

МОДУЛЬНОЕ ПРЕДСТАВЛЕНИЕ НЕЙРОННЫХ СЕТЕЙ

Abstract: In this article we consider the possibility of modular presentation of the most commonly used paradigms of artificial neural networks and discuss advantages of such presentation. As a basic element of modular presentation we propose to use "layer" of neurons that is a group of neurons working in parallel and performing the same functions. Model problem is described that illustrates the advantages of modular presentation and architecture of modular network of "new type" is given.

Key words: neural networks, neuron, modular neural networks.

Аномація: У статті розглядається можливість модульного представлення найпоширеніших парадигм штучних нейронних мереж та обговорюються переваги такого представлення. Як базовий елемент модульного представлення запропоновано використовувати «шар» нейронів, тобто групу нейронів, що працюють паралельно й виконують однакові функції. Розглянуто модельну задачу, яка ілюструє переваги модульного представлення й приведено архітектуру нейронної мережі «нового типу».

Ключові слова: нейронні мережі, нейрон, модульні нейронні мережі.

Аннотация: В статье рассматривается возможность модульного представления наиболее распространенных парадигм искусственных нейронных сетей и обсуждаются преимущества такого представления. В качестве базового элемента модульного представления предложено использовать «слой» нейронов, то есть группу нейронов, работающих параллельно и выполняющих одинаковые функции. Рассмотрена модельная задача, иллюстрирующая преимущества модульного представления и приведена архитектура нейронной сети «нового типа».

Ключевые слова: нейронные сети, нейрон, модульные нейронные сети.

1. Введение

В настоящее время известно большое число разнообразных парадигм искусственных нейронных сетей. Однако различие между парадигмами, зачастую, заключается только в методах обучения, то есть в правилах изменения настроечных параметров алгоритма, и наличии либо отсутствии латеральных и обратных связей. В рамках парадигм сети могут различаться правилами получения результата и вычисления ошибки. Например, многослойный персептрон может иметь на выходе пороговую функцию в случае использования в качестве классификатора или линейную, если используется как интерполятор.

Если условно разделить нейронные сети на «простые» и «комплексные», то архитектура «простых» сетей строится на основе нескольких структурных составляющих. Такими составляющими являются процессорный элемент (нейрон), слой или группа процессорных элементов, работающих параллельно, и набор слоев, образующих собственно нейронную сеть. В некоторых типах сетей особую роль играют правила передачи сигналов между нейронами слоя или слоями сети. «Комплексные» нейронные сети, обычно, можно представить как объединение нескольких «простых».

Ограниченное число структурных составляющих нейронных сетей приводит к естественному вопросу о возможности единообразного описания различных архитектур. Для ответа на этот вопрос необходимо определить, какие элементы потребуются для описания сетей, какими свойствами должны обладать такие элементы, а также определить правила использования этих элементов. Очевидно, что единообразное представление различных типов нейронных сетей должно позволять не только реализовать уже существующие парадигмы, но и предоставить средства для получения «новых типов» сетей.

Одним из возможных, и на наш взгляд, наиболее привлекательных путей единообразного описания нейронных сетей является их представление в виде модульных структур. В данной работе предлагается способ представления нейронных сетей в виде модульных структур, а также рассматриваются особенности такого представления.

Практически все широко известные и применяемые на практике нейронные сети могут использоваться в качестве классификаторов. Поэтому, без ограничения общности, идею модульного представления можно рассматривать на примере задачи классификации. Для большей определенности будем говорить о нейронных сетях, обучаемых «с учителем». То есть о таких сетях, в которых требуемый выход известен для каждого образа из обучающей выборки. Последнее важно в том смысле, что не только позволяет конкретизировать анализ, но и предполагает наличие дополнительного модуля в представлении сети.

Возможности модульного представления рассматриваются на примере наиболее известных сетей прямого распространения, таких как многослойный персептрон (MLP), радиальные базисные функции (RBF) и машины опорных векторов (SVM) [1, 2]. В качестве примеров модульного представления сетей с обратными и латеральными связями рассмотрены рекуррентные, ассоциативные [3] и ассоциативно-проективные сети [4]. В заключение приведен тривиальный пример построения сети «нового типа».

2. Модульное представление сетей прямого распространения

В основе большинства нейронных сетей лежит представление нейрона как процессорного элемента с настраиваемыми параметрами, например, в MLP. По аналогии с биологическим нейроном, процессорный элемент имеет произвольное число входов и один выход, причем каждому входу сопоставлен настраиваемый параметр, называемый весом связи. В некоторых сетях, таких как SVM, процессорные элементы не всегда выделяют в явном виде, однако и такие сети могут быть описаны при помощи «нейронов».

Пусть вектор X описывает образ, подаваемый на входы нейрона. Обозначим выход нейрона через y , тогда функционирование нейрона в общем виде может быть представлено следующим образом:

$$y = F(G(V, X)), \quad (1)$$

где V – вектор весов связей, F – передаточная функция нейрона, а G – некоторая функция «взвешивания». В зависимости от того, используется модель нейрона с порогом или без, размер вектора весов может быть равен либо быть больше числа входов нейрона. Существуют нейронные сети, для которых может быть использовано более одного настроенного параметра на связь. Пример такой сети обсуждается ниже.

Функция «взвешивания» является скалярной функцией векторного аргумента. Вид функции «взвешивания» определяет свойства нейрона в пространстве входов. Например, в наиболее часто встречающейся модели нейрона функция взвешивания определяется как скалярное произведение входного вектора и вектора весов с учетом порога:

$$Gp(V, X) = v_0 + (V', X) = v_0 + \sum_{i=1}^N v_i x_i, \quad (2)$$

где N – количество входов нейрона и v_0 – порог. Индекс p – «персептронная» в имени функции введен для удобства ссылок на различные функции «взвешивания». Результат вычисления функции (2) пропорционален расстоянию до гиперплоскости, заданной вектором нормали $V' = \{v_1, \dots, v_N\}$ и смещением v_0 в пространстве входов нейрона.

Для нейронов, использующих функцию взвешивания (2), в качестве передаточной функции наиболее часто используется сигмоида

$$Fp(t) = \frac{1}{1 + e^{-\alpha t}}.$$

Во многих других, практически важных случаях, результат функции взвешивания пропорционален расстоянию до некоторой точки во входном пространстве. Например, в сетях RBF функция взвешивания представляется в виде Евклидова расстояния до опорной точки, координаты которой можно рассматривать как веса связей:

$$Gr(V, X) = v_0 + \|V' - X\| = v_0 + \sqrt{\sum_{i=1}^N (v_i - x_i)^2}, \quad (3)$$

где индекс r – означает «радиальная» функция. Порог v_0 в функции (3) по аналогии с функцией (2) введен как весовой коэффициент, и в данном случае имеет смысл «радиуса действия» передаточной функции. В RBF-сетях обычно используют функции типа Гауссиана, то есть в качестве передаточной можем рассматривать функцию вида

$$Fr(t) = e^{-\alpha t}.$$

Следует добавить, что в сетях с латеральными связями, например, в сетях ассоциативной памяти, иногда рассматривают дополнительный «установочный» вход нейрона u , при этом перед пересчетом сети принимается $y = u$ и в дальнейших вычислениях этот вход обычно не используется.

Временно абстрагируясь от алгоритмов обучения, попробуем выделить то общее, что есть в работе (пересчете) выбранных сетей прямого распространения, то есть в MLP, RBF и SVM. Без ограничения общности можем рассматривать задачу классификации на два класса. В этом случае нейронная сеть имеет один выход. Введем следующие обозначения: X – вектор на входе сети и z – выход всей сети. Будем использовать нижние индексы для нумерации входов или нейронов в одном слое, а верхние для обозначения слоев сети.

Рассмотрим двухслойный персептрон. Пусть y – выход нейрона первого слоя. На выходе первого слоя получаем вектор $Y = \{y_1, \dots, y_M\}$, где M – число нейронов первого слоя. Тогда работа двухслойного персептрона описывается выражением

$$z = Fp^2(Gp^2(V^2, Y)) = Fp^2(v_0^2 + \sum_{i=1}^M v_i^2 \cdot Fp^1(Gp^1(V_i^1, X))).$$

В режиме классификатора на два класса выход персептрона (искомый класс) определяется сравнением с некоторым порогом T . Входной вектор будет отнесен к первому из двух классов, если $z < T$, иначе – ко второму. Поэтому нелинейность на выходе в данном случае является излишней и для выходного нейрона можно использовать линейную передаточную функцию $Fp(t) = t$. Следовательно, работу двухслойного персептрона можно описать выражением

$$z = v_0^2 + \sum_{i=1}^M v_i^2 \cdot Fp(Gp(V_i^1, X)). \quad (4)$$

Сети RBF и SVM основаны на использовании «опорных векторов», то есть определенным образом выбранных точек во входном пространстве. Пусть выбрано M опорных точек, тогда работа сети RBF описывается выражением

$$z = v_0 + \sum_{i=1}^M v_i \cdot Fr(Gr(V_i, X)), \quad (5)$$

а работа сети SVM

$$z = v_0 + \sum_{i=1}^M v_i \cdot K(V_i, X), \quad (6)$$

где K – так называемая функция ядра и имеет достаточно широкий спектр представлений. Однако оказывается, что используемые на практике функции ядра могут быть представлены в виде (1), при этом в качестве функции взвешивания будет использована одна из функций (2) или (3). В качестве примера можно привести Гауссово ядро:

$$K(V, X) = e^{-\alpha \|V-X\|^2},$$

которое в точности совпадает с одной из реализаций RBF, или полиномиальное ядро:

$$K(V, X) = (1 + (V, X))^d,$$

которое в случае $d = 1$ дает нейрон «персептронного» типа.

Сравнение выражений (4) – (6) показывает, что в смысле работы (прямого пересчета или экзамена) SVM является обобщением RBF и двухслойного персептрона. Тем не менее, RBF и SVM остаются различными парадигмами нейронных сетей, поскольку используют разные правила формирования набора опорных векторов и настройки весов выходного нейрона.

Возвращаясь к исходной задаче, а именно, к представлению нейронных сетей в виде модульных структур, видим, что рассматриваемые сети прямого распространения могут быть описаны при помощи двух слоев (модулей) со следующими правилами работы элементов модулей:

$$y = F(G(V^1, X)) \text{ и } z = Fp(Gp(V^2, Y)). \quad (7)$$

В принципе, рассматриваемые сети допускают модульное представление и в виде наборов нейронов, причем для каждой из таких сетей требуется не более двух типов процессорных элементов. Имеются несколько подходов к описанию нейронных сетей в виде наборов процессорных элементов, например, описания при помощи графов [5]. Однако существуют ассоциативные сети, которые требуют учета кооперативного поведения нейронов внутри слоя. Поэтому минимальным элементом модульного представления предлагается считать слой.

Нетрудно проверить, что, используя функцию взвешивания (3) и учитывая кооперативное поведение, модульное представление (7) описывает такие методы распознавания как метод потенциальных функций, а также методы ближайшего и k – ближайших соседей.

Первые преимущества модульного представления сетей прямого распространения вытекают из принципа решения задачи классификации при помощи таких сетей. Сети RBF и SVM относятся к так называемым классификаторам с преобразованием входного пространства. Суть использования этих классификаторов заключается в том, что находят такое преобразование входного пространства $\Omega \rightarrow R$, что в пространстве R классы становятся линейно разделимыми, и для определения границы между классами используется линейный персептрон (линейное разделяющее правило).

Доказано, что MLP и RBF являются универсальными аппроксиматорами. То есть, если граница между классами существует, то может быть построен классификатор, в точности описывающий такую границу [6–8]. Однако существуют две принципиальные проблемы применения классификаторов с преобразованием исходного пространства признаков. Первая – определение наиболее подходящих функций преобразования (в случае RBF – это функции Fr) и вторая, непосредственно связанная с первой, это количество опорных векторов, необходимое для достижения линейной разделимости. Возникает противоречие между требованием линейной разделимости и ограниченным числом опорных векторов.

Для разрешения этого противоречия (уменьшения числа опорных векторов), возможно использование компромиссного решения, когда после преобразования пространства, в нашем случае применения «RBF модуля», используется не линейный, а, например, двухслойный персептрон с небольшим числом нейронов в первом (скрытом) слое. Тогда можно не требовать линейной разделимости в новом пространстве.

Тем не менее модульное представление имеет смысл, если позволяет описывать процесс обучения нейронных сетей. Алгоритмы обучения MLP и RBF представлены в [1], а SVM в [2]. Не вдаваясь в детальное описание, отметим следующее: в рассматриваемых сетях алгоритмы изменения настроечных параметров относятся к слою, хотя и требуют знания об ошибке на выходе всей сети. Следовательно, для представления нейронной сети модульной структурой необходимо определить, какими свойствами должен обладать модуль (слой) и какие функции должна поддерживать модульная сеть в целом, независимо от типов входящих в нее модулей.

Определим свойства, которыми должны обладать слой и модульная сеть. В общем случае, полнофункциональный слой должен обладать следующими свойствами:

1) Архитектура – слой имеет входы, выходы и некоторое количество нейронов (процессорных элементов), работающих параллельно. Тут следует отметить, что параллельность работы нейронов слоя относится к «внешнему» наблюдателю и означает только то, что в каждый момент времени определены выходы всех нейронов слоя.

2) Работа – алгоритм прямого пересчета, то есть правила получения результата на выходах для текущих значений на входах.

3) Обучение – алгоритм изменения настроечных параметров и определение ошибки на каждом входе по сигналу ошибки или известным ошибкам на каждом из выходов модуля (слоя), а

также, что очень важно, правила удаления или добавления входов или выходов по указанному извне номеру и/или согласно внутреннему алгоритму слоя.

Рассмотрим назначение и использование функций слоя (модуля). Во-первых, следует отметить, что не все функции являются обязательными. В зависимости от типа и назначения модуль может не содержать нейронов и не иметь алгоритма обучения, то есть не содержать настроечных параметров. Кроме того, возможность определения ошибки на каждом входе является свойством алгоритма обучения данного модуля и определяет ограничения при построении модульной архитектуры.

Так, например, если предполагается использовать слой MLP с обучением путем обратного распространения ошибки, то последующий модуль обязательно должен определять ошибку для каждого своего входа, поскольку в алгоритмах обучения MLP-слоя используется величина этой ошибки. Для некоторых реализаций слоя RBF в режиме обучения достаточно просто знать о наличии ошибки на данном входе.

В случае моделирования SVM существуют две возможности организации обучения в модульном представлении: либо слой может получать сигнал о наличии ошибки на выходе и соответствующий опорный вектор (нейрон) будет удален, либо получить информацию о номерах удаленных входов последующего модуля. При этом соответствующие выходы (опорные векторы) будут удалены из слоя. Второй вариант является более универсальным, поскольку в модульном представлении аналогично реализуются алгоритмы удаления ненужных связей (pruning), используемые во многих сетях прямого распространения.

Наличие у модулей правил удаления входов и выходов обеспечивает реализацию механизмов удаления ненужных связей (настроечных параметров) и нейронов (процессорных элементов). Простейший алгоритм заключается в том, что удаляются связи, имеющие нулевой вес. В рассматриваемом примере, где второй слой каждой из сетей содержит один нейрон, удаление связи означает удаление входа этого слоя, а значит, выходы соответствующих нейронов первого слоя не оказывают влияния на результат работы сети и соответствующие нейроны первого слоя можно удалить. Описанный механизм соответствует последнему этапу обучения SVM.

Механизм добавления входов и выходов позволяет реализовать варианты сетей, предполагающие создание новых опорных векторов либо за счет включения новой части обучающей выборки, либо на основе специфических эвристик. Важным является то, что количество входов или выходов модуля влияет только на количество внешних связей соседних модулей, но не требует знания ни об алгоритме работы, ни об алгоритме обучения смежных модулей. Кроме того, число входов и выходов всей модульной сети остается неизменным.

Пусть модули обладают перечисленными свойствами, тогда нетрудно выделить функции, которые должны обеспечиваться модульной сетью в целом. Причем, очевидно, что функции сети не должны предполагать «знания» о том, какой именно алгоритм обучения или работы реализует конкретный модуль.

Тогда модульная сеть должна обеспечивать:

1) Архитектура – входы, выходы и некоторое количество модулей (слоев), связанных направленными связями. Направление связей задает прямой пересчет сети.

2) Работа – последовательность пересчета модулей и правило формирования выхода всей сети.

3) Обучение – получение информации об ошибке в сети, правила распространения ошибки по сети, последовательность пересчета модулей при обучении, а также, обеспечение согласования количества связей в случае изменения количества входов или выходов у модулей.

Кроме перечисленных функций, для моделирования некоторых типов сетей могут быть введены правила автоматического добавления или удаления модулей заданного типа.

3. Модульное представление сетей с обратными и латеральными связями

Большинство сетей с обратными и латеральными связями имеют естественное представление в виде модульных структур. Наиболее распространенными сетями с обратными и латеральными связями являются рекуррентные, ассоциативные и ассоциативно-проективные сети.

Рекуррентные сети наиболее широко применяются при анализе временных рядов, в задачах разбора предложений и других, обычно там, где данные представляют некоторую последовательность. Рекуррентные сети строятся на основе многослойного персептрона и включают все виды обратных связей. Среди рекуррентных сетей выделяют четыре типа, которые представлены на рис. 1. Сети различаются только способом образования петель обратных связей и обычно имеют двухслойную архитектуру.

Рекуррентная сеть Фраскони-Гори-Сода [9] показана на рис. 1а и является сетью с локальными обратными связями. В этой архитектуре выход каждого нейрона скрытого слоя соединен со входом того же нейрона. В архитектуре, предложенной Нарендра и Партасарати [10] (рис. 1б), вся сеть охвачена обратной связью, то есть, выход каждого нейрона последнего слоя соединен с каждым из нейронов первого скрытого слоя. Элман [11] (рис. 1в) рассматривал применение сети с обратной связью, охватывающей только скрытый слой. То есть выход каждого нейрона скрытого слоя соединен со входами всех нейронов этого же слоя. Вильямс и Циплер [12] (рис. 1г) рассматривали возможности применения «полносвязной» рекуррентной сети, в которой выход каждого нейрона сети соединен со входами всех нейронов на каждом слое.

Во всех рассмотренных рекуррентных сетях обратная связь выполняет функцию буфера задержки на один такт, то есть, кроме входного вектора на нейроны сети подается состояние выходов соответствующих нейронов на предыдущем такте распознавания. Существуют модификации рекуррентных сетей, в которых буфер задержки выполняет более сложную функцию, например, задержка с накоплением, когда текущее состояние нейронов добавляется к содержимому буфера, взятому с определенным коэффициентом. Новое состояние буфера определяется по формуле

$$B_{t+1} = \alpha Y_t + (1 - \alpha) B_t, \quad (8)$$

где B – состояние буфера, Y – выходы нейронов на такте t и $0 < \alpha < 1$ – коэффициент, который часто называют моментом.

Часто на вход рекуррентных сетей подают не отдельный элемент последовательности, а подпоследовательность определенной длины. В таком случае для распознавания последовательностей можно применять и обычный многослойный персептрон, а буфер задержки

используется вне сети для формирования входного вектора. На рис. 2 приведены примеры использования многослойного персептрона (рис. 2а) и рекуррентной сети (рис. 2б) для анализа последовательностей.

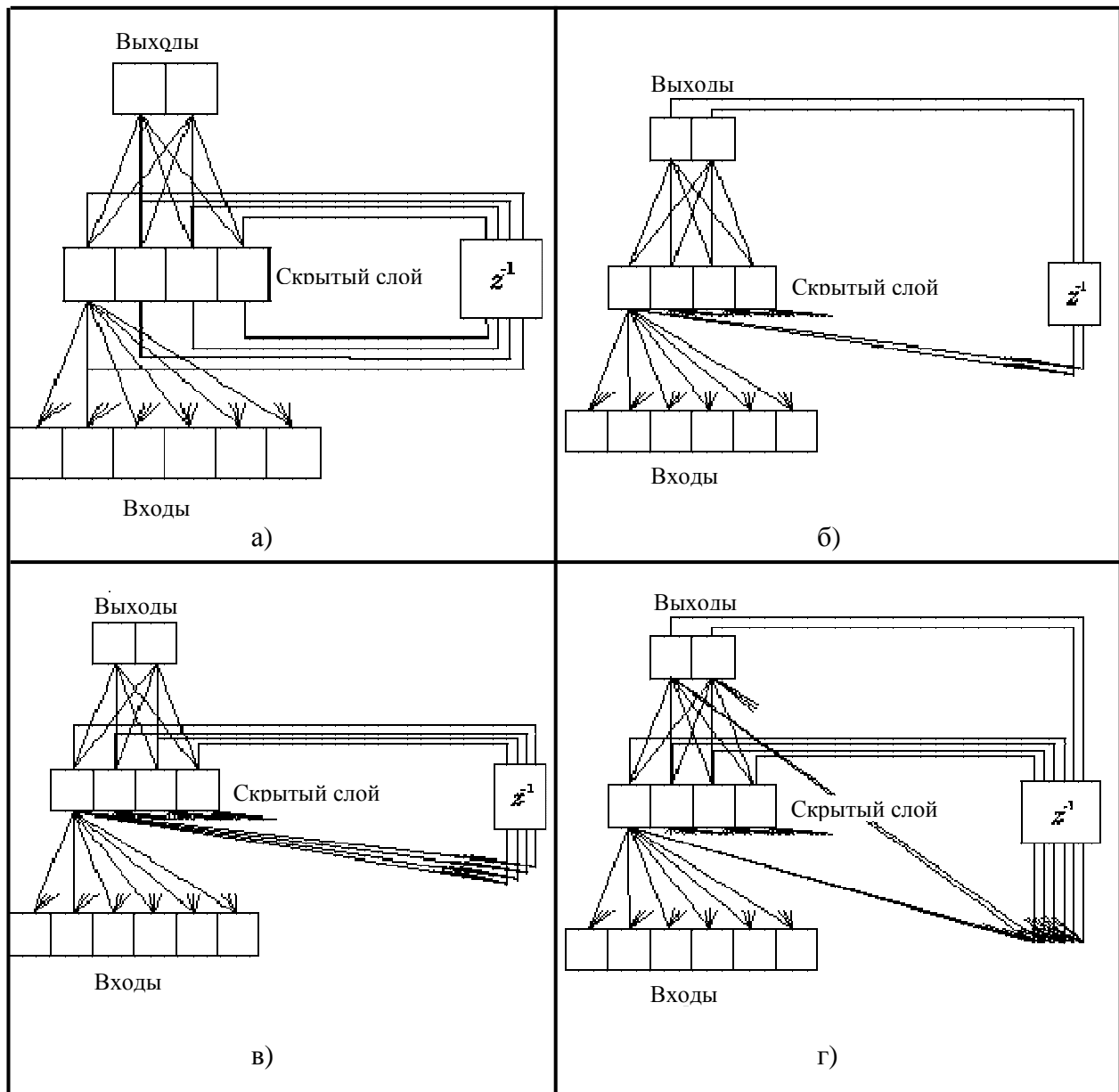


Рис. 1. Примеры рекуррентных сетей на основе многослойного персептрона

Непосредственно из рис. 2 видно, что, используя только два типа слоев – MLP и слой задержки (Delay), можно построить обе рассматриваемые сети. Пример построения этих сетей с использованием модулей, реализующих соответствующий слой, приведен на рис. 3. Очевидным образом представляются и остальные архитектуры рекуррентных сетей.

Ассоциативные сети типа Хопфилда содержат один слой нейронов с латеральными связями [3, 13]. В смысле функционирования основным отличием ассоциативных сетей от сетей прямого распространения и рекуррентных сетей является не столько тип связей, сколько процесс конвергенции сети. Конвергенция сети – это процесс последовательного пересчета сети несколько раз для одного и того же входного воздействия, что позволяет выходному вектору сети сместиться

в область центра аттрактора. Другими словами, после конвергенции на выходе ассоциативного слоя появляется один из ранее запомненных векторов.

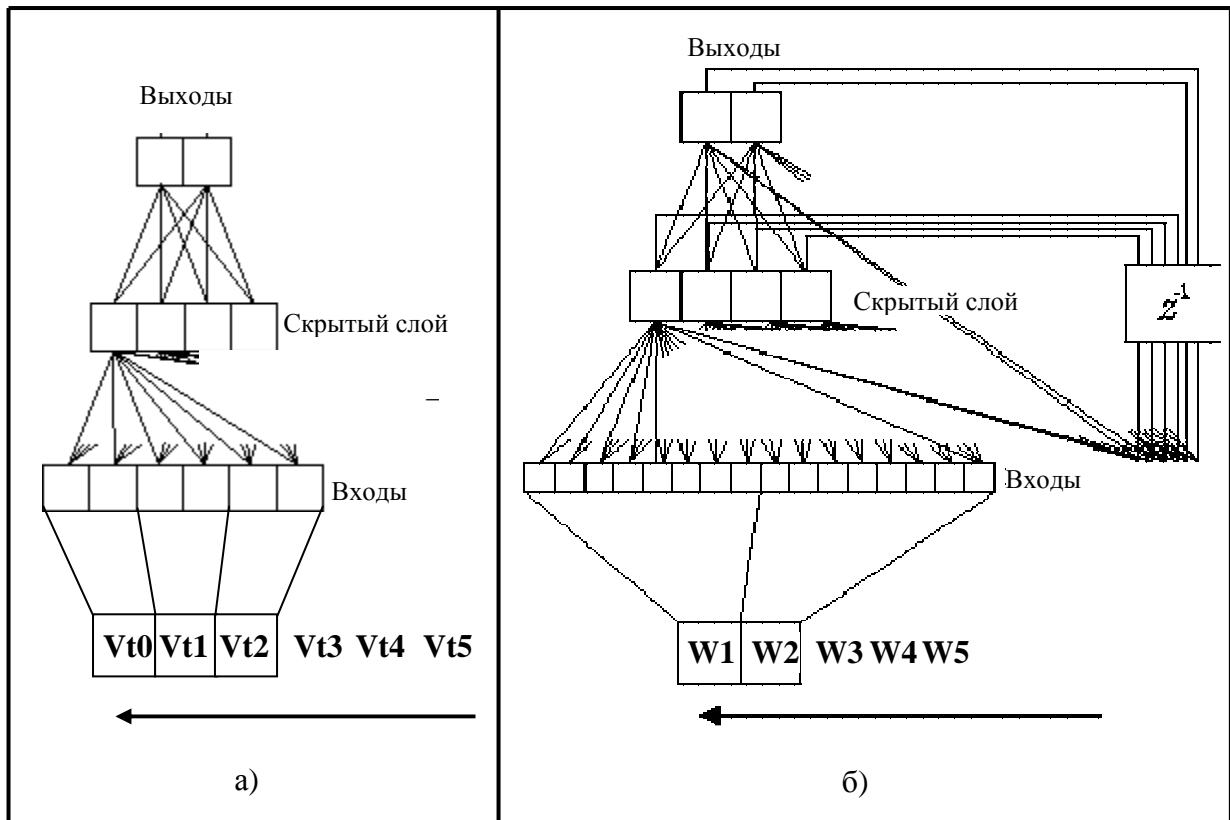


Рис. 2. Пример использования сетей для анализа временных рядов

Выше мы говорили, что для функционирования модульной нейронной сети нам нужен некоторый алгоритм управления потоками данных (маршрутизатор), который не зависит ни от количества, ни от типов модулей, входящих в сеть. Одним из требований к маршрутизатору является возможность автоматически определять наличие циклов в «сконструированной» модульной сети и возможность задавать параметры работы с каждым из циклов. Параметрами цикла, собственно, является количество пересчетов модулей внутри цикла на один входной вектор либо признак остановки пересчета. Тут входной вектор имеет смысл вектора, поданного на вход цикла, а не всей сети. Более подробно циклы в модульных сетях рассмотрены в [14].

В ассоциативных сетях типа Хопфилда алгоритм работы нейронов в точности совпадает с алгоритмом, используемым в MLP. Оказывается, если мы можем задавать признак окончания счета циклов, то в модульном представлении, для реализации такого типа ассоциативных сетей, нам не нужно вводить дополнительный тип слоя. Если на рис. 1 в убрать выходной слой нейронов, мы получим реализацию ассоциативной памяти, где для одного входного вектора пересчет в цикле MLP-задержка-MLP производится до тех пор, пока активность нейронов на выходе MLP слоя не перестанет изменяться.

В случае ассоциативно-проективных сетей [4], модульное представление вообще является естественным, поскольку в сетях такого типа не только определены функции каждого из слоев (в этих сетях существуют слои нескольких типов, отличающихся алгоритмом пересчета и выполняемыми функциями), но и допускаются почти произвольные потоки данных между слоями.

4. Использование модульного представления

Преимущества модульного представления рекуррентных и ассоциативных нейронных сетей, а также сетей прямого распространения, следуют из способа построения архитектуры в модульной реализации. Поскольку сеть для решения конкретной задачи более не реализуется единым алгоритмом, а представлена в виде унифицированного набора алгоритмов, относящихся исключительно к слою, то, имея средства динамически (или в интерактивном режиме) менять тип используемого слоя, произвольным образом объединять слои и задавать направления потоков данных, получаем возможность достаточно просто «сконструировать» нейронную сеть наиболее удачно решающую конкретную прикладную задачу.

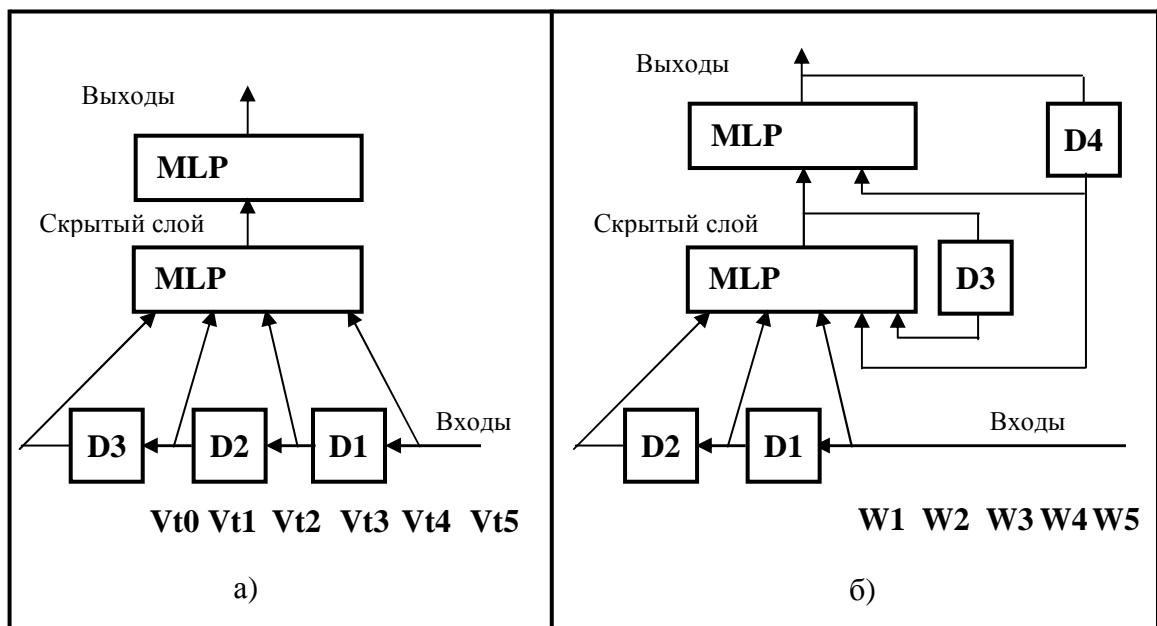


Рис. 3. Пример модульных сетей для анализа временных рядов

Инструментом, позволяющим интерактивно конструировать нейронную сеть из модулей для решения прикладных задач, является САПР МНН [15].

Рассмотрим модельную задачу, иллюстрирующую преимущества модульного представления. В качестве примера возьмем задачу классификации, известную как «исключающее ИЛИ». Задача формулируется так: на плоскости даны четыре точки с координатами (0,0), (1,1), (0,1) и (1,0). Первые две точки относятся к одному классу, третья и четвертая к другому. В свое время Минский и Пейперт показали, что однослойный персептрон не решает эту задачу, и она стала классической для проверки нейронных сетей. Известно, что минимальная архитектура MLP (рис. 4а), решающего данную задачу, содержит два нейрона и пять связей (весов или настраиваемых параметров).

Однако, если вместо использования «классических» нейронных сетей мы поставим последовательно два RBF-слоя, то для решения задачи нам потребуются всего два нейрона и три связи (рис. 4б). В данном случае связи – это три настраиваемых параметра, которые являются координатами центров опорных векторов. В нашем случае передаточная функция может быть практически любой, в частности, линейной. В качестве опорного вектора в первом слое может быть выбрана любая из четырех точек. Пусть выбрана точка с координатами (0,0) и линейная

передаточная функция. Тогда на выходе нейрона первого слоя для 4-х входных векторов (координат точек) будем иметь три точки теперь уже в одномерном пространстве (на линии), с координатами (0) , $(\sqrt{2})$ и (1) . Обе точки второго класса на выходе слоя не различимы. Взяв в качестве опорного вектора для второго слоя RBF точку с координатой (1) и линейную передаточную функцию, а также порог $0,1$ (3), получаем модульную сеть, решающую поставленную задачу.

Архитектура «нового типа», показанная на рис. 4б, достаточно полно иллюстрирует потенциальные возможности предложенного подхода к моделированию нейронных сетей. Известные нейросетевые парадигмы не предоставляют возможности, например, использовать RBF-нейроны или слои последовательно. Это связано с проблемами построения алгоритмов обучения для такой сети. Однако при решении прикладных задач, исследуя данные при помощи известных типов сетей и алгоритмов обучения, можно определить, какой характер преобразования входного пространства признаков «упрощает» решение. Выделяя отдельные слои в обученных сетях и комбинируя их со слоями других типов, мы получаем мощный инструмент решения задач классификации, и шире – распознавания, произвольной сложности.

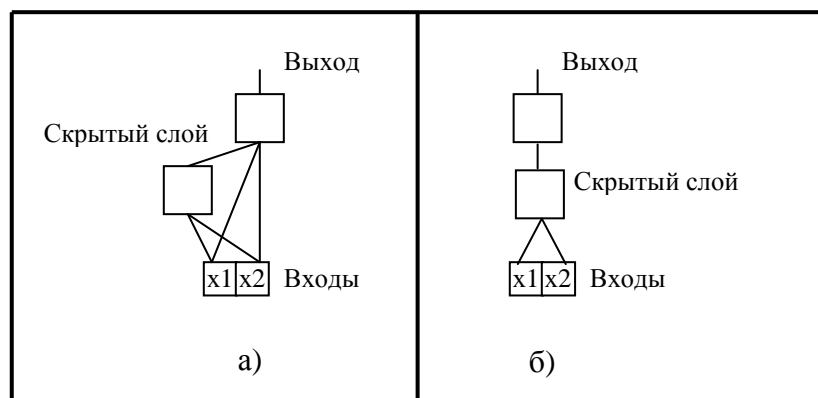


Рис. 4. Простейшие сети, решающие задачу «исключающее ИЛИ»

5. Выводы

В данной работе проведен анализ различных парадигм нейронных сетей, широко используемых на практике. Показано, что основным элементом архитектур этих сетей является слой нейронов. Установлено, что если рассматривать слои как независимые элементы архитектуры, то количество различных слоев, которые требуются для моделирования рассмотренных парадигм нейронных сетей, меньше, чем количество самих парадигм.

На основании проведенного анализа предложено рассматривать слои нейронов в качестве модулей или элементов архитектуры модульных сетей. Сформулированы требования к свойствам слоев и функционированию модульной сети в целом. На модельном примере показано, что представление нейронных сетей в виде модулей-слоев позволяет не только моделировать известные парадигмы, но и строить архитектуры не изучавшиеся ранее.

Отметим, что не любая комбинация рассматриваемых модулей дает работоспособную нейронную сеть. Использование модулей при построении нейронных сетей различных архитектур требует введения некоторых правил и ограничений. Правила применения модулей при построении нейросетевых архитектур будут обсуждаться в последующих публикациях.

Отдельный интерес представляют проблемы обучения модульных сетей с произвольным составом модулей. В произвольной сети часть модулей может обучаться с учителем, часть без, а некоторые модули могут вообще не требовать обучения. К последнему типу можно отнести метод главных компонент, позволяющий уменьшить размерность исходного пространства признаков. Большинство разработчиков обучают модули отдельно и получают общее решение задачи как комбинацию решений отдельных сетей. Однако большой интерес представляет кооперативное обучение, когда результаты на выходах одних модулей интерпретируются как сигналы для обучения других. Продолжение теоретических исследований в области модульных нейронных сетей мы видим, в частности, в исследовании методов кооперативного обучения.

СПИСОК ЛИТЕРАТУРЫ

1. Haykin S. *Neural Networks: A Comprehensive Foundation*. – 2-nd ed. – Englewood Cliffs, NJ. – Prentice-Hall, 1999. – P. 842.
2. Vapnik V. *Statistical Learning Theory*. – John Wiley & Sons. – Inc., NY, 1998. – P. 736.
3. Hopfield J.J. *Neural networks and physical systems with emergent collective computational abilities* // *Proc. Nat. Acad. Sci. USA*. – 1982. – N 79. – P. 2554–2558.
4. Кусскуль Э.М. *Ассоциативные нейроподобные структуры*. – Киев: Наукова думка, 1992. – 143 с.
5. Berthold M.R., Fischer I. *Formalizing neural networks using graph transformations* // *Proc. of IEEE International Conference on Neural Networks, (ICNN'97)*. – Houston. – 1997. – Vol. 1. – P. 275–280.
6. Kothari R., Ensley D. *Decision boundary and generalization performance of feed-forward networks with Gaussian lateral connections* // *Proceedings of SPIE*. – 1998. – Vol. 3390. – P. 314–321.
7. Hornik K., Stinchcombe M., White H. *Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks* // *Neural Networks*. – 1990. – Vol. 3. – P. 551–560.
8. Park J., Sandberg I.W. *Universal approximation using radial-basis function networks* // *Neural Computation*. – 1991. – Vol. 3. – P. 245–257.
9. Frasconi P., Gori M., Soda G. *Local feedback multilayered networks* // *Neural Computation*. – 1992. – Vol. 4(1). – P. 120–130.
10. Narendra K.S., Parthasarathy K. *Identification and control of dynamical systems using neural networks* // *IEEE Transactions on Neural Networks*. – 1990. – Vol. 1(1). – P. 4–27.
11. Elman J.L. *Distributed representations, simple recurrent networks, and grammatical structure* // *Machine Learning*. – 1991. – Vol. 7(2/3). – P. 195–226.
12. Williams R.J., Zipser D. *A learning algorithm for continually running fully recurrent neural networks* // *Neural Computation*. – 1989. – Vol. 1(2). – P. 270–280.
13. Резник А.М., Городничий Д.О., Сычев А.С. *Регулирование обратной связи в нейронных сетях с проекционным алгоритмом обучения* // *Кибернетика и системный анализ*. – 1996. – № 6. – С.153–162.
14. Кусскуль М.Э., Галинская А.А. *О свойствах циклов в модульных нейронных сетях* // *Математичні машини і системи*. – 2005. – № 2. – С. 54–62.
15. Резник А.М., Кусскуль М.Э., Сычев А.С., Садовая Е.Г., Калина Е.А. *Система проектирования модульных нейронных сетей САПР МНН* // *Математичні машини і системи*. – 2002. – № 3. – С. 28–37.