1.60) та (1.61) на модулі $l-1, l-2, \dots, 2$, одержимо

$$w_{l,J} = X^1 \cap w_{1,J} \cap w_{2,J} \cap \dots \cap w_{l,J}, \quad (1.62)$$

де J — номер нейрона-переможця у модулі l .

З (1.61) та (1.62) одержуємо

$$w_{l,J} = w_{l-1,J} \cap w_{l,J}, \quad (1.63)$$

тобто ARTO-мережа формує узагальнені ієрархічні категорії на більш високих рівнях.

Протилежні функції виконує розподільна ієрархічна ARTR-мережа, зображена на рис.1.29.

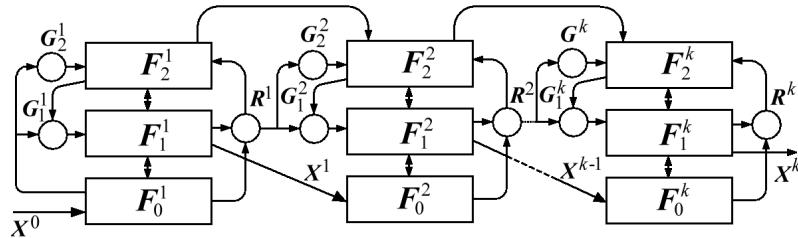


Рис.1.29. Нейронна мережа типу ARTR.

В такій мережі тільки відмінність між вхідним образом та його прототипом передається на наступний модуль ієрархічної структури. Зв'язки між модулями забезпечують чотири множини з'єднань, принципово новою з яких є множина V . Асоціативне з'єднання V , що має назву ієрархічної пам'яті, призначене для встановлення зв'язків між кластером та його одночасно

існуючими на мережі підкластерами. Якщо в першому модулі активований кластер, то в наступних модулях можуть бути активовані тільки підкластери даного кластера.

Локальний вхід модуля l визначається відповідно до виразу:

$$X^l = X^{l-1} - X^{l-1} \cap w_{l-1,J}, \quad l = 2, \dots, k. \quad (1.64)$$

Виходячи з того, що ми розглядаємо дискретне представлення вхідного образу та прототипу, (1.64) запишемо у вигляді:

$$X^l = X^{l-1} \cap \bar{w}_{l-1,J}, \quad (1.65)$$

де $\bar{w}_{l-1,J}$ є логічним запереченням $w_{l-1,J}$. Після рекурсивного застосування (1.65) до шарів $l, l-1, \dots, 2$ одержимо вираз для вхідного образу шару $l+1$:

$$\begin{aligned} X^{l+1} &= X^1 \cap \bar{w}_{1,J} \cap \dots \cap \bar{w}_{l,J} = \\ &= X^1 \cap (\bar{w}_{1,J} \cup \dots \cup \bar{w}_{l,J}) \quad l = 1, \dots, k. \end{aligned} \quad (1.66)$$

Позначимо глобальний прототип категорії J на рівні l як $W_{l,J}$. Тоді:

$$W_{l,J} = w_1 \cup w_2 \cup \dots \cup w_k, \quad l = 1, \dots, k. \quad (1.67)$$

Отже, у випадку ARTR-мережі, глобальний прототип є розділеним між її модулями, кожен з яких забезпечує формування підкластера на своєму ієрархічному рівні.

Розділ 2

МЕТОДИ НАВЧАННЯ З УЧИТЕЛЕМ

Як згадувалось у попередньому розділі, існує велика кількість сфер застосування сучасних штучних нейронних мереж. Найбільш поширеними є задачі розпізнавання образів, задачі класифікації та адаптивного управління. Швидкими темпами розвиваються нейронні мережі для реалізації адаптивних обчислювальних алгоритмів. Кожен з цих випадків застосування може розглядатися у загальному вигляді як перетворення вхідного векторного простору у вихідний. Навчання у такому контексті є встановленням правил згаданого перетворення. Існують три підходи до встановлення таких правил, які отримали назву методів навчання з учителем, навчання без учителя та навчання з підкріпленням. Кожен з них можна розглядати як процес, в ході якого відбувається адаптація вільних параметрів нейронної мережі шляхом моделювання поведінки зовнішнього середовища.

Навчання з учителем базується на використанні послідовності навчальних вибірок, що задають бажані значення вихідних векторів нейронної мережі для відповідних значень вхідних векторів. При навчанні без учителя замість навчальних вибірок використовується попередній досвід функціонування нейронної мережі. Але як у першому, так і у другому випадку єдиним залишається механізм навчання, який полягає в модифікації вагових коефіцієнтів нейронів.

Основним критерієм оцінки ефективності навчання є цільова функція, що дає функціональний опис того, наскільки модельний вихід співпадає з ідеальним для даної вхідної вибірки. Цільову функцію для методів навчання з учителем ще називають похибкою вихідного вектора ШНМ.

2.1. Принципи формування цільової функції

Нехай дана навчальна вибірка $\Psi = \left\{ \left(\mathbf{X}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^N$, що містить N пар: $\mathbf{X}^{(i)}, \mathbf{y}^{(i)}$ відповідно вхідних і вихідних векторів, а формування вихідного вектора відбувається у відповідності до виразу:

$$\mathbf{y}^{(i)} = f(\mathbf{X}^{(i)}) + \varepsilon, \quad (2.1)$$

де ε — складова білого шуму з відхиленням σ_ε^2 .

Задані умови визначають процес навчання, який полягає в пошуку найкращого сімейства функцій Φ та вибору найкращої функції з параметром \mathbf{W} , що входить до даного сімейства $\varphi_{\mathbf{W}}(\mathbf{X}^{(i)}) \in \Phi$. Критерієм такого вибору є цільова функція $E[\varphi_{\mathbf{W}}(\mathbf{X}^{(i)}, \mathbf{y}^{(i)})]$, яка визначає „відстань” між результатом дії апроксимуючої функції $\varphi_{\mathbf{W}}(\mathbf{X}^{(i)})$ та заданим навчальною вибіркою вихідним елементом даних $\mathbf{y}^{(i)}$.

Найпростішим сімейством апроксимуючих функцій є лінійне сімейство, що в загальному випадку задають виразом:

$$\varphi_{\mathbf{W}}(\mathbf{X}^{(i)}) = \mathbf{w}_0 + \sum_{j=1}^m \mathbf{w}_j x_j^{(i)} = \sum_{j=0}^m \mathbf{w}_j x_j^{(i)} = \mathbf{W}^T \mathbf{X}^{(i)}. \quad (2.2)$$

В (2.2) конструктивно введено член $x_0 = 1$ з метою спрощення виразу.

В ролі апроксимуючого сімейства широко використовують поліноми степеня p :

$$\varphi_{\mathbf{W}}(\mathbf{X}^{(i)}) = \mathbf{w}_0 + \sum_{j=1}^p \mathbf{w}_j (x^{(i)})^j = \sum_{j=0}^p \mathbf{w}_j (x^{(i)})^j. \quad (2.3)$$

В загальному вигляді лінійне апроксимуюче сімейство задамо виразом:

$$\varphi_W(\mathbf{X}^{(i)}) = \mathbf{w}_0 + \sum_{j=1}^m \mathbf{w}_j g_j(\mathbf{X}^{(i)}), \quad (2.4)$$

де $g_j(\mathbf{X}^{(i)})$ — деяка визначена на даній множині аргументів функція.

В залежності від типу задач використовують різні способи визначення цільової функції $E[\varphi_W(\mathbf{X}^{(i)}), \mathbf{y}^{(i)}]$. Широко поширена середньоквадратична цільова функція, яку задають таким чином:

$$E = \sum_{i=1}^N [\varphi_W(\mathbf{X}^{(i)}) - \mathbf{y}^{(i)}]^2. \quad (2.5)$$

Подібною до неї за оцінкою є цільова функція максимальної імовірності:

$$\begin{aligned} E &= -\ln P(\Psi|W) = -\ln \left[\prod_{i=1}^N p(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}|W) \right] = \\ &= -\sum_{i=1}^N \ln [p(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}|W)], \end{aligned} \quad (2.6)$$

де $p(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}|W)$ — імовірність існування вибірки $\{\mathbf{X}^{(i)}, \mathbf{y}^{(i)}\}$ для матриці параметрів W . У випадку нормального закону розподілу маємо:

$$p(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}|W) = \frac{1}{\sigma_\varepsilon \sqrt{2\pi}} e^{-\frac{[\varphi_W(\mathbf{X}^{(i)}) - \mathbf{y}^{(i)}]^2}{2\sigma_\varepsilon^2}}. \quad (2.7)$$

Підставивши (2.7) в (2.6), одержимо:

$$\begin{aligned} E &= -\ln P(\Psi|W) = \\ &= \ln \sigma_\varepsilon + \frac{1}{2} \ln 2\pi + \frac{1}{2\sigma_\varepsilon^2} \sum_{i=1}^N [\varphi_W(\mathbf{X}^{(i)}) - \mathbf{y}^{(i)}]^2. \end{aligned} \quad (2.8)$$

Після вилучення з (2.8) всіх членів, які не залежать від параметра W , одержимо цільову функцію, яка еквівалентна (2.5).

Відома також ентропійна цільова функція:

$$\begin{aligned} E &= \sum_{i=1}^N \left\{ \frac{1}{2} [1 + \varphi_W(\mathbf{X}^{(i)})] \log \frac{1 + \varphi_W(\mathbf{X}^{(i)})}{1 + \mathbf{y}^{(i)}} + \right. \\ &\quad \left. + \frac{1}{2} [1 - \varphi_W(\mathbf{X}^{(i)})] \log \frac{1 - \varphi_W(\mathbf{X}^{(i)})}{1 - \mathbf{y}^{(i)}} \right\}. \\ E &= \sum_{i=1}^N \left\{ \frac{1}{2} [1 + \varphi_W(\mathbf{X}^{(i)})] \log \frac{1 + \varphi_W(\mathbf{X}^{(i)})}{1 + \mathbf{y}^{(i)}} + \right. \\ &\quad \left. + \frac{1}{2} [1 - \varphi_W(\mathbf{X}^{(i)})] \log \frac{1 - \varphi_W(\mathbf{X}^{(i)})}{1 - \mathbf{y}^{(i)}} \right\}. \end{aligned}$$

2.2. Методи навчання одношарових ШНМ прямого поширення

2.2.1. Лінійна регресія

Лінійна апроксимуюча функція (2.2) характерна для одношарових ШНМ. Виходячи з цього, застосовують метод лінійної регресії, який є аналітичним методом пошуку мінімуму середньоквадратичної цільової функції (2.5).

Задамо процес функціонування одношарової ШНМ виразом:

$$\mathbf{y}^{(i)} = \mathbf{W}_{\text{var}}^T \mathbf{X}^{(i)} + \varepsilon. \quad (2.9)$$

Застосувавши лінійну апроксимуючу функцію (2.2), отримаємо:

$$\tilde{\mathbf{y}}^{(i)} = \mathbf{W}^T \mathbf{X}^{(i)}. \quad (2.10)$$

Таким чином, метою методу лінійної регресії стає визначення умов, за яких $W^T = W_{\text{var}}^T$. Для досягнення цієї рівності розглянемо можливість мінімізації цільової функції на навчальній вибірці Ψ , яка обумовлює наявність вхідної структури даних у вигляді матриці, кожен рядок якої містить елементи вхідного вектора, а кількість рядків дорівнює кількості навчальних вибірок

$$X = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{pmatrix} = \begin{pmatrix} X^{(1)} \\ X^{(2)} \\ \dots \\ X^{(N)} \end{pmatrix}, \quad (2.11)$$

вихідного вектора

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(N)} \end{pmatrix} \quad (2.12)$$

та наближеного вихідного вектора ШНМ $\tilde{y} = XW$.

Процес навчання одношарової ШНМ полягає в мінімізації цільової функції:

$$E = \sum_{i=0}^N (y^{(i)} - \tilde{y}^{(i)})^2. \quad (2.13)$$

Тоді середньоквадратична цільова функція матиме вигляд:

$$E = |y - \tilde{y}|^T |y - \tilde{y}| = W^T X^T XW - 2W^T X^T y + y^T y. \quad (2.14)$$

З метою визначення екстремуму цієї функції прирівнюємо до нуля першу похідну:

$$\frac{\partial E}{\partial W} = 2X^T XW - 2X^T y = 0. \quad (2.15)$$

З виразу (2.15) визначимо параметр W :

$$W = (X^T X)^{-1} Xy. \quad (2.16)$$

Оскільки в загальному випадку матриця X неквадратична, то $(X^T X)^{-1} X = X^\circ$ є псевдоінверсною матрицею. Тоді аналітичний вираз для визначення параметра W

$$W = X^\circ y. \quad (2.17)$$

У випадку виродженої матриці $X^T X$ застосовують заміну:

$$X^T X \rightarrow (X^T + \lambda I)X, \quad (2.18)$$

яка виконує функцію регуляризації матриці шляхом додавання до головної діагоналі невеликих чисел. Тоді розв'язок (2.16) матиме вигляд:

$$W = (X^T X + \lambda I)^{-1} X^T y. \quad (2.19)$$

Квадратична матриця W може бути інвертована, а одержані зайві власні значення, якщо вони менші, ніж λ , відфільтровуються. Такий підхід називають гребеневою регресією [176]. Застосування методів лінійної регресії має швидше теоретичний інтерес, оскільки практичне використання стикається з величезними обчислювальними труднощами. Традиційний спосіб подолання даної проблеми полягає в розробці ітераційних підходів, одним з яких є метод градієнтного спуску.

2.2.2. Метод градієнтного спуску

Ідея методу градієнтного спуску полягає в послідовній зміні параметрів ШНМ в напрямку, що забезпечує зменшення цільової функції E . Оскільки функція E диференційовна по кожному з параметрів, то існує можливість обчислення градієнтного вектора. Рухаючись у напрямку від'ємного градієнта по кожному з параметрів, знаходимо локальні мінімуми цільової функції. На рис.2.1 зображений принцип зміни параметра за методом градієнтного спуску.

Така зміна виражається формулою:

$$\begin{aligned} \Delta W &= -\eta \frac{\partial E}{\partial W} = \eta \sum_{i=1}^N (y^{(i)} - \tilde{y}^{(i)}) X^{(i)} = \\ &= \eta \sum_{i=1}^N (y^{(i)} - W^T X^{(i)}) X^{(i)}, \end{aligned} \quad (2.20)$$

де η — коефіцієнт навчання.

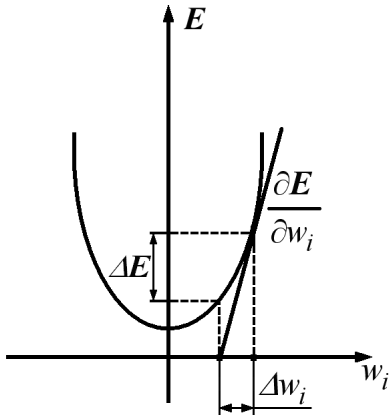


Рис.2.1. Зміна параметра за методом градієнтного спуску.

На рис.2.2 зображено блок-схему алгоритму, що побудований на базі методу градієнтного спуску. Цей алгоритм називають алгоритмом пакетного типу, тому що для визначення величини кроку зміни параметра необхідно провести обробку всієї навчальної вибірки.

У випадку стохастичної природи навчальної вибірки, виходячи з рівняння Вінера $R_{xx}W - r_{xy} = 0$ та провівши аналогію з (2.15), визначимо кореляційну матрицю R_{xx} і взаємно кореляційний вектор r_{xy} . Оскільки довільний елемент матриці

$$[R_{xx}]_{kl} = \frac{1}{N} \sum_{i=1}^N x_k^{(i)} x_l^{(i)},$$

а елемент вектора

$$[r_{xy}]_k = \frac{1}{N} \sum_{i=1}^N x_k^{(i)} y^{(i)},$$

то одержимо:

$$R_{xx} = \frac{1}{N} X^T X, \quad r_{xy} = \frac{1}{N} X^T y.$$

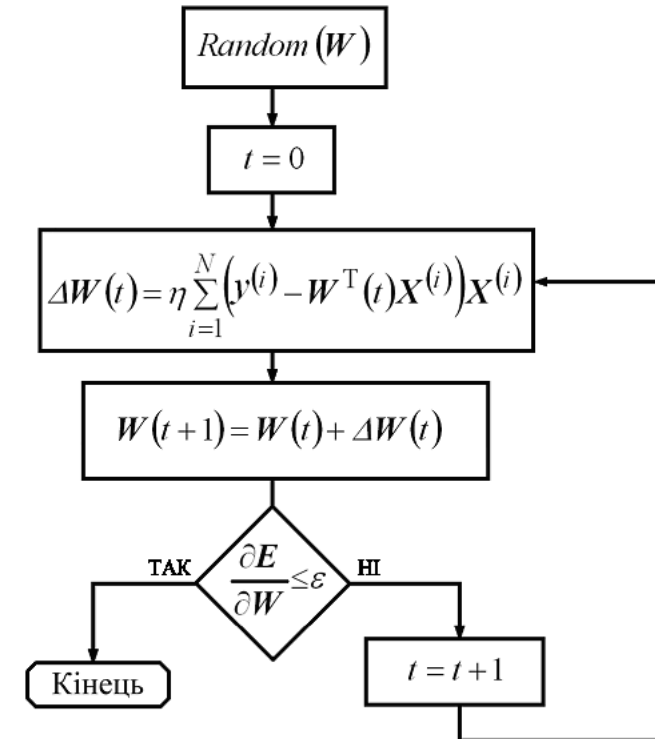


Рис.2.2. Алгоритм градієнтного спуску.

З використанням даної аналогії формула для визначення кроку зміни параметра набуде вигляду:

$$\Delta W = -\eta \frac{\partial E}{\partial W} = \eta (r_{xy} - R_{xx} W). \quad (2.21)$$

Практика показує високу стійкість даного алгоритму, що забезпечує незначні коливання модуля цільової функції за умови зміни характеру вхідних даних. Але невелика швидкість збіжності обумовлює необхідність значних затрат часу на навчання ШНМ. Основним параметром, що впливає на швидкість збіжності алгоритму градієнтного спуску, є коефіцієнт навчання η . Відповідно до методу, модифікація цього параметра може бути проведена тільки після обробки всієї вибірки Ψ . Швидкість збіжності також може бути збільшена у випадку, коли відома попередня оцінка кореляційних функцій $R_{xx}(t)$ та $r_{xy}(t)$. Але потреба в такій оцінці не впливає вже із самої постановки задачі, оскільки відомі значення $R_{xx}(t)$ та $r_{xy}(t)$ дають можливість аналітичного розв'язання рівняння (2.17), що виключає необхідність використання ітераційних методів.

2.2.3. Метод найменших квадратів

Метод найменших квадратів (Last Mean Square)[181] розроблений з метою покращення швидкості збіжності алгоритму шляхом підвищення динаміки регулювання коефіцієнта η . Він був уперше застосований для нейронної мережі Adaline [179] з лінійною активаційною функцією нейронів, а потім узагальнений для довільних ШНМ прямого поширення [178]. Динамічний характер методу полягає в тому, що процес корекції кроку параметра може відбуватися не після обробки всієї вибірки, як у методі градієнтного спуску, а після обчислення кожного елемента вибірки. На рис. 2.3 показана

блок-схема алгоритму найменших квадратів, який має необмежену навчальну вибірку.

Такий алгоритм називають алгоритмом реального часу. Критерієм закінчення алгоритму навчання є досягнення мінімуму цільової функції. Швидкість збіжності алгоритму критично залежить від величини кроку параметра, який регулюється коефіцієнтом навчання після кожної ітерації.

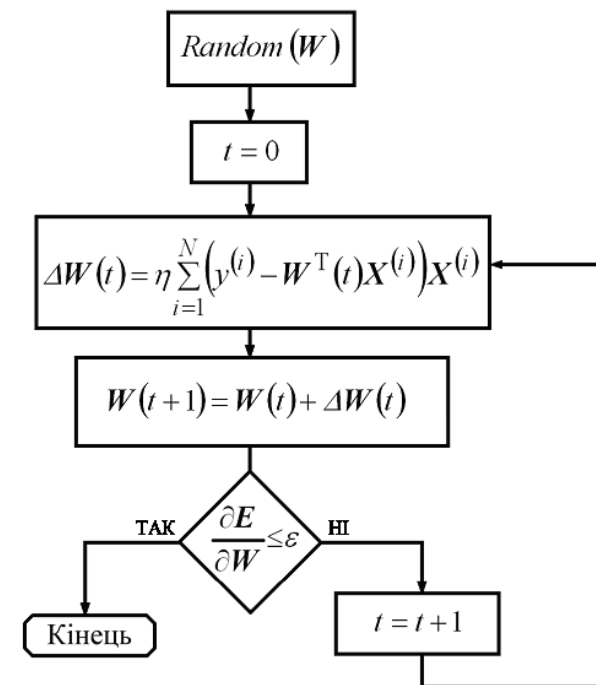


Рис.2.3. Алгоритм найменших квадратів.

При малих значеннях η зменшується швидкість збіжності алгоритму, а при надто великих — алгоритм розбіжний.

В залежності від конкретних умов визначають різні обмеження для діапазону зміни η . Для збереження збіжності алгоритму найменших квадратів необхідна умова:

$$0 < \eta < \frac{2}{\lambda_{\max}}, \quad (2.22)$$

де λ_{\max} — максимальне власне значення кореляційної матриці \mathbf{R}_{xx} .

Верхнє обмеження для λ_{\max} можна задати у вигляді $\lambda_{\max} < \text{Tr}(\mathbf{R}_{xx})$, де $\text{Tr}(\mathbf{R}_{xx}) = \sum_k \rho_{kk}$ — слід матриці \mathbf{R}_{xx} .

Тому відомі також оцінки діапазону зміни коефіцієнта навчання η , що базуються на величині сліду кореляційної матриці:

$$0 < \eta < \frac{2}{\text{Tr}(\mathbf{R}_{xx})} = \frac{2}{\text{Tr}(\mathbf{X}^T \mathbf{X})}. \quad (2.23)$$

Найбільш перспективною стратегією вибору коефіцієнта $\eta(i)$ на i -му ітераційному кроці є балансування на верхній межі даного обмеження, що дає можливість максимально збільшити швидкість збіжності алгоритму. Але визначення такої оцінки потребує більших обчислювальних ресурсів, ніж ті, що витрачаються на виконання ітерацій.

З цієї причини найчастіше використовують евристичні залежності, на зразок тієї, що наведена в блок-схемі рис. 2.3.

2.2.4. Метод навчання Adaline та персептрона Розенблатта

Завдяки застосуванню порогової активаційної функції штучні нейронні мережі Adaline та персептрон Розенблатта здатні класифікувати вхідні сигнали за двома класами. Вектори вхідних сигналів, які зумовлюють нижнє значення вихідного сигналу, зараховують до класу A , а вхідні вектори, які визначають його верхнє значення, формують клас B .

Методи навчання ШНМ з пороговими активаційними функціями відрізняються від уже розглянутих, що зумовлено формою активаційної функції.

Одношарова ШНМ Adaline (ADaptive LINear Element) використовує наступну порогову активаційну функцію:

$$y^{(i)} = \begin{cases} +1, & \mathbf{X}^{(i)} \in A, \\ -1, & \mathbf{X}^{(i)} \in B. \end{cases} \quad (2.24)$$

Наближене значення вихідного вектора в процесі навчання визначають за формулою:

$$\tilde{y}^{(i)} = \text{sign}(\mathbf{W}^T \mathbf{X}^{(i)}) = \begin{cases} +1, & \mathbf{W}^T \mathbf{X}^{(i)} \geq 0, \\ -1, & \mathbf{W}^T \mathbf{X}^{(i)} < 0. \end{cases} \quad (2.25)$$

Одношаровий персептрон Розенблатта використовує бінарну функцію:

$$y^{(i)} = \begin{cases} 1, & \mathbf{X}^{(i)} \in A, \\ 0, & \mathbf{X}^{(i)} \in B. \end{cases} \quad (2.26)$$

Відповідно вихідний навчальний вектор формується за функцією Хевісайда:

$$\tilde{y} = \Theta(\mathbf{W}^T \mathbf{X}^{(i)}) = \begin{cases} 1, & \mathbf{W}^T \mathbf{X}^{(i)} \geq 0, \\ 0, & \mathbf{W}^T \mathbf{X}^{(i)} < 0. \end{cases} \quad (2.27)$$

В роботі [160] запропонований один із перших методів навчання з учителем, що дозволяє забезпечити формування вихідних класів. Для кожного значення вхідного вектора порівнюють вихідний сигнал з відповідним класом вхідного вектора та визначають зміну параметра за формулою:

$$\Delta w_j = \begin{cases} 0, & (X^{(i)} \in A, \tilde{y}^{(i)} = -1), (X^{(i)} \in B, \tilde{y}^{(i)} = +1), \\ +\varepsilon x_j^{(i)}, & (X^{(i)} \in B, \tilde{y}^{(i)} = -1), \\ -\varepsilon x_j^{(i)}, & (X^{(i)} \in A, \tilde{y}^{(i)} = +1), \end{cases} \quad (2.28)$$

$$i = \overline{1, N}; j = \overline{1, m}.$$

Блок-схема алгоритму, яка ілюструє одну з можливих послідовностей дій по навчанню нейрона, показана на рис.2.4.

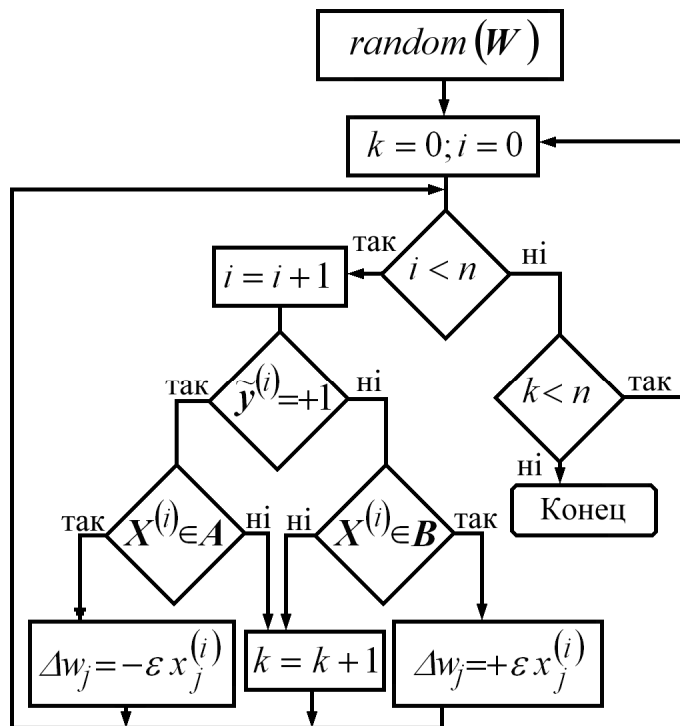


Рис.2.4. Алгоритм Розенблатта.

В роботі [160] доведено, що у випадку лінійної роздільності класів A та B даний метод збіжний і правильно класифікує вхідні вектори за скінченну кількість кроків. Швидкість збіжності регулюється коефіцієнтом $\varepsilon > 0$. Якщо не існує гіперплощини, яка б розділяла класи A та B , то виникає нескінченна осциляція значень вагових коефіцієнтів, що є істотним недоліком даного методу.

2.2.5. Метод „кишені”

Метод, що не допускає зацикловання алгоритму, запропонований у [82]. Метою даного методу є досягнення максимально можливої класифікації у випадку, коли не існує лінійної роздільності заданих класів. Графічно функціонування методу можна зобразити як побудову найкращої з можливих гіперплощин, які розділяють задані класи. В результаті такої побудови значення вхідного вектора, які не можуть бути розділеними, залишаються на протилежній стороні гіперплощини.

З огляду на принцип роботи даного методу, він отримав назву „метод кишені”. Основна відмінність нового підходу полягає в тому, що після кожного проходу алгоритму, побудованого за даним методом, порівнюють значення параметра $k^{(i)}$ зі значенням $k^{(i-1)}$, яке було одержане на попередньому етапі. У випадку, коли $k^{(i)} > k^{(i-1)}$, відповідне значення вектора W кладуть до „кишені”, тобто запам’ятовують. Таким чином, у нашому розпорядженні завжди знаходиться вектор вагових коефіцієнтів, який забезпечує кращий варіант класифікації серед досягнутих на даний момент. Недоліком методу є необхідність проведення значної кількості проходів алгоритму, щоб отримати найкращий спосіб класифікації.

2.2.6. Методи градієнтного спуску та найменших квадратів для Adaline та перцептрона Розенблатта

Єдиною причиною, що утруднює використання методів градієнтного спуску і найменших квадратів для навчання Adaline та перцептрона Розенблатта, є недиференційовність порогових функцій. Ця проблема може бути розв'язана шляхом введення заміन:

$$\mathbf{v}^{(i)} = \mathbf{W}^T \mathbf{X}^{(i)},$$

$$\tilde{y}^{(i)} = \text{sign}(\mathbf{v}^{(i)}) \rightarrow \tilde{y}^{(i)} = \tanh(\mathbf{v}^{(i)}) = \frac{e^{\mathbf{v}^{(i)}} - e^{-\mathbf{v}^{(i)}}}{e^{\mathbf{v}^{(i)}} + e^{-\mathbf{v}^{(i)}}}, \quad (2.29)$$

$$\tilde{y}^{(i)} = \Theta(\mathbf{v}^{(i)}) \rightarrow \tilde{y}^{(i)} = \frac{1}{1 + e^{-\mathbf{v}^{(i)}}}. \quad (2.30)$$

Відповідно до формули (2.20) крок зміни параметра за методом градієнтного спуску для Adaline матиме вигляд:

$$\begin{aligned} \Delta \mathbf{W} &= -\eta \frac{\partial \mathbf{E}}{\partial \mathbf{W}} = \\ &= 2\eta \sum_{i=1}^N \left[\tanh^3(\mathbf{v}^{(i)}) - y^{(i)} \tanh^2(\mathbf{v}^{(i)}) - \tanh(\mathbf{v}^{(i)}) + y^{(i)} \right] \mathbf{X}^{(i)}, \end{aligned} \quad (2.31)$$

а крок зміни параметра для перцептрона:

$$\Delta \mathbf{W} = -\eta \frac{\partial \mathbf{E}}{\partial \mathbf{W}} = 2\eta \sum_{i=1}^N \frac{y^{(i)} e^{-\mathbf{v}^{(i)}} (1 + e^{-\mathbf{v}^{(i)}})^2 - e^{-\mathbf{v}^{(i)}}}{(1 + e^{-\mathbf{v}^{(i)}})^3} \mathbf{X}^{(i)}. \quad (2.32)$$

Аналогічно можна вивести формули для обчислення кроку зміни параметра за методом найменших квадратів [177].

Пошук мінімуму цільової функції, як і у попередніх методах навчання одношарових ШНМ прямого поширення, інтерпретується як вирішення проблеми знаходження оптимальної гіперплощини, яка розділяє класи.

2.3. Методи навчання багатошарових ШНМ прямого поширення

Багатошарові ШНМ можуть розглядатися як послідовне з'єднання одношарових ШНМ прямого поширення. Структуру вагових коефіцієнтів у цих мережах організують таким чином, щоб класи більш складної форми оброблялися на шарах нейронів високих рівнів шляхом об'єднання та перетину простих класів, які формуються на нижчих рівнях ШНМ. Відомі строгі доведення того факту, що двошарова ШНМ здатна розпізнати довільний клас опуклої форми за умови, що існує можливість використати достатню кількість нейронів прихованого шару, а вагові коефіцієнти налаштовані належним чином [82]. ШНМ прямого поширення з кількома прихованими шарами потенційно здатна до розпізнавання класів довільної форми. Отже, постановка задачі на ШНМ прямого поширення включає визначення мінімально можливої кількості нейронів прихованого шару та вибір ефективного методу настройки вагових коефіцієнтів. На сьогодні обидві ці проблеми не є тривіальними.

Для пояснення основних принципів побудови методів навчання з учителем будемо розглядати двошарову ШНМ, показану на рис.2.5. Нульовий шар цієї мережі виконує допоміжну функцію розгалуження сигналів і не містить нейронів. З цієї причини його робота не приводить до модифікації вхідного вектора. Останній шар ШНМ називають вихідним шаром. Всі шари, що розташовані між нульовим та вихідним, є прихованими шарами з нелінійною активаційною

функцією нейронів. На даному прикладі будемо розглядати один прихований шар з m нейронами, які використовують у ролі активаційної функції гіперболічний тангенс.

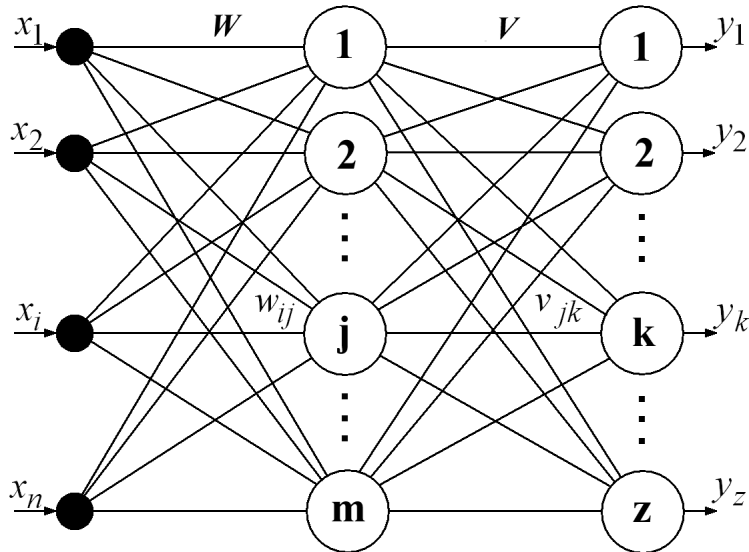


Рис.2.5. Узагальнена багатощарова ШНМ прямого поширення.

Роботу даної ШНМ можна описати такими рівняннями:

$$\tilde{y} = \varphi[s(x)], \quad (2.33)$$

$$s(x) = v_0 + \sum_{j=1}^m v_j h_j, \quad (2.34)$$

$$h_j = f(t_j) = f\left(w_{0j} + \sum_{i=1}^n w_{ij} x_i\right), \quad (2.35)$$

$$t_j = w_{0j} + \sum_{i=1}^n w_{ij} x_i. \quad (2.36)$$

де i, j — індекси елемента вхідного вектора та нейрона прихованого шару відповідно; x_i — елемент вхідного вектора; t_j — аргумент активаційної функції нейрона прихованого шару; h_j — елемент вихідного вектора прихованого шару; s — аргумент активаційної функції нейрона вихідного шару; \tilde{y} — вихідний вектор ШНМ; $f(t)$ — активаційна функція нейронів прихованого шару; $\varphi(s)$ — активаційна функція нейронів вихідного шару.

2.3.1. Метод зворотного поширення

Нехай дано навчальну вибірку $\Psi = \left\{ \left(\mathbf{X}^{(n)}, \mathbf{y}^{(n)} \right) \right\}_{n=1}^N$, що містить N пар: $\mathbf{X}^{(n)}$, $\mathbf{y}^{(n)}$ відповідно вхідних і вихідних векторів та множину параметрів $\mathbf{W} = \{\mathbf{w}, \mathbf{v}\}$, яка складається з параметрів нейронів прихованого шару \mathbf{w} і параметрів вихідного шару \mathbf{v} . Метод зворотного поширення полягає в мінімізації цільової функції:

$$E = \frac{1}{2N} \sum_{n=1}^N \left[\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}(x) \right]^2. \quad (2.37)$$

Для параметра v_j одержимо

$$\frac{\partial E}{\partial v_j} = -\frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)} \right) \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial v_j} =$$

$$\begin{aligned}
&= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) \frac{\partial \mathbf{s}^{(n)}}{\partial v_j} = \\
&= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) h_j. \quad (2.38)
\end{aligned}$$

Крок зміни вагових коефіцієнтів вихідного шару дорівнює:

$$\Delta v_j = -\eta \frac{\partial \mathbf{E}}{\partial v_j} = \frac{\eta}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) h_j. \quad (2.39)$$

Визначимо частинну похідну по w_{ij} :

$$\begin{aligned}
\frac{\partial \mathbf{E}}{\partial w_{ij}} &= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_{ij}} = \\
&= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) v_j \frac{\partial h_j^{(n)}}{\partial w_{ij}} = \\
&= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) v_j f'(t_j^{(n)}) \frac{\partial t_j^{(n)}}{\partial w_{ij}} = \\
&= -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) v_j f'(t_j^{(n)}) x_i^{(n)}. \quad (2.40)
\end{aligned}$$

Крок зміни вагових коефіцієнтів прихованого шару:

$$\Delta w_{ij} = -\eta \frac{\partial \mathbf{E}}{\partial w_{ij}} = \frac{\eta}{N} \sum_{n=1}^N (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}) v_j f'(t_j^{(n)}) x_i^{(n)}. \quad (2.41)$$

Зробивши заміни:

$$\delta^{(n)} = (\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}) \varphi'(\mathbf{s}^{(n)}), \quad (2.42)$$

$$\delta_j^{(n)} = \delta^{(n)} f'(t_j^{(n)}), \quad (2.43)$$

спростимо формули для визначення кроків зміни параметрів прихованого та вихідного шарів:

$$\Delta v_j = \frac{\eta}{N} \sum_{n=1}^N \delta^{(n)} h_j^{(n)}, \quad (2.44)$$

$$\Delta w_{ij} = \frac{\eta}{N} \sum_{n=1}^N \delta_j^{(n)} x_i^{(n)}. \quad (2.45)$$

Подібно до методів градієнтного спуску та найменших квадратів, що використовуються для одношарових ШНМ, метод зворотного поширення може використовуватись для пакетного алгоритму або алгоритму реального часу [100]. Приклад блок-схеми пакетного алгоритму зворотного поширення показаний на рис.2.6.

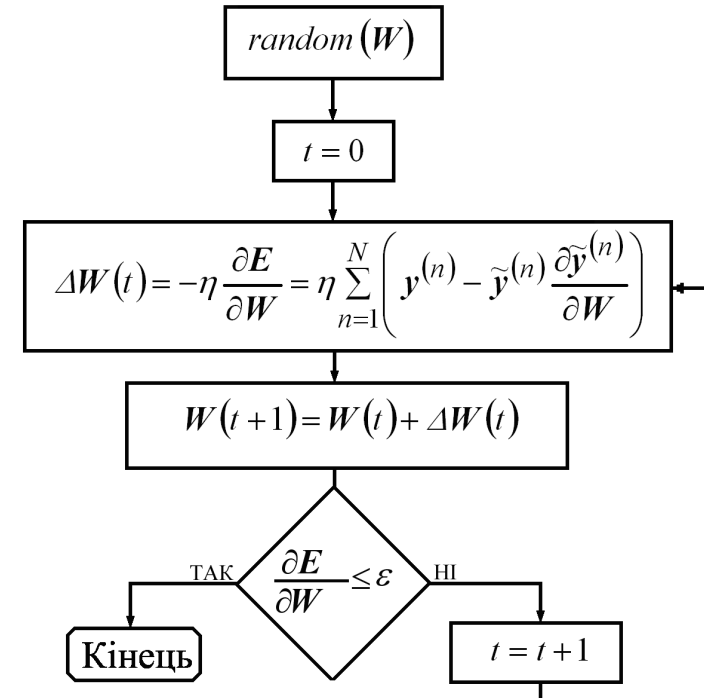


Рис.2.6. Пакетний алгоритм зворотного поширення.

Для обчислення чергового кроку параметра ΔW за даним алгоритмом необхідно виконати обробку всієї навчальної вибірки Ψ .

Ознакою закінчення процесу навчання може бути досягнення умови $\frac{\partial E}{\partial W} \leq \varepsilon$, або мінімальна зміна цільової функції

$$ABS\left(\frac{E(t+1) - E(t)}{E(t)}\right) < \varepsilon, \text{ де } \varepsilon > 0 \text{ — значення порога точності.}$$

Алгоритм реального часу дозволяє модифікацію кроку параметра після обробки кожної навчальної пари $(X^n, y^{(n)})$, що забезпечує можливість кращої адаптації алгоритму до зміни вхідної вибірки. Приклад блок-схеми алгоритму реального часу для методу зворотного поширення показаний на рис. 2.7.

Величина кроку параметра ΔW модифікується в реальному часі за формулою:

$$\Delta W^{(n)} = -\eta \frac{\partial E^{(n)}}{\partial W} + \alpha \Delta W^{(n-1)}, \quad (2.46)$$

де $0 < \alpha < 1$ — коефіцієнт впливу попередньої ітерації; $\lambda(n)$ — функція динамічного формування коефіцієнта навчання η .

Завдяки такому підходу до формування кроку параметра алгоритм може використовувати навчальну вибірку невизначеної довжини. Тому навчання за алгоритмом реального часу найчастіше використовують у структурах, що характеризуються необмеженим потоком вхідних даних.

Алгоритм реального часу дозволяє підвищити швидкість збіжності методу зворотного поширення у випадку ретельного підбору функції $\lambda(n)$. Але навіть за цієї умови метод потребує значного часу для настройки параметрів, що пояснюється незадовільною швидкістю збіжності. Незважаючи на це, метод зворотного поширення мав велике значення на початку 80-х років, оскільки завдяки йому було доведено універсальний характер ШНМ прямого поширення.

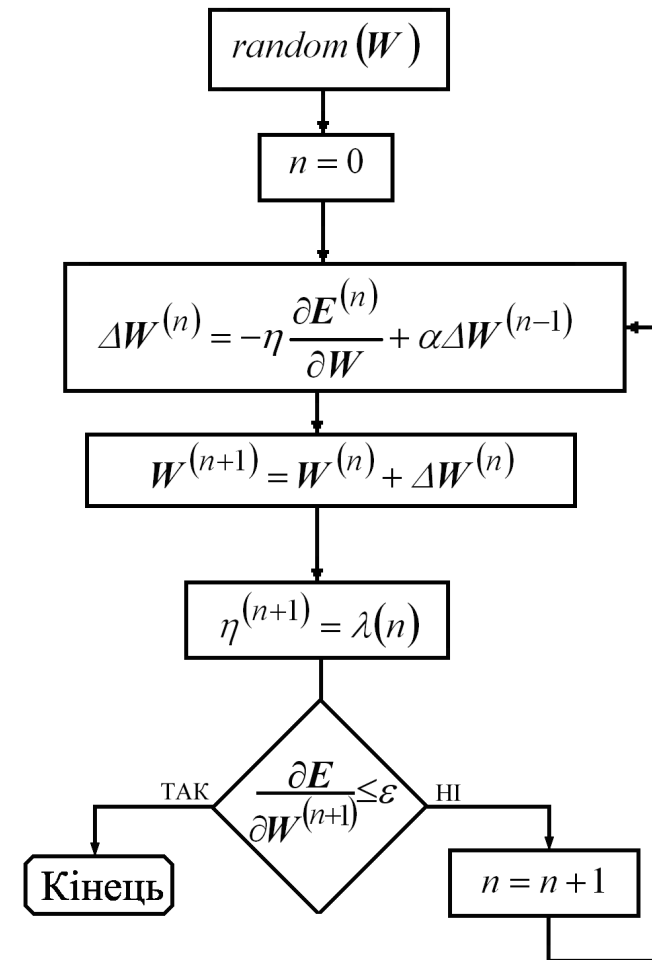


Рис.2.7. Алгоритм реального часу для методу зворотного поширення.

2.3.2. Метод еластичного зворотного поширення

Метод еластичного зворотного поширення („Resilient Backpropagation”) вперше запропонований в [158]. Характерними його особливостями є використання матриці коефіцієнтів навчання та спеціальних кроків зміни параметрів. Нехай W — матриця параметрів

$$W = \begin{pmatrix} w_{11} & \dots & w_{1j} & \dots & w_{1m} \\ \cdot & \dots & \cdot & \dots & \cdot \\ w_{i1} & \dots & w_{ij} & \dots & w_{im} \\ \cdot & \dots & \cdot & \dots & \cdot \\ w_{n1} & \cdot & w_{nj} & \cdot & w_{nm} \end{pmatrix}, \quad (2.47)$$

а матриця коефіцієнтів навчання з елементами, що задають індивідуальні коефіцієнти навчання для цих параметрів, —

$$\eta = \begin{pmatrix} \eta_{11} & \dots & \eta_{1j} & \dots & \eta_{1m} \\ \cdot & \dots & \cdot & \dots & \cdot \\ \eta_{i1} & \dots & \eta_{ij} & \dots & \eta_{im} \\ \cdot & \dots & \cdot & \dots & \cdot \\ \eta_{n1} & \cdot & \eta_{nj} & \cdot & \eta_{nm} \end{pmatrix}. \quad (2.48)$$

З використанням (2.48) будемо визначати величину зміни кожного з параметрів індивідуально за формулою:

$$\Delta w_{ij} = -\eta_{ij} \operatorname{sign} \left(\frac{\partial E}{\partial w_{ij}} \right). \quad (2.49)$$

Значення коефіцієнтів навчання η_{ij} формуються динамічно в процесі навчання відповідно до умови:

$$\eta_{ij}(t) = \begin{cases} \gamma^+ * \eta_{ij}(t-1) & \text{при } \frac{\partial E(t)}{\partial w_{ij}} * \frac{\partial E(t-1)}{\partial w_{ij}} > 0, \\ \gamma^- * \eta_{ij}(t-1) & \text{при } \frac{\partial E(t)}{\partial w_{ij}} * \frac{\partial E(t-1)}{\partial w_{ij}} < 0, \\ \eta_{ij}(t-1) & \text{при } \frac{\partial E(t)}{\partial w_{ij}} * \frac{\partial E(t-1)}{\partial w_{ij}} = 0, \end{cases} \quad (2.50)$$

де $0 < \gamma^- < 1 < \gamma^+$ — коефіцієнти зміни коефіцієнта навчання.

Даний метод реалізує правило адаптації, яке можна сформулювати наступним чином. Кожного разу, коли частинна похідна $\frac{\partial E}{\partial w_{ij}}$ для відповідного параметра w_{ij} змінює свій знак, це може бути інтерпретовано як той факт, що крок $\Delta w_{ij}(t-1)$ останньої зміни даного параметра був завеликим і спричинив перестрибування через локальний мінімум. Тому необхідно зменшити наступний крок $\Delta w_{ij}(t)$ шляхом зменшення коефіцієнта навчання в γ^- разів.

На рис 2.8 показаний принцип роботи алгоритму еластичного зворотного поширення.

Для більш докладного пояснення дії алгоритму по знаходженню w_{\min} , що відповідає значенню цільової функції $\min E$, розглянемо три послідовні фази даного процесу.

Фаза 1 починається безпосередньо перед обчисленням ітерації з номером t . Проаналізуємо чи змінюється знак частинної похідної $\frac{\partial E(t)}{\partial w_{ij}}$ у точці t порівняно з її знаком у точці

попередньої ітерації $t-1$. Відповідно до рис.2.8 $\frac{\partial E(t)}{\partial w_{ij}} > 0$ і

$\frac{\partial E(t-1)}{\partial w_{ij}} > 0$, тому збільшимо коефіцієнт навчання η_{ij} в γ^+ разів та обчислимо величину зміни параметра за формулою:

$$\Delta w_{ij} = -\eta_{ij} \operatorname{sign} \left(\frac{\partial E(t)}{\partial w_{ij}} \right),$$

$$\text{де } \operatorname{sign} \left(\frac{\partial E(t)}{\partial w_{ij}} \right) = \begin{cases} +1 & \text{при } \frac{\partial E(t)}{\partial w_{ij}} > 0, \\ -1 & \text{при } \frac{\partial E(t)}{\partial w_{ij}} < 0. \end{cases} \quad (2.51)$$

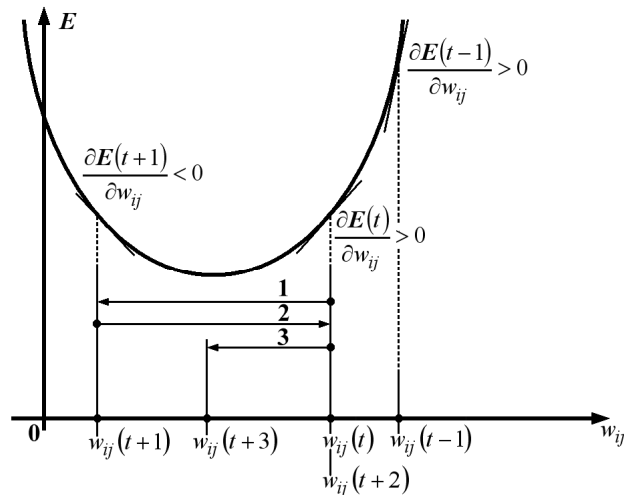


Рис.2.8. Принцип роботи алгоритму для методу еластичного зворотного поширення.

Фаза 1 завершується переходом до точки $w_{ij}(t+1)$ за формулою:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t). \quad (2.52)$$

Повторюючи у фазі 2 дії, які вже описані для фази 1, визначаємо, що в точці $w_{ij}(t+1)$ виникає зміна знаку частинної похідної цільової функції. Цей факт приводить до зменшення коефіцієнта навчання η_{ij} в γ^- разів. Далі відбувається повернення в точку $w_{ij}(t)$ шляхом виконання ітерації

$$w_{ij}(t+2) = w_{ij}(t+1) - \Delta w_{ij}(t) \quad (2.53)$$

без зміни кроку параметра. У цій фазі також прирівнюють до нуля змінну, що містить значення частинної похідної на даній ітерації, тобто встановлюють значення $\frac{\partial E(t+1)}{\partial w_{ij}} = 0$.

Фаза 3 починається з визначення того факту, що добуток частинних похідних на даній ітерації і попередній ітерації дорівнює нулю завдяки примусовому прирівнюванню до нуля частинної похідної попередньої ітерації:

$$\frac{\partial E(t+1)}{\partial w_{ij}} * \frac{\partial E(t+2)}{\partial w_{ij}} = 0. \quad (2.54)$$

Ця умова є ознакою того, що в даній ітерації немає необхідності модифікувати величину коефіцієнта навчання. Отже, формування кроку параметра $\Delta w_{ij}(t+2)$ відбувається з використанням η_{ij} , що був сформований під час фази 2. Виконуючи ітерацію

$$w_{ij}(t+3) = w_{ij}(t+2) + \Delta w_{ij}(t+2) \quad (2.55)$$

з меншим кроком та стартуючи з тієї ж точки, що і в ітерації (2.52), знаходимо:

$$w_{ij}(t+3) = (w_{ij})_{\min}. \quad (2.56)$$

Пакетний алгоритм даного методу показаний на рис. 2.9.

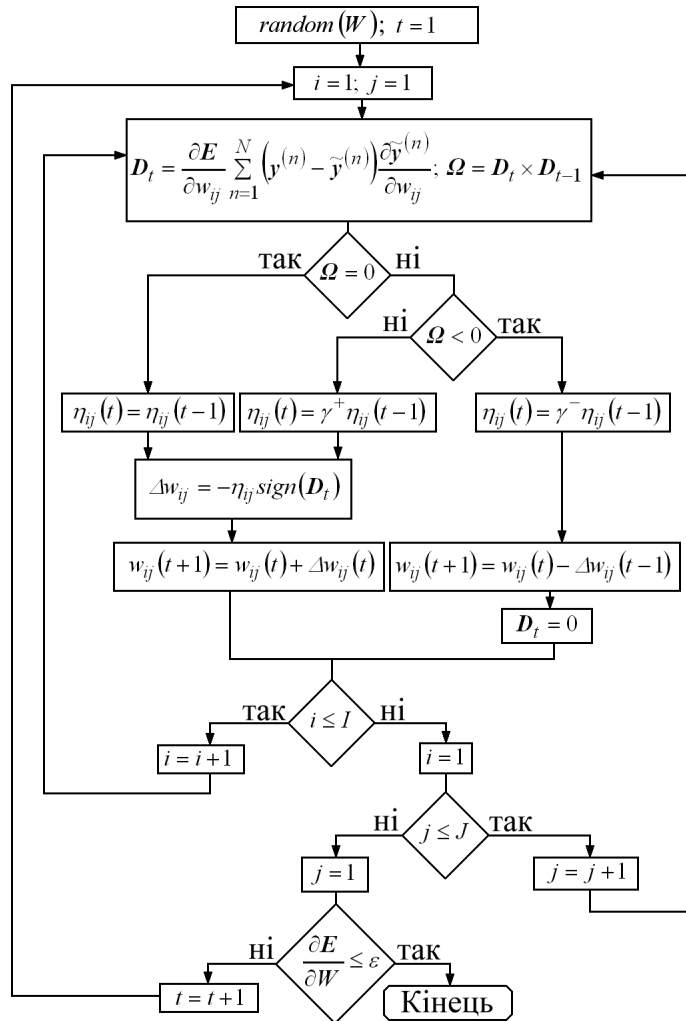


Рис.2.9. Пакетний алгоритм для методу еластичного зворотного поширення.

Невелике збільшення кількості обчислювальних операцій компенсується підвищенням швидкості збіжності методу. Деякого спрощення алгоритму можна досягти за рахунок фіксації γ^+ та γ^- . Практична реалізація алгоритму вимагає обмеження діапазону зміни коефіцієнта навчання

$$\eta_0 < \eta_{ij} < \eta_{\max}$$

з метою збереження збіжності методу. Оскільки зміна кроку $\Delta w_{ij}(t)$ носить „шумовий” характер і в межах ітерації не визначає напрямку зменшення цільової функції, то очевидно, що даний метод не може бути застосований для алгоритмів реального часу. Метод еластичного зворотного поширення забезпечує більшу швидкість збіжності, ніж більшість алгоритмів навчання ШНМ прямого поширення. Однак у деяких випадках він може збігатися до локального мінімуму, що призводить одержання невірних результатів навчання. Тому, в порівнянні з даним алгоритмом кращим є алгоритм реального часу для стандартного методу зворотного поширення.

2.3.3. Метод Гауса–Ньютона

Ідея розглянутих методів навчання ШНМ полягає у використанні похідної першого порядку для мінімізації цільової функції. Однак більш точні результати можуть бути досягнуті у випадку представлення цільової функції у вигляді ряду Тейлора з урахуванням члена цього ряду, що містить похідну другого порядку. Такий метод [33] має назву методу Гауса–Ньютона і забезпечує мінімізацію цільової функції:

$$\begin{aligned}
 E(W + \Delta W) &= \\
 &= E(W) + \left(\frac{\partial E}{\partial W} \right)^T \Delta W + \frac{1}{2} \Delta W^T \frac{\partial^2 E}{\partial W^2} \Delta W + O(\|\Delta W\|^3) \approx
 \end{aligned}$$

$$\approx \mathbf{E}(\mathbf{W}) + \left(\frac{\partial \mathbf{E}}{\partial \mathbf{W}} \right)^T \Delta \mathbf{W} + \frac{1}{2} \Delta \mathbf{W}^T \mathbf{H} \Delta \mathbf{W}, \quad (2.57)$$

де $\mathbf{W} = [w_1, w_2, w_3, \dots, w_n]^T$ — вектор параметрів ШНМ, що містить вагові коефіцієнти прихованого та вихідного шарів;

$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}} = \left[\frac{\partial \mathbf{E}}{\partial w_1}, \frac{\partial \mathbf{E}}{\partial w_2}, \frac{\partial \mathbf{E}}{\partial w_3}, \dots, \frac{\partial \mathbf{E}}{\partial w_n} \right]^T$$
 — вектор частинних похідних;

$\Delta \mathbf{W}$ — вектор приростів параметрів; \mathbf{H} — матриця Гессе похідних другого порядку.

$$\mathbf{H} = \frac{\partial^2 \mathbf{E}}{\partial \mathbf{W}^2} = \begin{bmatrix} \frac{\partial^2 \mathbf{E}}{\partial w_1^2} & \frac{\partial^2 \mathbf{E}}{\partial w_1 \partial w_2} & \frac{\partial^2 \mathbf{E}}{\partial w_1 \partial w_3} & \frac{\partial^2 \mathbf{E}}{\partial w_1 \partial w_n} \\ \frac{\partial^2 \mathbf{E}}{\partial w_n \partial w_1} & \frac{\partial^2 \mathbf{E}}{\partial w_n \partial w_2} & \frac{\partial^2 \mathbf{E}}{\partial w_n \partial w_3} & \frac{\partial^2 \mathbf{E}}{\partial w_n^2} \end{bmatrix}. \quad (2.58)$$

Мінімальне значення цільової функції одержимо шляхом прирівнювання похідної ряду Тейлора (2.57) до нуля:

$$\frac{\partial \mathbf{E}(\mathbf{W} + \Delta \mathbf{W})}{\partial \mathbf{W}} \approx \frac{\partial \mathbf{E}(\mathbf{W})}{\partial \mathbf{W}} + \mathbf{H} \Delta \mathbf{W} = 0. \quad (2.59)$$

З (2.59) випливає, що оптимальний вектор приростів параметрів:

$$\Delta \mathbf{W} = -\mathbf{H}^{-1} \frac{\partial \mathbf{E}}{\partial \mathbf{W}}. \quad (2.60)$$

Практичне застосування даного методу визначення оптимальних кроків для приросту параметрів стикається з проблемами, пов'язаними з інвертуванням матриці \mathbf{H} . Кількість

обчислень, необхідних для виконання даної операції для матриці з розмірністю $(n \times n)$, зростає пропорційно до n^3 .

Існують також обмеження на вигляд матриці \mathbf{H} . Для того, щоб гарантовано існував мінімум цільової функції, необхідно, щоб матриця \mathbf{H} була позитивно означеною, тобто всі її власні значення були додатними. У випадку нульових власних значень маємо сідлову точку, в якій \mathbf{H} не може бути обернена. Наявність від'ємних власних значень свідчить про те, що існує хоча б один максимальний ваговий коефіцієнт, тобто одержане значення цільової функції не відповідає умовам мінімуму.

2.3.4. Метод Левенберга–Маркара

Обмеження на вигляд матриці \mathbf{H} , що характерні для методу Гауса–Ньютона, усувають шляхом регуляризації. Для цього вводять заміну:

$$\mathbf{H} \rightarrow \mathbf{H} + \lambda \mathbf{I}$$

з метою фільтрації власних значень, які менші, ніж λ .

Підставивши в елемент h_{ij} матриці (2.58) значення цільової функції, яка задана середньоквадратичною похибкою, одержимо

$$h_{ij} = \frac{1}{2} \frac{\partial^2 \mathbf{E}}{\partial w_i \partial w_j} = \frac{1}{N} \sum_{n=1}^N \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_i} \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_j} + \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)} \right) \frac{\partial^2 \tilde{\mathbf{y}}^{(n)}}{\partial w_i \partial w_j}. \quad (2.61)$$

При достатньо малих значеннях $(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)})$ другий член формули (2.61) можна не враховувати в подальших обчисленнях. Тому зробимо спрощення:

$$h_{ij} \approx \frac{1}{N} \sum_{n=1}^N \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_i} \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_j},$$

яке дає можливість представити матрицю \mathbf{H} у вигляді:

$$\mathbf{H} \approx \frac{1}{N} \sum_{n=1}^N \mathbf{J}^{(n)} \left(\mathbf{J}^{(n)} \right)^T, \quad (2.62)$$

де

$$\mathbf{J}^{(n)} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{y}}^{(1)}}{\partial w_1} & \frac{\partial \tilde{\mathbf{y}}^{(1)}}{\partial w_2} & \dots & \frac{\partial \tilde{\mathbf{y}}^{(1)}}{\partial w_m} \\ \frac{\partial \tilde{\mathbf{y}}^{(2)}}{\partial w_1} & \frac{\partial \tilde{\mathbf{y}}^{(2)}}{\partial w_2} & \dots & \frac{\partial \tilde{\mathbf{y}}^{(2)}}{\partial w_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_1} & \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_2} & \dots & \frac{\partial \tilde{\mathbf{y}}^{(n)}}{\partial w_m} \end{bmatrix}. \quad (2.63)$$

Операції з якобіаном (2.63) потребують значно менших обчислювальних ресурсів, що пов'язано з використанням частинних похідних першого порядку.

Використовуючи згаданий підхід до регуляризації матриці, яка підлягає інвертуванню, та ввівши спрощення (2.62), одержимо формулу Левенберга–Маркара для формування вектора приросту параметрів:

$$\Delta \mathbf{W} = \left[\frac{1}{N} \sum_{n=1}^N \mathbf{J}^{(n)} \left(\mathbf{J}^{(n)} \right)^T + \lambda \mathbf{I} \right]^{-1} \frac{\partial \mathbf{E}}{\partial \mathbf{W}}. \quad (2.64)$$

Метод Левенберга–Маркара є узагальненням методів градієнтного спуску та Гауса–Ньютона. Метод градієнтного спуску одержуємо з (2.64) у випадку, коли $\eta = \frac{1}{\lambda}$, а λ набуває великих значень:

$$\Delta \mathbf{W} = \frac{1}{\lambda} \frac{\partial \mathbf{E}}{\partial \mathbf{W}} \text{ при } \lambda \rightarrow \infty.$$

За малих значень λ одержуємо метод Гауса–Ньютона:

$$\Delta \mathbf{W} = \left[\frac{1}{N} \sum_{n=1}^N \mathbf{J}^{(n)} \left(\mathbf{J}^{(n)} \right)^T \right]^{-1} \frac{\partial \mathbf{E}}{\partial \mathbf{W}} \text{ при } \lambda \rightarrow 0.$$

2.4. Методи навчання ШНМ типу РБФ

Відомо, що структура РБФ-мереж нагадує структуру ШНМ прямого поширення (рис.2.5) з одним прихованим шаром. Обидва види мереж мають прихований та вихідний шар нейронів з подібною структурою міжнейронних зв'язків. Але є в них і важлива відмінність, що полягає в різних функціональних можливостях нейронів, які входять до складу згаданих мереж.

ШНМ прямого поширення — це мережа з проективним функціональним базисом, який описують залежності (2.33)–(2.36).

Вигляд базису РБФ-мереж впливає вже з їх назви і є ключовим для організації функціонування мереж такого типу. **Радіальні базисні функції** задають виразом, що використовує радіальні відстані

$$t_j = \frac{\left\| \sum_{i=1}^n (w_{ij} - x_i) \right\|}{w_{0j}}$$

в ролі аргументів активаційних функцій для нейронів прихованого шару. Сама активаційна функція прихованого шару найчастіше — гаусіан. Тому вихідний сигнал прихованого шару обчислюють за формулою:

$$h_j = f(t_j) = \exp\left(-\frac{t_j^2}{2}\right) = \exp\left(-\frac{\left\| \sum_{i=1}^n (w_{ij} - x_i) \right\|^2}{2w_{0j}}\right). \quad (2.65)$$

Нейрони вихідного шару реалізують лінійну комбінацію сигналів прихованого шару. З цієї причини аргументи їх активаційних функцій визначають за формулою:

$$\mathbf{s}(x) = v_0 + \sum_{j=1}^m v_j h_j, \quad (2.66)$$

а вихідні сигнали у найпростішому випадку:

$$\tilde{\mathbf{y}} = \varphi[\mathbf{s}(x)] = v_0 + \sum_{j=1}^m v_j h_j. \quad (2.67)$$

2.4.1. Метод точного прямого центрування

Нехай маємо навчальну вибірку $\Psi = \left\{ \left(\mathbf{X}^{(n)}, \mathbf{y}^{(n)} \right) \right\}_{n=1}^N$, що містить N пар: $\mathbf{X}^{(n)}$, $\mathbf{y}^{(n)}$ відповідно вхідних і вихідних векторів, а вхідний алфавіт ШНМ співпадає з даною навчальною вибіркою. Тоді найпростіший метод навчання РБФ-мереж полягає у використанні такої кількості нейронів прихованого шару, яка б дорівнювала кількості елементів навчальної вибірки. При цьому параметри прихованого шару визначають із виразу:

$$\mathbf{w}_j = \mathbf{X}^{(j)}, \quad (2.68)$$

а коефіцієнт w_{0j} задають евристичним шляхом на проміжку:

$$w_{0j} \in \left[\min \left(\left\| \mathbf{X}^{(k)} - \mathbf{X}^{(l)} \right\| \right), \max \left(\left\| \mathbf{X}^{(k)} - \mathbf{X}^{(l)} \right\| \right) \right], \quad (2.69)$$

$k, l = \overline{1, N}, k \neq l.$

Коефіцієнт вихідного шару v_0 визначають з використанням лінійної регресії:

$$v_0 = \frac{1}{N} \sum_{n=1}^N \mathbf{y}^{(n)}, \quad (2.70)$$

а для визначення множини параметрів \mathbf{v}_j формують матрицю вихідних сигналів прихованого шару

$$\mathbf{H} = \begin{pmatrix} h_1^{(1)} & h_2^{(1)} & \dots & h_m^{(1)} \\ h_1^{(2)} & h_2^{(2)} & \dots & h_m^{(2)} \\ \dots & \dots & \dots & \dots \\ h_1^{(N)} & h_2^{(N)} & \dots & h_m^{(N)} \end{pmatrix},$$

кожен рядок якої містить множину вихідних сигналів для відповідної вибірки, а кожен стовпець — допустиму множину вихідних сигналів відповідного нейрона прихованого шару.

Значення параметрів вихідного шару задають через вихідні значення навчальної вибірки

$$\mathbf{v}_j = \left(\mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{y}. \quad (2.71)$$

Оскільки для даної системи рівнянь кількість невідомих співпадає з кількістю рівнянь, то результатом навчання є точний розв'язок, тобто $\tilde{\mathbf{y}}^{(n)} = \mathbf{y}^{(n)}$.

Практичне застосування методу точного центрування стикається з проблемою значних обчислювальних затрат та значної кількості елементів мережі, які зростають зі збільшенням навчальної вибірки.

2.4.2. Метод інтерполяційного прямого центрування

Деякого зменшення об'єму обчислень досягають шляхом введення процесу інтерполяції на навчальній вибірці Ψ . Для цього вводять поняття сусідства, яке задають відстанню:

$$\left\| \mathbf{X} - \mathbf{X}^{(n)} \right\| \leq d, \quad (2.72)$$

де \mathbf{X} — поточне значення вхідного вектора; $\mathbf{X}^{(n)}$ — найближчий вхідний вектор навчальної вибірки, що відповідає даному критерію сусідства.

Вихідний сигнал для поточного вхідного вектора формується з урахуванням тих вихідних сигналів навчальної вибірки, що

відповідають критерію сусідства (2.72). Найпростіший спосіб формування вихідного сигналу:

$$\mathbf{y} = \frac{1}{\delta} \sum_{i=1}^{\delta} \mathbf{y}^{(i)}, \quad (2.73)$$

де δ — кількість елементів підмножини $\left\{ \left(\mathbf{X}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^{\delta}$ навчальної вибірки $\Psi_{\delta} \subset \Psi$, які відповідають критерію (2.72).

Відомий також спосіб інтерполяції з використанням керн-функції $\sigma(z)$, що встановлює правило формування вихідного сигналу навколо кожного ядра $\mathbf{X}^{(i)}$ за формулою:

$$\mathbf{y} = \frac{1}{N} \sum_{i=1}^N \mathbf{y}^{(i)} \sigma \left(\left\| \mathbf{X} - \mathbf{X}^{(i)} \right\| \right). \quad (2.74)$$

Для ефективної інтерполяції керн-функція $\sigma(z)$ повинна бути спадною та нормалізованою таким чином, що $\int \sigma(z) dz = 1$.

2.4.3. Метод центрування на підмножині

Згадані методи прямого центрування з використанням всієї навчальної вибірки є доволі простими і дають точні результати. Єдиний недолік цих методів — це великий розмір прихованого шару, що дозволяє застосовувати їх лише для незначних навчальних вибірок. За необхідності збільшення вхідного алфавіту РБФ-мережі застосовують методи навчання, які у ролі навчальної вибірки використовують тільки підмножину цього алфавіту. Порядок формування згаданої підмножини започатковує різні методи навчання, найпростішим з яких є метод випадкового формування навчальної вибірки.

Із заданого алфавіту пар $\Phi = \left\{ \left(\mathbf{X}^{(n)}, \mathbf{y}^{(n)} \right) \right\}_{n=1}^N$ шляхом випадкового вибору формують навчальну вибірку $\Psi = \left\{ \left(\mathbf{X}^{(k)}, \mathbf{y}^{(k)} \right) \right\}_{k=1}^K$, що задовольняє умові $\Psi \subset \Phi$. Вибірку Ψ

формують з метою зменшення кількості нейронів прихованого шару, але такий підхід утруднює досягнення заданої точності навчання, що є його істотним недоліком.

Більшою ефективністю характеризується ідея використання принципу найменших квадратів для знаходження тих елементів заданого алфавіту, включення яких у навчальну вибірку може забезпечити максимальну точність відтворення мережею всього алфавіту. Для цього необхідно знайти умови мінімізації сумарної похибки

$$\mathbf{E} = \sum_{i=1}^N \left(\mathbf{y}^{(i)} - \tilde{\mathbf{y}}^{(i)} \right)^2 = (\mathbf{y} - \tilde{\mathbf{y}})^T (\mathbf{y} - \tilde{\mathbf{y}}). \quad (2.75)$$

Використовуючи вектор $\mathbf{X}^{(k)}$ як центр радіальної базисної функції, визначимо поточне значення вихідного вектора:

$$\tilde{\mathbf{y}}^{(i)} = B_1 f \left(\frac{\mathbf{X}^{(i)} - \mathbf{X}^{(k)}}{w_0} \right) = B_1 h_{ik}. \quad (2.76)$$

Підставивши (2.76) в (2.75), отримаємо:

$$\mathbf{E} = B_1^2 \left(\mathbf{H}^T \mathbf{H} \right)_{kk} - 2B_1 \left(\mathbf{H}^T \mathbf{y} \right)_k + \mathbf{y}^T \mathbf{y}, \quad (2.77)$$

де B_1 — коефіцієнт, який потрібно визначити;

\mathbf{H} — інтерполяційна матриця :

$$\mathbf{H} = \begin{pmatrix} h_1^{(1)} & h_2^{(1)} & \dots & h_m^{(1)} \\ h_1^{(2)} & h_2^{(2)} & \dots & h_m^{(2)} \\ \dots & \dots & \dots & \dots \\ h_1^{(N)} & h_2^{(N)} & \dots & h_m^{(N)} \end{pmatrix}.$$

Визначимо B_1 з умови мінімального значення похибки (2.77).

$$\frac{\partial \mathbf{E}}{\partial B_1} = 2B_1 \left(\mathbf{H}^T \mathbf{H} \right)_{kk} - 2 \left(\mathbf{H}^T \mathbf{y} \right)_k = 0. \quad (2.78)$$

Звідси

$$B_1 = \frac{(\mathbf{H}^T \mathbf{y})_k}{(\mathbf{H}^T \mathbf{H})_{kk}}. \quad (2.79)$$

Отже, мінімальне значення похибки для виразу (2.77) одержимо для випадку

$$\mathbf{E} = \mathbf{y}^T \mathbf{y} - \frac{(\mathbf{H}^T \mathbf{y})_k^2}{(\mathbf{H}^T \mathbf{H})_{kk}}. \quad (2.80)$$

Оскільки член $\mathbf{y}^T \mathbf{y}$ не залежить від вибору центру базисної радіальної функції, то його можна не враховувати в подальших розрахунках. Таким чином, першим центром μ_1 вибираємо такий елемент алфавіту $\mathbf{X}^{(k)}$, який задовольняє умову мінімізації похибки (2.80).

На наступному етапі необхідно відшукати другий центр РБФ $\mathbf{X}^{(p)}$ з мінімальним значенням цільової функції за умови, що нам уже відомий центр $\mathbf{X}^{(k)}$. Для цього знаходять лінійну комбінацію

$$\mathbf{Q}^{(n)} = A_1 h_{nk} + A_2 h_{np}, \quad (2.81)$$

для якої виконується умова ортогональності:

$$\sum_{n=1}^N \mathbf{Q}^{(n)} \mathbf{H}_{nk} = A_1 (\mathbf{H}^T \mathbf{H})_{kk} + A_2 (\mathbf{H}^T \mathbf{H})_{kp} = 0, \quad (2.82)$$

яка справедлива при $h_{nk} \neq h_{np}$.

Нормалізувавши коефіцієнти A_1 та A_2 за умови:

$$\begin{aligned} \sum_{n=1}^N \mathbf{Q}^{(n)} &= \\ &= A_1^2 (\mathbf{H}^T \mathbf{H})_{kk} + 2A_1 A_2 (\mathbf{H}^T \mathbf{H})_{kp} + A_2^2 (\mathbf{H}^T \mathbf{H})_{pp} = 1, \end{aligned} \quad (2.83)$$

представимо поточний вихідний вектор у вигляді:

$$\tilde{\mathbf{y}} = B_1 f \left(\frac{\|\mathbf{X} - \mathbf{X}^{(k)}\|}{w_0} \right) + B_2 \mathbf{Q}. \quad (2.84)$$

Значення сумарної похибки для даного вихідного вектора:

$$\begin{aligned} \mathbf{E} &= \mathbf{E}_1 + \mathbf{E}_2 = \\ &= B_2^2 \sum_{n=1}^N (\mathbf{Q}^{(n)})^2 - 2B_2 \sum_{n=1}^N \mathbf{Q}^{(n)} y^{(n)} + B_1 B_2 \sum_{n=1}^N \mathbf{Q}^{(n)} \mathbf{H}_{nk} + \\ &\quad + B_1^2 (\mathbf{H}^T \mathbf{H})_{kk} - 2B_1 (\mathbf{H}^T \mathbf{y})_k + (\mathbf{y}^T \mathbf{y}), \\ \mathbf{E}_1 &= B_2^2 \sum_{n=1}^N (\mathbf{Q}^{(n)})^2 - 2B_2 \sum_{n=1}^N \mathbf{Q}^{(n)} y^{(n)}, \\ \mathbf{E}_2 &= B_1 B_2 \sum_{n=1}^N \mathbf{Q}^{(n)} \mathbf{H}_{nk} + B_1^2 (\mathbf{H}^T \mathbf{H})_{kk} - 2B_1 (\mathbf{H}^T \mathbf{y})_k + (\mathbf{y}^T \mathbf{y}). \end{aligned} \quad (2.85)$$

Очевидно, що вираз для \mathbf{E}_2 не впливає на величину похибки, оскільки три останні члени даного виразу були мінімізовані на попередньому етапі, а змішаний член дорівнює нулю, оскільки має місце умова ортогональності.

Отже, для мінімізації сумарної похибки необхідно знайти $\min(\mathbf{E}_1)$. Для цього прирівняємо до нуля її частинну похідну, використавши умову ортогональності (2.82).

$$\begin{aligned} \mathbf{E}_1 &= B_2^2 \sum_{n=1}^N (\mathbf{Q}^{(n)})^2 - 2B_2 \sum_{n=1}^N \mathbf{Q}^{(n)} y^{(n)} = B_2^2 \sum_{n=1}^N (\mathbf{Q}^{(n)})^2 - \\ &\quad - 2B_2 [A_1 (\mathbf{H}^T \mathbf{H})_{kk} + A_2 (\mathbf{H}^T \mathbf{H})_{kp}], \end{aligned} \quad (2.86)$$

$$\frac{\partial \mathbf{E}_1}{\partial B_2} = B_2 - [A_1 (\mathbf{H}^T \mathbf{H})_{kk} + A_2 (\mathbf{H}^T \mathbf{H})_{kp}] = 0. \quad (2.87)$$

У відповідності до (2.87) значення коефіцієнта B_2

$$B_2 = \left[A_1 (\mathbf{H}^T \mathbf{H})_{kk} + A_2 (\mathbf{H}^T \mathbf{H})_{kp} \right]. \quad (2.88)$$

Підставивши (2.88) у (2.86), визначимо величину сумарної похибки вихідного сигналу для центру $\mathbf{X}(p)$:

$$E_1 = - \left[A_1 (\mathbf{H}^T \mathbf{H})_{kk} + A_2 (\mathbf{H}^T \mathbf{H})_{kp} \right]^2. \quad (2.89)$$

Виходячи з умов ортогональності та нормалізації, отримаємо:

$$A_1 = -A_2 \frac{(\mathbf{H}^T \mathbf{H})_{pp}}{(\mathbf{H}^T \mathbf{H})_{kk}}, \quad (2.90)$$

$$A_2 = \sqrt{\frac{(\mathbf{H}^T \mathbf{H})_{kk}}{(\mathbf{H}^T \mathbf{H})_{pp} \left[(\mathbf{H}^T \mathbf{H})_{kk} - 2(\mathbf{H}^T \mathbf{H})_{kp} + (\mathbf{H}^T \mathbf{H})_{pp} \right]}}. \quad (2.91)$$

Пошук наступних центрів повторюється аналогічно і може бути припинений за умови досягнення заданої точності навчання.

2.4.4. Метод K середніх

Метод K середніх найчастіше використовують у задачах класифікації. Він дозволяє провести узагальнення деякої множини елементів за заданими критеріями шляхом об'єднання в кластери елементів множини з близькими характеристиками. Процес, який називають кластеризацією, дозволяє значно спростити подальшу обробку інформації шляхом заміни обробки всього кластера обробкою елемента, який показники, близькі до усереднених.

Для навчання РБФ-мережі за даним методом вибирають K центрів μ_k та ітеративно змінюють їх за критерієм мінімізації цільової функції:

$$E = \sum_{n=1}^N \sum_{k=1}^K \delta_k(\mathbf{X}^{(n)}) \left\| \mathbf{X}^{(n)} - \mu_k \right\|^2, \quad (2.92)$$

де

$$\delta_k(\mathbf{X}^{(n)}) = \begin{cases} 1 & \text{при } \left\| \mathbf{X}^{(n)} - \mu_k \right\| < \left\| \mathbf{X}^{(n)} - \mu_j \right\|, j, k \leq K, j \neq k, \\ 0 & \text{в інших випадках.} \end{cases}$$

Функція $\delta_k(\mathbf{X})$ призначена для відбору тих векторів вхідної послідовності даних, які розміщуються ближче до центру μ_k відповідної радіальної базисної функції.

Використовуючи ітераційний процес градієнтного спуску, проводять оновлення центрів μ_k :

$$\begin{aligned} \mu_k(t+1) &= \mu_k(t) - \eta \frac{\partial E}{\partial \mu_k} = \\ &= \mu_k(t) + 2\eta \sum_{n=1}^N \delta_k(\mathbf{X}^{(n)}) \left[\mathbf{X}^{(n)} - \mu_k(t) \right] = \\ &= (1 - 2\eta N_k) \mu_k(t) + 2\eta \sum_{n=1}^{N_k} \mathbf{X}^{(n)}, \end{aligned} \quad (2.93)$$

де N_k — кількість вхідних векторів, які входять до кластера K .

Ітераційний процес продовжують до досягнення максимально допустимої зміни цільової функції. Після цього центри μ_k вважаються такими, що вибрані оптимально і можуть бути використані для настройки базисних радіальних функцій прихованого шару.

2.4.5. Імовірнісні методи

Структура імовірнісної мережі повністю тотожна узагальненій структурі ШНМ прямого поширення, що складається з n входів, m нейронів прихованого шару та z нейронів вихідного шару. Основною відмінністю є форма

базисної функції, що задає густину імовірності, розподілену за законом Гауса:

$$p(\mathbf{X}|j) = \frac{1}{\sqrt{(2\pi\sigma_j^2)^n}} \exp\left(-\frac{\|\mathbf{X}-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right), \quad (2.94)$$

де $j = 1, 2, \dots, m$ — номер базисної функції відповідного нейрона прихованого шару; $\boldsymbol{\mu}_j$ — вектор, що представляє центр базисної функції з номером j ; σ_j^2 — відхилення від центру, яке задає міру поширення вхідних даних навколо центру кластеризації.

Така форма базисної функції впливає з припущення про незалежність компонентів вхідного вектора \mathbf{X} та можливість задавання однакового відхилення для всіх центрів кластеризації.

Довільний елемент k вихідного шару імовірнісної мережі реалізує активаційну функцію густини імовірності:

$$p(\mathbf{X}|k) = \sum_{j=1}^m v_{jk} p(\mathbf{X}|j), \quad (2.95)$$

де параметр вихідного шару v_{jk} задовольняє умови:

$$\sum_{j=1}^m v_{jk} = 1, \quad v_{jk} \geq 0$$

і визначає апіорну імовірність того, що генерований нейроном j сигнал належить до класу k .

За теоремою Бейеса, апостеріорну імовірність $P(j|k, \mathbf{X})$ того, що нейрон j під впливом вхідного вектора \mathbf{X} сформував сигнал, який належить до класу k , можна виразити залежністю:

$$P(j|k, \mathbf{X}) = \frac{v_{jk} p(\mathbf{X}|j)}{\sum_l v_{lk} p(\mathbf{X}|l)}. \quad (2.96)$$

2.4.5.1. Метод максимізації математичного сподівання

Одним із відомих методів навчання імовірнісних ШНМ прямого поширення є метод максимізації математичного сподівання, запропонований у [66].

Для навчання за цим методом використовують векторний параметр θ , що узагальнює параметри $\boldsymbol{\mu}_j$ та σ_j^2 . Процес навчання полягає у визначенні апіорних імовірностей $P(k)$, $k = 1, 2, \dots, z$, того, що вихідний сигнал належить до класу k та згаданого параметра θ . Таким чином, узагальнений векторний параметр має вигляд: $\boldsymbol{\theta} = (\theta, P(1), P(2), \dots, P(z))$.

Задамо логарифмічну імовірнісну функцію від навчальної вибірки \mathbf{X} за умови незалежності кожного з елементів даних

$$L(\boldsymbol{\theta}) = p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{i=1}^N \log p(\mathbf{X}^{(i)}, k^{(i)}). \quad (2.97)$$

Враховуючи залежність $P(\mathbf{X}, k) = P(k)p(\mathbf{X}|k)$ і той факт, що множина даних \mathbf{X} складається з z незалежних кластерів \mathbf{X}_k , функцію (2.97) перепишемо у вигляді:

$$L(\boldsymbol{\theta}) = \sum_{k=1}^z N_k \log P(k) + \sum_{k=1}^z \sum_{i=1}^{N_k} p(\mathbf{X}^{(i)}|k). \quad (2.98)$$

Максимальне значення першого доданка досягається за умови:

$$P(k) = \frac{N_k}{N}, \quad k = 1, 2, \dots, z, \quad (2.99)$$

а максимізацію другого доданка забезпечують у процесі навчання.

Для навчання імовірнісної ШНМ застосовують функцію:

$$L(\theta) = \sum_{k=1}^z \sum_{i=1}^{N_k} \log p(\mathbf{X}^{(i)}|k). \quad (2.100)$$

Щоб застосовувати функцію (2.100), необхідно знати всю допустиму послідовність вхідних даних. Тому для забезпечення процесу навчання на навчальній вибірці \mathbf{X} , що є підмножиною допустимої послідовності, вводять проектуючий вектор [157]

$$\mathbf{Z} = (\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \dots, \mathbf{Z}^{(i)}, \dots, \mathbf{Z}^{(N)}), \quad (2.101)$$

де

$$\mathbf{Z}^{(i)} = \begin{cases} 1, & \text{якщо } \mathbf{X}^{(i)} \in \mu_j, \\ 0 & \text{в інших випадках.} \end{cases}$$

Доповнена таким чином множина даних тепер має вигляд:

$$\mathbf{D} = \{\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \dots, \mathbf{d}^{(i)}, \dots, \mathbf{d}^{(N)}\}, \text{ де } \mathbf{d}^{(i)} = (\mathbf{X}^{(i)}, k^{(i)}, \mathbf{Z}^{(i)}),$$

а модернізована логарифмічна імовірнісна функція

$$L_z = \sum_{i=1}^N \sum_{j=1}^M \mathbf{Z}_j^{(i)} \log(v_{jk^{(i)}} p(\mathbf{X}^{(i)}|j)). \quad (2.102)$$

Ітераційний процес методу максимізації математичного сподівання (EM Algorithm) складається з двох основних кроків, що мають назву Е-кроку та М-кроку.

Під час Е-кроку $(t+1)$ поточне математичне сподівання значення $Z_j^{(i)}$ при вхідному векторі $\mathbf{X}^{(i)}$ визначається як апостеріорна імовірність $P_t(j|k^{(i)}, \mathbf{X}^{(i)})$, що обчислена з урахуванням поточного узагальненого параметра θ_t .

М-крок полягає в обчисленні нових значень $(v_{jk})_{t+1}$, $(\mu_j)_{t+1}$ та $(\sigma_j^2)_{t+1}$, виходячи з умови максимізації виразу:

$$\begin{aligned} Q(\theta; \theta_t) &= \\ &= \sum_{k=1}^z \sum_{n=1}^{N_k} \sum_{j=1}^M P_t(j|k, \mathbf{X}^{(n)}) \left[\log v_{jk} + \log p(\mathbf{X}^{(n)}|j) \right]. \end{aligned} \quad (2.103)$$

Для цього представимо (2.103) у вигляді суми:

$$Q(\theta; \theta_t) = Q_1(\theta; \theta_t) + Q_2(\theta; \theta_t), \quad (2.104)$$

де

$$Q_1(\theta; \theta_t) = \sum_{k=1}^z \sum_{n=1}^{N_k} \sum_{j=1}^M P_t(j|k, \mathbf{X}^{(n)}) \log v_{jk}, \quad (2.105)$$

$$Q_2(\theta; \theta_t) = \sum_{k=1}^z \sum_{n=1}^{N_k} \sum_{j=1}^M P_t(j|k, \mathbf{X}^{(n)}) \log p(\mathbf{X}^{(n)}|j). \quad (2.106)$$

Єдиним параметром, що входить до виразу (2.105), є v_{jk} . Виконавши стандартну техніку пошуку екстремуму та врахувавши $\sum_{j=1}^m v_{jk} = 1$, $v_{jk} \geq 0$, одержимо:

$$v_{jk} = \frac{1}{N_k} \sum_{n=1}^{N_k} P_t(j|k, \mathbf{X}^{(n)}), \quad j = 1, 2, \dots, m, \quad k = 1, 2, \dots, z. \quad (2.107)$$

Знайдемо екстремуми функції (2.106). Для цього прирівняємо до нуля відповідні частинні похідні $\frac{\partial Q_2}{\partial \mu_j}$ та $\frac{\partial Q_2}{\partial \sigma_j^2}$. З одержаних рівнянь маємо:

$$(\mu_j)_{t+1} = \frac{\sum_{k=1}^z \sum_{n=1}^{N_k} P_t(j|k, \mathbf{X}^{(n)}) \mathbf{X}^{(n)}}{\sum_{k=1}^z \sum_{n=1}^{N_k} P_t(j|k, \mathbf{X}^{(n)})}, \quad (2.108)$$

$$\left(\sigma_j^2\right)_{t+1} = \frac{1}{r} \frac{\sum_{k=1}^z \sum_{n=1}^{N_k} P_t(j|k, \mathbf{X}^{(n)}) \left\| \mathbf{X}^{(n)} - (\boldsymbol{\mu}_j)_{t+1} \right\|^2}{\sum_{k=1}^z \sum_{n=1}^{N_k} P_t(j|k, \mathbf{X}^{(n)})}, \quad (2.109)$$

де $\mathbf{X} = (x_1, x_2, \dots, x_r)$.

Блок-схема алгоритму, що побудований за методом максимізації математичного сподівання, представлена на рис.2.10.

Даний алгоритм для методу базується на ітеративній процедурі оцінки параметра максимальної імовірності з прихованими змінними. Він може бути застосований до систем розпізнавання в просторово-часових моделях, коли просторові змінні приховані від спостерігача, а просторові параметри та параметри моделі повинні визначатися одночасно.

2.4.5.2. Метод максимізації математичного сподівання для нелінійних динамічних систем

Метод максимізації математичного сподівання може бути узагальненим для моделювання нелінійних динамічних систем із внутрішнім станом \mathbf{U}_t , вхідним вектором \mathbf{X}_t та вихідним вектором \mathbf{Y}_t . Внутрішній стан системи еволюціонує відповідно до стаціонарної *нелінійної* динаміки під впливом вхідного вектора, вихідного вектора та адитивного шуму:

$$\mathbf{X}_{t+1} = f(\mathbf{X}_t, \mathbf{U}_t) + \phi, \quad (2.110)$$

де ϕ — адитивний шум з розподілом Гауса, нульовим математичним сподіванням та відхиленням ω .

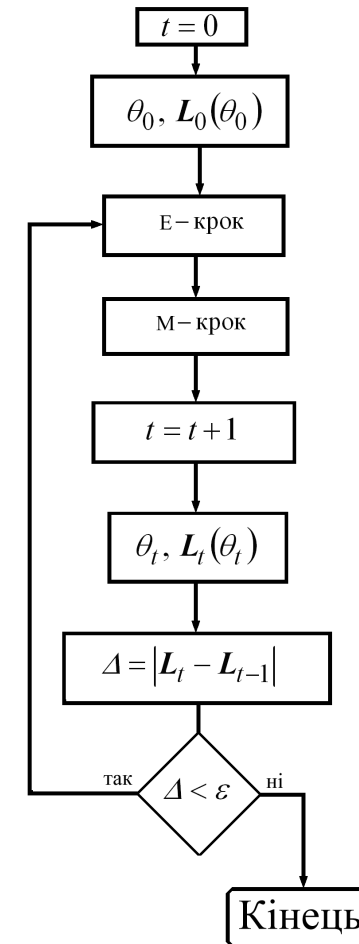


Рис.2.10. Алгоритм для методу максимізації математичного сподівання.

Вихід системи є нелінійним відносно входу та внутрішнього стану:

$$Y_t = g(X_t, U_t) + \varphi, \quad (2.111)$$

де φ — шум з розподілом Гауса, нульовим математичним сподіванням та відхиленням σ .

Е-крок методу максимізації математичного сподівання полягає у визначенні внутрішнього стану через апостеріорну густину імовірності $p(X_t | (U_1, U_2, \dots, U_T, Y_1, Y_2, \dots, Y_T))$, де $1 \leq t \leq T$.

Якщо функція $f(X_t, U_t)$ лінійна, і закон розподілу згаданої густини імовірності відповідає закону розподілу Гауса, то для визначення математичного сподівання та відхилення доцільно застосувати *калманівське згладжування* [112]. У випадку нелінійності функції $f(X_t, U_t)$ густина імовірності у загальному випадку не відповідає закону розподілу Гауса. Тому для визначення математичного сподівання та відхилення використовують різні підходи, серед яких і розширене калманівське згладжування, що забезпечує згладжування локальної лінеаризації нелінійної системи шляхом введення додаткових членів у (2.110) та (2.111)

$$X_{t+1} = f(\tilde{X}_t, U_t) + A(X_t - \tilde{X}_t) + \phi, \quad (2.112)$$

$$Y_t = g(\tilde{X}_t, U_t) + B(X_t - \tilde{X}_t) + \varphi. \quad (2.113)$$

де $A \equiv \frac{\partial f}{\partial X} \Big|_{X=\tilde{X}}$ та $B \equiv \frac{\partial g}{\partial X} \Big|_{X=\tilde{X}}$ — матриці, які визначають за частинними похідними від функцій $f(X, U)$ і $g(X, U)$ для кожної точки \tilde{X} в просторі вектора X .

Під час **М-кроку** виконують модифікацію залежностей f, g з шумовими відхиленнями ϕ, φ на основі вхідного вектора, вихідного вектора та сформованого внутрішнього стану. З цією

метою задають вихідні сигнали прихованого шару РБФ-мережі за допомогою виразу:

$$Z = \sum_{i=1}^m w_i \rho_i(X) + AX + BU + b + \phi, \quad (2.114)$$

де b — вихідний діагональний вектор; w_i — вагові коефіцієнти i -го нейрона прихованого шару з радіальною базисною функцією

$$\rho_i(X) = \frac{1}{\sqrt{2\pi S_i}} \exp\left(-\frac{(X - \mu_i)^T (X - \mu_i)}{2S_i}\right), \quad (2.115)$$

яка містить центр μ_i та матрицю відхилень S_i .

Вираз, що підлягає мінімізації для однієї точки [166]:

$$Q(\theta) = -\frac{1}{2} \frac{[Z - \tilde{Z}_\theta(X, U)]^T [Z - \tilde{Z}_\theta(X, U)]}{\phi} - \frac{1}{2} \ln|\phi|, \quad (2.116)$$

де $\tilde{Z}_\theta(X, U) = \sum_{i=1}^m w_i \rho_i(X) + AX + BU + b + \phi$,

$$\theta = \{w_1, w_2, \dots, w_m, A, B, b\}.$$

Із виразу $\frac{\partial Q}{\partial \theta} = 0$ знаходять оптимальні значення для узагальненого параметра θ та адитивного шуму ϕ .

Розділ 3

ОПТИМІЗАЦІЯ АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ

Оптимізація роботи нейронної мережі полягає не тільки у настройці її синаптичних коефіцієнтів для досягнення бажаної реакції на вхідні дані, а й в оптимізації всієї архітектури ШНМ. Такий підхід включає оптимізацію топології штучної нейронної мережі та узагальнення з метою забезпечення ефективного функціонування в значно більшому просторі, ніж той, що використовувався в процесі її навчання.

Тривіальним підходом до розширення сфери ефективного функціонування ШНМ є пряме застосування знань про всю множину вхідних даних на етапі формування архітектури та навчання. Але такий підхід, як правило, є недосяжним в основному з двох причин, якими є:

- відсутність знань про всю можливу множину вхідних даних;
- великий об'єм роботи, яку необхідно виконати у випадку великих масивів вхідних даних.

Тому жоден із сучасних методів оптимізації архітектури не може гарантувати досягнення найкращого результату для кожного конкретного випадку. Методи, що виникли в процесі розв'язування практичних задач на ШНМ, базуються на досягненні компромісу між часом навчання мережі, кількістю її елементів та ефективністю узагальнення.

Критерієм оптимізації, як і під час навчання, є мінімізація цільової функції, в ролі якої найчастіше використовують середньоквадратичну похибку, що поділяється на навчальну та реальну.

3.1. Методи визначення реальної похибки

Реальною похибкою називають похибку, з якою функціонує ШНМ в умовах реальних даних, на відміну від *навчальної*

похибки, що визначається при роботі з навчальною вибіркою. Як правило, реальна похибка завжди перевищує навчальну. Тому для визначення ступеня коректності роботи ШНМ дуже важливо мати оцінку реальної похибки. Очевидний підхід, що широко застосовується для визначення реальної похибки, полягає у використанні *оцінки на незалежній підмножині*. Незалежна підмножина повинна включати діапазон зміни даних, який планується використовувати в умовах реального функціонування ШНМ, бути сформованою до проведення навчання ШНМ і не повинна використовуватися в майбутньому. Такий підхід характерний для ШНМ, які працюють з неперервними потоками звукової, відео- або телекомунікаційної інформації. В цьому випадку не виникає труднощів при формуванні незалежної підмножини даних.

Коли вхідний масив даних обмежений, використовують *метод багаторазової оцінки* [168]. Цей метод полягає в реалізації об'єднаної процедури навчання ШНМ та визначення її реальної похибки.

Представимо множину вхідних даних D у вигляді підмножин, які задовольняють умови:

$$D = D_1 \cup D_2 \cup \dots \cup D_q, \quad D_1 \cap D_2 \cap \dots \cap D_q = 0. \quad (3.1)$$

Виконаємо процедуру q раз, кожного разу вважаючи одну з вибраних підмножин за тестову, а інші $q-1$ за навчальні. Після кожного проходу визначимо тестову похибку $E_t^{(i)}$, де i — поточний номер проходу. Тоді реальна похибка визначатиметься за формулою:

$$E_r = \frac{1}{q} \sum_{i=1}^q E_t^{(i)}. \quad (3.2)$$

Алгоритм методу багаторазової оцінки показаний на рис.3.1. Суть його полягає у визначенні реальної похибки у вигляді стандартного відхилення тестових похибок, обчислених на вибраних підмножинах.

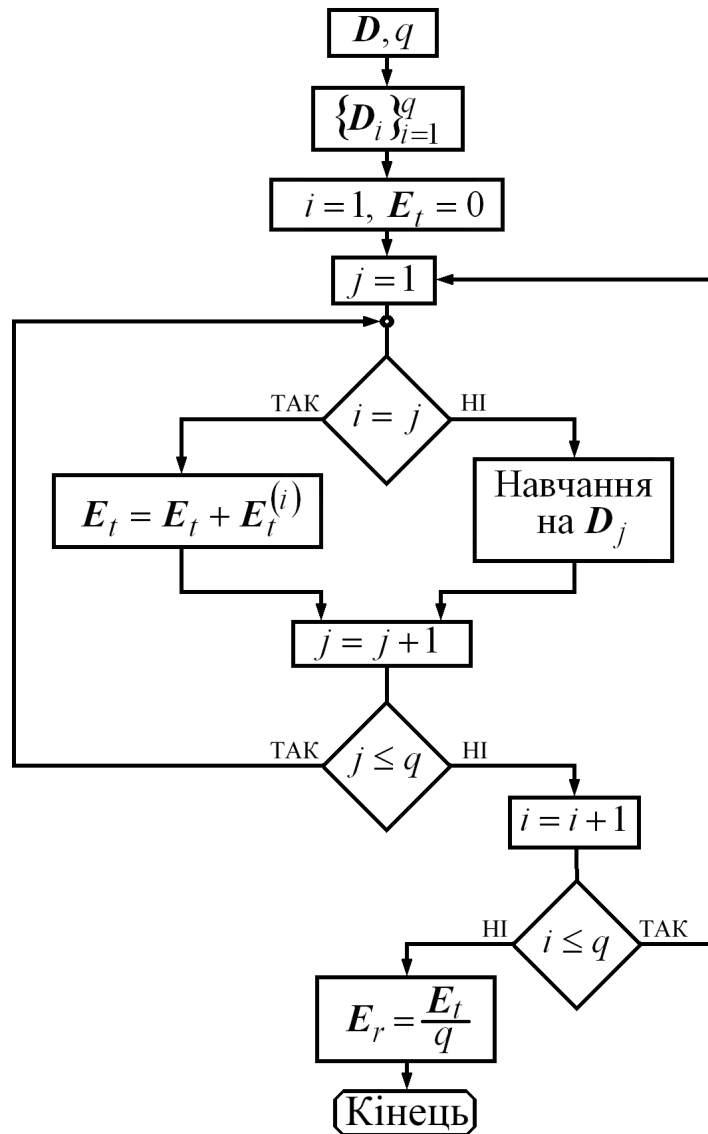


Рис.3.1. Алгоритм багаторазової оцінки.

Мінімізація реальної похибки подібна до процесу навчання тим, що може використовувати методи, які пов'язані з модифікацією вагових коефіцієнтів, але, на відміну від нього, включає методи, що модифікують архітектуру ШНМ.

3.2. Дилема відхилення / дисперсії

У загальному випадку мінімізація реальної похибки визначається зі співвідношення *відхилення та дисперсії* [84].

Статистичне відхилення задає обмеження, що впливають із архітектури ШНМ та забезпечують точність реалізації цільової функції. Його визначення відбувається на множині навчальної вибірки і не залежить від функціонування нейронної мережі в умовах реальних вхідних даних.

Статистична дисперсія задає діапазон флуктуації ефективності процесу навчання ШНМ на різних вхідних вибірках за умови використання однієї і тієї ж моделі визначення цільової функції.

Як уже згадувалось, надто точна підгонка параметрів ШНМ на навчальній вибірці зовсім не означає, що в результаті буде досягнутим мінімальне значення реальної похибки. Для визначення всіх факторів, що впливають на величину реальної

похибки, розглянемо декомпозицію $E_{\Psi} \left[(f(X) - \tilde{f}(X))^2 \right]$ на

навчальній вибірці $\Psi = \left\{ (X^{(i)}, y^{(i)}) \right\}_{i=1}^N$:

$$\begin{aligned}
 E_{\Psi} \left[(f(X) - E[y|X])^2 \right] &= \\
 &= (E_{\Psi} [f(X)] - E[y|X])^2 + E_{\Psi} \left[(f(X) - E_{\Psi} [f(X)])^2 \right] = \\
 &= B^2(f(X)) + V(f(X)), \tag{3.3}
 \end{aligned}$$

де $\tilde{f}(X) = E[y|X]$ — наближене значення активаційної функції, що було досягнуте на навчальній вибірці Ψ ;

$B(f(X)) = E_{\psi}[f(X)] - E[y|X]$ — статистичне відхилення;

$V(f(X)) = E_{\psi}[(f(X) - E_{\psi}[f(X)])^2]$ — статистична дисперсія.

Оптимальне функціонування ШНМ досягається як результат одночасної мінімізації і статистичного відхилення і дисперсії. Але реальна ситуація полягає в тому, що точна підгонка параметрів на навчальній вибірці приводить до зменшення відхилення і одночасного зростання дисперсії. Навпаки, спроби зменшення статистичної дисперсії приводять до збільшення відхилення. Тому таке співвідношення між даними величинами має назву дилеми відхилення / дисперсії.

Нарощування архітектури ШНМ дає можливість зменшити рівень статистичного відхилення, а використання методів відсікання мінімізує дисперсію. Можливі стратегії одночасної мінімізації відхилення та дисперсії базуються на збільшенні об'єму вхідних даних і навчальної вибірки.

3.3. Методи оптимізації параметрів

З метою оптимізації ШНМ прямого поширення використовують ряд методів. Найбільш відомі з них: попередня зупинка, регуляризація, усереднення, відсікання та нарощування.

3.3.1. Метод попередньої зупинки

Ретельна підгонка параметрів ШНМ на фіксованій навчальній вибірці може призвести до надто точної настройки на особливості конкретних даних, що неминуче викличе збільшення реальної похибки. Очевидним виходом із цієї ситуації є зупинка процесу навчання до того моменту, доки реальна похибка не почне зростати з причини надлишкового регулювання. Такий підхід, що отримав назву методу попередньої зупинки [155,174], часто застосовується на практиці

і має різні модифікації, пов'язані з використанням різноманітних методик визначення моменту зупинки процесу навчання.

У найпростішому випадку регулярно через певні інтервали проводять обчислення навчальної та реальної похибки. Зупинку виконують за наявності тенденції зростання реальної похибки. Але таке визначення моменту зупинки має небезпеку попадання у локальний мінімум, якщо у вхідному сигналі присутній шум або значна нелінійність. Для того, щоб уникнути попадання в локальний мінімум, оцінку реальної похибки виконують на підмножині параметрів.

Існують також більш складні методики, що базуються на встановленні порога чутливості, перевищення якого реальною похибкою є умовою попередньої зупинки. Конкретну величину такого порога визначають у результаті практичної роботи з ШНМ, оскільки він залежить як від характеру вхідних даних, так і від архітектури нейронної мережі.

3.3.2. Регуляризація

Методи, які називають регуляризацією, були започатковані в [16,17] у межах нових підходів до вирішення некоректно поставлених задач. Реальну похибку представляють у вигляді суми середньоквадратичної похибки та деякої функції $R(W)$, що задає попередній стан ШНМ:

$$E = \frac{1}{N} \sum_{n=1}^N (y^{(n)} - \tilde{y}^{(n)})^2 + kR(W), \quad (3.4)$$

де k — коефіцієнт регуляризації, який задає ступінь впливу $R(W)$ на реальну похибку.

Мінімізація (3.4) прямо відповідає вирішенню дилеми відхилення / дисперсії, оскільки середньоквадратична похибка задає статистичне відхилення, а функція $R(W)$ — величину дисперсії. Коефіцієнт регуляризації k відіграє роль параметра, що формує співвідношення між статистичним

відхиленням та дисперсією при вирішенні дилеми відхилення / дисперсії і змінюється в діапазоні $0 < k < \frac{2\sigma^2}{W^T W}$.

Його зростання призводить до збільшення відхилення, оскільки частка середньоквадратичної похибки у виразі (3.4) зменшується. Відповідно, зменшення коефіцієнта k збільшує вплив середньоквадратичної складової, що призводить до зростання дисперсії.

Оскільки реальна похибка зменшується при застосуванні методик пригнічення вагових коефіцієнтів, найчастіше її задають у такому вигляді:

$$E = \frac{1}{N} \sum_{n=1}^N \left(y^{(n)} - \tilde{y}^{(n)} \right)^2 + k \sum_{i=1}^M w_i^2. \quad (3.5)$$

Використання функції пригнічення зумовлено тим, що більшість методів навчання пов'язані зі зменшенням значень вагових коефіцієнтів. Так, наприклад, працює метод градієнтного спуску, що забезпечує модифікацію параметрів ШНМ за формулою:

$$w_i(t+1) = (1 - k\eta)w_i(t) - \eta \frac{\partial E}{\partial w_k}. \quad (3.6)$$

У [86] запропонований альтернативний підхід, що задає в ролі дисперсійної характеристики функціонал, який формують, виходячи з поняття гладкості.

Нехай дана навчальна вибірка $\Psi = \left\{ \left(X^{(i)}, y^{(i)} \right) \in R^d \times R \right\}_{i=1}^N$, що містить множину даних, яка характеризується функціональною залежністю $y = f(X)$ при $X \in R^d$, сформованою в умовах шуму. Завдання оптимізації ШНМ полягає в настройці параметрів таким чином, щоб досягти відтворення даної функції f за умови мінімально можливої реальної похибки. Така задача очевидно може бути віднесеною до класу некоректно поставлених, оскільки має необмежену кількість розв'язків. Для того, щоб одержати можливість вибору

одного з них, необхідно володіти деякими апріорними знаннями про функцію, яку потрібно моделювати на ШНМ. В межах даного підходу найбільш загальною умовою, якій повинна задовольняти функція $y = f(X)$, є гладкість.

Гладкою будемо називати таку функцію, що має неперервну другу похідну на всій множині визначення. Іншими словами, двом як завгодно близьким значенням аргументу повинні відповідати два як завгодно близькі значення функції.

Отже, головна ідея регуляризації полягає в представленні проблеми оптимізації ШНМ у вигляді некоректно поставленої задачі. Розв'язок даної задачі одержують, виходячи з варіаційного принципу, при реалізації якого використовують вхідний набір даних та попередню інформацію про гладкість функції. Гладкість визначається за допомогою функціонала гладкості $R[y]$ таким чином, що менші його значення відповідають більшій гладкості функції.

Оскільки критеріями мінімізації реальної похибки є одночасно подібність результатів до тих, що задані у навчальній вибірці, а разом з тим і гладкість функції, вираз (3.4) для реальної похибки набуде вигляду:

$$E[y] = \frac{1}{N} \sum_{n=1}^N \left(y^{(n)} - \tilde{y}^{(n)} \right)^2 + kR[y], \quad (3.7)$$

Гладкість може розглядатися також як міра „осциляційної” поведінки функції. Тому в класі диференційовних функцій більш гладкою вважають ту функцію, яка менше осцилює. В частотній області прояв осциляційних властивостей відповідає зміщенню енергії в напрямку високих частот, а рівень гладкості може бути вимірний шириною смуги частот. Виходячи з цього, функціонал гладкості $R[y]$ задають формулою:

$$R[y] = \int_{R^d} \frac{|\tilde{f}(s)|^2}{\tilde{G}(s)}, \quad (3.8)$$

де знак „ \checkmark ” над функцією означає, що взято перетворення Фур’є; $\frac{1}{\checkmark G(s)}$ — фільтр верхніх частот, що утворений за допомогою деякої додатної спадної функції $\checkmark G(s)$.

В [17] доведено, що функція f , яка мінімізує даний функціонал, має вигляд:

$$\mathbf{y} = f(\mathbf{X}) = \sum_{i=1}^N c_i \mathbf{G}(\mathbf{X} - \mathbf{X}^{(i)}) + \sum_{l=1}^k d_l p_l(\mathbf{X}), \quad (3.9)$$

де $\{p_l(\mathbf{X})\}_{l=1}^k$ — базис у k -вимірному нуль-просторі N функціонала $\mathbf{R}[\mathbf{y}]$; коефіцієнти d_l та c_i залежать від вхідних даних і задовольняють лінійній системі:

$$\begin{cases} (\mathbf{G} + \lambda \mathbf{I})\mathbf{c} + \mathbf{P}\mathbf{d} = \mathbf{y}, \\ \mathbf{P}\mathbf{c} = 0, \end{cases} \quad (3.10)$$

де \mathbf{I} — одинична матриця.

Існування розв’язку системи (3.10) визначається існуванням розв’язку відповідної варіаційної задачі. При $\lambda = 0$ маємо випадок, для якого розв’язок системи залежить від властивостей функції \mathbf{G} .

Отже, регуляризація ШНМ може бути проведена шляхом формування вихідних даних у відповідності до функції (3.9) для заданого класу базисних функцій \mathbf{G} .

3.3.3. Усереднення

Зниження реальної похибки реалізації деякої функції $\mathbf{y} = f(\mathbf{X})$ на ШНМ досягають також шляхом усереднення результатів на множині моделей. Різні варіанти результатів можуть бути одержаними у випадку застосування різних методик навчання ШНМ на одній і тій самій навчальній вибірці. Нехай маємо K різних методик навчання, кожна з яких забезпечує одержання відповідного наближення $\checkmark \mathbf{y}_k$ функції \mathbf{y} .

Очевидно, що дані наближення знаходяться в деякому околі точного значення. Оскільки невідомо, який із результатів має мінімальну реальну похибку, то очевидно, що середнє значення матиме меншу похибку, ніж найгірший результат. Тому одержимо середнє значення для кожного елементу навчальної вибірки:

$$\checkmark \mathbf{y}^{(n)} = \frac{1}{K} \sum_{k=1}^K \checkmark \mathbf{y}_k^{(n)}. \quad (3.11)$$

Середньоквадратична похибка для k -ї методики навчання:

$$\mathbf{E}_k = \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \checkmark \mathbf{y}_k^{(n)} \right)^2. \quad (3.12)$$

Тоді очікувана середньоквадратична похибка після усереднення дорівнюватиме:

$$\mathbf{E} = \frac{1}{K} \sum_{k=1}^K \mathbf{E}_k. \quad (3.13)$$

Середньоквадратичну похибку можемо також одержати, використавши середнє вихідне значення:

$$\checkmark \mathbf{E} = \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \checkmark \mathbf{y}^{(n)} \right)^2. \quad (3.14)$$

Для порівняння \mathbf{E} та $\checkmark \mathbf{E}$ застосуємо нерівність Коші–Шварца:

$$\left(\mathbf{a}^T \checkmark \mathbf{y} \right)^2 \leq \|\mathbf{a}\|^2 \|\checkmark \mathbf{y}\|^2, \quad (3.15)$$

для двох K -вимірних векторів:

$$\mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix}, \quad \checkmark \mathbf{y} = \begin{pmatrix} \checkmark \mathbf{y}_1 \\ \checkmark \mathbf{y}_2 \\ \dots \\ \checkmark \mathbf{y}_K \end{pmatrix}. \quad (3.16)$$

Представимо ліву та праву частину нерівності (3.15) у покомпонентному вигляді:

$$\left(\mathbf{a}^T \tilde{\mathbf{y}}\right)^2 = \left(\sum_{k=1}^K \tilde{\mathbf{y}}_k\right)^2, \quad (3.17)$$

$$\|\mathbf{a}\|^2 \|\tilde{\mathbf{y}}\|^2 = K \sum_{k=1}^K \tilde{\mathbf{y}}_k^2. \quad (3.18)$$

Тоді нерівність (3.15) набуде вигляду:

$$\left(\sum_{k=1}^K \tilde{\mathbf{y}}_k\right)^2 \leq K \sum_{k=1}^K \tilde{\mathbf{y}}_k^2. \quad (3.19)$$

Розкриємо квадрат різниці у виразі (3.13) після підстановки в нього (3.12):

$$\begin{aligned} \mathbf{E} &= \frac{1}{KN} \sum_{k=1}^K \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}_k^{(n)}\right)^2 = \\ &= \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)}\right)^2 - \\ &\quad - \frac{2}{KN} \sum_{k=1}^K \sum_{n=1}^N \mathbf{y}^{(n)} \tilde{\mathbf{y}}_k^{(n)} + \frac{1}{KN} \sum_{k=1}^K \sum_{n=1}^N \left(\tilde{\mathbf{y}}_k^{(n)}\right)^2. \end{aligned} \quad (3.20)$$

Розкриємо також квадрат різниці у виразі (3.14) після підстановки в нього (3.11):

$$\begin{aligned} \tilde{\mathbf{E}} &= \frac{1}{N} \sum_{n=1}^N \left[\mathbf{y}^{(n)} - \left(\frac{1}{K} \sum_{k=1}^K \tilde{\mathbf{y}}_k^{(n)} \right) \right]^2 = \\ &= \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)}\right)^2 - \\ &\quad - \frac{2}{KN} \sum_{k=1}^K \sum_{n=1}^N \mathbf{y}^{(n)} \tilde{\mathbf{y}}_k^{(n)} + \frac{1}{K^2 N} \sum_{n=1}^N \left(\sum_{k=1}^K \tilde{\mathbf{y}}_k^{(n)} \right)^2. \end{aligned} \quad (3.21)$$

Порівняємо (3.20) і (3.21) та проведемо спрощення:

$$\tilde{\mathbf{E}} \approx \mathbf{E},$$

$$\left(\sum_{k=1}^K \tilde{\mathbf{y}}_k\right)^2 \approx K \sum_{k=1}^K \tilde{\mathbf{y}}_k^2 \quad (3.22)$$

Порівнюючи (3.22) з нерівністю Коші–Шварца (3.19), одержимо нерівність $\tilde{\mathbf{E}} \leq \mathbf{E}$. Отже, використання усередненого вихідного вектора веде до зниження середньоквадратичної похибки, яка є складовою частиною реальної похибки ШНМ.

Усереднення також може бути застосованим при використанні різних типів ШНМ для реалізації однієї і тієї ж функції $\mathbf{y} = f(\mathbf{X})$. У цьому випадку усереднене вихідне значення є середньозваженим:

$$\tilde{\mathbf{y}}^{(n)} = \sum_{k=1}^K \boldsymbol{\mu}_k \tilde{\mathbf{y}}_k^{(n)}, \quad (3.23)$$

$$\text{де } \sum_{k=1}^K \boldsymbol{\mu}_k = 1, \quad \boldsymbol{\mu}_k = \frac{\mathbf{E}_k^{-1}}{\sum_{i=1}^K \mathbf{E}_i^{-1}}.$$

Підставивши (3.23) в (3.14), визначимо середньоквадратичну похибку:

$$\begin{aligned} \tilde{\mathbf{E}} &= \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}^{(n)}\right)^2 = \frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \sum_{k=1}^K \boldsymbol{\mu}_k \tilde{\mathbf{y}}_k^{(n)} \right)^2 = \\ &= \sum_{k=1}^K \boldsymbol{\mu}_k^2 \left[\frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}_k^{(n)}\right)^2 \right] - \sum_{k=1}^K \sum_{\substack{j=1 \\ j \neq k}}^K \boldsymbol{\mu}_j \boldsymbol{\mu}_k \times \\ &\quad \times \left[\frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}_k^{(n)}\right) \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}_j^{(n)}\right) \right]. \end{aligned} \quad (3.24)$$

У випадку відсутності кореляції між моделями ШНМ одержимо наближено:

$$\tilde{\mathbf{E}} \approx \sum_{k=1}^K \boldsymbol{\mu}_k^2 \left[\frac{1}{N} \sum_{n=1}^N \left(\mathbf{y}^{(n)} - \tilde{\mathbf{y}}_k^{(n)}\right)^2 \right] = \sum_{k=1}^K \boldsymbol{\mu}_k^2 \mathbf{E}_k. \quad (3.25)$$

Підставивши значення μ_k в (3.25), отримаємо нерівність:

$$\bar{E} = \frac{1}{\sum_{k=1}^K E_k^{-1}} < \min_k(E_k). \quad (3.26)$$

Отже, використання середньозваженого усереднення дозволяє одержати середньоквадратичну похибку, меншу від похибки найточнішої моделі за умови відсутності кореляції між моделями та коректній оцінці середньоквадратичної похибки на кожній з них.

3.3.4. Методи відсікання

Відсікання вперше було застосовано для методу штучного інтелекту, який має назву дерева альтернатив, з метою його спрощення без втрати точності. У нейронних мережах цей метод дозволяє виявити та відкинути зайві міжнейронні зв'язки і надлишкові нейрони.

За ознакою вибору критерію відсікання розрізняють методи відсікання зв'язків, що мають мінімальні значення вагових коефіцієнтів, та методи відсікання зв'язків з мінімальною дисперсією.

Методи відсікання з $\min(W)$ можуть використовувати як детерміновані, так і імовірнісні оцінки. Наведемо деякі приклади таких оцінок:

- при мінімізації структури ШНМ шляхом відкидання зв'язків з мінімальними ваговими коефіцієнтами збільшення похибки компенсується додаванням середнього внеску відкинутого зв'язку до вагових коефіцієнтів тих зв'язків, що залишилися;
- оцінка чутливості ШНМ до відкидання одного з її елементів визначається як сума змін вагових коефіцієнтів в процесі повторного сеансу навчання [113];
- оцінка результатів відсікання проводиться на основі формування цільової функції [141];

- відсікання зв'язку відбувається як результат оцінки імовірності того, що величина даного вагового коефіцієнта прямує до нуля [74].

Методи відсікання з $\min(\sigma^2)$ базуються на тому факті, що сигнали, які мало змінюються на протязі всього періоду обробки вхідних даних, не потребують обробки, і можуть бути замінені деякими константами. Оскільки такі константи не потребують введення додаткових елементів архітектури, то їх застосування спрощує топологію ШНМ.

Відкидання зв'язків для багатьох методів відсікання є першим етапом оптимізації топології. Другий етап полягає у відкиданні тих нейронів, вхідні зв'язки яких задовольняють відповідні умови. Для методів відсікання ці умови найчастіше визначаються в залежності від способу обчислення коефіцієнта впливу.

Один із перших відомих методів відсікання [140] полягає у визначенні „каркасу” нейронної мережі, що містить тільки необхідні для вирішення задачі елементи. Критерієм важливості того чи іншого елемента мережі є різниця середньоквадратичних похибок навчальних послідовностей E , які обчислюються з даним елементом і без нього. Неважливі за даним критерієм елементи можуть бути відкинуті.

Розглянемо найбільш відомі методи відсікання, що застосовуються на сучасному етапі розвитку ШНМ прямого поширення з метою їх оптимізації.

3.3.4.1. Метод „пригнічення”

Метод „пригнічення”, який забезпечує відсікання тільки зайвих зв'язків [42], базується на застосуванні подвійної цільової функції E_W на етапі навчання нейронної мережі.

$$E_W = E + E_C, \quad (3.27)$$

де E_C — функція „пригнічення”.

Значення функції „пригнічення” в значній мірі залежить від типу вирішуваних задач та прийнятих стратегій навчання мережі. Одним із поширених критеріїв „пригнічення” є

$$E_C = \varepsilon \sum_{j=1}^m n_j \sum_{i=1}^{n_j} w_{ij}^2, \quad (3.28)$$

де n_j — кількість вагових коефіцієнтів w_{ij} нейрона j ; ε — коефіцієнт, що задає вплив критерію „пригнічення”.

Функція „пригнічення” може бути ускладнена шляхом введення індивідуального коефіцієнта впливу ε_i для кожного вагового коефіцієнта. Цей метод не завжди відносять до методу відсікання з тієї причини, що найчастіше він використовується лише для зменшення впливу деяких вагових коефіцієнтів і не містить критерію відсікання. Якщо ввести такий критерій, наприклад, $w_{ij} < \text{const}$, одержуємо всі можливості для застосування операції відсікання.

Загально визнаний недолік методу „пригнічення” полягає в тому, що відсутні формалізовані підходи ефективного вибору коефіцієнта ε , який є ключовою ланкою даного методу. До того ж, крім величини вагових коефіцієнтів, які істотно впливають на похибку вихідного вектора, існують ще і додаткові фактори. До таких факторів найчастіше відносять амплітуду вхідного сигналу та взаємний вплив вагових коефіцієнтів із різних шарів багат шарової ШНМ.

3.3.4.2. OBD-метод

З точки зору узагальненого підходу значно ефективнішим є так званий метод оптимального ушкодження („Optimal Brain Damage”) [124], скорочено — OBD-метод. Цей метод використовує апроксимацію другого порядку цільової функції, представленої рядом Тейлора:

$$E = E_0 + \sum_{j=1}^m \frac{\partial E}{\partial w_j} \delta w_j + \frac{1}{2} \sum_{j=1}^m \frac{\partial^2 E}{\partial w_j^2} \delta w_j^2 + \frac{1}{2} \sum_{j=1}^m \sum_{\substack{i=1 \\ i \neq j}}^n \frac{\partial^2 E}{\partial w_j \partial w_i} \delta w_j \delta w_i + o(\|\Delta W\|^3), \quad (3.29)$$

де δw_j — компоненти матриці δW , яка задає кроки зміни вагових коефіцієнтів.

Якщо відсікання проводять для добре навченої ШНМ, то $\sum_{j=1}^m \frac{\partial E}{\partial w_j} \delta w_j \approx 0$, оскільки в цьому випадку E — мінімальна.

При достатньо малих змінах доданком $o(\|W\|^3)$ також можна знехтувати. Тому критерієм відсікання в даному методі є коефіцієнт впливу S_j , для визначення якого використовують другі похідні цільової функції E , що утворюють матрицю H :

$$H = \begin{bmatrix} \frac{\partial E^2}{\partial w_1^2} & \frac{\partial E^2}{\partial w_1 \partial w_2} & \frac{\partial E^2}{\partial w_1 \partial w_3} & \dots & \frac{\partial E^2}{\partial w_1 \partial w_m} \\ \frac{\partial E^2}{\partial w_2 \partial w_1} & \frac{\partial E^2}{\partial w_2^2} & \frac{\partial E^2}{\partial w_2 \partial w_3} & \dots & \frac{\partial E^2}{\partial w_2 \partial w_m} \\ \frac{\partial E^2}{\partial w_3 \partial w_1} & \frac{\partial E^2}{\partial w_3 \partial w_2} & \frac{\partial E^2}{\partial w_3^2} & \dots & \frac{\partial E^2}{\partial w_3 \partial w_m} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial E^2}{\partial w_n \partial w_1} & \frac{\partial E^2}{\partial w_n \partial w_2} & \frac{\partial E^2}{\partial w_n \partial w_3} & \dots & \frac{\partial E^2}{\partial w_n \partial w_m} \end{bmatrix}. \quad (3.30)$$

Значне зростання розмірів ($\sim n^3$) цієї матриці за умови збільшення елементів ШНМ є основною обчислювальною

проблемою методу оптимального ушкодження. Матрицю \mathbf{H} можна спростити, якщо не враховувати вплив нейрона, який відсікається, на цільові функції інших нейронів. У цьому випадку матриця \mathbf{H} міститиме тільки діагональні елементи. Відкинувши член ряду Тейлора

$$\frac{1}{2} \sum_{j=1}^m \sum_{\substack{i=1 \\ i \neq j}}^n \frac{\partial \mathbf{E}^2}{\partial \mathbf{w}_i \partial \mathbf{w}_j} \delta \mathbf{w}_i \delta \mathbf{w}_j = 0,$$

одержимо формулу для обчислення зміни цільової функції:

$$\delta \mathbf{E} = \frac{1}{2} \sum_{j=1}^m \frac{\partial \mathbf{E}}{\partial \mathbf{w}_j^2} \delta \mathbf{w}_j^2. \quad (3.31)$$

Позначивши через h_{jj} вектор діагональних елементів матриці \mathbf{H} , визначатимемо коефіцієнт впливу як похибку, що вноситься у вихідний сигнал відсіканням нейрона j :

$$S_j = \delta \mathbf{E} = \frac{h_{jj} \delta \mathbf{w}_j^2}{2}. \quad (3.32)$$

У випадку достатньо малих значень коефіцієнта впливу виконують операцію відсікання, тобто вилучають відповідний нейрон з усіма його зв'язками. Така дія може вплинути на величину похибки. Тому виконують повторний сеанс навчання нейронної мережі. Даний процес може циклічно повторюватись і обмежений тільки допустимою величиною похибки нейронної мережі. Однак ця похибка може значно зрости у випадку:

$$h_{ij} \delta \mathbf{w}_i \delta \mathbf{w}_j > h_{jj} \delta \mathbf{w}_j^2.$$

3.3.4.3. OBS-метод

Для врахування впливу всіх елементів матриці \mathbf{H} розроблено метод оптимальної хірургії (Optimal Brain Surgeon), що отримав назву OBS-методу [95,167]. Він також базується на апроксимації локальної похибки, яка виникає при зміні вагових коефіцієнтів \mathbf{W} на величину $\Delta \mathbf{W}$, за формулою:

$$\mathbf{E}(\mathbf{W} + \delta \mathbf{W}) = \mathbf{E}(\mathbf{W}) + \frac{1}{2} \delta \mathbf{W}^T \mathbf{H} \delta \mathbf{W}. \quad (3.33)$$

Однією з важливих умов реалізації методу є повільні зміни вектора вагових коефіцієнтів $\mathbf{w}_j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})$ довільного нейрона j навколо локального мінімуму з метою досягнення співвідношення

$$\mathbf{w}_j + \delta \mathbf{w}_j = 0 \quad (3.34)$$

і, таким чином, зменшення внеску даного нейрона у формування вихідного вектора. Використаємо вектор

$$\mathbf{e}_j = \begin{pmatrix} 0 \\ \cdot \\ 0 \\ 1 \\ 0 \\ \cdot \\ 0 \end{pmatrix}, \quad (3.35)$$

що складатиметься з нульових елементів за виключенням одиничного елементу з номером j одиниці. Це дозволяє записати умову корекції вагових коефіцієнтів (3.34) у вигляді:

$$\delta \mathbf{W} \mathbf{e}_j = -\mathbf{w}_j. \quad (3.36)$$

Коефіцієнт впливу S_j для \mathbf{w}_j визначається величиною зміни похибки:

$$S_j = \delta E_j = E(W + \delta W) - E(W) = \frac{1}{2} \delta W^T H \delta W. \quad (3.37)$$

Обчислення S_j виконують за методом множників Лагранжа. Для цього необхідно сформулювати функцію:

$$C(\delta W) = \frac{1}{2} \delta W^T H \delta W + \lambda (\delta W e_j + w_j). \quad (3.38)$$

З метою визначення екстремуму візьмемо похідну $\frac{C(\delta W)}{\delta W}$ та прирівняємо її до нуля

$$\frac{\partial C(\delta W)}{\delta W} = \delta W H + \lambda e_j = 0. \quad (3.39)$$

З (3.39) отримаємо:

$$\delta W = -\lambda H^{-1} e_j. \quad (3.40)$$

Домноживши (3.40) на e_j^T та використавши (3.36), визначимо λ :

$$\left. \begin{aligned} e_j^T \delta W &= e_j^T H^{-1} e_j \\ e_j^T \delta W &= -w_j \end{aligned} \right\} \Rightarrow \lambda = \frac{w_j}{e_j^T H^{-1} e_j}. \quad (3.41)$$

Підставивши (3.41) в (3.40), одержимо мінімальний крок зміни вагових коефіцієнтів:

$$\delta W = \frac{-w_j H^{-1} e_j}{e_j^T H^{-1} e_j}. \quad (3.42)$$

Використовуючи (3.37) та (3.42), запишемо вираз для коефіцієнта впливу:

$$S_j = \frac{1}{2} \delta W^T H \delta W = \frac{w_j^2}{e_j^T H^{-1} e_j}. \quad (3.43)$$

Як і в OBD-методі, формула для визначення S_j є ядром, що забезпечує ефективне функціонування ШНМ.

Послідовність дій також нагадує OBD-метод і складається з наступних стадій:

1. Проводять навчання ШНМ з урахуванням мінімальних можливих значень W та умови $\delta E = 0$.
2. Повторюють наступні дії, доки збільшення середньоквадратичної похибки не досягне критичної межі.
 - 2.1. Відсікають елементи з найменшим коефіцієнтом впливу S_j .
 - 2.2. Змінюють вагові коефіцієнти W з кроком δW , що визначається за формулою (3.42).
 - 2.3. Проводять навчання нової ШНМ.

Основним недоліком OBS-методу є труднощі, що пов'язані з обчисленням прямої та оберненої матриці H . В [96] запропоновано спосіб ітеративних наближених обчислень з метою зменшення їх об'єму.

3.3.4.4. OLS-метод

Метод відсікання на основі найменших квадратів (Orthogonal-Least-Squares), або скорочено OLS-метод [99], був розроблений переважно з метою значного зменшення обчислень, необхідних для визначення коефіцієнтів впливу.

Розглянемо лінійну функцію регресії:

$$z = P\theta + \xi, \quad (3.44)$$

де $z = (z_1, z_2, \dots, z_n)^T \in \mathbf{R}^n$ — вектор залежних змінних;

$$P = \begin{pmatrix} p_{11} & \dots & p_{1m} \\ \dots & \dots & \dots \\ p_{n1} & \dots & p_{nm} \end{pmatrix} \in \mathbf{R}^n \times \mathbf{R}^m$$
 — матриця коефіцієнтів регресії;

$\theta = (\theta_1, \theta_2, \dots, \theta_m) \in \mathbf{R}^m$ — вектор параметрів, що підлягають оцінці; $\xi = (\xi_1, \xi_2, \dots, \xi_n) \in \mathbf{R}^n$ — вектор поправочних коефіцієнтів.

Для моделі (3.44) наближення за методом найменших квадратів вектора параметрів θ визначається з виразу:

$$\tilde{\theta} = (P^T P)^{-1} P^T z. \quad (3.45)$$

Формування чисельного розв'язку рівняння (3.45) базується на процедурі ортогоналізації, що наведена в [43]. Представимо матрицю регресії у вигляді добутку:

$$P^T P = A^T D A, \quad (3.46)$$

де A — верхньотрикутна матриця з одиничними діагональними елементами; D — діагональна матриця з додатними діагональними елементами.

Ввівши заміни $B = P A^{-1}$ та $g = A \theta$, виразимо функцію регресії (3.44) через ортогональну матрицю

$$B = \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{nm} \end{pmatrix} \in \mathbf{R}^n \times \mathbf{R}^m,$$

$$z = B g + \xi. \quad (3.47)$$

Необхідність такого перетворення впливає з того, що елементи допоміжної матриці B можуть бути одержані рекурсивно з виразу:

$$\begin{aligned} b_{1l} &= p_{1l}, \\ b_{kl} &= p_{kl} - \sum_{i=1}^{k-1} \alpha_{ik} b_{il}, \quad k = 2, \dots, m, \end{aligned} \quad (3.48)$$

$$\text{де } \alpha_{ik} = \frac{\sum_{l=1}^n b_{il} p_{kl}}{\sum_{l=1}^n b_{il}^2}.$$

Допоміжний параметр $g = (g_1, \dots, g_m)^T$ визначається з виразу

$$g = D^{-1} B^T z - D^{-1} B^T \xi, \quad (3.49)$$

де матриця D , виходячи з (3.46), дорівнює:

$$D = B^T B = (P A^{-1})^T (P A^{-1}).$$

Наближено, без складової шуму, отримуємо з (3.49)

$$\tilde{g} = D^{-1} B^T z, \quad (3.50)$$

або в поелементному вигляді:

$$\tilde{g}_i = \frac{\sum_{l=1}^n b_{il} z_l}{\sum_{l=1}^n b_{il}^2}, \quad i = 1, \dots, m. \quad (3.51)$$

Використовуючи наближені значення допоміжного параметра \tilde{g} , задамо ітераційний процес обчислення наближених значень $\tilde{\theta}$ основного параметра θ :

$$\tilde{\theta} = \tilde{g} - (A - I) \tilde{\theta}, \quad (3.52)$$

або в поелементному вигляді:

$$\begin{aligned} \tilde{\theta}_m &= \tilde{g}_m, \\ \tilde{\theta}_i &= \tilde{g}_i - \sum_{k=i+1}^m \alpha_{ik} \tilde{\theta}_k, \quad i = m-1, \dots, 1. \end{aligned} \quad (3.53)$$

Помноживши (3.47) на $z^T = g^T B^T + \xi^T$, одержимо

$$z^T z = g^T B^T B g + \xi^T \xi + g^T B^T \xi + \xi^T B g. \quad (3.54)$$

Спростимо (3.54), враховуючи відсутність кореляції білого шуму з вхідними даними,

$$z^T z = g^T B^T B g + \xi^T \xi = \sum_{i=1}^m g_i^2 b_i^T b_i + \xi^T \xi, \quad (3.55)$$

де $b_i = (b_{1i}, \dots, b_{ni})^T$.

Тоді середнє значення на навчальній вибірці

$$\frac{z^T z}{n} = \sum_{i=1}^m \frac{g_i^2 b_i^T b_i}{n} + \frac{\xi^T \xi}{n},$$

де $\sigma^2 = \frac{\xi^T \xi}{n}$ — дисперсія білого шуму, що виникає в процесі моделювання функції регресії. Тому вклад, який вносить вхідний сигнал, дорівнює:

$$S = \sum_{i=1}^m s_i = \sum_{i=1}^m \frac{g_i^2 b_i^T b_i}{n}, \quad (3.56)$$

де $s_i = \frac{g_i^2 b_i^T b_i}{n}$ — коефіцієнт впливу нейрона i , який дорівнює його внеску в дисперсію залежної змінної.

Для використання даного підходу до визначення тих нейронів ШНМ, що підлягають відсіканню, задають функціонування прихованого шару

$$\begin{cases} \varphi = \mathbf{W}\mathbf{X} + \varepsilon, \\ \mathbf{v} = f(\varphi), \end{cases} \quad (3.57)$$

а в ролі функції регресії (3.44) застосовують перше рівняння системи (3.57).

Алгоритм, який реалізує OLS-метод відсікання, показаний на рис. 3.2. Через d позначено величину порога коефіцієнта впливу s_i i -го нейрона, змінна K задає поточну кількість нейронів після чергового етапу відсікання. Лічильник k відображає кількість нейронів, для яких коефіцієнт впливу перевищує поріг на поточному етапі. Лічильник i відповідно задає поточний номер нейрона, для якого визначається коефіцієнт впливу.

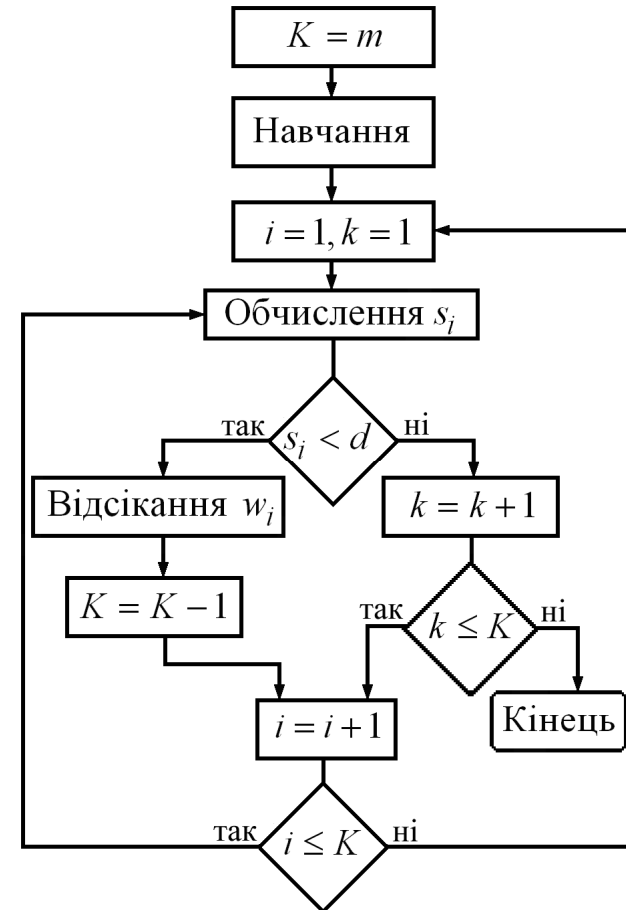


Рис.3.2. Алгоритм OLS-методу.

Предметом розгляду методів відсікання є попередньо створена нейронна мережа. Параметри її топології, як правило, є надлишковими, що зумовлює збільшення часу навчання та інші негативні явища. Узагальнення за допомогою методів відсікання дозволяє оптимізувати топологію ШНМ при середньому значенні реальної похибки. Тому актуальним є питання розгляду можливості прямої побудови ШНМ з оптимальною топологією.

3.3.5. Методи нарощування

Методи нарощування призначені для поступового нарощування нейронної мережі за умови збереження її оптимальності на кожному з етапів [183]. Такий підхід дозволяє спростити процедуру створення ШНМ.

3.3.5.1. Метод каскадної кореляції

Одним із перших методів нарощування, що набули практичного застосування, вважають метод каскадної кореляції [71]. Цей метод задає процедуру побудови каскадної ШНМ, яка складається з вихідного шару нейронів та деякої кількості прихованих шарів, кожен з яких містить лише один нейрон. Процес нарощування відбувається шляхом послідовного додавання чергового нейрона, що формує свій прихований шар. За допомогою кожного наступного нейрона намагаються зменшити реальну похибку ШНМ за рахунок пошуку максимуму кореляції між вихідним сигналом нового нейрона та тією залишковою частиною похибки, яку необхідно нейтралізувати. Процес нарощування нейронів прихованих шарів показаний на рис. 3.3.

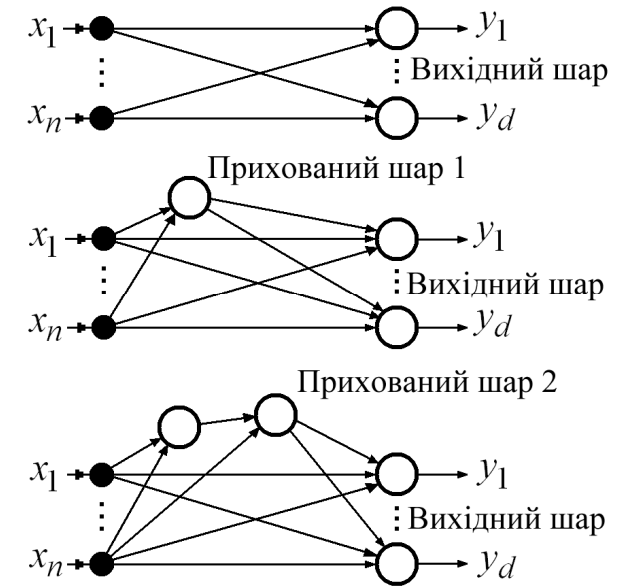


Рис.3.3. Процес нарощування ШНМ.

Він починається зі структури з n входами та d виходами, у якій повністю відсутні приховані шари. Додавання прихованих шарів відбувається послідовно таким чином, що на кожний новий нейрон, який спочатку одержує статус *кандидата*, подаються всі вхідні сигнали, а також вихідні сигнали всіх попередніх прихованих шарів. При цьому вихід даного нейрона залишається непідключеним.

В такому стані починають черговий сеанс навчання з використанням навчальної вибірки, метою якого є досягнення максимального значення суми

$$S = \sum_{k=1}^d \left| \sum_{i=1}^N (\mathbf{v}^{(i)} - \bar{\mathbf{v}}) (\mathbf{E}_k^{(i)} - \bar{\mathbf{E}}_k) \right|, \quad (3.58)$$

де k — поточний вихід нейронної мережі, для якого відбувається вимірювання похибки; i — номер поточного елементу навчальної вибірки; N — кількість елементів навчальної вибірки; $\mathbf{v}^{(i)}$ — поточне значення вихідного сигналу кандидата; $\bar{\mathbf{v}}, \bar{\mathbf{E}}_k$ — середні значення виходу кандидата та похибки ШНМ, взятих на навчальній вибірці.

Для одержання екстремуму (3.58) визначимо частинні похідні $\frac{\partial S}{\partial \mathbf{w}_j}$ по кожному із входів кандидата

$$\frac{\partial S}{\partial \mathbf{w}_j} = \sum_{k=1}^d \sum_{i=1}^N \sigma_k (\mathbf{E}_k^{(i)} - \bar{\mathbf{E}}_k) \varphi'(\mathbf{s}^{(i)}) \mathbf{I}_j^{(i)}, \quad (3.59)$$

де σ_k — знак кореляції між виходом кандидата та відповідним виходом ШНМ; $\varphi'(\mathbf{s}^{(i)})$ — похідна активаційної функції кандидата від суми входів $\mathbf{s}^{(i)}$, що сформовані i -м елементом навчальної вибірки; $\mathbf{I}_j^{(i)}$ — вхідний сигнал, який отримує кандидат від нейрона j .

Обчисливши $\left\{ \frac{\partial S}{\partial \mathbf{w}_j} \right\}_{j=1}^J$, можемо максимізувати S ,

використовуючи ознаку зростання градієнта. Досягнувши таким чином бажаної кореляції вихідного сигналу кандидата з похибкою ШНМ, фіксують значення вагових коефіцієнтів на його входах та підключають вихід до входів нейронів, як

показано на рис. 3.3. Оскільки формула (3.58) відображає тільки абсолютне значення кореляції, то знак вихідного сигналу нейрона, при підключенні його до активної нейронної мережі, вибирають таким чином, щоб зменшити величину похибки її вихідних сигналів.

Даний підхід також допускає використання групи нейронів у ролі кандидатів на формування чергового прихованого шару. Практична перевірка показала, що в цьому випадку неможливо досягти кращого результату, ніж той, що виникає після підключення нейрона з максимальною кореляцією. Однак застосування групи нейронів все ж має свої переваги, до яких відносять зменшення імовірності попадання в локальний екстремум, а також зменшення часу навчання у випадку застосування паралельних алгоритмів пошуку максимуму кореляції для всіх нейронів групи одночасно.

Недоліком методу каскадної кореляції є занадто швидкий перехід ШНМ до стану зарегульованості, що супроводжується зростанням реальної похибки.

3.3.5.2. Метод ітераційного нарощування

Метод ітераційного нарощування дозволяє уникнути небезпеки зарегульованості. Він є простим повторенням процедури синтезу ШНМ з послідовною оцінкою реальної похибки після застосування одного з методів регуляризації. Фактично, такий підхід полягає у послідовності спроб моделювання однієї і тієї ж функції, заданої вхідним набором даних та навчальною вибіркою, на упорядкованій множині топологій ШНМ зі зростаючою кількістю нейронів прихованого шару. Алгоритм даного методу показаний на рис. 3.4.

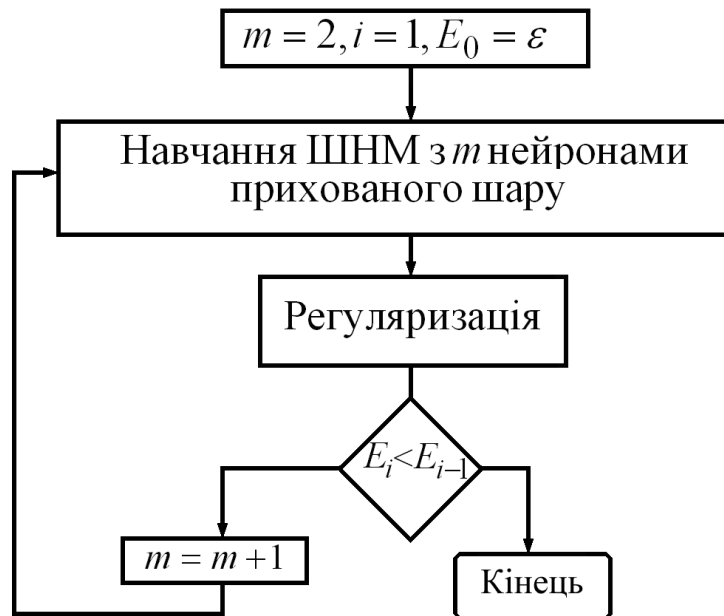


Рис.3.4. Алгоритм ітераційного нарощування.

Нарощування починають зі структури, що містить мінімум 2 нейрони прихованого шару, оскільки структура з одним нейроном еквівалентна задачі лінійної регресії, яка має аналітичне вирішення. Чергування етапів навчання з регуляризацією забезпечує нарощування ШНМ в узагальненому вигляді. Якщо результат нарощування привів до зменшення реальної похибки, то процес нарощування продовжується, у протилежному випадку вважаємо, що оптимальної топології досягнуто.

Розділ 4

МЕТОДИ НАВЧАННЯ БЕЗ УЧИТЕЛЯ

Під час навчання з учителем критерієм якості є мінімізація цільової функції, яка визначає відхилення реакції ШНМ на вхідний сигнал від ідеальної реакції, заданої у навчальній вибірці. Тому навчальна вибірка виконує роль учителя, і саме від неї залежить ефективність функціонування ШНМ у реальних умовах.

Навчанням без учителя будемо називати такі методи, що не потребують зовнішнього сигналу, який інформував би нейронну мережу про те, наскільки правильною є її реакція на вхідний сигнал. Для оцінки реакції ШНМ процедура навчання без учителя повинна синтезувати свій внутрішній прихований сигнал, що базується на здатності нейронної мережі передбачати характер зміни вхідних даних або, у загальному випадку, визначати деяку внутрішню міру якості. Тому при навчанні без учителя дуже важливою є мета навчання. Необхідно знати, якою ми хочемо бачити нейронну мережу, тобто, для розв'язання яких задач планують її використати.

Таких застосувань, що мають визначатися за характером обробки вхідних даних, існує значна кількість. Серед них найвідомішими є:

- Аналіз важливих компонентів (Principal Component Analysis-PCA);
- Аналіз незалежних компонентів (Independent Component Analysis - ICA);
- Кластерний аналіз (Clustering);
- Багатовимірне масштабування (Multidimensional Scaling);
- Виявлення асоціативних зв'язків (Associative rule discovery).

Всі ці застосування базуються на фундаментальному підході, який полягає у тому, що процедура навчання без учителя завжди намагається сформувати у нейронної мережі здатність віднайти

деяку упорядковану структуру у вхідному потоці даних. У реальному житті на вхідні дані впливає значна кількість некорельованих факторів. Проте такі дані є відображенням реальних фізичних процесів і, як правило, далекі від випадковості. Навчання без учителя має на меті навчити ШНМ розпізнавати ці фізичні процеси. Основною проблемою даного підходу є значна кількість таких процесів та зв'язків між ними. Намагання розв'язати цю проблему є причиною створення все нових і нових методів навчання без учителя.

4.1. Аналіз важливих компонентів

Аналіз важливих компонентів (PCA) [27,111] — це широко відомий метод, що застосовується для зниження розмірності елементів даних. Зазвичай елемент даних задають n -вимірним вектором, який описують вектором меншої розмірності. Ідея методу важливих компонентів полягає у побудові ефективної лінійної апроксимації багатовимірних даних. Між деякими елементами x_i вхідного вектора \mathbf{x} , як правило, існує лінійна кореляція. У такому випадку вектор даних може бути представлений меншою кількістю елементів, які взаємно не корельовані.

Нехай маємо вхідний набір даних $\mathbf{X} = \{\mathbf{x}^{(k)}\}_{k=1}^N$, де $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$ — n -вимірний вектор вхідної послідовності даних. Необхідно за допомогою множини базисних векторів $\{\mathbf{c}_k\}_{k=1}^N$ сформувати апроксимацію вхідного набору $\mathbf{Z} = \{\mathbf{z}^{(k)}\}_{k=1}^M$, де $\mathbf{z}^{(k)} = (z_1^{(k)}, z_2^{(k)}, \dots, z_m^{(k)})^T$ за умови $m < n$. Матриця коваріантності вхідного набору \mathbf{X} має такий вигляд:

$$C = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1j} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{21} & \dots & \sigma_{2j} & \dots & \sigma_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sigma_{i1} & \sigma_{i2} & \dots & \sigma_{ij} & \dots & \sigma_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nj} & \dots & \sigma_{nn} \end{pmatrix}, \quad (4.1)$$

$$\text{де } \sigma_{ij} = \frac{1}{N} \sum_{k=1}^N x_i^{(k)} x_j^{(k)} - \left(\frac{1}{N} \sum_{k=1}^N x_i^{(k)} \right) \left(\frac{1}{N} \sum_{k=1}^N x_j^{(k)} \right).$$

Матриця C задає кореляцію між елементами вектора вхідного набору X . Отже, для досягнення поставленої мети необхідно за допомогою лінійного перетворення

$$z = Gx \quad (4.2)$$

сформувані набір даних з використанням діагональної матриці коваріантності, що свідчило б про відсутність кореляції між елементами вектора даних.

Виходячи з означення елементів матриці коваріантності, робимо висновок, що вона завжди є симетричною. Тому, використовуючи спектральну теорему, можна виконати діагоналізацію матриці C за допомогою ортогонального перетворення. Для цього застосуємо деяку ортогональну матрицю Q , для якої справедливе співвідношення $Q^T Q = I$ і яка дозволяє отримати діагональну матрицю A із виразу:

$$A = Q^T C Q. \quad (4.3)$$

Якщо для лінійного перетворення (4.2) виберемо матрицю G із умови $G = Q^T$, то матриця коваріантності для апроксимованого набору Z буде еквівалентною діагональній матриці A .

Власні вектори матриці Q співпадають зі стовпцями матриці C , а власні значення матриці коваріантності C дорівнюють діагональним елементам матриці A .

Елементи апроксимованого вектора z одержують із виразу:

$$z_i = c_i^T x, \quad (4.4)$$

де c_i^T — i -й власний вектор матриці коваріантності C .

Виходячи з означення власного вектора, робимо висновок, що справедливе співвідношення:

$$C c_i = \lambda_i c_i, \quad (4.5)$$

де λ_i — i -те власне значення матриці C і одночасно відповідний діагональний елемент матриці A .

Множина $\{c_i\}_{i=1}^N$ є шуканою множиною базисних ортогональних векторів, які після нормалізації можуть бути використані для представлення вектора x у вигляді:

$$x = \sum_{i=1}^N a_i c_i. \quad (4.6)$$

Множину власних значень $\{\lambda_i\}_{i=1}^N$ та відповідних власних векторів $\{c_i\}_{i=1}^N$ упорядковують таким чином, щоб

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_i \geq \dots \geq \lambda_N. \quad (4.7)$$

Помноживши (4.6) на c^T , одержимо $a_i = z_i$. Звідси

$$x = \sum_{i=1}^N z_i c_i. \quad (4.8)$$

Ми отримали співвідношення для реконструкції вхідного вектора x по відомому вихідному вектору z за умови їх однакової вимірності та відсутності кореляції між елементами вихідного вектора. Однак за умовою: $m < n$, тобто апроксимуючий вектор повинен мати меншу розмірність, ніж вхідний вектор. З цієї причини необхідно оцінити похибку, що виникає в результаті такого процесу апроксимації.

Представимо загальну похибку у вигляді квадратичної суми похибок від представлення кожного вектора вхідного набору X :

$$E = \sum_{k=1}^N \left\| \mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)} \right\|^2, \quad (4.9)$$

де $\tilde{\mathbf{x}}^{(k)} = \sum_{i=1}^M z_i^{(k)} \mathbf{c}_i$ — реконструйований вхідний вектор.

Визначимо похибку реконструкції за умови нульового математичного сподівання апроксимуючого набору даних:

$$E = \sum_{k=1}^N \left\| \sum_{i=1}^N z_i^{(k)} \mathbf{c}_i - \sum_{i=1}^M z_i^{(k)} \mathbf{c}_i \right\|^2 = \sum_{k=1}^N \left\| \sum_{i=M+1}^N z_i^{(k)} \mathbf{c}_i \right\|^2. \quad (4.10)$$

У зв'язку з нормалізацією та ортогональністю власних векторів вираз (4.10) може бути переписаний у вигляді:

$$E = \sum_{k=1}^N \left(\sum_{i=M+1}^N z_i^{(k)} \right)^2 = N \sum_{i=M+1}^N \lambda_i. \quad (4.11)$$

Виходячи з упорядкування (4.7), робимо висновок, що похибка (4.11) містить мінімальні власні значення, тобто є мінімально можливою.

4.2. Аналіз незалежних компонентів

4.2.1. Технологія сліпого відокремлення стаціонарних сигналів

Активний розвиток технології аналізу незалежних компонентів (Independent Component Analysis — ICA), який спостерігаємо останнім часом, можна пояснити рядом його успішних застосувань до розв'язання задач розпізнавання мови, телекомунікаційних задач та обробки сигналів у медицині. Основною метою даного аналізу є одержання сигналів від недоступних джерел інформації шляхом виділення їх із лінійної

суміші, що постачається датчиками на вхід нейронної мережі. Тому його ще називають сліпим відокремленням стаціонарних сигналів. Аналіз незалежних компонентів містить функції декореляції сигналів, які властиві методу РСІ, в межах застосування статистичних технологій другого порядку. Крім того, методи ІСА забезпечують зменшення статистичних залежностей більш високих порядків, що дає можливість сформувати максимально незалежні сигнали.

Розглянемо вектор взаємно незалежних в кожний фіксований момент часу t сигналів $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_N(t)]^T$ від недоступних джерел інформації. В цьому випадку багатовимірна функція густини імовірності дорівнює добутковій граничній густині імовірностей:

$$p(\mathbf{s}) = \prod_{i=1}^N p_i(s_i). \quad (4.12)$$

Вектор фіксованих датчиками сигналів суміші $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$ визначається моделлю датчиків:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad (4.13)$$

де $\mathbf{A} \in \mathbf{R}^{N \times N}$ — невідома невироджена матриця змішування.

Завдання полягає у відновленні сигналів, що генеруються джерелами інформації з вектора суміші $\mathbf{x}(t)$ без знання матриці суміші за допомогою лінійного перетворення:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t), \quad (4.14)$$

де $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_N(t)]^T$ — відновлений вектор;

$\mathbf{W} \in \mathbf{R}^{N \times N}$ — матриця розшарування.

В ідеальному випадку матриця розшарування \mathbf{W} еквівалентна оберненій матриці змішування \mathbf{A}^{-1} . Крім того, вектор невідомих сигналів джерел $\mathbf{s}(t)$ повинен дорівнювати векторові відновлених сигналів $\mathbf{y}(t)$. Але одержати відновлений вектор, що співпадає би з вектором $\mathbf{s}(t)$, неможливо через

невизначеність перестановок його елементів у ході формування моделі (4.13) та проблеми масштабування, яка виникає в зв'язку з добутком двох невідомих: матриці змішування A та елементів вектора невідомих джерел інформації $s(t)$. Ідентифікації невідомих сигналів джерела можна досягти введенням матриці перестановок та використанням коефіцієнтів масштабування $c_i s_i(t)$.

Таким чином, завдання методу ІСА зводиться до пошуку матриці розшарування, яка б задовольняла співвідношення:

$$W = APA^{-1}, \quad (4.15)$$

де A — невироджена діагональна матриця;

P — матриця перестановок.

Методи пошуку матриці розшарування базуються на наступних припущеннях [59,60]:

1. Кількість датчиків має дорівнювати або перевищувати кількість невідомих сигналів для формування повнорангової матриці A .
2. Невідомі сигнали джерел повинні бути повністю незалежними в кожний конкретний момент часу, що відповідає правилу формування багатовимірної функції густини імовірності (4.12).
3. Не більше ніж один із невідомих сигналів джерел повинен описуватись нормальним законом розподілу, оскільки задача розшарування на два сигнали з нормальними законами розподілу є некоректною.
4. Не дозволяється присутність шумів у суміші сигналів. Оскільки ефективне розшарування можливе лише в умовах низького рівня шумів, допускається тільки низький рівень білого шуму. Виходом із цього становища може бути інтерпретація шуму як одного з невідомих джерел інформації.

Технологія сліпого відокремлення стаціонарних сигналів базується на методах пошуку максимуму ентропії, мінімуму взаємної інформації та максимуму імовірності.

4.2.1.1. Максимум ентропії

У [36] запропоновано метод навчання для ШНМ, що забезпечує розшарування невідомих сигналів джерела s з лінійної суміші x шляхом максимізації ентропії. Максимізуючи загальну ентропію $H(z)$, ШНМ здатна мінімізувати взаємну інформацію своїх вихідних сигналів $z_i = f(y_i)$, де $f(y_i)$ — оборотна нелінійна функція; $y = Wx$.

Взаємна інформація $I(z_1, z_2, \dots, z_N)$ вперше визначена Шенноном як величина, що задає ступінь перекриття між випадковими змінними z_1, z_2, \dots, z_N . Вона завжди додатна або, у випадку повної незалежності змінних z_1, z_2, \dots, z_N , дорівнює нулю.

Виразимо взаємну інформацію через загальну ентропію $H(z_1, z_2, \dots, z_N)$ з використанням правила ланцюга:

$$H(z_1, z_2, \dots, z_N) = H(z_1) + H(z_2) + \dots + H(z_i) + \dots + H(z_N) - I(z_1, z_2, \dots, z_N), \quad (4.16)$$

де $H(z_i)$ — гранична ентропія для виходу i ШНМ.

Максимізація загальної ентропії полягає в максимізації граничних ентропій виходів ШНМ та мінімізації їх взаємної інформації. При $I(z_1, z_2, \dots, z_N) = 0$ загальна ентропія дорівнює сумі граничних ентропій:

$$H(z_1, z_2, \dots, z_N) = H(z_1) + H(z_2) + \dots + H(z_N). \quad (4.17)$$

Максимальне значення загальної ентропії може бути досягнутим за умови, що змінні z_1, z_2, \dots, z_N — амплітудно-обмежені та мають однакові закони розподілу. Саме з цієї причини необхідно використовувати логістичні нелінійні функції $z_i = f(y_i)$. Такий підхід може бути реалізований на одношаровій ШНМ прямого поширення з нелінійними активаційними функціями вихідного шару.

Співвідношення між $p(z)$ та $p(x)$ може бути заданим через детермінант матриці Якобі [151]:

$$p(z) = \frac{p(x)}{|\det J(x)|}. \quad (4.18)$$

З використанням (4.18), вираз для оцінки загальної ентропії в логарифмічному представленні матиме вигляд:

$$H(z) = -E\{\log p(z)\} = E\{\log |\det J(x)|\} - E\{\log p(x)\}. \quad (4.19)$$

Для пошуку екстремуму (4.19) відносно W використаємо метод градієнтного спуску:

$$\frac{\partial W}{\partial t} = \alpha \frac{\partial H(z; W)}{\partial W}. \quad (4.20)$$

Враховуючи (4.19), відмітимо, що пошук екстремуму загальної ентропії зводиться до пошуку максимуму матриці Якобі для вагових коефіцієнтів:

$$\frac{\partial W}{\partial t} = \frac{\partial \log |\det(W)|}{\partial W} + \frac{\partial \log \prod_{i=1}^N \left| \frac{\partial z_i}{\partial y_i} \right|}{\partial W}. \quad (4.21)$$

Спростимо перший доданок виразу (4.21):

$$\frac{\partial \log |\det(W)|}{\partial W} = (W^T)^{-1} = W^{-T}. \quad (4.22)$$

Добуток у другому доданку правої частини (4.21) може бути замінений сумою логарифмів, кожен із доданків якої залежить від конкретного вагового коефіцієнта w_{ij} . Тому введемо заміну:

$$\frac{\partial \log \prod_{i=1}^N \left| \frac{\partial z_i}{\partial y_i} \right|}{\partial W} = \frac{\partial \sum_{i=1}^N \log \left| \frac{\partial z_i}{\partial y_i} \right|}{\partial W} - \varphi(y) x^T, \quad (4.23)$$

де $\varphi(y)$ — градієнтний вектор, запропонований у [20,48].

$$\begin{aligned} \varphi(y) &= -\frac{\partial p(y)}{\partial y} = \left[-\frac{\partial p(y_1)}{\partial y_1}, -\frac{\partial p(y_2)}{\partial y_2}, \dots, -\frac{\partial p(y_N)}{\partial y_N} \right] = \\ &= -\frac{f''(y)}{f'(y)} = \left[-\frac{f''(y_1)}{f'(y_1)}, -\frac{f''(y_2)}{f'(y_2)}, \dots, -\frac{f''(y_N)}{f'(y_N)} \right]. \end{aligned} \quad (4.24)$$

Основний метод навчання (4.20) тепер матиме вигляд:

$$\Delta W = \frac{\partial W}{\partial t} = \alpha \left[(W^T)^{-1} - \varphi(y) x^T \right]. \quad (4.25)$$

Ітераційна формула для пошуку вагових коефіцієнтів за даним методом:

$$W(t+1) = W(t) + \alpha \left[(W^T(t))^{-1} - \varphi(y) x^T \right]. \quad (4.26)$$

Аналогічного результату можна досягти шляхом мінімізації взаємної інформації.

4.2.1.2. Мінімум взаємної інформації

Метод градієнтного спуску для мінімізації взаємної ентропії задають аналогічно до (4.20):

$$\frac{\partial W}{\partial t} = -\alpha \frac{\partial I(W)}{\partial W}. \quad (4.27)$$

Мірою взаємної інформації є дивергенція Куллбека-Лейблера між загальною та граничними функціями густини імовірності:

$$I = KL \left(p, \prod_i p_i \right) = \int p(y) \log \frac{p(y)}{\prod_{i=1}^N p_i(y_i)} dy. \quad (4.28)$$

Провівши ряд елементарних перетворень:

$$\begin{aligned}
I &= \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_{i=1}^N p_i(y_i)} d\mathbf{y} = \\
&= \int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} - \int p(\mathbf{y}) \log \prod_{i=1}^N p_i(y_i) d\mathbf{y} = \\
&= -H(\mathbf{y}) - \sum_{i=1}^N \int p(\mathbf{y}) \log p_i(y_i) d\mathbf{y} = \sum_{i=1}^N H(y_i) - H(\mathbf{y}) = \\
&= \sum_{i=1}^N H(y_i) - H(\mathbf{x}) - \log |\det(\mathbf{W})|, \quad (4.29)
\end{aligned}$$

отримаємо вираз для визначення взаємної інформації, що відповідає визначенню Шеннона (4.15).

Для обчислення $H(y_i)$ використовують ряди Еджворса або Грема–Чарлі з метою апроксимації відповідних граничних функцій густини імовірності $p_i(y_i)$. В [114] запропонована така апроксимація:

$$p_i(y_i) \approx \gamma(y_i) \left[1 + \frac{(k_3)_i}{3!} H_3(y_i) + \frac{(k_4)_i}{4!} H_4(y_i) \right], \quad (4.30)$$

де $(k_3)_i = (m_3)_i$, $(k_4)_i = (m_4)_i - 3$, $(m_k)_i = E[y_i^k]$ — момент функції y_i з порядком k ;

$$\gamma(y_i) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{y_i^2}{2}\right\};$$

$H_k(y_i)$ — поліноми Чебишева–Ерміта, що визначаються з виразу:

$$(-1)^k \frac{\partial^k \gamma(y_i)}{\partial y_i^k} = H_k(y_i) \gamma(y_i). \quad (4.31)$$

Після ряду перетворень [20], кінцева апроксимація для граничної ентропії має вигляд:

$$\begin{aligned}
H(y_i) &= \\
&= \frac{1}{2} \log(2\pi e) - \frac{(k_3)_i^2}{2 \cdot 3!} - \frac{(k_4)_i^2}{2 \cdot 4!} + \frac{5}{8} (k_3)_i^2 (k_4)_i + \frac{1}{16} (k_4)_i^3. \quad (4.32)
\end{aligned}$$

Підставивши (4.32) в (4.29), одержимо кінцеву формулу для визначення взаємної інформації:

$$\begin{aligned}
I &= -H(\mathbf{x}) - \log |\det(\mathbf{W})| + \frac{1}{2} \log(2\pi e) - \\
&\sum_{i=1}^N \left[\frac{(k_3)_i^2}{2 \cdot 3!} - \frac{(k_4)_i^2}{2 \cdot 4!} - \frac{5}{8} (k_3)_i^2 (k_4)_i - \frac{1}{16} (k_4)_i^3 \right]. \quad (4.33)
\end{aligned}$$

Сформуємо градієнт окремо для кожного з вагових коефіцієнтів ШНМ на основі виразу (4.33) для взаємної інформації з урахуванням таких співвідношень:

$$\begin{aligned}
\frac{\partial \log |\det(\mathbf{W})|}{\partial w_{ij}} &= (\mathbf{W}^T)_{ij}^{-1} \text{ — } ij\text{-й елемент матриці } (\mathbf{W}^T)^{-1}, \\
\frac{\partial (k_3)_i}{\partial w_{ij}} &= 3E[(y_i)^2 x_j], \\
\frac{\partial (k_4)_i}{\partial w_{ij}} &= 4E[(y_i)^3 x_j].
\end{aligned}$$

Формула градієнта для методу мінімізації інформації:

$$\begin{aligned}
\frac{\partial I}{\partial w_{ij}} &= -(\mathbf{W}^T)_{ij}^{-1} + \\
&+ \varphi[(k_3)_i, (k_4)_i] \cdot E[(y_i)^2 x_j] + \\
&+ \theta[(k_3)_i, (k_4)_i] \cdot E[(y_i)^3 x_j], \quad (4.34)
\end{aligned}$$

де $\varphi[(k_3)_i, (k_4)_i] = -\frac{1}{2} (k_3)_i + \frac{15}{4} (k_4)_i$;

$$\theta[(k_3)_i, (k_4)_i] = -\frac{1}{6} (k_4)_i + \frac{5}{2} (k_3)_i + \frac{3}{4} (k_4)_i.$$

Модифікацію вагових коефіцієнтів згідно з методом градієнтного спуску задають виразом:

$$\begin{aligned} \frac{\partial w_{ij}}{\partial t} &= -\alpha \frac{\partial \mathbf{I}}{\partial w_{ij}} = \\ &= \alpha \left\{ \left(\mathbf{W}^T \right)_{ij}^{-1} - \varphi \mathbf{E} \left[(y_i)^2 x_j \right] - \theta \mathbf{E} \left[(y_i)^3 x_j \right] \right\}. \end{aligned} \quad (4.35)$$

Для обчислення функцій φ і θ вводять додаткові адаптивні алгоритми, що забезпечують обчислення поточних значень кумулянтів $(k_3)_i$ та $(k_4)_i$:

$$\begin{aligned} \frac{\partial (k_3)_i}{\partial t} &= -\beta \left[(k_3)_i - y_i^3 \right], \\ \frac{\partial (k_4)_i}{\partial t} &= -\beta \left[(k_4)_i - y_i^4 + 3 \right], \end{aligned} \quad (4.36)$$

де β — коефіцієнт навчання.

Замінивши в (4.35) моменти випадкових змінних на їх поточні значення, одержуємо наближену формулу для модифікації вагових коефіцієнтів:

$$\frac{\partial w_{ij}}{\partial t} = \alpha \left[\left(\mathbf{W}^T \right)_{ij}^{-1} - \varphi(y_i) x_j \right], \quad (4.37)$$

де $\varphi(z_i) = \frac{29}{4} y_i^3 + \frac{47}{4} y_i^5 + \frac{14}{3} y_i^7 + \frac{25}{4} y_i^9 + \frac{3}{4} y_i^{11}$.

Переписавши (4.37) в матричній формі, отримуємо метод модифікації матриці вагових коефіцієнтів, аналогічний до того, який був одержаний методом максимізації ентропії (4.25):

$$\Delta \mathbf{W} = \frac{\partial \mathbf{W}}{\partial t} = \alpha \left[\left(\mathbf{W}^T \right)^{-1} - \varphi(\mathbf{y}) \mathbf{x}^T \right]. \quad (4.38)$$

Відповідно задаємо ітераційну формулу для обчислення вагових коефіцієнтів:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha \left[\left(\mathbf{W}^T(t) \right)^{-1} - \varphi(\mathbf{y}) \mathbf{x}^T \right]. \quad (4.39)$$

4.2.1.3. Природний градієнт

Метод природного градієнта було запропоновано в [19] для (4.38) з метою прискорення збіжності та спрощення обчислювального процесу.

Введемо заміну в (4.38), враховуючи, що $\mathbf{x}^T \mathbf{W}^T = \mathbf{y}^T$:

$$\frac{\partial \mathbf{W}}{\partial t} = \alpha \left[\left(\mathbf{W}^T \right)^{-1} - \varphi(\mathbf{y}) \mathbf{x}^T \right] = \alpha \left[\mathbf{I} - \varphi(\mathbf{y}) \mathbf{y}^T \right] \left(\mathbf{W}^T \right)^{-1}. \quad (4.40)$$

Для одержання обчислення приросту вагових коефіцієнтів за методом природного градієнта помножимо праву частину (4.40) на $\mathbf{W}^T \mathbf{W}$:

$$\begin{aligned} \frac{\partial \mathbf{W}}{\partial t} &= \alpha \left[\left(\mathbf{W}^T \right)^{-1} - \varphi(\mathbf{y}) \mathbf{y}^T \right] \left(\mathbf{W}^T \right)^{-1} \mathbf{W}^T \mathbf{W} = \\ &= \alpha \left[\mathbf{I} - \varphi(\mathbf{y}) \mathbf{y}^T \right] \mathbf{W}. \end{aligned} \quad (4.41)$$

Ітераційна формула методу природного градієнта:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha \left[\mathbf{I} - \varphi(\mathbf{y}) \mathbf{y}^T \right] \mathbf{W}. \quad (4.42)$$

Незважаючи на такий тривіальний перехід від методу звичайного градієнтного спуску до методу природного градієнта, він не є штучно створеною евристичною технікою, а відображає істотні закономірності оптимізації, що мають широке застосування на практиці.

Відомим фактом є те, що градієнт $\left(\frac{\partial \mathbf{J}}{\partial y_1}, \frac{\partial \mathbf{J}}{\partial y_2}, \dots, \frac{\partial \mathbf{J}}{\partial y_N} \right)$ визначає в точці \mathbf{y} напрям найкоротшого спуску за умови заданості функції $\mathbf{J}: \mathbf{R}^N \rightarrow \mathbf{R}$.

Якщо функцію задано у вигляді матричних перетворень $J: \mathbf{R}^{N \times N} \rightarrow \mathbf{R}$, то градієнт не визначає напрям найкоротшого спуску, і тому не є оптимальним. Згідно з [19], для досягнення високої збіжності у випадку матричних операцій необхідно застосовувати метод природного градієнта, який забезпечує найкоротший спуск.

Зменшення кількості обчислень для методу природного градієнта пояснюється тим, що у даному випадку немає необхідності інвертувати матрицю W .

4.2.1.4. Оцінка максимуму імовірності

Метод оцінки максимуму імовірності застосовують у випадку, коли маємо деяку інформацію про характер моделі датчиків, які задають лінійним перетворенням (4.13). Він базується на використанні внутрішніх параметрів цієї генеруючої моделі і полягає в моделюванні процесу формування вектора x із використанням вектора невідомих сигналів джерел та матриці A .

Нехай маємо послідовність $X = \{x(1), x(2), \dots, x(T)\}$ векторів з густиною імовірності $p(X)$. Задамо параметричну генеруючу модель P за допомогою параметричних оцінок розподілу $p(x; a)$:

$$P = \{p(x; a) | a \in A\}. \quad (4.43)$$

Тоді імовірність $L_T(a)$ послідовності X , сформованої в термінах моделі:

$$L_T(a) = \frac{1}{T} \log \prod_{t=1}^T p(x(t); a) = \frac{1}{T} \sum_{t=1}^T \log p(x(t); a). \quad (4.44)$$

При зростанні T середнє арифметичне (4.44) збігається за законом великих чисел до відповідного математичного сподівання:

$$L(A) = \int p(X) \log p(X; A) dX \quad (4.45)$$

Виходячи з того, що різницю між $p(X; A)$ та $p(X)$ задають за допомогою дивергенції Куллбека–Лейблера:

$$\begin{aligned} D(p(X), p(X; A)) &= \int p(X) \log \frac{p(X)}{p(X; a)} dX = \\ &= \int p(X) \log p(X) dX - \int p(X) \log p(X; A) dX = \\ &= H(X) - \int p(X) \log p(X; A) dX \end{aligned} \quad (4.46)$$

Підставивши (4.45) в (4.46), одержимо:

$$L(A) = H(X) - D(p(X), p(X; A)). \quad (4.47)$$

Оскільки ентропія $H(X)$ не залежить від W , то пошук максимуму імовірності зводиться до мінімізації дивергенції Куллбека–Лейблера:

$$\frac{\partial L(A)}{\partial t} = -\beta \frac{\partial D(p(X), p(X; A))}{\partial W}. \quad (4.48)$$

Матриця змішування A є оборотною, що впливає з необхідності існування матриці W , еквівалентної A^{-1} . Враховуючи цей факт та інваріантність дивергенції Куллбека–Лейблера до оборотних перетворень, можемо стверджувати, що мінімізація дивергенції (4.48) еквівалентна мінімізації дивергенції між функцією густини імовірності недоступних сигналів джерел та їх параметричною оцінкою на ШНМ:

$$\frac{\partial L(A)}{\partial t} = -\beta \frac{\partial D(p(S), p(Y; W))}{\partial W}. \quad (4.49)$$

Застосувавши метод градієнтного спуску і метод природного градієнта до (4.49), одержимо залежність для приросту вагових коефіцієнтів:

$$\Delta W = \frac{\partial W}{\partial t} = \beta [I - \varphi(y)y^T] W \quad (4.50)$$

та ітераційну формулу:

$$W(t+1) = W(t) + \beta [I - \varphi(y)y^T] W. \quad (4.51)$$

4.2.2. Технологія сліпого відокремлення нестационарних сигналів

Методи пошуку максимальної ентропії, мінімальної взаємної інформації та максимальної імовірності виконують реконструкцію невідомих джерел за допомогою лінійного перетворення (4.14). Така задача є оберненою до тієї, що визначається моделлю датчиків (4.13). На основі накопиченої інформації про залежність між елементами вектора змішування формується матриця розшарування W . Методи пошуку максимальної ентропії, мінімальної взаємної інформації та максимальної імовірності базуються на визначенні граничних законів розподілу, виходячи з інформації про суміш сигналів. Орієнтовані вони на стаціонарні сигнали, оскільки теоретично неможливо побудувати закон розподілу випадкової величини тільки за одним спостереженням, тому що такий закон розподілу постійно змінюється від спостереження до спостереження.

В роботі [130] вперше запропоновано ефективну технологію сліпого відокремлення нестационарних сигналів з використанням статистики другого порядку. Суть її полягає в декореляції нестационарних сигналів.

Задамо модель змішування лінійним перетворенням аналогічно моделі датчиків (4.12):

$$\mathbf{x}(t) = A\mathbf{s}(t), \quad (4.52)$$

де $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T$ — вектор сигналів датчиків;

$\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_n(t)]^T$ — вектор статистично незалежних сигналів від невідомих джерел;

$A \in \mathbf{R}^{m \times n}$ — невідома матриця змішування; $m \geq n$.

Завдання полягає у реконструкції $\mathbf{s}(t)$ з вектора суміші $\mathbf{x}(t)$ без знання матриці змішування A та вектора невідомих сигналів джерел за допомогою лінійного перетворення:

$$\mathbf{y}(t) = W\mathbf{x}(t). \quad (4.53)$$

Методи відокремлення нестационарних сигналів базуються на таких припущеннях [130]:

1. Матриця A повинна бути повноранговою по стовпцях.
2. Сигнали джерела $\{s_i\}$ повинні бути статистично незалежними з нульовим математичним сподіванням, що забезпечує матриці коваріантності $C(t) = E\{\mathbf{s}(t)\mathbf{s}(t)^T\}$

діагональний вигляд:

$$C(t) = \text{diag}\{\sigma_1(t), \sigma_2(t), \dots, \sigma_n(t)\}, \quad (4.54)$$

де $\sigma_i(t) = E\{s_i(t)^2\}$;

E — оператор математичного сподівання.

3. $\frac{\sigma_i(t)}{\sigma_j(t)} \neq \frac{\sigma_i(t+\tau)}{\sigma_j(t+\tau)}$, $i, j = 1, 2, \dots, n$; $i \neq j$; $\tau \neq 0$.

4.2.2.1. Метод поточної декореляції

У [130] показано, що для нестационарних сигналів джерел $\mathbf{s}(t)$ можливо здійснити відокремлення елементів суміші $\mathbf{x}(t)$ з взаємною кореляцією елементів $y_i(t)$ та $y_j(t)$, $i, j = 1, 2, \dots, n$; $i \neq j$, яка дорівнює нулю. Для виключення взаємної кореляції між елементами реконструйованого вектора $\mathbf{y}(t)$ застосовують цільову функцію:

$$\Theta(W) = \frac{1}{2} \left[\sum_{i=1}^n \log E\{y_i^2(t)\} - \log \det \left(E\{\mathbf{y}(t)\mathbf{y}(t)^T\} \right) \right]. \quad (4.55)$$

Функція (4.55) є додатною функцією, значення якої дорівнює нулю тоді і тільки тоді, коли $E\{\mathbf{y}(t)\mathbf{y}(t)^T\} = 0$.

Задачу відокремлення нестационарних сигналів розв'язують за допомогою рекурентної ШНМ (рис.4.1), роботу якої описує система рівнянь:

$$\begin{cases} y_1(t) = x_1(t) + w_{11}y_1(t) + w_{12}y_2(t) + \dots + w_{1n}y_n, \\ y_2(t) = x_2(t) + w_{21}y_1(t) + w_{22}y_2(t) + \dots + w_{2n}y_n, \\ \dots \\ y_n(t) = x_n(t) + w_{n1}y_1(t) + w_{n2}y_2(t) + \dots + w_{nn}y_n. \end{cases} \quad (4.56)$$

У матричному запису система (4.56) має вигляд:

$$\mathbf{y}(t) = \mathbf{x}(t) + \mathbf{W}\mathbf{y}(t) = (\mathbf{I} - \mathbf{W})^{-1} \mathbf{x}(t). \quad (4.57)$$

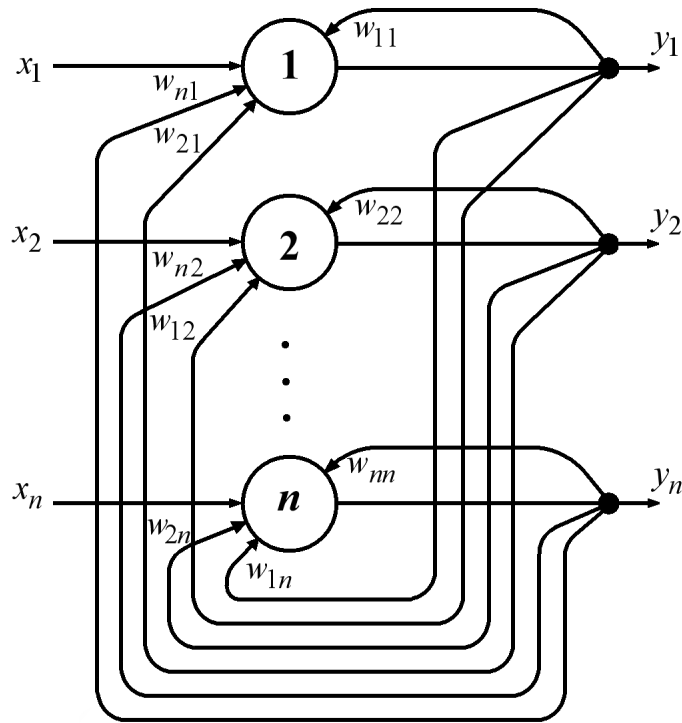


Рис.4.1. Рекурентна ШНМ для сліпого відокремлення нестационарних сигналів.

Навчання даної ШНМ формують на основі методу градієнтного спуску:

$$\frac{\partial \mathbf{W}}{\partial t} = -\alpha \frac{\partial \Theta}{\partial \mathbf{W}}. \quad (4.58)$$

Для цього визначимо диференціал $d\Theta(\mathbf{W})$ по відношенню до $d\mathbf{W}$:

$$\begin{aligned} d\Theta(\mathbf{W}) &= \Theta(\mathbf{W} + d\mathbf{W}) - \Theta(\mathbf{W}) = \\ &= \frac{1}{2} d \left(\sum_{i=1}^n \mathbb{E} \{ y_i^2(t) \} \right) - \frac{1}{2} d \left(\log \det \left(\mathbb{E} \{ \mathbf{y}(t) \mathbf{y}(t)^T \} \right) \right). \end{aligned} \quad (4.59)$$

Введемо заміну $\mathbf{G}(t) = \mathbb{E} \{ \mathbf{x}(t) \mathbf{x}(t)^T \}$. Тоді для другого доданка з (4.59) одержимо:

$$\begin{aligned} d \left(\log \det \left(\mathbb{E} \{ \mathbf{y}(t) \mathbf{y}(t)^T \} \right) \right) &= \\ &= 2 \text{tr} \left\{ (\mathbf{I} - \mathbf{W})^{-1} d\mathbf{W} \right\} + d \left(\log \det (\mathbf{G}(t)) \right). \end{aligned} \quad (4.60)$$

Другий доданок в (4.60) не залежить від \mathbf{W} , тому його можна не враховувати. Позначимо $d\mathbf{V} = (\mathbf{I} - \mathbf{W})^{-1} d\mathbf{W}$ і підставимо в спрощений вираз (4.60):

$$d \left(\log \det \left(\mathbb{E} \{ \mathbf{y}(t) \mathbf{y}(t)^T \} \right) \right) = 2 \text{tr} \{ d\mathbf{V} \}. \quad (4.61)$$

Аналогічно для першого доданка (4.59):

$$d \left(\sum_{i=1}^n \log \mathbb{E} \{ y_i^2(t) \} \right) = 2 \mathbb{E} \{ \mathbf{y}^T(t) \mathbf{D}^{-1}(t) d\mathbf{V} \mathbf{y}(t) \}, \quad (4.62)$$

де $\mathbf{D} = \text{diag} \{ \mathbb{E} \{ y_1^2(t) \}, \mathbb{E} \{ y_2^2(t) \}, \dots, \mathbb{E} \{ y_n^2(t) \} \}$.

Підставивши (4.61) та (4.62) в (4.59), одержимо похідну відносно \mathbf{V} :

$$\frac{d\Theta(\mathbf{W})}{d\mathbf{V}} = \mathbb{E} \{ \mathbf{D}^{-1} \mathbf{y}(t) \mathbf{y}(t)^T \} - \mathbf{I}. \quad (4.63)$$

Виходячи з (4.63), задамо ітераційний процес для \mathbf{V} :

$$\mathbf{V}(t+1) = \mathbf{V}(t) + \alpha \left[\mathbf{I} - \mathbf{D}^{-1}(t) \mathbb{E} \{ \mathbf{y}(t) \mathbf{y}(t)^T \} \right], \quad (4.64)$$

де $\Phi(t) = \text{diag}(\varphi_1(t), \varphi_2(t), \dots, \varphi_n(t))$;

$$\varphi_i(t) = (1 - \delta)\varphi_i(t-1) + \delta y_i(t)^2; \quad \delta = 0,01.$$

Приріст $\Delta V(t) = V(t+1) - V(t) = \alpha [I - \Phi(t)^{-1} y(t)y(t)^T]$. В той же час:

$$\Delta W(t) = [I - W(t)]^{-1} \Delta W(t).$$

Звідси одержимо приріст вагових коефіцієнтів:

$$\Delta W(t) = \alpha (I - W(t)) [I - \Phi(t)^{-1} y(t)y(t)^T] \quad (4.65)$$

та ітераційну формулу:

$$W(t+1) = W(t) + \alpha (I - W(t)) [I - \Phi(t)^{-1} y(t)y(t)^T]. \quad (4.66)$$

4.2.2.2. Метод автокореляції

Метод автокореляції [137] в процесі відокремлення суміші вимагає не тільки відсутності кореляції у фіксовані моменти часу, а також відсутності кореляції між сигналами, які зняті в різні моменти часу. Виходячи з цієї умови, можемо визначити матрицю кореляції сигналів суміші:

$$\begin{aligned} E\{x(t)x(t+\tau)^T\} &= A \cdot E\{s(t)s(t+\tau)\} \cdot A^T = \\ &= A \cdot \begin{pmatrix} R_{s1}(\tau) & 0 & \dots & 0 \\ 0 & R_{s2}(\tau) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & R_{sn}(\tau) \end{pmatrix} \cdot A^T, \end{aligned} \quad (4.67)$$

де $R_{si}(\tau)$ — автокореляційна функція сигналу джерела $s_i(t)$.

Ідея методу полягає в тому, що у випадку правильного вибору матриці W одержимо діагональну матрицю для елементів реконструйованого вектора:

$$E\{y(t)y(t+\tau)\} = E\{(WAs(t))(WAs(t+\tau))^T\}. \quad (4.68)$$

Перепишемо умову (4.15) у вигляді:

$$WA = AP. \quad (4.69)$$

Підставимо (4.69) в (4.68):

$$\begin{aligned} E\{y(t)y(t+\tau)\} &= E\{(APs(t))(APs(t+\tau))^T\} = \\ &= \begin{pmatrix} \lambda^{11} R_{s1'} & 0 & \dots & 0 \\ 0 & \lambda^{21} R_{s2'} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda^{n1} R_{sn'} \end{pmatrix}, \end{aligned} \quad (4.70)$$

де $1', 2', \dots, n'$ — задані матрицею перестановок P перестановки індексів $1, 2, \dots, n$ відносно моделі змішування;

λ^i — i -й діагональний елемент матриці A .

У [22] задачу (4.70) зводять до розв'язання системи рівнянь:

$$W \cdot E\{x(t)x(t+\tau)\} \cdot W^T = D_i, \quad i = 1, 2, \dots, r, \quad (4.71)$$

де D_i — i -та діагональна матриця.

Для розв'язання (4.71) застосовують метод градієнтного спуску, який мінімізує цільову функцію:

$$L_1(W) = \sum_{i \neq j} (E\{y_i(t)y_j(t)\})^2 + \sum_{i \neq j} (E\{y_i(t)y_j(t+\tau)\})^2 \quad (4.72)$$

Мінімум $L_1(W)$ досягається у випадку, коли мінімального значення набуває кожен із доданків, який залежить від взаємної кореляції реконструйованих сигналів, взятих відповідно в один і в різні моменти часу. Для цільової функції (4.72) дуже важливим є вибір параметра зсуву за часом τ , оскільки саме він визначає, чи будуть кореляційні матриці містити максимально різну інформацію.

Для вирішення проблеми вибору τ в [187] запропоновано цільову функцію, яка містить ряд кореляційних матриць:

$$\begin{aligned}
L_2(\mathbf{W}) &= \\
&= \sum_{i \neq j} \left(\mathbb{E} \{ y_i(t) y_j(t) \} \right)^2 + \sum_{k=1}^N \sum_{i \neq j} \left(\mathbb{E} \{ y_i(t) y_j(t + \tau_k) \} \right)^2 = \\
&= \sum_{k=0}^N \sum_{i \neq j} \left(\mathbb{E} \{ y_i(t) y_j(t + \tau_k) \} \right)^2, \quad (4.73)
\end{aligned}$$

одержаних для різних значень зсуву за часом τ_k .

Або в спрощеному вигляді згідно з [137]:

$$L(\mathbf{W}) = \sum_{k=0}^N \sum_{i \neq j} \left| \mathbf{W} \mathbb{E} \{ x_i(t) x_j(t + \tau_k) \} \mathbf{W}^T \right|^2. \quad (4.74)$$

Для мінімізації цієї функції по \mathbf{W} можна застосувати традиційний метод градієнтного спуску. В [187] запропоновано прямий метод, що дозволяє значно спростити обчислення.

Нехай $\mathbf{V} = \mathbb{E} \{ \mathbf{x}(t) \mathbf{x}(t)^T \}$ — матриця коваріантності вектора суміші. Тоді корінь квадратний із цієї інверсної матриці коваріантності: $\sqrt{\mathbf{V}^{-1}} = \sqrt{\mathbf{A}^{-1}} \mathbf{S}^T$, де \mathbf{S} — ортогональна матриця, а \mathbf{A} — діагональна матриця, які задовольняють умову: $\mathbf{V} = \mathbf{S} \mathbf{A} \mathbf{S}^T$. Виконавши перетворення:

$$\mathbf{x}'(t) = \sqrt{\mathbf{V}^{-1}} \mathbf{x}(t), \quad (4.75)$$

одержимо вектор суміші, який характеризується одиничною матрицею коваріантності:

$$\mathbb{E} \{ \mathbf{x}'(t) \mathbf{x}'(t)^T \} = \sqrt{\mathbf{V}^{-1}} \mathbf{V} \sqrt{\mathbf{V}^{-1}}^T = \mathbf{I}. \quad (4.76)$$

У межах даного підходу вектор суміші розглядають як проекцію вектора невідомих сигналів джерел на деяких напрямках. У загальному випадку ці напрямки не є ортогональними. Тому перетворення (4.75) має на меті формування ортогональних проекцій.

Другий крок методу полягає у вилученні недіагональних елементів кореляційної матриці. З цією метою шукають діагональну поворотну матрицю \mathbf{C} , що мінімізує вираз:

$$l(\mathbf{C}) = \sum_{k=0}^N \sum_{i \neq j} \left| \left(\mathbf{C} \mathbf{M}_k \mathbf{C}^T \right)_{ij} \right|^2, \quad (4.77)$$

де $\left(\mathbf{C} \mathbf{M}_k \mathbf{C}^T \right)_{ij}$ — ij -елемент матриці $\left(\mathbf{C} \mathbf{M}_k \mathbf{C}^T \right)$;

$$\mathbf{M}_k = \mathbb{E} \{ \mathbf{x}'(t) \mathbf{x}'(t + \tau_k)^T \}, \quad k = 1, 2, \dots, N.$$

Для розв'язання (4.77) в [49] запропонований метод, суть якого полягає в тому, що глобальну поворотну матрицю \mathbf{C} апроксимують за допомогою послідовності елементарних поворотів $\mathbf{T}_k(\phi_k)$, кожен з яких мінімізує діагональні елементи матриці коваріантності \mathbf{V} шляхом наближеного обчислення кутів повороту ϕ_k .

Значення елементів матриці \mathbf{C} обчислюють із виразу:

$$\mathbf{C} = \prod_{k=1}^N \mathbf{T}_k(\phi_k). \quad (4.78)$$

У результаті одержують наближене значення матриці розшарування:

$$\mathbf{W} = \mathbf{C} \sqrt{\mathbf{V}^{-1}}. \quad (4.79)$$

4.2.3. Конволютивне сліпе відокремлення сигналів

У реальних умовах сигнали джерела проходять через датчики, що характеризуються різним часом затримки. Основний сигнал може змішуватись із сигналами, відбитими від навколишніх предметів (наприклад, якщо таким датчиком є мікрофон). У цьому випадку модель миттєвого змішування, що використовувалась раніше, стає некоректною.

Для обробки таких сигналів застосовують математичний апарат згортки (convolution). Тому процес їх сліпого

відокремлення отримав назву конволютивного процесу. Середовище, в якому існують відбиті сигнали, називають ще мультитраховим середовищем. Задамо модель змішування невідомих сигналів $\mathbf{s}(t) = \{s_i(t)\}_{i=1}^N, t = \dots, -1, 0, 1, \dots$, які надходять із N взаємно незалежних джерел:

$$\begin{aligned} x_i(t) &= \sum_{j=1}^N \sum_{\tau=-\infty}^{\infty} a_{ij}(\tau) s_j(t-\tau) = \\ &= \sum_{j=1}^N \bar{a}_{ij}(z) s_j(t), \quad i = 1, 2, \dots, N, \end{aligned} \quad (4.80)$$

де $x_i(t)$ — i -й елемент вектора суміші;

$a_{ij}(z) = \sum_{\tau=-\infty}^{\infty} a_{ij}(\tau) z^{-\tau}$ — функція змішування сигналу від

джерела j з сигналом i суміші;

$z^{-\tau}$ — оператор затримки: $s_i(t) z^{-\tau} = s_i(t-\tau)$.

Конволютивна модель змішування у векторному вигляді:

$$\mathbf{x}(t) = \mathbf{A}(z) \mathbf{s}(t). \quad (4.81)$$

У межах даної моделі прийняті наступні припущення:

1. Матриця змішування $\mathbf{A}(z)$ не повинна мати полюсів та нулів у одиничному колі $|z|=1$.

2. Сигнали джерела $s_i(t), i = 1, 2, \dots, N$ повинні бути статистично незалежними, з нульовим математичним сподіванням. Виходячи з цього, автокореляційна матриця

$\mathbf{R}_s(t, \tau) = E\{\mathbf{s}(t) \mathbf{s}(t-\tau)^T\}$ має діагональний вигляд:

$$\begin{aligned} \mathbf{R}_s(t, \tau) &= \text{diag}(E\{s_1(t) s_1(t-\tau)\}, \dots, E\{s_N(t) s_N(t-\tau)\}) = \\ &= \text{diag}(r_1(t, \tau), \dots, r_N(t, \tau)). \end{aligned}$$

3. Сигнали джерела $s_i(t), i = 1, 2, \dots, N$, є нестационарними сигналами, тобто для кожного фіксованого τ елементи матриці автокореляції змінюються з часом t незалежно один від одного.

4.2.3.1. Метод конволютивної поточної декореляції

Метод відокремлення конволютивних сигналів [114] орієнтований на застосування одношарової ШНМ прямого поширення з формуванням елементів вихідного вектора:

$$\begin{aligned} y_i(t) &= \sum_{j=1}^N \sum_{\tau=0}^M w_{ij}(\tau) x_j(t-\tau) = \\ &= \sum_{j=1}^N \bar{w}_{ij}(z) x_j(t), \quad i = 1, 2, \dots, N, \end{aligned} \quad (4.82)$$

де $\bar{w}_{ij}(z) = \sum_{\tau=0}^M w_{ij}(\tau) z^{-\tau}$.

Або у векторному вигляді:

$$\mathbf{y}(t) = \mathbf{W}(z) \mathbf{x}(t), \quad (4.83)$$

де $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_N(t)]^T$ — вектор реконструйованих сигналів;

$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$ — вектор сигналів суміші;

$\mathbf{W}(z) = \begin{pmatrix} \bar{w}_{11}(z) & \dots & \bar{w}_{1N}(z) \\ \dots & \dots & \dots \\ \bar{w}_{N1}(z) & \dots & \bar{w}_{NN}(z) \end{pmatrix}$ — матриця розшарування.

Підставимо (4.81) у рівняння (4.83):

$$\mathbf{y}(t) = \mathbf{W}(z) \mathbf{A}(z) \mathbf{s}(t). \quad (4.84)$$

Аналогічно до (4.71), модель відокремлення конволютивних сигналів повинна задовольняти умову:

$$\mathbf{W}(z) \mathbf{A}(z) = \mathbf{A}(z) \mathbf{P}, \quad (4.85)$$

де \mathbf{P} — довільна матриця перестановок;

$A(z) = \text{diag} \left(\sum_{\tau=-\infty}^{\infty} d_1(\tau) z^{-\tau}, \dots, \sum_{\tau=-\infty}^{\infty} d_N(\tau) z^{-\tau} \right)$ — діагональна

матриця.

Отже, задача зводиться до пошуку такої матриці розширення W , яка б задовольняла умову (4.85).

Для випадку миттєвого змішування в [130] запропоновано цільову функцію (4.55), мінімізація якої дозволяє знайти матрицю розширення W . В [114] цю функцію узагальнено для випадку конволютивних сигналів:

$$\begin{aligned} \Theta(t, W(z)) &= \\ &= \frac{1}{2} \left[\sum_{i=1}^N \log E \left\{ y(t-L)^2 \right\} - \log \det \left(E \left\{ y(t-L) y(t-L)^T \right\} \right) \right], \end{aligned} \quad (4.86)$$

де L — параметр затримки, що задає момент часу, для якого відбувається оцінка взаємної кореляції.

Для мінімізації цільової функції (4.86) в [114] застосовують метод градієнтного спуску:

$$\Delta W(\tau) = -\alpha \frac{\partial \Theta(t, W(z))}{\partial W(\tau)} W(z^{-1}) W(z), \quad (4.87)$$

де α — додатний коефіцієнт навчання; $\tau = 0, 1, 2, \dots, M$.

Підставивши значення цільової функції в (4.87), одержимо:

$$\begin{aligned} \Delta W(\tau) &= \alpha z^{-\tau} \left[I - \left(\text{diag} \left(E \left\{ y(t-L) y(t-L)^T \right\} \right) \right)^{-1} \times \right. \\ &\quad \left. \times E \left\{ y(t-L) y(t-L)^T \right\} \right] W(z), \end{aligned} \quad (4.88)$$

де $\tau = 0, 1, 2, \dots, M$.

Для визначення діагональної матриці

$\text{diag} \left(E \left\{ y(t-L) y(t-L)^T \right\} \right)$ застосовують ітераційний процес:

$$\phi_i(t) = \beta \phi_i(t-1) + (1-\beta) y_i(t-L)^2, \quad (4.89)$$

де $i = 1, 2, \dots, N$, $0 < \beta < 1$.

З урахуванням (4.89) та заміни математичного сподівання $E \left\{ y(t-L) y(t-L)^T \right\}$ на миттєве значення $y(t-L) y(t-L)^T$ вираз (4.88) набуде вигляду:

$$\Delta W(\tau) = \alpha z^{-\tau} \left[I - \Phi(t)^{-1} y(t-L) y(t-L)^T \right] W(z), \quad (4.90)$$

де $\tau = 0, 1, 2, \dots, M$; $\Phi(t) = \text{diag}(\phi_1(t), \phi_2(t), \dots, \phi_N(t))$.

Для того, щоб позбавитися невизначеності, пов'язаної з масштабуванням параметрів $d_i(\tau)$ в діагональній матриці $A(z)$, застосовують різні підходи, одним із яких є нормалізація $\frac{\bar{w}_{ij}(z)}{w_{ij}(L)}$, $i, j = 1, 2, \dots, N$, елементів матриці $W(z)$.

4.2.3.2. Метод конволютивної автокореляції

Для даного методу [153] розглядають модифіковану модель конволютивного змішування:

$$\mathbf{x}(t) = A(z) \mathbf{s}(t) + \mathbf{n}(t), \quad (4.91)$$

де $\mathbf{n}(t)$ — додатковий шум від датчиків.

Визначимо матрицю взаємної кореляції елементів вектора змішування:

$$\begin{aligned} R_x(t, t+\tau) &= E \left\{ \mathbf{x}(t) \mathbf{x}(t+\tau)^T \right\} = \\ &= A E \left\{ \mathbf{s}(t) \mathbf{s}(t)^T \right\} A^T + E \left\{ \mathbf{n}(t) \mathbf{n}(t)^T \right\} = \\ &= A A_s A^T + A_n, \end{aligned} \quad (4.92)$$

де A_s, A_n — діагональні матриці коваріантності сигналів від джерел та шуму від датчиків.

У випадку стаціонарних сигналів параметри взаємної кореляції не залежать від абсолютного значення моменту часу, а визначаються періодом зсуву τ . Тому $R_x(t, t+\tau) = R_x(\tau)$.

Запишемо z -перетворення матриці $R_x(\tau)$:

$$\mathbf{R}_z(z) = \mathbf{A}(z)\mathbf{A}_s(z)\mathbf{A}^H(z) + \mathbf{A}_n(z), \quad (4.93)$$

де $\mathbf{A}(z)$ — матриця z -перетворень фільтрів зі скінченними імпульсними характеристиками (СІХ-фільтрів) матриці $\mathbf{A}(\tau)$; $\mathbf{A}_s(z), \mathbf{A}_n(z)$ — z -перетворення автокореляції сигналів джерел та шуму; $\mathbf{A}^H(z)$ — транспонована та комплексно спряжена матриця $\mathbf{A}(z)$.

На практиці необхідно обмежити кількість z -точок для розв'язання задачі. З цією метою беруть T -рівновіддалені вибірки на одиничному колі, що дає можливість застосувати дискретне перетворення Фур'є (ДПФ). Для періодичних сигналів ДПФ дозволяє виконати кругову згортку (4.93).

Оскільки постановка задачі відокремлення конволютивної суміші базується на лінійній згортці, наближено представимо модель змішування в частотній області за умови, що кількість частотних каналів набагато менша, ніж довжина реалізації:

$$\mathbf{x}(t, \omega) \approx \mathbf{A}(\omega)\mathbf{s}(t, \omega) + \mathbf{n}(t, \omega), \quad (4.94)$$

де $\mathbf{x}(t, \omega)$ — ДПФ реалізації $[\mathbf{x}(t), \mathbf{x}(t+1), \dots, \mathbf{x}(t+T)]$ довжиною T , що визначається за формулою:

$$\mathbf{x}(t, \omega) = \sum_{\tau=0}^{T-1} \exp\{-i2\pi\omega\tau\} \mathbf{x}(t+\tau).$$

Наближене значення взаємної кореляційної в частотній області:

$$\begin{aligned} \tilde{\mathbf{R}}_x(t, \omega) &= \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}(t+nT, \omega)\mathbf{x}(t+nT, \omega)^H \approx \\ &\approx \mathbf{A}(\omega)\mathbf{A}_s(t, \omega)\mathbf{A}(\omega)^H + \mathbf{A}_n(t, \omega). \end{aligned} \quad (4.95)$$

При достатньо великій кількості використаних реалізацій N матриця $\mathbf{A}_s(t, \omega)$ може розглядатися як діагональна, що зумовлено незалежністю сигналів джерел.

Існування моделі змішування не є гарантією існування моделі відокремлення. Тому припустимо, що конволютивне змішування

для нашого випадку допускає існування моделі (4.83). Виходячи зі статистичної незалежності реконструйованих сигналів $\tilde{\mathbf{s}}(t, \omega)$, їх спектральна матриця взаємної кореляції в ідеальному випадку повинна бути діагональною:

$$\mathbf{A}_s(t, \omega) = \mathbf{W}(\omega)[\tilde{\mathbf{R}}(t, \omega) - \mathbf{A}_n(t, \omega)]\mathbf{W}(\omega)^H. \quad (4.96)$$

Використовуючи цю умову, задамо цільову функцію оцінки точності представлення \mathbf{W} для кожного фіксованого $t_k = kNT$:

$$\mathbf{E}(t_k, \omega) = \mathbf{W}(\omega)[\tilde{\mathbf{R}}(t_k, \omega) - \mathbf{A}_n(t, \omega)]\mathbf{W}(\omega)^H - \mathbf{A}_s(t_k, \omega). \quad (4.97)$$

Тоді задача зводиться до пошуку значень матриць $\mathbf{W}, \mathbf{A}_s, \mathbf{A}_n$, які б задовольняли умовам:

$$\begin{aligned} \tilde{\mathbf{W}}, \tilde{\mathbf{A}}_s, \tilde{\mathbf{A}}_n = \arg \min_{\substack{\mathbf{W}, \mathbf{A}_s, \mathbf{A}_n \\ \mathbf{W}(\tau)=0, \tau < Q \ll T, \\ \mathbf{W}_{ii}(\omega)=1}} \sum_{\omega=1}^T \sum_{k=1}^K \|\mathbf{E}(t_k, \omega)\|^2. \end{aligned} \quad (4.98)$$

Додаткова часова область Q показує, що для досягнення заданої точності необхідно вибирати або короткі фільтри, або достатньо тривалі за часом реалізації. Протиріччя (4.98) полягає в тому, що, з одного боку, для досягнення статистичної незалежності необхідно використовувати частоти $\omega = 1, 2, \dots, T$, а, з іншого боку, формування $\mathbf{W}(\omega)$ обмежене частотними фільтрами, що дають нульовий відгук при $\tau < Q \ll T$. Цю проблему можна вирішити шляхом настройки параметрів фільтрів з метою досягнення максимальної точності в діапазоні Q .

Мінімізація (4.98) може бути проведена шляхом застосування методу градієнтного спуску. Згідно з [45] для дійсної функції комплексної змінної градієнти дійсної та уявної частини отримують шляхом взяття частинних похідних від спряженого комплексного числа:

$$\frac{\partial f(z)}{\partial \Re} + i \frac{\partial f(z)}{\partial \Im} = 2 \frac{\partial f(z)}{\partial z^*}, \quad (4.99)$$

де z^* — спряжена комплексна змінна.

З урахуванням (4.99) градієнти цільової функції

$$J = \sum_{k=1}^K \|E(t_k, \omega)\|^2 \text{ по кожній зі змінних:}$$

$$\frac{\partial J}{\partial W^*(\omega)} = 2 \sum_{k=1}^K E(t_k, \omega) W(\omega) [\tilde{R}(t_k, \omega) - A_n(t_k, \omega)], \quad (4.100)$$

$$\frac{\partial J}{\partial A_s^*(t_k, \omega)} = -\text{diag}(E(t_k, \omega)), \quad (4.101)$$

$$\frac{\partial J}{\partial A_n^*(t_k, \omega)} = -\text{diag}(W(\omega) E(t_k, \omega) W(\omega)^H). \quad (4.102)$$

За методом градієнтного спуску знаходять мінімуми $W(\omega)$ та $A(t_k, \omega)$ шляхом застосування відповідних градієнтів (4.100) та (4.102). Оптимальне значення $A_s(t_k, \omega)$ при даних $W(\omega)$ та $A(t_k, \omega)$ на кожному градієнтному кроці може бути обчислене шляхом прирівнювання градієнта (4.101) до нуля, що забезпечує

$$A_s(t_k, \omega) = \text{diag}[W(\omega) \tilde{R}(t_k, \omega) W(\omega)^H - A_n(t_k, \omega)]. \quad (4.103)$$

Недоліком даного методу є висока чутливість до перестановок. Відокремлення може бути виконаним із достатньою точністю тільки у випадку стійких перестановок. При довільних перестановках застосовують додаткові заходи, головним із яких є можливість вибору достатньо великої реалізації для кожного діапазону Q .

4.3. Правило Хебба

Історично так склалося, що одним із перших набув широкого поширення принцип синаптичного навчання Хебба [97], суть якого полягає в тому, що ефективність зв'язку між нейронами зростає пропорційно до кореляції між пре- та постсинаптичною активністю. Або, за словами самого Хебба: «Коли аксон клітини

A знаходиться безпосередньо перед збудженням клітини B та неодноразово або постійно приймає участь у її збудженні, то в одній або обох клітинах виникає деякий наростаючий процес або зміни метаболізму, які приводять до підвищення ефективності збудження B клітиною A » [97]. На основі правила навчання Хебба запропоновано значну кількість моделей роботи мозку та методів його навчання [22,75].

Розглянемо принцип синаптичного навчання Хебба на прикладі роботи ШНМ прямого поширення, яка показана на рис.4.2.

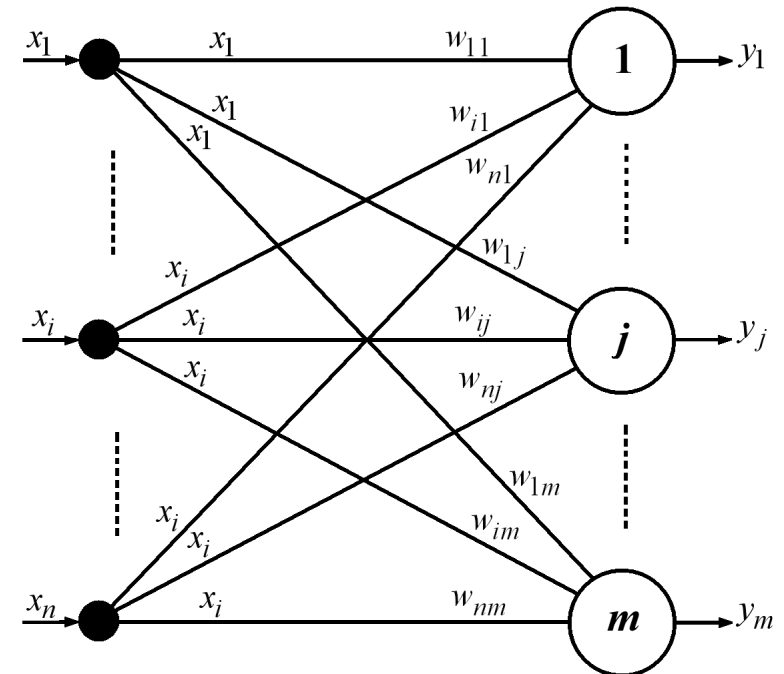


Рис.4.2. Одношарова ШНМ прямого поширення.

Позначимо через $\mathbf{w}_j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})$ вагові коефіцієнти нейрона j , а ефективність зв'язку між x_i та y_j задамо ваговим коефіцієнтом w_{ij} . Тоді зміна цієї ефективності за Хеббом буде пропорційною

$$\Delta w_{ij} = \alpha x_i y_j, \quad (4.104)$$

де $\alpha > 0$ — коефіцієнт навчання.

Залежність (4.104) задає механізм навчання, що полягає у модифікації вагових коефіцієнтів w_{ij} на величину Δw_{ij} . У випадку лінійної активаційної функції нейронів вихідного шару:

$$y_j = \sum_{i=1}^n w_{ij} x_i, \quad (4.105)$$

$$\Delta w_{ij} = \alpha x_i \sum_{k=1}^n w_{kj} x_k. \quad (4.106)$$

Можна також використовувати нелінійні активаційні функції:

$$y_j = f\left(\sum_{i=1}^n w_{ij} x_i\right), \quad (4.107)$$

$$\Delta w_{ij} = \alpha x_i f\left(\sum_{k=1}^n w_{kj} x_k\right). \quad (4.108)$$

Запишемо правило Хебба (4.104) у загальному вигляді:

$$\Delta \mathbf{W} = \alpha \mathbf{x} \mathbf{y}. \quad (4.109)$$

Для застосування його на практиці необхідно визначити момент зупинки, тобто задати умови, при яких $\Delta \mathbf{W} = 0$. Саме це є основною проблемою при навчанні за правилом Хебба. Вагові коефіцієнти можуть зростати необмежено за рахунок додатного зворотного зв'язку.

Розглянемо коротко математичну оцінку стабільності навчання за правилом Хебба. Нехай кореляційна матриця \mathbf{R} має власний вектор $\mathbf{W} = (w_1, w_2, \dots, w_m)$ та власне значення λ :

$$\mathbf{R}\mathbf{W} = \lambda \mathbf{W}. \quad (4.110)$$

Тоді множення вектора \mathbf{W} на матрицю \mathbf{R} є рівнозначним множенню його на скаляр λ . Отже, у даному випадку змінюється лише модуль вектора, а не його напрямок. Припустимо, що процес навчання Хебба збігається до стабільного напрямку \mathbf{W}' . Тоді градієнт в околі цього напрямку повинен прямувати до нуля:

$$0 = \mathbf{E}(\Delta \mathbf{w}'_i) = \mathbf{E}(\mathbf{y} x_i) = \mathbf{E}\left(\sum_{k=1}^n \mathbf{w}_k x_k x_i\right) = \sum_{k=1}^n \mathbf{R}_{ik} \mathbf{w}_k. \quad (4.111)$$

У випадку, коли (4.111) виконується для всіх i :

$$\mathbf{R}\mathbf{W} = 0, \quad (4.112)$$

де \mathbf{R} — симетрична, додатна, квазівизначена матриця з невід'ємними власними значеннями.

Враховуючи (4.112), приходимо до висновку, що вектор \mathbf{W} повинен мати власне значення $\lambda = 0$. Розглянемо додатний окіл ε вектора \mathbf{W}' з ненульовим власним значенням:

$$\mathbf{E}(\Delta \mathbf{W}') = \mathbf{R}(\mathbf{W}' + \varepsilon) = \mathbf{R}\varepsilon > 0. \quad (4.113)$$

З (4.113) випливає, що вагові коефіцієнти зростають у довільному напрямку з ненульовим власним значенням. Отже, існує точка, для якої $\mathbf{W} = 0$, але ця точка є нестабільною, оскільки довільність зміщення призводить до ситуації, коли всі вагові коефіцієнти стають відмінними від нуля. Розглядаючи у цьому контексті процес навчання за правилом Хебба, можна зробити висновок, що таке навчання викликає зміну вагових коефіцієнтів у напрямку зростання власних значень.

4.3.1. Методи обмеження для навчання за правилом Хебба

Існує ряд методів запобігання необмеженому зростанню вагових коефіцієнтів при навчанні за правилом Хебба. Найпростіший із них полягає в обмеженні зростання тих вагових коефіцієнтів, що вже досягли деякої наперед встановленої межі w_j^+ . Основним недоліком даного підходу є той факт, що це

обмеження викликає зміну напрямку вектора W і, таким чином, знижує адекватність моделі. Зі збільшенням кількості операцій обмеження починається деградація моделі. Тому залишається не визначеним момент зупинки алгоритму, який задовольняв би умову максимальної адекватності.

Відомий також підхід, який полягає в нормалізації вагових коефіцієнтів після кожного етапу модифікації. Така операція дає можливість зберегти напрямок вектора W за умови зміни його модуля в межах від нуля до одиниці.

Ітераційний процес навчання за правилом Хебба для вагового коефіцієнта w_{ij} задають формулою:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t). \quad (4.114)$$

Згідно з цим методом, кожна ітерація має супроводжуватися нормалізацією:

$$w_{ij}(t+1) = \frac{w_{ij}(t+1)}{\|w_{ij}(t+1)\|}, \quad (4.115)$$

а у випадку суміщення операції нормалізації з ітераційним процесом:

$$w_{ij}(t+1) = \frac{w_{ij}(t) + \Delta w_{ij}(t)}{\|w_{ij}(t) + \Delta w_{ij}(t)\|}. \quad (4.116)$$

До зменшення темпів зростання також приводить застосування диференційного правила Хебба, яке базується на модифікації (4.114):

$$w_{ij}(t+1) = w_{ij}(t) + \alpha [(x_i(t+1) - x_i(t))(y_j(t+1) - y_j(t))] \quad (4.117)$$

Згідно з (4.117), більшого зростання набудуть вагові коефіцієнти, які відповідають зв'язкам нейронів із більшою динамікою зростання. У випадку одношарової ШНМ розглядають динаміку зростання відповідного вхідного та вихідного сигналу нейрона прихованого шару.

Оскільки найважливішою властивістю правила навчання Хебба є його локальність, то дуже важливо було знайти внутрішній критерій, що дозволив би обмежити зростання вагових коефіцієнтів. Такий критерій був запропонований Гроссбергом [89]. Суть його полягає у фіксації загальної суми вагових коефіцієнтів на вході нейрона. За таких умов величина кожного конкретного вагового коефіцієнта уже не буде зростати необмежено. Формально згаданий локальний критерій виражають залежністю:

$$\frac{\partial w_{ij}}{\partial t} = \alpha x_i y_j - w_{ij}. \quad (4.118)$$

Очевидно, що зупинка зростання вагових коефіцієнтів відповідає умові $\frac{\partial w_{ij}}{\partial t} = 0$.

Тоді точка стабілізації $w_{ij} = \alpha E(x_i y_j)$.

Використовуючи даний підхід до навчання ШНМ за правилом Хебба, одержимо у загальному вигляді:

$$\frac{\partial W}{\partial t} = \alpha RW - W. \quad (4.119)$$

Тоді умова збіжності має вигляд:

$$\frac{W}{\alpha} = RW. \quad (4.120)$$

Отже, для досягнення стабільного стану вектор W сам повинен бути власним вектором кореляційної матриці вхідних даних R з власним значенням $\frac{1}{\alpha}$. Деякі ускладнення даного критерію пізніше були запропоновані Гроссбергом для вхідної зірки:

$$\frac{\partial w_{ij}}{\partial t} = \alpha (y_j - w_{ij}) x_i, \quad (4.121)$$

де величина обмеження зростання вагового коефіцієнта w_{ij} масштабується величиною відповідного вхідного сигналу x_i .

Аналогічно для вихідної зірки:

$$\frac{\partial w_{ij}}{\partial t} = \alpha(x_i - w_{ij})y_j, \quad (4.122)$$

де зростання w_{ij} масштабується величиною вихідного сигналу y_j .

4.3.2. Мультиплікативні та субтрактивні обмеження

У [134] проведені узагальнення відомих підходів до формування критеріїв обмежень зростання вагових коефіцієнтів при використанні навчання ШНМ за правилом Хебба. У результаті цих досліджень були запропоновані мультиплікативні та субтрактивні обмеження, що базуються на критерії Гроссберга про фіксовану суму вагових коефіцієнтів для кожного нейрона. Мультиплікативне обмеження дає таку динаміку зміни вагових коефіцієнтів, в результаті якої виконується (4.120), а субтрактивне обмеження приводить до кінцевого стану, у якому майже всі вагові коефіцієнти набувають максимально або мінімально допустимого значення.

Згідно з простим правилом навчання Хебба для одного нейрона $\frac{\partial \mathbf{W}}{\partial t} = \mathbf{R} \cdot \mathbf{W}$, де \mathbf{R} — матриця коваріантності, що

характеризується множиною власних ортонормальних векторів \mathbf{e}^a , які задовольняють співвідношення: $\mathbf{R} \cdot \mathbf{e}^a = \lambda^a \mathbf{e}^a$, де λ^a — відповідні власні значення. Позначивши через w_a ваговий коефіцієнт власного вектора \mathbf{e}^a , представимо вектор вагових коефіцієнтів нейрона $\mathbf{W} = \sum_{a=0}^n w_a \mathbf{e}^a$.

Розглянемо рівняння мультиплікативного обмеження:

$$\mathbf{W}(t + \Delta t) = \beta(\mathbf{W})[\mathbf{W}(t) + \mathbf{R}\mathbf{W}(t)\Delta t], \quad (4.123)$$

де $\beta(\mathbf{W})$ — обмежувальна функція.

Після проведення елементарних алгебраїчних перетворень одержимо:

$$\frac{\mathbf{W}(t + \Delta t) - \mathbf{W}(t)}{\Delta t} = \mathbf{R}\mathbf{W}(t) - \gamma(\mathbf{W})\mathbf{W}(t + \Delta t), \quad (4.124)$$

де $\gamma(\mathbf{W}) = \frac{1 - \beta(\mathbf{W})}{\beta(\mathbf{W})\Delta t}$.

При $\Delta t \rightarrow 0$ визначаємо мультиплікативне обмеження:

$$\frac{\partial \mathbf{W}(t)}{\partial t} = \mathbf{R}\mathbf{W}(t) - \gamma(\mathbf{W})\mathbf{W}(t). \quad (4.125)$$

Аналогічно задають субтрактивне обмеження:

$$\frac{\partial \mathbf{W}(t)}{\partial t} = \mathbf{R}\mathbf{W}(t) - \varepsilon(\mathbf{W})\mathbf{n}, \quad (4.126)$$

в якому обмеження вагових коефіцієнтів є керованим за допомогою добутку функції $\varepsilon(\mathbf{W})$ та векторної константи \mathbf{n} . У більшості випадків $\mathbf{n} = (1, 1, \dots, 1)^T$.

Метою методів мультиплікативного та субтрактивного обмеження є формування проєкції вектора вагових коефіцієнтів на обмежувальну гіперплощину. У залежності від вибору критерію обмеження розрізняють два типи таких методів.

Критерієм методів першого типу є збереження постійної величини загальної суми вагових коефіцієнтів:

$$\sum_j w_j n_j = \mathbf{W} \cdot \mathbf{n}, \quad (4.127)$$

де $\mathbf{n} = (1, 1, \dots, 1)^T$ — константний вектор.

Обмеження (4.127) може бути досягнутим у випадку виконання умови:

$$\frac{\partial \mathbf{W}(t)}{\partial t} = \frac{\partial (\mathbf{W} \cdot \mathbf{n})}{\partial t} = 0. \quad (4.128)$$

Виходячи з (4.128), задають значення обмежувальних функцій мультиплікативного методу:

$$\gamma(W) = \frac{n \cdot RW(t)}{n \cdot W(t)}, \quad (4.129)$$

при $W(t)|_{t=0} \neq 0$.

Аналогічно задають обмежувальну функцію субтрактивного методу:

$$\varepsilon(W) = \frac{n \cdot RW(t)}{n \cdot n}. \quad (4.130)$$

Критерій методів другого типу полягає у збереженні постійної величини середньоквадратичної суми вагових коефіцієнтів:

$$\sum_j w_j^2 = W \cdot W. \quad (4.131)$$

Даний критерій обмеження не має біологічного аналога і використовується в основному для теоретичних досліджень [186]. Обмеження (4.131) виконується за умови:

$$2W \frac{\partial W(t)}{\partial t} = \frac{\partial (W \cdot W)}{\partial t} = 0. \quad (4.132)$$

Мультиплікативна обмежувальна функція $\gamma(W)$ для методу другого типу, що відповідає умові (4.132):

$$\gamma(W) = \frac{W(t) \cdot RW(t)}{W(t) \cdot W(t)}. \quad (4.133)$$

Випадок субтрактивної обмежувальної функції другого типу не розглядаємо, оскільки у цьому простий метод навчання за правилом Хебба не збігається.

Динаміку вектора вагових коефіцієнтів у загальному випадку визначають за методом градієнтного спуску:

$$\frac{\partial W(t)}{\partial t} = P \cdot R \cdot W(t), \quad (4.134)$$

де P — проектуючий оператор, який забезпечує проектування необмежених динамічних змін вектора вагових коефіцієнтів на обмежувальну гіперплощину.

Форма цього оператора залежить від виду обмеження та типу методу:

$$P = I - \frac{W(t) \cdot n^T}{W(t) \cdot n} \quad \text{— мультиплікативний оператор першого}$$

типу;

$$P = I - \frac{n \cdot n^T}{n \cdot n} \quad \text{— субтрактивний оператор першого типу};$$

$$P = I - \frac{W(t) \cdot W(t)^T}{W(t) \cdot W(t)} \quad \text{— мультиплікативний оператор другого}$$

типу.

Оператор P виконує аналогічні функції для всіх згаданих типів методів, тому для пояснення його дії представимо попередні формули у загальному вигляді:

$$P = I - \frac{s \cdot r^T}{s \cdot r}, \quad (4.135)$$

де s — вектор віднімання; r — вектор обмеження; I — одинична матриця.

Як показано на рис.4.3, дія оператора P на необмежений вектор похідної RW розділяється на дві складові.

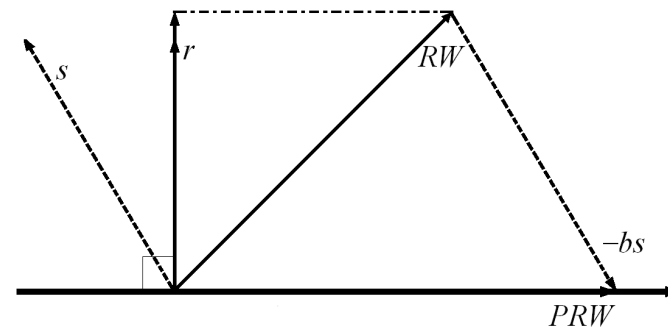


Рис.4.3. Проектування на обмежувальну гіперплощину.

Він видаляє з вектора \mathbf{RW} колінеарну до \mathbf{r} складову шляхом проектування його на обмежувальну гіперплощину:

$$\mathbf{r} \cdot \mathbf{P} \cdot \mathbf{R} \cdot \mathbf{W} = \mathbf{r} \cdot \frac{\partial \mathbf{W}}{\partial t} = 0 \quad (4.136)$$

та обмежує величину даного вектора шляхом віднімання вектора, що є колінеарним до \mathbf{s} :

$$\mathbf{P} \cdot \mathbf{R} \cdot \mathbf{W} = \mathbf{R} \cdot \mathbf{W} - \mathbf{b}\mathbf{s}, \quad (4.137)$$

де $\mathbf{b} = \frac{\mathbf{r} \cdot \mathbf{R} \cdot \mathbf{W}}{\mathbf{r} \cdot \mathbf{s}}$ – коефіцієнт пропорційності для вектора \mathbf{s} .

Для мультиплікативних обмежень $\mathbf{s} = \mathbf{W}$, а для субтрактивних обмежень $\mathbf{s} = \mathbf{r} = \mathbf{n}$.

Застосування мультиплікативних та субтрактивних обмежень у більшості випадків дозволяє одержати принципово різні результати. Так, для мультиплікативного методу першого типу характерною є збіжність до стабільної точки з ненульовим значенням критерію (4.127), а для субтрактивного методу – збіжність до нульового значення даного критерію.

Якщо основний власний вектор \mathbf{e}^1 матриці \mathbf{R} характеризується ненульовою сумою за умови тільки додатних кореляцій [134], то:

- мультиплікативний метод першого типу формує стабільну

$$\text{лінійну комбінацію } \mathbf{W} = \sum_{a=1}^n w_a \mathbf{e}^a;$$

- субтрактивний метод забезпечує створення вектора з нульовою сумою, елементи якого зростають до граничних значень.

У випадку змін знаку кореляції між вхідними сигналами як мультиплікативний, так і субтрактивний методи першого типу спричиняють граничне зростання вагових коефіцієнтів.

Мультиплікативний метод другого типу завжди збігається до стабільної точки, що відповідає основному власному вектору \mathbf{e}^1 .

4.4. Правило Ойя

4.4.1. Правило Ойя для одного нейрона

Правило Ойя для одного нейрона є однією з модифікацій правила Хебба. Його можна визначити як мультиплікативне обмеження другого типу ($y(\mathbf{W}) = y^2$), для якого характерною є збіжність до основного власного вектора вхідної матриці коваріантності \mathbf{R} . Критерій обмеження $\sum_i w_i^2 \rightarrow 1$, тобто вагові коефіцієнти при навчанні за правилом Ойя повинні бути нормалізованими.

Розглянемо однеїронну модель (рис.4.4), що реалізує лінійну активаційну функцію:

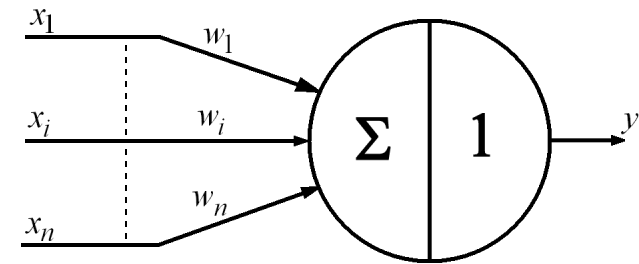


Рис.4.4. Однеїронна модель Ойя.

$$\mathbf{y} = \sum_{i=1}^n w_i x_i = \mathbf{W}\mathbf{x} = \mathbf{x}^T \mathbf{W}^T. \quad (4.138)$$

Розглянемо простий метод навчання за правилом Хебба з нормалізацією (4.116). Очевидно, що для його реалізації необхідні значні затрати обчислювальних ресурсів. Ідея правила Ойя [146] полягає у спрощенні обчислень шляхом апроксимації ітераційної формули за допомогою ряду Тейлора:

$$\begin{aligned} \frac{W + \alpha yx^T}{\|W + \alpha yx^T\|} &\approx \frac{W + \alpha yx^T}{\sqrt{(W + \alpha yx^T)(W + \alpha yx^T)^T}} \approx \\ &\approx \frac{W + \alpha yx^T}{\sqrt{WW^T + 2\alpha yWx + \alpha^2 y^2 x^T x}}. \end{aligned}$$

Беручи до уваги нормалізованість початкових значень вагових коефіцієнтів, тобто, $WW^T = \|W\| = 1$, та достатньо мале значення коефіцієнта навчання $\alpha \ll 1$, для того, щоб знехтувати членом α^2 , запишемо:

$$\begin{aligned} \frac{W + \alpha yx^T}{\|W + \alpha yx^T\|} &\approx \frac{W + \alpha yx^T}{\sqrt{1 + 2\alpha yWx}} \approx (W + \alpha yx^T)(1 - \alpha yWx) \approx \\ &\approx W + \alpha yx^T - \alpha y^2 W - \alpha^2 y^2 x^T \approx W + \alpha y(x^T - yW). \end{aligned} \quad (4.139)$$

Звідси одержуємо ітераційну формулу для навчання за правилом Ойя:

$$W(t+1) = W(t) + \alpha y(x^T - yW(t)), \quad (4.140)$$

з модифікацією вагових коефіцієнтів:

$$\Delta W = \alpha y(x^T - yW(t)). \quad (4.141)$$

Член зі знаком мінус у (4.141) забезпечує стабільність процесу навчання. Щоб показати це, спроєкуємо модифікуючий вектор ΔW на поточний вектор вагових коефіцієнтів W :

$$\Delta W \cdot W^T = x^T \cdot W^T - y \cdot W \cdot W^T = y(1 - \|W\|^2). \quad (4.142)$$

Із (4.142) видно, що модифікуючий вектор ΔW забезпечує наближення поточного вектора до одиничного кола.

Покажемо також, що навчання за правилом Ойя (4.140) забезпечує збіжність вектора вагових коефіцієнтів до основного власного вектора e^1 матриці коваріантності R . Підставивши (4.138) у (4.141), отримаємо:

$$\Delta W = \alpha(Wxx^T - Wxx^T W^T W). \quad (4.143)$$

Математичне сподівання модифікуючого вектора одержимо шляхом визначення математичних сподівань (4.143):

$$E\{\Delta W\} = \alpha(W \cdot E\{xx^T\} - W \cdot E\{xx^T\} \cdot W^T W). \quad (4.144)$$

Припустимо, що поточний вектор W досяг свого сталого значення. Тоді математичне сподівання модифікуючого вектора $E\{\Delta W\} = 0$, а математичне сподівання $R = E\{xx^T\}$ визначає вхідну матрицю коваріантності R . З урахуванням сказаного, (4.144) набуде вигляду:

$$W \cdot R = (W \cdot R \cdot W^T) \cdot W. \quad (4.145)$$

Вектор вагових коефіцієнтів буде відповідати власному вектору матриці коваріантності e^1 за умови:

$$\lambda^1 = W \cdot R \cdot W^T, \quad (4.146)$$

де λ^1 — відповідне власне значення матриці коваріантності R .

Підставивши (4.146) у (4.145), одержимо:

$$W \cdot R = \lambda^1 \cdot W. \quad (4.147)$$

Власне значення λ^1 та власний вектор W є характеристичними величинами матриці R . Дія даної матриці на вектор W рівнозначна до множення його на скаляр λ^1 , тобто, в результаті цієї операції змінюється лише модуль вектора, а напрямок його залишається незмінним.

4.4.2. Правило Ойя для підпростору

Попередньо розглянутий метод навчання Ойя дозволяє знайти тільки один власний вектор, що задовольняє умовам навчання ШНМ. Більш складний підхід, що базується на технології аналізу важливих компонентів, представлений у роботах

[147,110]. Цей підхід полягає у застосуванні нового критерію, що отримав назву підпросторового.

Нехай $\mathbf{x} = (x_1, x_2, \dots, x_N)$ — вхідний вектор з нульовим математичним сподіванням і $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^T\}$ — матриця коваріантності розмірності $N \times N$. Тоді підпросторовий критерій задають виразами:

$$\begin{aligned} J(\mathbf{W}) &= E\left\{\sum_{k=1}^M y^2\right\} = E\left\{\sum_{k=1}^M (\mathbf{w}_k^T \mathbf{x})^2\right\} = \\ &= \max\left(\sum_{k=1}^M \mathbf{w}_k^T \mathbf{R} \mathbf{w}_k\right) = \text{trace}(\mathbf{W}^T \mathbf{R} \mathbf{W}), \end{aligned} \quad (4.148)$$

$$\mathbf{W}^T \mathbf{W} = \mathbf{I}. \quad (4.149)$$

Для спрощення (4.148) введемо ортогональну матрицю \mathbf{P} з властивостями:

$$\mathbf{P} = \mathbf{W} \mathbf{W}^T, \quad \mathbf{P}^2 = \mathbf{P}, \quad \mathbf{P}^T = \mathbf{P}.$$

Тоді

$$\text{trace}(\mathbf{W}^T \mathbf{R} \mathbf{W}) = E\{\|\mathbf{P}\|^2\}. \quad (4.150)$$

Таким чином, задачу зводимо до пошуку такого M -вимірного підпростору, для якого середньоквадратична норма проекції \mathbf{x} є максимальною [169]. Такий підпростір утворює множина власних векторів $\mathbf{E} = (\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^M)$ матриці коваріантності \mathbf{R} , які відповідають максимальним власним значенням $\lambda^1 > \lambda^2 > \dots > \lambda^M$. Ортогональна матриця \mathbf{P} може бути вираженою у термінах власних векторів:

$$\mathbf{P} = \mathbf{E} \mathbf{E}^T. \quad (4.151)$$

Оскільки матриця \mathbf{W} не може бути однозначно визначеною за критерієм (4.148), представимо її у вигляді:

$$\mathbf{W} = \mathbf{E} \cdot \mathbf{C}, \quad (4.152)$$

де \mathbf{C} — довільна невинуджена $k \times k$ матриця повороту, яка задовольняє умову $\mathbf{C}^{-1} = \mathbf{C}^T$.

Відображення даних на підпростір є процесом стиску, точність якого можна оцінити шляхом реконструкції $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$.

Квадратична похибка реконструкції:

$$\varepsilon = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 = \|\mathbf{x}\|^2 - \|\mathbf{P}\mathbf{x}\|^2, \quad (4.153)$$

досягає мінімального значення за умови, що $E\{\|\mathbf{P}\mathbf{x}\|^2\} \rightarrow \min$.

Метод навчання за правилом Ойя, що використовує підпросторовий критерій [147]:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha \left[\mathbf{y}\mathbf{x} + \mathbf{y} \sum_{k=1}^M y_k \mathbf{w}_k(t) \right], \quad (4.154)$$

де $y_k = \mathbf{w}_k^T \mathbf{x}$.

Даний метод не дозволяє отримати асимптотично стабільний вектор вагових коефіцієнтів. У результаті ітераційного процесу він буде еволюціонувати у напрямку деякого неоднозначно заданого ортонормального базису в M -вимірному підпросторі власних векторів вхідної матриці коваріантності, що цілком співпадає з підпросторовим критерієм (4.152).

У [148] запропоновано введення асиметрії в процес обмеження зростання вагових коефіцієнтів з метою прискорення їх збіжності до власних векторів матриці коваріантності. Ітераційна формула такого асиметричного методу має вигляд:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha \mathbf{y} \left[\mathbf{x} + \boldsymbol{\theta} \sum_{k=1}^M y_k \mathbf{w}_k(t) \right], \quad (4.155)$$

де $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_N)$ — вектор коефіцієнтів асиметрії з упорядкованими елементами: $\theta_1 < \theta_2 < \dots < \theta_N$.

При такому співвідношенні коефіцієнтів асиметрії перший нейрон завжди матиме найвищу швидкість збіжності у напрямку відповідного власного вектора матриці \mathbf{R} . Доведено [148], що вектори вагових коефіцієнтів будуть збігатися до важливих власних векторів у порядку, який відповідає спаданню власних

значень. Цей висновок цілком узгоджується з мультиплікативним обмеженням $y(\mathbf{W}) = \mathbf{Ox} \sum_{k=1}^M y_k \mathbf{w}_k$.

4.4.3. Узагальнене правило Хебба

Узагальнене правило Хебба (метод Санджера) забезпечує використання технології пошуку важливих компонент для навчання одношарової ШНМ [163]. Процес навчання за узагальненим правилом Хебба складається з послідовності дій:

$$\begin{aligned} y_1 &= \mathbf{w}_1 \mathbf{x}, \quad x_1 = \mathbf{x} - y_1 \mathbf{w}_1^T, \quad \Delta \mathbf{w}_1 = \alpha y_1 \tilde{\mathbf{x}}_1^T, \\ y_2 &= \mathbf{w}_2 \mathbf{x}, \quad x_2 = \tilde{\mathbf{x}}_1 - y_2 \mathbf{w}_2^T, \quad \Delta \mathbf{w}_2 = \alpha y_2 \tilde{\mathbf{x}}_2^T, \\ &\dots \\ y_j &= \mathbf{w}_j \mathbf{x}, \quad \mathbf{x}_j = \tilde{\mathbf{x}}_{j-1} - y_j \mathbf{w}_j^T, \quad \Delta \mathbf{w}_j = \alpha y_j \tilde{\mathbf{x}}_j^T. \end{aligned} \quad (4.156)$$

Виходячи з (4.156), виразимо в матричній формі приріст вхідних даних:

$$\tilde{\mathbf{x}}_j^T = \mathbf{x}^T - [y_1, y_2, \dots, y_j, 0, \dots, 0] \cdot \mathbf{W} = \mathbf{x}^T - \tilde{\mathbf{y}}_j^T \mathbf{W}, \quad (4.157)$$

де $\tilde{\mathbf{y}}_j^T = [y_1, y_2, \dots, y_j, 0, \dots, 0]$ — вихідний вектор \mathbf{y} , у якому останні $m - j$ елементів замінено нулями.

Підставивши (4.157) у вираз для визначення приросту коефіцієнтів $\Delta \mathbf{w}_j$ нейрона j , одержимо:

$$\Delta \mathbf{w}_j = \alpha (y_j \mathbf{x}^T - y_j \tilde{\mathbf{y}}_j^T \mathbf{W}). \quad (4.158)$$

Або у загальному вигляді для всієї ШНМ:

$$\Delta \mathbf{W} = \alpha (\mathbf{y} \mathbf{x}^T - \text{tril}(\mathbf{y} \mathbf{y}^T) \cdot \mathbf{W}), \quad (4.159)$$

де $\text{tril}(\cdot)$ — нижньотрикутна матриця з нульовими елементами, які розташовані над головною діагоналлю.

Застосуємо оператор математичного сподівання до обох частин рівняння (4.159), враховуючи той факт, що у випадку збіжності методу $\Delta \mathbf{W} \rightarrow 0$:

$$0 = E\{\mathbf{W} \mathbf{x} \mathbf{x}^T\} - E\{\text{tril}(\mathbf{W} \mathbf{x} \mathbf{x}^T \mathbf{W}^T) \cdot \mathbf{W}\}. \quad (4.160)$$

Перепишемо (4.160) з урахуванням означення матриці коваріантності $\mathbf{R} = E\{\mathbf{x} \mathbf{x}^T\}$:

$$\mathbf{W} \cdot \mathbf{R} = \text{tril}(\mathbf{W} \mathbf{R} \mathbf{W}^T) \cdot \mathbf{W}. \quad (4.161)$$

Трикутна матриця в (4.161) має вигляд:

$$\mathbf{A} = \text{tril}(\mathbf{W} \mathbf{R} \mathbf{W}^T) = \begin{pmatrix} \lambda^1 & 0 & \dots & 0 \\ \lambda^{21} & \lambda^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \lambda^{m1} & \lambda^{m2} & \dots & \lambda^m \end{pmatrix}, \quad (4.162)$$

де $\lambda^{ji} = \mathbf{w}_j \mathbf{R} \mathbf{w}_i^T$.

Перепишемо (4.161) з урахуванням (4.162):

$$\mathbf{W} \mathbf{R} = \mathbf{A} \mathbf{W}. \quad (4.163)$$

Із (4.163) випливає рівняння для першого нейрона ШНМ:

$$\mathbf{w}_1 \mathbf{R} = \lambda^1 \mathbf{w}_1, \quad \text{де } \lambda^1 = \mathbf{w}_1 \mathbf{R} \mathbf{w}_1^T. \quad (4.164)$$

Вираз (4.164) повністю узгоджується з правилом Ойя для одного нейрона (4.147), оскільки вектор вагових коефіцієнтів першого нейрона ШНМ \mathbf{w}_1 за наявності власного значення λ^1 збігається до власного вектора вхідної матриці коваріантності \mathbf{R} . У той же час, всі недіагональні власні значення λ^{ji} збігаються до нуля через ортогональність власних векторів.

Блок-схему базового алгоритму Санджера для навчання за узагальненим правилом Хебба показано на рис.4.5. На етапі початкових установок вибирають значення вектора похибки $e_0(k)$ таким, що дорівнює значенню вхідного вектора даних $\mathbf{x}(k)$.

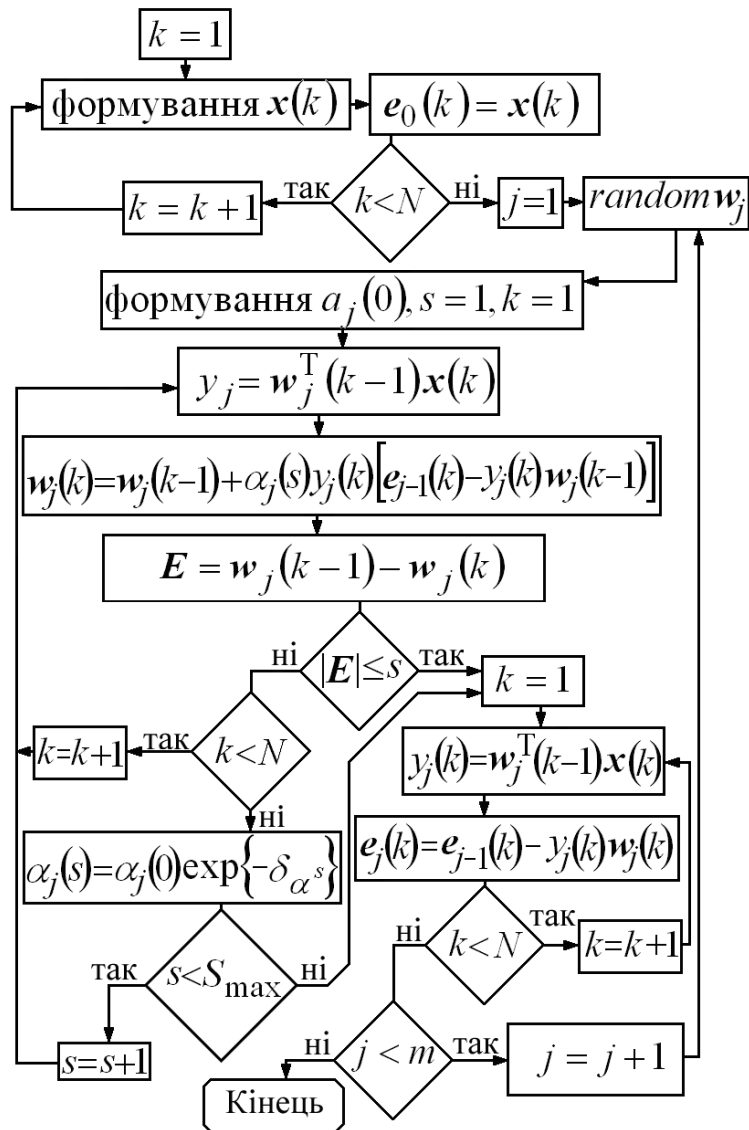


Рис.4.5. Блок-схема алгоритму навчання ШНМ за узагальненим правилом Хебба.

Значення векторів w_j вагових коефіцієнтів нейронів задають випадковим чином із нормальним розподілом густини імовірності.

В межах даного алгоритму вибір елементів вектора коефіцієнта навчання $\alpha_j(0)$ не має чіткого математичного обґрунтування. Його формують шляхом проб та помилок, враховуючи дисперсію вхідних даних. Основу алгоритму складають три вкладені цикли. Зовнішній цикл забезпечує почергове навчання нейронів, які входять до ШНМ. Кожен із нейронів має обмежену кількість допустимих стадій навчання S_{\max} . Якщо в межах поточної стадії не досягнуто збіжності вагових коефіцієнтів для даного нейрона, то відбувається експоненціальне зменшення величини коефіцієнта навчання за формулою:

$$\alpha_j(s) = \alpha_j(0)e^{-\delta_\alpha s}, \quad (4.165)$$

де s — поточний номер стадії навчання; δ_α — крок коефіцієнта навчання, який задає діапазон його зміни:

$$\delta_\alpha = \frac{\ln\left(\frac{\alpha_j(0)}{(\alpha_j)_{\min}}\right)}{S_{\max}}. \quad (4.166)$$

Власне внутрішній цикл алгоритму забезпечує навчання на вхідному наборі даних, що складається з N векторів. Якщо в процесі навчання досягнуто збіжності вектора вагових коефіцієнтів, тобто

$$|w_j(k-1) - w_j(k)| < \varepsilon, \quad (4.167)$$

то запускається кінцевий цикл для поточного номера нейрона. Призначення даного циклу полягає в корекції вектора помилки $e(k)$ та вихідного вектора $y(k)$. Ознакою закінчення алгоритму вважають збіжність (4.167) для всіх нейронів ШНМ.

Для кожного конкретного випадку застосування описаного алгоритму існують певні труднощі у підборі коефіцієнтів навчання. Вирішення цього питання запропоновано у [18]. Початкове значення коефіцієнта навчання вибирають, спираючись на стохастичні характеристики усього вхідного набору даних, на якому проводиться сеанс навчання:

$$\alpha_j(0) = \frac{1}{\sigma^2 - \sum_{i=1}^{j-1} \lambda^i}, \quad (4.168)$$

де σ^2 — дисперсія вхідного набору даних; $\lambda^i = \frac{1}{N} \sum_{k=1}^N y_i(k)^2$ —

власні значення, що відповідають попереднім власним векторам для нейрона j .

Подальша адаптація коефіцієнта навчання відбувається за допомогою ітераційного процесу, що забезпечує мінімізацію деякої евристичної функції:

$$\alpha_j(q_{k+1}) = \alpha_j(q_k) + cN \left[1 - \alpha_j(q_k) \sigma^2(q_k) \right] \times \left[\sigma^2(q_k) + \frac{2}{N} \sum_{i=1}^N y_i(e_j(q_k) \Delta w_j(q_k)) \right], \quad (4.169)$$

де $c = \frac{0,01}{\lambda^{j-1}}$ — евристична константа.

Такий підхід до визначення коефіцієнта навчання не може забезпечити його оптимальну змінюваність в процесі навчання, оскільки для визначення коефіцієнта необхідні дані про весь вхідний набір даних. Це є істотним недоліком у випадках, коли вхідні дані не є відомими заздалегіть. Коригування коефіцієнта навчання необхідно проводити також для забезпечення гарантованої збіжності алгоритму. В [57] приведено

модифікацію алгоритму з рекурсивним визначенням коефіцієнта навчання:

$$K_j(k) = \frac{P_j(k-1)y_j(k)}{1 + P_j(k-1)y_j(k)^2}, \quad (4.170)$$

$$P_j(k) = [1 - K_j(k)y_j(k)]P_j(k-1).$$

Ітераційна формула у даному випадку матиме вигляд:

$$w_j(k) = w_j(k-1) + K_j(k)[e_j(k) - w_j(k-1)y_j(k)]. \quad (4.171)$$

Переваги згаданих модифікацій алгоритмів для узагальненого методу навчання за правилом Хебба об'єднані в алгоритмі, що представлений у [126]. Цей алгоритм забезпечує пошук важливих компонентів з використанням ШНМ каскадної структури (рис.4.6).

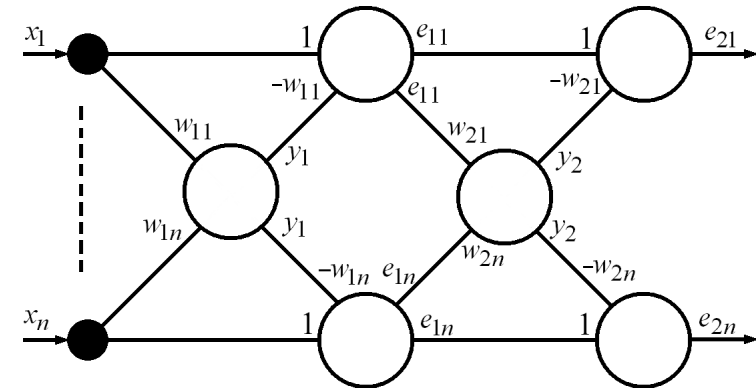


Рис.4.6. Каскадна ШНМ для пошуку важливих компонентів.

Суть даного підходу полягає в тому, що кожен наступний важливу компоненту шукають аналогічно до першої важливої компоненти, але для такого пошуку використовують не вектор вхідних даних x , а відповідний вектор помилки. Блок-схема алгоритму показана на рис.4.7.

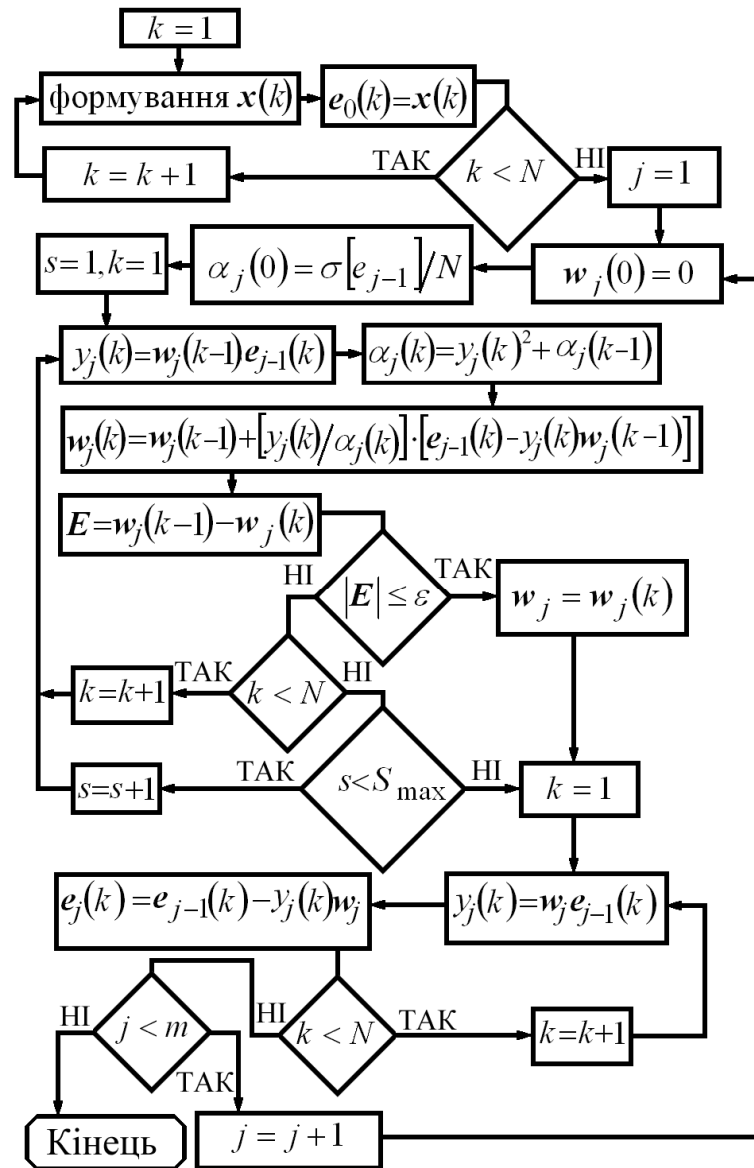


Рис.4.7. Блок-схема каскадного алгоритму навчання ШНМ за узагальненим правилом Хебба.

Для визначення власного значення $\lambda^j = E\{y_j^2\}$ застосовують дефляційний сигнал $E_j = [e_j(1), e_j(2), \dots, e_j(N)]$. Поточний вихідний сигнал ШНМ тепер формується за формулою:

$$y_j(k) = w_j(k-1)e_j(k), \quad (4.172)$$

де $e_j(k) = [e_{j1}(k), e_{j2}(k), \dots, e_{jn}(k)]^T$, $k = 1, 2, \dots, N$.

Така модифікація алгоритму дозволяє значно підвищити точність визначення вагових коефіцієнтів, оскільки вона вільна від найістотнішого недоліку, характерного для всіх вищезгаданих алгоритмів для узагальненого правила Хебба. Згаданий недолік проявляється у тому, що величини компонентів обернено пропорційні до ступеня їх важливості. Принцип каскадності дає можливість отримувати важливі компоненти, відсортовані у порядку спадання величини власного значення.

4.5. Методи навчання конкурентних нейронних мереж

ШНМ конкурентного типу базуються на двох основних типах шарів нейронів:

- шар вимірювання відстані;
- шар змагання типу „переможець забирає все” (WTA-Winner Take All).

Шар вимірювання відстані генерує вектор d , який задає відстань між вхідним вектором x та кожним із векторів вагових коефіцієнтів w_j , $j = 1, 2, \dots, m$.

Шар змагання типу „переможець забирає все” генерує вихідні сигнали y_i , де $i = 1, \dots, m$. Сигнал від нейрона-переможця, якому відповідає мінімальна відстань

$$d_j = \min_j D(x, w_j), \quad (4.173)$$

позначають логічною одиницею, а всі інші сигнали набувають значення логічного нуля. Структура конкурентної ШНМ показана на рис.4.8.

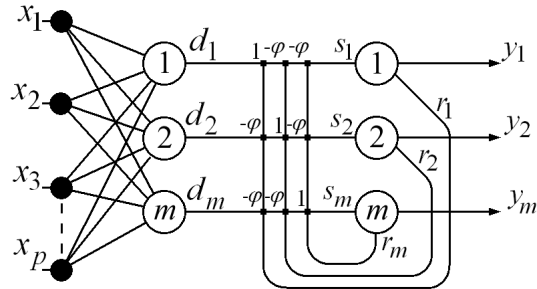


Рис.4.8. Конкурентна ШНМ.

Для вимірювання відстані використовують ряд оцінок, серед яких найбільш поширені: евклідова відстань, квадрат евклідової відстані, манхетенська відстань, відстань Чебишева, степенева відстань та проекція.

Розглянемо способи визначення згаданих відстаней між векторами a і b .

Евклідова відстань — широко поширений спосіб визначення відстані. Евклідова відстань є геометричною відстанню в багатовимірному просторі (рис.4.9).

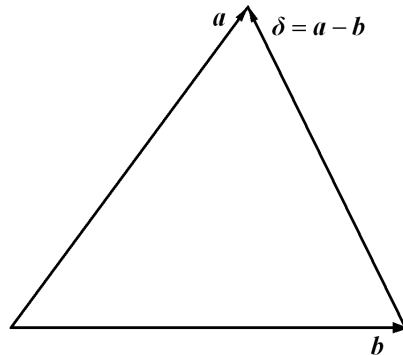


Рис.4.9. Евклідова відстань.

$$d = D(a, b) = \|a - b\| = \|\delta\| = \sqrt{\sum_i (a_i - b_i)^2} = \sqrt{\delta^T \delta}. \quad (4.174)$$

Операція добування квадратного кореня вносить певну обчислювальну складність у використання евклідової відстані. Тому для спрощених оцінок застосовують квадрат евклідової відстані.

Квадрат евклідової відстані використовують також у випадках, коли необхідно надати більшого значення віддаленим об'єктам. Її визначають із виразу:

$$d = D(a, b) = \|a - b\|^2 = \|\delta\|^2 = \sum_i (a_i - b_i)^2 = \sum_i \delta_i^2 = \delta^T \delta. \quad (4.175)$$

Манхетенська відстань відповідає сумі покоординатних різниць векторів. Таку відстань застосовують при необхідності згладжування впливу викидів на загальну величину відстані, що задана виразом:

$$d = D(a, b) = \sum_i |a_i - b_i| = \sum_i |\delta_i| = \text{sum}(\text{abs}(\delta)). \quad (4.176)$$

Відстань Чебишева. Такий спосіб задавання відстані використовують, коли необхідно розпізнати об'єкти, що відрізняються за однією координатою. Формула для відстані Чебишева має вигляд:

$$d = D(a, b) = \max_i (|a_i - b_i|) = \max_i (|\delta_i|). \quad (4.177)$$

Степенева відстань. Цей спосіб є узагальненням евклідової відстані. Формула степеневої відстані містить параметри p і r , які задають для кожного конкретного випадку:

$$d = D(a, b) = r \sqrt[r]{\sum_i |a_i - b_i|^p} = r \sqrt[r]{\sum_i |\delta_i|^p}. \quad (4.178)$$

Проекція вектора a на вектор b — це найпростіший спосіб вимірювання подібності нормалізованих векторів (рис.4.10):

$$d = D(a, b) = b^T a = \|b\| \cdot \|a\| \cdot \cos \varphi. \quad (4.179)$$

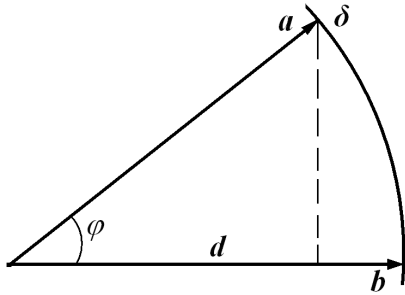


Рис.4.10. Проекція.

При нормалізації векторів \mathbf{a} та \mathbf{b} , тобто при $\|\mathbf{a}\| = \|\mathbf{b}\| = 1$, одержимо відстань $d = \cos \varphi$.

Визначимо відстань-проекцію вектора \mathbf{a} на вектор \mathbf{b} через вектор різниці $\delta = \mathbf{a} - \mathbf{b}$ з метою використання проекції як міри подібності векторів. Для цього зробимо деякі перетворення:

$$\begin{aligned} \|\delta\|^2 &= (\mathbf{a} - \mathbf{b})^T (\mathbf{a} - \mathbf{b}) = \mathbf{a}^T \mathbf{a} - 2\mathbf{a}^T \mathbf{b} + \mathbf{b}^T \mathbf{b} = \\ &= \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a}^T \mathbf{b} = 2 - 2\mathbf{a}^T \mathbf{b}. \end{aligned} \quad (4.180)$$

Звідси одержимо:

$$d = 1 - \frac{1}{2} \|\delta\|^2. \quad (4.181)$$

Нормалізація вхідних векторів даних може бути виконана без втрати інформації шляхом формування додаткового виміру в процесі обробки даних. Нехай маємо послідовність $(p-1)$ -вимірних вхідних векторів $\tilde{\mathbf{x}}(n)$, де $n = 1, 2, \dots, N$.

За умовою вектори $\tilde{\mathbf{x}}(n)$ нормалізовані, тобто

$$\|\tilde{\mathbf{x}}(n)\| \leq 1 \quad \forall n = 1, \dots, N. \quad (4.182)$$

Додамо до вектора $\tilde{\mathbf{x}}(n)$ компоненту $x_p(n)$, виходячи зі співвідношення:

$$\|\mathbf{x}\|^2 = \|\tilde{\mathbf{x}}\|^2 + x_p^2. \quad (4.183)$$

З урахуванням нормалізації додаткова компонента вхідного вектора має вигляд:

$$x_p^2 = 1 - \|\tilde{\mathbf{x}}\|^2. \quad (4.184)$$

Вираз (4.184) задає проекцію $(p-1)$ -вимірної гіперплощини на p -вимірну гіперсферу.

Функціонування шару змагання задають правилами визначення нейрона-переможця:

$$y_j = \begin{cases} 1 & \text{при } j = \arg \min_k \mathbf{D}(\mathbf{x}, \mathbf{w}_k), \\ 0 & \text{в інших випадках.} \end{cases} \quad (4.185)$$

Топологія шару змагання являє собою рекурентну ШНМ з фіксованою матрицею зворотних зв'язків:

$$\mathbf{M} = \begin{pmatrix} 1 & \dots & -\varphi_{1j} & \dots & -\varphi_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ -\varphi_{i1} & \dots & 1 & \dots & -\varphi_{im} \\ \dots & \dots & \dots & \dots & \dots \\ -\varphi_{n1} & \dots & -\varphi_{nj} & \dots & 1 \end{pmatrix}. \quad (4.186)$$

У більшості випадків $\varphi_{ij} = \varphi < 1 \quad \forall (i \neq j)$.

Стан шару змагання виражають вектором \mathbf{s} , що еволюціонує відповідно до ітераційного рівняння:

$$\mathbf{s}(n+1) = \mathbf{M}\mathbf{r}(n) + \mathbf{d}\delta(n), \quad (4.187)$$

де $\delta(n)$ – дельта функція, яка дозволяє участь вектора \mathbf{d} тільки на першому етапі ітераційного процесу:

$$\delta(n) = \begin{cases} 1 & \text{при } n = 0, \\ 0 & \text{при } n \neq 0. \end{cases} \quad (4.188)$$

Вектор $\mathbf{r}(n)$ — це вектор зворотного зв'язку, формування якого відбувається шляхом відкидання від'ємних значень елементів вектора \mathbf{s} :

$$r(n) = \max(0, s_j(n)) = \begin{cases} s_j(n) & \text{при } s_j(n) \geq 0, \\ 0 & \text{при } s_j(n) < 0. \end{cases} \quad (4.189)$$

$j = 1, 2, \dots, m.$

Формуючи зворотний зв'язок за правилом (4.189), перепишемо умову визначення сигналу від нейрона-переможця (4.185) у такому вигляді:

$$y_j(n) = \begin{cases} 1 & \text{при } r_j(n) > 0, \\ 0 & \text{при } r_j(n) = 0. \end{cases} \quad (4.190)$$

Основною метою навчання нейронних мереж конкурентного типу є адаптивне квантування вхідного простору, що характеризується організацією даних у вигляді кластерів. Розглянемо ряд методів, що реалізують цю ідею, починаючи з простого методу конкурентного навчання.

4.5.1. Простий метод конкурентного навчання

За цим методом спочатку задають випадкові значення вагових коефіцієнтів для шару вимірювання відстані:

$$\mathbf{w}_j(0) = \mathbf{x}^T(\text{rand}(j)), \quad j = 1, 2, \dots, m \quad (4.191)$$

Для кожного вхідного вектора $\mathbf{x}(n)$ на шарі змагання визначають нейрон-переможця j з умови мінімальної відстані вхідного вектора від вектора вагових коефіцієнтів даного нейрона $\mathbf{w}_j(n)$:

$$y_j(n) = \begin{cases} 1 & \text{при } d_j(n) = \min_j \mathbf{D}(\mathbf{x}(n), \mathbf{w}_j(n)), \\ 0 & \text{в інших випадках.} \end{cases} \quad (4.192)$$

Наступним кроком є ітеративне формування змін до вагових коефіцієнтів:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \alpha(n) y_j(n) [\mathbf{x}^T - \mathbf{w}_j(n)]. \quad (4.193)$$

Як видно з рис.4.11, зміна вектора \mathbf{w}_j відбувається в напрямку вектора \mathbf{x} . У зв'язку з урахуванням вихідного сигналу $y_j(n)$ в ітераційній формулі (4.193), модифікація вагових коефіцієнтів є актуальною тільки для нейрона-переможця.

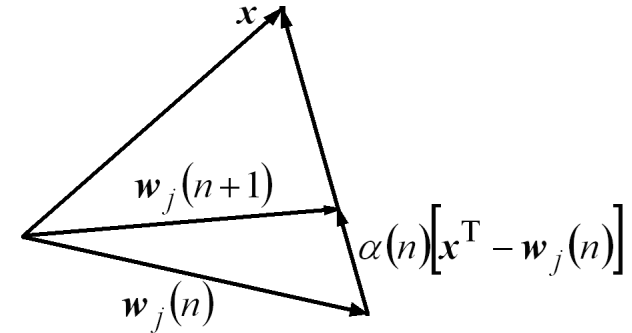


Рис.4.11. Ітераційний процес наближення вектора \mathbf{w}_j до вектора \mathbf{x} .

Для покращення збіжності методу використовують динамічне зменшення коефіцієнта навчання. Найбільш поширеним є евристичний підхід до такого зменшення:

$$\alpha(n) = 0,1 \cdot \left(1 - \frac{n}{N}\right). \quad (4.194)$$

Більш складними є кластерні методи навчання, які також можуть бути реалізованими на ШНМ конкурентного типу.

4.5.2. Кластерні методи навчання

Кластерні методи навчання без учителя застосовують із метою пошуку деяких структур (кластерів) у вхідному потоці даних [126]. У протилежності до технології аналізу важливих компонентів, де дані задають проекціями у ортогональному базисі, кластерний аналіз працює з даними у вигляді радіальних

базисних функцій. Основною ідеєю даного підходу є пошук множини кластерних центрів $W = (w_1, w_2, \dots, w_k, \dots, w_m)$, яка найкращим чином описує вхідний набір даних. Вектор x , що є елементом вхідних даних, досліджують на приналежність до кластера k шляхом визначення його відстані від центру w_k .

Вектор x вважають таким, що належить до кластера w_k , якщо

$$D(x, w_k) < D(x, w_j) \quad \forall k \neq j. \quad (4.195)$$

Одним із широко відомих кластерних методів є метод векторного квантування. Цей метод забезпечує пошук кластерних центрів шляхом мінімізації цільової функції:

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^m \Phi_k[x(n)] \cdot \|x(n) - w_k\|^2, \quad (4.196)$$

де $K \ll N$; $\Phi_k[x(n)]$ — функція приналежності, яка набуває значення:

$$\Phi_k[x(n)] = \begin{cases} 1 & D(x, w_k) < D(x, w_j) \quad \forall k \neq j, \\ 0 & \text{в інших випадках.} \end{cases} \quad (4.197)$$

Застосувавши до (4.196) метод градієнтного спуску:

$$\frac{\partial W}{\partial t} = -\alpha \frac{\partial E}{\partial W}, \quad (4.198)$$

одержимо ітераційну формулу методу векторного квантування:

$$w_k(n+1) = w_k(n) + \alpha \sum_{n=1}^N \Phi_k[x(n)] \cdot [x(n) - w_k]. \quad (4.199)$$

Алгоритм навчання ШНМ за методом векторного квантування показаний на рис.4.12. За цим алгоритмом спочатку надають випадкові значення векторам вагових коефіцієнтів w_k , кожен з яких відповідає центру кластера. Далі на вхід ШНМ подають перший вектор вхідних даних $x(0)$ і обчислюють відстані між ним та кожним із заданих центрів кластерів.

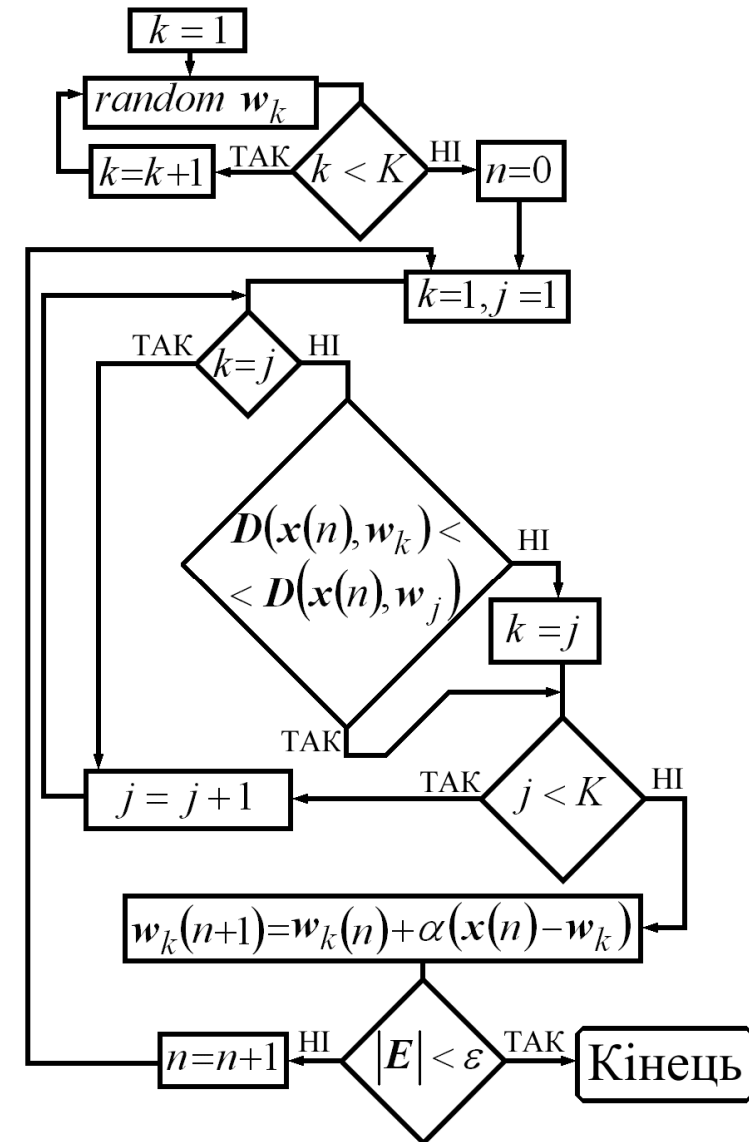


Рис.4.12. Алгоритм навчання за методом векторного квантування.

У даному циклі модифікувати потрібно лише той центр, відстань якого від вхідного вектора є мінімальною. За умови, що перевірка глобальної ознаки збіжності не дала позитивного результату, на вхід ШНМ подають черговий вектор вхідних даних $\mathbf{x}(n)$, і цикл навчання повторюється. Ознакою закінчення алгоритму є мінімальне значення цільової функції або мінімальні зміни вагових коефіцієнтів $\mathbf{w}_k(n)$.

Після того, як знайдено центри кластерів, вони можуть бути використані для кодування вхідного потоку даних. Найпростіший спосіб кодування – це заміна векторів, що належать до даного кластера, однією змінною, яка містить кількість знайдених векторів.

Існують також більш складні способи, в які крім інших параметрів входить і відстань до центру.

4.5.3. Карти Кохонена

З 1988 року набули широкого поширення карти, що мають здатність до самоорганізації. Вони вперше були запропоновані Кохоненом [115]. Карти Кохонена є подальшим розвитком методу векторного квантування, що дає можливість формування нових топологій ШНМ у відповідності до вигляду функції „сусідства” Φ_{c_j} . На відміну від методів векторного квантування, метою яких є встановлення оптимальних меж кластерів при їх заданій кількості, карти Кохонена мають властивість автономного пошуку та формування нових кластерів [117].

Існує велика кількість топологій карт Кохонена, одна з яких описана в розділі 1 цієї книги. Переважна більшість з них забезпечує відображення вхідного простору даних у одновимірний або двовимірний масив вихідних сигналів [41]. На рис.13 показано спрощену структуру ШНМ Кохонена, що забезпечує відображення потоку n -вимірних вхідних векторів

$$\mathbf{x}(i) = (x_1(i), x_2(i), \dots, x_n(i)),$$

де $i = 1, 2, \dots, P$, у вихідний вектор $\mathbf{y} = (y_1, y_2, \dots, y_m)$.

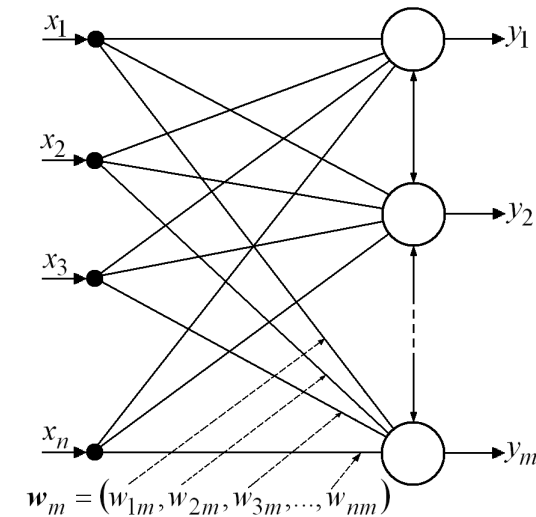


Рис.4.13. Спрощена структура ШНМ Кохонена.

З кожним із вихідних нейронів асоціюється вектор вагових коефіцієнтів $\mathbf{w}_j = (w_{1j}, w_{2j}, \dots, w_{nj})$, елементам якого спочатку надають невеликі випадкові значення. Вхідний вектор \mathbf{x} порівнюють з усіма векторами вагових коефіцієнтів \mathbf{w}_j .

Критерієм такого порівняння найчастіше вибирають квадрат евклідової відстані.

Відстань між вектором вагових коефіцієнтів нейрона-переможця та вхідним вектором:

$$d_j = \min_j \left\| \sum_{i=1}^n (x_i - w_{ij}) \right\|^2 = \min_j \|\mathbf{x} - \mathbf{w}_j\|^2 = \|\mathbf{x} - \mathbf{w}_c\|^2, \quad (4.200)$$

де \mathbf{w}_c — вектор вагових коефіцієнтів нейрона-переможця c .

Визначення нейрона-переможця ініціює процес навчання області, яка включає сам нейрон та його деякий отік, за ітераційною формулою:

$$w_j(n+1) = w_j(n) + \Phi_{cj}(n) [x(n) - w_j(n)]. \quad (4.201)$$

Функція сусідства $\Phi_{cj}(n) = \Phi(\|l_c - l_j\|, n)$ залежить від відстані між координатою l_c нейрона-переможця c та координатою l_j поточного нейрона j на карті Кохонена. Найпростіша функція сусідства фіксує область активізації процесу навчання $L_c(n)$, розташовану навколо нейрона-переможця:

$$\Phi_{cj}(n) = \begin{cases} \alpha(n) \text{ при } j \in L_c(n), \\ 0 \text{ в інших випадках,} \end{cases} \quad \lim_{n \rightarrow \infty} L_c(n) = 1, \quad (4.202)$$

де $\alpha(n)$ — коефіцієнт навчання.

Широко поширені функції сусідства, що базуються на законі розподілу Гауса:

$$\Phi_{cj}(n) = \alpha(n) \exp\left(-\frac{\|l_c - l_j\|^2}{2\sigma^2(n)}\right), \quad (4.203)$$

де $\sigma(n)$ — параметр, що задає розмір області сусідства та монотонно зменшується з часом.

На рис.4.14 показаний алгоритм самоорганізації карт Кохонена [116]. У даному алгоритмі, подібно до алгоритму векторного квантування, спочатку задають випадкові значення векторам вагових коефіцієнтів w_k . Далі на вхід ШНМ подають перший вектор вхідних даних $x(0)$ і виконують пошук нейрона, який характеризується найменшою відстанню між даним вхідним вектором і відповідним вектором вагових коефіцієнтів w_k . Такий нейрон є нейроном-переможцем c даного етапу навчання.

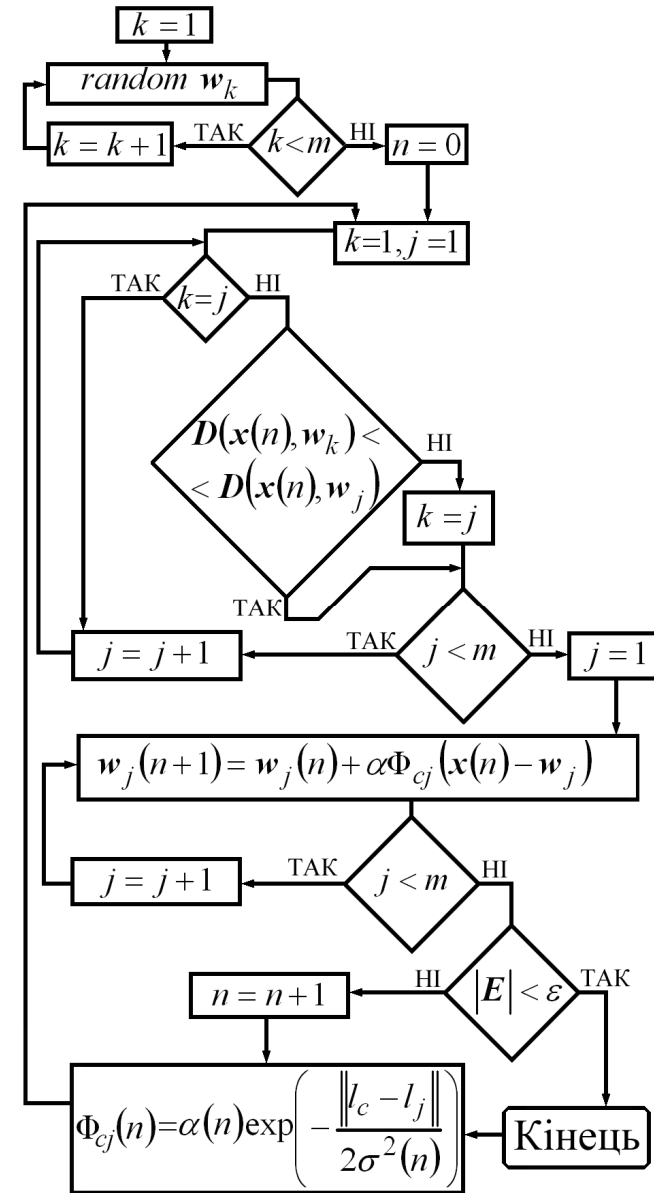


Рис.4.14. Алгоритм самоорганізації карт Кохонена.

На наступному етапі виконують модифікацію вагових коефіцієнтів за формулою (4.201) з урахуванням функції сусідства, яка задає допустиму область модифікації. На рис.4.14 спрощено показаний етап перевірки збіжності алгоритму, спосіб реалізації якого залежить від конкретних завдань, що вирішують за допомогою самоорганізації карт Кохонена. У випадку негативного результату перевірки критерію збіжності починають новий цикл навчання. Першим кроком цього циклу є модифікація функції сусідства Φ_{cj} , яка спричиняє зменшення області збудження навколо нейрона-переможця. Далі на вхід надходить черговий вектор вхідних даних $\mathbf{x}(n)$, і процес навчання продовжується до досягнення позитивного результату оцінки критерію збіжності.

4.5.4. Нейронний газ

Метод нейронного газу, запропонований у [128], отримав свою назву завдяки деякій математичній аналогії між процесами навчання та дифузними процесами у газовому середовищі. Даний метод адаптації вагових коефіцієнтів базується на використанні поняття **ранжирування околу** векторів вагових коефіцієнтів \mathbf{w}_i для даного вхідного вектора даних \mathbf{x} . Розходження між вхідним вектором \mathbf{x} та векторами вагових коефіцієнтів нейронів задано множиною викривлень:

$$\mathbf{D}_x = \{d_i(\mathbf{x}, \mathbf{w})\}_{i=1}^m = \{\|\mathbf{x} - \mathbf{w}_i\|\}_{i=1}^m, \quad (4.204)$$

де m — кількість нейронів у ШНМ.

У випадку послідовної реалізації чергова множина викривлень \mathbf{D}_x підлягає процесу сортування з затратами часу, пропорційними $O(m \log m)$. Після сортування першому елементу множини d_1 відповідає вектор вагових коефіцієнтів \mathbf{w}_i , найближчий до \mathbf{x} . Останній елемент d_m формується за допомогою вектора вагових коефіцієнтів, що є найвіддаленішим

від \mathbf{x} . У зв'язку з ранжуванням векторів вагових коефіцієнтів говорять про заміну принципу „переможець забирає все” на принцип „переможець забирає найбільше”. Для модифікації вагових коефіцієнтів за останнім принципом використовують ітераційну формулу:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha \cdot f_i(\mathbf{D}_x) \cdot (\mathbf{x}(t) - \mathbf{w}_i), \quad (4.205)$$

де $\alpha \in [0, 1]$ — коефіцієнт навчання; $f_i(\mathbf{D}_x) \in [0, 1]$ — функція, що відповідає топологічному розміщенню векторів \mathbf{w}_i у вхідному просторі.

В [127] запропоновано ефективне аналітичне визначення функції топологічного розміщення:

$$f_i(\mathbf{D}_x) = \exp\left(-\frac{k_i(\mathbf{x}, \mathbf{w})}{\lambda}\right), \quad (4.206)$$

де λ — параметр, що задає кількість векторів вагових коефіцієнтів із ранжуваного околу, які потрібно модифікувати під час ітераційного кроку; $k_i(\mathbf{x}, \mathbf{w}) = 0, 1, \dots, m-1$ — індекс ранжування околу векторів \mathbf{w}_i для поточного вектора \mathbf{x} .

Нехай нейрону-переможцю відповідає вектор \mathbf{w}_c , що є найближчим до поточного вхідного вектора \mathbf{x} . Тоді індекс ранжування для нього $k_c = 0$. Індексом ранжування $k_i = 1$ може бути позначений довільний із $m-1$ залишкових векторів вагових коефіцієнтів \mathbf{w}_i за умови, що він знаходиться на другому місці по близькості до вхідного вектора. Кандидата на третє місце вибирають уже з множини $m-2$ залишкових векторів \mathbf{w}_i . Враховуючи той факт, що кожному з векторів \mathbf{w}_i може відповідати довільний допустимий індекс ранжирування, запишемо ранжирувану послідовність векторів вагових коефіцієнтів у загальному вигляді:

$$(\mathbf{w}_{i_0}, \mathbf{w}_{i_1}, \dots, \mathbf{w}_{i_{m-1}}), \quad (4.207)$$

де \mathbf{w}_{i_0} — вектор з індексом ранжирування $k_{i_0} = 0$.

Алгоритм навчання за методом нейронного газу показаний на рис.4.15.

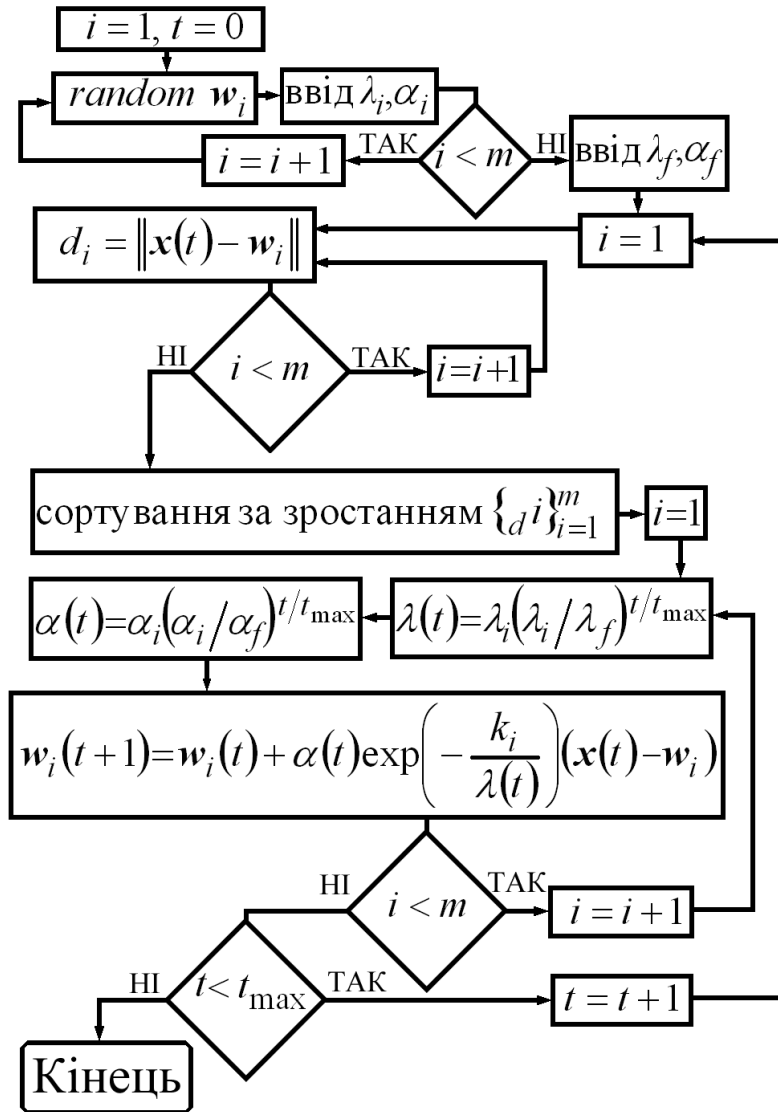


Рис.4.15. Алгоритм нейронного газу.

Традиційно для кластерних алгоритмів спочатку задають випадкові значення векторів вагових коефіцієнтів w_i . В цьому ж циклі вводять початкові значення параметрів λ_i та коефіцієнтів навчання α_i . Перед початком процесу навчання вводять також кінцеві значення λ_f та α_f . Перший етап навчання полягає в обчисленні викривлень $d_i = \|\mathbf{x}(t) - \mathbf{w}_i\|$ для кожного нейрона ШНМ. Елементи одержаної множини викривлень вхідного вектора $\mathbf{D}_x = \{d_i(\mathbf{x}, \mathbf{w})\}_{i=1}^m$ сортують за зростанням. Ітераційний процес модифікації вагових коефіцієнтів виконується з попереднім обчисленням відповідного параметра:

$$\lambda(t) = \lambda_i \left(\lambda_i / \lambda_f \right)^{\frac{t}{t_{\max}}},$$

та коефіцієнта навчання: $\alpha(t) = \alpha_i \left(\alpha_i / \alpha_f \right)^{\frac{t}{t_{\max}}}$.

Для кожного нейрона мережі виконують адаптацію вектора вагових коефіцієнтів до поточного вхідного вектора шляхом проведення t_{\max} ітерацій. Після цього на вхід нейронної мережі подають черговий вхідний вектор даних. Ознакою закінчення навчання є вичерпання вхідного потоку даних. Одним із вузьких місць даного алгоритму є затрати часу на виконання процесу сортування. Тому в [21] запропоновано спосіб прискорення виконання алгоритму шляхом обмеження кількості елементів, які потрібно розсортувати, лише деякою кількістю найближчих до вхідного вектора. Ця кількість може зменшуватися в процесі навчання, що приводить до злиття даного методу з методом K -середніх.

4.5.5. Нейронний газ із конкурентним правилом Хебба

Метод нейронного газу часто застосовують у поєднанні з конкурентним правилом Хебба. Особливість цього правила

полягає у тому, що воно не містить механізму модифікації вагових коефіцієнтів і може бути використаним як додатковий механізм у поєднанні з різними методами навчання. Конкурентне правило Хебба лише забезпечує модифікацію топології нейронної мережі шляхом введення додаткових міжнейронних зв'язків.

Задамо множину $\mathcal{S} = \{s_1, s_2, \dots, s_i, \dots, s_m\}$, що містить m нейронів s_i та множину міжнейронних зв'язків $\mathcal{C} \subset \mathcal{S} \times \mathcal{S}$ з початковим станом $\mathcal{C} = \emptyset$. Кожному нейрону s_i відповідає вектор вагових коефіцієнтів w_{s_i} , початкове значення якого визначається густиною імовірності $p(\mathbf{x})$. Процес побудови міжнейронних зв'язків за конкурентним правилом Хебба починається з пошуку двох нейронів з векторами вагових коефіцієнтів, найближчими до поточного вхідного вектора. Перший нейрон знаходять із виразу

$$v_1 = \arg \min_{s_i \in \mathcal{S}} (\|\mathbf{x} - w_{s_i}\|) \quad (4.208)$$

шляхом пошуку вектора вагових коефіцієнтів, що максимально співпадає з вхідним вектором даних. Другий нейрон повинен задовольняти умову:

$$v_2 = \arg \min_{s_i \in \mathcal{S} \setminus \{s_1\}} (\|\mathbf{x} - w_{s_i}\|), \quad (4.209)$$

яка задає пошук на множині нейронів за виключенням уже знайденого елемента v_1 . Якщо ж між нейроном v_1 та нейроном v_2 не існує зв'язку, то такий зв'язок встановлюється шляхом доповнення множини зв'язків:

$$\mathcal{C} = \mathcal{C} \cup \{v_1, v_2\}. \quad (4.210)$$

Спільне використання конкурентного правила Хебба та методу нейронного газу потребує введення додаткового механізму відкидання міжнейронних зв'язків, які не використовуються [129]. Такий механізм називають механізмом старіння зв'язків. Суть його полягає у тому, що з кожним

зв'язком співвідносять деяку змінну, яка відображає вік. Безпосередньо після створення вік нового зв'язку дорівнює нулю. Операція його введення супроводжується інкрементом віку всіх зв'язків нейрона-переможця v_1 . Застарілий зв'язок відкидається у випадку, коли його вік перевищує критичну межу. Алгоритм, що поєднує конкурентне правило Хебба та метод нейронного газу, показаний на рис.4.16.

Для покращення збіжності алгоритму застосовують змінну величину порога старіння:

$$T(t) = T_i (T_i / T_f)^{t/t_{\max}}, \quad (4.211)$$

де T_i — початкове значення порога старіння для $a_{\{v_1, s_i\}} \geq T(t)$ зв'язків i -го нейрона; T_f — кінцеве значення порога старіння.

Для кожного нейрона-переможця проводиться перевірка віку його зв'язків. За умови, яка вказує на факт перевищення зв'язком $\{v_1, s_i\}$ порога старіння $T(t)$, даний зв'язок виключають із множини зв'язків за допомогою операції виключення:

$$\mathcal{C} = \mathcal{C} \cap \overline{\{v_1, s_i\}} = \mathcal{C} \setminus \{v_1, s_i\}. \quad (4.212)$$

Ознакою завершення алгоритму може бути досягнення допустимої кількості ітерацій t_{\max} , стабілізація вагових коефіцієнтів \mathcal{W} або елементів множини \mathcal{C} .

4.5.6. Нейронний газ із нарощуванням

Метод нейронного газу з нарощуванням нагадує уже розглянутий метод нейронного газу з конкурентним правилом Хебба. Власне, він містить всі основні механізми цього методу, до яких слід віднести ранжування околу нейрона-переможця, правило встановлення міжнейронних зв'язків та механізм старіння.

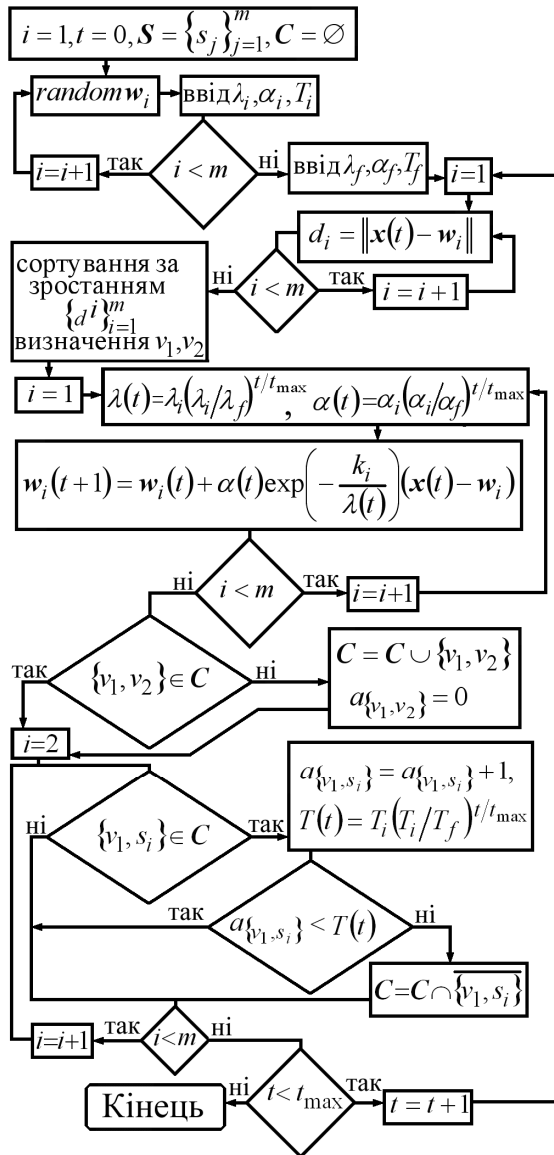


Рис.4.16. Алгоритм нейронного газу з конкурентним правилом Хебба.

Принципово новою властивістю даного методу є можливість нарощування кількості нейронів ШНМ в процесі її навчання [76]. Механізм методу вказує на місце включення нових елементів у топологію нейронної мережі. Критерієм, за яким визначають місце модифікації ШНМ, є локальна похибка, що підлягає безперервній оцінці в ході процесу навчання. Новий нейрон завжди додають у тому місці, яке характеризується максимальною локальною похибкою. Блок-схему алгоритму показано на рис.4.17а,б. Робота алгоритму починається на ШНМ, множина нейронів якої $S = \{s_1, s_2\}$ містить лише два елементи. Множина міжнейронних зв'язків $C \subset S \times S$ має початковий стан $C = \emptyset$. Із кожним нейроном s_i асоціюється вектор вагових коефіцієнтів w_{s_i} , початкове значення якого визначається густиною імовірності $p(x)$. Перші етапи роботи даного алгоритму подібні до етапів алгоритму нейронного газу з конкурентним правилом Хебба. Процес навчання також починається з роботи механізму побудови міжнейронних зв'язків, що забезпечує зв'язок між нейроном-переможцем та нейроном, який посідає друге місце за близькістю його вектора вагових коефіцієнтів до вхідного вектора. Умови визначення даних нейронів у загальному вигляді задані виразами (4.198) та (4.199) і можуть мати різну алгоритмічну реалізацію.

Найпростіший спосіб їх пошуку застосований у алгоритмі, що показаний на рис.4.17а. Поточну множину нейронів спочатку упорядковують за критерієм зростання відстані між вхідним вектором та векторами вагових коефіцієнтів:

$$d_i = \|x(t) - w_i\|, \quad (4.213)$$

а потім відбирають перший v_1 та другий v_2 елементи з упорядкованої множини. Якщо перевірка множини C виявляє відсутність зв'язку між нейронами v_1 та v_2 , то такий зв'язок встановлюється за правилом (4.210) з величиною віку $a_{\{v_1, v_2\}} = 0$.

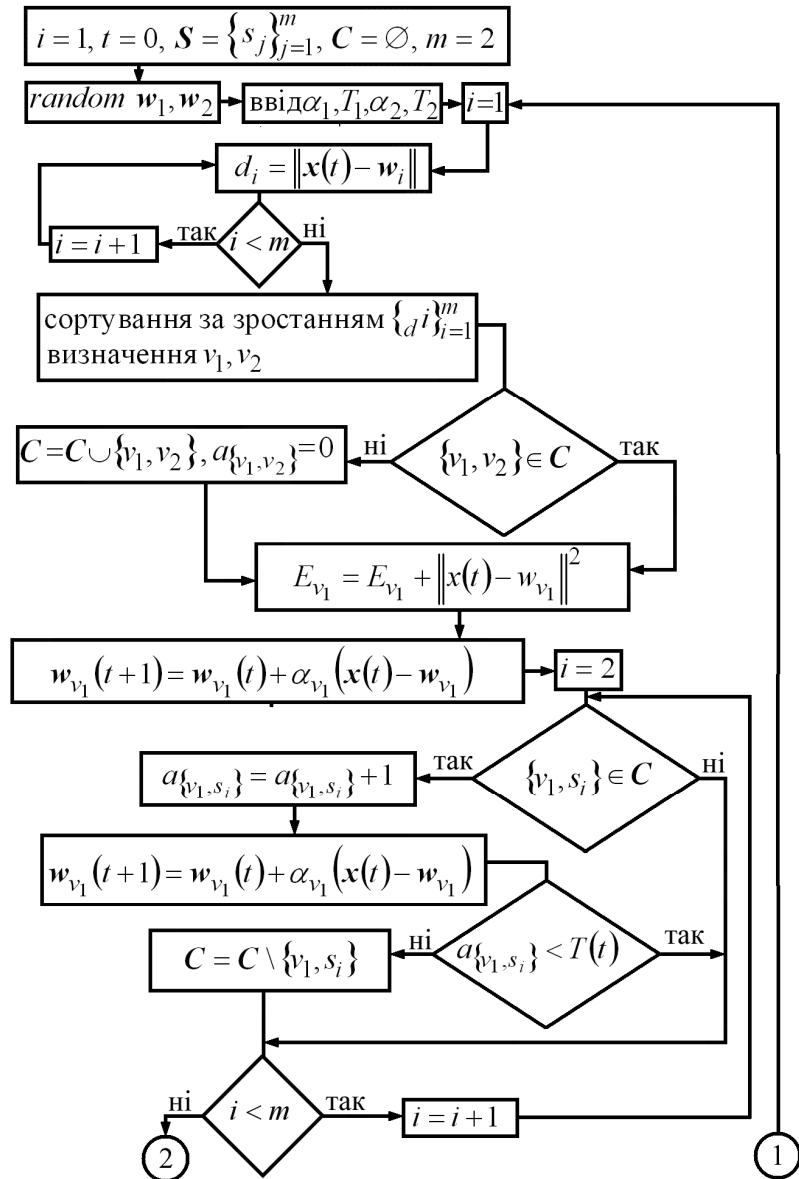


Рис.4.17а. Алгоритм нейронного газу з нарощуванням.

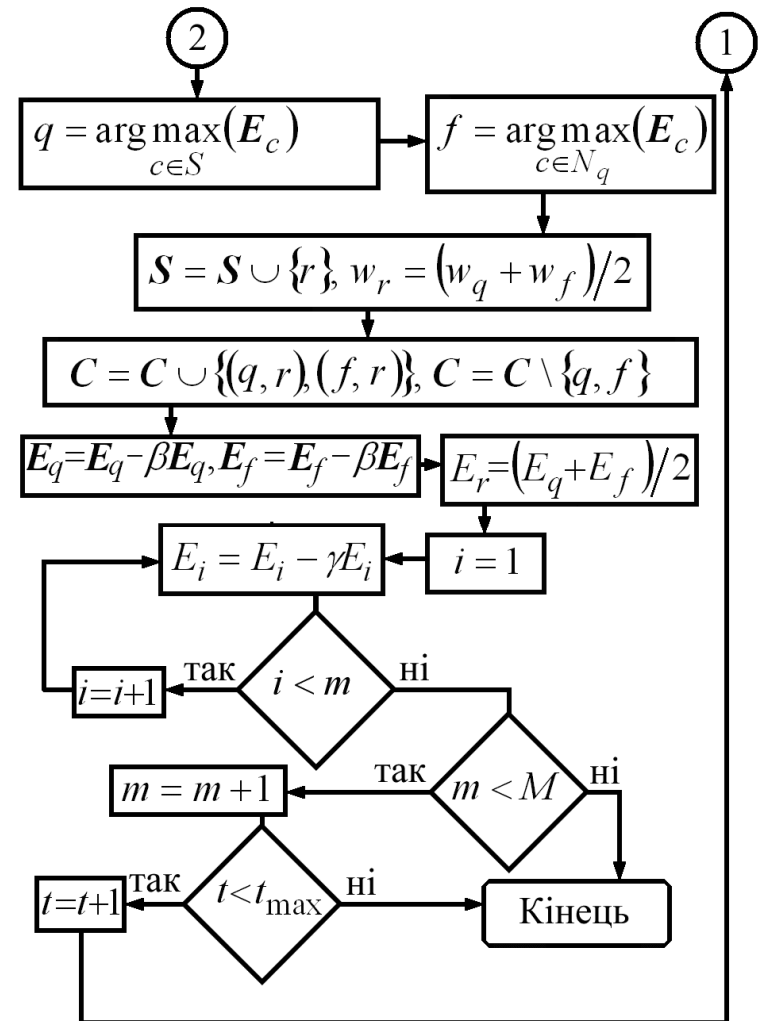


Рис.4.17б. Алгоритм нейронного газу з нарощуванням.

З кожним нейроном ШНМ співвідносять локальну похибку:

$$E_{v_1} = E_{v_1} + \|\mathbf{x}(t) - \mathbf{w}_{v_1}\|^2, \quad (4.214)$$

яку потрібно модифікувати тільки для нейрона-переможця. Для цього ж нейрона виконується ітеративна модифікація вектора вагових коефіцієнтів:

$$\mathbf{w}_{v_1}(t+1) = \mathbf{w}_{v_1}(t) + \alpha_{v_1}(\mathbf{x}(t) - \mathbf{w}_{v_1}). \quad (4.215)$$

Наступний етап алгоритму реалізує операції модифікації віку зв'язків нейрона-переможця та ітеративної зміни векторів вагових коефіцієнтів усіх нейронів, що безпосередньо зв'язані з нейроном-переможцем. У випадку, коли вік зв'язку $\{v_1, s_i\}$ перевищує допустимий для нього вік T_i , даний зв'язок розривається шляхом виключення його з множини міжнейронних зв'язків C .

Частину алгоритму, що відповідає за зміну топології ШНМ, показано на рис.4.17б. Робота даного етапу починається з пошуку нейрона q , який характеризується максимальним значенням локальної похибки E_c :

$$q = \arg \max_{c \in S} (E_c). \quad (4.216)$$

Далі виконують пошук на множині N_q , що містить нейрони, безпосередньо зв'язані з нейроном q , з метою знаходження нейрона f , який характеризується максимальною на даній множині N_q локальною похибкою:

$$f = \arg \max_{c \in N_q} (E_c). \quad (4.217)$$

Нейрони q та f є предками нового нейрона r , який приєднують до ШНМ шляхом включення його до множини $S = S \cup \{r\}$. Вектор вагових коефіцієнтів нейрона r визначають як середнє арифметичне векторів нейронів q та f :

$$\mathbf{w}_r = (\mathbf{w}_q + \mathbf{w}_f)/2. \quad (4.218)$$

Зміна топології ШНМ відбувається шляхом введення зв'язків $C = C \cup \{(q, r), (f, r)\}$ та розриву зв'язку $C = C \setminus \{q, f\}$, якщо такий існує. Для новоутвореної ШНМ проводять модифікацію локальних похибок:

$$\begin{cases} E_q = E_q - \beta E_q, \\ E_f = E_f - \beta E_f, \\ E_r = (E_q + E_f)/2, \\ E_i = E_i - \gamma E_i \quad \text{при } i \neq q, i \neq f, i \neq r. \end{cases} \quad (4.219)$$

Критерієм завершення алгоритму є зростання кількості нейронів до деякої критичної межі M або досягнення критичної кількості ітерацій t_{\max} .

4.6. Методи навчання ШНМ типу ART

ART-мережі є найбільш подібними до біологічних об'єктів. Суть такої подібності полягає в здатності до реалізації процесів самоорганізації та до ефективного навчання без учителя. Ці процеси в ART-мережах зводяться до кластеризації вхідних образів та розвитку процедур категоризації шляхом використання адаптивного резонансу. Поняття „резонанс” у даному випадку визначає деякий резонансний стан нейронної мережі, що виникає тоді, коли вхідний вектор даних відповідає критерію близькості до вектора категоризованого прототипу. Величина цієї близькості є одним із параметрів, які можна змінювати в залежності від типу задачі. Такий резонанс називають адаптивним, оскільки в ході навчання нейронної мережі типу ART можуть бути модифікованими як вектор прототипу, так і загальна кількість кластерів.

4.6.1. ART-подібні нейронні мережі

Поняття про функціональну еквівалентність підсистеми навчання у вигляді мереж прямого поширення і класичної

структури з прямим та зворотним поширенням узагальнено в роботі [28]. Використовуючи це узагальнення, можна зробити висновок, що властивість адаптивного резонансу поширюється на нейронні мережі, які складаються з підсистеми навчання та підсистеми керування. В ролі підсистеми навчання можуть використовуватись одношарові нейронні мережі прямого поширення з навчанням без учителя, які застосовуються для розпізнавання образів і реалізують мету навчання, яку задають виразом:

$$\delta(t) = \arg \max_{j=1, \dots, c(t)} \left\{ f[\mathbf{X}(t), \mathbf{W}_j(t)] \right\}, \quad (4.220)$$

де $\mathbf{X}(t) = (x_1(t), x_2(t), \dots, x_i(t), \dots, x_d(t))$ — d -вимірний вектор вхідного сигналу, що надходить на вхід ШНМ в момент часу t , $x_i(t) \in [0, 1]$ — для бінарних мереж і $x_i(t) \in \mathbf{R}$ — для аналогових мереж; $\mathbf{W}_j(t) = (w_{1j}(t), w_{2j}(t), \dots, w_{dj}(t))$ — d -вимірний вектор шаблону або прототип кластера; $c(t)$ — розмір нейронної мережі, який може залежати від часу.

Підсистема керування — це інтерфейсний блок, який поєднує навчання з учителем із навчанням без учителя. Якість розпізнавання вхідного образу, який поширюється в прямому напрямку (аспект навчання без учителя), порівнюється з шаблоном кластера, що поширюється в зворотному напрямку і є заданим зовнішнім середовищем (аспект навчання з учителем). Саме у випадку співпадіння цих векторів виникає явище „резонансу”. Це означає, що в процесі навчання нейронної мережі відбулася адаптація прототипу таким чином, що він почав відображати властивості зовнішнього середовища.

Модульна архітектура ART-подібних нейронних мереж показана на рис.4.18.

Підсистема навчання даної архітектури представлена двома шарами. F -шар (Feature Representation Field) складається з вхідних нейронів $F_i, i=1, \dots, d$, де d — розмірність вхідного

простору. E -шар (Exemplar Representation Field) містить змінну кількість нейронів $c(t)$, що позначаються $E_j, j=1, \dots, c$.

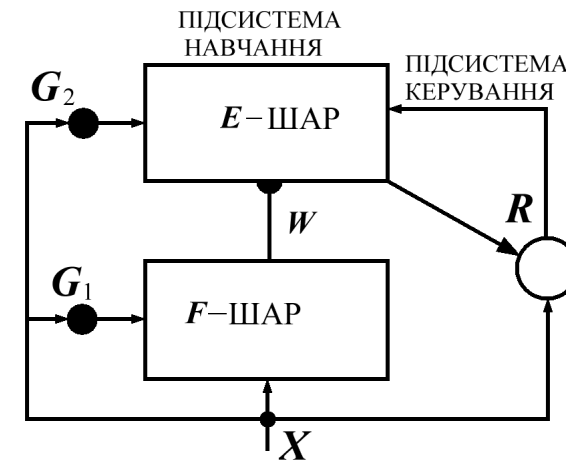


Рис.4.18. Архітектура ART-подібних мереж.

Елементи цього шару, в залежності від типу задач, можуть мати назву вихідні вузли, компоненти, категорії або кластери.

Алгоритм функціонування ART-подібних нейронних мереж починається з присвоєння нульових значень змінним c і t . При надходженні на вхід нейронної мережі чергового вектора вхідного образу $\mathbf{X}(t)$ значення лічильника образів збільшується на одиницю: $t = t + 1$. Активація нейронів F -шару, яка спричинена надходженням вхідного образу, визначається за формулою:

$$a(t) = f(\mathbf{X}(t), \mathbf{W}_j(t)), \quad j=1, \dots, c \quad (4.221)$$

де $f(\mathbf{X}(t), \mathbf{W}_j(t))$ — однонаправлена функція активації, яка має назву функції вибору. Вона задає ступінь подібності вектора $\mathbf{X}(t)$ до вектора $\mathbf{W}_j(t)$, але не навпаки.

Якщо деяка група нейронів була активізованою вектором вхідного образу, то говорять, що ця група утворює резонансний домен. Оскільки в ART-подібних мережах реалізований жорсткий конкурентний механізм, то серед можливих прототипів вибирається лише один, той, що максимально наближений до вхідного образу. Пошук нейрона, який містить необхідний прототип, відбувається за формулою:

$$\delta = \arg \max_{j=1, \dots, c} \left\{ f[\mathbf{X}(t), \mathbf{W}_j(t)] \right\}. \quad (4.222)$$

Вибраний прототип є ключем до формування резонансного домену, елементи якого відбирають з урахування порога чутливості:

$$\varphi(\mathbf{W}_\delta(t), \mathbf{X}(t)) \leq \rho, \quad (4.223)$$

де $\rho \in [0, 1]$ — поріг чутливості, що задають на етапі налаштування нейронної мережі на конкретну задачу;

$\varphi(\mathbf{W}_\delta(t), \mathbf{X}(t))$ — однонаправлена порівняльна функція, що задає ступінь подібності вектора $\mathbf{W}_j(t)$ до вектора $\mathbf{X}(t)$, але не навпаки.

Процес формування резонансного домену ще називають тестом чутливості або тестом гіпотези.

У випадку негативного завершення тесту чутливості стан „резонансу” не настає, а підсистема керування видає команду скиду. Ця команда вилучає прототип, який не пройшов перевірку на чутливість, із подальшого розгляду.

Наступний крок алгоритму полягає у виборі чергового активізованого нейрона відповідно до (4.222). Для вибраного нейрона виконується тест чутливості. Якщо даний нейрон також не пройшов тесту чутливості, то процес повторюється, доки не будуть проаналізовані всі активізовані нейрони E -шару. У

випадку позитивного завершення тесту чутливості виникає стан „резонансу”, який дозволяє підсистемі навчання модифікувати вибраний прототип. Така модифікація відбувається відповідно до встановлених правил і має на меті наблизити вектор прототипу до вектора поточного вхідного образу $\mathbf{X}(t)$.

Якщо множина резонансного домену пуста, то стан „резонансу” не настає. У цьому випадку динамічно збільшується кількість нейронів E -шару, що супроводжується відповідним нарощуванням параметра: $c = c + 1$.

Новий прототип відповідає вектору вхідного образу, значення якого не було адаптованим ні одним із існуючих прототипів. Описаний процес повторюється після подання на вхід нейронної мережі чергового вектора вхідного образу.

4.6.2. Нейронна мережа Fuzzy ART

Fuzzy ART — це нейронна мережа, функціонування якої відповідає алгоритму, описаному для ART-подібних нейронних мереж. Принциповою відмінністю є застосування багатозначної (fuzzy) логіки [52, 182].

У нейронних мережах Fuzzy ART першим кроком обробки вхідного образу є його нормалізація, що спричинено необхідністю обмеження зростання кількості прототипів. Один із способів нормалізації [52]:

$$\mathbf{X}(t) = (X_1(t), \dots, X_d(t)) = \frac{\mathbf{Z}(t)}{\|\mathbf{Z}(t)\|}, \quad (4.224)$$

де $\mathbf{Z}(t)$ — початкове значення вхідного образу, а

$$\|\mathbf{Z}(t)\| = \sqrt{(Z_1(t))^2 + \dots + (Z_d(t))^2}.$$

Існує також спосіб нормалізації за допомогою кодів доповнення [107]:

$$\begin{aligned} \mathbf{X}(t) &= (X_1(t), \dots, X_d(t)) = (\mathbf{Z}(t), \mathbf{Z}_c(t)) = \\ &= (Z_1(t), \dots, Z_q(t), Z_{1,c}(t), \dots, Z_{q,c}(t)), \end{aligned} \quad (4.225)$$

де $\mathbf{Z}(t) \in \mathbf{R}^q$, $d = dq$, $\mathbf{Z}_{k,c} = 1 - \mathbf{Z}_k(t)$, $k = 1, \dots, q$, $|\mathbf{X}(t)| = q$,

$$|\mathbf{X}(t)| = \sum_{k=1}^d \mathbf{X}_k(t).$$

Використання нормалізації з кодами доповнення дозволяє уникнути втрати інформації від нормалізації та ввести оцінку мінімальних та максимальних величин для кожного кластера і розмірності. Таким чином, для нейронних мереж Fuzzy Art прототипи можна представити у вигляді деяких замкнутих гіперповерхонь у просторі вихідних даних. Кожна гіперповерхня починає розвиватися з однієї точки, з часом вона зростає і досягає розмірів, які обмежуються порогом чутливості. Введення нормалізації також запобігає „зацикленню” прототипів.

Функція активації вхідного шару нейронів для даного випадку матиме вигляд:

$$f(\mathbf{X}(t), \mathbf{W}_j(t)) = \frac{\sum_{k=1}^d \min\{\mathbf{X}_k(t), \mathbf{W}_{kj}(t)\}}{\alpha + \sum_{k=1}^d \mathbf{W}_{kj}(t)}, \quad (4.226)$$

$$\mathbf{X}_k(t), \mathbf{W}_{kj}(t) \in [0, 1],$$

де $j = 1, \dots, c$ — поточний номер нейрона E -шару;

α змінюється в діапазоні $[0.001, 1]$.

Порівняльна функція:

$$\varphi(\mathbf{W}_\delta(t), \mathbf{X}(t)) = \frac{\sum_{k=1}^d \min\{\mathbf{W}_{k\delta}(t), \mathbf{X}_k(t)\}}{\sum_{k=1}^d \mathbf{X}_k(t)}, \quad \mathbf{X}_k, \mathbf{W}_{k\delta} \in [0, 1]. \quad (4.227)$$

Для бінарних нейронних мереж Fuzzy ART порівняльна функція має вигляд:

$$\varphi(\mathbf{X}(t), \mathbf{W}_\delta(t)) = \frac{\sum_{k=1}^d \mathbf{X}_k(t), \mathbf{W}_{k\delta}(t)}{\sum_{k=1}^d \mathbf{X}_k(t)}, \quad (4.228)$$

$$\mathbf{X}_k, \mathbf{W}_{k\delta} \in [0, 1].$$

Після підстановки в (4.228) параметрів нормалізації можемо отримати геометричну інтерпретацію:

$$\begin{aligned} \varphi(\mathbf{X}(t), \mathbf{W}_\delta(t)) &= \frac{\mathbf{X}(t) \times \mathbf{W}_\delta(t)}{\|\mathbf{X}_k(t)\|^2} = \frac{\|\mathbf{X}(t)\| \cdot \|\mathbf{W}_\delta(t)\| \cos \theta_\delta}{\|\mathbf{X}_k(t)\|^2} = \\ &= \frac{\|\mathbf{W}_\delta(t)\| \cos \theta_\delta}{\|\mathbf{X}_k(t)\|} \in [0, 1], \quad \mathbf{X}_k, \mathbf{W}_{k\delta} \in [0, 1], \end{aligned} \quad (4.229)$$

де " \times " — скалярний добуток; θ_δ — кут між $\mathbf{X}(t)$ та $\mathbf{W}_\delta(t)$. Порівняльна функція для даного випадку дорівнює проекції $\mathbf{W}_\delta(t)$ на $\mathbf{X}(t)$.

Правило навчання для Fuzzy ART визначає порядок адаптації прототипу до вхідного образу, який пройшов тест чутливості. У більшості випадків його задають виразом:

$$\begin{aligned} \mathbf{W}_{k,\delta}(t) &= (1 - \beta) \mathbf{W}_{k,\delta} + \beta \min\{\mathbf{X}_k(t), \mathbf{W}_{k,\delta}(t)\}, \\ k &= 1, \dots, d, \end{aligned} \quad (4.230)$$

де $\beta \in [0, 1]$ — коефіцієнт навчання.

З (4.230) видно, що лише прототип-переможець $\mathbf{W}_{k,\delta}$ може бути модифікований вектором вхідного образу $\mathbf{X}(t)$, що дозволяє класифікувати Fuzzy ART як конкурентну нейронну мережу. За умови нормалізації вхідних образів (4.227) і $\beta = 1$ прототип у найпростіший спосіб відображає властивості вхідних даних.

Експлуатація нейронних мереж Fuzzy ART показала, що вони мають ті ж недоліки, які характерні і для мереж ART-1.

Головними з них є підвищена чутливість до шуму та викидів у вхідному потоці даних, неефективне формування структури прототипів та залежність структури прототипів від порядку подання даних на вхід нейронної мережі.

З метою підвищення ефективності роботи нейронних мереж, що функціонують із використанням принципу адаптивного резонансу, запропоновано цілий ряд нових архітектур.

4.6.3. Нейронні мережі S-Fuzzy ART та SART

Нейронна мережа S-Fuzzy ART (симетрична багатозначна ART) є модифікацією Fuzzy ART, яка характеризується симетричністю функцій порівняння та активації [122]. Завдяки такій симетрії взаємодія між вхідним образом та прототипом може бути описана лише однією функцією:

$$\psi(X, W): D^d \times D^d \rightarrow [0, 1]. \quad (4.231)$$

У випадку, коли $X = W$, функція $\psi(X, X)$ досягає максимуму і є мінімальною при мінімальному співпадінні. Властивість симетрії цієї функції поширюється на весь діапазон вхідних образів та прототипів:

$$\psi(X, W) = \psi(W, X), \quad \forall X, W \in D^d. \quad (4.232)$$

Вона може бути утворена, наприклад, шляхом об'єднання (4.226) та (4.227):

$$f(X, W_j) = \varphi(X, W_j) = \psi(X, W_j) = \frac{\left(\sum_{k=1}^d \min\{X_k, W_{k,j}\} \right)^2}{\sum_{k=1}^d X_k \sum_{k=1}^d W_{k,j}} \propto \frac{\sum_{k=1}^d \min\{X_k, W_{k,j}\}}{\sqrt{\sum_{k=1}^d X_k \sum_{k=1}^d W_{k,j}}} \in [0, 1], \quad (4.233)$$

де $X, W \in D^d$.

Властивість симетрії функцій активації та порівняння дозволила поліпшити точність формування кластерів і підвищити живучість мережі на визначених наборах даних.

Для розширення можливостей мереж типу ART запропоновано ряд нових підходів [28,64], які полягають у застосуванні принципу абсолютної та відносної приналежності до прототипу як для бінарних, так і для аналогових даних. Такі мережі отримали назву SART (спрощені ART).

Нехай вектор вхідного образу $X(t)$ належить до аналогового простору R^d , де d — розмірність вхідного простору, а кластерна структура $W_j, j=1, \dots, c(t)$, належить до простору параметрів R^p при $p \geq d$.

У випадку застосування нормалізації принцип абсолютної приналежності задають відображенням:

$$A(X, W): R^d \times R^p \rightarrow [0, 1], \quad (4.234)$$

яке є симетричним у випадку $p = d$. При $X = W$ значення $A(X, X)$ досягає максимуму і, відповідно до властивості симетричності,

$$A(X, W) = A(W, X), \quad X \in R^d, W \in R^d. \quad (4.235)$$

Принцип відносної приналежності до однієї кластерної структури $W_j(t)$, яка є елементом вектора

$W(t) = (W_1(t), \dots, W_{c(t)}(t))$, задають відображенням:

$$P_j(X(t), W_j(t)): R_1^p \times R_2^p \times \dots \times R_{c(t)}^p \times R^d \rightarrow [0, 1], \quad (4.236)$$

таким чином, що

$$\sum_{j=1}^{c(t)} P_j(X(t), W_j(t)) = 1. \quad (4.237)$$

Функція $P_j(X(t), W(t))$ є монотонно незростаючою для відстані $d(X(t), W_j(t))$ і монотонно неспадною для

$$d(X(t), W_h(t)), \quad h=1, \dots, c(t), \quad h \neq j.$$

При $p=d$ таке відображення також має бути симетричним:

$$P_j(X(t), W(t)) = P_j(W(t), X(t)). \quad (4.238)$$

Це означає, що існує взаємно однозначне співвідношення між векторами кластерних параметрів і точками вхідного простору.

Для даного абсолютного відображення можна побудувати відносно відображення, виходячи з виразу:

$$P_j(X(t), W(t)) = \frac{A(X(t), W_j(t))}{\sum_{k=1}^{c(t)} A(X(t), W_k(t))}, \quad j=1, \dots, c(t) \quad (4.239)$$

Отже, відносна приналежність вхідного образу до того чи іншого прототипу залежить від його абсолютної приналежності до кожного з кластерів, які сформовані на нейронах Е-шару. Тому кожний нейрон цього шару стає контекстно-залежним, що дає можливість використовувати процеси міжнейронної взаємодії в ході моделювання задач із відносною приналежністю.

Негативна сторона такого підходу проявляється у випадку наявності шуму та викидів у вхідному сигналі. Тоді помилкова інформація розповсюджується на всі кластери і може призвести до значної їх деформації.

4.6.4. Нейронна мережа FOSART

На основі технології SART створено нейронну мережу FOSART (fully self-organizing simplified adaptive resonance theory) [44], яка реалізує ідентифікацію вхідного образу шляхом мінімізації цільової функції:

$$E = \sum_{i=1}^n \sum_{j=1}^c \|X_i - W_j\|^2 k_j(d(X_i, W)), \quad (4.240)$$

де $d(X_i, W) = (d(X_i, W_1), \dots, d(X_i, W_c))$ — множина відстаней між вхідним образом X_i та кожним із прототипів, які в сукупності утворюють вектор $W = (W_1, \dots, W_c)$.

Множник $k_j(d(X_i, W))$ називають функцією ядра [24].

Необхідна умова мінімізації цільової функції [30]:

$$\frac{\partial E}{\partial W_j} = -2 \sum_{i=1}^n (X_i - W_j) k_j(d(X_i, W)) + \sum_{i=1}^n \sum_{h=1}^c (X_i - W_h)^2 \frac{\partial k_h(d(X_i, W))}{\partial W_j} = 0, \quad (4.241)$$

де $j=1, \dots, c$.

Прирівнявши другий доданок (4.241) до нуля, одержимо:

$$W_j(\tau+1) = \frac{\sum_{i=1}^n X_i k_j(d(X_i, W(\tau)))}{\sum_{g=1}^c k_j(d(X_g, W(\tau)))}, \quad j=1, \dots, c. \quad (4.242)$$

Рекурсивна формула методу навчання для нейронної мережі FOSART [30]:

$$\begin{aligned} W_j(\tau+1) &= W_j(\tau) - \beta(\tau) \frac{\partial E}{\partial W_j} = \\ &= W_j(\tau) + \beta(\tau) \sum_{i=1}^n (X_i - W_j(\tau)) k_j(d(X_i, W(\tau))), \end{aligned} \quad (4.243)$$

де $\beta(\tau)$ — коефіцієнт навчання.

Для збіжності процесу (4.243) коефіцієнт навчання $\beta(\tau)$ повинен задовольняти такі умови:

$$\lim_{\tau \rightarrow \infty} \beta(\tau) = 0, \quad \sum_{\tau=1}^{\infty} \beta(\tau) = \infty, \quad \sum_{\tau=1}^{\infty} \beta^2(\tau) < \infty.$$

Алгоритм роботи мережі складається з типових для ART-мереж кроків.

1. Ввід параметрів. Для правильного функціонування мережі необхідно ввести ряд параметрів, серед яких величина коефіцієнта чутливості $\rho \in [0,1]$, мінімальна кількість епох для існування кожного нейрона $\tau_{\min} \geq 1$ тощо.

2. Завантаження вхідного образу. На цьому етапі на вхід нейронної мережі подають черговий вектор вхідного образу $X(t)$ зі збільшенням лічильника $t = t + 1$.

3. Спрацьовування активаційних функцій та пошук найближчих прототипів. Результатом роботи етапу є визначення номера нейрона, який містить максимально наближений прототип, та нейрона, який посідає друге місце відповідно до критерію близькості:

$$\begin{aligned} \delta_1(t) &= \arg \max_{j=1, \dots, c(t)} \{ \psi(X(t), W_j(t)) \} = \\ &= \arg \min_{j=1, \dots, c(t)} \{ \|X(t) - W_j(t)\| \}, \end{aligned} \quad (4.244)$$

$$\begin{aligned} \delta_2(t) &= \arg \max_{\substack{j=1, \dots, c(t) \\ j \neq \delta_1(t)}} \{ \psi(X(t), W_j(t)) \} = \\ &= \arg \min_{\substack{j=1, \dots, c(t) \\ j \neq \delta_1(t)}} \{ \|X(t) - W_j(t)\| \}, \end{aligned} \quad (4.245)$$

де функція активації співпадає з порівняльною функцією і обчислюється в такий спосіб:

$$\psi(X(t), W_j(t)) = \exp \left(- \frac{\|X(t) - W_j(t)\|^2}{\sigma^2} \right) \in [0,1].$$

4. Формування резонансного домену (тест чутливості). У випадку, виконується тест чутливості $\psi(X(t), W_{\delta_1(t)}(t)) \geq \rho$ вважають, що наявним є стан „резонансу”, і переходять до пункту 4.1. У протилежному випадку виконують пункт 4.2.

4.1. Навчання мережі в стані „резонансу”.

4.1.1. Лічильник резонансів r_{δ_1} для нейрона E_{δ_1} збільшують на одиницю: $r_{\delta_1}(t+1) = r_{\delta_1}(t) + 1$.

4.1.2. Якщо між нейронами E_{δ_1} та E_{δ_2} існує літеральний зв'язок $L_{\delta_1, \delta_2}(t)$, то модифікують лічильник резонансів цього зв'язку: $r_{\delta_1, \delta_2}(t+1) = r_{\delta_1, \delta_2}(t) + 1$.

4.1.3. Якщо між нейронами E_{δ_1} та E_{δ_2} не існує літерального зв'язку, то такий зв'язок $L_{\delta_1, \delta_2}(t)$ генерують і встановлюють його лічильник резонансів: $r_{\delta_1, \delta_2}(t+1) = 1$.

4.1.4. Модифікують параметри прототипу W_{δ_1} і літерально пов'язаних прототипів, використовуючи ітераційний процес (4.243).

4.2. Генерування нового прототипу за відсутності „резонансу”.

4.2.1. Нарощують лічильник прототипів: $c(t+1) = c(t) + 1$.

4.2.2. Активізують додатковий нейрон E - шару: $E_{\delta(t+1)}(t+1)$.

4.2.3. Формують новий прототип: $W_{\delta(t+1)}(t+1) = X(t)$.

4.2.4. Встановлюють лічильник епох: $\tau_{c(t+1)}(t+1) = 0$.

4.2.5. Встановлюють лічильник резонансів $r_{c(t+1)}(t+1) = 1$.

5. Перехід до нової епохи. Після повної обробки вхідного набору даних відбувається перехід до нової епохи, який супроводжується такими діями.

5.1. Вилучення зайвих прототипів. Перевіряють значення лічильника резонансів для кожного нейрона E -шару. Якщо

$r_j(t) = 0$, то це означає, що за останню епоху нейрон $E_j(t)$, який містить прототип $W_j(t)$, не утворив резонансного стану ні з одним із вхідних образів. Такий нейрон вилучають із подальшого розгляду, а лічильник прототипів зменшують на одиницю: $c(t+1) = c(t) - 1$.

5.2. Вилучення зайвих літеральних зв'язків. Розглядають попарно нейрони E -шару: $\forall i \in \{1, c(t)\}, \forall j \in \{1, c(t)\}, i \neq j$. Якщо для пари (i, j) існує літеральний зв'язок $L_{i,j}(t)$, то проводять перевірку лічильника резонансів $r_{ij}(t)$. Умовою вилучення зв'язку $L_{i,j}(t) \in r_{ij}(t) = 0$.

5.3. Нарощування лічильника епох. Для нейронів $E_j(t)$, що залишилися після вилучень за умовами 5.1 та 5.2, проводять нарощування лічильника епох: $c(t+1) = c(t) + 1$.

5.4. Скидання лічильників резонансів. Для нейронів $E_j(t)$, $j = 1, \dots, c(t)$ встановлюють $r_j(t) = 0$. Для всіх літеральних зв'язків $L_{i,j}(t)$, $i \in \{1, c(t)\}, j \in \{1, c(t)\}, i \neq j$ встановлюють $r_{ij}(t) = 0$.

6. Перевірка закінчення алгоритму. Якщо лічильник епох $\tau_j(t) \geq \tau_{\min}$, $j = 1, \dots, c(t)$, то припиняють роботу алгоритму. У протилежному випадку переходять до кроку 1.

Застосування FOSART до розв'язання задач класифікації, векторного квантування та перцепційного групування показали її високу стабільність у порівнянні з іншими розглянутими нейронними мережами типу ART.

Розділ 5

ІТЕРАЦІЙНІ МЕТОДИ НА НЕЙРОННИХ МЕРЕЖАХ

5.1. Принципи взаємодії нейронів при реалізації ітераційних методів

Ітераційні методи широко застосовуються в науці і техніці, наприклад, в медицині, моделюванні електричних кіл, комунікаційних мереж та фінансових моделях. Метою більшості таких алгоритмів є пошук нерухомої точки лінійних або нелінійних операторів. У випадку значних розмірностей та складних обчислень, що є характерними для задач реального світу, реалізація ітераційного методу потребує значних обчислювальних ресурсів. Природним вирішенням цієї проблеми є створення паралельних ітераційних методів. Але ефективність таких методів не можна оцінювати окремо від засобів їх реалізації. Сам принцип паралелізму допускає наявність деякої кількості елементів обробки інформації (процесорних модулів або нейронів), що поєднані каналами зв'язку. У випадку, коли оброблювана інформація має дискретний характер, продуктивність паралельної структури не є загальною сумою продуктивностей окремих її складових. Така оцінка може бути зроблена лише у випадку ідеального паралельного алгоритму, що зумовлює однакове та повне завантаження всіх елементів структури. Реальні алгоритми характеризуються неоднорідними об'ємами обробки інформації, що, в свою чергу, породжує різний ступінь завантаження каналів зв'язку. Тому одним із найважливіших факторів, які впливають на продуктивність паралельних методів, є взаємодія між елементами паралельної структури. Процес взаємодії завжди залежить від трьох таких елементів: передавача, приймача та каналу зв'язку. У нейронній мережі передавачами та приймачами є нейрони, які з'єднані між собою синаптичними

зв'язками. Для досягнення високих показників взаємодії необхідно використовувати елементи, які характеризуються високою продуктивністю. Але не тільки продуктивність елементів архітектури впливає на взаємодію. Вона в значній мірі також залежить від процедури проведення, тобто від послідовності дій всіх учасників цього процесу. Якщо для проведення взаємодії необхідна одночасна готовність всіх учасників, то таку взаємодію називають синхронною взаємодією. Асинхронна взаємодія потребує готовності тільки деякої підмножини учасників процесу. Загальну структуру форм взаємодії у паралельних методах показано на рис.5.1.

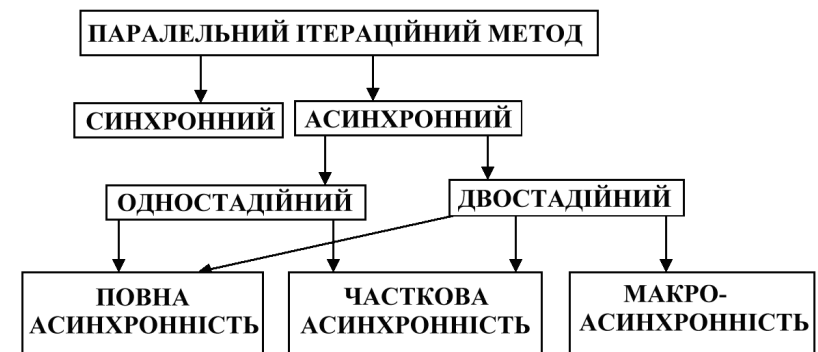


Рис.5.1. Структура форм взаємодії у паралельних ітераційних методах.

Паралельні ітераційні методи, що використовують синхронну взаємодію елементів архітектури, називають синхронними паралельними ітераційними методами. Відповідно, для асинхронних паралельних ітераційних методів необхідна асинхронна взаємодія елементів архітектури. Синхронні паралельні ітераційні методи, у більшості випадків, можуть бути одержані зі звичайних послідовних ітераційних методів шляхом їх розпаралелювання. Привабливість такого підходу полягає у тому, що збіжність паралельного синхронного методу легко

може бути доведеною, оскільки вона завжди впливає зі збіжності його послідовного аналога.

Нехай штучна нейронна мережа складається з одного шару n нейронів, що мають зв'язки кожного з кожним, як показано на рис.5.2, та з керуючого нейрона C .

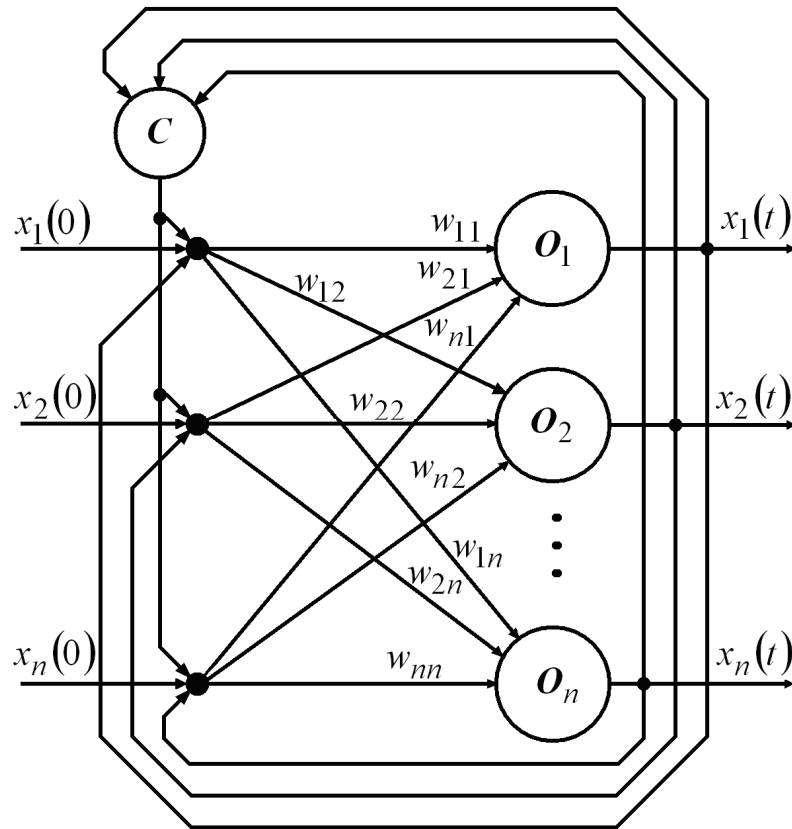


Рис.5.2. Синхронна ШНМ.

Ітераційний процес у такій мережі починається з надходження на вхід вектора даних $\mathbf{x}(0) = (x_1(0), x_2(0), \dots, x_n(0))$. Кожен нейрон реалізує свою частину O_i загального оператора O ,

використовуючи свої дані та ті, які були одержані від інших нейронів. Позначивши вектор даних, який відповідає поточній ітерації з номером t через $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$, а дію компоненти оператора $O(\mathbf{x}(t), \mathbf{W})$ через

$$O_i(x_1(t), x_2(t), \dots, x_n(t), w_{1i}, w_{2i}, \dots, w_{ni}),$$

формулу синхронного ітераційного процесу на нейроні O_i представимо у вигляді:

$$x_i(t+1) = O_i(x_1(t), x_2(t), \dots, x_n(t), w_{1i}, w_{2i}, \dots, w_{ni}), \quad (5.1)$$

$$i = 1, \dots, n, \quad t \in N$$

Очевидно, що у загальному випадку синхронний ітераційний процес містить дві стадії обробки інформації: обчислювальну та комунікаційну. Ці стадії завжди ідуть одна за одною і починаються синхронно у всіх нейронах під дією сигналу керуючого нейрона. Обчислювальна стадія полягає в обчисленні кожним із нейронів відповідної компоненти оператора $O(\mathbf{x}(t))$, а на комунікаційній стадії відбувається передача на входи всіх нейронів результатів обчислень кожної з компонент.

Часову діаграму синхронного ітераційного процесу показано на рис.5.3.



Рис.5.3. Часова діаграма синхронного ітераційного процесу.

Вона відображає ситуацію, коли час обробки компонент оператора $\mathcal{O}(W, x(t))$ істотно відрізняється для кожного з нейронів. З цієї причини тривалість обчислювальної стадії ітераційного процесу дорівнює тривалості обчислювальної стадії найповільнішого нейрона. Очевидна неефективність організації синхронного процесу є однією з головних причин пошуку можливостей забезпечення асинхронної його реалізації.

Поняття асинхронної взаємодії при реалізації паралельних ітераційних методів не є однозначним. В залежності від принципів формування обчислювальної стадії розрізняють одностадійний та двостадійний асинхронні ітераційні процеси. Одностадійний асинхронний ітераційний процес за структурою ітерацій подібний до синхронного. У двостадійному асинхронному процесі розрізняють ітерації двох рівнів: локальні або внутрішні ітерації та глобальні або макроітерації.

Розглянемо спочатку принципи взаємодії нейронів при реалізації одностадійних асинхронних процесів, які характеризуються повним або частковим асинхронізмом. Архітектуру ШНМ з асинхронною взаємодією нейронів показано на рис.5.4. Одним із перших результатів, які заклали основи принципів асинхронної взаємодії, вважають роботу [54]. Останнім часом цей принцип набув подальшого розвитку в зв'язку з використанням у різних сферах паралельної обробки інформації. Основною відмінністю принципу асинхронної взаємодії від принципу синхронної взаємодії є те, що при його застосуванні немає необхідності оновлення всіх компонент вхідного вектора як умови початку чергової обчислювальної стадії. Така взаємодія є характерною для нейронних мереж.

Як і в синхронному випадку, асинхронний ітераційний процес містить обчислювальну та комунікаційну стадії обробки інформації. Оскільки принцип асинхронної взаємодії не потребує синхронізації початку цих стадій, то немає потреби у загальній синхронізації мережі, яку б виконував керуючий нейрон.

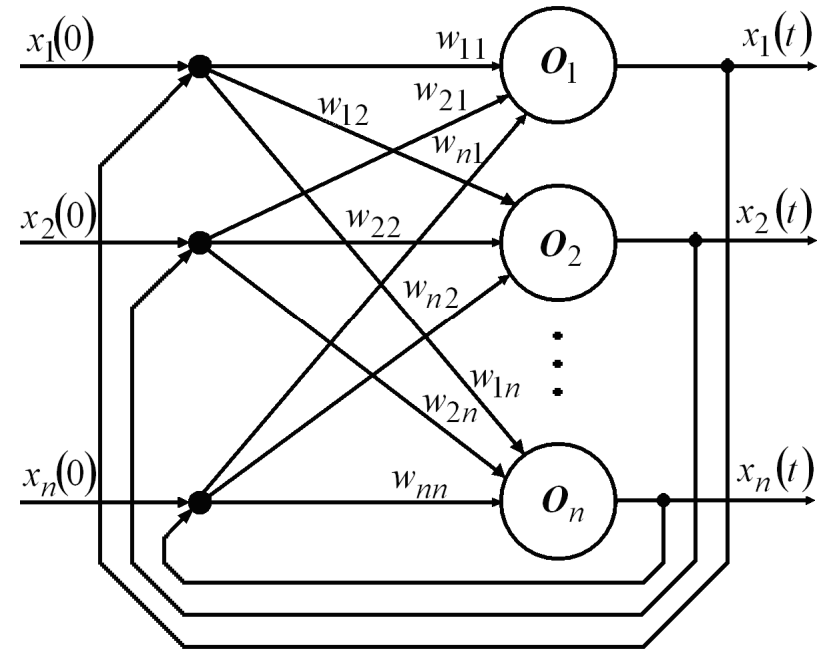


Рис.5.4. Асинхронна ШНМ.

Часову діаграму асинхронного ітераційного процесу показано на рис.5.5. Час обробки компонент оператора $\mathcal{O}(W, x(t))$, як і раніше, істотно відрізняється для кожного з нейронів, але початок комунікаційної стадії настає безпосередньо після закінчення його обчислювальної стадії. Отже, такий підхід дозволяє уникнути простоїв, і тому є більш ефективним при організації обчислювального процесу.

На рис.5.5. зображено часову діаграму з приблизно однаковими обчислювальними стадіями. Але на практиці часто виникають ситуації, коли обчислювальні стадії у різних нейронах можуть істотно відрізнятися.

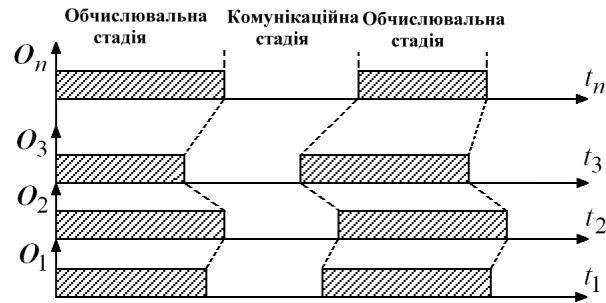


Рис.5.5. Часова діаграма асинхронного ітераційного процесу.

На рис.5.6 показано часову діаграму для таких процесів. У даному випадку послідовно $\tau = 12$ ітерацій процесу O_1 використовують результати першої ітерації процесу O_2 .

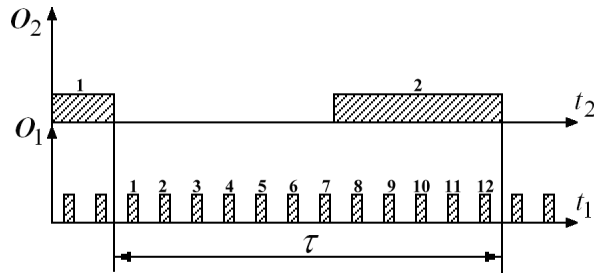


Рис.5.6. Часова діаграма обчислювальних процесів з різними періодами.

Асинхронний ітераційний метод, для якого величина параметра τ є фіксованою, називають ітераційним методом з частковою асинхронністю. Якщо величина τ не регламентується, тобто не існує обмежень на глибину використання даних, то такий асинхронний ітераційний метод називають методом з повною асинхронністю.

5.2. Одностадійні асинхронні ітераційні методи

5.2.1. Базові поняття

Розглядатимемо паралельні ітераційні методи для розв'язання систем лінійних рівнянь типу

$$Ax = b, \quad (5.2)$$

де A — матриця розмірності $n \times n$ з додатними симетричними коефіцієнтами $a_{i,j} \in \mathbf{R}$, $i, j = \overline{0, n-1}$; $x, b \in \mathbf{R}^n$.

Позначимо через D діагональну матрицю, для якої

$$d_{ii} = a_{ii}, \quad E = D - A.$$

Сформуємо ітераційний метод $x = D^{-1}Ex + D^{-1}b$, виходячи з визначення нерухомої точки для (5.2). Позначивши $B = D^{-1}E$ та $C = D^{-1}$, одержимо

$$x = Bx + Cb. \quad (5.3)$$

Оскільки B, C і b задані, то оператор F має вигляд:

$$F(x) = Bx + Cb. \quad (5.4)$$

Будемо говорити, що матриця B є невід'ємною $B \geq 0$, якщо всі її елементи b_{ij} невід'ємні. Модуль матриці $|B| \in \mathbf{R}^n$ також задають через модулі її елементів:

$$|B|_{ij} = |b_{ij}|, \quad i, j = 1, 2, \dots, n. \quad (5.5)$$

Позначимо через $\rho(B)$ спектральний радіус матриці B , тобто найбільший із модулів власних значень даної матриці.

Означення 5.1. Нормований лінійний простір — це векторний простір X з нормою $\|x\|$, що задовольняє умови:

- 1) $\|x\| \geq 0 \quad \forall x \in X$,
- 2) $\|x\| = 0$ тоді і тільки тоді, коли $x = 0$,
- 3) $\|x + y\| \leq \|x\| + \|y\|$,

$$4) \|\alpha x\| = |\alpha| \cdot \|x\|.$$

Означення 5.2. Метричний простір — це пара (X, d) , яка складається з деякої множини (простору) точок X і відстані, яка однозначно задана невід'ємною дійсною функцією $d(x, y)$ і для неї виконуються такі аксіоми:

$$1) d(x, y) = 0 \quad \text{тоді і тільки тоді, коли } x = y, \quad (5.6)$$

$$2) d(x, y) = d(y, x) \quad (\text{аксіома симетрії}), \quad (5.7)$$

$$3) d(x, z) \leq d(x, y) + d(y, z) \quad (\text{аксіома трикутника}). \quad (5.8)$$

Для спрощення метричний простір (X, d) позначають однією буквою $R = (X, d)$, а функцію $d(x, y)$ називають метрикою простору R .

Якщо простір складається з n точок, то функція відстані для них може бути заданою, наприклад, такими способами:

$$1) d(x, y) = \|x - y\|_1 = \sum_{i=1}^n |x_i - y_i|.$$

$$2) d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}.$$

$$3) d(x, y) = \|x - y\|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|.$$

Означення 5.3. Простір R називають повним метричним простором, якщо у ньому збігається довільна фундаментальна послідовність.

Послідовність $\{x_n\}$ метричного простору R будемо називати фундаментальною, якщо вона задовольняє критерій Коші, тобто, якщо для довільного $\varepsilon > 0$ існує таке число $k \in N$, що $d(x_n, x_m) < \varepsilon \quad \forall m, n \geq k$.

Означення 5.4. Відображення A метричного простору R в себе $A: R \rightarrow R$ називають стискаючим відображенням, якщо існує таке число $\alpha < 1$, що для довільних двох точок $x, y \in R$ виконується нерівність

$$d(Ax, Ay) \leq \alpha d(x, y). \quad (5.9)$$

Теорема 5.1 (Теорема Банаха про нерухому точку).

Нехай R — повний метричний простір і $A: R \rightarrow R$ — стискаюче відображення. Тоді це відображення має одну і тільки одну нерухому точку в R . Точку x називають нерухомою точкою відображення A , якщо $Ax = x$.

Означення 5.5. Повний нормований простір R з нормою $\|\cdot\|$ називають банаховим простором або B -простором, якщо його метрика задана виразом $d(x, y) = \|x - y\|$. Згідно з [132], $(R, \|\cdot\|)$ та $(\|\cdot\|, R)$ є повними метричними та банаховими просторами.

5.2.2. Метод хаотичних ітерацій

Метод хаотичних ітерацій був уперше запропонований в [54].

Означення 5.6. Метод хаотичних ітерацій (F, J) — це клас послідовностей n -вимірних векторів $x(j)$, $j = 0, 1, 2, \dots$. Кожну з послідовностей цього класу визначають рекурсивно з виразу:

$$\begin{aligned} x_i(j+1) &= \\ &= \begin{cases} x_i(j), & i \neq k_{n+1}(j), \\ f_i(x_1(j - k_1(j)), \dots, x_n(j - k_n(j))), & i = k_{n+1}(j), \end{cases} \end{aligned} \quad (5.10)$$

де $x(0)$ — заданий початковий вектор.

Оператор $F: R^n \rightarrow R^n$ задають виразом

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x)).$$

Послідовність $J = (k_1(j), k_2(j), \dots, k_{n+1}(j))$, $j = 0, 1, 2, \dots$

містить $(n+1)$ -вимірні вектори. При деякому заданому цілому $s > 0$ ці вектори характеризуються такими властивостями:

а) $0 \leq k_i(j) < s$ при $i = 1, 2, \dots, n$ та $j = 1, 2, \dots$;

б) $1 \leq k_{n+1}(j) \leq n$ при $j = 1, 2, \dots$;

в) $k_{n+1}(j) = i$ необмежено часто для кожного i , $1 \leq i \leq n$.

Ітераційна схема (5.10) задає процес, у якому для кожного поточного моменту часу j відбувається модифікація компоненти з номером $k_{n+1}(j)$ вектора $\mathbf{x}(j)$ за умови незмінності решти компонент. Для модифікації компоненти $x_{k_{n+1}(j)}(j)$ використовують n компонент, значення яких були отримані на одному з s попередніх ітераційних кроків. Затримка для кожної з таких компонент $x_i(j - k_i(j))$ визначається часовим зсувом $k_i(j)$. Виходячи з властивостей хаотичних ітерацій, значення компонент, які були отримані пізніше, ніж за s кроків, не враховуються. Тому для одержання правильного результату необхідно забезпечити оновлення кожної з компонент вектора $\mathbf{x}(j)$. Оскільки послідовність такого оновлення може бути довільною, то очевидним стає той факт, що метод хаотичних ітерацій є узагальненням ітераційних методів із обумовленим порядком модифікації компонент. Задавши конкретний вид послідовності \mathbf{J} та величину s , можемо одержати конкретний ітераційний метод.

Наприклад, при

$$k_1(j) = k_2(j) = k_3(j) = \dots = k_n(j) = 0, \\ k_{n+1}(j) \equiv j \pmod{n} + 1$$

і $s=1$ одержимо $k_{n+1}(j) = i$ для $i=1, \dots, n$. Отже, при заданих умовах кожна компонента вектора ітерацій змінюється тільки один раз за n ітерацій, що відповідає методу Гауса–Зейделя.

Якщо виразити послідовність \mathbf{J} співвідношеннями

$$k_{n+1}(j) \equiv j \pmod{n} + 1, k_1(j) = k_2(j) = \dots = k_n(j) = [k_{n+1}(j) + 1],$$

а значення затримки покласти $s=n$, то також одержимо $k_{n+1}(j) = i$ для $i=1, \dots, n$. Але у цьому випадку ітераційна схема відповідає методу Якобі.

Умову збіжності методу хаотичних ітерацій задають наступною теоремою [54].

Теорема 5.2. Нехай (\mathbf{F}, \mathbf{J}) — метод хаотичних ітерацій, побудований за ітераційною схемою (5.10). Для того, щоб послідовність ітерацій над вектором $\mathbf{x}(j)$, виконаних за методом хаотичних ітерацій (\mathbf{F}, \mathbf{J}) , збігалася до розв'язку рівняння (5.2), необхідно і достатньо, щоб спектральний радіус матриці $|\mathbf{B}|$ був меншим за одиницю.

При доведенні умов необхідності та достатності цієї теореми застосовують лему 5.1.

Лема 5.1. Нехай матриця \mathbf{M} є невід'ємною з елементами $\{m_{ij}\}_{i,j=1}^n$. Для того, щоб спектральний радіус даної матриці був меншим за одиницю, необхідно і достатньо, щоб існував додатний вектор $\mathbf{v} \in \mathbf{R}^n$ і деяке дійсне число θ , $0 < \theta < 1$, які б задовольняли нерівність

$$\mathbf{M}\mathbf{v} \leq \theta\mathbf{v}. \quad (5.11)$$

Доведення умови достатності теореми 5.2. випливає з леми 5.1. Оскільки спектральний радіус $\rho(\mathbf{B}) < 1$, то існують додатний вектор $\mathbf{v} \in \mathbf{R}^n$ і додатне число $\theta < 1$ такі, що

$$|\mathbf{B}|\mathbf{v} \leq \theta\mathbf{v}. \quad (5.12)$$

Нехай $\mathbf{x} \in \mathbf{R}^n$ та $|\mathbf{x}| < \mathbf{v}$. Тоді справедлива нерівність

$$|\mathbf{B}\mathbf{x}| \leq |\mathbf{B}||\mathbf{x}| \leq |\mathbf{B}|\mathbf{v} \leq \theta\mathbf{v}, \quad (5.13)$$

яка підтверджує те, що матриця $|\mathbf{B}|$ є матрицею стискування і переводить паралелепіпед $\{\mathbf{x} : |\mathbf{x}| \leq \mathbf{v}\}$ в себе. Використовуючи цей факт, доводять, що послідовність $\{\mathbf{x}(j)\}_{j=1}^{\infty}$ прямує до нуля.

При доведенні умови необхідності теореми 5.2 показують, що якщо не існує такого \mathbf{v} , для якого виконується нерівність

$$|\mathbf{B}|\mathbf{v} \leq \alpha\mathbf{v}, \quad (5.14)$$

то існує така послідовність J_0 , для якої ітерації $\{x(j)\}_{j=1}^{\infty}$ розбігаються. Тоді з леми 5.1 випливає, що для даного випадку $\rho(|B|) \geq 1$. Отже, теорема 5.2 повністю доведена.

Розглянемо застосування методу хаотичних ітерацій для релаксації з параметром ω . Ітераційна схема для даного випадку має вигляд:

$$x = \omega(Bx + C) + (1 - \omega)x. \quad (5.15)$$

Звівши подібні члени, одержимо

$$x = (I - \omega D^{-1}A)x + \omega Cb. \quad (5.16)$$

Введемо матриці $B^\omega = I - \omega D^{-1}A$ і $C^\omega = \omega C$. Позначивши через (F^ω, J) метод хаотичної релаксації, наведемо умови його збіжності.

Теорема 5.3. Якщо метод хаотичної релаксації (F^ω, J) побудований за ітераційною схемою (5.10), то необхідні і достатні умови його збіжності:

$$\rho(|B|) < 1, \quad 0 < \omega < \frac{2}{1 + \rho(|B|)}.$$

В [132,133] розглянуто загальний випадок методу хаотичної ітерації, коли більше, ніж одна компонента вектора $x(j)$ може бути оновленою в кожний момент часу j , і оператор F є нелінійним.

Нехай $\Psi = \{J_j\}_{j=1}^{\infty}$ — послідовність непустих підмножин $J_j \subset \{1, 2, \dots, n\}$. Тоді ітераційну схему (5.10) перепишемо у вигляді:

$$x_i(j+1) = \begin{cases} x_i(j), & i \notin J_j, \\ f_i(x_1(j - k_1(j)), \dots, x_n(j - k_n(j))), & i \in J_j. \end{cases} \quad (5.17)$$

Властивості методу.

- 1) Оператор $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$, заданий у покоординатному вигляді $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$, може бути нелінійним;
- 2) $i \in \Psi$ нескінченну кількість разів;
- 3) $x_i(j)$ підлягає модифікації хоча б один раз на протязі s послідовних кроків.

Узагальнений метод хаотичних ітерацій задає обчислювальний процес, у якому на ітераційному кроці j всі компоненти вектора $x(j)$, номери яких входять у підмножину J_j , підлягають модифікації, а всі інші компоненти цього вектора залишаються незмінними. Модифікація компонент $x_i(j)$ з номерами $i \in J_j$ відбувається одночасно. Для цього використовують n компонент, значення яких були одержані на одному з s попередніх ітераційних кроків. Затримку для кожної з таких компонент визначають часовим зсувом $x_i(j - k_i(j))$, що має значення $k_i(j)$. Як і в попередньому випадку, значення компонент, які були отримані пізніше, ніж за s кроків, не враховуються.

Означення 5.7. Нехай задано відображення $F : D(F) \rightarrow B$, при $D(F) \subset B$ і $B = \prod_{i=1}^n B_i$, де $\{B_i\}_{i=1}^n$ — множина банахових просторів. Для кожної даної точки $v \in D(F)$ говорять, що оператор F є стискаючим у точку v по векторній нормі η , якщо:

- 1) існує невід'ємна матриця T розмірності $n \times n$, яка відповідає умові

$$\eta(F(v) - F(v)) \leq T\eta(v - v) \quad \forall v \in D(F), \quad (5.18)$$
- 2) спектральний радіус матриці T менший за одиницю: $\rho(T) < 1$.

Для $\mathbf{B}_i \in \mathbf{R}$, $i \leq i \leq n$, векторна норма має вигляд:

$$\|\mathbf{x}\| = (\|x_1\|, \dots, \|x_n\|) \quad \forall \mathbf{x} \in \mathbf{B} = \mathbf{R}^n.$$

Враховуючи це, перепишемо нерівність (5.18):

$$\|\mathbf{F}(\mathbf{v}) - \mathbf{F}(\mathbf{v}')\| \leq T \|\mathbf{v} - \mathbf{v}'\| \quad \forall \mathbf{v} \in \mathbf{R}^n. \quad (5.19)$$

Теорема про збіжність даного методу хаотичних ітерацій спирається на (5.19).

Теорема 5.4. Нехай оператор \mathbf{F} має фіксовану точку $\mathbf{x}^\circ \in \mathbf{D}(\mathbf{F})$ і стискає в \mathbf{x}° для векторної норми η . Тоді ітераційна послідовність $\{\mathbf{x}(j)\}_{j=1}^\infty$, що задана ітераційною схемою (5.17), збігається в нерухому точку \mathbf{x}° .

Висновок. Нехай оператор $\mathbf{F} : \mathbf{R}^n \rightarrow \mathbf{R}^n$ має форму $\mathbf{F}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) + \mathbf{C}$. Якщо $\rho(|\mathbf{B}|) < 1$, то \mathbf{F} має єдину нерухому точку \mathbf{x}° і стискає в цю точку для векторної норми η .

Д о в е д е н н я. Розглянемо нормований простір $(\mathbf{R}^n, \|\cdot\|_\omega)$ з векторною нормою

$$\|\mathbf{x}\|_\omega = \max_{i=1, n} \left| \frac{x_i}{\omega_i} \right|, \quad \omega_i > 0. \quad (5.20)$$

Згідно з [62] такий простір є повним метричним простором. Тому для доведення того, що оператор \mathbf{F} має єдину нерухому точку \mathbf{x}° , достатньо показати, що він є стискаючим на \mathbf{R}^n по нормі $\|\cdot\|_\omega$.

Нехай $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$. Тоді

$$\eta(\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{y})) = |\mathbf{B}(\mathbf{x} - \mathbf{y})| \leq |\mathbf{B}| \|\mathbf{x} - \mathbf{y}\| = \mathbf{B}\eta(\mathbf{x} - \mathbf{y}). \quad (5.21)$$

В [132] показано, що оскільки матриця $|\mathbf{B}|$ — невід'ємна і $\rho(|\mathbf{B}|) < 1$, то за умови $\alpha < 1$ справедлива нерівність:

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\|_\omega \leq \alpha \|\mathbf{x} - \mathbf{y}\|_\omega. \quad (5.22)$$

Нехай відстань $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_\omega$. Тоді за теоремою Банаха про нерухому точку для оператора \mathbf{F} існує єдина нерухома точка $\mathbf{x}^\circ \in \mathbf{R}^n$. Оскільки \mathbf{F} — стискаючий оператор на \mathbf{R}^n , то він стискає в $\mathbf{x}^\circ \in \mathbf{R}^n$ по векторній нормі η .

5.2.3. Метод асинхронних ітерацій

Важливий внесок у розвиток паралельних ітераційних методів було зроблено в [34,35] завдяки розробці асинхронного ітераційного методу. Основна відмінність даного методу від розглянутих методів хаотичних ітерацій полягає в знятті обмеження на глибину ітераційних кроків s , що використовують при обчисленні ітераційної послідовності $\{\mathbf{x}(j)\}_{j=1}^\infty$. При цьому оператор \mathbf{F} може бути як лінійним, так і нелінійним.

Означення 5.8. Нехай $\mathbf{F}(\mathbf{x}) = (f_1(x), f_2(x), \dots, f_n(x))$ — оператор, що задає відображення $\mathbf{F} : \mathbf{R}^n \rightarrow \mathbf{R}^n$. Тоді асинхронною ітераційною послідовністю, яка відповідає оператору \mathbf{F} , називають послідовність $\{\mathbf{x}(j)\}_{j=1}^\infty$ векторів $\mathbf{x}(j) \in \mathbf{R}^n$, що визначається рекурсивно за ітераційною схемою:

$$x_i(j+1) = \begin{cases} x_i(j), & i \notin \mathbf{J}_j, \\ f_i[x_1(s_1(j)), x_2(s_2(j)), \dots, x_n(s_n(j))], & i \in \mathbf{J}_j, \end{cases} \quad (5.23)$$

де $\Psi = \{\mathbf{J}_j\}_{j=1}^\infty$ — послідовність непустих підмножин

$$\mathbf{J}_j \subset \{1, 2, \dots, n\};$$

$$\mathbf{S} = \{s_i(j)\}_{j=1}^\infty, \quad i = 1, 2, \dots, n, \text{ — послідовність цілих}$$

невід'ємних чисел, що відповідають умовам:

$$0 \leq s_i(j) \leq j \text{ при } j = 1, 2, \dots;$$

$$\lim_{j \rightarrow \infty} (s_i(j)) = \infty;$$

$i \in \Psi$ нескінченну кількість разів.

Асинхронний ітераційний метод, який відповідає оператору F з початковим вектором $x(0)$ та множинам Ψ і S , позначають $(F, x(0), \Psi, S)$.

Виходячи з визначення хаотичного ітераційного методу, існує деяке τ , для якого справедлива нерівність $s_i(j) \geq j - \tau \geq 0$ при $j = 1, 2, \dots$, $i = 1, 2, \dots$. Ця умова впливає з того, що $\lim_{j \rightarrow \infty} (s_i(j)) = \infty$. Отже, метод асинхронних ітерацій можна розглядати як узагальнення методів хаотичних ітерацій.

Наведемо означення стискаючого оператора F [35].

Означення 5.9. Оператор F , що задає відображення $F: R^n \rightarrow R^n$, є стискаючим оператором на множині $D \subset R^n$ за умови, що:

– існує невід'ємна матриця T розмірності $n \times n$, яка відповідає умові:

$$|F(x) - F(y)| \leq T|x - y| \quad \forall x, y \in D, \quad (5.24)$$

– спектральний радіус матриці T менший за одиницю: $\rho(T) < 1$.

В [34] доведена теорема, що задає умови збіжності методу асинхронних ітерацій.

Теорема 5.5. Якщо F є стискаючим оператором на замкнутій підмножині $D \subset R^n$ і $F(D) \subset D$, то асинхронний ітераційний метод $(F, x(0), \Psi, S)$, що відповідає оператору F з початковим вектором $x(0)$ на підмножині D , збігається до нерухомої точки оператора F з підмножини D .

Оскільки умовою збіжності асинхронного ітераційного методу $(F, x(0), \Psi, S)$ є наявність властивості стискання для

оператора F у всіх точках замкнутої підмножини, то таку умову вважають більш жорсткою, ніж умова збіжності методу хаотичних ітерацій (F, J) , яка впливає з властивості стискання оператора F тільки в його нерухомій точці.

5.2.4. Метод асинхронних ітерацій з нерухомими точками

У [38] вперше запропоновано паралельні асинхронні методи з нерухомими точками. Ці методи [39,40] можна розглядати як подальше узагальнення методів [54,133,35].

Означення 5.10. Нехай задано відображення $F: R^n \rightarrow R^n$. Тоді асинхронною ітераційною послідовністю, яка відповідає оператору F , будемо називати послідовність $\{x(t)\}_{t=1}^{\infty}$ векторів $x(t) \in R^n$, що визначається рекурсивно за ітераційною схемою:

$$x_i(t+1) = \begin{cases} x_i(t) & \forall t \notin T^i, \\ f_i[x_1(s_1^i(t)), x_2(s_2^i(t)), \dots, x_n(s_n^i(t))] & \forall t \in T^i, \end{cases} \quad (5.25)$$

де $S^i = \{s_k^i(t)\}_{t=1}^{\infty}$, $i, k = 1, 2, \dots, n$, — множини цілих

невід'ємних чисел, що відповідають умовам:

$$0 \leq s_k^i(t) \leq t, \quad \forall t \geq 0.$$

На відміну від асинхронного ітераційного методу (5.23), у якому всім компонентам $f_i(x)$ оператора F відповідає одна і та ж множина S , для даного методу з кожною компонентою $f_i(x)$ співставляється своя множина S^i . З цієї причини вибір конкретної множини $S = S^i, i = 1, \dots, n$, та T^i зводить ітераційну схему (5.25) до одного з розглянутих раніше випадків, які називають сценаріями.

Розрізняють ітераційні методи з повною та частковою асинхронністю.

Означення 5.11. Ітераційним методом з повною асинхронністю називають метод, що базується на ітераційній послідовності (5.25) за умов:

$$1) t_l \in T^i, i=1,2,\dots,n, \lim_{l \rightarrow \infty} t_l \rightarrow \infty ;$$

$$2) \lim_{k \rightarrow \infty} s_k^i(t_l) \rightarrow \infty .$$

Означення 5.12. Ітераційним методом з частковою асинхронністю називають метод, що базується на ітераційній послідовності (5.25), якщо існує міра асинхронності, яка визначається константою τ , і справджуються наступні залежності:

1) для кожного $t \geq 0$ і для кожного $i=1,2,\dots,n$ існує хоча б один елемент множини $\{t, t+1, t+2, \dots, t+\tau-1\}$, який належить до T^i ;

2) для елементів множин S^i справджується нерівність $t-\tau < s_k^i(t) \leq t, \forall t \in T^i$;

$$3) s_i^i(t) = t, \forall t \in T^i, i=1,2,\dots,n .$$

Умови достатності для збіжності ітераційного методу з повною асинхронністю задають теоремою [39]:

Теорема 5.6. Нехай $X = \prod_{i=1}^n X_i$ та для кожного $i \in \{1,2,\dots,n\}$

існує послідовність $\{X_i(k)\}$ підмножин множини X_i , для якої виконуються умови:

$$1) X(k) = X_1(k) \times X_2(k) \times \dots \times X_n(k) —$$

умова існування блоків;

$$2) X_i(k+1) \subseteq X_i(k) —$$

умова існування вкладених множин;

$$3) F(X(k)) \subseteq X(k+1) —$$

умова існування вкладених відображень множин;

4) для $x(k) \in X(k), k \in N, \lim_{k \rightarrow \infty} x(k) \rightarrow x^\circ$, тобто кожна

гранична точка послідовності $\{x(k)\}$ є нерухомою

точкою оператора F — умова синхронної збіжності.

Тоді при $x(0) \in X(0)$ кожна гранична точка послідовності $\{x(k)\}$, яка одержана шляхом застосування ітераційної схеми (5.25), є нерухомою точкою оператора F .

Згідно з означенням ітераційного методу з частковою асинхронністю кожне наступне значення вектора $x(t+1)$ формується на основі обмеженої підмножини попередніх значень $z(t) = \{x(t), x(t-1), \dots, x(t-\tau+1)\}$, кількість елементів якої залежить від міри асинхронності τ . Підмножини $z(t)$ формують множину Z при $t \geq 0$. Позначимо через X° всі можливі нерухомі точки $\{x \in X | x = f(x)\}$, а через Z° множину всіх можливих елементів Z , що мають вигляд $\{x^\circ, \dots, x^\circ\}$, де $x^\circ \in X^\circ$.

Теорема 5.7 (теорема Ляпунова). Нехай F — неперервний оператор в умовах часткової асинхронності. Припустимо, що існує додатне ціле t' і неперервна функція $d: Z \rightarrow [0, \infty)$ з властивостями:

1) для кожного $z(0) \in Z^\circ$ і для кожного сценарію справедлива нерівність $d(z(t')) < d(z(0))$;

2) для кожного $z(t) \in Z$, для кожного $t \geq 0$ і для кожного сценарію справедлива нерівність $d(z(t+1)) < d(z(t))$.

Тоді $z^\circ \in Z^\circ$ для кожної граничної точки послідовності $\{z(t)\}$.

Доведення теореми 5.6 та теореми 5.7 базуються на ідеї вкладених множин. Нехай всі компоненти на початку

ітераційного процесу належать до $X(k)$. Після кількох ітерацій, відповідно до умови теореми, отримаємо $x_i \in X_i(l)$, $l > k$. Всі отримані значення задовольняють умову: $X_i(l) \subset X(k+1)$.

Висновок. Замінімо в теоремі 5.6 глобальну умову 3 на локальну $f_i(X(k)) \subseteq X(k+1)$. Тоді асинхронний нестационарний ітераційний метод, що базується на ітераційній схемі (5.25), збігатиметься до єдиної нерухомої точки x° , що є спільною для всіх f_i .

Існують теореми, що дають строге доведення твердження, наведеного у висновку, для конкретних випадків. Одним із них є випадок, коли кожний з компонентів X_i простору X є нормованим лінійним простором $(X_i, \|\cdot\|_i)$. Визначимо норму $\|\cdot\|_\omega$ за формулою:

$$\|x\|_\omega = \max_{i=1,n} \frac{\|x\|_i}{\omega_i}, \quad (5.26)$$

де $\omega_i > 0$, $i = 1, 2, \dots, n$.

Теорема 5.8. Нехай існує нерухома точка $f_i(x^\circ) = x^\circ$ для $x^\circ \in X$, $i = 1, \dots, n$, і деяке таке число $\alpha \in [0, 1)$, що виконуються нерівності

$$\|f_i(x) - x^\circ\|_\omega \leq \alpha \|x - x^\circ\|_\omega. \quad (5.27)$$

Тоді асинхронний нестационарний ітераційний метод, що базується на ітераційній схемі (5.25), збігатиметься до єдиної нерухомої точки x° , яка є спільною для всіх f_i .

В [67] доведено стаціонарний випадок $F(x^\circ) = x^\circ$, а в [68, 69] розглянуті різні варіанти доведення нестационарного випадку $f_i(x^\circ) = x^\circ$, $i = 1, \dots, n$.

Ще один випадок, для якого знайдено строге доведення збіжності ітераційного методу з повною асинхронністю, базується на застосуванні операторів з властивістю парастискання.

Означення 5.13. Оператор $F: R^n \rightarrow R^n$ називають оператором з властивістю парастискання для нерухомої точки x° , якщо існує така невід'ємна матриця $P \in R^{n \times n}$ зі спектральним радіусом $\rho(P) < 1$, що для всіх $x \in X$ справедлива нерівність

$$\begin{pmatrix} \|(Fx)_1 - x_1^\circ\|_1 \\ \dots \\ \|(Fx)_n - x_n^\circ\|_n \end{pmatrix} \leq P \cdot \begin{pmatrix} \|x_1 - x_1^\circ\|_1 \\ \dots \\ \|x_n - x_n^\circ\|_n \end{pmatrix}, \quad (5.28)$$

яка є покомпонентною нерівністю в просторі R^n [149].

Теорема 5.9. Нехай f_i має властивість парастискання по відношенню до нерухомої точки x° з матрицею P незалежно від i . Тоді асинхронний нестационарний метод, що базується на ітераційній схемі (5.25), збігається до нерухомої точки x° , яка є єдиною для всіх f_i .

Ця теорема була вперше доведена в [34] для асинхронного ітераційного методу з ітераційною послідовністю (5.23), а потім поширена на ітераційну послідовність (5.25).

Умова стискання (5.27) може бути ослабленою при використанні M -функцій та діагонально-домінантних функцій [139]. Якщо умова стискання

$$\|F(x) - x^\circ\|_\omega < \|x - x^\circ\|_\omega \quad (5.29)$$

при $\mathbf{x} \neq \mathbf{x}^\circ$ є справедливою в обох напрямках, то таке відображення називають парастисканням [68,154].

5.3. Двостадійні ітераційні методи

5.3.1. Початкові поняття про мультирозщеплюючі методи

Розглянемо мультирозщеплюючі паралельні ітераційні методи розв'язування системи лінійних рівнянь (5.2) з невідродженою матрицею \mathbf{A} розмірності $n \times n$. Ці методи вперше були представлені в [145], і з того часу активно досліджуються [77,78]. Розвиток їх почався з узагальнення блочного методу Якобі [173], який використовує представлення матриці \mathbf{A} у вигляді блочної матриці з $L \times L$ блоками,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \dots & \mathbf{A}_{1L} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \dots & \mathbf{A}_{2L} \\ \dots & \dots & \dots & \dots \\ \mathbf{A}_{L1} & \mathbf{A}_{L2} & \dots & \mathbf{A}_{LL} \end{pmatrix}, \quad (5.30)$$

з невідродженими діагональними елементами \mathbf{A}_{ll} розмірності $n_l \times n_l$ при

$$n = \sum_{l=1}^L n_l. \quad (5.31)$$

Вектори \mathbf{x} та \mathbf{b} для системи (5.2) представлені у вигляді:

$$\mathbf{x}^\Gamma = \left((\mathbf{x}^1)^\Gamma, (\mathbf{x}^2)^\Gamma, \dots, (\mathbf{x}^l)^\Gamma, \dots, (\mathbf{x}^L)^\Gamma \right), \quad (5.32)$$

$$\mathbf{b}^\Gamma = \left((\mathbf{b}^1)^\Gamma, (\mathbf{b}^2)^\Gamma, \dots, (\mathbf{b}^l)^\Gamma, \dots, (\mathbf{b}^L)^\Gamma \right). \quad (5.33)$$

Ітераційна схема для блочного методу Якобі має такий вигляд:

$$\mathbf{A}_{ll} \mathbf{x}_i^l = \mathbf{b}^l - \sum_{\substack{j=1 \\ j \neq l}}^L \mathbf{A}_{lj} \mathbf{x}_{i-1}^j, \quad (5.34)$$

$$\text{де } \mathbf{x}_0^\Gamma = \left((\mathbf{x}_0^1)^\Gamma, (\mathbf{x}_0^2)^\Gamma, \dots, (\mathbf{x}_0^l)^\Gamma, \dots, (\mathbf{x}_0^L)^\Gamma \right),$$

$$i = 1, 2, \dots,$$

$$l = 1, \dots, L.$$

Очевидно, що використання даної ітераційної схеми дозволяє застосовувати паралельну обробку інформації для формування вектора ітерації.

$$\mathbf{x}_i^\Gamma = \left((\mathbf{x}_i^1)^\Gamma, (\mathbf{x}_i^2)^\Gamma, \dots, (\mathbf{x}_i^l)^\Gamma, \dots, (\mathbf{x}_i^L)^\Gamma \right). \quad (5.35)$$

Наведемо означення [79], які будуть корисними для розуміння принципів побудови мультирозщеплюючих двостадійних ітераційних методів.

Означення 5.14. Вектор $\mathbf{x} \geq 0$, ($\mathbf{x} > 0$) називають невід'ємним (додатним), якщо всі його елементи $x_i \geq 0$ ($x_i > 0$) при $i = 1, 2, \dots, n$.

Означення 5.15. Матрицю $\mathbf{B} \geq 0$, ($\mathbf{B} > 0$) називають невід'ємною (додатною), якщо всі її елементи $b_{ij} \geq 0$, ($b_{ij} > 0$) при $i, j = 1, 2, \dots, n$.

Означення 5.16. Матриці \mathbf{A} і \mathbf{B} або вектори \mathbf{x} і \mathbf{y} порівнюють $\mathbf{A} \geq \mathbf{B}$, \mathbf{B}_j , $j = 1, \dots, n$, якщо $\mathbf{A} - \mathbf{B} \geq 0$ ($\mathbf{A} - \mathbf{B} > 0$) або $\mathbf{x} \geq \mathbf{y}$ ($\mathbf{x} > \mathbf{y}$), якщо $\mathbf{x} - \mathbf{y} \geq 0$ ($\mathbf{x} - \mathbf{y} > 0$).

Означення 5.17. Для даної матриці $\mathbf{A} = (a_{ij})$, $i, j = 1, 2, \dots, n$, завжди існує модуль $|\mathbf{A}| = (|a_{ij}|)$ такий, що $|\mathbf{A}| \geq 0$.

Означення 5.18. Для матриць $A^{n \times n}$, $B^{n \times n}$ справедлива нерівність $|AB| \leq |A| \cdot |B|$.

Означення 5.19. Нехай $Z^{n \times n}$ — множина всіх дійсних матриць розмірності $n \times n$, які містять недодатні недиагональні елементи [37]. Тоді невироджену матрицю $A \in Z^{n \times n}$ називають M -матрицею за умови, що $A^{-1} \geq 0$, тобто A є монотонною матрицею [173].

Означення 5.20. Порівняльною матрицею для матриці $A = (a_{ij}) \in R^{n \times n}$ називають таку матрицю $G(A) = (g_{ij}) \in R^{n \times n}$, елементи якої визначаються з виразу:

$$a_{ij} = \begin{cases} |a_{ii}|, & i = j, \\ -|a_{ij}|, & i \neq j. \end{cases} \quad (5.36)$$

Означення 5.21. Матрицю A називають H -матрицею, якщо її порівняльна матриця $G(A) \in M$ -матрицею [150].

Лема 5.2. Нехай матриці $A, B \in R^{n \times n}$. Тоді справедливі такі співвідношення:

- 1) Якщо $A \in M$ -матрицею, $B \in Z^{n \times n}$ і $A \leq B$, то $B \in M$ -матрицею [149].
- 2) Якщо $A \in H$ -матрицею, то $|A^{-1}| \leq G(A)^{-1}$ [143].
- 3) Якщо $|A| \leq B$, то $\rho(A) \leq \rho(B)$ [149].

Лема 5.3. Нехай $A \in R^{n \times n}$. Представлення матриці A у вигляді різниці $A = M - N$ називається розщепленням, якщо матриця M невироджена, і збіжним розщепленням, якщо $\rho(M^{-1}N) < 1$.

Розщеплення $A = M - N$ називають:

- 1) звичайним розщепленням, якщо $M^{-1} \geq 0$, $N \geq 0$ [173];
- 2) слабким звичайним розщепленням,

якщо $M^{-1} \geq 0$, $M^{-1}N \geq 0$ [149];

- 3) H -розщепленням, якщо $G(M) - |N| \in M$ -матрицею [80];
- 4) H -сумісним розщепленням, якщо $G(A) = G(M) - |N|$ [80].

Лема 5.4.

- 1) Якщо $A = M - N \in$ слабким регулярним розщепленням, то $\rho(M^{-1}N) < 1$ тоді і тільки тоді, коли $A^{-1} \geq 0$ [173].
- 2) Якщо $A = M - N \in H$ -розщепленням, тоді A та $M \in H$ -матрицями і $\rho(M^{-1}N) \leq \rho(G(M)^{-1}|N|) < 1$ [143].
- 3) Якщо $A = M - N \in H$ -сумісним розщепленням і $A \in H$ -матрицею, тоді таке розщеплення є H -розщепленням і збігається [80].

Лема 5.5. Невиродженій матриці $A \in R^{n \times n}$ і матриці $T \in R^{n \times n}$, для якої існує $(I - T)^{-1}$, завжди відповідає пара матриць $M = A(I - T)^{-1}$, $N = AT(I - T)^{-1}$ таких, що $T = M^{-1}N$ і $A = M - N$, де M — невироджена матриця [123].

Лема 5.6. Нехай матриця $A \in R^{n \times n}$ і вектор $v \in R^n$, $v > 0$, а також існує деяке число $\alpha \geq 0$ таке, що $|A|v \leq \alpha v$. Тоді для всіх

$$\|Ay\|_v \leq \alpha \|y\|_v, \text{ де } \|y\|_v = \max_{i=1, n} \frac{y_i}{v_i}.$$

Лема 5.7. Нехай $T_1, T_2, \dots, T_j, \dots$ — послідовність невід'ємних матриць в R^n . Якщо існує дійсне число $0 \leq \theta < 1$ і вектор $v > 0$ в R^n , які задовольняють нерівності: $T_j v \leq \theta v$, $j = 1, 2, \dots$, то $\lim_{j \rightarrow \infty} T_j, T_{j-1}, \dots, T_1 = 0$ [121].

Мультирозщеплюючий метод [145] базується на розщепленні матриці A на

$$A = M_l - N_l \quad (5.37)$$

та діагональні невід'ємні зважені матриці E_l .

Ітераційна схема мультирозщеплюючого методу має вигляд:

$$x_i = \sum_{l=1}^L E_l \cdot M_l^{-1} (N_l x_{i-1} + b). \quad (5.38)$$

Доведення збіжності методу (5.38) дано в [145] для $A^{-1} \geq 0$, коли (5.37) є слабким звичайним розщепленням, тобто $M_l^{-1} \geq 0$, $M_l^{-1} N_l \geq 0$. Обчислювальний алгоритм, побудований за даним методом, показано на рис.5.7. Мультирозщеплюючий метод може бути зведений до блочного методу Якобі за умов:

$$\begin{cases} M_l = \text{Diag}(I, \dots, I, A_{ll}, I, \dots, I), \\ E_l = \text{Diag}(0, \dots, 0, I, 0, \dots, 0). \end{cases} \quad (5.39)$$

Як видно з рис.5.7, алгоритм мультирозщеплюючого методу (5.38) передбачає попереднє розв'язування лінійної системи

$$M_l y_l = N_l x_{i-1} + b. \quad (5.40)$$

Очевидно, що не завжди можуть бути використані прямі методи для одержання розв'язку (5.40). У цих випадках застосовують ітераційні методи. Використання ітераційних методів для виконання внутрішніх ітерацій започаткувало розвиток двостадійних мультирозщеплюючих методів.

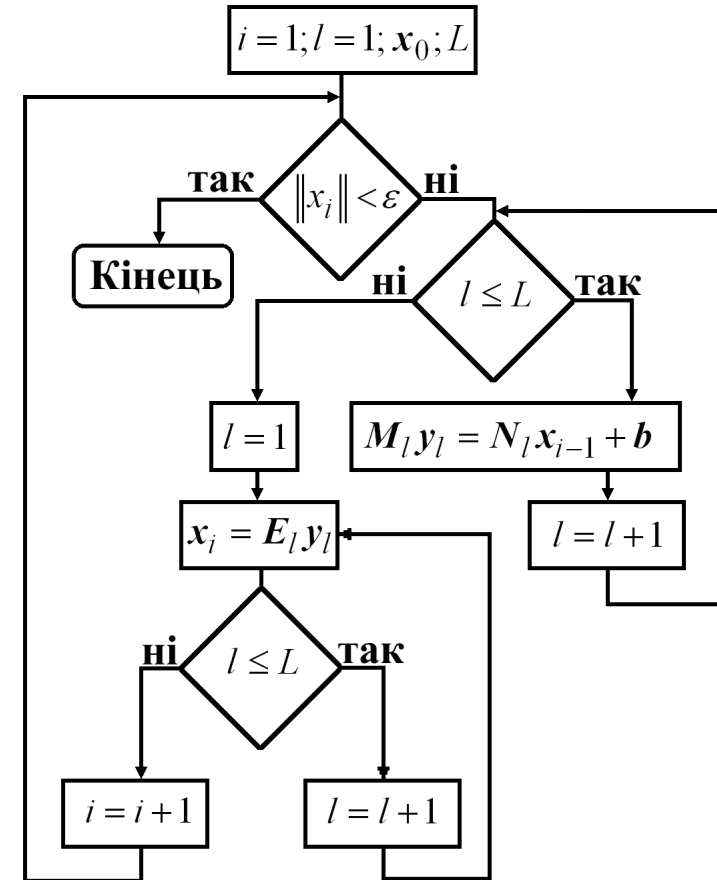


Рис.5.7. Блок-схема алгоритму мультирозщеплюючого методу.

5.3.2. Мультирозщеплюючі ітераційні методи

Багатостадійні ітераційні методи, що представлені в [47,78], [79,87,144], для розщеплення (5.37) додатково застосовують розщеплення множини матриць M_l :

$$M_l = B_l - C_l, \quad l=1, \dots, L. \quad (5.41)$$

Для розв'язування (5.40) задають кількість локальних ітерацій $s(i,l)$, що може бути постійною на протязі всього періоду обчислень (стаціонарний метод) або змінюватися в залежності від номера блоку чи макроітерації (нестационарний метод) відповідно до критерію збіжності [87]. На рис.5.8 наведено блок-схему алгоритму двостадійного мультирозщеплюючого методу.

Ітераційна схема цього методу, що враховує залежність від кількості локальних ітерацій, має вигляд:

$$x_i = \sum_{l=1}^L E_l \left[R_l^{s(i,l)} x_{i-1} + \sum_{j=0}^{s(i,l)-1} R_l^j B_l^{-1} (N_l x_{i-1} + b) \right], \quad (5.42)$$

де $R_l = B_l^{-1} C_l$.

В [170] доведено збіжність методу для довільної кількості локальних ітерацій $s(i,l)$ за умов $A^{-1} \geq 0$ і $M_l^{-1} \geq 0$, $N_l \geq 0$, $B_l^{-1} \geq 0$, $B_l^{-1} C \geq 0$, що задають звичайне макророзщеплення (5.37) та слабке звичайне локальне розщеплення (5.41).

Розглянемо локальне розщеплення з параметром:

$$M_l = \frac{B_l}{\omega} - \frac{(1-\omega)B_l}{\omega} - C_l. \quad (5.43)$$

При $\omega=1$ розщеплення (5.43) зводиться до (5.41), а при $\omega > 0$ одержуємо нестационарний двостадійний релаксаційний метод [77] з ітераційною схемою обчислення внутрішніх ітерацій

$$B_{l,j} = \omega(C_{l,j-1} + N_l x_{i-1} + b) + (1-\omega)B_l y_{l,j-1}. \quad (5.44)$$

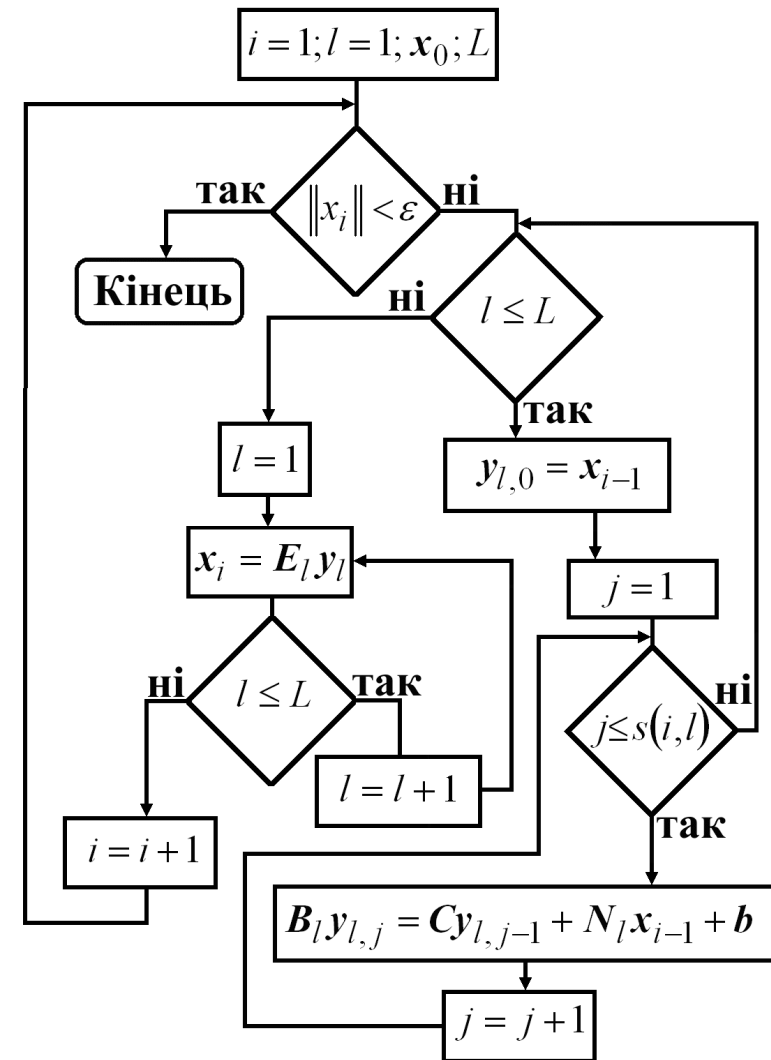


Рис.5.8. Блок-схема алгоритму двостадійного мультирозщеплюючого методу.

5.3.2.1. Синхронні двостадійні ітераційні методи

Позначимо через \mathbf{x}° розв'язок системи лінійних рівнянь (5.2). Тоді поточну похибку на макроітерації i визначають як різницю $e_i = \mathbf{x}^\circ - \mathbf{x}_i$.

Розглянемо дві блочно-діагональні матриці:

$$\mathbf{R}(i) = \text{Diag}(\mathbf{R}_l^{s(i,l)}), \quad \mathbf{T} = [\mathbf{I} - \mathbf{R}(i)]\mathbf{M}^{-1} \quad (5.45)$$

та зробимо перетворення для виразу, який входить до (5.42):

$$\begin{aligned} \sum_{j=0}^{s(i,l)-1} \mathbf{R}_l^j \mathbf{B}_l &= (\mathbf{I} - \mathbf{R}_l^{s(i,l)}) (\mathbf{I} - \mathbf{R}_l)^{-1} \mathbf{B}_l^{-1} = \\ &= (\mathbf{I} - \mathbf{R}_l^{s(i,l)}) \mathbf{M}_l^{-1}. \end{aligned} \quad (5.46)$$

Враховуючи дані перетворення, представимо ітераційну схему (5.42) у вигляді:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{R}(i)\mathbf{x}_{i-1} + \mathbf{T}(i)(\mathbf{N}\mathbf{x}_{i-1} + \mathbf{b}) = \\ &= \mathbf{R}(i)\mathbf{x}_{i-1} + \mathbf{T}(i)\mathbf{N}\mathbf{x}_{i-1} + \mathbf{T}(i)\mathbf{b}. \end{aligned} \quad (5.47)$$

Введемо заміну $\mathbf{D}(i) = \mathbf{R}(i) + \mathbf{T}(i)\mathbf{N}$ для виразу (5.47). Тоді

$$\mathbf{x}_i = \mathbf{D}(i)\mathbf{x}_{i-1} + \mathbf{T}(i)\mathbf{b}. \quad (5.48)$$

Виходячи з ітераційної схеми (5.48) та враховуючи, що виконуються умови збіжності методу (5.42), можна записати для нерухомої точки:

$$\mathbf{x}^\circ = \mathbf{D}(i)\mathbf{x}^\circ + \mathbf{T}(i)\mathbf{b}. \quad (5.49)$$

Тому похибка на ітераційному кроці i може бути виражена через початкову похибку e_0 за допомогою виразу:

$$e_i = \mathbf{D}(i)e_{i-1} = \mathbf{D}(i)\mathbf{D}(i-1)\cdots\mathbf{D}(1)e_0. \quad (5.50)$$

В [123] доведена теорема про збіжність методу (5.48) за умови монотонності матриці \mathbf{A} .

Теорема 5.10. Нехай $\mathbf{A} \geq 0$ і $\mathbf{A} = \mathbf{M} - \mathbf{N}$ — звичайне розщеплення при $\mathbf{M} = \text{Diag}(\mathbf{M}_l)$, а $\mathbf{M}_l = \mathbf{B}_l - \mathbf{C}_l, l = 1, 2, \dots, L$, — множина слабких звичайних розщеплень. Тоді двостадійний ітераційний метод (5.48) збігається до нерухомої точки $\mathbf{x}^\circ = \mathbf{A}^{-1}\mathbf{b}$ при довільному початковому векторі \mathbf{x}_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1, l = 1, 2, \dots, L, i = 1, 2, \dots$.

Д о в е д е н н я. Використовуючи введені раніше спрощення, отримуємо:

$$\mathbf{D}(i) = \mathbf{R}(i) + \mathbf{T}(i)\mathbf{N} = \mathbf{T}(i) [\mathbf{T}(i)^{-1}\mathbf{R}(i) + \mathbf{N}], \quad (5.51)$$

$$\mathbf{D}(i) = \mathbf{R}(i) + \mathbf{T}(i)\mathbf{N} = \mathbf{R}(i) + [\mathbf{I} - \mathbf{R}(i)]\mathbf{M}^{-1}\mathbf{N}. \quad (5.52)$$

Спіраючись на лему 5.5, запишемо:

$$\mathbf{A} = \mathbf{T}(i)^{-1} - [\mathbf{T}(i)^{-1}\mathbf{R}(i) + \mathbf{N}]. \quad (5.53)$$

Існування оберненої матриці $\mathbf{T}(i)^{-1}$ впливає з леми 5.4 та того факту, що $\mathbf{M} \geq 0$ тоді і тільки тоді, коли $\mathbf{M}_l \geq 0, l = 1, 2, \dots, L$. Матриці $\mathbf{T}(i) \geq 0, \mathbf{D}(i) \geq 0$, що впливає з еквівалентності перетворень, в результаті яких одержано (5.52). Враховуючи (5.51), запишемо: $\mathbf{T}(i)\mathbf{A} = \mathbf{I} - \mathbf{D}(i)$. Звідси

$$\mathbf{D}(i) = \mathbf{I} - \mathbf{T}(i)\mathbf{A}. \quad (5.54)$$

Нехай $\mathbf{v} > 0$ — довільний вектор і $\mathbf{w} = \mathbf{A}^{-1}\mathbf{v}$. Тоді, виходячи з (5.54) та деяких додаткових умов:

$$\begin{aligned} \mathbf{D}(i)\mathbf{w} &= [\mathbf{I} - \mathbf{T}(i)\mathbf{A}]\mathbf{w} = \mathbf{w} - \mathbf{T}(i)\mathbf{v} = \\ &= \mathbf{w} - \mathbf{B}^{-1}\mathbf{v} - \text{Diag} \left(\sum_{j=0}^{s(i,l)-1} \mathbf{R}_l^j \mathbf{B}_l^{-1} \right) \mathbf{v}. \end{aligned} \quad (5.55)$$

Оскільки $\text{Diag}\left(\sum_{j=0}^{s(i,l)-1} \mathbf{R}_l^j \mathbf{B}_l^{-1}\right) \mathbf{v} \geq 0$, $\mathbf{D}(i)\mathbf{w} \geq 0$,

$\mathbf{w} - \mathbf{B}^{-1}\mathbf{v} < \mathbf{w}$, то існує таке $0 \leq \theta < 1$, що $\mathbf{w} - \mathbf{B}^{-1}\mathbf{v} \leq \theta \mathbf{w}$.
Отже,

$$\mathbf{D}(i)\mathbf{w} \leq \theta \mathbf{w} \text{ при } i=1,2,\dots \quad (5.56)$$

За лемою 5.8 добуток $\mathbf{D}(i) \cdot \mathbf{D}(i-1) \cdot \dots \cdot \mathbf{D}(1)$ прямує до нуля зі зростанням кількості ітерацій i . Тому $\lim_{i \rightarrow \infty} e_i \rightarrow 0$, і нерівність (5.56) не залежить від кількості локальних ітерацій $s(i,l)$, $l=1,2,\dots,L$, $i=1,2,\dots$.

В [79] доведена теорема про збіжність методу (5.48) за умови, коли $\mathbf{A} \in H$ -матрицею.

Теорема 5.11. Нехай $\mathbf{A} \in H$ -матрицею і $\mathbf{A} = \mathbf{M} - \mathbf{N}$ — H -розщеплення при $\mathbf{M} = \text{Diag}(\mathbf{M}_l)$, а $\mathbf{M}_l = \mathbf{B}_l - \mathbf{C}_l$, $l=1,2,\dots,L$, — множина H -сумісних розщеплень. Тоді двостадійний ітераційний метод (5.48) збігається до нерухомої точки $\mathbf{x}^\circ = \mathbf{A}^{-1}\mathbf{b}$ при довільному початковому векторі \mathbf{x}_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1$, $l=1,2,\dots,L$, $i=1,2,\dots$.

Д о в е д е н н я. Виходячи з умови теореми:

$$\begin{aligned} \mathbf{D}(i) = & \text{Diag}\left(\left(\mathbf{B}_l^{-1}\mathbf{C}_l\right)^{s(i,l)}\right) + \\ & + \text{Diag}\left(\sum_{j=0}^{s(i,l)-1} \left(\mathbf{B}_l^{-1}\mathbf{C}_l\right)^j \mathbf{B}_l^{-1}\right) \mathbf{N}. \end{aligned} \quad (5.57)$$

Спираючись на лему 5.2, одержимо наступні граничні значення:

$$\begin{aligned} |\mathbf{D}(i)| \leq & \text{Diag}\left(\left(\left\|\mathbf{B}_l^{-1}\right\|\left\|\mathbf{C}_l\right\|\right)^{s(i,l)}\right) + \\ & + \text{Diag}\left(\sum_{j=0}^{s(i,l)-1} \left(\left\|\mathbf{B}_l^{-1}\right\|\left\|\mathbf{C}_l\right\|\right)^j \left\|\mathbf{B}_l^{-1}\right\|\right) |\mathbf{N}| \leq \\ \leq & \text{Diag}\left(\left(\mathbf{G}(\mathbf{B}_l^{-1})\mathbf{C}_l\right)^{s(i,l)}\right) + \\ & + \text{Diag}\left(\sum_{j=0}^{s(i,l)-1} \left(\mathbf{G}(\mathbf{B}_l^{-1})\mathbf{C}_l\right)^j \mathbf{G}(\mathbf{B}_l^{-1})\right) |\mathbf{N}|. \end{aligned}$$

Права частина даної нерівності $\overline{\mathbf{D}(i)}$ є матрицею, яка відповідає ітерації i двостадійного ітераційного методу з монотонною матрицею $\mathbf{G}(\mathbf{M}) - |\mathbf{N}|$ при звичайному розщепленні і $\mathbf{G}(\mathbf{M}) = \text{Diag}(\mathbf{G}(\mathbf{M}_l))$, $\mathbf{G}(\mathbf{M}_l) = \mathbf{G}(\mathbf{B}_l) - |\mathbf{C}_l|$. Оскільки дані умови відповідають теоремі 5.10, можемо записати нерівність

$$|\mathbf{D}(i)|\mathbf{w} \leq \overline{\mathbf{D}(i)}\mathbf{w} \leq \theta \mathbf{w}, \quad (5.58)$$

де $\mathbf{w} \in \mathbf{R}^n$, $\mathbf{w} > 0$, $\theta \in [0,1)$, $i=1,2,\dots$.

Нехай

$$\mathbf{P}(i) = \mathbf{D}(i) \cdot \mathbf{D}(i-1) \cdot \dots \cdot \mathbf{D}(1).$$

Тоді

$$|\mathbf{P}(i)| \leq |\mathbf{D}(i)| \cdot |\mathbf{D}(i-1)| \cdot \dots \cdot |\mathbf{D}(1)|.$$

Виходячи з (5.58) та леми 5.7, $\mathbf{P}(i)$ прямує до нуля зі зростанням кількості ітерацій i . Тому $\lim_{i \rightarrow \infty} e_i \rightarrow 0$, і нерівність

(5.58) не залежить від кількості локальних ітерацій $s(i,l)$, $l=1,2,\dots,L$, $i=1,2,\dots$.

Двостадійний метод (5.42) з алгоритмом, який представлено на рис.5.8, є синхронним ітераційним методом. Принцип синхронності полягає у тому, що кожний наступний

макроітераційний крок $x_i = \sum_{l=1}^L E_l y_l$ може бути зроблений тільки

після виконання всіх локальних ітераційних кроків $M_l y_l = N_l x_{i-1} + b$, $l=1,2,\dots,L$. Для підвищення ефективності обчислювального процесу застосовують асинхронні двостадійні методи.

5.3.2.2. Макроасинхронні двостадійні методи

Обчислювальний процес за даним методом може бути організований на нейронній структурі з повним графом міжнейронних зв'язків. Основною особливістю асинхронних обчислень є можливість продовжувати ітераційний процес у групі нейронів, які відповідають за блок, не чекаючи закінчення макроітераційного кроку в інших блоках. Результатом обчислень у блоці l є значення $x_i^{(l)} = E_l y_i$, що формується на основі даних поточних макроітерацій в інших блоках. Для формального опису двостадійного асинхронного макроітераційного методу задамо деяку множину послідовностей $\{J_i\}_{i=1}^{\infty}$, де $J_i \subseteq \{1,2,\dots,L\}$. Поточний блок $l \in J_i$ за умови, що його обробка відбувається на ітераційному кроці i . Особливість принципу макроасинхронності полягає в тому, що обмін інформацією з іншими блоками відбувається тільки перед початком чергової макроітерації. Оскільки цей момент обміну настає незалежно для кожного блоку, то дані, одержані від інших блоків, відповідають різним номерам ітерацій у цих блоках. Позначимо ці номери у вигляді індексів $r(k,i)$. Ітераційна схема, яка побудована за даним методом, базується на декомпозиції матриці на блочні оператори. Для початкового вектора

$$x_0 = (x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(L)})$$

задамо послідовність

$$x_i^{(l)} = \begin{cases} x_{i-1}^{(l)}, & l \notin J_i, \\ \mathbf{D}^{(l)}(i) \left(x_{r(1,i)}^{(1)}, x_{r(2,i)}^{(2)}, \dots, x_{r(l,i)}^{(l)} \right)^T + \\ \quad + \mathbf{T}^{(l)}(i) \mathbf{b}, & l \in J_i, \end{cases} \quad (5.59)$$

де $\mathbf{D}^{(l)}(i)$ і $\mathbf{T}^{(l)}(i)$ — елементи декомпозиції матриць $\mathbf{D}(i)$ і $\mathbf{T}(i)$, які визначають ітераційну схему (5.48).

Застосувавши до (5.59) ітераційну схему (5.42), одержимо більш детальний запис, що враховує локальні ітерації:

$$x_i^{(l)} = \begin{cases} x_{i-1}^{(l)}, & l \notin J_i, \\ \mathbf{E}_l \left[\mathbf{R}_l^{s(i,l)} x_{r(l,i)}^{(l)} + \right. \\ \quad \left. + \sum_{j=0}^{s(i,l)-1} \mathbf{R}_l^j \mathbf{B}_l^{-1} \left(N_l \sum_{k=1}^L x_{r(k,i)}^{(k)} + \mathbf{b} \right) \right], & l \in J_i, \end{cases} \quad (5.60)$$

де

- а) $r(l,i) < i$ для всіх $l=1,2,\dots,L$, $i=1,2,\dots$;
- б) $\lim_{i \rightarrow \infty} r(l,i) \rightarrow \infty$ для всіх $l=1,2,\dots,L$;
- в) множина $\{i | l \in J_i\}$ є необмеженою для всіх $l=1,2,\dots,L$.

Розглянемо теореми про збіжність даного методу, що сформульовані в [79] з доведеннями, які базуються на [67].

Теорема 5.12. Нехай $\mathbf{K}(i) \in \mathbf{R}^n$ — послідовність операторів, які характеризуються спільною нерухомою точкою x° , існує норма:

$$\|x\|_a = \max_{i=1,L} \frac{\|x^{(i)}\|_l}{a_l} \text{ для } l=1,\dots,L, \mathbf{a} \in \mathbf{R}^L, \mathbf{a} > 0 \text{ і справедлива}$$

нерівність:

$$\|K(i)x - x^\circ\|_a \leq \alpha \|x - x^\circ\| \text{ при } x \in \mathbf{R}^n, \alpha \in [0,1], i=1,2,\dots$$

Нехай також послідовність $r(l,i)$ і множини $J_i, l=1,\dots,L, i=1,2,\dots$, задовольняють умови а,б,в з (5.60). Тоді послідовність асинхронних ітерацій:

$$x_i^{(l)} = \begin{cases} x_{i-1}^{(l)}, & l \notin J_i, \\ \mathbf{K}^{(l)}(i) [x_{r(1,i)}^{(1)}, x_{r(2,i)}^{(2)}, \dots, x_{r(L,i)}^{(L)}], & l \in J_i, \end{cases} \quad (5.61)$$

$$i=1,2,\dots,$$

збігається до нерухомої точки x° для довільного початкового вектора x_0 .

Теорема 5.13. Нехай $A \geq 0$ і $A = M - N$ — звичайне розщеплення при $M = \text{Diag}(M_l)$, а $M_l = B_l - C_l, l=1,2,\dots,L$, — множина слабких звичайних розщеплень. Нехай також послідовність $r(l,i)$ і множини $J_i, l=1,\dots,L, i=1,2,\dots$, задовольняють умови а,б,в з (5.60). Тоді двостадійний макроасинхронний мультирозщеплюючий ітераційний метод (5.60) збігається до нерухомої точки $x^\circ = A^{-1}b$ при довільному початковому векторі x_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1, l=1,2,\dots,L, i=1,2,\dots$.

Теорема 5.14. Нехай $A \in H$ -матрицею і $A = M - N$ — H -розщеплення при $M = \text{Diag}(M_l)$, а $M_l = B_l - C_l, l=1,2,\dots,L$, — множина H -сумісних розщеплень. Нехай також послідовність $r(l,i)$ і множини $J_i, l=1,\dots,L, i=1,2,\dots$, задовольняють умови а,б,в з (5.60). Тоді двостадійний макроасинхронний мультирозщеплюючий ітераційний метод (5.60) збігається до нерухомої точки $x^\circ = A^{-1}b$ при довільному початковому

векторі x_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1, l=1,2,\dots,L, i=1,2,\dots$.

5.3.2.3. Повністю асинхронні двостадійні методи

Повністю асинхронні двостадійні методи базуються на принципі отримання нової інформації від інших блоків, на відміну від макроасинхронних методів, що використовують тільки результати чергової макроітерації. Оскільки поточне значення $x_{r(l,i)}^{(l)}$ залежить також від номера локальної ітерації $j=1,2,\dots,s(i,l)-1$, то для відображення цього факту сформуємо послідовність $r(j,l,i)$. Правило побудови ітераційної послідовності повністю асинхронного двостадійного методу є узагальненням уже розглянутого макроасинхронного методу:

$$x_i^{(l)} = \begin{cases} x_{i-1}^{(l)}, & l \notin J_i, \\ \mathbf{E}_l \left[\mathbf{R}_l^{s(i,l)} x_{r(j,l,i)}^{(l)} + \sum_{j=0}^{s(i,l)-1} \mathbf{R}_l^j \mathbf{B}_l^{-1} \left(N_l \sum_{k=1}^L x_{r(j,l,i)}^{(k)} + \mathbf{b} \right) \right], & l \in J_i, \end{cases} \quad (5.62)$$

де

а) $r(j,l,i) < i$

для всіх $j=0,\dots,s(i,l)-1, l=1,2,\dots,L, i=1,2,\dots$;

б) $\lim_{i \rightarrow \infty} \left(\min_{j=0, s(i,l)-1} r(j,l,i) \right) \rightarrow \infty$ для всіх $l=1,2,\dots,L$;

в) множина $\{i | l \in J_i\}$ є необмеженою для всіх $l=1,2,\dots,L$.

Теорема 5.15. Нехай $A \geq 0$ і $A = M - N$ — звичайне розщеплення при $M = \text{Diag}(M_l)$, а $M_l = B_l - C_l, l=1,2,\dots,L$ —

множина слабких звичайних розщеплень. Нехай також послідовність $r(j,l,i)$ і множини J_i , $j=0,\dots,s(i,l)-1$, $l=1,\dots,L$, $i=1,2,\dots$, задовольняють умови a, b, v з (5.62). Тоді двостадійний повністю асинхронний мультирозщеплюючий ітераційний метод (5.62) збігається до нерухомої точки $x^\circ = A^{-1}b$ при довільному початковому векторі x_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1$, $l=1,2,\dots,L$, $i=1,2,\dots$.

Теорема 5.16. Нехай $A \in H$ -матрицею і $A = M - N$ — H -розщеплення при $M = \text{Diag}(M_l)$, а $M_l = B_l - C_l$, $l=1,2,\dots,L$, — множина H -сумісних розщеплень. Нехай також послідовність $r(j,l,i)$ і множини J_i , $j=0,\dots,s(i,l)-1$, $l=1,\dots,L$, $i=1,2,\dots$, задовольняють умови a, b, v з (5.62). Тоді двостадійний макроасинхронний мультирозщеплюючий ітераційний метод (5.62) збігається до нерухомої точки $x^\circ = A^{-1}b$ при довільному початковому векторі x_0 та при довільній послідовності кількостей локальних ітерацій $s(i,l) \geq 1$, $l=1,2,\dots,L$, $i=1,2,\dots$.

5.4. Локально-асинхронні ітераційні методи

5.4.1. Початкові поняття про локально-асинхронні методи

Розглянемо систему нелінійних алгебраїчних рівнянь:

$$x = F(x), \quad (5.63)$$

де оператор $F: B \rightarrow B$ означений на банаховому просторі B .

Означення 5.22. Оператор F називається p -ліпшицевим в області його визначення $D(F)$, якщо існує невід'ємна матриця $L^{n \times n}$, яка задовольняє нерівність:

$$\forall x, y \in D(F): p(F(x) - F(y)) \leq Lp(x - y). \quad (5.64)$$

Матрицю L називають матрицею Ліпшиця оператора F .

Означення 5.23. Оператор F називається p -стискаючим, якщо він є p -ліпшицевим в $D(F)$ і його спектральний радіус $\rho(L) < 1$.

Означення 5.24. Нехай $\{J_i\}_{i=1}^\infty$ — хаотична послідовність множин. Зростаюча послідовність чисел $\{m_r\}_{r=0}^\infty$ називається мінімальною послідовністю, якщо виконуються такі умови:

- 1) $m_0 = 0$,
- 2) $\bigcup_{i=m_r+1}^{m_{r+1}} J_i = \{1, 2, \dots, n\}$, $r = 0, 1, 2, \dots$,
- 3) $\bigcup_{i=m_r+1}^{m_{r+1}-1} J_i \subset \{1, 2, \dots, n\}$, $r = 0, 1, 2, \dots$.

Мінімальна послідовність $\{m_r\}_{r=0}^\infty$ забезпечує максимальну ефективність асинхронних алгоритмів.

Означення 5.25. Остачею хаотичної послідовності множин $\{J_i\}_{i=1}^\infty$ називається множина:

$$\mathfrak{R} = \{r \in \{1, 2, \dots, n\} \mid \forall n \in \mathbb{N} \exists m \geq n: r \in J_m\},$$

яка складається з таких елементів множини $\{1, 2, \dots, n\}$, які входять нескінченно часто в $\{J_i\}_{i=1}^\infty$.

Означення 5.26. Хаотична послідовність множин $\{J_i\}_{i=1}^\infty$ має максимальну остачу при $\mathfrak{R} = \{1, 2, \dots, n\}$.

Лема 5.8. Хаотична послідовність множин $\{J_i\}_{i=1}^\infty$ має максимальну остачу тоді і тільки тоді, коли існує мінімальна послідовність $\{m_r\}_{r=0}^\infty$.

Означення 5.27. Гільбертовим простором H називається лінійний простір з таким скалярним добутком $\langle x, y \rangle$, що норма, означена як

$$\|x\| = \sqrt{\langle x, x \rangle},$$

переводить простір H у повний метричний простір із властивостями:

- 1) $\langle x, y \rangle = \langle y, x \rangle$,
- 2) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$,
- 3) $\langle \lambda x, y \rangle = \lambda \langle x, y \rangle$,
- 4) $\langle x, x \rangle \geq 0$, $\langle x, x \rangle = 0$ тоді і тільки тоді, коли $x = 0$,
- 5) $|\langle x, y \rangle| \leq \|x\| \cdot \|y\|$,
- 6) $|\langle x, y \rangle| = \|x\| \cdot \|y\|$ тоді і тільки тоді, коли $x = \lambda y$ або $y = 0$,
- 7) $\langle x, y \rangle = 0$ при $x = 0$,
- 8) $\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$.

Означення 5.28. Нехай X, Y — нормовані лінійні простори і $F: X \rightarrow Y$ в деякій відкритій підмножині $O \subset X$. Відображення F є диференційовним по Фреше в $x \in O$, якщо існує такий обмежений лінійний оператор $A_x \in A(X, Y)$, який задовольняє умову:

$$\lim_{\|h\| \rightarrow 0} \frac{\|F(x+h) - F(x) - A_x h\|}{\|h\|} = 0. \quad (5.65)$$

З (5.65) випливає, що диференційовне в точці x відображення є неперервним у цій точці. Вираз $A_x h$, для кожного $h \in X$ — це елемент простору Y , який називають сильним диференціалом або диференціалом Фреше відображення F в точці x . Сам лінійний оператор $F'(x) = A_x$ називають сильною похідною або похідною Фреше.

Означення 5.29. Нехай X, Y — нормовані лінійні простори, і відображення $F: X \rightarrow Y$. Слабким диференціалом або диференціалом Гато відображення F в точці x називається границя:

$$DF(x, h) = \frac{dF(x+th)}{dt}\bigg|_{t=0} = \lim_{t \rightarrow 0} \frac{F(x+th) - F(x)}{t}, \quad (5.66)$$

де під збіжністю розуміють збіжність по нормі $\|\cdot\|_Y$.

У випадку лінійності по h диференціала (5.66) справедливе співвідношення:

$$DF(x, h) = F'(x)h. \quad (5.67)$$

Обмежений лінійний оператор $F'_s(x)$ називають слабкою похідною або похідною Гато.

Теорема 5.17. Якщо існує похідна Гато $F'_s(x)$ в деякому околі O точки x_0 , і вона є операторною функцією від x , неперервною в точці x_0 , то в цій точці існує похідна Фреше, і вона співпадає з похідною Гато.

5.4.2. Локально-асинхронний метод розв'язування операторного рівняння

Якщо хаотична послідовність множин $\{J_i\}_{i=1}^{\infty}$ має максимальну остачу, то достатньою умовою збіжності розглянутих асинхронних методів є p -збіжність оператора переходу від однієї ітерації до іншої. Асинхронність у даному випадку розуміють як можливість обчислення кожної компоненти вектора ітерації незалежно від решти компонент шляхом дії на неї відповідної компоненти оператора переходу. При цьому існує збіжність послідовності ітерацій до нерухомої точки, що є спільною для всіх компонент оператора переходу.

Асинхронні ітераційні методи успішно використовують для розв'язування систем алгебраїчних рівнянь, які є результатом застосування однорідних різницевих схем для дискретизації

крайових задач. Ці рівняння характеризуються стрічковою матрицею, і при обчисленні компоненти вектора ітерації використовують обмежене число компонент, яке не залежить від кількості вузлів сіткової області. Тому такі методи мають назву локально-асинхронних методів [9]. При застосуванні цих методів умова парастискання призводить до суттєвих обмежень на величину кроків дискретизації. У зв'язку з цим в [5] проведений пошук достатніх умов, що дозволяють знизити вимоги до параметрів дискретизації.

Нехай $\mathbf{H} = \prod_{i=1}^n \mathbf{H}_i$ — прямий добуток n дійсних гільбертових

просторів, на яких задані скалярні добутки $\langle \cdot, \cdot \rangle_i, i=1, \dots, n$.

Позначимо через $\|\cdot\|_i, i=1, \dots, n$ норми, які породжені відповідними скалярними добутками.

Розглянемо операторне рівняння:

$$\mathbf{y} = \mathbf{F}(\mathbf{y}), \quad (5.68)$$

де $\mathbf{F}: \mathbf{H} \rightarrow \mathbf{H}$, \mathbf{y} — розв'язок рівняння $\mathbf{y} \in \mathbf{H}$.

Нехай для оператора \mathbf{F} існують такі константи $\gamma_1 < 0, \gamma_2 < 0$, що для всіх $\mathbf{y}, \mathbf{z} \in \mathbf{H}$ справедливі такі нерівності:

$$\langle \mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{z}), \mathbf{y} - \mathbf{z} \rangle \leq \gamma_1 \|\mathbf{y} - \mathbf{z}\|^2, \quad (5.69)$$

$$\|\mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{z})\|^2 \leq \gamma_2 \langle \mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{z}), \mathbf{y} - \mathbf{z} \rangle. \quad (5.70)$$

Тоді оператор \mathbf{F} будемо називати сильно монотонним оператором. Для сильно монотонних операторів справедлива така теорема [12].

Теорема 5.18. *Нехай у скінченному гільбертовому просторі \mathbf{H} задано неперервний сильно монотонний оператор \mathbf{A} , для якого справедлива нерівність:*

$$\langle \mathbf{A}(\mathbf{y}) - \mathbf{A}(\mathbf{z}), \mathbf{y} - \mathbf{z} \rangle \geq \delta \|\mathbf{y} - \mathbf{z}\|^2, \quad \delta > 0, \mathbf{y}, \mathbf{z} \in \mathbf{H}. \quad (5.71)$$

Тоді операторне рівняння $\mathbf{A}\mathbf{y} = \mathbf{f}$ має єдиний розв'язок у кулі:

$$\mathbf{K} = \left\{ \mathbf{y} : \|\mathbf{y}\| \leq \frac{\|\mathbf{A}(0) - \mathbf{f}\|}{\delta} \right\}. \quad (5.72)$$

Наслідок. Якщо оператор \mathbf{F} неперервний в скінченному просторі \mathbf{H} , то операторне рівняння (5.68) має єдиний розв'язок у кулі:

$$\left\{ \mathbf{y} : \|\mathbf{y}\| \leq \frac{\|\mathbf{F}(0) - \mathbf{F}(\mathbf{y})\|}{1 - \gamma_1} \right\}. \quad (5.73)$$

Використовуючи (5.69), одержимо нерівність:

$$\begin{aligned} \langle \mathbf{y} - \mathbf{F}(\mathbf{y}) - \mathbf{z} + \mathbf{F}(\mathbf{z}), \mathbf{y} - \mathbf{z} \rangle &= \|\mathbf{y} - \mathbf{z}\|^2 - \langle \mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{z}), \mathbf{y} - \mathbf{z} \rangle \geq \\ &\geq \|\mathbf{y} - \mathbf{z}\|^2 - \gamma_1 \|\mathbf{y} - \mathbf{z}\|^2 = (1 - \gamma_1) \|\mathbf{y} - \mathbf{z}\|^2. \end{aligned}$$

Тому оператор $\mathbf{I} - \mathbf{F}$ є сильно монотонним у просторі \mathbf{H} .

Введемо оператор $\Phi(\mathbf{y})$ з урахуванням релаксаційного параметра ω :

$$\Phi(\mathbf{y}) = 2\omega \mathbf{F}(\mathbf{y}) + (1 - \omega)\mathbf{y}.$$

Теорема 5.19. *Нехай $\mathbf{y} = \Phi(\mathbf{y})$ — операторне рівняння з оператором $\Phi: \mathbf{H} \rightarrow \mathbf{H}$, для якого справеджується нерівність (5.69), і релаксаційний параметр задовольняє умову:*

$$\omega < \frac{2}{2 - \gamma_2}. \quad (5.74)$$

Тоді дане операторне рівняння має в \mathbf{H} єдиний розв'язок для довільного $\mathbf{F}: \mathbf{H} \rightarrow \mathbf{H}$.

Схема доведення.

Запишемо нерівність (5.70) з використанням умови (5.74):

$$\begin{aligned} \|\Phi(\mathbf{y}) - \Phi(\mathbf{z})\|^2 &\leq \\ &\leq \left[(1 - 2\gamma_1 + \gamma_1\gamma_2)\omega^2 - 2(1 - \gamma_1)\omega + 1 \right] \|\mathbf{y} - \mathbf{z}\|^2. \end{aligned} \quad (5.75)$$

Умовою стискання оператора Φ є обмеження для коефіцієнта Лібшиця:

$$\rho^2(\omega) = (1 - 2\gamma_1 + \gamma_1\gamma_2)\omega^2 - 2(1 - \gamma_1)\omega + 1 < 1. \quad (5.76)$$

Звідси випливає, що

$$\frac{2(1 - \gamma_1)}{1 - 2\gamma_1 + \gamma_1\gamma_2} > \frac{2}{2 - \gamma_2}. \quad (5.77)$$

Отже, оператор Φ є стискаючим оператором в \mathbf{H} і, відповідно, існує розв'язок рівняння $\mathbf{y} = \Phi(\mathbf{y})$. Оскільки це рівняння еквівалентне рівнянню (5.58), то звідси також випливає існування розв'язку рівняння (5.58).

Визначимо значення ω , при якому коефіцієнт Лібшиця (5.76) досягає мінімального значення. Оскільки $(1 - 2\gamma_1 + \gamma_1\gamma_2) > 0$, то мінімальне значення $\rho(\omega)$ отримаємо при такому значенні релаксаційного параметра ω :

$$\omega_0 = \frac{1 - \gamma_1}{1 - 2\gamma_1 + \gamma_1\gamma_2}. \quad (5.78)$$

Знайдемо умову, за якої значення ω_0 входить у множину допустимих параметрів ω , заданих нерівністю (5.74):

$$\frac{1 - \gamma_1}{1 - 2\gamma_1 + \gamma_1\gamma_2} < \frac{2}{2 - \gamma_2}. \quad (5.79)$$

Звідси одержимо, що за умови

$$\gamma_1 < \frac{\gamma_2}{2 - \gamma_2} \quad (5.80)$$

ω міститься в множині $\left\{ \omega : \omega < \frac{2}{2 - \gamma_2} \right\}$.

$$\text{Тоді } \rho(\omega_0) = \frac{\gamma_1(\gamma_2 - \gamma_1)}{1 - 2\gamma_1 + \gamma_1\gamma_2}.$$

З іншого боку, для того, щоб коефіцієнт Лібшиця (5.76) був додатним, необхідною умовою є від'ємне значення його

дискримінанта, тобто $\gamma_1^2 - \gamma_1\gamma_2 < 0$. Звідси $\gamma_2 < \gamma_1$. Отже, з урахуванням (5.80), отримуємо:

$$\gamma_2 < \gamma_1 < \frac{\gamma_2}{2 - \gamma_2}. \quad (5.81)$$

Оскільки, за умовою $\gamma_2 < 0$, то нерівність (5.81) виконується завжди.

В [5] доведені теореми збіжності локально-асинхронних методів, параметри яких задовольняють розглянуті умови.

Теорема 5.20. *Нехай хаотична послідовність множин $\{\mathbf{J}_i\}_{i=1}^{\infty}$ має максимальну остачу, виконуються умови (5.69), (5.70), (5.81), і параметр релаксації ω знаходиться в діапазоні:*

$$\frac{1 - \gamma_1 - \sqrt{(1 - \gamma_1)^2 - (1 - n^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2} < \omega < \frac{1 - \gamma_1 + \sqrt{(1 - \gamma_1)^2 - (1 - n^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2}. \quad (5.82)$$

Тоді асинхронний ітераційний метод

$$\mathbf{y}_j(i) = \begin{cases} \mathbf{y}_j(i-1), & j \notin \mathbf{J}_i, \\ \Phi_j(\mathbf{y}(i-1)), & j \in \mathbf{J}_i, \end{cases} \quad i = 1, 2, \dots, \quad (5.83)$$

збігається до єдиного розв'язку φ операторного рівняння (5.68) для будь-якого початкового вектора $\mathbf{y}(0) \in \mathbf{H}$ зі швидкістю збіжності:

$$\|\mathbf{y}(i) - \varphi\| \leq \rho^{\lambda(i)}(\omega) \|\mathbf{y}(0) - \varphi\|. \quad (5.84)$$

де $\|\mathbf{y}\| = \max_{i=1, n} \|y_i\|$, $\lambda(i)$ — кількість макроітерацій, які містить

послідовність ітерацій $\{\mathbf{y}(m)\}_{m=1}^i$.

Наслідок. Нехай кожна компонента оператора \mathbf{F} залежить від обмеженої кількості компонент вектора $\mathbf{y} = (y_1, \dots, y_n)$, і їх число не перевищує β . Тоді асинхронний ітераційний метод

(5.83) називають локально-асинхронним ітераційним методом. Для даного методу може бути розширений допустимий діапазон зміни релаксаційного параметра ω шляхом модифікації нерівності (5.82):

$$\frac{1 - \gamma_1 - \sqrt{(1 - \gamma_1)^2 - (1 - \beta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2} < \omega < \frac{1 - \gamma_1 + \sqrt{(1 - \gamma_1)^2 - (1 - \beta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2}. \quad (5.85)$$

При цьому не порушуються умови теореми 5.20, і тому оцінка швидкості збіжності (5.84) залишається вірною. Звідси визначають кількість макроітерацій $\lambda(i)$, які необхідно виконати для досягнення заданої точності ε :

$$\lambda(i) \geq \frac{\ln \varepsilon}{\ln \beta \rho(\omega)}. \quad (5.86)$$

Очевидно, що метод має найвищу швидкість збіжності при $\omega = \omega_0$. Тому нижня оцінка кількості макроітерацій (5.86) має вигляд:

$$\lambda_{\min} \geq \frac{\ln \varepsilon}{\ln \beta + \ln \gamma_1(\gamma_2 - \gamma_1) - \ln(1 - 2\gamma_1 + \gamma_1\gamma_2)}. \quad (5.87)$$

Швидкість збіжності даного локально-асинхронного методу може бути збільшена у випадку введення обмежень на оператор F . Нехай для F існує диференціал по Гато в деякій кулі $K_r = \{y : \|y\|_r \leq r\}$, тобто

$$\forall y, z \in K_r \lim_{\eta \rightarrow 0} \left\| \frac{F(y + \eta z) - F(y)}{\eta} - F'(y)z \right\| = 0.$$

Існування диференціала по Гато для оператора F у кулі K_r зумовлює виконання для нього нерівності Лагранжа:

$$\|F(y) - F(z)\| \leq \sup_{0 \leq \eta \leq 1} \|F'(y + \eta(z - y))\| \cdot \|y - z\| \quad \forall y, z \in K_r.$$

Крім того, для будь-яких $y, z, v \in K_r$ $\exists \eta \in [0, 1]$ таке, що

$$\langle F(y) - F(z), v \rangle = \langle F'(y + \eta(z - y)) \rangle \langle y - z, v \rangle. \quad (5.88)$$

Умови збіжності локально-асинхронного ітераційного методу для даного оператора визначають за теоремою.

Теорема 5.21. Нехай існує диференціал оператора F по Гато в кулі, при $\gamma_3 < 0, \gamma_4 < 0$ справедливі нерівності:

$$\|F(z)y\|^2 \leq \gamma_3 \langle F'(z)y, y \rangle, \quad (5.89)$$

$$\|F'(z)y, y\| \leq \gamma_4 \|y\|^2, \quad (5.90)$$

хаотична послідовність множин $\{J_i\}_{i=1}^{\infty}$ має максимальну остачу, а релаксаційний параметр ω задовольняє умови:

$$\omega < \frac{2}{2 - \gamma_4},$$

$$\frac{1 - \gamma_1 - \sqrt{(1 - \gamma_1)^2 - (1 - \beta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2} < \omega < \frac{1 - \gamma_1 + \sqrt{(1 - \gamma_1)^2 - (1 - \beta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2}.$$

Тоді послідовність ітерацій, побудованих за локально-асинхронним методом (5.83), збігається до єдиного розв'язку φ операторного рівняння (5.68) з оцінкою швидкості збіжності:

$$\|y(i) - \varphi\| \leq \rho^{\lambda(i)}(\omega) \|y(0) - \varphi\|, \quad (5.91)$$

при $y(i) \in K_r, i = 1, 2, \dots, \rho(\omega) = (1 - \omega)^2 + \omega(2 - 2\omega + \omega\gamma_3)\gamma_4$.

5.4.3. Локально-асинхронний метод розв'язування послідовності операторних рівнянь

Дискретизація рівнянь параболічного типу полягає у формуванні послідовності операторних рівнянь, кожне з яких є системою нелінійних алгебраїчних рівнянь. У цьому випадку застосування згаданих асинхронних методів не приводить до розв'язку крайової задачі в цілому, оскільки виникає необхідність синхронізації переходу на черговий часовий шар. У випадку складних реальних процесів, описуваних рівняннями параболічного типу, на величину часових шарів можуть накладатися жорсткі обмеження, що породжує деградацію обчислювального процесу у випадку застосування нейронної мережі як паралельного обчислювального середовища.

Тому для рівнянь параболічного типу актуальною є проблема створення ітераційних методів, які дозволяють уникнути фази синхронізації при розв'язуванні послідовності рівнянь:

$$\begin{cases} \mathbf{y} = \mathbf{F}(\mathbf{y}, t) + \check{\mathbf{y}}^K, \\ \mathbf{y} = \mathbf{F}(\mathbf{y}, t-1) + \check{\mathbf{y}}^{K-1}, \\ \dots \dots \dots \\ \mathbf{y} = \mathbf{F}(\mathbf{y}, 1) + \check{\mathbf{y}}^1, \end{cases} \quad (5.92)$$

з операторами $\{\mathbf{F}(\mathbf{y}, t), t=1, \dots, K\}$, які визначені на прямому добутку банахових просторів $\mathbf{B} = \prod_{j=1}^n \mathbf{B}_j$ з векторною нормою:

$$\rho(\mathbf{y}) = (\|\mathbf{y}_1\|_1, \dots, \|\mathbf{y}_n\|_n)^\Gamma, \quad \mathbf{y}_j \in \mathbf{B}_j, \quad j=1, \dots, n,$$

де $\|\cdot\|_j$ — норма простору $\mathbf{B}_j, j=1, \dots, n$.

Побудуємо ітераційний процес розв'язування послідовності рівнянь (5.92), використовуючи правило [5]. Введемо множину

упорядкованих послідовностей $\mathbf{S} = \{\mathbf{S}_j\}_{j=1}^n$, де $\mathbf{S}_j = \{1, 2, \dots, K\}$.

Через $s_j(k) \in \mathbf{S}_j$ позначимо номер часового шару, на якому знаходиться компонента $\mathbf{y}_j^{s_j(k)}(k)$ під час ітераційного кроку k .

Для заданих початкового наближення $\mathbf{y}^{S(0)}(0) = \mathbf{y}(0) \in \mathbf{B}$, точності $\varepsilon > 0$ та хаотичної послідовності множин $\{\mathbf{J}_k\}_{k=1}^\infty$ при $k=1, 2, \dots$ визначатимемо компоненти вектора $\mathbf{y}(k)$ з виразів:

$$\mathbf{y}_j^{s_j(k)}(k) = \begin{cases} \mathbf{y}_j^{s_j(k-1)}(k-1), & j \notin \mathbf{J}_k, \\ \mathbf{F}_j(\mathbf{y}_j^{s_j(k-1)}(k-1), s_j(k-1) + \\ + \mathbf{y}_j^{s_j(k-\delta_i(k))}(k-\delta_i(k))), & j \in \mathbf{J}_k \end{cases} \quad (5.93)$$

$$s_j(k) = \begin{cases} s_j(k-1), & j \notin \mathbf{J}_k, \\ \begin{cases} s_j(k-1), \\ \|\mathbf{y}_j^{s_j(k)}(k) - \mathbf{y}_j^{s_j(k-1)}(k-1)\| > \varepsilon, \quad j \in \mathbf{J}_k, \end{cases} \\ s_j(k-1) + 1, \\ \|\mathbf{y}_j^{s_j(k)}(k) - \mathbf{y}_j^{s_j(k-1)}(k-1)\| \leq \varepsilon, \quad j \in \mathbf{J}_k, \end{cases} \quad (5.94)$$

$$\{\mathbf{y}^{S(k)}(k)\}_{k=1}^\infty = \{\mathbf{y}_1^{s_1(k)}(k), \dots, \mathbf{y}_n^{s_n(k)}(k)\}_{k=1}^\infty,$$

за умов:

$$s_j(k) \leq K, \quad j=1, \dots, n \quad (5.95)$$

$$s_j(0) = 1, \quad j=1, \dots, n, \quad (5.96)$$

$$\delta_j(k) = \min_{i=1,n} \left\{ m : \left\| \mathbf{y}_j^{s_j(k-m)}(k-m) - \mathbf{y}_j^{s_j(k-m-1)}(k-m-1) \right\|_j \leq \varepsilon \right\}. \quad (5.97)$$

Даний метод є узагальненням локально-асинхронного методу, оскільки кожна компонента вектора ітерації обчислюється незалежно від решти компонент. Істотна відмінність його від уже відомих асинхронних методів полягає у застосуванні спеціального механізму асинхронізації обчислень на часових шарах. Тому метод складається з двох взаємопов'язаних правил формування послідовностей (5.93), (5.94) та умов (5.95), (5.96), (5.97).

Ітераційна схема (5.93) задає загальний цикл формування компоненти вектора $\mathbf{y}_j^{s_j(k)}(k) \in \mathbf{y}(k)$ з використанням принципів повної асинхронності. Для даної ітераційної схеми реалізований механізм покомпонентного переходу до наступного часового шару, який заданий виразом (5.94).

Цей механізм базується на локальній оцінці нерівності:

$$\left\| \mathbf{y}_j^{s_j(k)}(k) - \mathbf{y}_j^{s_j(k-1)}(k-1) \right\|_j \leq \varepsilon. \quad (5.98)$$

Якщо нерівність (5.98) виконується, то індекс $s_j(k)$, який відображає поточний часовий шар для компоненти $\mathbf{y}_j^{s_j(k)}(k)$, збільшується на одиницю. Оскільки умова (5.98) є локальною, то перехід до наступного часового шару даної компоненти не залежить від переходу інших компонент. Після зростання індексу $s_j(k)$ до величини максимального часового шару K припиняється його збільшення, але обчислювальний процес продовжується до виконання таких умов:

$$s_j(k) = K, \quad j = 1, \dots, n, \quad (5.99)$$

$$\left\| \mathbf{y}_j(k) - \mathbf{y}_j(k - \delta_j(k)) \right\|_j \leq \varepsilon, \quad j = 1, \dots, n. \quad (5.100)$$

Величина $\delta_j(k)$ є характеристикою запізнення при обчисленні компоненти вектора $\mathbf{y}_j(k)$. Величина запізнення $\mathbf{y}_j(k)$ має задовольняти умову (5.97) на кожному ітераційному кроці.

Метод (5.93), (5.94) не дозволяє отримати розв'язок послідовності операторних рівнянь (5.92) на проміжному часовому шарі, оскільки асинхронний принцип обчислень допускає одночасне існування компонент ітераційного вектора на різних часових шарах.

Достатні умови збіжності локально-асинхронного ітераційного методу розв'язування послідовності операторних рівнянь задає теорема [5].

Теорема 5.22. Нехай $\{\mathbf{F}(\mathbf{y}, t), t = 1, \dots, K\}$ — послідовність p -стискаючих операторів, визначених на прямому добутку кінцевого числа банахових просторів зі спільною матрицею стискування \mathbf{L} , і хаотична послідовність множин $\{\mathbf{J}_k\}_{k=1}^{\infty}$ має максимальну остачу.

Тоді послідовність ітерацій, побудованих по асинхронному ітераційному методу (5.93)–(5.97), збігається у \mathbf{B} для довільного початкового наближення $\mathbf{y}(0) \in \mathbf{B}$.

Умова парастискування призводить до істотних обмежень на кроки просторово-часової сітки, на якій задана різницєва крайова задача. Тому, подібно до методу, який описано у попередньому параграфі, в [5] введено ослаблення умов збіжності шляхом застосування монотонних операторів.

Нехай $\{\mathbf{F}(\mathbf{y}, t), t = 1, \dots, K\}$ — множина операторів, які визначені на прямому добутку скінченного числа гільбертових просторів $\mathbf{H} = \prod_{j=1}^n \mathbf{H}_j$ і нехай існують від'ємні константи $\gamma_1(t), \gamma_2(t), t = 1, \dots, K$, які задовольняють такі умови:

$$\forall \mathbf{y}, \mathbf{z} \in \mathbf{H} : \langle \mathbf{F}(\mathbf{y}, t) - \mathbf{F}(\mathbf{z}, t), \mathbf{y} - \mathbf{z} \rangle \leq \gamma_1(t) \|\mathbf{y} - \mathbf{z}\|^2, \quad (5.101)$$

$$\|\mathbf{F}(\mathbf{y}, t) - \mathbf{F}(\mathbf{z}, t)\|^2 \leq \gamma_2 \langle \mathbf{F}(\mathbf{y}, t) - \mathbf{F}(\mathbf{z}, t), \mathbf{y} - \mathbf{z} \rangle. \quad (5.102)$$

Виходячи з умов сильної монотонності ітераційних операторів, нерівності (5.100), (5.101) мають вигляд:

$$\forall \mathbf{y}, \mathbf{z} \in \mathbf{H} : \|\mathbf{F}(\mathbf{y}, t) - \mathbf{F}(\mathbf{z}, t)\| \geq -\gamma_1(t) \|\mathbf{y} - \mathbf{z}\|, \quad (5.103)$$

$$\forall \mathbf{y}, \mathbf{z} \in \mathbf{H} : \|\mathbf{F}(\mathbf{y}, t) - \mathbf{F}(\mathbf{z}, t)\| \leq -\gamma_2 \|\mathbf{y} - \mathbf{z}\|. \quad (5.104)$$

Звідси $\gamma_2(t) \leq \gamma_1(t)$.

Введемо ітераційний оператор:

$$\Phi(\mathbf{y}, t) = \mathbf{y} + \omega(\mathbf{F}(\mathbf{y}, t) - \mathbf{y} + \bar{\mathbf{y}}^t), \quad t = 1, \dots, K, \quad (5.105)$$

з релаксаційним параметром:

$$\omega < \min_{t=1, K} \frac{2}{2 - \gamma_2(t)}. \quad (5.106)$$

Використовуючи умову (5.104) та (5.106), запишемо:

$$\begin{aligned} \|\Phi(\mathbf{y}, t) - \Phi(\mathbf{z}, t)\|^2 &\leq \\ &\leq \left[(1 - \omega)^2 + \omega(2 - 2\omega + \omega\gamma_2(t))\gamma_1(t) \right] \|\mathbf{y} - \mathbf{z}\|^2, \quad t = 1, \dots, K. \end{aligned}$$

Враховуючи $(2 - 2\omega - \omega\gamma_2(t)) > 0, t = 1, \dots, K$, отримуємо умову збіжності оператора Φ :

$$\|\Phi(\mathbf{y}, t) - \Phi(\mathbf{z}, t)\|^2 \leq \rho(\omega, t) \|\mathbf{y} - \mathbf{z}\|^2, \quad t = 1, \dots, K,$$

де $\rho^2(\omega, t) = (1 - \omega)^2 + \omega(2 - 2\omega + \omega\gamma_2(t))\gamma_1(t)$, $t = 1, \dots, K$.

Асинхронний ітераційний метод розв'язування послідовності операторних рівнянь:

$$\begin{cases} \mathbf{y} = \Phi(\mathbf{y}, t) + \bar{\mathbf{y}}^K, \\ \mathbf{y} = \Phi(\mathbf{y}, t-1) + \bar{\mathbf{y}}^{K-1}, \\ \dots \dots \dots \\ \mathbf{y} = \Phi(\mathbf{y}, 1) + \bar{\mathbf{y}}^1, \end{cases} \quad (5.107)$$

для заданих $\mathbf{y}(0) \in \mathbf{H}, \varepsilon > 0$ та при хаотичній послідовності множин $\{\mathbf{J}_k\}_{k=1}^{\infty}$ має вигляд:

$$\mathbf{y}_j^{s_j(k)}(k) = \begin{cases} \mathbf{y}_j^{s_j(k-1)}(k-1), & j \notin \mathbf{J}_k, \\ (1 - \omega)\mathbf{y}_j^{s_j(k-1)}(k-1) + \\ + \omega(\varphi_j(\mathbf{y}^{s(k-1)}(k-1), s_j(k-1)) + \\ + \mathbf{y}_j^{s_j(k-\delta_i(k))}(k-\delta_i(k))), & j \in \mathbf{J}_k, \end{cases} \quad (5.108)$$

$$s_j(k) = \begin{cases} s_j(k-1), j \notin \mathbf{J}_k, \\ s_j(k-1), \\ \left\| \mathbf{y}_j^{s_j(k)}(k) - \mathbf{y}_j^{s_j(k-1)}(k-1) \right\| > \varepsilon, j \in \mathbf{J}_k, \\ s_j(k-1) + 1, \\ \left\| \mathbf{y}_j^{s_j(k)}(k) - \mathbf{y}_j^{s_j(k-1)}(k-1) \right\| \leq \varepsilon, j \in \mathbf{J}_k. \end{cases} \quad (5.109)$$

$$s_j(k) \leq K, \quad j = 1, \dots, n, \quad (5.110)$$

$$s_j(0) = 1, \quad j = 1, \dots, n, \quad (5.111)$$

$$\delta_j(k) = \min \left\{ m : \left\| \mathbf{y}_j^{s_j(k-m)}(k-m) - \mathbf{y}_j^{s_j(k-m-1)}(k-m-1) \right\| \leq \varepsilon \right\}. \quad (5.112)$$

Умови збіжності даного методу визначає теорема 5.23.

Теорема 5.23. Нехай виконуються задані умови (5.101), (5.102), (5.106), хаотична послідовність має максимальну остачу, і заданий наступний діапазон зміни релаксаційного параметра:

$$\frac{\left(1 - \gamma_1(t) - \sqrt{(1 - \gamma_1(t))^2 - (1 - n^{-2}) (1 - 2\gamma_1(t) + \gamma_1(t)\gamma_2(t))}\right)}{(1 - 2\gamma_1(t) + \gamma_1(t)\gamma_2(t))} <$$

$$< \frac{\left(1 - \gamma_1(t) + \sqrt{(1 - \gamma_1(t))^2 - (1 - n^{-2}) (1 - 2\gamma_1(t) + \gamma_1(t)\gamma_2(t))}\right)}{(1 - 2\gamma_1(t) + \gamma_1(t)\gamma_2(t))} \quad (5.113)$$

Тоді послідовність ітерацій, побудованих за асинхронним ітераційним методом (5.108)–(5.112), збігається в \mathbf{H} до єдиного розв'язку задачі (5.107)

З умови (5.106) випливає, що $\rho(\omega, t) < 1, t = 1, \dots, K$. Для кожного оператора $\Phi(y, t)$ існує своє оптимальне значення $\rho(\omega, t)$. Тому при розв'язуванні послідовності рівнянь (5.107) необхідно добирати значення параметра ω . При реалізації даного методу на нейронних мережах процес такого підбору є основою алгоритму навчання нейронної мережі.

5.5. Ітераційні методи розв'язування крайових задач математичної фізики

5.5.1. Базові поняття

Для крайових задач математичної фізики, що не мають аналітичного розв'язання, застосовують дискретизацію, яка складається з двох основних етапів[14]:

- заміна області зміни неперервного аргументу дискретною множиною точок (сіткою);
- заміна диференціального рівняння різницеvim рівнянням.

Розмірність сітки визначається розмірністю крайової задачі. Наприклад, для одновимірного відрізка $0 \leq x \leq 1$ з кроком h сітку задають множиною вузлів $\mathbf{G}_h = \omega_h \cup \gamma_h$, де

$\omega_h = \{x_i = ih, i = 1, \dots, N - 1, h = 1/N\}$ — множина внутрішніх вузлів; $\gamma_h = \{x_i = ih, i = 0, N, h = 1/N\}$ — множина граничних вузлів. На даній множині неперервну функцію $y(x)$ задають сітковою функцією $y(x_i) = y_i$, де $x_i, i = 0, 1, \dots$, — значення x у вузлах сіткової області. Оскільки довільна сіткова функція може бути представлена у вигляді вектора з координатами, що відповідають її значенням у вузлах сітки, то множина таких функцій завжди утворює скінченновимірний простір \mathbf{H} .

Диференціальний оператор замінюють різницеvim оператором, який діє в просторі. Процес такої заміни називають різницевою апроксимацією диференціального оператора.

Нехай область $\mathbf{G} = \Omega \cup \Gamma$ складається з внутрішнього простору $\Omega \in \mathbf{R}^n$ та границі $\Gamma \in \mathbf{R}^n$. На \mathbf{G} заданий диференціальний оператор $L: \mathbf{R}^n \rightarrow \mathbf{R}^n$, який діє на функцію $u(x), x \in \Omega$

Покриємо область \mathbf{G} сіткою \mathbf{G}_h і будемо розглядати сіткову функцію $u^h(x)$, визначену у вузлах сіткової області, які відповідають значенням аргументу $x_i \in \omega_h$. Тоді різницева апроксимація полягає у заміні Lu в деякій точці $x_i \in \omega_h$ лінійною комбінацією значень функції u^h у вузлах шаблону.

$$\left(L_h u^h\right)_i = \sum_{x_j \in \sigma_i} a_{ij}^h u^h(x_j), \quad (5.114)$$

де a_{ij}^h — коефіцієнти лінійної комбінації, що задають характер дії різницевого оператора; σ_i — множина значень аргументу, які формують апроксимаційний шаблон для вузла i .

Похибкою апроксимації оператора L оператором L_h називають різницю:

$$\Delta = L_h u^h - Lu. \quad (5.115)$$

Зазвичай диференціальне рівняння

$$Lu(x) = f(x) \quad (5.116)$$

розв'язують із деякими додатковими умовами. До таких умов відносять задані початкові значення:

$$u_0 = u(x_0), \quad x_0 \in \Omega, \quad (5.117)$$

та значення на границях області:

$$u(x) = \mu(x), \quad x \in \Gamma. \quad (5.118)$$

Відповідну задачу називають задачею Коші або крайовою задачею. Перехід від крайової задачі (5.116)–(5.118) до різницевої крайової задачі відбувається шляхом заміни диференціального оператора Lu на різницевий оператор $L_h u^h$, а правої частини $f(x)$ — відповідною сітковою функцією $\varphi_h(x)$. В результаті отримують систему різницевих рівнянь:

$$L_h u(x) = \varphi_h(x), \quad x \in G_h. \quad (5.119)$$

Нехай оператор L_h заданий у вигляді матриці коефіцієнтів системи різницевих рівнянь (5.119). Тоді, представивши його у вигляді різниці матриць $L_h = B_h - C_h$, для розв'язання (5.119) можна застосувати залежність:

$$B_h u_{k+1}^h = C_h u_k^h + \varphi_h. \quad (5.120)$$

Звідси:

$$\begin{aligned} u_{k+1}^h &= B_h^{-1} (C_h u_k^h + \varphi_h) = B_h^{-1} ((B_h - L_h) u_k^h + \varphi_h) = \\ &= u_k^h - B_h^{-1} (L_h u_k^h - \varphi_h), \end{aligned} \quad (5.121)$$

де $k = 0, 1, \dots$ — номер ітерації; $\forall u_0^h \in H$; $L_h : H \rightarrow H$ — різницевий оператор; $B_h : H \rightarrow H$ — лінійний оператор, що має зворотний оператор B_h^{-1} .

Ітераційна схема (5.121) має назву методу простої ітерації і застосовується не тільки для систем різницевих рівнянь, а й для довільних систем лінійних алгебраїчних рівнянь. Така ітераційна

схема широко застосовується в паралельних асинхронних обчислювальних структурах [142].

5.5.2. Локально-асинхронний метод для рекурентних нейронних мереж

Даний метод базується на принципах локальності та асинхронності, які застосовувались при створенні локально-асинхронних методів [5,9], орієнтованих на застосування у паралельних обчислювальних структурах. Вони також є актуальними і для нейронних мереж, оскільки впливають із природних потреб ефективного завантаження кожного елемента мережі при необхідності обмеження міжнейронних зв'язків [8]. Нейронну мережу, орієнтовану на даний метод [11], можна розглядати як скінченну сукупність H елементів, що реалізують глобальний оператор F , який складається з множини парастискаючих операторів:

$$F = \{F_h | h \in H\}.$$

Означення 5.30. Глобальний оператор $F : R^n \rightarrow R^n$ називають парастискаючим [70], якщо він неперервний, і для нерухомої точки η та довільної точки $x \in R^n$ справедлива нерівність:

$$\|F(x) - \eta\| < \|x - \eta\|, \quad (5.122)$$

за умови, що $x \neq \eta$.

Асинхронні паралельні ітераційні методи, розглянуті вище, спираються на поняття хаотичної послідовності, приналежність до якої чергового ітераційного кроку визначається вузлами паралельної обчислювальної структури. Такий підхід потребує відповідних ресурсів для проведення аналізу приналежності до послідовності, яка у загальному випадку є глобальною.

Даний локально-асинхронний метод орієнтований на застосування в нейронних структурах і забезпечує використання

ітераційного оператора виключно при виникненні умов спрацювання нейрона.

Означення 5.31. Нехай задано парастискаючий глобальний оператор $F = \{F_h | h \in H\}$, $F: D^{k_h} \rightarrow D$, де $k_h \in \{1, \dots, k\}$ $\forall h \in H$, $D^{k_h} \subset R^{nk_h}$, $D \subset R^n$, і множину початкових векторів X_0 . Тоді асинхронною ітераційною послідовністю будемо називати послідовність $\{x(t)\}_{t=1}^{\infty}$ векторів $x(t) \in R^n$, які визначаються за ітераційною схемою:

$$x(t+1) = F_{h(t)} \left(x(s_1^h(t)), \dots, x(s_{k_h(t)}^h(t)) \right), \quad t = 1, 2, \dots, \quad (5.123)$$

де $\{h(t) | t = 1, 2, \dots\}$ — послідовність елементів $h(t) \in H$,

$$S^h = \left\{ s_{k_h}^h(t) \right\}_{t=0}^{\infty}, \quad h = 1, \dots, H, \quad k_h = 1, \dots, k, \quad \text{— множини цілих}$$

невід'ємних чисел, які відповідають умовам:

$$\begin{cases} 0 < s_{k_h(t)}^h(t) \leq t, & t > 0, \\ -M \leq s_{k_h(t)}^h(t) \leq 0, & t = 0, \end{cases}$$

$$X_0 = \left\{ x(s_1^h(0)), \dots, x(s_{k_h}^h(0)) \right\}_{h=1}^H \quad \text{— множина початкових}$$

векторів, яка визначається початковим станом нейронної мережі.

Згідно з означеннями 5.10 та 5.11, метод (5.123) може бути повністю асинхронним або частково асинхронним. Якщо оператор $F_{h(t)}(\cdot)$ допускає довільну кількість k_h аргументів з довільною відстанню $s_l^h(t)$ від поточного ітераційного кроку аргументу з номером l , то метод (5.123) називатимемо повністю асинхронним за умов:

- 1) $\lim_{t \rightarrow \infty} s_l^h(t) \rightarrow \infty \quad \forall h(t) \in H, \quad \forall l = 1, \dots, k_{h(t)},$
- 2) $\{h(t)\} \cup \{h(t+1)\} \cup \dots = H.$

Умова 1 вказує, що ітераційна схема методу дозволяє розглядати елементи даних, які віддалені від поточного ітераційного кроку на довільну відстань як аргументи оператора $F_{h(t)}(\cdot)$. Умова 2 забезпечує існування необмеженої допустимої кількості ітерацій. Метод, який відповідає таким умовам, має лише теоретичну цінність. На практиці використовують виключно частково асинхронні методи. Метод (5.122) називатимемо частково асинхронним при виконанні таких умов:

- 1) $s^h = \max_t \left(\max_l (t - s_l^h(t)) \right) \quad \forall h \in H,$
- 2) $\{h(t)\} \cup \{h(t+1)\} \cup \dots \cup \{h(t+c)\} = H, \quad c \in N, \quad \forall t \in N.$

У такому методі вводять обмежувальні константи s^h максимальної відстані від поточного ітераційного кроку та константи c або c_h , які обмежують максимально допустиму кількість ітерацій.

Для аналізу умов стискання даного глобального оператора F будемо використовувати узагальнення означень неперервності [4] операторів, які задані на структурі даних:

$$X = \begin{Bmatrix} x_1^1 & \dots & x_1^{k_h} & \dots & x_1^k \\ \dots & \dots & \dots & \dots & \dots \\ x_n^1 & \dots & x_n^{k_h} & \dots & x_n^k \end{Bmatrix},$$

де $X \in R^{nk}$; $x^i \in R^n$ при $i = 1, \dots, k$.

Означення 5.32. Нехай $F = \{F_h | h \in H\}$ — глобальний оператор, який задає відображення $F: D^{k_h} \rightarrow D$, і існує множина натуральних чисел $\{k_h\}_{h=1}^H$ таких, що відповідають умові $k_h \leq k \quad \forall h \in H$. Тоді множину функціоналів $\{f_h: D^{k_h} \rightarrow R\}$ називатимемо рівномірно неперервною

на $X = \{x^1, \dots, x^k\} \in \mathbf{D}^k$, якщо для довільного $\varepsilon > 0$ існує таке $\delta > 0$, для якого за умови $\max_{i=1, \dots, k} \|y^i - x^i\| < \delta$

виконується нерівність:

$$\left| f_h(y^1, \dots, y^{k_h}) - f_h(x^1, \dots, x^{k_h}) \right| < \varepsilon \quad \forall h \in \mathbf{H}.$$

Означення 5.33. Нехай $X, Y \in \mathbf{D}^{k_h}$, і існує коефіцієнт $0 \leq \beta < 1$. Тоді за умови

$$\|F_h(X) - F_h(Y)\| \leq \beta \max_i \|x^i - y^i\| \quad \forall h \in \mathbf{H},$$

F називатимемо стискаючим в \mathbf{D} .

Означення 5.34. Нехай $X \in \mathbf{D}^{k_h}$, і оператор F_h — неперервний в \mathbf{D}^{k_h} . Тоді F є неперервно парастискаючим в \mathbf{D} [69], якщо для довільної нерухомої точки $\eta \in \mathbf{R}^n$ з F_h справедлива нерівність:

$$\|F_h(X) - \eta\| < \max_i \|x^i - \eta\| \quad \forall h \in \mathbf{H},$$

або $X = (x, \dots, x)$ і x є нерухомою точкою з F_h .

Означення 5.35. Якщо для кожного вектора $x \in \mathbf{D}$ існує норма $\|\cdot\|_x$ в \mathbf{R}^n , для якої виконується нерівність $\|z\|_x \leq c \|z\|_y \quad \forall x, y \in \mathbf{D}, z \in \mathbf{R}^n, c \geq 1$, і $\eta \in \mathbf{R}^n$ є нерухомою точкою глобального оператора $F = \{F_h | h \in \mathbf{H}\}$, який задає відображення $F: \mathbf{D}^{k_h} \rightarrow \mathbf{D}$, то такий оператор називають нормовано стискаючим в \mathbf{D} , якщо для всіх $h \in \mathbf{H}_\eta$:

$$\|F_h(X) - \eta\|_\eta \leq \beta_\eta^h(X) \max_i \|x^i - \eta\|_\eta,$$

де $\{\beta_\eta^h(\cdot): \mathbf{D} \rightarrow [0, 1] | h \in \mathbf{H}_\eta\}$ — рівномірно неперервна в \mathbf{D} множина функціоналів, і для всіх $X \in \mathbf{D}^{k_h}$, для яких множина

$$\mathbf{H}_{X, \eta} = \left\{ h \in \mathbf{H}_\eta \mid \exists j \in \{1, \dots, k_h\}: F_h(x^1, \dots, x^j, \dots, x^{k_h}) \neq x^j \right\}$$

є не пустою множиною, справедлива нерівність

$$\sup_{h \in \mathbf{H}_{X, \eta}} \beta_\eta^h(x^1, \dots, x^{k_h}) < 1.$$

Ітераційний процес може бути представлений у вигляді деякого направленного графа (\mathbf{T}, \mathbf{E}) , який складається з вершин $\mathbf{T} = N_0 \cup \{-1, \dots, -M\}$ та направлених ребер \mathbf{E} . Топологія такого графа задана правилом формування підмножин вхідних направлених ребер для довільної вершини $t \in \mathbf{T}$. Між вершинами p і t існує ребро (p, t) тоді і тільки тоді, коли існує l , $1 \leq l \leq k_{h(t-1)}$, таке, що $p = s_l^h(t-1)$ при $p \geq -M, t \geq 1$.

Сформулюємо умову збіжності даного методу [154] при

$$\mathbf{S}^h = \mathbf{S} = \left\{ s_{k_h}(t) \right\}_{t=0}^\infty, \quad h = 1, \dots, H, \quad k_h = 1, \dots, k.$$

Теорема 5.24. Нехай нормально парастискаючий глобальний оператор F , що визначений на $\mathbf{D} \subset \mathbf{R}^n$, має нерухому точку $\eta \in \mathbf{D}$. Тоді частково асинхронний ітераційний метод збігається до спільної нерухомої точки η , якщо для графа (\mathbf{T}, \mathbf{E}) ітераційного процесу існують числа $a_0, b \in \mathbf{N}$ та послідовність $\{b_p | b_p = a_0, a_0 + 1, \dots\}$, які при $p \geq a_0$ задовольняють умови:

- 1) для кожної вершини з номером $t \geq p$ існує напрямлене ребро з b_p в t ;
- 2) $p - b_p \leq b$;
- 3) для довільного $h(t) \in \mathbf{H}$ існує число $c \in \mathbf{N}$ таке, що при $p \geq a_0$ завжди можна знайти вершину t_p^h , яка є спадкоємицею b_p і попередницею b_{p+c} .

Оскільки множина \mathcal{S} задає довільний сценарій для методу (5.123), то ітераційний процес зі сценарієм \mathcal{S}^h може бути зведений до сценарію \mathcal{S} . Отже, теорема 5.24 є справедливою також для методу (5.123).

За умови нормованого парастискання неможливо довести умову достатності, яка полягає у тому, що ітераційний метод (5.123) збігається тільки у випадку, коли існує нерухома точка $\eta \in \mathcal{D}$. Такий висновок базується на факті, що глобальний оператор F у загальному випадку не є неперервним. При накладанні умови неперервності на оператор F та скасування обмеження на допустиму кількість ітерацій доведення умови достатності зводиться до вже відомого випадку [70].

5.5.3. Ітераційний метод адитивної корекції для кліткових нейронних мереж

Клітковими нейронними мережами називають однорідні багатопарові структури з локальними зв'язками на кожному шарі [56,55]. У [88] представлений метод розв'язування диференціальних рівнянь з частинними похідними на клітковій нейронній мережі за умови, що її поведінку описує система диференціальних рівнянь:

$$C \frac{dX}{dt} = -DX + Tg(X) + I, \quad (5.124)$$

де $X = (x_1, \dots, x_n)^T$ — вектор, що задає стан нейронної мережі; $D = \text{diag}\{d_{11}, \dots, d_{nn}\}$ — діагональна матриця з елементами $d_{ii} = \text{const} > 0$; T — матриця міжнейронних зв'язків з елементами $t_{ij} = t_{ji}$, $t_{ii} = 0$; $g(X) = (g_1(x_1), \dots, g_n(x_n))^T$ — функція активації нейронів; $I = (i_1, \dots, i_n)^T$ — вектор додаткових входів; $C = \{c_1, \dots, c_n\}$ — вектор вхідних параметрів.

Нехай необхідно розв'язати систему:

$$AX = F, \quad (5.125)$$

де A — квадратна матриця; $X = (x_1, \dots, x_n)^T$ — вектор невідомих; $F = (f_1, \dots, f_n)^T$ — вектор правої частини.

При постановці задачі (5.125) на клітковій мережі потрібно задати параметри $D, T, I, g(X)$ таким чином, щоб стаціонарний стан нейронної мережі відповідав розв'язку.

Нехай матриця A є додатно визначеною. Тоді задамо параметри нейронної мережі, виходячи із співвідношень:

$$d_{ii} = a_{ii}, t_{ij} = a_{ij}, i_i = f_i, g_i(x_i) = -x_i \text{ при } i, j = 1, \dots, n. \quad (5.126)$$

З урахуванням (5.126) рівняння (5.124) перепишемо у вигляді:

$$F - AX = R, \quad (5.127)$$

де R — нев'язка системи (5.125).

Для розв'язування системи (5.127) застосовують ітераційний метод адитивної корекції. За цим методом кожний ітераційний крок супроводжується обчисленням вектора нев'язки:

$$R^{(k)} = F - AX^{(k)}. \quad (5.128)$$

З урахуванням (5.128) знаходять чергову корекцію розв'язку системи (5.125):

$$B \Delta X^{(k+1)} = R^{(k)}, \quad (5.129)$$

де B — матриця, яка задана параметрами нейронної мережі.

Ітераційна формула для одержання наближеного значення розв'язку системи (5.125):

$$X^{(k+1)} = X^{(k)} + \Delta X^{(k+1)}. \quad (5.130)$$

Необхідну та достатню умову збіжності даного методу задають виразом:

$$[(2B - A)X, X] > 0, \quad (5.131)$$

при $X \neq 0$.

Загальне представлення даного методу дозволяє застосовувати його для реалізації процесу розв'язування систем лінійних різницевих рівнянь на нейронних мережах Хопфілда.

Розділ 6

МУЛЬТИСІТКОВІ МЕТОДИ

6.1. Двосіткові ітераційні методи

Мультисіткові методи можна вважати новою технологією, яка дозволяє покращувати параметри вже відомих паралельних асинхронних методів. Основи даної технології закладені ще на початку 1960-х років в [72], де був описаний мультисітковий алгоритм розв'язування рівняння Пуассона на стандартному п'ятиточковому шаблоні. Цей метод перевідкрив W. Hackbush, який у [92,93,94] дав його строге математичне обґрунтування і запропонував ряд нових підходів. Подальше стрімке зростання досліджень мультисіткових методів забезпечило їх широке застосування у різних сферах, включаючи і нейронні мережі. Нейронні мультисіткові методи, започатковані в [25,26], базуються на застосуванні правил навчання нейронних мереж в умовах наявності асинхронних принципів взаємодії між нейронами.

Розглянемо систему різницевих рівнянь

$$\mathbf{L}_h \mathbf{u}_h(\mathbf{x}) = \mathbf{f}_h(\mathbf{x}), \mathbf{x} \in \mathbf{G}, \quad (6.1)$$

яка є апроксимацією крайової задачі $\mathbf{L}\mathbf{u} = \mathbf{f}$ на сітці $\mathbf{G} = \omega \cup \gamma$, де $\omega_h = \{x_i = ih, i = 1, \dots, N-1, h = 1/N\}$ — множина внутрішніх вузлів; $\gamma_h = \{x_i = ih, i = 0, N, h = 1/N\}$ — множина граничних вузлів.

Нехай $\mathbf{u}_h(n)$ — апроксимація розв'язку \mathbf{u}_h , який відповідає ітераційному кроку n . Тоді $\mathbf{v}_h(n)$ називають помилкою такої апроксимації за умови:

$$\mathbf{v}_h(n) = \mathbf{u}_h - \mathbf{u}_h(n), \quad (6.2)$$

а нев'язку задають у вигляді різниці

$$\mathbf{d}_h(n) = \mathbf{f}_h - \mathbf{L}_h \mathbf{u}_h(n). \quad (6.3)$$

Використовуючи (6.2) та (6.3), запишемо рівняння для нев'язки:

$$\mathbf{L}_h \mathbf{v}_h(n) = \mathbf{d}_h(n), \quad (6.4)$$

що є еквівалентним рівнянню (6.1).

Введемо спрощений оператор $\widehat{\mathbf{L}}_h$, для якого завжди визначений обернений оператор $\widehat{\mathbf{L}}_h^{-1}$. Тоді ітераційний процес для визначення чергового наближення $\mathbf{v}_h(n)$ можна задати рівнянням

$$\mathbf{v}_h(n+1) = (\mathbf{I}_h - \widehat{\mathbf{L}}_h^{-1} \mathbf{L}_h) \mathbf{v}_h(n), \quad (6.5)$$

де $(\mathbf{I}_h - \widehat{\mathbf{L}}_h^{-1} \mathbf{L}_h): \mathbf{H}_h \rightarrow \mathbf{H}_h$ — оператор переходу, \mathbf{I}_h — тотожний оператор.

Аналогічно можна задати ітераційний процес для визначення нев'язки:

$$\mathbf{d}_h(n+1) = \mathbf{L}_h (\mathbf{I}_h - \widehat{\mathbf{L}}_h^{-1} \mathbf{L}_h) \mathbf{L}_h^{-1} \mathbf{d}_h(n) = (\mathbf{I}_h - \widehat{\mathbf{L}}_h^{-1} \mathbf{L}_h) \mathbf{d}_h(n). \quad (6.6)$$

Розглянемо можливість апроксимації даної задачі на множині сіток $\mathbf{G}^{(0)}, \mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(M)}$, починаючи з деякої базової сітки $\mathbf{G}^{(0)}$, яка характеризується найменшим кроком дискретизації h_0 . Величину кроку дискретизації для довільної сітки $\mathbf{G}^{(i)}$ можна записати:

$$h_i = k^{(i)} h_{i-1}, \quad (6.7)$$

де $k^{(i)}$ — коефіцієнт зростання кроку дискретизації.

У більшості випадків $k^{(i)} = k = 2$.

Перехід від сітки $\mathbf{G}^{(i)}$ до сітки $\mathbf{G}^{(i+1)}$ відбувається за допомогою лінійного оператора рестрикції

$$\mathbf{R}_i^{i+1}: \mathbf{H}_i \rightarrow \mathbf{H}_{i+1}, \quad (6.8)$$

а зворотний перехід задає лінійний оператор пролонгації P_{i+1}^i :

$$P_{i+1}^i : H_{i+1} \rightarrow H_i. \quad (6.9)$$

Для параметрів $d^{(i)}(n)$ та $v^{(i)}(n)$, що відповідають значенням нев'язки та похибки у вузлах сітки $G^{(i)}$, оператор R_i^{i+1} задає перехід:

$$d^{(i+1)}(n) = R_i^{i+1} d^{(i)}(n), \quad (6.10)$$

а оператор P_{i+1}^i :

$$v^{(i)}(n) = P_{i+1}^i v^{(i+1)}(n). \quad (6.11)$$

Розглянемо послідовність мультисіткових обчислень на прикладі методу дворівневої корекції [73], заданого на послідовності сіток $G^{(0)}, G^{(1)}$.

Позначимо через $S[u^{(0)}(n), L^{(0)}, f^{(0)}]$ ітераційну схему на сітці $G^{(0)}$, а через $S[v^{(i)}(n), L^{(i)}, d^{(i)}(n)]$ — ітераційну схему на довільній грубій сітці $G^{(i)}$, $i > 0$, на ітераційному кроці n . Тоді наступна послідовність операцій складатиме алгоритм реалізації згаданого методу.

1. Згладжування на сітці $G^{(0)}$:

$$u^{(0)}(n) = S[u^{(0)}(n), L^{(0)}, f^{(0)}]. \quad (6.12)$$

2. Обчислення нев'язки на точній сітці:

$$d^{(0)}(n) = f^{(0)} - L^{(0)} u^{(0)}(n). \quad (6.13)$$

3. Операція рестрикції на грубу сітку:

$$d^{(1)}(n) = R_0^1 d^{(0)}(n). \quad (6.14)$$

4. Згладжування похибки на сітці $G^{(1)}$:

$$v^{(1)}(n) = S[v^{(1)}(n), L^{(1)}, d^{(1)}(n)]. \quad (6.15)$$

5. Пролонгація похибки:

$$v^{(0)}(n) = P_1^0 v^{(1)}(n). \quad (6.16)$$

6. Обчислення нової апроксимації:

$$u^{(0)}(n+1) = u^{(0)}(n) + v^{(0)}(n). \quad (6.17)$$

7. Повторення пунктів 1–6 до досягнення збіжності.

Даний підхід став базовим для подальших досліджень мультисіткових методів з метою покращення швидкості збіжності. Такі дослідження охоплюють ряд основних напрямків:

- застосування нелінійних операторів рестрикції R_i^{i+1} та пролонгації P_{i+1}^i ;
- використання більш точних методів дискретизації для рівнянь із частинними похідними;
- одержання точного розв'язку на грубій сітці;
- використання складних ітераційних схем згладжування.

6.2. Мультисіткові ітераційні методи

Природним узагальненням методу дворівневої кореляції є перехід від двох сіток $G^{(0)}, G^{(1)}$ до використання деякої послідовності сіток $G^{(0)}, G^{(1)}, \dots, G^{(M)}$, де M — номер сітки, яка містить лише одну точку.

На рис.6.1 показано приклад сіткової послідовності, яка складається з чотирьох сіток і характеризується типовим коефіцієнтом зростання кроку дискретизації $h_i = 2h_{i-1}, 0 < i < 4$. Сітку $G^{(0)}$, яка характеризується найменшим кроком

дискретизації, називають точною сіткою, а множина $\{\mathbf{G}^{(i)} | i > 0\}$ містить грубі сітки.

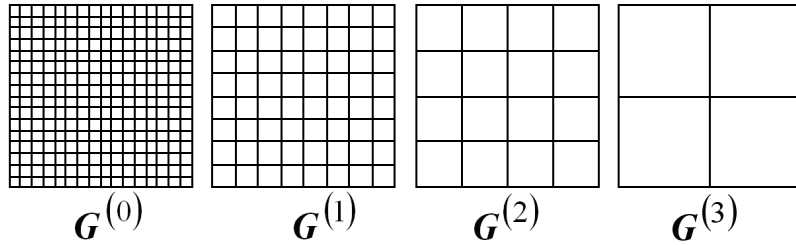


Рис.6.1. Послідовність сіток.

Оскільки обчислення на грубих сітках потребують значно меншої кількості ресурсів, ніж обчислення на точній сітці, мультисіткові методи містять, як правило, значну кількість переходів на сітках з великим кроком дискретизації. Алгоритми переходу між елементами заданої сіткової послідовності називають циклами, структуру яких визначають за коефіцієнтом циклічності γ .

На рис.6.2. показані V -цикли та W -цикли, які характеризуються коефіцієнтами циклічності $\gamma = \{1, 2, 4\}$, що вказують на те, скільки разів у даному циклі досягають максимально грубої сітки

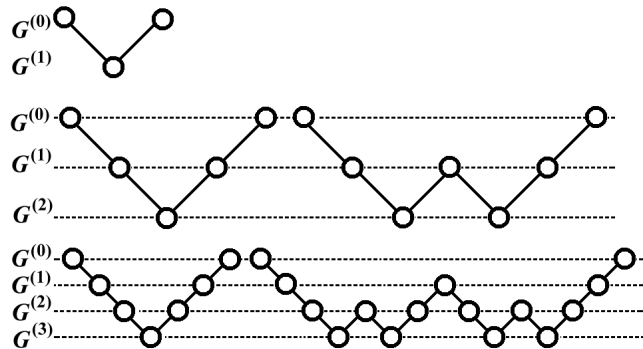


Рис. 6.2. V -цикли та W -цикли

Нехай задано систему різницевих рівнянь (6.1), яку необхідно розв'язати на послідовності сіток $\{\mathbf{G}^{(i)} | i = 0, 1, \dots\}$ з кроками $h_i, i = 0, 1, \dots, h_{i_1} > h_{i_2}$, де $i_1 > i_2$. Для цього визначимо лінійні оператори (6.8), (6.9) та обернений оператор

$$\mathbf{L}^{(i)}: \mathbf{H}_i \rightarrow \mathbf{H}_i. \quad (6.18)$$

Багатосітковий метод у загальному випадку містить таку послідовність етапів: перше згладжування, уточнення та друге згладжування.

Перше згладжування полягає у виконанні ν_1 разів ітераційної схеми

$$\mathbf{u}^{(0)}(n) = \mathbf{S}^{\nu_1} [\mathbf{u}^{(0)}(n), \mathbf{L}^{(0)}, \mathbf{f}^{(0)}] \quad (6.19)$$

на сітці $\mathbf{G}^{(0)}$. На етапі уточнення виконується алгоритм, який складається з переходів між сітками за схемою V -циклу або W -циклу та виконання ітераційного процесу розв'язування рівняння нев'язки на кожній з грубих сіток.

Обчисливши нев'язку на сітці $\mathbf{G}^{(0)}$ за формулою (6.13), здійснюємо її пролонгацію на сітку $\mathbf{G}^{(1)}$, використовуючи загальну формулу пролонгації (6.10), яку будемо застосовувати завжди при передачі значень нев'язки у вузли сітки $\mathbf{G}^{(i+1)}$ від сітки $\mathbf{G}^{(i)}$.

Похибку $\mathbf{v}^{(i)}(n)$ на сітці $\mathbf{G}^{(i)}$ задають рівнянням нев'язки

$$\mathbf{L}^{(i)} \mathbf{v}^{(i)}(n) = \mathbf{d}^{(i)}(n) \quad (6.20)$$

і визначають шляхом застосування ν_2 разів ітераційної схеми

$$\mathbf{v}^{(i)}(n) = \mathbf{S}^{\nu_2} [\mathbf{v}^{(i)}(n), \mathbf{L}^{(i)}, \mathbf{d}^{(i)}(n)]. \quad (6.21)$$

Згідно з вибором типу циклу, за яким виконується ітерація мультисіткового методу, операції пролонгації чергуються з операціями інтерполяції (6.11), які визначають значення похибки на більш точній сітці. Останнім кроком етапу уточнення завжди

є корекція апроксимації $\mathbf{u}^{(0)}(n)$ на сітці $\mathbf{G}^{(0)}$ за формулою (6.17).

Етап другого згладжування полягає у застосуванні ν_3 разів ітераційної схеми (6.19) до скоригованого на етапі уточнення значення $\mathbf{u}^{(0)}(n)$:

$$\mathbf{u}^{(0)}(n+1) = \mathbf{S}^{\nu_3} \left[\mathbf{u}^{(0)}(n) + \mathbf{v}^{(0)}(n), \mathbf{L}^{(0)}, \mathbf{f}^{(0)} \right]. \quad (6.22)$$

Існує велика кількість методик вибору параметрів ν_1, ν_2, ν_3 , які залежать від конкретного виду крайової задачі.

6.2.1. Узагальнення ітераційної схеми згладжування

Найпоширенішими ітераційними схемами, які застосовуються на етапах згладжування, є ітераційні схеми, побудовані за методами Гауса–Зейделя та Якобі.

Згідно з методом Гауса–Зейделя обчислення апроксимації на сітці $\mathbf{G}^{(i)}$ відбувається за рекурсивною формулою:

$$\mathbf{u}^{(i)}(n+1) = (\mathbf{D}^{(i)} - \mathbf{Q}^{(i)})^{-1} (\mathbf{M}^{(i)} \mathbf{u}^{(i)}(n) + \mathbf{f}^{(i)}). \quad (6.23)$$

Діагональну матрицю $\mathbf{D}^{(i)}$, нижню трикутну матрицю без діагональних елементів $\mathbf{Q}^{(i)}$ та верхню трикутну матрицю без діагональних елементів $\mathbf{M}^{(i)}$ одержують шляхом розщеплення матриці $\mathbf{L}^{(i)}$:

$$\mathbf{L}^{(i)} = \mathbf{D}^{(i)} - \mathbf{Q}^{(i)} - \mathbf{M}^{(i)}. \quad (6.24)$$

Метод Гауса–Зейделя часто використовують для послідовних обчислень вузлів сіткової області, оскільки ефективність його застосування залежить від способу обходу даних вузлів. Фіксований обхід задає залежність між компонентами, що звужує можливості розпаралелювання ітераційного процесу. Цей метод може бути розпаралеленим шляхом введення деякого порядку обчислення взаємно незалежних блоків.

Повну незалежність обчислень у вузлах сіткової області забезпечує метод Якобі, ітераційна схема якого має вигляд:

$$\mathbf{u}^{(i)}(n+1) = \mathbf{u}^{(i)}(n) - (\mathbf{D}^{-1})^{(i)} (\mathbf{L}^{(i)} (\mathbf{u}^{(i)}(n) - \mathbf{f}^{(i)})). \quad (6.25)$$

Зробивши еквівалентні перетворення, одержимо другий варіант даної ітераційної схеми:

$$\mathbf{u}^{(i)}(n+1) = (\mathbf{D}^{-1})^{(i)} \left[(\mathbf{A}^{(i)} + \mathbf{B}^{(i)}) \mathbf{u}^{(i)}(n) + \mathbf{f}^{(i)} \right], \quad (6.26)$$

або у загальному вигляді:

$$\mathbf{u}^{(i)}(n+1) = \mathbf{A}^{(i)} (\mathbf{L}^{(i)} \mathbf{u}^{(i)}(n) - \mathbf{f}^{(i)}). \quad (6.27)$$

Існує строге доведення збіжності ітераційної схеми (6.27) у випадку паралельного синхронного обчислення апроксимації $u_i(n)$ в кожній точці сіткової області [15]. Цей підхід потребує засобів синхронізації в ході обчислювального процесу, що знижує його ефективність у випадку застосування однорідних обчислювальних середовищ із великою кількістю елементів. Саме таким середовищем є нейронна мережа. Тому для ефективної організації обчислювального процесу у нейронних структурах доцільно використовувати сучасні принципи введення асинхронності, що лежать в основі створення методу асинхронних ітерацій (5.23). За цим методом ітераційну схему на сітці $\mathbf{G}^{(i)}$ задають такою ітераційною послідовністю:

$$u_j^{(i)}(n+1) = \begin{cases} u_j^{(i)}(n) & \forall n \notin \mathbf{J}_n^{(i)}, \\ \mathbf{A}^{(i)} \left[\mathbf{L}^{(i)} (u_1^{(i)}(s_1(n)), \dots, u_i^{(i)}(s_i(n))) - \mathbf{f}^{(i)} \right] & \forall n \in \mathbf{J}_n^{(i)}, \end{cases} \quad (6.28)$$

де $\mathbf{J}_n^{(i)} = \{ \mathbf{J}_n^{(i)} \}_{n=1}^{\infty}$ — послідовність непустих підмножин

$$\mathbf{J}_n^{(i)} \subset \{1, 2, \dots, i\},$$

$\mathbf{S}^{(i)}(n) = \{ s_k(n) | k = 1, 2, \dots, i \}_{n=1}^{\infty}$ — множини цілих невід'ємних чисел, які задовольняють умови:

$$0 \leq s_k(n) \leq n \quad \forall n \geq 0; \quad \lim_{n \rightarrow \infty} (s_k(n)) = \infty.$$

Для побудови ітераційних схем згладжування можна також застосовувати асинхронні ітераційні методи з нерухомими точками та мультирозщеплюючі ітераційні методи, які були розглянуті в попередньому розділі.

Відомо, що згадані методи зменшують високочастотні компоненти похибки. Тобто, зменшення похибки $e^{(i)}(n) = u - u^{(i)}(n)$ на сітці $G^{(i)}$ замінюють її згладжуванням (рис.6.3).

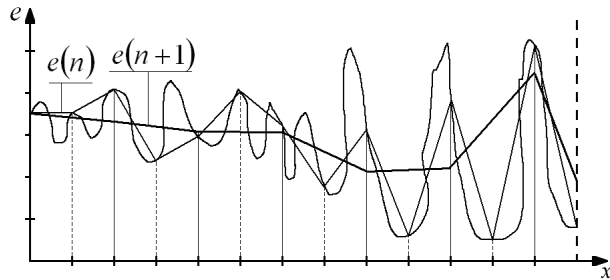


Рис.6.3. Ефект згладжування похибки.

З цієї причини ітераційні методи часто називають методами згладжування.

Наприклад, для ітераційної схеми Якобі на сітці G

$$u(n+1) = u(n) - \omega D^{-1}(Lu - f) \quad (6.29)$$

з демпфіруванням $\omega \leq 1$, власні вектори $A_{(k,l)}^{(\nu,\mu)}$ матриць L та

$S = I - \omega D^{-1}L$ визначають із виразу:

$$A_{(k,l)}^{(\nu,\mu)} = \sin(\pi \nu h k) \sin(\pi \mu h l), \quad \nu, \mu = 1, 2, \dots, m. \quad (6.30)$$

Відповідні власні значення $\lambda^{(\nu,\mu)}(L)$ та $\lambda^{(\nu,\mu)}(S)$:

$$\lambda^{(\nu,\mu)}(L) = 4 - 2 \cos(\pi \nu h) - 2 \cos(\pi \mu h), \quad (6.31)$$

$$\lambda^{(\nu,\mu)}(S) = 1 - \omega - 0.5\omega(\cos(\pi \nu h) + \cos(\pi \mu h)). \quad (6.32)$$

З (6.31),(6.32) можна зробити висновок, що низькочастотні компоненти похибки підлягають невеликому зменшенню навіть при зміні у великому діапазоні коефіцієнта демпфірування ω .

Тому ідея мультисіткових методів полягає в обчисленні похибки у низькочастотному діапазоні, де низькочастотні компоненти похибки зсуваються у високочастотну область поточного діапазону, що збільшує швидкість збіжності.

6.2.2. Пролонгація та рестрикція

Існують два основних способи покриття сіткою області розв'язання крайової задачі: вузловий та домінний. Вузловий спосіб полягає в заміні неперервного значення функції множиною значень, які визначені у вузлах сіткової області:

$$G = \left[\begin{array}{l} x \in \mathbf{R}^n, x = \mathbf{j}h, \mathbf{j} = (j_1, \dots, j_n), h = (h_1, \dots, h_n), \\ j_\alpha = (0, 1, \dots, n_\alpha), h_\alpha = \frac{1}{n_\alpha}, \alpha = 1, \dots, n \end{array} \right]. \quad (6.33)$$

На рис.6.4 показані приклади одновимірного та двовимірного вузлових способів дискретизації.

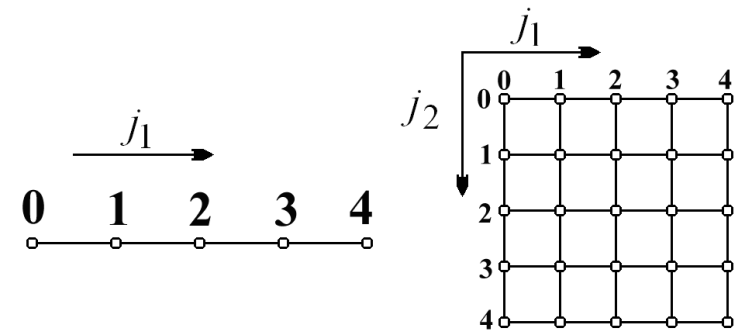


Рис.6.4. Вузлова дискретизація.

При доменному способі дискретизації неперервне значення функції замінюють множиною значень, що розміщені у геометричних центрах доменів сіткової області.

$$\mathbf{G} = \left[\begin{array}{l} \mathbf{x} \in R^n, \mathbf{x} = (\mathbf{j} - \mathbf{d})\mathbf{h}, \mathbf{j} = (j_1, \dots, j_n), \mathbf{h} = (h_1, \dots, h_n), \\ \mathbf{d} = (1, \dots, 1)/2, j_\alpha = (0, 1, \dots, n_\alpha), h_\alpha = \frac{1}{n_\alpha}, \alpha = 1, \dots, n. \end{array} \right] \quad (6.34)$$

На рис.6.5 показані приклади одновимірного та двовимірного доменних способів дискретизації.

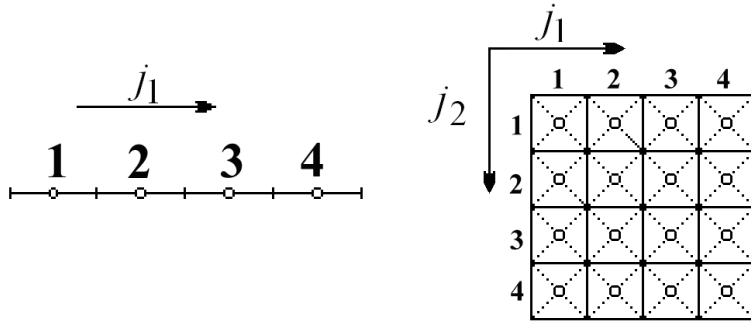


Рис.6.5. Доменна дискретизація.

Для скорочення запису дії операторів рестрикції та пролонгації використаємо шаблонний запис дії лінійного оператора, представивши систему різницьових рівнянь (6.1) у вигляді $\mathbf{L}\mathbf{U} = \mathbf{F}$. Тоді на сітці i :

$$\left(\mathbf{L}^{(i)} \mathbf{u} \right)_j = \sum_{k \in Z^n} \mathbf{L}^{(i)}(\mathbf{j}, k) \mathbf{u}_{j+k}, \quad \mathbf{j} \in \mathbf{G}, \quad (6.35)$$

де $\mathbf{Z} = (0, \pm 1, \pm 2, \dots)$ — простір сусідства, $\mathbf{j} = (j_1, j_2, \dots, j_n)$ — індекси розмірностей сітки.

Структуру оператора \mathbf{L} задають множиною

$$\mathbf{S}_A = \left\{ k \in Z^n : \exists \mathbf{j} \in \mathbf{G}, \mathbf{L}(\mathbf{j}, k) \neq 0 \right\}, \quad (6.36)$$

а множину значень $\mathbf{L}(\mathbf{j}, k)$ при фіксованому $k \in \mathbf{S}_A$ визначають як шаблон даного оператора в точці \mathbf{j} .

Позначивши координатні зсуви для двовимірної сітки (рис.6.3) відповідно через $\mathbf{e}_1 = (1, 0)$ і $\mathbf{e}_2 = (0, 1)$, представимо шаблон для точки \mathbf{j} у загальному вигляді:

$$\left[\mathbf{L}^{(i)} \right]_{\mathbf{j}} = \left[\begin{array}{ccc} & \mathbf{L}^{(i)}(\mathbf{j} - \mathbf{e}_2) & \\ \mathbf{L}^{(i)}(\mathbf{j} - \mathbf{e}_1) & \mathbf{L}^{(i)}(\mathbf{j}) & \mathbf{L}^{(i)}(\mathbf{j} + \mathbf{e}_1) \\ & \mathbf{L}^{(i)}(\mathbf{j} + \mathbf{e}_2) & \end{array} \right]. \quad (6.37)$$

6.2.2.1. Вузлова пролонгація

Розглянемо одновимірні сітки $\mathbf{G}^{(i)}$ та $\mathbf{G}^{(i+1)}$ зі співвідношенням кроків дискретизації $\frac{h_{i+1}^1}{h_i^1} = 2$.

Тоді дія оператора пролонгації

$$\mathbf{P}_{i+1}^i : \mathbf{u}^{(i+1)} \rightarrow \mathbf{u}^{(i)}, \quad \mathbf{u}^{(i)} \in R^{n_i}, \quad \mathbf{u}^{(i+1)} \in R^{n_{i+1}}, \\ n_i = 2n_{i+1} + 1,$$

визначається з виразу:

$$\mathbf{u}_{j_1}^{(i)} = \begin{cases} \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j_1} = \mathbf{u}_{j_1}^{(i+1)}, \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j_1+1} = \frac{1}{2} \left(\mathbf{u}_{j_1}^{(i+1)} + \mathbf{u}_{j_1+1}^{(i+1)} \right) \end{cases} \quad (6.38)$$

На рис.6.6 показано графічну інтерпретацію принципу дії одновимірного оператора вузлової пролонгації.

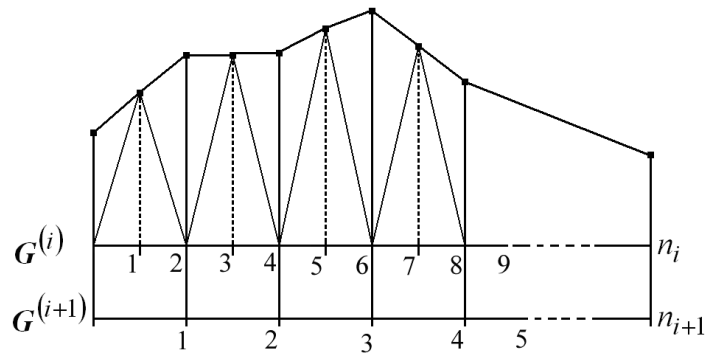


Рис.6.6. Приклад дії оператора вузлової пролонгації на одновимірній сітці.

Лінійна пролонгація на двовимірних сітках $G^{(i)}$ та $G^{(i+1)}$ зі співвідношенням кроків дискретизації $\frac{h_{i+1}^1}{h_i^1} = \frac{h_{i+1}^2}{h_i^2} = 2$ показана на рис.6.7. Обчислення значень функції $u_{j_1, j_2}^{(i)}$ у даному випадку є результатом дії оператора пролонгації $\mathbf{u}^{(i)} = \mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)}$ на семиточковому шаблоні.

$$\mathbf{u}_{j_1, j_2}^{(i)} = \begin{cases} (\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)})_{2j_1, 2j_2} = \mathbf{u}_{j_1, j_2}^{(i+1)}, \\ (\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)})_{2j_1+1, 2j_2} = \frac{1}{2} (\mathbf{u}_{j_1, j_2}^{(i+1)} + \mathbf{u}_{j_1+1, j_2}^{(i+1)}), \\ (\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)})_{2j_1, 2j_2+1} = \frac{1}{2} (\mathbf{u}_{j_1, j_2}^{(i+1)} + \mathbf{u}_{j_1, j_2+1}^{(i+1)}), \\ (\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)})_{2j_1+1, 2j_2+1} = \frac{1}{2} (\mathbf{u}_{j_1, j_2}^{(i+1)} + \mathbf{u}_{j_1+1, j_2+1}^{(i+1)}). \end{cases} \quad (6.39)$$

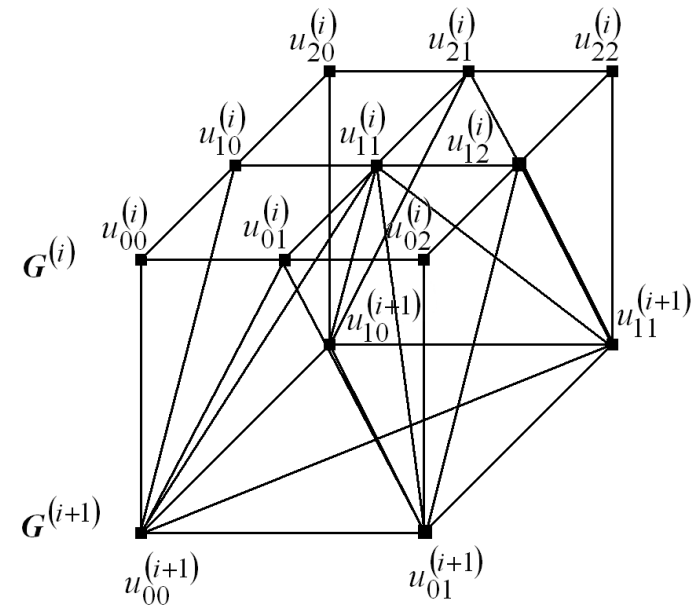


Рис.6.7. Приклад лінійної пролонгації на двовимірній сітці.

Основним недоліком використання лінійної пролонгації для реалізації операції пролонгації є виділення пріоритетного напрямку, що визначається вибором інтерполяційної діагоналі, наприклад: $((00), (11))$. При такому підході існує загроза втрати симетрії за рахунок зростання похибки у лінійно незалежному напрямку. Тому частіше застосовують більш складну білінійну пролонгацію, яка відрізняється від розглянутого випадку способом обчислення значень функції у вузлах $\mathbf{u}_{2j_1+1, 2j_2+1}^{(i)}$. Позначимо приріст координат по осі j_1 через $\mathbf{e}_1 = (1, 0)$, а по осі j_2 — через $\mathbf{e}_2 = (0, 1)$. Тоді отримаємо:

$$\left\{ \begin{array}{l} \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j} = \mathbf{u}_j^{(i+1)}, \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1} = \frac{1}{2} \left(\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_1}^{(i+1)} \right), \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_2} = \frac{1}{2} \left(\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_2}^{(i+1)} \right), \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1+e_2} = \\ = \frac{1}{4} \left(\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_1}^{(i+1)} + \mathbf{u}_{j+e_2}^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} \right) \end{array} \right. \quad (6.40)$$

Аналогічно може бути задана тривимірна пролонгація з використанням приросту координат

$$\mathbf{e}_1 = (1,0,0), \mathbf{e}_2 = (0,1,0), \mathbf{e}_3 = (0,0,1):$$

$$\left\{ \begin{array}{l} \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j} = \mathbf{u}_j^{(i+1)}, \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_k} = \frac{1}{2} \left(\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_k}^{(i+1)} \right), \\ \hspace{15em} k = 1,2,3, \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_k+e_l} = \\ = \frac{1}{4} \left(\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_k}^{(i+1)} + \mathbf{u}_{j+e_l}^{(i+1)} + \mathbf{u}_{j+e_k+e_l}^{(i+1)} \right), \\ \hspace{15em} k, l = 1,2,3, \quad k < l, \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1+e_2+e_3} = \\ = \frac{1}{8} \left(\mathbf{u}_j^{(i+1)} + \sum_{k=1}^3 \mathbf{u}_{j+e_k}^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} + \right. \\ \left. + \mathbf{u}_{j+e_2+e_3}^{(i+1)} + \mathbf{u}_{j+e_1+e_3}^{(i+1)} + \mathbf{u}_{j+e_1+e_2+e_3}^{(i+1)} \right). \end{array} \right. \quad (6.41)$$

6.2.2.2. Доменна пролонгація

Доменна пролонгація базується на способі дискретизації, який показано на рис.6.5. Нехай на одновимірних сітках $\mathbf{G}^{(i)}$ та $\mathbf{G}^{(i+1)}$ задано співвідношенням кроків дискретизації $\frac{h_{i+1}^1}{h_i^1} = 2$. Тоді дія оператора пролонгації визначається з виразу:

$$\mathbf{u}_{j_1}^{(i)} = \begin{cases} \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j_1} = \frac{3}{4} \mathbf{u}_{j_1}^{(i+1)} + \frac{1}{4} \mathbf{u}_{j_1+1}^{(i+1)}, \\ \left(P_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j_1-1} = \frac{1}{4} \mathbf{u}_{j_1-1}^{(i+1)} + \frac{3}{4} \mathbf{u}_{j_1}^{(i+1)}. \end{cases} \quad (6.42)$$

На рис.6.8 показано графічну інтерпретацію принципу дії одновимірного оператора доменної пролонгації.

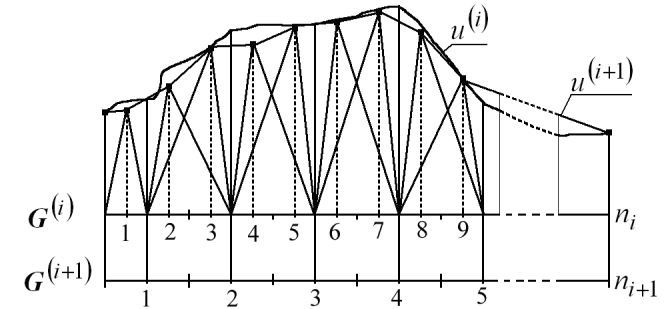


Рис.6.8. Приклад дії оператора доменної пролонгації на одновимірній сітці.

Лінійна доменна пролонгація у двовимірному випадку, яка сформована на семиточковому шаблоні, визначається подібно до відповідної вузлової пролонгації:

$$\left\{ \begin{aligned} \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j} &= \frac{1}{4} \left(3\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1} &= \frac{1}{4} \left(\mathbf{u}_j^{(i+1)} + 2\mathbf{u}_{j+e_1}^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_2} &= \frac{1}{4} \left(\mathbf{u}_j^{(i+1)} + 2\mathbf{u}_{j+e_2}^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1+e_2} &= \frac{1}{4} \left(\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_{j+e_1+e_2}^{(i+1)} \right) \end{aligned} \right. \quad (6.43)$$

Точки з координатами

$$\left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1}, \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_2}$$

на точній сітці формуються по трьох найближчих точках грубої сітки, а для формування точок з координатами

$$\left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j}, \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1+e_2}$$

необхідно попередньо вибрати напрям інтерполяційної діагоналі. Для виразу (6.43) таку діагональ визначають за координатами

$$(2j, 2j + e_1 + e_2).$$

Подібно до вузлової лінійної пролонгації, доменна двовимірна лінійна пролонгація спричиняє втрату симетрії по одному з вимірів. Тому в даному випадку частіше застосовують білінійну пролонгацію, яка симетрична по кожному з вимірів:

$$\left\{ \begin{aligned} \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j} &= \\ &= \frac{1}{16} \left(9\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_{j+e_1}^{(i+1)} + 3\mathbf{u}_{j+e_2}^{(i+1)} + \mathbf{u}_{j+e_1+e_2}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1} &= \\ &= \frac{1}{16} \left(9\mathbf{u}_{j+e_1}^{(i+1)} + 9\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_{j+e_1+e_2}^{(i+1)} + \mathbf{u}_{j+e_2}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_2} &= \\ &= \frac{1}{16} \left(9\mathbf{u}_{j+e_2}^{(i+1)} + 9\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_j^{(i+1)} + 3\mathbf{u}_{j+e_1+e_2}^{(i+1)} + \mathbf{u}_{j+e_1}^{(i+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{u}^{(i+1)} \right)_{2j+e_1+e_2} &= \\ &= \frac{1}{16} \left(9\mathbf{u}_{j+e_1+e_2}^{(i+1)} + 3\mathbf{u}_{j+e_1}^{(i+1)} + 3\mathbf{u}_{j+e_2}^{(i+1)} + \mathbf{u}_j^{(i+1)} \right). \end{aligned} \right. \quad (6.44)$$

На рис.6.9 показано структуру зв'язків при формуванні білінійної доменної пролонгації для довільних чотирьох точок з координатними зсувами $\mathbf{e}_1 = (1, 0)$ і $\mathbf{e}_2 = (0, 1)$ відносно координати $\mathbf{j} = (j_1, j_2)$.

У тривимірному випадку домен сітки $\mathbf{G}^{(i+1)}$ має вигляд куба з точкою дискретизації, яка розміщена на перетині його діагоналей, як показано на рис.6.10.

Для досягнення тривимірної симетричної пролонгації точки $\mathbf{u}_{2j}^{(i)}$ застосовують таку інтерполяційну формулу:

$$\left(\mathbf{P}\mathbf{u}^{(i+1)} \right)_{2j} = \frac{1}{64} \left(27\mathbf{u}_j^{(i+1)} + \mathbf{u}_{j+e_1+e_2+e_3}^{(i+1)} + 9\mathbf{u}_{j+e_1}^{(i+1)} + 9\mathbf{u}_{j+e_2}^{(i+1)} + 9\mathbf{u}_{j+e_3}^{(i+1)} + 3\mathbf{u}_{j+e_1+e_2}^{(i+1)} + 3\mathbf{u}_{j+e_2+e_3}^{(i+1)} + 3\mathbf{u}_{j+e_1+e_3}^{(i+1)} \right) \quad (6.45)$$

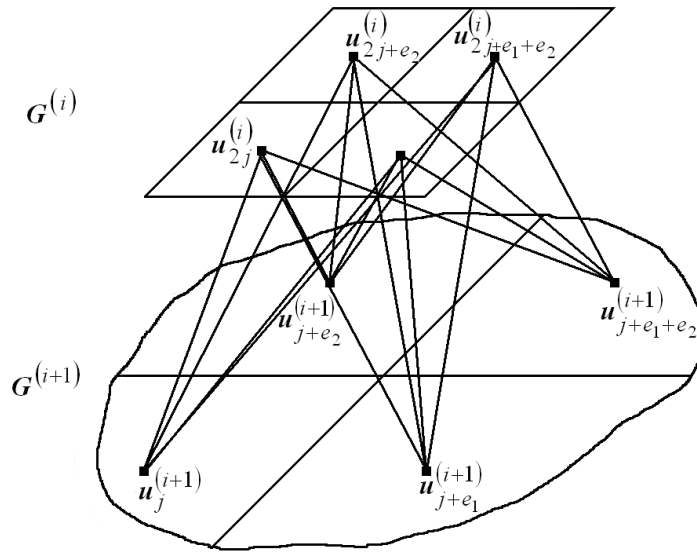


Рис.6.9. Приклад дії оператора білінійної доменної пролонгації на двовимірній сітці.

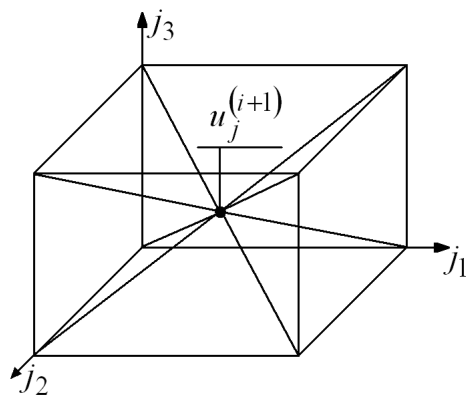


Рис.6.10. Домен тривимірної сітки.

яка може бути аналогічно виведена для решти вершин куба, утвореного точками, які розміщені в геометричних центрах доменів.

Для цього необхідно взяти з коефіцієнтом 27 значення функції в точці, яка лежить безпосередньо під точкою пролонгації, додати до нього з коефіцієнтами 9 значення функцій в точках, зсунутих по одній з координат, потім додати з коефіцієнтами 3 значення функцій в точках, зсунутих по двох координатах та додати значення функції в точці, зсунутій по трьом координатах. На рис.6.11 дано графічне зображення правила формування коефіцієнтів тривимірної симетричної пролонгації.

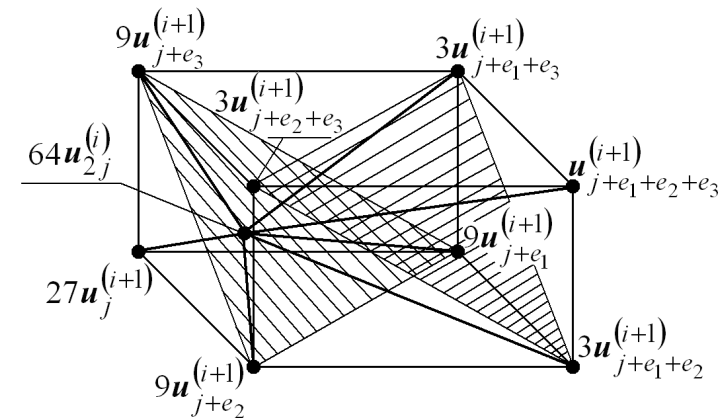


Рис.6.11. Тривимірна симетрична пролонгація.

Ми навели найпростіші приклади дії операторів пролонгації. У деяких випадках застосовують пролонгацію більш високих порядків з ускладненими формулами, які використовують значення функції з віддалених вузлів. У випадку доменної пролонгації існує також ряд проблем, пов'язаних з обчисленням значень функції на границі області. Відомі також нетривіальні способи пролонгації, які застосовуються у нелінійних

мультисіткових методах, виклад яких виходить за межі даної роботи [65].

6.2.2.3. Рестрикція

Перехід від точної сітки до більш грубої визначається дією оператора рестрикції R_i^{i+1} (6.8). У найпростішому випадку з вузловою дискретизацією ця дія полягає у переході від сітки R_i^{i+1} до сітки $i+1$, яка утворена шляхом вилучення кожного другого рядка і кожної другої колонки з сітки $i+1$. Приклад рестрикції двовимірної сітки показаний на рис.6.12.

Позначимо величину кроку дискретизації по координаті j_1 на сітці $G^{(i)}$ через $h_i^{j_1}$. Тоді у двовимірному випадку, який представлений на рис.6.12, маємо такі співвідношення:

$$\frac{h_{i+1}^{j_1}}{h_i^{j_1}} = \frac{h_{i+1}^{j_2}}{h_i^{j_2}} = 2. \quad (6.46)$$

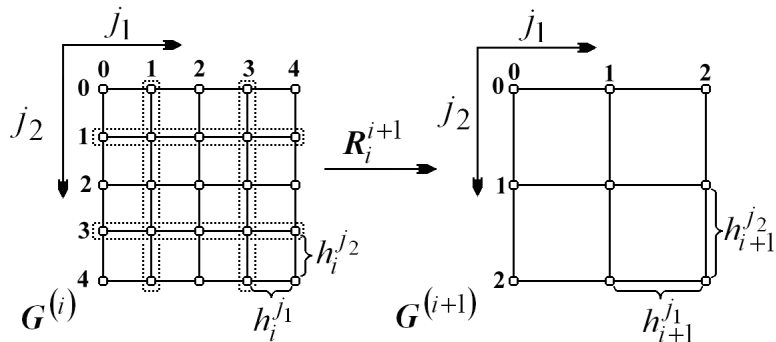


Рис.6.12. Вузлова двовимірна рестрикція.

При доменній рестрикції відбувається об'єднання просторів сусідніх доменів, як показано на рис.6.13.

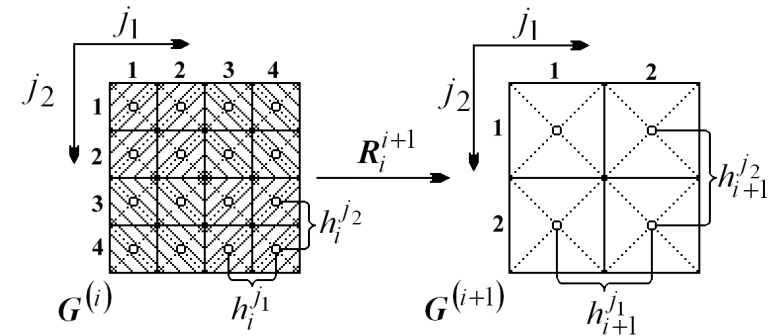


Рис.6.13. Доменна двовимірна рестрикція.

У даному випадку зростання кроку дискретизації також визначається співвідношенням (6.46), але просторове розташування точок змінюється при застосуванні операції рестрикції. Така рестрикція є найпростішою. Вона має назву ін'єктивної або взаємно-однозначної рестрикції. Дія ін'єктивної рестрикції може бути описана ін'єктивним оператором:

$$\mathbf{d}^{(i+1)} = (\mathbf{R}_{in} \mathbf{d}^{(i)})_j = (\mathbf{d}^{(i)})_{2j}. \quad (6.47)$$

Стандартну рестрикцію найчастіше визначають як операцію, обернену до пролонгації:

$$\mathbf{R}_i^{i+1} = (\mathbf{P}_{i+1}^i)^* = \left(\frac{h_i}{h_{i+1}} \right)^n (\mathbf{P}_{i+1}^i)^T, \quad (6.48)$$

де коефіцієнт $\left(\frac{h_i}{h_{i+1}} \right)^n$ задає вплив розмірності на співвідношення пролонгації та рестрикції.

Розглянемо одновимірні сітки $G^{(i+1)}$ та $G^{(i)}$ зі співвідношенням кроків дискретизації $\frac{h_i^1}{h_{i+1}^1} = \frac{1}{2}$. Тоді дія оператора рестрикції

$R_i^{i+1} : d^{(i)} \rightarrow d^{(i+1)}$, $d^{(i)} \in R^{n_i}$, $d^{(i+1)} \in R^{n_{i+1}}$, $n_i = 2n_{i+1} + 1$
визначається в такий спосіб:

$$d_j^{(i+1)} = (R_i^{i+1} d^{(i)})_j = \frac{1}{2} \left(\frac{d_{2j-1}^{(i)}}{2} + d_{2j}^{(i)} + \frac{d_{2j+1}^{(i)}}{2} \right) \quad (6.49)$$

На рис.6.14 показано графічну інтерпретацію принципу дії одновимірного оператора вузлової рестрикції. Аналогічно можна вивести залежності для опису дії оператора рестрикції з розмірністю $n = 2$. Вираз для двовимірної лінійної рестрикції задамо, використовуючи координатні зсуви $e_1 = (1,0)$, $e_2 = (0,1)$ та інтерполяційну діагональ $(j - e_1 - e_2, j + e_1 + e_2)$:

$$d_j^{(i+1)} = \frac{1}{2} \left(2d_{2j}^{(i)} + d_{2j-e_1}^{(i)} + d_{2j+e_1}^{(i)} + d_{2j-e_2}^{(i)} + d_{2j+e_2}^{(i)} + d_{2j-e_1-e_2}^{(i)} + d_{2j+e_1+e_2}^{(i)} \right). \quad (6.50)$$

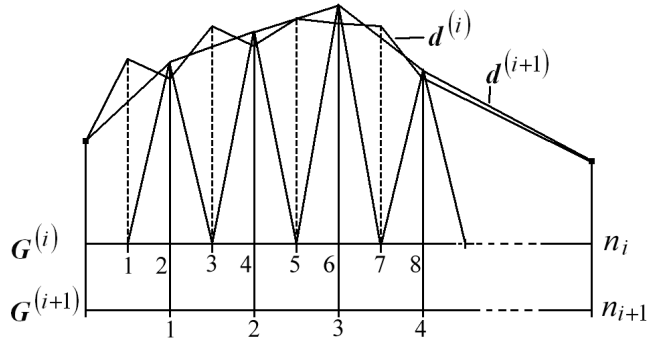


Рис.6.14. Приклад дії оператора вузлової рестрикції на одновимірній сітці.

Для збереження симетрії при переході на грубу сітку застосовують білінійну рестрикцію:

$$d_j^{(i+1)} = \frac{1}{4} \left(4d_{2j}^{(i)} + 2d_{2j-e_1}^{(i)} + 2d_{2j+e_1}^{(i)} + 2d_{2j-e_2}^{(i)} + 2d_{2j+e_2}^{(i)} + d_{2j-e_1-e_2}^{(i)} + d_{2j+e_1+e_2}^{(i)} + d_{2j-e_1+e_2}^{(i)} + d_{2j+e_1-e_2}^{(i)} \right) \quad (6.51)$$

Для тривимірного випадку також існують різні варіанти лінійної і білінійної рестрикції та симетрична трилінійна рестрикція, яку задають виразом:

$$d_j^{(i+1)} = \frac{1}{8} \left(8d_{2j}^{(i)} + 4d_{2j-e_1}^{(i)} + 4d_{2j+e_1}^{(i)} + 4d_{2j-e_2}^{(i)} + 4d_{2j+e_2}^{(i)} + 2d_{2j-e_1-e_2}^{(i)} + 2d_{2j+e_1+e_2}^{(i)} + 2d_{2j-e_1+e_2}^{(i)} + 2d_{2j+e_1-e_2}^{(i)} + 2d_{2j+e_1-e_3}^{(i)} + 2d_{2j+e_1+e_3}^{(i)} + 2d_{2j-e_1+e_3}^{(i)} + 2d_{2j+e_1-e_3}^{(i)} + 2d_{2j+e_2+e_3}^{(i)} + 2d_{2j-e_2+e_3}^{(i)} + 2d_{2j+e_2-e_3}^{(i)} + d_{2j-e_1-e_2-e_3}^{(i)} + 2d_{2j+e_1+e_2-e_3}^{(i)} + 2d_{2j-e_1+e_2-e_3}^{(i)} + 2d_{2j+e_1-e_2-e_3}^{(i)} \right). \quad (6.52)$$

Вигляд оператора рестрикції R_i^{i+1} в загальному випадку залежить від номера сітки i , але на практиці часто застосовують той самий оператор при переході між різними сітками. Тоді оператор рестрикції позначають без використання індексів: $R_i^{i+1} = R$.

Отже, у мультисітковому методі можуть бути застосовані різні ступені апроксимації операторів пролонгації, рестрикції та диференціального оператора. Позначимо через $(m_p - 1)$ та

$(m_R - 1)$ — степені поліномів, які апроксимують оператори P та $R = P^*$, а порядок диференціального оператора — через $2m$. Тоді необхідною умовою збіжності мультисіткових методів є просте співвідношення [92]:

$$m_P + m_R > 2m. \quad (6.53)$$

Із теорем збіжності мультисіткових методів, які детально розглянуті в [184,185], випливає, що дані методи характеризуються високою швидкістю збіжності для задач з гладкими коефіцієнтами. За наявності розривів та за умови анізотропних коефіцієнтів досягнення позитивних результатів значно ускладнюється.

6.3. Мультисітковий ітераційний метод для лінійних крайових задач

6.3.1. Алгоритм мультисіткового методу

Нехай дано деяку лінійну крайову задачу $Lu = f$, яка може бути апроксимована дискретною крайовою задачею:

$$L_h u_h = f_h, \quad (6.54)$$

де $L_h : H_h \rightarrow H_h$ — лінійний оператор, визначений у просторі сіткових функцій; u_h, f_h — функції, визначені у вузлах сітки G_h .

Нехай u_h — точний розв'язок рівняння (6.54). Тоді, одержавши певне наближення $u_h(n)$, можемо визначити поточну величину похибки $v_h(n)$ за формулою (6.2), а величину нев'язки $d_h(n)$ — з виразу (6.3). Очевидно, що нев'язка дорівнює нулю тоді і тільки тоді, коли наближений розв'язок $u_h(n)$ співпадає з точним.

У загальному випадку чергове наближення $u_h(n)$ визначається за допомогою ітераційного оператора:

$$u_h(n) = \Phi_h(u_h(n-1), f_h). \quad (6.55)$$

Індекс h задає крок дискретизації на сітці G_h . У випадку мультисіткового методу задачу розв'язують за допомогою послідовності сіток із кроками $h_0 < h_1 < \dots < h_{i-1} < h_i$; їх позначають $G^{(i)} = G_{h_i}$, де параметр i задає поточний рівень сітки.

Задавши оператори пролонгації P та рестрикції R , одержимо алгоритм мультисіткового локально-асинхронного методу, який складається з такої послідовності кроків:

I. *Перше згладжування наближення.*

1) Виконання v_1 разів ітераційної формули (6.55) на сітці $G^{(i)}$ з найменшим кроком дискретизації:

$$u^{(i)}(n + v_1) = \Phi_{v_1}^{(i)}(u^{(i)}(n), f^{(i)}). \quad (6.56)$$

2) Обчислення нев'язки:

$$d^{(i)}(n) = (f^{(i)} - L^{(i)}u^{(i)}(n)). \quad (6.57)$$

II. *Обчислення похибки на грубих сітках.* Цей крок полягає в послідовному розв'язуванні рівняння нев'язки на грубих сітках. Алгоритм переходів між грубими сітками визначається циклом мультисіткового методу. Найчастіше застосовують V -цикли або W -цикли. На кожній з грубих сіток, за винятком останньої, визначені такі види операцій:

1. Операція рестрикції:

$$d^{(i)}(n) = R d^{(i-1)}(n), \quad 0 < i < i_{\max}. \quad (6.58)$$

2. Перше згладжування похибки:

$$\mathbf{v}^{(i)}(n + \nu_1) = \Phi_{\nu_1}^{(i)}(\mathbf{v}^{(i)}(n), \mathbf{d}^{(i)}) \quad (6.59)$$

для рівняння нев'язки $\mathbf{L}^{(i)}\mathbf{v}^{(i)} = \mathbf{d}^{(i)}$ на сітці $\mathbf{G}^{(i)}$.

3. Операція пролонгації та корекції похибки:

$$\mathbf{v}^{(i)}(n) = \mathbf{v}^{(i)}(n) + \mathbf{P}\mathbf{v}^{(i-1)}(n), \quad 0 < i < i_{\max}. \quad (6.60)$$

4. Друге згладжування похибки:

$$\mathbf{v}^{(i)}(n + \nu_2) = \Phi_{\nu_2}^{(i)}(\mathbf{u}^{(i)}(n), \mathbf{f}^{(i)}). \quad (6.61)$$

III. Корекція на точній сітці.

1. Операція полягає в корекції наближення $\mathbf{u}^{(0)}(n)$ за

допомогою обчисленої на грубих сітках похибки $\mathbf{v}^{(0)}(n)$:

$$\mathbf{u}^{(0)}(n) = \mathbf{u}^{(0)}(n) + \mathbf{v}^{(0)}(n). \quad (6.62)$$

2. Друге згладжування наближення:

$$\mathbf{u}^{(0)}(n + \nu_2) = \Phi_{\nu_2}^{(0)}(\mathbf{u}^{(0)}(n), \mathbf{f}^{(0)}). \quad (6.63)$$

Визначення мультисіткового алгоритму

Нехай дана ієрархія сіток

$$\mathbf{G}^{(0)} \supset \mathbf{G}^{(1)} \supset \dots \supset \mathbf{G}^{(i)} \supset \dots \supset \mathbf{G}^{(i_{\max})},$$

оператори пролонгації \mathbf{P}_{i+1}^i і рестрикції \mathbf{R}_i^{i+1} для переходу між сітками $\mathbf{G}^{(i)}$ та $\mathbf{G}^{(i+1)}$, диференціальний оператор $\mathbf{L}^{(i)}$ та ітераційний оператор $\Phi_{\nu}^{(i)}$. Тоді мультисітковий ітераційний метод описують рекурсивною процедурою $MGM(i, j, Up)$, у якій параметр i вказує на поточний номер сітки, параметр j — на поточний номер циклу, а логічний параметр Up відповідає за поточний напрямок руху вздовж сіток.

У випадку, коли $j_{\max} = 1$, мультисітковий метод містить V -цикли, а при $j_{\max} > 1$ виникають W -цикли. При $Up = \text{true}$ відбувається процес рестрикції, а при $Up = \text{false}$ — процес пролонгації.

Procedure $MGM(i, j, Up)$;

Begin

If Not Up then

Begin

If $i < i_{\max}$ *then*

begin

$$\mathbf{u}[i] := \Phi_{\nu_1}^{(i)}(\mathbf{u}[i], \mathbf{f}[i])$$

$$\mathbf{f}[i+1] := \mathbf{R}_i^{i+1}(\mathbf{L}^{(i)} \cdot \mathbf{u}[i] - \mathbf{f}[i]);$$

$$\mathbf{u}[i+1] := 0;$$

If $j > 0$ *then* $MGM(i+1, j, Up)$ *else Exit*;

end else

begin

$Up := \text{Not } Up$;

$j := j - 1$;

$MGM(i-1, j, Up)$

end;

End else

begin

$$\mathbf{u}[i] := \mathbf{u}[i] + \mathbf{P}_{i+1}^i \cdot \mathbf{u}[i-1];$$

$$\mathbf{u}[i] := \Phi_{\nu_2}^{(i)}(\mathbf{u}[i], \mathbf{f}[i]);$$

If $i < 2$ *then*

begin

$Up := \text{Not } Up$;

If $j = 0$ *then* $MGM(i-1, j, Up)$ *else* $MGM(i, j, Up)$;

end else $MGM(i-1, j, Up)$;

end;

End.

На рис.6.15 показано блок-схему мультисіткового алгоритму, який складається з двох гілок, які розділені умовою $i < i_{\max}$.

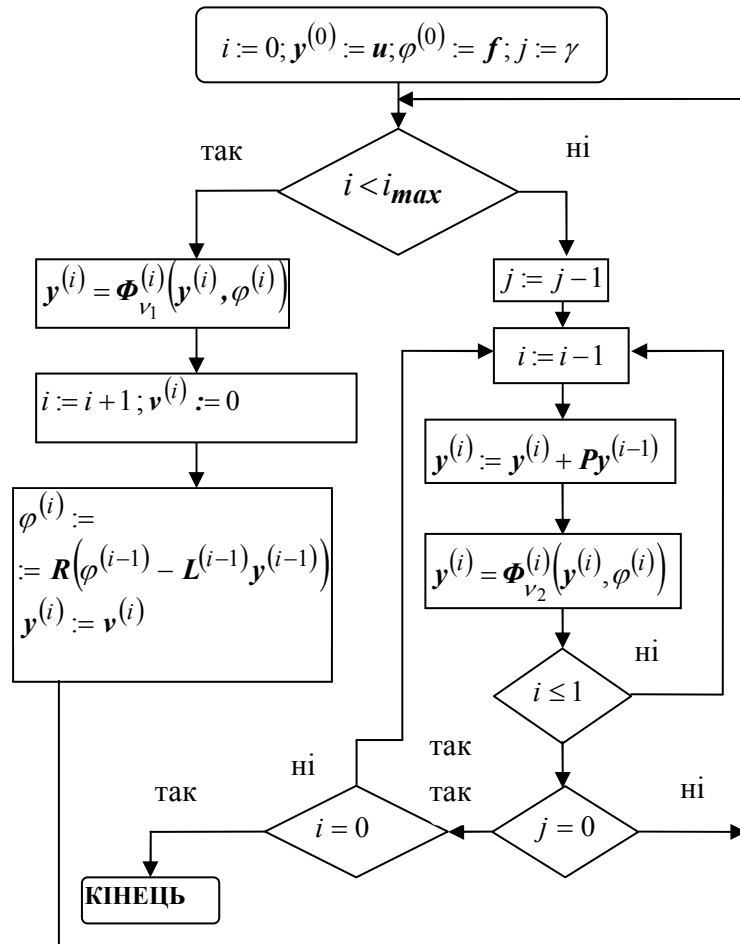


Рис.6.15. Блок-схема мультисіткового алгоритму.

Ця умова контролює досягнення алгоритмом максимально грубої сітки з номером i_{\max} . Оскільки обчислення завжди починають із точної сітки, то першим етапом алгоритму є згладжування наближення. Далі відбувається рух у напрямку

зростання кроку сітки з почерговим згладжуванням похибки та рестрикцією на більш грубу сітку.

При досягненні умови $i = i_{\max}$ переходять до гілки, яка забезпечує рух у напрямку підвищення точності сітки. При цьому виконуються операції згладжування та пролонгації на більш точну сітку. Кількість спусків на грубу сітку i_{\max} задають параметром γ , а поточний номер спуску контролюється змінною j . При $j = 0$ виникає умова завершальної пролонгації на точну сітку з другим згладжуванням шуканого наближення до розв'язку крайової задачі.

6.3.2. Збіжність мультисіткового методу

Позначимо через $A_{v_1, v_2}^{(i)}$ ітераційний оператор мультисіткового методу. Дія його, із врахуванням операцій пролонгації та рестрикції, визначається залежністю:

$$A_{v_1, v_2}^{(i)} = \Phi_{v_2}^{(i)} \left(I - P_{i+1}^i \left(L^{(i-1)} \right)^{-1} R_j^{i+1} L^{(i)} \right) \Phi_{v_1}^{(i)}, \quad (6.64)$$

де $\Phi_{v_1}^{(i)}$ — ітераційний оператор першого згладжування;

$\Phi_{v_2}^{(i)}$ — ітераційний оператор другого згладжування;

P_{i+1}^i — оператор пролонгації;

R_i^{i+1} — оператор рестрикції;

$L^{(i)}$ — диференціальний оператор на сітці i .

Базуючись на класичному підході, необхідна та достатня умова збіжності мультисіткового методу можна сформулювати у вигляді нерівності:

$$\rho(A_{v_1, v_2}^{(i)}) < 1, \quad (6.65)$$

де ρ — спектральний радіус ітераційного оператора $A_{\nu_1, \nu_2}^{(i)}$.

Оскільки відомо, що величина ρ залежить від суми $\nu = \nu_1 + \nu_2$, оператор $A_{\nu_1, \nu_2}^{(i)}$ представимо у еквівалентному вигляді — оператором $A_{\nu}^{(i)} = A_{\nu, 0}^{(i)}$, для якого

$$\rho(A_{\nu_1, \nu_2}^{(i)}) = \rho(A_{\nu}^{(i)}). \quad (6.66)$$

Для даного оператора вираз (6.64) матиме вигляд:

$$A_{\nu}^{(i)} = \left((L^{(i-1)})^{-1} - P_{i+1}^i (L^{(i-1)})^{-1} R_j^{i+1} \right) L^{(i)} \Phi_{\nu}^{(i)}. \quad (6.67)$$

Виходячи з нерівності Коші–Буняковського, можемо зробити висновок, що норма оператора $A_{\nu}^{(i)}$ є обмеженою у просторі сіткових функцій:

$$\|A_{\nu}^{(i)}\| \leq \left\| (L^{(i-1)})^{-1} - P_{i+1}^i (L^{(i-1)})^{-1} R_j^{i+1} \right\| \cdot \|L^{(i)} \Phi_{\nu}^{(i)}\|. \quad (6.68)$$

Розглянемо властивості співмножників правої частини нерівності (6.68), які пов'язані з величиною кроку дискретизації.

Властивість згладжування. Оператор $\Phi_{\nu}^{(i)}$ має властивість згладжування, якщо існує функція $\eta(\nu)$ та константи α, C_{Φ} такі, що

$$\|L^{(i)} \Phi_{\nu}^{(i)}\| \leq C_{\Phi} \eta(\nu) h_i^{-\alpha}, \quad (6.69)$$

де $\eta(\nu) \rightarrow 0$ при $\nu \rightarrow \infty$.

Властивість апроксимації. Для диференціального оператора $L^{(i)}$ виконується властивість апроксимації, якщо існують константи C_L, α такі, що

$$\left\| (L^{(i)})^{-1} - P_{i+1}^i (L^{(i)})^{-1} R_i^{i+1} \right\| \leq C_L h_i^{\alpha} \quad (6.70)$$

Теорема 6.1. Нехай виконуються умови (6.69), (6.70) та існує константа $\varepsilon > 0$. Тоді, при необмеженій максимальній

кількості ітерацій згладжування $\nu_{\max} \rightarrow \infty$ існує така їх мінімальна кількість ν_{\min} , починаючи з якої справджується нерівність

$$\|A_{\nu}^{(i)}(\nu)\| \leq C_L \eta(\nu) \leq \varepsilon. \quad (6.71)$$

Послідовність ітерацій мультисіткового методу, яка задовольняє умову (6.71), збігається до єдиного розв'язку рівняння (6.54).

Схема доведення. При $\nu_{\max} \rightarrow \infty$ з (6.69), (6.70) випливає, що $\|A_{\nu}^{(i)}\| \leq C_L \eta(\nu)$. Оскільки $\eta(\nu) \rightarrow 0$ при $\nu \rightarrow \infty$, то послідовність ітерацій з оператором $A_{\nu}^{(i)}$ збігається.

Отже, оцінка похибки мультисіткового ітераційного методу можна виразити формулою:

$$\|u^{(i)}(n) - u^{(i)}\| \leq \|A_{\nu}^{(i)}\| \cdot \|u^{(i)}(0) - u^{(i)}\|. \quad (6.72)$$

З (6.71) випливає, що права частина (6.72) не залежить від i , тобто від кількості сіток, які використовує мультисітковий метод. Величина i завжди обернено пропорційна величині кроку дискретизації h_i . Тому швидкість збіжності мультисіткового методу не залежить від величини кроку дискретизації.

6.4. Ієрархічні методи

6.4.1. Ієрархічний базис

Мультисіткові методи характеризуються підвищеною швидкістю збіжності за умови незалежності від величини кроку дискретизації. Ці властивості привели до широкого їх застосування для розв'язування крайових задач математичної фізики. Однак, поряд із безперечними перевагами, розглянуті мультисіткові методи мають кілька недоліків. Перш за все слід відмітити зростання обчислювальної складності методу, що

спричиняє необхідність застосування значних обчислювальних ресурсів для його реалізації. Крім цього, традиційний мультисітковий метод характеризується фіксованим порядком переходу між сітками, що відповідає V -циклом або W -циклом, і тому потребує синхронізації обчислень. Згадані недоліки знижують ефективність застосування мультисіткових методів у нейронних структурах, функціонування яких базується на принципах асинхронності.

Ієрархічний сітковий метод об'єднує переваги звичайного мультисіткового методу з асинхронними ітераційними методами, що були розглянуті у попередньому розділі. При його застосуванні послідовність сіток для обчислення можна вибирати довільним чином.

Розглянемо систему лінійних рівнянь

$$Lu = f. \quad (6.73)$$

Основна ідея методу полягає в перетворенні матриці системи L з вузлового базису в ієрархічний базис (рис.6.16). Нехай існують дві вкладені сітки $G^{(i)} \supset G^{(i+1)}$ і два відповідних простори сіткових функцій $H_i \supset H_{i+1}$. Стандартний базис для H_i — це вузловий базис:

$$\left\{ \varphi_j^{(i)} \right\}_{j=1}^{n_i}, \quad \varphi_j^{(i)}(x_k) = \delta_{jk} \quad (6.74)$$

Враховуючи наявність вкладених сіток, можна задати ієрархічний базис:

$$\left\{ \varphi_j^{(i)} \Big|_{x_j \in G^{(i)} \setminus G^{(i+1)}}, \varphi_j^{(i+1)} \Big|_{x_j \in G^{(i)} \cap G^{(i+1)}} \right\}, \quad (6.75)$$

де базисні функції $\varphi_j^{(i)}$, що відповідають спільним вузлам сіток $G^{(i)} \cap G^{(i+1)}$, замінюють відповідними вузловими базисними функціями $\varphi_j^{(i+1)}$. Розщеплення вузлів дискретизації

$$G = S \oplus C, \quad S = G^{(i)} \setminus G^{(i+1)}, \quad C = G^{(i)} \cap G^{(i+1)}, \quad (6.76)$$

як показано на рис.6.16, дозволяє переписати систему рівнянь (6.73) у такому вигляді:

$$\begin{pmatrix} L_{SS} & L_{SC} \\ L_{CS} & L_{CC} \end{pmatrix} \cdot \begin{pmatrix} u_S \\ u_C \end{pmatrix} = \begin{pmatrix} f_S \\ f_C \end{pmatrix}. \quad (6.77)$$

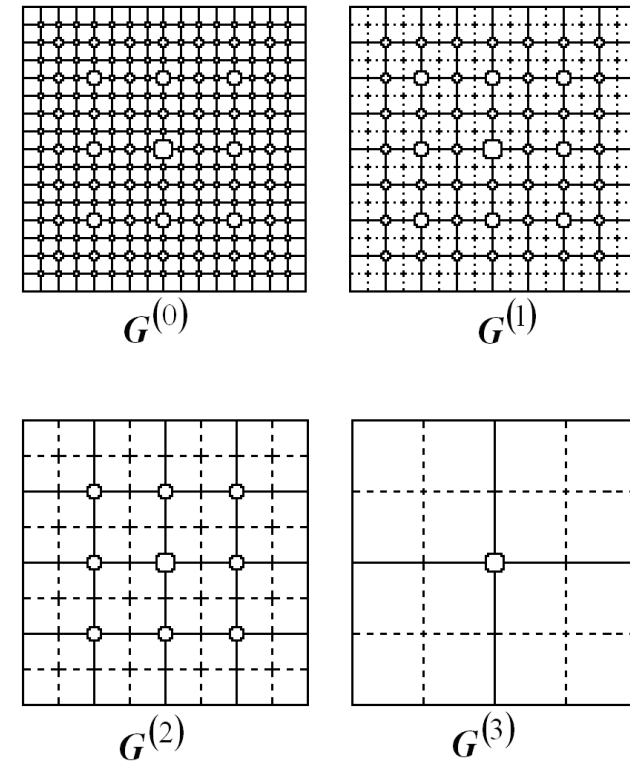


Рис.6.16. Приклад двовимірного ієрархічного базису.

Довільна сіткова функція $\varphi(x)$ може бути представлена в сітковому просторі H_i за допомогою вузлового базису (6.74):

$$\varphi(x) = \sum_{j=1}^{n_i} \beta_j \varphi_j^{(i)}, \quad (6.78)$$

або з використанням ієрархічного базису(6.75):

$$\varphi(x) = \sum_{\beta_j \in S} \widehat{\beta}_j \varphi_j^{(i)} + \sum_{\beta_j \in C} \widehat{\beta}_j \varphi_j^{(i+1)}, \quad (6.79)$$

де $\beta = (\beta_1, \dots, \beta_{n_i})$ — вектор коефіцієнтів вузлового базису;

$\widehat{\beta} = (\widehat{\beta}_1, \dots, \widehat{\beta}_{n_i})$ — вектор коефіцієнтів ієрархічного базису.

Співвідношення між векторами коефіцієнтів β та $\widehat{\beta}$ задають за допомогою відображення J :

$$\beta = J\widehat{\beta}, \quad (6.80)$$

яке визначають, виходячи з (6.78) та (6.79):

$$J = \begin{pmatrix} I_{n_S} & J_{SC} \\ 0 & I_{n_C} \end{pmatrix}. \quad (6.81)$$

Матриця системи в ієрархічному базисі:

$$\widehat{L} = \begin{pmatrix} \widehat{L}_{SS} & \widehat{L}_{SC} \\ \widehat{L}_{CS} & \widehat{L}_{CC} \end{pmatrix} = J^T L J, \quad (6.82)$$

$$\text{де } \begin{pmatrix} \widehat{L}_{SS} = L_{SS}, & \widehat{L}_{SC} = L_{SC} + L_{SS} J_{SC}, \\ \widehat{L}_{CS} = L_{CS} + J_{SC}^T L_{SS}, & \widehat{L}_{CC} = L_{CC} + J_{SC}^T L_{SS} J_{SC}. \end{pmatrix}$$

Дані залежності виведені для двох довільних сусідніх сіток $G^{(i)}$ та $G^{(i+1)}$. Тому перехід до базису, що містить більше двох сіток, як, наприклад, показано на рис. 6.16, виконують шляхом рекурсивного застосування формули (6.79).

6.4.2 Метод ієрархічної релаксації

Представимо матрицю L лінійної системи рівнянь (6.73) за допомогою верхньотрикутної матриці T і нижньотрикутної матриці з нульовою діагоналлю D як $L = T - D$. Нехай також

$F = T^{-1}D$, $\varphi = T^{-1}f$. Тоді система рівнянь (6.73) матиме вигляд:

$$u = Fu + \varphi. \quad (6.83)$$

Якщо $u(0)$ — деяка початкова апроксимація точного розв'язку u^* системи (6.73), то початкове значення нев'язки $r(0)$ та похибки $\varepsilon(0)$ визначають таким чином:

$$\begin{aligned} r(0) &= u(0) - Fu(0) - \varphi, \\ \varepsilon(0) &= u(0) - u^*. \end{aligned} \quad (6.84)$$

З урахуванням (6.84) перепишемо систему (6.83) у вигляді:

$$(I - F)\varepsilon(0) = r(0). \quad (6.85)$$

Задамо ітераційну схему з використанням релаксаційного параметра α :

$$u(n) = R(\alpha)u(n-1) - \alpha\varphi, \quad (6.86)$$

де $R(\alpha) = (1 + \alpha)I - \alpha F$.

З (6.86) виведемо загальну ітераційну схему:

$$u(n) = R^n(\alpha)u(0) - \alpha \sum_{k=1}^n R^{n-k} \varphi. \quad (6.87)$$

Розглянемо множину матриць

$$\left\{ L_k^{n_k \times n_k} \right\}_{k=1}^m \text{ при } n_1 > \dots > n_k > \dots > n_m > 0$$

з відповідними значеннями:

$$F_k = T_k^{-1}D_k, \quad \varphi_k = T_k^{-1}f_k \text{ та } R_k(\alpha_k) = (1 + \alpha_k)I - \alpha_k F_k.$$

Оператор F_k , $1 < k \leq m$ отримаємо з оператора F_1 шляхом заміни деяких рядків на вектори з нульовими елементами, крім елемента, який дорівнює одиниці і знаходиться у стовпці з номером, що дорівнює номеру рядка. Таким же чином отримують вектор φ_k з вектора φ_1 . Множину пар $\{F_k, \varphi_k\}_{k=1}^m$ називатимемо ієрархією, що складається з рівнів (F_k, φ_k) .

Означення 6.1. Ієрархія $\{F_k, \varphi_k\}_{k=1}^m$ називається δ -коректною, якщо на кожному її рівні виконується хоча б одна ітерація, тобто існує оператор

$$\Phi_s = \prod_{i=1}^s R_i, \quad (6.88)$$

що є результатом послідовного застосування операторів R_i , та справедлива нерівність $\|r_k(u^*)\| < \delta$, де $\delta > 0$ — константа.

Метод ієрархічної релаксації — це метод, згідно з яким послідовність ітерацій обчислюється за ітераційною схемою:

$$u(n+s) = \Phi_s u(n), \quad (6.89)$$

де $u(n) \in H_1$.

Визначимо початкове значення похибки $\varepsilon(0)$ як відхилення від точного розв'язку u^* для системи (6.83).

$$\begin{aligned} \varepsilon(0) &= \Phi_s u(0) - u^* = F_{i_s} \Phi_{s-1} u(0) + \varphi_{i_s} - u^* = \\ &= F_{i_s} (\Phi_{s-1} u(0) - u^*) + F_{i_s} u^* - u^* + \varphi_{i_s} = \\ &= F_{i_s} \varepsilon_{s-1} - r_{i_s}(u^*) = \dots = \\ &= \prod_{j=s}^1 F_{i_j} \varepsilon_0(0) - \sum_{j=0}^{s-1} \left(\prod_{k=j}^1 F_{i_s-k+1} \right) r_{i_s-j}(u^*), \end{aligned}$$

Застосуємо нерівність Коші–Буняковського:

$$\begin{aligned} \|\varepsilon_s(0)\| &\leq \left\| \prod_{j=s}^1 F_{i_j} \right\| \|\varepsilon_0\| + \sum_{j=0}^{s-1} \left\| \prod_{k=1}^j F_{i_s-k+1} \right\| \|r_{i_s-j}(u^*)\| \leq \\ &\leq \left\| \prod_{j=s}^1 F_{i_j} \right\| \|\varepsilon_0\| + \delta \sum_{j=0}^{s-1} \left\| \prod_{k=1}^{s-1} F_{i_s-k+1} \right\|. \end{aligned}$$

Ввівши такі заміни:

$$P_s = \prod_{j=s}^1 F_{i_j} \text{ та } P_{s,j} = \prod_{k=1}^s F_{i_s-k+1}$$

отримаємо:

$$\|\varepsilon_s\| \leq \|P_s\| \cdot \|\varepsilon_s\| + \delta \sum_{j=0}^{s-1} \|P_{s,j}\|. \quad (6.90)$$

Після використання оператора P_s n разів:

$$\varepsilon_{ns} = P_s^n \varepsilon_0 - \sum_{k=0}^{n-1} P_s^k \sum_{j=0}^{k-1} P_{s,j} r_{k-j}(u^*). \quad (6.91)$$

Підставивши (6.91) в (6.90) та визначивши суму геометричної прогресії, отримаємо нерівність:

$$\|\varepsilon_{ns}\| \leq \|P_s\|^n \cdot \|\varepsilon_0\| + \delta \sum_{j=0}^{s-1} \|P_{s,j}\| \frac{1 - \|P_s\|^n}{1 - \|P_s\|}. \quad (6.92)$$

З (6.92) випливає, що послідовність ітерацій (6.89) збігається за певних умов, які задаються означенням δ -збіжності.

Означення 6.2. Ієрархічна релаксація називається δ -збіжною, якщо виконується нерівність:

$$\lim_{n \rightarrow \infty} \|\varepsilon_n\| < \delta, \quad (6.93)$$

де $\delta > 0$ — константа.

6.4.3. Асинхронний метод ієрархічної релаксації

Використовуючи властивість ієрархічних методів, що забезпечує довільний вибір послідовності сіток, розглянемо підхід до застосування асинхронних принципів обчислень для методу ієрархічної релаксації.

Нехай $\{J_i\}_{i=1}^m$ — послідовність множин, кожен елемент якої містить номери тих рядків матриці оператора F_i , які не

дорівнюють координатному вектору-рядку. Тоді $\mathbf{J}_1 = (1, \dots, n_1)$, а з визначення Φ_s випливає, що $\bigcup_{i=1}^s \mathbf{J}_i = \{1, \dots, n_1\}$.

Означення 6.3. Нехай існує ієрархія, яка складається з пар $\{F_k, \varphi_k\}_{k=1}^m$ при $F_k, \varphi_k \in \mathbf{H}_1$ та початкове наближення $\mathbf{u}(0) \in \mathbf{H}_1$. Тоді асинхронним методом ієрархічної релаксації називають метод, для якого послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^\infty$ побудована за правилом

$$u_i(n) = \begin{cases} (F_{k(n)} \mathbf{u}(n-s(n)) - \varphi)_i, & i \in \mathbf{J}_{k(n)}, \\ u_i(n-1), & i \notin \mathbf{J}_{k(n)}, \end{cases} \quad (6.94)$$

де $s(n)$ — елемент послідовності чисел $\{s(n)\}_{n=1}^\infty$, які задовольняють такі умови:

- 1) $s(n) \geq 1, n = 1, 2, \dots$;
- 2) $\lim_{n \rightarrow \infty} (n - s(n)) = \infty$;
- 3) кожен елемент $k(n) = \{1, \dots, m\}$ зустрічається в послідовності $\{k(n)\}_{n=1}^\infty$ нескінченну кількість разів.

Даний метод може бути реалізований на рекурсивній нейронній мережі, яка складається з m нейронів. При цьому нейрон k на ітерації з номером n буде використовувати значення компонент вектора ітерації з номерами $i \in \mathbf{J}_{k(n)}$ та виконувати обчислення за формулою $u_i(n) = (F_{k(n)} \mathbf{u}(n-s(n)) - \varphi)_i$. Якщо виходити з означення асинхронного релаксаційного методу, то можна зробити висновок, що основні міркування з ним пов'язані, аналогічні міркуванням, пов'язаним з методом асинхронних ітерацій. Звідси послідовність $\{\mathbf{J}_{k(n)}\}_{n=1}^\infty$ є хаотичною послідовністю

множин з максимальною остачею, що забезпечує оптимальні умови побудови оператора Φ_s .

Означення 6.4. Ієрархію $\{F_k, \varphi_k\}_{k=1}^m$ з невід'ємними матрицями F_k називають стискаючою, якщо існує константа $0 < \alpha < 1$ і додатний вектор \mathbf{v} такі, що

$$F_k \check{\mathbf{v}} \leq \alpha \check{\mathbf{v}}, \quad k = 1, \dots, m, \quad (6.95)$$

де $\check{\mathbf{v}} = \check{\mathbf{v}}(k) \in \mathbf{R}^{n_k}$ складається лише з тих елементів вектора $\mathbf{v} \in \mathbf{R}^{n_1}$, номери яких належать множині \mathbf{J}_k .

Виходячи з того, що хаотична послідовність $\{\mathbf{J}_{k(n)}\}_{n=1}^\infty$ має максимальну остачу, побудована для рівнянь $\mathbf{u} = F_k \mathbf{u} + \varphi$ послідовність асинхронних ітерацій збігається до єдиного в \mathbf{R}^{n_1} розв'язку рівняння (6.83).

Теорема 6.2. Нехай дана δ -коректна стискаюча ієрархія $\{F_k, \varphi_k\}_{k=1}^m$. Тоді послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^\infty$ є δ -збіжною.

Д о в е д е н н я. Перший етап полягає у доведенні того факту, що величина похибки $\varepsilon(n)$ є обмеженою, тобто

$$|\varepsilon(n)| \leq \omega \mathbf{v}, \quad n = 0, 1, \dots, \quad (6.96)$$

де ω — константа, \mathbf{v} — додатний вектор.

Для цього виберемо

$$\omega = \max \left(\max_{i=1, \dots, n_1} |\varepsilon_i(0)|, \delta \beta \right), \quad (6.97)$$

де $\beta = \frac{1}{(1-\alpha) \cdot \min_{i=1, \dots, n_1} (v_i)}$.

Для $n = 0$ нерівність справджується, оскільки залежить від вигляду параметра ω . При $n > 0$ можливі два випадки. У першому з них $i \notin \mathbf{J}_{k(n)}$, і тому завдяки математичній індукції

$$|\varepsilon_i(n)| = |\varepsilon_i(n-1)| \leq \omega v_i.$$

Справедливість (6.96) у другому випадку, при $i \in \mathbf{J}_{k(n)}$, випливає з δ -коректності ієрархії та умов існування асинхронного методу ієрархічної релаксації: $s(n) \geq 1$. При цьому одержуємо значення похибки:

$$\varepsilon_{k(n)}(n) = \mathbf{F}_{k(n)}\varepsilon(n-s(n)) - \mathbf{r}_{k(n)} \leq \mathbf{F}_{k(n)}\omega\check{\mathbf{v}} + \delta \mathbf{e},$$

де $\mathbf{e} \in \mathbf{R}^{n_k(n)}$ — одиничний вектор.

Враховуючи (6.94), (6.97) та умову стискання для оператора $\mathbf{F}_{k(n)}$, запишемо

$$\varepsilon_{k(n)}(n) \leq \alpha\omega\check{\mathbf{v}} + \delta \mathbf{e} \leq \omega\check{\mathbf{v}}.$$

Звідси випливає, що

$$|\varepsilon_{k(n)}(n)| \leq \omega\check{\mathbf{v}}. \quad (6.98)$$

Для одержання нерівності (6.96) необхідно (6.98) доповнити нульовими компонентами, які відповідають вузлам сітки $\mathbf{G}^{(1)}$, для яких обчислення ще не завершені.

Виберемо зростаючу послідовність n_0, n_1, \dots таким чином, що

$$|\varepsilon_k(n)| \leq (\omega\alpha^p + \delta\beta)\check{\mathbf{v}}, \quad n \geq n_p, \quad k = 1, \dots, m. \quad (6.99)$$

З (6.98) випливає справедливість нерівності (6.99) при $p = 0$, $n_0 = 0$. Використовуємо математичну індукцію. Припустимо, що (6.99) справджується при $p < p_*$. Доведемо справедливість даної нерівності при $p = p_*$, $n > n_{p_*}$. Нехай

$$n_p = \min \left\{ n : \forall n \geq n_p, n - s(n) \geq n_{p_* - 1} \right\}.$$

Існування n_p випливає з означення асинхронного методу ієрархічної релаксації, для якого виконується умова існування максимальної остачі хаотичної послідовності множин.

Оскільки $s(n) \geq 1$, то $n_p - s(n) \geq n_{p_* - 1}$. Звідси, за математичною індукцією $n > n_{p_*}$ справедлива нерівність

$$|\varepsilon_{k(n)}(n)| \leq (\omega\alpha^{p_* - 1} + \delta\beta)\check{\mathbf{v}}, \quad k = 1, \dots, m.$$

Для $n \geq n_p$ з δ -коректності ієрархії релаксаційного методу випливає, що

$$\begin{aligned} \varepsilon_{k(n)}(n) &= \\ &= \mathbf{F}_{k(n)}\varepsilon(n-s(n)) - \mathbf{r}_{k(n)} \leq \mathbf{F}_{k(n)}(\omega\alpha^{p_* - 1} + \delta\beta)\check{\mathbf{v}} + \delta \check{\mathbf{e}}. \end{aligned}$$

Оскільки оператор $\mathbf{F}_{k(n)}$ є стискаючим оператором, то

$$\begin{aligned} |\varepsilon_{k(n)}(n)| &\leq (\omega\alpha^{p_*} + \delta\beta\alpha)\check{\mathbf{v}} + \delta \check{\mathbf{e}} \leq \\ &\leq \left[\omega\alpha^{p_*} + \delta \left(\beta\alpha + \frac{1}{\min_{k=1, \dots, n_k(n)} (v_i)} \right) \right] \check{\mathbf{v}} \leq (\omega\alpha^{p_*} + \delta\beta)\check{\mathbf{v}}. \end{aligned}$$

Отже, $|\varepsilon_{k(n)}(n)| \leq (\omega\alpha^{p_*} + \delta\beta)\check{\mathbf{v}}$ для всіх $n > n_p$.

Нехай тепер

$$n_{p_*} = \min \left\{ n : n_p \leq n \leq n_{p_*} \cup_n \mathbf{J}_{k(n)} = \{1, 2, \dots, n_1\} \right\}.$$

Звідси випливає, що

$$n_{p_*} = \min \left\{ n : n \geq n_p, \{k_1, \dots, k_m\} \subset \bigcup_{n_p < n < n_{p_*}} k(n) \right\}.$$

У цьому випадку $|\varepsilon_{k(n)}(n)| \leq (\omega\alpha^{p_*} + \delta\beta)\check{\mathbf{v}}$ для всіх $n \geq n_{p_*}$, що доводить збіжність ітераційної послідовності за умов, заданих у теоремі.

6.5. Асинхронний мультисітковий метод для лінійних крайових задач

Мультисіткові методи у поєднанні з принципами асинхронності, які були обґрунтовані для поняття ієрархії (6.94), можуть бути успішно застосовані для рекурентних нейронних структур. Такий підхід дозволяє побудувати паралельний алгоритм розв'язування крайової задачі, виходячи безпосередньо з її постановки.

Розглянемо в банаховому просторі \mathbf{B} лінійне рівняння

$$\mathbf{u} = \mathbf{F}\mathbf{u} + \varphi, \quad (6.100)$$

де $\mathbf{B} = \prod_{j=1}^{\eta} \mathbf{B}_j$, $\varphi, \mathbf{u} \in \mathbf{B}$, $\mathbf{F} : \mathbf{B} \rightarrow \mathbf{B}$.

Нехай рівняння (6.100) є шаблоном послідовності рівнянь

$$\mathbf{u} = \mathbf{F}_k \mathbf{u} + \varphi_k, \quad k = 1, \dots, m, \quad (6.101)$$

де $\{\mathbf{F}_k : \mathbf{B} \rightarrow \mathbf{B}\}_{k=1}^m$, $\{\varphi_k\}_{k=1}^m$.

Тоді для довільного початкового наближення $\mathbf{u}(0)$ визначимо похибку:

$$\begin{aligned} \mathbf{u}(1) &= \mathbf{F}_k \mathbf{u}(0) + \varphi_k, \\ \varepsilon(1) &= \mathbf{u}(1) - \mathbf{u}^* = \mathbf{F}_k (\mathbf{u}(0) - \mathbf{u}^*) + (\mathbf{F}_k - \mathbf{F}_1) \mathbf{u}^* + \varphi_k - \varphi_1 = \\ &= \mathbf{F}_k (\mathbf{u}(0) - \mathbf{u}^*) + \mathbf{F}_k (\mathbf{u}^*) - \mathbf{u}^* + \varphi_k = \mathbf{F}_k (\varepsilon(0)) - \mathbf{r}_k (\mathbf{u}^*), \end{aligned}$$

де $\mathbf{r}_k (\mathbf{u}^*) = \mathbf{u}^* - \mathbf{F}_k (\mathbf{u}^*) - \varphi_k$.

Означення 6.5. Послідовність операторів \mathbf{F}_k , $k = 1, \dots, m$, називають δ -коректною, якщо для деякого додатного δ та норми $\|\cdot\|$ у просторі \mathbf{B} виконується нерівність $\|\mathbf{r}_k (\mathbf{u}^*)\| \leq \delta$.

Нехай $\{\mathbf{J}_n\}_{n=1}^{\infty}$ — хаотична послідовність множин з елементами $\{1, \dots, \eta\}$.

Визначимо послідовність векторів $\mathbf{t}(n) = \{t_1(n), \dots, t_\eta(n)\}$ з натуральними числовими компонентами: $0 < t_j(n) \leq m$, $j = 1, \dots, \eta$.

Означення 6.6. Послідовність $\{t_j(n)\}_{n=1}^{\infty}$ називають повною, якщо кожен з елементів множини $\{1, \dots, m\}$ зустрічається в $\{t_j(n)\}_{n=1}^{\infty}$ нескінченну кількість разів. Повна послідовність $\{t_j(n)\}_{n=1}^{\infty} \forall j = 1, \dots, \eta$ завжди має максимальну остачу.

Означення 6.7. Для асинхронного мультисіткового методу послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^{\infty}$ будується за правилом

$$u_i(n) = \begin{cases} \left[\mathbf{F}_{t_j(n)} \mathbf{u}(n - s_j(n)) + \varphi_{t_j(n)} \right]_j, & j \in \mathbf{J}_n, \\ u_j(n-1), & j \notin \mathbf{J}_n, \end{cases} \quad (6.102)$$

де $n = 1, 2, \dots$, $j = 1, \dots, \eta$, $\{s(n)\}_{n=1}^{\infty}$ — послідовність векторів $s(n) = (s_1(n), s_2(n), \dots, s_\eta(n))$, що задовольняють умовам:

- 1) $s_j(n) > 0$, $n = 1, 2, \dots$;
- 2) $\lim_{n \rightarrow \infty} (n - s_j(n)) = \infty$, $j = 1, \dots, \eta$;
- 3) $s(n) \leq n$.

Асинхронний мультисітковий метод відрізняється від асинхронного релаксаційного методу тим, що в процесі обчислень на ітераційному кроці n оновлюються ті компоненти вектора ітерації $\mathbf{u}(n) \in \mathbf{B}$, номери яких відповідають елементам множини \mathbf{J}_n . При обчисленні за асинхронним релаксаційним методом номери кожної групи компонент $\mathbf{J}_{k(n)}$ відомі

заздалегідь і обчислюються одночасно, що істотно знижує рівень асинхронності.

Очевидно, що асинхронний мультисітковий метод (6.102) є узагальненням асинхронного релаксаційного методу (6.94) та асинхронного ітераційного методу (5.23) одночасно. Даний метод еквівалентний методу (6.94) за умови $\mathbf{J}_n = \mathbf{J}_{k(n)}$ і методу (5.23) при $m = 1$.

Означення 6.8. Послідовність операторів $\{\mathbf{F}_k\}_{k=1}^m$ називають δ -паракоректною, якщо для кожного \mathbf{F}_k властивість δ -коректності виконується покоординатно, тобто у кожному з підпросторів $\mathbf{V}_1, \dots, \mathbf{V}_\eta$.

Означення 6.9. Послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^\infty$, побудованих за методом (6.102), називають δ -паразбіжною при

$$\lim_{n \rightarrow \infty} \|\varepsilon_j(n)\|_j < \delta,$$

де $\delta > 0$ — константа, $j = 1, \dots, \eta$.

Очевидно, що δ -паразбіжність є більш сильною вимогою, ніж δ -збіжність, оскільки потребує покоординатної збіжності векторів ітерацій.

Теорема 6.3. Нехай хаотична послідовність $\{\mathbf{J}_n\}_{n=1}^\infty$ має максимальну остачу, а оператори \mathbf{F}_k , $k = 1, \dots, m$, є парастискаючими і δ -паракоректними. Тоді послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^\infty$, побудованих за методом (6.102), є δ -паразбіжною.

Д о в е д е н н я. Якщо оператори є δ -паракоректними, то завжди можна знайти норму, в якій ці оператори будуть δ -коректними. Зокрема, такою нормою може бути норма простору \mathbf{X} :

$$\|\mathbf{u}\| = \max_{j=1, \dots, \eta} \|u_j\|_j, u_j \in \mathbf{X}_j.$$

У цьому випадку з умови $\|u_j\|_j \leq \delta$ випливає, що $\|\mathbf{u}\| \leq \delta$.

Нехай для методу (6.102) існує повна послідовність векторів $\{t_j(n)\}_{j=1}^\infty$. Тоді для обчислення компоненти вектора $u_j(n)$ може бути використана одна з m формул, оскільки метод (6.102) дозволяє на кожному ітераційному кроці застосовувати свій оператор переходу, що впливає з вимоги повноти послідовності векторів $\{t_j(n)\}_{j=1}^\infty$, яка забезпечує існування всіх операторів із послідовності $\{\mathbf{F}_k\}_{k=1}^m$ на всьому просторі \mathbf{X} . Незалежно від вибору \mathbf{F}_k буде виконуватися нерівність, яка задає умову парастискання:

$$\mathbf{L}_k \mathbf{v} \leq \alpha \mathbf{v}, k = 1, \dots, m \quad (6.103)$$

де $0 < \alpha < 1$, \mathbf{L}_k — матриця стискання для оператора \mathbf{F}_k , $\mathbf{v} \in \mathbf{R}^\eta$ — додатний вектор.

Застосовуючи теорему 6.2 про δ -збіжність асинхронного методу ієрархічної релаксації із врахуванням того факту, що хаотична послідовність множин має максимальну остачу, можемо стверджувати, що в даному випадку існує δ -паракоректна δ -паразбіжна асинхронна ієрархічна релаксація. Теорема доведена.

Для дискретизації стаціонарної крайової задачі скористаємось різницевиими схемами. Тоді сіткові рівняння, що виникають у результаті такої дискретизації, можуть бути інтерпретовані як операторні рівняння в просторі сіткових функцій \mathbf{H}_h :

$$\begin{cases} \mathbf{u} = \mathbf{F}^{(1)}\mathbf{u} + \varphi^{(1)}, \\ \dots \\ \mathbf{u} = \mathbf{F}^{(k)}\mathbf{u} + \varphi^{(k)}, \\ \dots \\ \mathbf{u} = \mathbf{F}^{(m)}\mathbf{u} + \varphi^{(m)}, \end{cases} \quad (6.104)$$

де $\mathbf{F}^{(k)} : \mathbf{H}_h \rightarrow \mathbf{H}_{h_k}$, $\varphi^{(k)} \in \mathbf{H}_{h_k}$

$$0 < h_1 < \dots < h_k < \dots < h_m.$$

Сітка, що об'єднує всі вузли:

$$\mathbf{G} = \bigcup_{k=1}^m \mathbf{G}^{(k)},$$

де $\mathbf{G}^{(k)}$ — підсітка, що містить вузли з кроком дискретизації h_k .

Враховуючи введені позначення, різницеве рівняння для вузла j на підсітці $\mathbf{G}^{(k)}$ матиме вигляд:

$$u_j = \mathbf{F}_j^{(k)} u_j + \varphi_j^{(k)}. \quad (6.105)$$

Для того, щоб уникнути довізначення функції $\varphi^{(k)}$ у вузлах сіткової області $\mathbf{G}^{(k)}$, використовуємо ієрархічний базис (6.75). В результаті приходимо до послідовності різницевих операторів $\{\mathbf{F}^{(k)}\}_{k=1}^m$, кожен з яких визначений на сітці $\mathbf{G}^{(m)}$ за умови δ -паразбіжності.

При реалізації даного методу на нейронній мережі, що характеризується повною асинхронністю, можуть виникати ситуації, коли, наприклад, для формування ітерації у вузлі j на сітці $\mathbf{G}^{(k)}$ будуть використовуватись дані з вузлів сітки $\mathbf{G}^{(k-1)}$. Згідно з описаними властивостями різницевих операторів, така ситуація не призводить до втрати збіжності методу. Отже, згаданий метод забезпечує розв'язання крайової задачі в умовах

реалізації на асинхронних структурах, і тому має назву асинхронного мультисіткового методу.

6.6. Асинхронний мультисітковий метод для нелінійних крайових задач

Розглянемо послідовності нелінійних рівнянь

$$\mathbf{u} = \mathbf{F}_k \mathbf{u} + \varphi_k, \quad k = 1, \dots, m, \quad (6.106)$$

де $\{\mathbf{F}_k : \mathbf{B} \rightarrow \mathbf{B}\}_{k=1}^m$, $\{\varphi_k\}_{k=1}^m$.

Кожний з δ -паракоректних операторів \mathbf{F}_k є парастискаючим на \mathbf{B} з однаковою матрицею стискування \mathbf{L} :

$$\|\mathbf{F}_k(\mathbf{u}) - \mathbf{F}_k(\mathbf{u}^*)\|_{\eta} \leq \mathbf{L} \|\mathbf{u} - \mathbf{u}^*\|_{\eta}, \quad (6.107)$$

де $\mathbf{u}, \mathbf{u}^* \in \mathbf{B}$, $k = 1, \dots, m$.

Визначимо величину похибки $\boldsymbol{\varepsilon}_s(\mathbf{0})$:

$$\begin{aligned} \boldsymbol{\varepsilon}_s(\mathbf{0}) &= \boldsymbol{\Phi}_s \mathbf{u}(\mathbf{0}) - \mathbf{u}^* = \mathbf{F}_{k_s} \boldsymbol{\Phi}_{s-1} \mathbf{u}(\mathbf{0}) + \varphi_{k_s} - \mathbf{u}^* = \\ &= \mathbf{F}_{k_s} \boldsymbol{\Phi}_{s-1} \mathbf{u}(\mathbf{0}) - \mathbf{F}_{k_s} \mathbf{u}^* + \mathbf{F}_{k_s} \mathbf{u}^* - \mathbf{u}^* + \varphi_{k_s} = \\ &= \mathbf{F}_{k_s} \boldsymbol{\Phi}_{s-1} \mathbf{u}(\mathbf{0}) - \mathbf{F}_{k_s} \mathbf{u}^* - \mathbf{r}_{k_s}(\mathbf{u}^*). \end{aligned}$$

Застосувавши властивість (6.107), одержимо:

$$\begin{aligned} \|\boldsymbol{\varepsilon}_s(\mathbf{0})\|_{\eta} &\leq \mathbf{L} \|\boldsymbol{\varepsilon}_{s-1}(\mathbf{0})\|_{\eta} + \|\mathbf{r}_{k_s}(\mathbf{u}^*)\|_{\eta} \leq \\ &\leq \mathbf{L}^s \prod_{j=s}^1 \|\boldsymbol{\varepsilon}_0(\mathbf{0})\|_{\eta} + \sum_{j=0}^{s-1} \left\| \prod_{i=j}^{s-1} \mathbf{F}_{k_{s-i+1}} \mathbf{r}_{k_{s-j}} \right\|_{\eta} \leq \quad (6.108) \\ &\leq \mathbf{L}^s \prod_{j=s}^1 \|\boldsymbol{\varepsilon}_0(\mathbf{0})\|_{\eta} - \delta \sum_{j=0}^{s-1} \left\| \prod_{i=j}^{s-1} \mathbf{F}_{k_{s-i+1}} \right\|_{\eta}. \end{aligned}$$

Теорема 6.4. Нехай хаотична послідовність множин має максимальну остачу, нелінійні δ -паракоректні оператори F_k , $k = 1, \dots, t$ є парастискаючими з однією і тією ж матрицею стискання L . Тоді послідовність ітерацій $\{u(n)\}_{n=1}^{\infty}$, побудованих за асинхронним мультисітковим методом (6.102), є δ -паразбіжною.

Доведення даної теореми аналогічне доведенню теореми 6.3 яке наведено для асинхронного мультисіткового методу розв'язування лінійних крайових задач.

Жорстка умова парастискання для послідовності операторів $\{F_k\}_{k=1}^m$ є причиною жорстких обмежень на кроки дискретизації. Тому для ослаблення їх має сенс використання властивостей монотонних операторів.

Нехай для оператора F_k в гільбертовому просторі H існують такі константи $\gamma_1 < 0, \gamma_2 < 0$, що для всіх $y, z \in H$ справедливі такі нерівності:

$$\langle F_k(y) - F_k(z), y - z \rangle \leq \gamma_1 \|y - z\|^2, \quad (6.109)$$

$$\|F_k(y) - F_k(z)\|^2 \leq \gamma_2 \langle F_k(y) - F_k(z), y - z \rangle. \quad (6.110)$$

Теорема 6.5. Нехай виконуються умови (6.109), (6.110) та

$$\sigma < \frac{2}{2 - \gamma_2}.$$

Тоді операторне рівняння $u = F_k u + \varphi_k$, $k = 1, \dots, t$, для довільного $\varphi_k \in H$ має в H єдиний розв'язок.

В H введемо послідовність операторів

$$\Theta_k(u) = u + \sigma(F_k u - u + \varphi_k), \quad k = 1, \dots, t,$$

і будемо розглядати рівняння

$$u = \Theta_k(u), \quad k = 1, \dots, t.$$

Теорема 6.6. Нехай хаотична послідовність множин має максимальну остачу, і виконуються умови (6.109), (6.110),

$$\gamma_2 < \gamma_1 < \frac{\gamma_2}{2 - \gamma_2},$$

$$\frac{1 - \gamma_1 - \sqrt{(1 - \gamma_1)^2 - (1 - \eta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2} < \sigma < \frac{1 - \gamma_1 + \sqrt{(1 - \gamma_1)^2 - (1 - \eta^{-2})(1 - 2\gamma_1 + \gamma_1\gamma_2)}}{1 - 2\gamma_1 + \gamma_1\gamma_2},$$

а оператори F_k , $k = 1, \dots, t$ є δ -паракоректними в H . Тоді послідовність ітерацій $\{u(n)\}_{n=1}^{\infty}$, яка побудована за асинхронним мультисітковим методом для операторів $\{\Theta_k\}_{k=1}^m$ є δ -паразбіжною в H при довільному $u(0) \in H$.

6.7. Асинхронний мультисітковий метод для нестационарних крайових задач

Якщо крайова задача залежить від часу, то її дискретизація потребує застосування просторово-часової сітки. Після вибору такої сітки отримаємо різницеву крайову задачу, суть якої полягає у необхідності розв'язання послідовності систем алгебраїчних рівнянь, кількість яких відповідає кількості часових шарів.

Розглянемо принципи побудови асинхронного мультисіткового методу для розв'язування послідовності операторних рівнянь:

$$\begin{cases} \mathbf{u} = \mathbf{F}_k(\mathbf{u}, 1) + \tilde{\mathbf{u}}^1, \\ \dots \quad \dots \quad \dots \\ \mathbf{u} = \mathbf{F}_k(\mathbf{u}, s-1) + \tilde{\mathbf{u}}^s, & s = 1, \dots, K, \\ \dots \quad \dots \quad \dots & k = 1, \dots, m. \\ \mathbf{u} = \mathbf{F}_k(\mathbf{u}, s) + \tilde{\mathbf{u}}^K, \end{cases} \quad (6.111)$$

Для заданого початкового значення

$$\mathbf{u}^{s(0)}(0) = \mathbf{u}(0) \in \mathbf{B}$$

та хаотичної послідовності множин $\{\mathbf{J}_n\}_{n=1}^{\infty}$ побудуємо послідовність ітерацій за таким правилом:

$$u_j^{s_j(n)}(n) = \begin{cases} \left[\begin{array}{l} \mathbf{F}_{k_j(n)}(\mathbf{u}^{s(n-1)}(n-1), s_j(n-1)) + \\ + \mathbf{u}^{s_j(n-\sigma_j(n))}(n-\sigma_j(n)) \end{array} \right]_j, & j \in \mathbf{J}_n, \\ u_j^{s_j(n-1)}(n-1), & j \notin \mathbf{J}_n, \end{cases} \quad (6.112)$$

де послідовність векторів $\{s(n)\}_{n=1}^{\infty}$ будується за правилом

$$s_j(n) = \begin{cases} s_j(n-1), \\ \left\| u_j^{s_j(n)}(n) - u_j^{s_j(n-1)}(n-1) \right\|_j > \varepsilon, j \in \mathbf{J}_n, \\ s_j(n-1) + 1, \\ \left\| u_j^{s_j(n)}(n) - u_j^{s_j(n-1)}(n-1) \right\|_j < \varepsilon, j \in \mathbf{J}_n, \\ s_j(n-1), & j \notin \mathbf{J}_n, \end{cases} \quad (6.113)$$

$$\begin{aligned} \sigma_j(n) &= \\ &= \min \left\{ l : \left\| y_j^{s_j(n-l)}(k-l) - y_j^{s_j(n-l-1)}(n-l-1) \right\|_j \leq \varepsilon \right\}. \end{aligned} \quad (6.114)$$

Умови збіжності даного методу визначаються теоремою 6.7.

Теорема 6.7. Нехай хаотична послідовність множин має максимальну остачу, для кожного фіксованого s ε -паракоректні оператори $\mathbf{F}_k(\mathbf{u}, s)$ є парастискаючими з однією і тією ж матрицею стискування $\mathbf{L}^{\eta \times \eta}$, для якої виконується нерівність

$$\left\| \mathbf{F}_k(\mathbf{u}, s) - \mathbf{F}_k(\mathbf{z}, s) \right\|_{\eta} \leq \mathbf{L}^{\eta \times \eta} \left\| \mathbf{u} - \mathbf{z} \right\|_{\eta} \quad \forall \mathbf{u}, \mathbf{z} \in \mathbf{B},$$

$$s = 1, \dots, K, \quad k = 1, \dots, m.$$

Тоді послідовність ітерацій, побудованих за асинхронним мультисітковим методом (6.112)–(6.114) при довільному початковому значенні $\mathbf{u}^{s(0)}(0) \in \mathbf{B}$ є ε -паразбіжною.

Д о в е д е н н я. У випадку $K=1$ метод (6.112)–(6.114) еквівалентний методу (6.102), що забезпечується умовами його збіжності. Для випадку $K > 1$, при $s_j(0) = 1, j = 1, \dots, \eta$

послідовність ітерацій $\{\mathbf{u}(n)\}_{n=1}^{\infty}$ є ε -паразбіжною. Отже, існує такий номер ітерації n_0 та така компонента j_0 , що виконується умова

$$\left\| u_{j_0}(n_0) - u_{j_0}(n_0 - 1) \right\|_{i_0} \leq \varepsilon, \quad i_0 \in \mathbf{J}_n. \quad (6.115)$$

Нехай n_0 — мінімальний номер ітерації, для якого існує таке j_0 , що має місце нерівність (6.115). Тоді при $n < n_0$

$$\forall i \in \{1, \dots, \eta\} : \left\| u_j(n) - u_j(n-1) \right\|_j > \varepsilon, \quad j \in \mathbf{J}_n.$$

Враховуючи той факт, що оператори

$$\mathbf{F}_k(n) = (\mathbf{F}_{k_1(n)}, \dots, \mathbf{F}_{k_{\eta}(n)})$$

є парастискаючими на \mathbf{B} з однією і тією ж матрицею стискання та врахувавши таку умову:

$$\forall n = 1, 2, \dots, \forall j \in \{1, \dots, \eta\}: \bigcup_{n=t(n)}^{t(n+1)-1} \mathbf{J}_n \setminus \{j\} \neq \{1, \dots, \eta\}, \quad (6.116)$$

яка накладена на хаотичну послідовність $\{\mathbf{J}_n\}_{n=1}^{\infty}$ з максимальною остачею і мінімальною послідовністю $\{t(n)\}_{n=1}^{\infty}$, робимо висновок про те, що компоненти вектора $s(n)$ можуть змінитися лише на протязі однієї макроітерації. Тому через скінченну кількість ітерацій одержимо $s_j(n) = K \quad \forall j = 1, \dots, \eta$. Після цього послідовність ітерацій відповідатиме методу (6.102) для одного з операторних рівнянь

$$\mathbf{u} = \mathbf{F}_k(\mathbf{u}, K) + \tilde{\mathbf{u}}_k^K, \quad k = 1, \dots, m. \quad (6.117)$$

Оскільки розв'язки всіх операторних рівнянь відрізняються покоординатно один від одного не більше, ніж на ε , то одержаний результат є шуканим розв'язком.

Для збіжності послідовності ітерацій за методом (6.112)–(6.114) необхідно, щоб виконувалась нерівність $\varepsilon \geq \delta$. Умова δ -паразбіжності гарантує лише існування деякого n_* , для якого має місце нерівність

$$\left\| u_j^{s_j(n)}(n_*) - u_j^{s_j(n_*-1)}(n_* - 1) \right\|_j \leq \delta. \quad (6.118)$$

Тому при $\varepsilon < \delta$ умови δ -паракоректності операторів $\mathbf{F}_k(\mathbf{u}, s)$ не достатньо для доведення існування такого номера ітерації n_* , для якого б виконувалася умова:

$$\left\| u_j^{s_j(n_*)}(n_*) - u_j^{s_j(n_*-1)}(n_* - 1) \right\|_j \leq \delta, \quad j \in \mathbf{J}_n \quad (6.119)$$

Для монотонних операторів $\mathbf{F}_k(\mathbf{u}, s)$ обмеження на величину кроків дискретизації можна ослабити за допомогою підходу, аналогічного описаному в попередньому параграфі.

Нехай кліткова нейронна мережа складається з η елементів. Тоді кожному вузлу сіткової області може бути поставлений у відповідність свій нейрон, який забезпечує обчислення відповідної компоненти вектора ітерацій шляхом реалізації одного з операторів із множини $\{\mathbf{F}_k(\mathbf{u}, s)\}_{k=1}^m$. Асинхронний характер методу допускає довільну послідовність реалізації цих операторів, що є фундаментальною властивістю нейронних мереж. За теоремою 6.7 про збіжність всі нейрони виконуватимуть обчислення для різних $s_j(n)$ лише в межах однієї макроітерації. Подальші обчислення виконуються при фіксованому $s_j(n)$.

6.8. Нейронний мультисітковий метод в мережах прямого поширення

Вченими, що належать до Гамбурзької мультисіткової групи [26], запропонований метод, який дозволяє розв'язувати крайові задачі еліптичного типу за допомогою нейронних мереж прямого поширення. Для цього нейрони ототожнюють з вузлами сіткової області, для якої відбувається розв'язування крайової задачі. Приклад нейронної мережі прямого поширення, яка реалізує мультисітковий метод для одновимірної крайової задачі, показаний на рис.6.17. Якщо для розв'язання задачі необхідно застосувати $N+1$ сітку дискретизації з кроками $h_N > h_{N-1} > \dots > h_0$, то кожній з сіток, за винятком сітки $\mathbf{G}^{(N)}$, ставлять у відповідність 2 шари нейронів.

Оператори пролонгації $\mathbf{P}^i = \left\{ \mathbf{P}_{z,x}^i \mid z \in \mathbf{G}^{(i)}, x \in \mathbf{G}^{(i-1)} \right\}$ та

рестрикції $\mathbf{R}^i = \left\{ \mathbf{R}_{x,z}^i \mid x \in \mathbf{G}^{(i)}, z \in \mathbf{G}^{(i+1)} \right\}$ визначають граф

міжнейронних з'єднань і вагові коефіцієнти відповідно у верхній та нижній половині нейронної мережі. Додатково введені зв'язки

з ваговими коефіцієнтами $\omega_i d_{i,a}^{-1}$ ($d_{i,a}$ — діагональний елемент $(D_i)_{aa}$, ω_i — параметр згасання), які об'єднують нейрони, ототоженні з однією і тією ж точкою дискретизації у верхній і нижній половинах ШНМ.

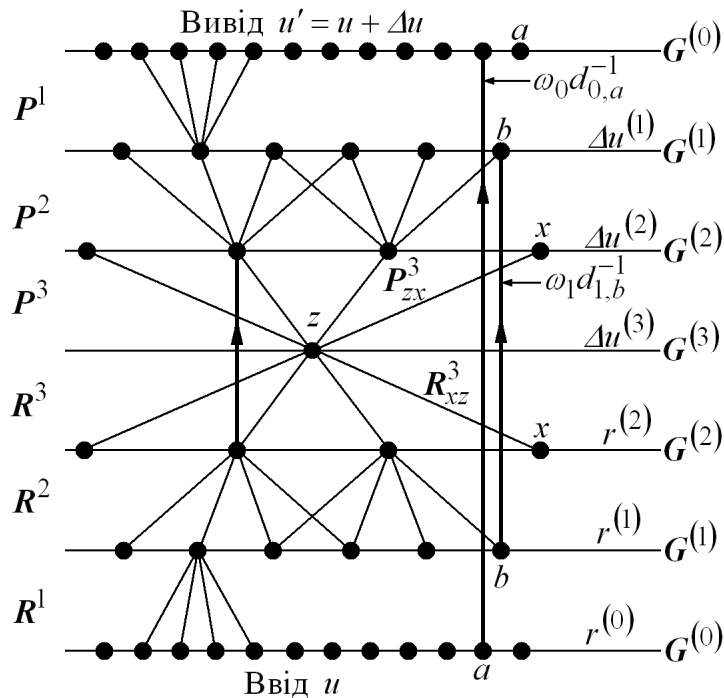


Рис.6.17. Архітектура мультисіткової ШНМ.

Відповідно до гіпотези Хебба, нейронна мережа здатна до навчання шляхом модифікації вагових коефіцієнтів

міжнейронних зв'язків. На вхід нейронної мережі подають апроксимацію розв'язку лінійної системи

$$Du = f \quad (6.120)$$

на сітці $G^{(0)}$, що забезпечує обчислення початкової нев'язки

$$r^0 = f^0 - D_0 u.$$

Результатом роботи ШНМ є корекція Δu , яка враховується шляхом додавання до початкового значення апроксимації: $u' = u + \Delta u$. Ідеальним розв'язком вважають значення вихідного вектора, що дорівнює $u = D_0^{-1} f^0$. За винятком першого шару, на вхід кожного нейрона подають зважену суму виходів тих нейронів, які розташовані на попередньому шарі та мають з'єднання з даним нейроном. У випадку лінійної крайової задачі активаційні функції нейронів є лінійними, що впливає з лінійної залежності $\Delta u(r^0)$.

Загальна формула, яку реалізує дана нейронна мережа:

$$\Delta u = \left(\omega_0 d_0^{-1} + \sum_{i=1}^N P^{(i,1)} \omega_i d_i^{-1} L^{(1,i)} \right) r^0, \quad (6.121)$$

де $P^{(i,1)} = P^i \cdot P^{i-1} \cdot \dots \cdot P^1$, $L^{(1,i)} = L^1 \cdot L^2 \cdot \dots \cdot L^i$,

$$d_i = (d_{i,0}, d_{i,1}, \dots, d_{i,n}).$$

Довільний оператор L^i визначають з виразу:

$$L^i = R^i (1 - \omega_{i-1} D_{i-1} d_{i-1}^{-1}). \quad (6.122)$$

Вагові (синаптичні) коефіцієнти $R_{x,z}^i$, $P_{z,x}^i$ та параметр згасання ω_i знаходять у процесі навчання нейронної мережі за методом навчання з учителем [100].

Нехай дана навчальна вибірка $\Psi = \{(u(j), y(j))\}_{j=1}^N$, що містить N пар вхідних $u(j)$ і вихідних $y(j)$ векторів. Виходячи

зі стандартної процедури навчання з учителем, результат роботи нейронної мережі оцінюють за цільовою функцією

$$E = \sum_j \|\mathbf{u}'(j) - \mathbf{y}(j)\|^2, \quad (6.123)$$

де j —поточний ітераційний крок; $\mathbf{u}'(j)$ — поточний вектор розв'язку;

З лінійності рівняння (6.120) випливає, що $\mathbf{y}(j)=0$ при $\mathbf{f}^0=0$ і $\mathbf{u}'(j)=\boldsymbol{\rho} \cdot \mathbf{u}(j)$, де $\boldsymbol{\rho}$ — ітераційна матриця. Якщо в процесі навчання на вхід нейронної мережі подати повну ортонормовану систему функцій, то одержимо

$$E = \sum_{j=1}^N \|\boldsymbol{\rho} \cdot \mathbf{u}(j)\|^2 = \text{tr } \boldsymbol{\rho} \boldsymbol{\rho}^* = \|\boldsymbol{\rho}\|^2. \quad (6.124)$$

Розв'язок системи одержують за виконання умови: $\mathbf{u}' = \lim_{E \rightarrow 0} (\mathbf{u})$.

6.9. Нейронний мультисітковий метод у рекурентних мережах

На відміну від традиційного для нейромережних структур підходу, рекурентні ШНМ можуть бути застосовані до розв'язання крайових задач математичної фізики за допомогою асинхронних мультисіткових ітераційних методів. Архітектура такої нейронної мережі складається з кліткової двошарової субмережі та субмережі прямого поширення. Кожен із шарів кліткової субмережі виконує специфічні обчислення, що зумовлює їх різну структуру. Нехай нейронна мережа забезпечує розв'язування системи різницевих рівнянь $\mathbf{L}_h \mathbf{u}_h = \mathbf{f}_h$. Тоді на першому шарі кліткової субмережі виконують обчислення мультисіткової корекції за формулою:

$$\mathbf{u}_h(n) = \mathbf{W} \mathbf{u}_h(n) + \Delta \mathbf{u}(n). \quad (6.125)$$

Нейрон, що забезпечує такі обчислення, показаний на рис.6.18.

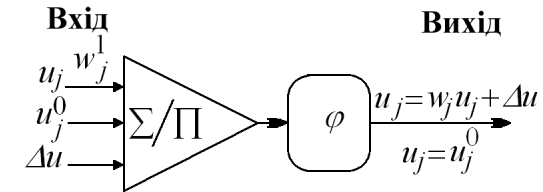


Рис.6.18. Структура нейрона першого шару.

Він складається з операційного блоку Σ/Π та блоку активаційної функції φ . Призначення операційного блоку полягає в обчисленні корекції елемента u_j вектора $\mathbf{u} = (u_0, \dots, u_j, \dots, u_m)$ для поточного ітераційного кроку n . Активаційна функція φ для нейронів першого і другого шарів кліткової нейронної субмережі є кусочно лінійною, що характерно для обчислювальних нейронних мереж. Матриця вагових коефіцієнтів \mathbf{W} задає коефіцієнти затухання.

На рис.6.19 показаний нейрон другого шару кліткової підмережі, який забезпечує формування чергової апроксимації вектора розв'язку згаданої системи різницевих рівнянь шляхом реалізації ітераційної схеми :

$$\mathbf{u}_h(n+1) = \mathbf{u}_h(n) - \boldsymbol{\rho}(\mathbf{L}_h \mathbf{u}_h(n) - \mathbf{f}_h). \quad (6.126)$$

До його складу входять три операційні блоки: $\mathbf{G}_1 - \mathbf{G}_3$, блок активаційної функції φ , блок затримки s_j та вихідний суматор. Зворотний зв'язок забезпечує виконання внутрішніх ітерацій, зумовлених застосуванням локально-асинхронного методу. Нейрон другого шару можна розглядати як математичний процесор, який отримує вхідний сигнал у вигляді певного k -

вимірному вектору $\mathbf{u}^k(n)$ (k — кількість точок, що входять до обчислювального шаблону) і формує скалярний вихідний сигнал $u_i(n+1)$, тобто реалізує відображення:

$$Ne: \mathbf{u}^k(n) \rightarrow u_i(n+1). \quad (6.127)$$

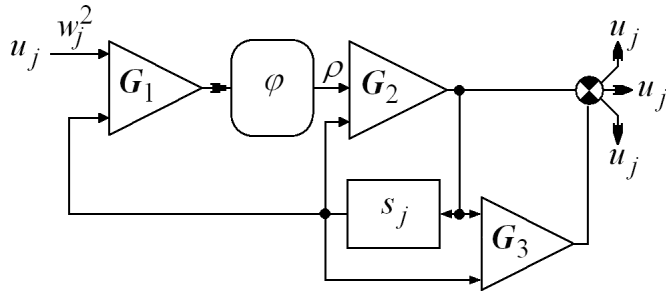


Рис.6.19. Структура нейрона другого шару.

Даний нейрон може бути реалізований у вигляді аналогового чи цифрового контролера або ж представлений моделюючою програмою. Структура його залежить від вибору конкретної ітераційної формули.

У загальному випадку відображення (6.127) має вигляд:

$$u_j(n) = u_i(n) + \rho(\mathbf{W}\mathbf{u}(n) + \mathbf{C}) = u_j(n) + \rho \left(\sum_k w_{k,j} u_k(n) + c_{k,j} \right), \quad (6.128)$$

де $w_{k,j} \in \mathbf{W}$, $c_{k,j} \in \mathbf{C}$.

Відображення (6.128) є фізичним аналогом ітераційної схеми (6.126), що дозволяє задати спосіб визначення вагових коефіцієнтів \mathbf{W} . В загальному випадку \mathbf{W} — це матриця, яка інтерпретує дію диференціального оператора \mathbf{L}_h .

З цієї причини процес формування вагових коефіцієнтів $w_{i,j}$, які утворюють матрицю \mathbf{W} , є частиною алгоритму створення

кліткової нейронної субмережі і залежить від вибору типу обчислювального шаблону.

Характер формування ітераційної послідовності $\{\mathbf{u}_h(n)\}_{n=1}^{\infty}$ базується на розгляді хаотичної послідовності множин $\{\mathbf{J}_n\}_{n=1}^{\infty}$, яка складається з непустих підмножин множини $\{1, \dots, k\}$. Використовуючи цю послідовність, будують послідовність ітерацій за правилом:

$$u_j(n+1) = \begin{cases} u_j(n), & j \notin \mathbf{J}_n, \\ \phi_j(u_1(s_1(n)), \dots, u_k(s_k(n))), & j \in \mathbf{J}_n, \end{cases} \quad (6.129)$$

де $\{s_l(n)\}_{l=1}^{\infty}$, $l=1, \dots, k$ — множина натуральних чисел, що задовольняють умовам: $s_l(n) \leq n$, $\lim_{n \rightarrow \infty} (s_l(n)) = \infty$.

Величина затримки s_l дозволяє використовувати в поточній ітерації значення компонент вектора, які входять до обчислювального шаблону і які були отримані на одній із попередніх ітерацій. На цьому факті ґрунтується використання принципу асинхронності при реалізації ітераційного процесу розв'язування системи різницевих рівнянь на клітковій нейронній субмережі.

Умова завершення ітераційного процесу має вигляд:

$$\frac{|\mathbf{u}_h(n+1) - \mathbf{u}_h(n)|}{|\mathbf{u}_h(n)|} < \varepsilon. \quad (6.130)$$

Справдження нерівності (6.130) свідчить про збіжність ітераційної послідовності $\{\mathbf{u}_h(n)\}_{n=1}^{\infty}$ до розв'язку. Для підвищення швидкості збіжності незалежно від величини кроків дискретизації застосовують мультисіткові методи. В ШНМ, архітектура якої показана на рис.6.20, для досягнення згаданих цілей використано нейронну підмережу прямого поширення.

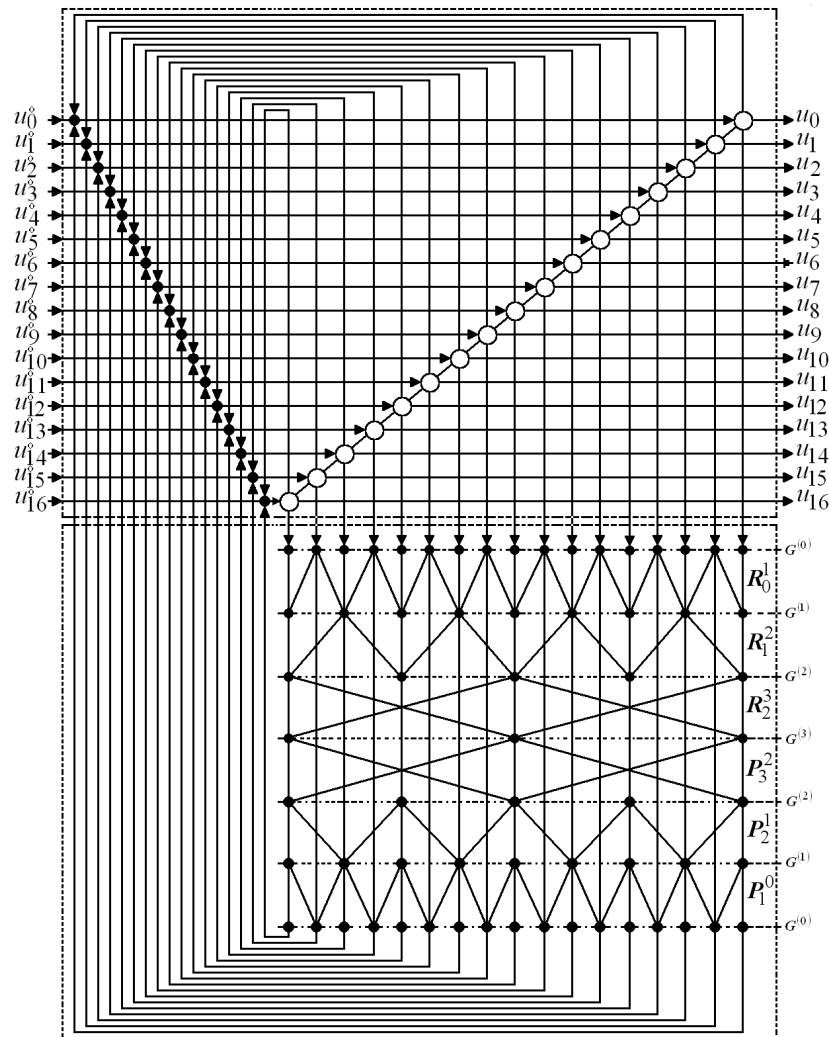


Рис.6.20. Архітектура рекурентної ШНМ для розв'язування крайових задач.

Для реалізації процесу мультисіткового згладжування застосовують $2N+1$ шарів нейронів, оскільки обчислення на кожній із сіток, за винятком сітки $G^{(N)}$, виконують два шари нейронів. Загальна кількість сіток, які застосовуються для реалізації мультисіткового згладжування, знаходиться в діапазоні $1 < i < \log_2 m$, де m — кількість елементів вектора u_h , і залежить від вибору циклу мультисіткового методу та виду операторів пролонгації та рестрикції.

Для одновимірної нейронної субмережі, яка зображена на рис.6.20, використані одновимірні сітки $G^{(i)}$ та $G^{(i+1)}$ зі співвідношенням кроків дискретизації $\frac{h_{i+1}^1}{h_i^1} = 2$.

Результат дії оператора пролонгації

$$P_{i+1}^i : v^{(i+1)} \rightarrow v^{(i)}, \quad v^{(i)} \in R^{n_i}, \quad v^{(i+1)} \in R^{m_{i+1}},$$

$$m_i = 2m_{i+1} + 1$$

визначають за формулою:

$$v_h^{(i)} = P_{i+1}^i v_h^{(i+1)}. \quad (6.131)$$

Результат дії оператора рестрикції

$$R_i^{i+1} : d^{(i)} \rightarrow d^{(i+1)}, \quad d^{(i)} \in R^{m_i}, \quad d^{(i+1)} \in R^{m_{i+1}},$$

$$m_i = 2m_{i+1} + 1$$

визначають за формулою:

$$d^{(i+1)} = R_i^{i+1} d^{(i)}. \quad (6.132)$$

На першому шарі мультисіткового згладжування виконують обчислення значень нев'язки d_h^0 :

$$d_h^0 = f - L_h u_h^0. \quad (6.133)$$

Передача одержаного значення нев'язки на наступний шар нейронів, ототожнюваний з більш грубою сіткою, відбувається з використанням операторів рестрикції (6.132), які виконують роль вагових коефіцієнтів для верхньої половини мультисіткової субмережі. На грубих сітках розв'язують рівняння нев'язки:

$$L_h^{(i)} v_h^{(i)} = d_h^{(i)}. \quad (6.134)$$

Для цього можна застосувати ітераційну схему

$$v_h^{(i)}(n+1) = v_h^{(i)}(n) - \rho \left(L_h^{(i)} v_h^{(i)} - d_h^{(i)} \right) \quad (6.135)$$

у випадку використання цифрової моделі нейрона або ж застосувати прямі методи обчислення при використанні неперервних моделей.

Одержані значення похибки передають до наступного шару, і застосовують оператори пролонгації (6.131), які визначаються як вагові коефіцієнти для нижньої половини мультисіткової субмережі. Результатом роботи мультисіткової субмережі є апроксимоване значення похибки, яке відповідає поточній макроітерації, визначуваною клітковою субмережею. Процес корекції поточного значення вектора розв'язку відбувається на першому шарі кліткової субмережі. Загальний алгоритм роботи нейронної мультисіткової субмережі може бути представлений у вигляді рекурсивної процедури.

Нехай дана ієрархія сіток

$$G^{(0)} \supset \dots \supset G^{(i)} \supset \dots \supset G^{(\log_2 m)}$$

$$G^{(\log_2 m)} \subset G^{(\log_2 m + 1)} \subset \dots \subset G^{(2 \log_2 m)}$$

оператори рестрикції R_i^{i+1} для переходу

$$G^{(i)} \rightarrow G^{(i+1)}, \quad i < \log_2 m,$$

оператори пролонгації P_{i+1}^i для переходу

$$G^{(i+1)} \rightarrow G^{(i)}, \quad \log_2 m < i \leq 2 \log_2 m,$$

диференціальний оператор $L^{(i)}$.

Тоді мультисітковий ітераційний метод описують рекурсивною процедурою $AMGM(i)$, у якій параметр i вказує на поточний номер сітки.

Procedure $AMGM(i)$;

Begin

If $i \leq 2 \log_2 m$ **then**

begin

If $i < \log_2 m$ **then**

begin

$$u[i] := u[i] - \rho \left(L^{(i)} u[i] - f[i] \right);$$

$$f[i+1] := R_i^{i+1} \left(L^{(i)} \cdot u[i] - f[i] \right);$$

end else

begin

$$u[i] := u[i] + P_{i+1}^i \cdot u[i-1];$$

$$u[i] := u[i] - \rho \left(L^{(i)} u[i] - f[i] \right);$$

end;

$AMGM(i+1)$;

end ;

End.

Ітераційний процес на кожному з нейронних шарів складається з внутрішніх нейронних ітераційних субпроцесів і виконує функцію згладжування після операцій пролонгації та рестрикції. Навчання нейронної мережі полягає у модифікації матриці вагових коефіцієнтів, яка задає дію цих операторів. Основною метою навчання є досягнення максимальної швидкості збіжності в результаті ефективних операцій корекції.

6.10. Приклади розв'язування крайових задач на рекурентних нейронних мережах

Розглянемо задачу Дирихле для рівняння:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u = -(1 + n\pi) \sin(n\pi(x + y)), \quad n = \{1, 4\}.$$

Значення функції u на границі Γ будемо визначати з виразу:

$$u|_{\Gamma} = (1 + n\pi) \sin(n\pi(x + y)).$$

Неперервний розв'язок даної задачі шукатимемо в квадраті

$$\bar{\Omega} = \Omega \cup \Gamma = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}.$$

Перехід до різницевої крайової задачі виконаємо в два етапи:

1. Замінімо область неперервної зміни аргументу областю його дискретної зміни.
2. Замінімо диференціальний оператор деяким різницевим оператором та сформулюємо різницевий аналог для граничної умови та початкових даних.

Результатом цієї процедури буде перехід до алгебраїчної системи рівнянь, яка може бути розв'язана за допомогою асинхронного мультисіткового методу.

У межах реалізації першого етапу введемо множину вкладених сіток:

$$\begin{aligned} \bar{G}^{(k)} &= G^{(k)} \cup \Gamma^{(k)} = \\ &= \left\{ \begin{array}{l} x_i = ih_i^{(k)}, y_j = jh_j^{(k)}, \sum_{i=0}^{64} h_i^{(k)} = 1, \sum_{j=0}^{64} h_j^{(k)} = 1, \\ h_i^{(k)} = h_j^{(k)} = h^{(k)}, \\ 0 \leq k \leq 6 \end{array} \right\}. \end{aligned}$$

На другому етапі використовуємо один із методів апроксимації диференціального оператора. Найпростіший підхід полягає в заміні похідних, які входять у $\mathbf{L}u$, на їх різницеві аналоги, які визначають $\mathbf{L}^{(k)}u^{(k)}$.

Для даної задачі різницевий оператор має вигляд:

$$\begin{aligned} \left(\mathbf{L}^{(k)}u^{(k)}\right)_{i,j} &= u_{i,j}^{(k)} - \left(u_{i+h,j}^{(k)} - 2u_{i,j}^{(k)} + u_{i-h,j}^{(k)}\right)/h^2 - \\ &\quad - \left(u_{i,j+h}^{(k)} - 2u_{i,j}^{(k)} + u_{i,j-h}^{(k)}\right)/h^2. \end{aligned}$$

Використовуючи різницевий оператор

$$\mathbf{L}^{(k)}u^{(k)} = \left\{ \left(\mathbf{L}^{(k)}u^{(k)}\right)_{i,j} \mid i = 0, \dots, 64, j = 0, \dots, 64 \right\},$$

будемо проводити обчислення значень функції за ітераційною формулою:

$$u^{(k)}(n+1) = u^{(k)}(n) - \omega \frac{h^2}{4} \left(\mathbf{L}^{(k)}u^{(k)} - \mathbf{F}^{(k)} \right),$$

де $\mathbf{F}^{(k)} = \left\{ F_{i,j}^{(k)} \mid i = 0, \dots, 64, j = 0, \dots, 64 \right\}$,

$$F_{i,j}^{(k)} = (1 + n\pi) \sin(n\pi h^{(k)}(i + j)), \quad \omega — \text{коефіцієнт затухання.}$$

Значення нев'язки визначимо за формулою

$$d_{i,j}^{(k)}(n) = \left(F_{i,j}^{(k)} - \left(\mathbf{L}^{(k)}u^{(k)}\right)_{i,j} \right)$$

та спроекуємо її на більш грубу сітку за допомогою оператора рестрикції:

$$d_{i,j}^{(k+1)} = \left(\mathbf{R}_k^{k+1} \mathbf{d}^{(k)} \right)_{i,j} =$$

$$= \frac{1}{8} \left(4d_{2i,2j}^{(k)} + d_{2i-1,2j}^{(k)} + d_{2i+1,2j}^{(k)} + d_{2i,2j-1}^{(k)} + d_{2i,2j+1}^{(k)} \right)$$

Проміжне згладжування на грубих сітках виконуватимемо за ітераційною формулою:

$$\mathbf{v}^{(k)}(n+1) = \mathbf{v}^{(k)}(n) - \omega \frac{h^2}{4} \left(\mathbf{L}^{(k)} \mathbf{v}^{(k)} - \mathbf{d}^{(k)} \right),$$

де $\mathbf{v}^{(k)}$ — поточне значення похибки на сітці $\mathbf{G}^{(k)}$.

Операцію пролонгації

$$\mathbf{P}_{k+1}^k : \mathbf{G}^{(k+1)} \rightarrow \mathbf{G}^{(k)}$$

виконуватимемо за формулою:

$$v_{i,j}^{(k)} = \begin{cases} \left(\mathbf{P}_{k+1}^k \mathbf{v}^{(k+1)} \right)_{2i,2j} = v_{i,j}^{(k+1)}, \\ \left(\mathbf{P}_{k+1}^k \mathbf{v}^{(k+1)} \right)_{2i+1,2j} = \frac{1}{2} \left(v_{i,j}^{(k+1)} + v_{i+1,j}^{(k+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{v}^{(i+1)} \right)_{2i,2j+1} = \frac{1}{2} \left(v_{i,j}^{(k+1)} + v_{i,j+1}^{(k+1)} \right), \\ \left(\mathbf{P}_{i+1}^i \mathbf{v}^{(i+1)} \right)_{2i+1,2j+1} = \frac{1}{2} \left(v_{i,j}^{(k+1)} + v_{i+1,j+1}^{(k+1)} \right). \end{cases}$$

Корекцію наближеного значення функції на сітці $\mathbf{G}^{(0)}$ виконують за формулою:

$$\left(u_{i,j}^{(0)} \right) = u_{i,j}^{(0)} + v_{i,j}^{(0)}.$$

Приклади розв'язку з різними параметрами показані на рис. 6.21–рис. 6.23.

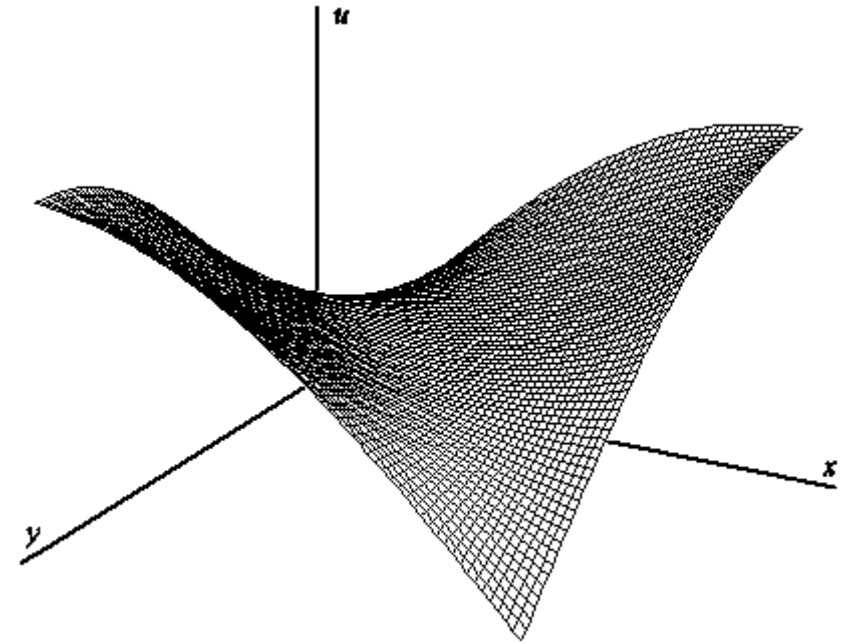


Рис. 6.21. Графік розв'язку рівняння

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u = -(1 + \pi) \sin(\pi(x + y)).$$

Параметри розв'язування крайової задачі	
Назва	Значення
Параметр функції правої частини	$n = 1$
Коефіцієнт затухання	$\omega = 1,3$
Точність	$\varepsilon = 0,0000000001$
Максимальна кількість ітерацій	$It_{\max} = 990$
Кількість ітерацій згладжування	$\nu = 1$

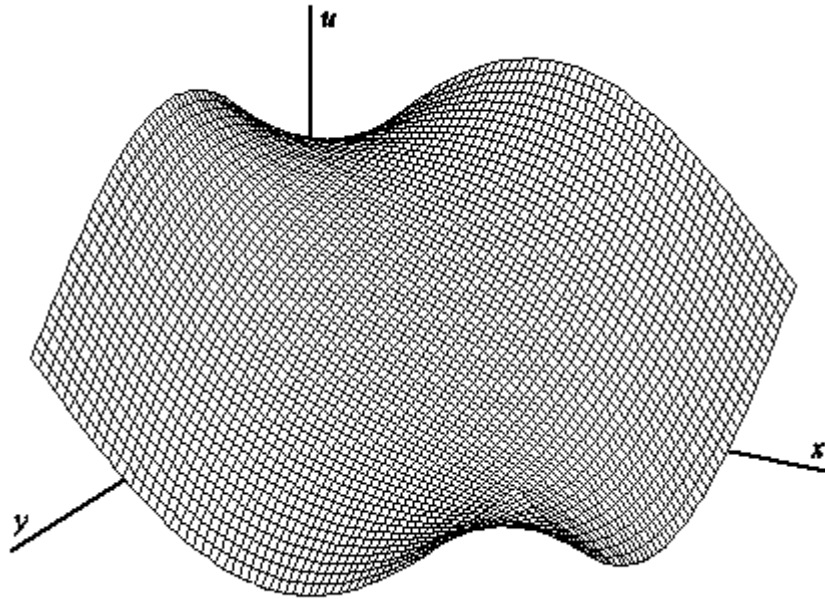


Рис. 6.22. Графік розв'язку рівняння

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u = -(1 + 2\pi)\sin(2\pi(x + y)).$$

Параметри розв'язування крайової задачі	
Назва	Значення
Параметр функції правої частини	$n = 2$
Коефіцієнт затухання	$\omega = 1,3$
Точність	$\varepsilon = 0,0000000001$
Максимальна кількість ітерацій	$It_{\max} = 856$
Кількість ітерацій згладжування	$\nu = 1$

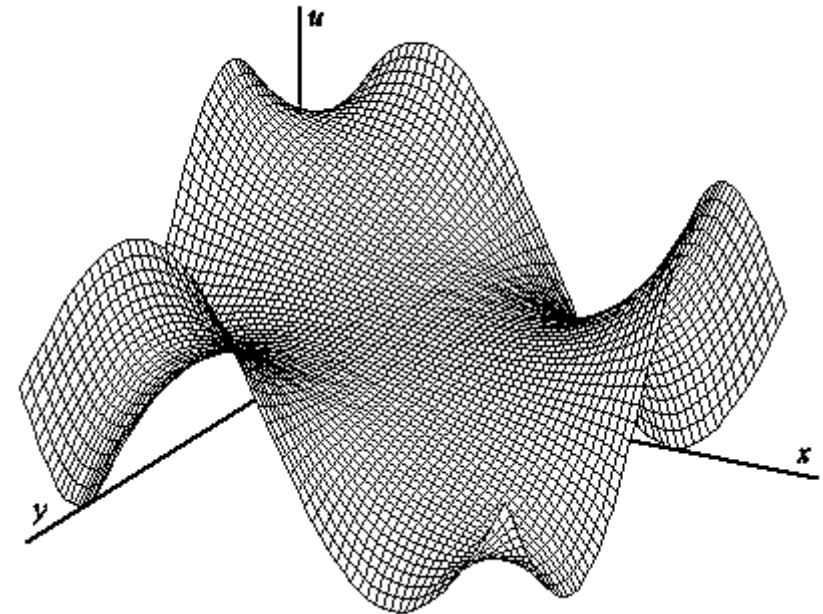


Рис. 6.23. Графік розв'язку рівняння

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u = -(1 + 4\pi)\sin(4\pi(x + y)).$$

Параметри розв'язування крайової задачі	
Назва	Значення
Параметр функції правої частини	$n = 4$
Коефіцієнт затухання	$\omega = 1,3$
Точність	$\varepsilon = 0,0000000001$
Максимальна кількість ітерацій	$It_{\max} = 796$
Кількість ітерацій згладжування	$\nu = 1$

Проведені дослідження залежності між величиною локального критерію збіжності асинхронного мультисіткового алгоритму

$$|u_{i,j}(n+1) - u_{i,j}(n)| / |u_{i,j}(n)| < \varepsilon$$

та глобальною точністю розв'язку

$$\|u - u^*\| < \delta.$$

На рис.6.24 показані приклади графіків розв'язку та відповідних ітераційних поверхонь, що характеризують рівень спрацювання локального критерію збіжності. Дослідження показали необхідність збереження співвідношення $\varepsilon = 10^{-3} \delta$ для досягнення задовільної точності розв'язку. В умовах асинхронних обчислень можуть виникати випадки „передчасного” спрацювання локального критерію збіжності, що прводить до істотного зростання локальної похибки. Засобом боротьби з цим явищем є відповідне зменшення локального критерію збіжності ε . Додатковим критерієм якості одержаного розв'язку при використанні локального критерію збіжності є співвідношення між максимальною кількістю ітерацій It_{\max} , середньою кількістю ітерацій It_{mid} та мінімальною кількістю It_{\min} . При зближенні цих трьох показників якість розв'язку зростає.

$$\alpha = \frac{It_{\min}}{It_{\max}} 100\% \text{ – екстремальний критерій.}$$

$$\beta = \frac{It_{\text{mid}}}{It_{\max}} 100\% \text{ – середній критерій.}$$

Альтернативним підходом є введення глобального критерію збіжності

$$|u(n+1) - u(n)| / |u(n)| < \varepsilon,$$

який забезпечує зупинку глобального обчислювального процесу лише після досягнення критерію збіжності в кожній точці області.

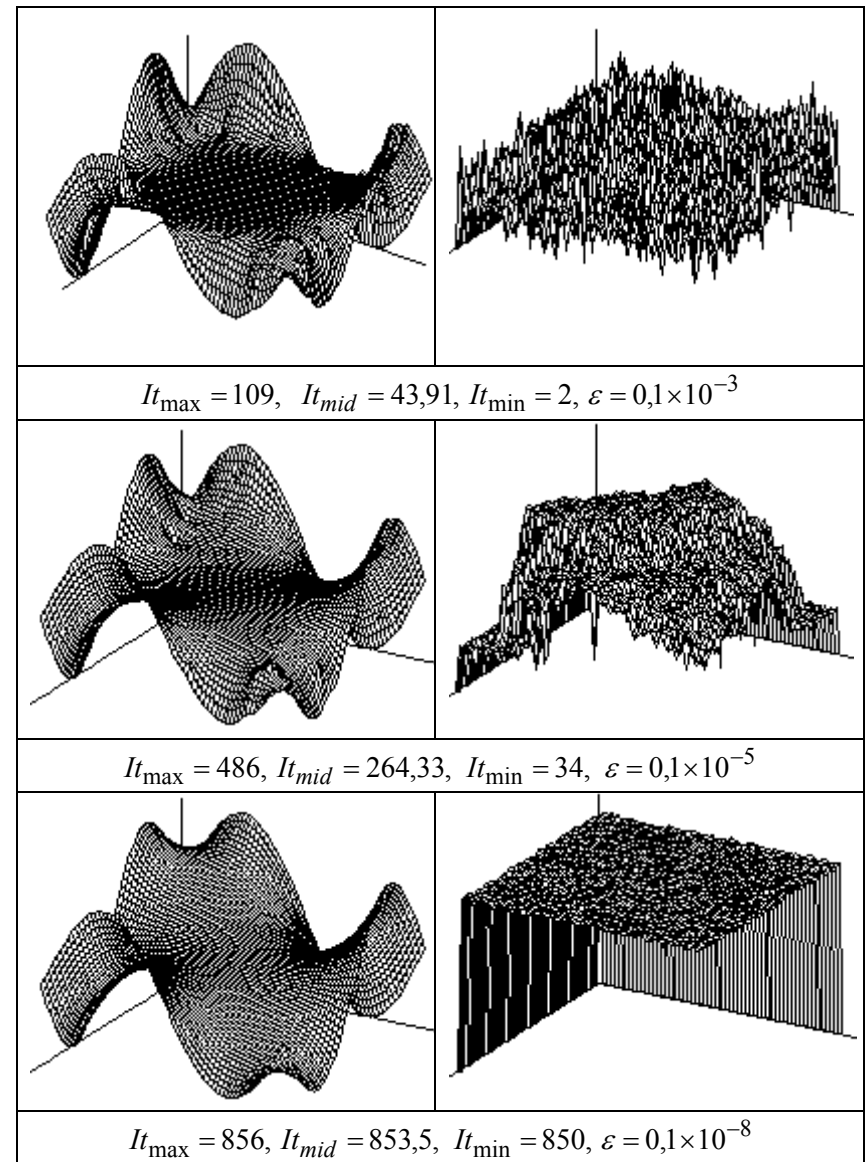


Рис. 6.24. Приклади розв'язку рівняння з різною величиною критерію збіжності.

СПИСОК ЛІТЕРАТУРИ

1. Греченко Т.Н., Соколов Е.Н. Нейрофизиология памяти и обучения // Руководство по физиологии. Механизмы памяти / Под ред. И.П. Ашмарина и др. Л.: Наука, 1987.–С.132-172.
2. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных. Докл. АН СССР, 1956.–Т.108, №2.–С.179-182.
3. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного. Докл. АН СССР, 1957.–Т.114, №5.–С. 953-956.
4. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа.–М.: Наука, 1981.–544 с.
5. Марчук В.А., Нестеренко Б.Б. Асинхронные методы параллельных вычислений.// Праці Інституту математики НАН України.– К.: Ін-т математики НАН України, 1998.–Т.22.–308 с.
6. Минский М., Пейперт С. Перцептроны.–М.: Мир, 1971.–261 с.
7. Нестеренко Б.Б., Новотарский М.А. Мультипроцессорные системы.–Киев: Институт математики, 1995.–408 с.
8. Нестеренко Б.Б., Новотарський М.А. Асинхронні паралельні обчислення в нейронних структурах.–Київ, 2002.–67с.–(Препр. / НАН України. Ін-т математики; 2002.1).
9. Нестеренко Б.Б., Марчук В.А. Основы асинхронных методов параллельных вычислений. Киев: Наукова думка, 1989.–176 с.
10. Новотарский М.А. PRO-сетевая модель цифрового

- нейрона // Радиоэлектроника. Информатика. Управление, 2002.–№1.–С.74-77.
11. Новотарский М.А. Имитационное моделирование нейросетей для решения уравнений математической физики локально-асинхронными методами // Радиоэлектроника. Информатика. Управление, 2001.–№1.–С.113-116.
 12. Ортега Дж., Рейнболдт В. Итерационные методы решения операторных уравнений, М: Мир, 1975.–558 с.
 13. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность.–М.:Мир, 1989.–512 с.
 14. Самарский А.А. Теория разностных схем.–М.:Наука, 1983.–616 с.
 15. Самарский А.А., Гулин А.В. Численные методы.–М.: Наука, 1989.–432с.
 16. Тихонов А.Н. О регуляризации некорректно поставленных задач // Докл. АН СССР, 1963.–Т.153, №1.–С.49-53.
 17. Тихонов А.Н. О решении некорректно поставленных задач и методе регуляризации //Материалы совместного советско- американского симпозиума по уравнениям с частными производными (август 1963 г.) / АН СССР, сиб. отд.- Новосибирск, 1963.–С.3-7.
 18. Abbas H.M., Fahmy M.M. Neural model for Karhunen-Loeve transform with application to adaptive image compression // IEE Proceedings-I, 1993.–vol.140, №2.–P.135-143.
 19. Amari S. Natural Gradient Works Efficiently in Learning // Neural Computation, 1998.–vol.10.–P.251-276.
 20. Amari S., Cichocki A., Yang H.H. A new learning algorithm for blind signal separation // Advances in Neural Information Processing Systems, Touretzki D.S., Mozer M.C., Hasselmo M.E., Eds., MIT Press, 1996.–vol.8.–P.757-763.
 21. Ancona F., Ridella S., Rovetta S., Zunino R. On the

- Importance of Sorting in Neural Gas Training of Vector Quantizers // Proceedings of ICNN-97, 1997.–P.1804-1808.
22. Anderson J.A., Golden R.M., Murphy G.L. Concepts of distributed systems // SPIE Optical and Hybrid Computing, 1986.–vol.634.–P.260-276.
 23. Ashby W.R. Design for a Brain.–New York: Wiley, 1952.–306 p.
 24. Atkeson C.G., Schall S.A., Moore A.W. Locally weighted learning // AI Rev, 1997.–vol.11.–P.11-73.
 25. Bäker M. Another look on multigrid // International Journal of Modern Physics C, 1997.–vol.8, №2.–P.191-205.
 26. Bäker M., Kalkreuter T., Mack G., Speh M. Neural multigrid for gauge theories and other disordered systems // Proc. Physics Computing'92, 1992.–P.12-21.
 27. Baldi P., Hornik K. Neural networks and principal component analysis // Neural Networks, 1989.–vol.2, №1.–P.53-58.
 28. Baraldi A., Alpaydın E. Simplified ART: A new class of ART algorithms // Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. TR-98-004, 1998.–42 p.
 29. Baraldi A., Parmiggiani F. A neural network for unsupervised categorization of multivalued input patterns: an application to satellite image clustering // IEEE Trans. Geosci. Remote Sensing, 1995.–vol.4, № 3.–P.393-396.
 30. Baraldi A., Shenato L. Soft-to-hard model transition in clustering: A review // Int. Comput. Sci. Inst., Berkeley, CA, TR-99-010, 1999.–34 p.
 31. Bartfai G, White R. ART-based Modular Networks for Incremental Learning of Hierarchical Clustering // Technical Report CS-TR-96/11, 1996.–28 p.
 32. Bartfai G.A. Comparison of Two ART-base Neural Networks for Hierarchical Clustering // ANNES'95, The Second New Zealand International Two-Stream Conference On Artificial Neural Networks and Expert Systems, IEEE Computer Society Press, 1995.–P.83-86.

33. Battiti R. First and second-order methods for learning: Between steepest descent and newton's method // Neural Computation, 1992.–vol.4, №2.–P.141-166.
34. Baudet G.M. Asynchronous Iterative Methods for Multiprocessors // Journal of the Association for Computing Machinery, 1978.–vol.25, №2.–P.226-244.
35. Baudet G.M. The Design and Analysis of the Algorithms for Asynchronous Multiprocessors // Ph.D.Diss., Pittsburg: Carnegie-Mellon Univ., PA, 1978.–182 p.
36. Bell A.J., Sejnovski T.J. An information-maximization approach to blind separation and blind deconvolution // Neural Computation, 1995.–vol.7, №6.–P.1029-1059.
37. Berman A., Plemmons R.J. Nonnegative Matrices in the Mathematical Sciences. SIAM, Philadelphia, 1994.–340 p.
38. Bertsekas D.P. Distributed asynchronous computation on fixed points. // Mathematical programming, 1983.–vol.27.–P.107-120.
39. Bertsekas D.P., Tsitsiklis J.N. Parallel and Distributed Computation: Numerical Methods. // Athena Scientific, 1997.–718 p.
40. Bertsekas D.P., Tsitsiklis J.N. Some aspects of parallel and distributed algorithms – a survey. // Automatica, 1991.–vol.27, №1.–P.3-21.
41. Bezdek J.C., Pal N.R. Generalized clustering networks and Kohonen's self-organizing scheme // IEEE Trans. Neural Networks, 1993.–vol.4.–P.549-557.
42. Bhat N.V., McAvoy T.J. Determining model structure for neural models by network pruning // Computers in Chemical Engineering, 1992.–vol.16.–P.271-281.
43. Billings S.A., Chen S., Korenberg M.J. Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. // International Journal of Control, 1989.–vol.49.–P.2157-2189.
44. Blonda P., Satalino G., Baraldi A., De Blasi R. Segmentation

- of multiple sclerosis lesions in MRI by fuzzy neural networks: FLVQ and FOSART // Proc. NAFIPS '98, Pensacola Beach, FL, August 1998.–P. 39-43.
45. Brandwood D. A complex gradient operator and its application in adaptive array theory // IEE Proc., 1983.–vol.130, № 1.–P 11-16.
 46. Broomhead D.S., Lowe D. Multivariable functional interpolation and adaptive networks // Complex Systems, 1988.–№ 2.–P.321-355.
 47. Bru R., Elsner L., Neumann M. Convergence of infinite products of matrices and inner-outer iterations schemes // Electronic Transactions on Numerical Analysis, 1994.–№2.–P.183-193.
 48. Cardoso J.F. Infomax and maximum likelihood for blind source separation // IEEE Signal Processing Letters, 1997.–№4.–P.109-111.
 49. Cardoso J.F., Souloumiac A. Jacobi angles for simultaneous diagonalization. // SIAM J. Mat. Anal. Appl., 1996.–vol.17, №1, P.161-164.
 50. Carpenter G. A. Grossberg S. ART2: Self-organization of stable category recognition codes for analogue input patterns // Appl. Opt., 1987.–vol.26, № 21.–P.4919-4930.
 51. Carpenter G.A., Grossberg S., Markuzon N., Reynolds J.H., Rosen D.B. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps // IEEE Trans. Neural Networks, 1992.–vol.3.–P.698-713.
 52. Carpenter G.A., Grossberg S., Rosen D.B. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system // Neural Networks, 1991.–vol.4.–P.759-771.
 53. Carpenter G., Grossberg S. A massively parallel architecture for a self-organizing neural pattern recognition machine // Computing Vision, Graphics, and Image Processing, 1987.–

- vol.37.–P.54-115.
54. Chazan D., Miranker W. Chaotic relaxation // Linear Algebra and Applications, 1969.–vol.2.–P.199-222.
 55. Chua L.O., Yang L. Cellular Neural Networks. Applications. // IEEE Transactions on Circuits and Systems, 1988.–vol.35, № 10.–P.1273-1290.
 56. Chua L.O., Yang L. Cellular Neural Networks. Theory. // IEEE Transactions on Circuits and Systems, 1988.–vol.35, № 10.–P.1257-1272.
 57. Cichocki A., Kasprzak W. Adaptive Learning Algorithm for Principal Component Analysis for Partial Data // Cybernetics and Systems'96, 1996.–vol.2.–P.1014-1019.
 58. Cohen M.A., Grossberg S.G. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks // IEEE Transactions on Systems, Man and Cybernetics, 1983.–vol.13.–P.815-826.
 59. Comon P. Independent Component Analysis, a new concept? // Signal Processing, 1994.–vol.36, № 3.–P.287-314.
 60. Cordoso J.F. Blind Signal Separation: statistical principles // Proceedings of the IEEE, 1998.–vol.9, № 10.–P.2009-2025.
 61. Cowan J.D. A Mathematical Theory of Central Nervous Activity. Ph. D. Thesis, University of London, UK, 1967.–178 p.
 62. Croom F.H. Principles of Topology. Saunders College Publishing, 1989.–312 p.
 63. Cybenko G. Approximation by Superpositions of a Sigmoidal Function // Mathematical Control Signals Systems, 1989.–vol.2, №4.–P.303-314.
 64. Davè R. N., Krishnapuram R. Robust clustering method: A unified view // IEEE Trans. Fuzzy Syst, 1997.–vol.5.–P.270-293.
 65. de Zeeuw P.M. Matrix-dependent prolongations and restrictions in blackbox multigrid solver.// J.Comput. Appl. Math., 1990.–vol.3.–P.1-27.

66. Dempster A.P., Laird N.M., Rubin D.B. Maximum Likelihood Estimation from Incomplete Data via the EM algorithm // *Journal of the Royal Statistical Society B*, 1977.–vol.39.–P.1-38.
67. El Tarazi M.N. Some convergence results for asynchronous algorithms // *Numerische Mathematik*, 1982.–vol.39.–P.325-340.
68. Elsner L., Koltracht I., Neumann M. Convergence of sequential and asynchronous nonlinear paracontractions // *Numerische Mathematik*, 1992.–vol.62.–P.305-319.
69. Elsner L., Koltracht I., Neumann M. On the convergence asynchronous paracontractions with application to tomographic reconstruction from incomplete data // *Linear Algebra and its Applications*, 1990.–vol.130.–P.65-82.
70. Elsner L., Koltracht I., Neumann N. On the convergence of sequential and asynchronous nonlinear paracontractions // *Numerische Mathematik*, 1992.–vol.62.–P.305-319.
71. Fahlman S.E., Labiere C. The Cascade-Correlation Learning Architecture // *Advances in Neural Information Processing Systems*, 1990.–P.524-532.
72. Fedorenko R.P. A relaxation method for solving elliptic difference equations // *Comput. Math. And Math Phys.*, 1962.–vol.1.–P.1092-1096.
73. Fedorenko R.P. The speed of convergence of one iterative process // *USSR Comput Math. And Math. Phys.*, 1964.–vol.4, №3.– P.227-235.
74. Finoff W., Hergert F. Zimmermann H. G. Improving model selection by nonconvergent method // *Neural Networks*, 1993.–vol.6, № 6.–P.771-783.
75. Freeman W.J., Burke B. Central pattern generation and recognition in olfactory bulb: A correlation learning rule // *Neural Networks*, 1988.–№1.–P.277-288.
76. Fritzke B. Growing cell structures – a self-organizing network for unsupervised and supervised learning // *Neural Networks*,

- 1994.–vol.7, №9.–P.1441-1460.
77. Frommer A., Mayer G. Convergence of relaxed parallel multisplitting methods // *Linear Algebra Appl.*, 1989.–vol.119.–P.141-152.
78. Frommer A., Mayer G. On the theory and practice of multisplitting methods in parallel computation // *Computing*, 1992.–vol.49.–P.63-74.
79. Frommer A., Szyld D.B. Asynchronous two-stage iterative methods // *Numerische Mathematik*, 1994.–vol.69.–P.141-153.
80. Frommer A., Szyld D.B. H-splitting and two-stage iterative methods // *Numerische Mathematik*, 1992.–vol.63.–P.345-356.
81. Funahashi K. On the Approximate Realization of Continuous Mappings by Neural Networks // *Neural Networks*, 1989.–v.2, № 3.–P.183-192.
82. Gallant S. Optimal linear discriminants. // *Eighth international Conference on Pattern Recognition. IEEE*, October 1986.– P.849-852.
83. Gelenbe, E., and Fourneau, J.M. Random neural networks with multiple classes of signals // *Neural Computation*, 1999.–vol.11, №4.–P.953-963.
84. Geman S., Bienenstock E., Doursat R. Neural Networks and the Bias/Variance Dilemma // *Neural Computation*, 1992, vol.4, №1.–P.1-58.
85. Geman S., Geman D. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images // *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.–vol.6.– P.721-741.
86. Girosi F., Jones M., Poggio T. Regularization theory and neural network architectures // *Neural Computation*, 1995.–vol.7.–P.219-269.
87. Golub G.H., Overton M.L. The convergence of inexact Chebyshev and Richardson iterative methods for solving

- linear systems // *Numerische Mathematik*, 1988.–vol.53.–P.571-593.
88. Gorbachenko V.I. Methods for solving partial differential equations // *Neurocomputers: Design and Applications*, 2000.–vol.1, №2.–P.16-29.
 89. Grossberg S. Classical and instrumental learning by neural networks // *Progress in theoretical biology*, New York: Academic Press, 1974.–vol.3.–P.51-141.
 90. Grossberg S. Contour enhancement, short-term memory, and consistencies in reverberating neural networks // *Studies in Applied Mathematics*, 1973.–L11.–P.213-257.
 91. Grossberg S. Embedding fields: Underlying philosophy, mathematics, and applications of psychology, physiology, and anatomy // *Journal of Cybernetics*, 1971.–vol.1.–P.28-50.
 92. Hackbush W. Multi-grid methods and applications. Springer series in computational mathematics 4, Berlin:Springer-Verlag, 1985.–542 p.
 93. Hackbush W. On the convergence of multi-grid iterations // *Beit. Numer. Math.*, 1981.–vol.9.–P.231-329.
 94. Hackbush W. On the multi-grid method applied to difference equations // *Computing*, 1978.–vol.20.–P.291-306.
 95. Hassibi B., Stork D., Wolff G.J. Optimal Brain Surgeon and General Network Pruning // *IEEE Int. Conf. Neural Networks*, San Francisco, 1993.–P.293-299.
 96. Haykin S. *Neural Networks — A Comprehensive Foundation*. Second edn, Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.–842 p.
 97. Hebb D.O. *The Organization of Behavior: A Neuropsychological Theory*.–New York: Wiley, 1949.–358 p.
 98. Hecht-Nielsen R., Kolmogorov's Mapping Neural Network Existence Theorem // *IEEE First Annual Int. Conf. on Neural Networks*, San Diego, 1987.–vol.3.–P.11-13.
 99. Henrique H.M., Lima E.L., Seborg D.E. Model structure determination in neural network model // *Chemical*

- Engineering Science*, 2000.–vol.55.–P.5457-5469.
100. Hertz J., Krogh A. Palmer R.G. *Introduction to the Theory of Neural Computation*.–New York:Addison-Wesley, 1991.–327p.
 101. Hinton G.E., Sejnowski T.J. Learning and Relearning in Boltzmann Machines // *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.–P.282-317.
 102. Hopfield J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities // *Proc. National Academy of Sciences, USA* 79, 1982.–P.2554-2558.
 103. Hopfield J.J., Tank D. Neural computation of decisions in optimization problems // *Biological Cybernetics*, 1985.–vol.52.–P.141-152.
 104. Hopfield J.J. Neural with graded response have collective computational properties like those of two-state neurons // *Proceedings of the National Academy of Science of the USA*, 1984.–vol.81.–P.3088-3092.
 105. Hopfield, J.J. Neural networks and physical systems with emergent collective computational properties // *Proceedings of the National Academy of Sciences of the USA*, 1982.–vol.79.–P.2554 - 2588.
 106. Hornick H., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // *Neural Networks*, 1989.–vol.2, № 5.–P.359-366.
 107. Huang J., Georgiopoulos M., Heileman G.L. Fuzzy ART properties. // *Neural Network*, 1995.–vol.8, № 2.–P. 203-213.
 108. Hung C., Lin S. Adaptive Hamming Net: A fast-learning ART 1 model without searching // *Neural Networks*, 1995.–vol.8, № 4.–P.605-618.
 109. Ishihara S., Ishihara K., Nagamachi M., Matsubara Y. ArboArt: ART-based Hierarchical Clustering and its application to questionnaire data analysis // *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, 1995.–vol.1.–P.532-537.

110. Jankovic M.A New Modulated Hebbian Learning Rule – Method for Local Computation of a Principal Subspace // Proceedings of 8-th International Conference on Neural Information Processing, China, 2001.–6 p.
111. Jolliffe I.T. Principal Component Analysis.–New York: Springer-Verlag, 1986.–271 p.
112. Kalman R.E., Bucy R.S. New results in linear filtering and prediction. // Journal of Basic Engineering (ASME), 1961.–vol.83.–P.95-108.
113. Karnin E.D. A simple procedure for pruning back-propagation trained neural networks // IEEE transactions on neural networks, 1990.–vol.1, №2.–P.239-242.
114. Kawamoto M., Barros A.K., Ohnishi N. A neural network for blind separation of convolved non-stationary signals // Proc. International ICSC Workshop on Independence and Artificial Neural Networks 98, Tenerife, Spain, 1998.–P.1374-1379.
115. Kohonen T. Self-Organization and Associative Memory. Springer Series in Information Sciences, 1989.–vol.8.–312 p.
116. Kohonen T. Self-organizing maps: Optimization approaches // Proceedings of the International Conference on Artificial Neural Networks, Espoo, Finland, June 1991.–P. 981-990.
117. Kohonen T. The self-organizing map // Proceedings of the IEEE, 1990.–vol.78, №9.–P.1464-1480.
118. Kohonen T. Self-organized formation of topologically correct feature maps // Biological Cybernetics, 1982.–vol.43.–P.59-69.
119. Kohring G.A. Artificial Neurons with Arbitrarily Complex Internal Structures // Internet http://xyz.lanl.gov/cs/0108009_2001.-22 p.
120. Kosko B. Bi-directional associative memories // IEEE Transactions on Systems, Man and Cybernetics, 1987.–vol.18, №1.–P.49-60.
121. Krieger U.R. On a two-level multigrid solution method for finite Markov chains // Linear Algebra and its Applications,

- 1995.–vol.223/224.–P.415-439.
122. Krishnapuram R., Keller J.M. A possibilistic approach to clustering // IEEE Trans. Fuzzy Syst., 1993.–vol.1.–P.98-110.
123. Lanzkrom P.J., Rose D.J., Szyld D.B. Convergence of nested classical iterative methods for linear systems // Numerische Mathematik, 1991.–vol.58.–P.685-702.
124. Le Cun Y, Denker J, Solla S. Optimal brain damage // Advances in Neural Network Information Processing Systems, 1990.–vol.2.–P.598-605.
125. LeCun, Y. Une procedure d'apprentissage pour reseau a seuil assymetrique // Cognitiva 85, 1985.–P.599-604.
126. Linde Y., Buzo A., Gray R.M. An algorithm for vector quantizer design // IEEE Transactionson Communications, 1980.–vol.28.–P.84-95.
127. Martinetz T.M., Berkovich S.G., Schulten K.J. “Neural-Gas” Network for Vector Quantization and its Application to the Time-Series Prediction // IEEE Transactions on Neural Networks, 1993–vol.4, №4–P.558-569.
128. Martinetz T.M., Schulten K.J. A Neural-Gas Network Learns Topologies // In Kohonen T., Mäkisara K., Simula O.,Kangas J (Eds.), Artificial Neural Networks, North-Holland, Amsterdam, 1991.–P.397-402.
129. Martinetz T.M., Schulten K.J. Topology representing networks // Neural Networks, 1994.–vol.3, №3.–P.507-522.
130. Matsuoka K., Ohya M., Kawamoto M. Neural net for blind separation of nonstationary signals // Neural Networks, 1995.–vol.8, №3.–P.411-419.
131. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics, 1943.–№ 5.–P.115-133.
132. Miellou J.C. Algorithmes de relaxation chaotique a retards // RAIRO-R1, 1975.–P.55-82.
133. Miellou J.C. Iterations chaotiques a retards, etudes de la concergence dans le cas d'espaces partielement or donnees. //

- CRAS, serie A t.278.– 1974.–P.957-960.
134. Miller K.D., MacKay D.J.C. The Role of Constraints in Hebbian Learning // *Neural Computation*, 1994.–vol.6.–P.100-126.
135. Minsky M.L. Steps toward artificial intelligence // *Proceedings of the Institute of Radio Engineers*, 1961.–vol.49.–P.8-30.
136. Minsky M.L. Theory of neural-analog reinforcement systems and its application to the brain-model problem. Ph.D. Thesis, Princeton University, Princeton, NJ, 1954.–143 p.
137. Molgedey L., Schuster H.G. Separation of a mixture of independent signals using time delayed correlations // *Phys.Rev.Lett.*, 1994.–vol.72, №23.–P.3633-3637.
138. Moody J, Darken C.J., Fast Learning In Networks of Locally Tuned Processing Units // *Neural Computation*, 1995.–vol.1.–P.281-294.
139. Morè J.J. Nonlinear generalizations of matrix diagonal dominance with application to Gauss-Seidel iterations // *SIAM Journal on Numerical Analysis*, 1972.–vol.9.–P.357-378.
140. Mozer M.C., Smolensky P. Skeletonization: A technique for trimming the fat from a network via relevance assessment // *Advances in Neural Network Information Processing Systems*, 1989.–vol.1.–P.1568-1574.
141. Mozer M.C., Smolensky P. Using relevance to reduce network size automatically // *Connection Science*, 1989.–vol.1, №1.–P.3-16.
142. Nesterenko B.B., Novotarskiy M.A. Mathematical simulation for parallel asynchronous methods of boundary value problems of mathematical physics // 16th IMACS World Congress, 2000.–6 p.
143. Neumaier A. New techniques for the analysis of linear interval equations // *Linear Algebra Appl.*, 1984.–vol.58.–P.273-325.

144. Nichols N.K. On the two-stage iterative processes for solving linear equations // *SIAM Journal on Numerical Analysis*, 1973.–vol.10.–P.460-469.
145. O'Leary D.P., White R.E. Multi-splitting of matrices and parallel solution of linear systems // *SIAM Journal of Algebraic and Discrete Mathematic*, 1985.–vol.6.–P.630-640.
146. Oja E. A Simplified neuron model as a principal component analyzer // *J. Math. Biol.*, 1982.–vol.15.–P.267-273.
147. Oja E. Neural networks, principal components and subspaces // *Int. J. Neural Systems*, 1989.–vol.1.–P.61-68.
148. Oja E., Ogawa H., Wangviwattana J. Principal component analysis by homogeneous neural networks, part I: The weighted subspace criterion, 1992 .–E75-D.–vol.3.–P.366-375.
149. Ortega J.M., Rheinboldt W.C. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York and London, 1970.–572 p.
150. Ostrovski M.A. Determinanten mit überwiegender Hauptdiagonale und die absolute Konvergenz von linearen Iterationsprozessen // *Comentarii Mathematici Helvetici*, 1956.–vol.30.–P.175-210.
151. Papoulis A. Probability, Random Variables and Stochastic Processes, 3-d edition, McGraw-Hill, New York, 1991.–668 p.
152. Parker D.B. Learning-logic: Casting the cortex of the human brain in silicon // Technical Report, Center for Computational Research in Economics and Management Science, TR-47, Cambridge, MA: MIT, 1985.–73 p.
153. Parra L., Spence C. Convolutional blind source separation of non-stationary sources // *IEEE Trans. on Speech and Audio Processing*, 2000.–P.320-327.
154. Pott M. On the convergence of asynchronous iteration methods for nonlinear paracontactions and consistent linear systems // *Linear Algebra and its Applications*, 1998.–

- vol.283.–P.1-33.
155. Prechelt L. Early stopping – but when?// *Neural Networks: Tricks of the Trade*, Springer-Verlag, Berlin, 1998.–P.55-70
 156. Ramon y Cajal S. *Histologie du systeme nerveux de l'homme et des vertebres*. - Paris: Maloine, 1911.–714 p.
 157. Redner R., Walker H. Mixture densities, maximum likelihood and the EM algorithm. // *SIAM Review*, 1984.–vol.26, №2.–P.195-239.
 158. Riedmiller M., Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm // H. Ruspini, ed., *Proc. of the IEEE Intl. Conference on Neural Networks*, San Fransisco, California, 1993.–P.586-591.
 159. Rochester N., Holland J.H., Haibt L.H., Duda W.L. Tests on a cell assembly theory of the action of the brain, using a large digital computer. // *IRE Transactions on Information Theory*, 1956.–N IT-2.–P.80-93.
 160. Rosenblatt F. Two theorems of statistical separability in the perceptron. // *Symposium on the Mechanization of Thought Processes*, 1959.–P.421-456.
 161. Rosenblatt F. The Perceptron: A probabilistic model for information storage and organization in the brain // *Psychological Review*, 1958.–vol.65.–P.386-408.
 162. Rumelhart D.E., Hinton G.E., Williams R.J. Learning representations by back-propagating errors // *Nature* (London), 1986.–vol.323.- P.533-536.
 163. Sanger T.D. Optimal Unsupervised Learning in a Single-Layer Neural Network. // *Neural Networks*, 1989.–vol.2.–P.459-473.
 164. Shepherd G. M., Koch C. Introduction to synaptic circuits // *The Synaptic Organization of the Brain* (G. M. Shepherd, ed.).- New York: Oxford University Press, 1990.–P.3-31.
 165. Shih F. Y., Moh J., Chang F. A new ART-based neural architecture for pattern classification and image enhancement without prior knowledge // *Pattern Recognition*, 1992.–vol.25,

- №. 5.–P.533-542.
166. Shumway R.H., Stoffer D.S. An approach to time series smoothing and forecasting using the EM algorithm // *Journal Series Analysis*, 1982.–vol.3, №4.–P.253-264.
 167. Stahlberger A., Riedmiller M.: Fast Network Pruning and Feature Extraction using unit-OBS Algorithm // *Advances in Neural Information Processing Systems*, 1997.–vol.9.–P.655–661.
 168. Stone F. Cross - validatory choice and assessment of statistical prediction // *Journal of the Royal Statistical Society*, 1974.–vol.34.–P.111-147.
 169. Sugiyama M., Ogawa H. Subspace information criterion for model selection // *Neural Computation*, 2001.–vol.13, № 8.–P.1863-1889.
 170. Szyld D.B., Jones M.T. Two-stage and multisplitting methods for the parallel solution of linear systems // *SIAM J. Matrix Anal. Appl.*, 1992.–vol.2.–P.671-679.
 171. Tanzi E. A. *Text-Book of Mental Diseases*. Rebman, 1909.–345 p.
 172. Uttley A.M. A theory of the mechanism of learning based on conditional probabilities // *Proc. of the 1st International Conference on Cybernetics*, Namur, Gauthier-Villars, Paris (France), 1956.–P.83-92.
 173. Varga R.S. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1962.–322 p.
 174. Weigend A.S., Huberman B.A., Rumelhart D.E. Predicting the future: a connectionist approach.// *International Journal of Neural Systems*, 1990.– vol.1 №3.– P.193-209.
 175. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral science. Ph.D. Thesis.- Harvard University, Cambridge, MA, 1974.–120 p.
 176. Wetherill G. B. *Regression Analysis with Applications*. Monographs of Statistic and Applied Probability, Chapman and Hall, London, 1986.–324 p.

177. Widrow B, Winter R.G. Neural nets for adaptive filtering and adaptive pattern recognition // IEEE Transactions on Computers, 1988.–vol.37, №3.–P.25-39.
178. Widrow B, Winter R.G., Baxter R.A. Learning phenomena in layered neural networks.// Proc. First International Conference on Neural Networks, IEEE, 1987.–vol. II.–P.411-429.
179. Widrow B. Generalisation and information storage in networks of adaline “neurons” // Self-Organizing Systems (M.C. Yovitz, G.T. Jacobi, and G.D. Goldstein, eds.), Washington, D.C.: Sparta, 1962.–P.435-461.
180. Widrow B., Hoff M.E., Jr. Adaptive switching circuits // IRE WESCON Convention Record, 1960.–P.96-104.
181. Widrow, B., Walach, E. On the statistical efficiency of the LMS algorithm with nonstationary inputs // IEEE Trans.on Information Theory, 1984.–vol.30, №2.–P.211-221.
182. Williamson J. R. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps // Neural Networks, 1996.–vol. 9, №5.–P.881-897.
183. Wynne-Jones M. Node splitting: A constructive algorithm for feed-forward neural networks // Neural Computing and Applications, 1993.–vol.1, №1.–P.17-22.
184. Xu J. Iterative methods by space decomposition and subspace correction // SIAM Review, 1992.–vol.34.–P.581-613.
185. Yserentant H. Old and new convergence proofs for multigrid methods // Acta Numerica, 1993.–P.285-326.
186. Yuille A.L., Kammen D.M., Cohen D.S. Quadrature and the development of orientation of selective cortical cells by Hebb rules // Biological Cybernetics, 1984.–vol.61.–P.183-194.
187. Ziehe A., Müller K.R. TDSEP – an efficient algorithm for blind separation using time structure // ICANN'98, Skovde, 1998.–P.675-680.

ЗМІСТ

ПЕРЕДМОВА	5
----------------------------	---

Розділ 1.

Нейронні мережі: означення та основні властивості	12
1.1. Історичні аспекти розвитку штучних нейронних мереж	14
1.2. Нейрон та його моделювання.	18
1.2.1. Біологічний нейрон.	18
1.2.2. Формальний нейрон	22
1.2.3. Аналогова та дискретна реалізації формального нейрона	28
1.3. Задачі штучних нейронних мереж.	31
1.3.1. Проблема представлення неперервних функцій.	32
1.3.2. Проблема класифікації та категоризації	34
1.4. Архітектура штучних нейронних мереж.	37
1.4.1. Ієрархічні структури	43
1.4.1.1. Одношарові нейронні мережі прямого поширення.	43
1.4.1.2. Багатошарові нейронні мережі прямого поширення.	45
1.4.1.3. РБФ – мережі	47
1.4.2. Конкурентні структури	48
1.4.2.1. WTA структура Ліпмана–Хеммінга.	48
1.4.2.2. Нейронна мережа Кохонена.	51
1.4.2.3. Нейронні мережі зустрічного поширення.	53
1.4.3. Рекурентні структури	56

1.4.3.1.	Нейронна мережа Хопфілда	57
1.4.3.1.1.	Бінарна модель	58
1.4.3.1.2.	Асоціативна пам'ять	62
1.4.3.1.3.	Неперервна модель	63
1.4.3.1.4.	Комбінаторна оптимізація	64
1.4.3.2.	Двонаправлена асоціативна пам'ять	65
1.4.3.3.	Стохастичні рекурентні мережі	68
1.4.4.	Нейронні мережі типу ART	70
1.4.4.1.	Нейронна мережа типу ART-1	71
1.4.4.2.	Нейронна мережа типу АНН	75
1.4.4.3.	Ієрархічні мережі типу ART	76

Розділ 2.

Методи навчання з учителем.	80	
2.1.	Принципи формування цільової функції	81
2.2.	Методи навчання одношарових ШНМ прямого поширення	83
2.2.1.	Лінійна регресія	83
2.2.2.	Метод градієнтного спуску	85
2.2.3.	Метод найменших квадратів	88
2.2.4.	Метод навчання Adaline та персептрона Розенблатта	90
2.2.5.	Метод „кишені”	93
2.2.6.	Методи градієнтного спуску та найменших квадратів для Adaline та персептрона Розенблатта	94
2.3.	Методи навчання багатошарових ШНМ прямого поширення	95
2.3.1.	Метод зворотного поширення	97

2.3.2.	Метод еластичного зворотного поширення	102
2.3.3.	Метод Гауса–Ньютона	107
2.3.4.	Метод Левенберга–Маркара	109
2.4.	Методи навчання ШНС типу РБФ	111
2.4.1.	Метод точного прямого центрування	112
2.4.2.	Метод інтерполяційного прямого центрування	113
2.4.3.	Метод центрування на підмножині	114
2.4.4.	Метод К середніх	118
2.4.5.	Імовірнісні методи	119
2.4.5.1.	Метод максимізації математичного сподівання	121
2.4.5.2.	Метод максимізації математичного сподівання для нелінійних динамічних систем	124

Розділ 3

Оптимізація архітектури нейронних мереж	128	
3.1.	Методи визначення реальної похибки	128
3.2.	Дилема відхилення / дисперсії	131
3.3.	Методи оптимізації параметрів	132
3.3.1.	Метод попередньої зупинки	132
3.3.2.	Регуляризація	133
3.3.3.	Усреднення	136
3.3.4.	Методи відсікання	140
3.3.4.1.	Метод „пригнічення”	141
3.3.4.2.	ОБД-метод	142
3.3.4.3.	ОBS-метод	145
3.3.4.4.	OLS-метод	147

3.3.5.	Методи нарощування	152
3.3.5.1.	Метод каскадної кореляції.	152
3.3.5.2.	Метод ітераційного нарощування	155

Розділ 4

Методи навчання без учителя		157
4.1.	Аналіз важливих компонентів.	158
4.2.	Аналіз незалежних компонентів.	161
4.2.1.	Технологія сліпого відокремлення стаціонарних сигналів.	161
4.2.1.1.	Максимум ентропії.	164
4.2.1.2.	Мінімум взаємної інформації.	166
4.2.1.3.	Природний градієнт	170
4.2.1.4.	Оцінка максимуму імовірності.	171
4.2.2.	Технологія сліпого відокремлення нестационарних сигналів.	173
4.2.2.1.	Метод поточної декореляції.	174
4.2.2.2.	Метод автокореляції	177
4.2.3.	Конволютивне сліпе відокремлення сигналів	180
4.2.3.1.	Метод конволютивної поточної декореляції.	182
4.2.3.2.	Метод конволютивної автокореляції.	184
4.3.	Правило Хебба.	187
4.3.1.	Методи обмеження для навчання за правилом Хебба.	190
4.3.2.	Мультиплікативні та субтрактивні обмеження.	193
4.4.	Правило Ойя	198
4.4.1.	Правило Ойя для одного нейрона.	198
4.4.2.	Правило Ойя для підпростору	200
4.4.3.	Узагальнене правило Хебба	203

4.5.	Методи навчання конкурентних нейронних мереж	210
4.5.1.	Простий метод конкурентного навчання.	215
4.5.2.	Кластерні методи навчання.	216
4.5.3.	Карти Кохонена	219
4.5.4.	Нейронний газ	223
4.5.5.	Нейронний газ із конкурентним правилом Хебба.	226
4.5.6.	Нейронний газ із нарощуванням.	228
4.6.	Методи навчання ШНС типу ART.	234
4.6.1.	ART-подібні нейронні мережі	234
4.6.2.	Нейронна мережа Fuzzy ART	238
4.6.3.	Нейронні мережі S-Fuzzy ART та SART	241
4.6.4.	Нейронна мережа FOSART.	243

Розділ 5

Ітераційні методи на нейронних мережах.		248
5.1.	Принципи взаємодії нейронів при реалізації ітераційних методів.	248
5.2.	Одностадійні асинхронні ітераційні методи.	255
5.2.1.	Базові поняття	255
5.2.2.	Метод хаотичних ітерацій.	257
5.2.3.	Метод асинхронних ітерацій.	263
5.2.4.	Метод асинхронних ітерацій з нерухомими точками.	265
5.3.	Двостадійні ітераційні методи	270
5.3.1.	Початкові поняття про мультирозщеплюючі методи	270
5.3.2.	Мультирозщеплюючі ітераційні методи.	276

5.3.2.1.	Синхронні двостадійні ітераційні методи	278
5.3.2.2.	Макроасинхронні двостадійні методи	282
5.3.2.3.	Повністю асинхронні двостадійні методи	285
5.4.	Локально-асинхронні ітераційні методи	286
5.4.1.	Початкові поняття про локально-асинхронні методи	286
5.4.2.	Локально-асинхронний метод розв'язування операторного рівняння	289
5.4.3.	Локально-асинхронний метод розв'язування послідовності операторних рівнянь	296
5.5.	Ітераційні методи розв'язування крайових задач математичної фізики	302
5.5.1.	Базові поняття	302
5.5.2.	Локально – асинхронний метод для рекурентних нейронних мереж	305
5.5.3.	Ітераційний метод адитивної корекції для кліткових нейронних мереж	310

Розділ 6

Мультисіткові методи	312	
6.1.	Двосіткові ітераційні методи	312
6.2.	Мультисіткові ітераційні методи	315
6.2.1.	Узагальнення ітераційної схеми згладжування	318
6.2.2.	Пролонгація та рестрикція	321
6.2.2.1.	Вузлова пролонгація	323
6.2.2.2.	Доменна пролонгація	327
6.2.2.3.	Рестрикція	332
6.3.	Мультисітковий ітераційний метод для лінійних крайових задач	336

6.3.1.	Алгоритм мультисіткового методу	336
6.3.2.	Збіжність мультисіткового методу	341
6.4.	Ієрархічні методи	343
6.4.1.	Ієрархічний базис	343
6.4.2.	Метод ієрархічної релаксації	346
6.4.3.	Асинхронний метод ієрархічної релаксації	349
6.5.	Асинхронний мультисітковий метод для лінійних крайових задач	354
6.6.	Асинхронний мультисітковий метод для нелінійних крайових задач	359
6.7.	Асинхронний мультисітковий метод для нестационарних крайових задач	361
6.8.	Нейронний мультисітковий метод в мережах прямого поширення	365
6.9.	Нейронний мультисітковий метод у рекурентних мережах	368
6.10.	Приклади розв'язування крайових задач на рекурентних нейронних мережах	376

СПИСОК ЛІТЕРАТУРИ	384
------------------------------------	------------

Наукове видання

**НОВОТАРСЬКИЙ Михайло Анатолійович
НЕСТЕРЕНКО Борис Борисович**

**ШТУЧНІ НЕЙРОННІ МЕРЕЖІ:
ОБЧИСЛЕННЯ**

Комп'ютерний набір
С.В. Новотарська

Комп'ютерна верстка
М.А. Новотарський

Підп. до друку 01.10.2003. Формат 60×84/16. Папір тип. №1. Офс.друк.
Умовн. друк. арк. 23,72. Обл.-вид. арк. 27,33. Зам. №

Інститут математики НАН України
01601, Київ 4, ГСП, вул. Терещенківська, 3.