

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Российский химико-технологический университет имени Д. И. Менделеева

**С. П. Дударов, П. Л. Папаев**

**Теоретические основы и практическое  
применение искусственных нейронных сетей**

Утверждено Редакционным  
советом университета  
в качестве учебного пособия

Москва

2014

УДК 004.8  
ББК 32.813  
Д81

Рецензенты:

Доктор технических наук, профессор, декан  
Санкт-Петербургского государственного технологического института  
(технического университета)

*А. А. Мусаев*

Кандидат технических наук, доцент  
Российского химико-технологического университета  
имени Д. И. Менделеева

*П. Г. Михайлова*

**Дударов С. П.**

Д81 Теоретические основы и практическое применение искусственных нейронных сетей: учеб. пособие/ С. П. Дударов, П. Л. Папаев. – М. : РХТУ им. Д. И. Менделеева, 2014. – 104 с.  
ISBN 978-5-7237-1222-5

Изложены основная терминология, классификация, математические основы и примеры практического применения различных архитектур искусственных нейронных сетей. Описываются особенности решения задач аппроксимации и кластеризации данных, распознавания и классификации образов с помощью нейросетевого подхода.

Рекомендуется студентам бакалавриата и магистратуры, обучающимся по направлениям подготовки «Информационные системы и технологии», «Информатика и вычислительная техника» и другим направлениям подготовки, использующим для решения своих задач методы вычислительной математики, кибернетики и нейроинформатики.

УДК 004.8  
ББК 32.813

ISBN 978-5-7237-1222-5

© Российский химико-технологический  
университет имени Д. И. Менделеева, 2014  
© Дударов С. П., Папаев П. Л., 2014

## Перечень принятых обозначений

- $a$  – параметр насыщения нормализующей сигмоидной функции
- $b_0, b_1$  – коэффициенты линейного фрагмента кусочной активационной функции
- $c$  – центр (координата оси симметрии) радиально-симметричной функции
- $D_d$  – показатель репрезентативности диапазона  $d$  из интервала изменения переменной
- $d$  – индекс диапазона из интервала изменения переменной
- $d_p$  – уровень допустимого отклонения показателя репрезентативности диапазона от равномерного распределения
- $dx$  – расстояние между входными векторами двух примеров выборки
- $dx_{\text{пр}}$  – предельное значение параметра сходства входных векторов примеров выборки
- $dy$  – расстояние между выходными векторами двух примеров выборки
- $dy_{\text{пр}}$  – предельное значение параметра сходства выходных векторов примеров выборки
- $E$  – ошибка расчета по нейронной сети для отдельного выхода в исходных единицах измерения
- $\tilde{E}$  – нормализованная ошибка расчета по нейронной сети для отдельного выхода
- $\tilde{E}_{\text{об}}$  – общая нормализованная ошибка расчета по нейронной сети
- $E_{\text{отн}}$  – относительная ошибка расчета по нейронной сети для отдельного выхода в процентах
- $f(s)$  – активационная функция любого вида
- $h(x)$  – радиально-симметричная функция
- $i$  – индекс входной переменной или выхода нейрона предыдущего слоя
- $j$  – индекс выходной переменной или нейрона текущего слоя
- $K$  – количество выходных переменных нейронной сети
- $k$  – индекс примера выборки
- $L$  – количество скрытых слоев
- $l$  – индекс скрытого слоя
- $M$  – количество входных переменных нейронной сети (нейрона)
- $m$  – количество нейронов в скрытом слое
- $N$  – количество примеров выборки
- $N_C$  – количество классов образов, выделяемых нейронной сетью
- $N_{\text{пр}}$  – предельно допустимое количество примеров для формирования кластера
- $N_d$  – число примеров в диапазоне  $d$
- $N_w$  – количество весовых коэффициентов нейронной сети

$N_d$  – среднее количество примеров в диапазоне из интервала изменения переменной

$n_d$  – количество диапазонов в интервале изменения переменной

$R$  – расстояние от центра кластера до примера выборки

$R_{\max}$  – максимальное по кластерам значение меры сходства

$R_{\min}$  – расстояние от нейрона-победителя до примера обучающей выборки

$R_{\text{кр}}$  – критическое расстояние, при котором допускается отнесение обучающего примера к существующему кластеру

$r$  – радиус рассеяния входной переменной радиально симметричной функции

$s$  – состояние нейрона

$T$  – координата порога функции единичного скачка

$t$  – весовой (синаптический) коэффициент долговременной памяти

$v$  – коэффициент скорости обучения нейронной сети

$w$  – весовой (синаптический) коэффициент (для АРТ-1 – кратковременной памяти)

$x$  – входная переменная в исходных единицах измерения

$x_c$  – центр нормализуемого интервала изменения входной переменной

$x_{\min}$  – минимальное значение входной переменной в выборке

$x_{\max}$  – максимальное значение входной переменной в выборке

$x^*$  – зашумленная входная переменная

$\tilde{x}$  – нормализованная входная переменная

$y$  – выходная переменная в исходных единицах измерения

$y_c$  – центр нормализуемого интервала изменения выходной переменной

$y_{\min}$  – минимальное значение выходной переменной в выборке

$y_{\max}$  – максимальное значение выходной переменной в выборке

$\tilde{y}$  – нормализованная выходная переменная

$\tilde{y}^n$  – нормализованное выходное значение по примеру выборки

$\tilde{y}^p$  – рассчитанное по нейронной сети нормализованное выходное значение

$\alpha$  – параметр насыщения активационной функции

$\beta$  – параметр влияния близости нейрона (центра кластера) к обучающему примеру

$\Delta$  – погрешность (ошибка) расчета выходного значения

$\varepsilon$  – абсолютное значение веса ингибиторного синапса

$\lambda$  – константа влияния нового образа на кратковременную память в нейронных сетях АРТ-1

$\psi$  – ширина коридора экстраполирования в долях исходного интервала изменения переменной

## Введение

Методы искусственного интеллекта – это современное направление развития методов математического моделирования свойств объектов, динамических процессов и поведения систем. Зародившись в середине XX века, данное направление динамично развивается в настоящее время, и уже в рамках него можно выделить самостоятельные, практически независимые области знаний со своими методами и математическим аппаратом: нейроинформатика, нечеткая логика и теория нечетких множеств, эволюционное моделирование, многоагентные системы и другие.

В данном учебном пособии представлены теоретические и практические аспекты применения искусственных нейронных сетей, по праву считающихся одним из главных направлений развития методов искусственного интеллекта. Многообразие существующих архитектур нейронных сетей позволяет использовать методы нейроинформатики для решения практически любых классов задач. Чаще всего среди этих задач фигурируют аппроксимация данных, прогнозирование временных рядов, математическое моделирование свойств объектов, распознавание образов, классификация, кластеризация данных, управление.

Искусственные нейронные сети – достаточно сложный математический аппарат. Для их использования зачастую требуются большие объемы исходной информации и значительные вычислительные ресурсы. В этой связи для решения достаточно простых задач представляется целесообразным использовать другие известные и широко применяемые методы. Применение методов нейроинформатики представляет наибольший интерес в случаях работы с большими объемами исходной информации, для многомерных задач, а также в условиях неполноты, неопределенности и изменчивости исходной информации.

В пособии рассмотрены примеры использования всех описанных архитектур нейронных сетей на достаточно простых примерах, чтобы заострить внимание на математических и методических аспектах и поспособствовать лучшему пониманию алгоритмов их работы. Более сложные и масштабные задачи, как правило, не будут сильно отличаться в применении методов нейроинформатики от рассмотренных в пособии примеров.

## **1. История развития искусственных нейронных сетей**

Термин «нейронная сеть» появился в середине XX века. Первые работы, в которых были получены основные результаты в данном направлении, были проделаны Мак-Каллоком и Питтсом. В 1943 году ими была разработана компьютерная модель нейронной сети на основе математических алгоритмов и теории деятельности головного мозга. Они выдвинули предположение, что нейроны можно упрощенно рассматривать как устройства, оперирующие двоичными числами, и назвали эту модель «пороговой логикой». Подобно своему биологическому прототипу нейроны Мак-Каллока–Питтса были способны обучаться путем подстройки параметров, описывающих синаптическую проводимость. Исследователи предложили конструкцию сети из электронных нейронов и показали, что подобная сеть может выполнять практически любые вообразимые числовые или логические операции. Мак-Каллок и Питтс предположили, что такая сеть в состоянии также обучаться, распознавать образы, обобщать, т. е. обладает всеми чертами интеллекта.

Данная модель заложила основы двух различных подходов исследований нейронных сетей. Один подход был ориентирован собственно на изучение биологических процессов в головном мозге, другой – на применение нейронных сетей как метода искусственного интеллекта для решения различных прикладных задач.

В 1949 году канадский физиолог и психолог Хебб высказал идеи о характере соединения нейронов мозга и их взаимодействии. Он первым предположил, что обучение заключается, в первую очередь, в изменениях силы синаптических связей. Теория Хебба считается типичным случаем самообучения, при котором испытываемая система спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. В более поздних вариантах теория Хебба легла в основу описания явления долговременной потенциации.

В 1954 году в Массачусетском технологическом институте с использованием компьютеров Фарли и Кларк разработали имитацию сети Хебба. Также исследования нейронных сетей с помощью компьютерного модели-

рования были проведены Рочестером, Холландом, Хебитом и Дудой в 1956 году.

В 1957 году Розенблаттом были разработаны математическая и компьютерная модели восприятия информации мозгом на основе двухслойной обучающейся нейронной сети. При обучении данная сеть использовала арифметические действия сложения и вычитания. Розенблатт описал также схему не только основного перцептрона, но и схему логического сложения. В 1958 году им была предложена модель электронного устройства, которое должно было имитировать процессы человеческого мышления, а два года спустя была продемонстрирована первая действующая машина, которая могла научиться распознавать некоторые из букв, написанных на карточках, которые подносили к его «глазам», напоминающим кинокамеры.

Интерес к исследованию нейронных сетей угас после публикации работы по машинному обучению Минского и Пейперта в 1969 году. Ими были обнаружены основные вычислительные проблемы, возникающие при компьютерной реализации искусственных нейронных сетей. Первая проблема состояла в том, что однослойные нейронные сети не могли совершать «сложение по модулю 2», то есть реализовать функцию «Исключающее ИЛИ». Второй важной проблемой было то, что компьютеры не обладали достаточной вычислительной мощностью, чтобы эффективно обрабатывать огромный объем вычислений, необходимый для больших нейронных сетей.

Исследования нейронных сетей замедлились до того времени, когда компьютеры достигли больших вычислительных мощностей. Одним из важных шагов, стимулировавших дальнейшие исследования, стала разработка в 1975 году Вербосом метода обратного распространения ошибки, который позволил эффективно решать задачу обучения многослойных сетей и решить проблему со «сложением по модулю 2».

В 1975 году Фукусимой был разработан когнитрон, который стал одной из первых многослойных нейронных сетей. Фактическая структура сети и методы, используемые в когнитроне для настройки относительных весов связей, варьировались от одной стратегии к другой. Каждая из стратегий имела свои преимущества и недостатки. Сети могли распространять информацию только в одном направлении или перебрасывать информацию

из одного конца в другой, пока не активировались все узлы и сеть не приходила в конечное состояние. Достичь двусторонней передачи информации между нейронами удалось лишь в сети Хопфилда (1982), и специализация этих узлов для конкретных целей была введена в первых гибридных сетях.

Алгоритм параллельной распределенной обработки данных в середине 1980 годов стал популярен под названием коннективизма. В 1986 году в работе Руммельхарта и Мак-Клелланда коннективизм был использован для компьютерного моделирования нейронных процессов.

Несмотря на большой энтузиазм, вызванный в научном сообществе разработкой метода обратного распространения ошибки, это также породило многочисленные споры о том, может ли такое обучение быть на самом деле реализовано в головном мозге. Отчасти это связывали с тем, что механизм обратного прохождения сигнала не был очевидным в то время, так как не было явного источника обучающего и целевого сигналов. Тем не менее, в 2006 году было предложено несколько неконтролируемых процедур обучения нейронных сетей с одним или несколькими слоями с использованием так называемых алгоритмов глубокого обучения. Эти алгоритмы могут быть использованы для изучения промежуточных представлений как с выходным сигналом, так и без него, чтобы понять основные особенности распределения сенсорных сигналов, поступающих на каждый слой нейронной сети.

Как и во многих других случаях, задачи высокой сложности требуют применения не одного, а нескольких методов решения или их синтеза. Не исключение и искусственные нейронные сети. С самого начала нынешнего столетия в работах различных исследователей активно описываются нейро-нечеткие сети, ячеечно-нейросетевые модели. Также нейронные сети используются, например, для настройки параметров нечетких систем управления. В общем, нет никаких сомнений и в дальнейшей интеграции методов искусственного интеллекта между собой и с другими методами решения задач.



## 2. Основные положения и классификация искусственных нейронных сетей

**Искусственные нейронные сети** – это математический аппарат, основанный на аналогиях с функционированием центральной нервной системы и предназначенный для решения широкого круга задач, таких, как распознавание образов, классификация, кластеризация и аппроксимация данных, прогнозирование временных рядов, адаптивное управление и многих других. Круг решаемых задач может быть различен для разных архитектур. **Архитектура нейронной сети** – совокупность, объединяющая принципы организации и функционирования нейронной сети и отдельных искусственных нейронов, алгоритмы и расчетные соотношения для ее обучения.

Теория искусственных нейронных сетей – одно из основных направлений методов искусственного интеллекта, активно развивающихся со второй половины XX века. Направление, занимающееся изучением теоретических и прикладных аспектов искусственных нейронных сетей с использованием методов компьютерного моделирования, получило название **нейроинформатики**.

Центральная нервная система человека и животных состоит из сотен миллиардов соединенных между собой нейронов, по которым передаются сигналы. Структурно биологический нейрон состоит из клеточного тела с ядром, множества входных каналов (дендритов), проводящих нервные импульсы к телу, и одного длинного отростка (аксона), служащего источником выходного сигнала нейрона. Фильтрация и обработка входных сигналов осуществляется за счет синапсов – участка пространства, разделяющего мембраны нервных окончаний контактирующих клеток. В процессе прохождения через синапс могут изменяться и, как правило, меняются амплитуда и частота проходящего сигнала. Механизм передачи и преобразования такого сигнала может быть химическим, электрическим или смешанным, однако наиболее часто встречаются все же синапсы первого типа.

**Искусственный нейрон** – это элементарная структурная единица искусственной нейронной сети, выполняющая функции по обработке входных сигналов  $x_i$ , поступающих с других нейронов, и представлению

результата в форме выходного значения. В общем случае, входные сигналы подвергаются в теле нейрона преобразованию, зачастую нелинейному, с участием **весовых (синаптических) коэффициентов**  $w_i$ , характеризующих значимость каждой связи, и функции активации. При необходимости для более качественного обучения в некоторых нейросетевых архитектурах вектор синаптических коэффициентов дополняется **коэффициентом смещения**  $w_0$  – весом фиктивного единичного входа. Структура искусственного нейрона схематично показана на рис. 2.1.

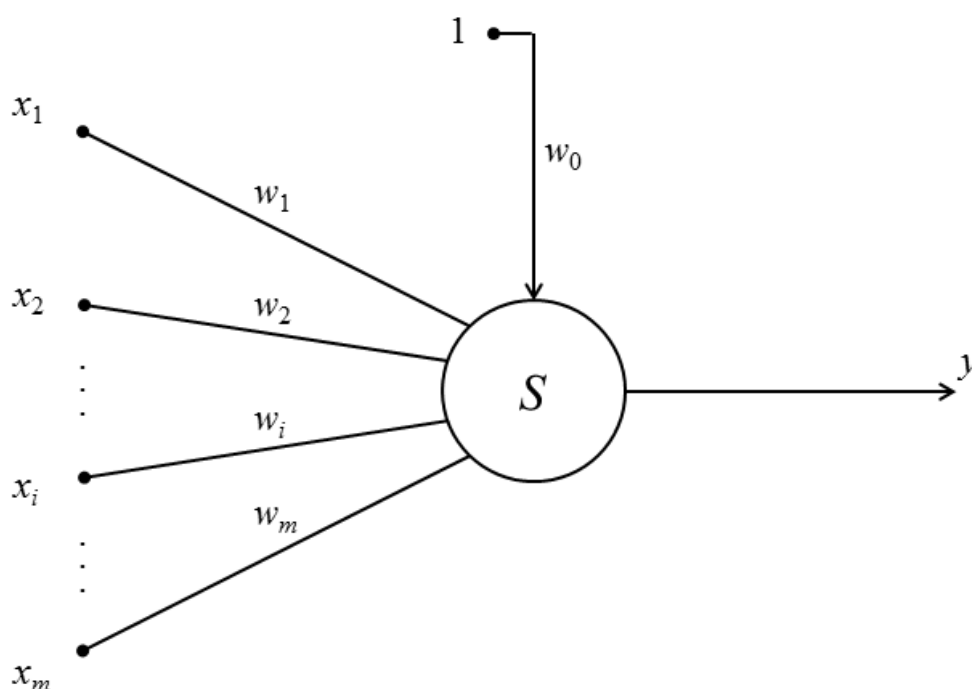


Рис. 2.1. Структура искусственного нейрона

Множество определенным образом связанных между собой искусственных нейронов формируют искусственную нейронную сеть. Нейроны могут быть сгруппированы в слои. Количество слоев, нейронов в каждом слое, а также количество внешних входных и выходных переменных определяют **структуру нейронной сети**.

Для практического использования нейронной сети ее необходимо предварительно обучить. **Обучение нейронной сети** представляет собой процедуру точного или итерационного расчета всех ее весовых коэффициентов, соединяющих внешние входы с нейронами и нейроны между собой. Алгоритм и расчетные соотношения данной процедуры зависят от архи-

тектуры и структуры нейронной сети.

Для обучения нейронной сети используется специально полученная обучающая выборка, представляющая собой множество примеров, относящихся к одному объекту при определенных условиях его исследования.

Под **эпохой обучения** понимается цикл однократного предъявления (как правило, в случайном порядке) всех примеров обучающей выборки для коррекции весовых коэффициентов при одинаковых настройках самой сети и алгоритма ее обучения.

Поскольку происхождение входных сигналов искусственных нейронов может быть различно: это могут быть как выходы других нейронов, так и входные сигналы из внешней среды (входы нейронной сети), эти сигналы целесообразно использовать в нормализованной форме, то есть изменяющимися в пределах  $[0, 1]$  или  $[-1, 1]$ . Выбор пределов нормализации зависит, в первую очередь, от планирующейся для использования архитектуры нейронной сети, а во вторую, – от предпочтений самого исследователя, так как отдельные архитектуры сетей позволяют работать только со строго определенными значениями. В зависимости от природы используемых переменных и степени влияния их различных значений могут быть использованы методы линейной и нелинейной нормализации.

**Состояние нейрона**  $s$  определяется как взвешенная с помощью синаптических коэффициентов сумма его входных значений:

$$s = w_0 + \sum_{i=1}^m w_i x_i, \quad (2.1)$$

где  $m$  – количество входных сигналов нейрона.

**Функция активации** преобразует состояние нейрона в значение в форме и масштабе выходного сигнала:

$$y = f(s). \quad (2.2)$$

В большинстве случаев для активации нейронов используются два типа функций: гистерезиса (насыщения) и импульсные. Выбор активационной функции выходных нейронов должен определяться, в первую очередь, характером описываемых ими зависимостей. Для скрытых нейронов вид и настройки активационных функций в большинстве случаев могут совпадать.

Группа искусственных нейронов, соединенных связями одного уров-

ня с нейронами других групп, входами нейронной сети или друг с другом, образует отдельный **слой нейронов**.

Слой искусственных нейронов, выходы которого являются выходами всей нейронной сети, называется **выходным слоем**. Таким образом, количество нейронов выходного слоя должно совпадать с количеством выходов всей нейронной сети. Все остальные слои искусственных нейронов, кроме выходного, называются **скрытыми слоями**.

Часто в литературе множество входных переменных называется **входным слоем**, или **слоем входных значений**. Однако ошибочно считать его слоем нейронов.

Искусственная нейронная сеть с единственным выходным слоем называется **однослойной сетью**. Искусственная нейронная сеть с одним скрытым слоем называется **двухслойной сетью** и т. д.

Пример схемы двухслойной искусственной нейронной сети с  $M$  входами,  $m_1$  нейронами единственного скрытого слоя и  $K$  выходами представлен на рис. 2.2.

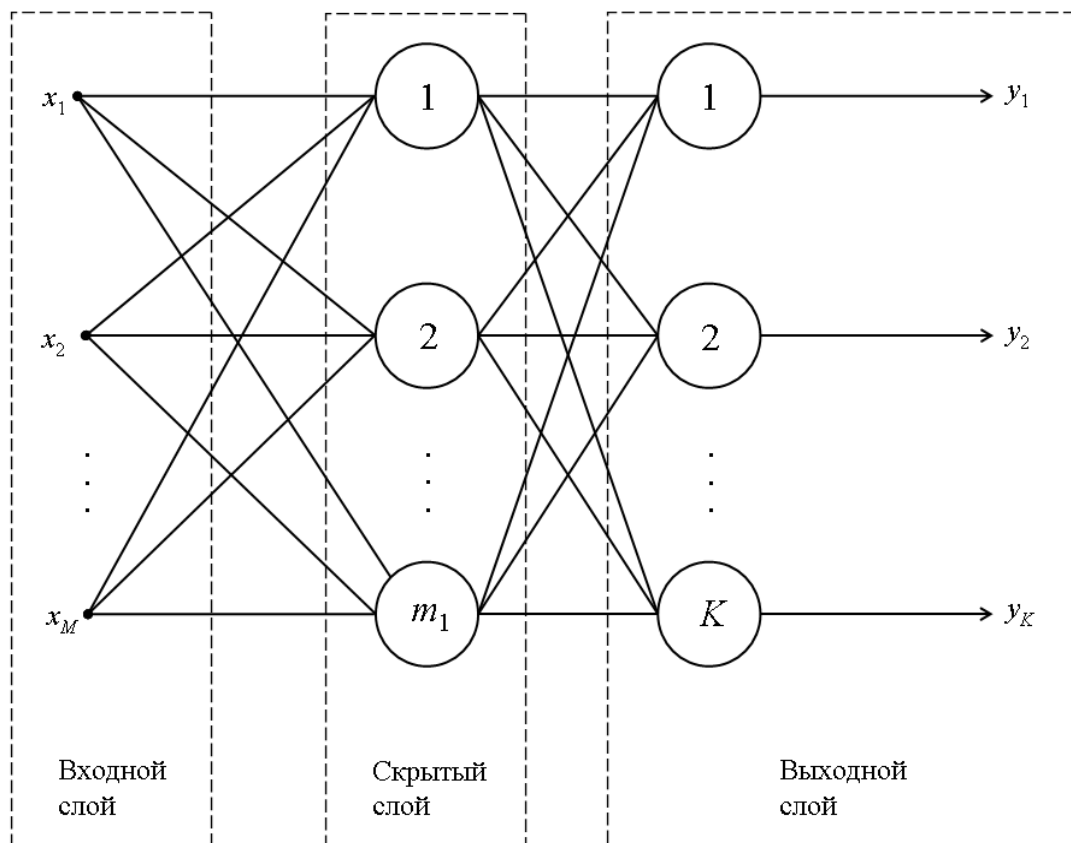


Рис. 2.2. Схема двухслойной искусственной нейронной сети

Существует несколько способов классификации искусственных нейронных сетей. Основной способ классификации – **по используемой архитектуре**, и зачастую под классом нейронной сети в узком смысле понимается именно архитектура. В соответствии с данным классифицирующим признаком архитектуры делятся:

- на однослойные и многослойные перцептроны;
- сети на основе радиально-симметричных функций;
- сети Кохонена;
- сети адаптивного резонанса;
- сети на основе автоассоциативной и гетероассоциативной памяти;
- сети Хэмминга и другие.

Подробная теория и примеры работы со многими архитектурами искусственных нейронных сетей приводятся в следующих главах пособия.

Нейронные сети можно классифицировать **по принципу обучения**. Наиболее распространены следующие разновидности обучения:

- с учителем (когда требуется получить нейросетевую модель, обобщающую примеры, в каждом из которых определенному набору значений входных переменных должен соответствовать набор значений выходных переменных);

- без учителя (когда ведется поиск скрытых закономерностей в массиве обучающих примеров без использования значений выходных переменных).

Еще один классифицирующий признак – **тип алгоритма обучения**. Здесь также имеется две основные группы:

- итерационные алгоритмы (когда многократно, в цикле повторяется определенная последовательность действий, предусмотренных алгоритмом, до стабилизации весовых коэффициентов);

- точные алгоритмы (когда весовые коэффициенты рассчитываются по обучающему набору с использованием операций линейной алгебры).

В свою очередь, итерационные алгоритмы делятся:

- на детерминированные (когда задана строго определенная последовательность действий по минимизации ошибки обучения);

- стохастические (в ходе работы которых поиск лучших значений весовых коэффициентов подчиняется случайным процессам).

**По области допустимых значений выходного сигнала** нейронные сети классифицируются:

- на бинарные (когда выходная переменная может принимать одно из двух возможных значений);
- аналоговые (когда возможно бесконечное множество значений выходной переменной, принадлежащее пределам ее нормализации).

Область допустимых значений выходного сигнала определяется выбранной активационной функцией соответствующего выходного нейрона. В общем случае возможно одновременное использование бинарных значений для одних и аналоговых значений для других выходов одной и той же нейронной сети.

**По структуре связей** среди нейронных сетей можно выделить:

- сети прямого распространения (когда сигналы распространяются непосредственно от входов к выходам);
- сети с обратными связями (когда выходной сигнал нейрона может возвращаться на его же вход, входы других нейронов того же слоя или входы нейронов предшествующих слоев).

Если выходные сигналы нейрона распространяются по обратным связям ко всем остальным нейронам без исключения, такая сеть называется **полносвязной**.

В целом, сети прямого распространения проще в реализации и гарантируют получение ответа после прохождения сигналов по слоям. В свою очередь, сети с обратными связями более компактны и быстрее обучаются.

Нейронные сети могут быть разделены на классы **по способу изменения состояний нейронов**:

- на асинхронные (у которых в определенный момент времени только один нейрон меняет свое состояние);
- синхронные (у которых в определенный момент времени сразу несколько нейронов (как правило, целый слой) меняют свои состояния).

Следует отметить, что любая архитектура нейронной сети может быть отнесена также к одному из классов приведенной выше классификации.

### 3. Способы нормализации переменных

Необходимость нормализации выборок данных обусловлена самой природой используемых переменных нейросетевых моделей. Будучи разными по физическому смыслу, они зачастую могут сильно различаться между собой по абсолютным величинам. Так, например, выборка может содержать и концентрацию, измеряемую в десятых или сотых долях процентов, и давление в сотнях тысяч паскаль. Нормализация данных позволяет привести все используемые числовые значения переменных к одинаковой области их изменения, благодаря чему появляется возможность свести их вместе в одной нейросетевой модели.

Чтобы выполнить нормализацию данных, нужно точно знать пределы изменения значений соответствующих переменных (минимальное и максимальное теоретически возможные значения). Тогда им и будут соответствовать границы интервала нормализации. Когда точно установить пределы изменения переменных невозможно, они задаются с учетом минимальных и максимальных значений в имеющейся выборке данных.

Наиболее распространенный способ нормализации входных и выходных переменных – **линейная нормализация**.

Примем следующие обозначения:

–  $x_{ik}, y_{jk}$  –  $i$ -е входное и  $j$ -е выходное значения  $k$ -го примера исходной выборки в традиционных единицах измерения, принятых в решаемой задаче;

–  $\tilde{x}_{ik}, \tilde{y}_{jk}$  – соответствующие им нормализованные входное и выходное значения;

–  $N$  – количество примеров обучающей выборки.

Тогда переход от традиционных единиц измерения к нормализованным и обратно с использованием метода линейной нормализации осуществляется с использованием следующих расчетных соотношений:

– при нормализации и денормализации в пределах  $[0, 1]$ :

$$\tilde{x}_{ik} = \frac{x_{ik} - x_{\min i}}{x_{\max i} - x_{\min i}}; \quad (3.1)$$

$$y_{jk} = y_{\min j} + \tilde{y}_{jk} (y_{\max j} - y_{\min j}); \quad (3.2)$$

– при нормализации и денормализации в пределах  $[-1, 1]$ :

$$\tilde{x}_{ik} = 2 \cdot \frac{x_{ik} - x_{\min i}}{x_{\max i} - x_{\min i}} - 1;$$

$$y_{jk} = y_{\min j} + (\tilde{y}_{jk} + 1)(y_{\max j} - y_{\min j})/2,$$

где

$$x_{\min i} = \min_{k=1, N}(x_{ik}); x_{\max i} = \max_{k=1, N}(x_{ik});$$

$$y_{\min j} = \min_{k=1, N}(y_{jk}); y_{\max j} = \max_{k=1, N}(y_{jk}).$$

Если обучающая выборка не содержит примеров с потенциально возможными меньшими или большими выходными значениями, можно задаться шириной коридора экстраполяции  $\psi$  для левой, правой или обеих границ в долях от длины всего первоначального интервала изменения переменной, обычно не более 10 % от нее. В этом случае происходит переход от фактических границ из обучающей выборки к гипотетическим:

$$y_{\min j} = (\psi + 1) \min_{k=1, N}(y_{jk}) - \psi \max_{k=1, N}(y_{jk}); y_{\max j} = (\psi + 1) \max_{k=1, N}(y_{jk}) - \psi \min_{k=1, N}(y_{jk}).$$

Один из способов **нелинейной нормализации** – с использованием сигмоидной логистической функции или гиперболического тангенса. Переход от традиционных единиц измерения к нормализованным и обратно в данном случае осуществляется следующим образом:

– при нормализации и денормализации в пределах  $[0, 1]$ :

$$\tilde{x}_{ik} = \frac{1}{e^{-a(x_{ik} - x_{ci})} + 1};$$

$$y_{jk} = y_{cj} - \frac{1}{a} \ln \left( \frac{1}{\tilde{y}_{jk}} - 1 \right),$$

где  $x_{ci}$ ,  $y_{cj}$  – центры нормализуемых интервалов изменения входной и выходной переменных:

$$x_{ci} = (x_{\min i} + x_{\max i})/2; y_{cj} = (y_{\min j} + y_{\max j})/2;$$

– при нормализации и денормализации в пределах  $[-1, 1]$ :

$$\tilde{x}_{ik} = \frac{e^{a(x_{ik} - x_{ci})} - 1}{e^{a(x_{ik} - x_{ci})} + 1};$$

$$y_{jk} = y_{cj} - \frac{1}{a} \ln \left( \frac{1 - \tilde{y}_{jk}}{1 + \tilde{y}_{jk}} \right).$$



Параметр  $a$  влияет на степень нелинейности изменения переменной в нормализуемом интервале. Кроме того, при использовании значений  $a < 0,5$  нет необходимости дополнительно задаваться шириной коридора экстраполяции.

Рассмотрим в сравнении методы линейной и нелинейной нормализации. На рис. 3.1 приведены графики нормализации входной переменной для пределов  $[-1; 1]$ . Для нелинейной нормализации с использованием функции гиперболического тангенса принято значение параметра  $a = 1,0$ . Следует отметить, что совпадение нормализованного значения в обоих случаях имеет место лишь в точке, соответствующей центру нормализуемого интервала.

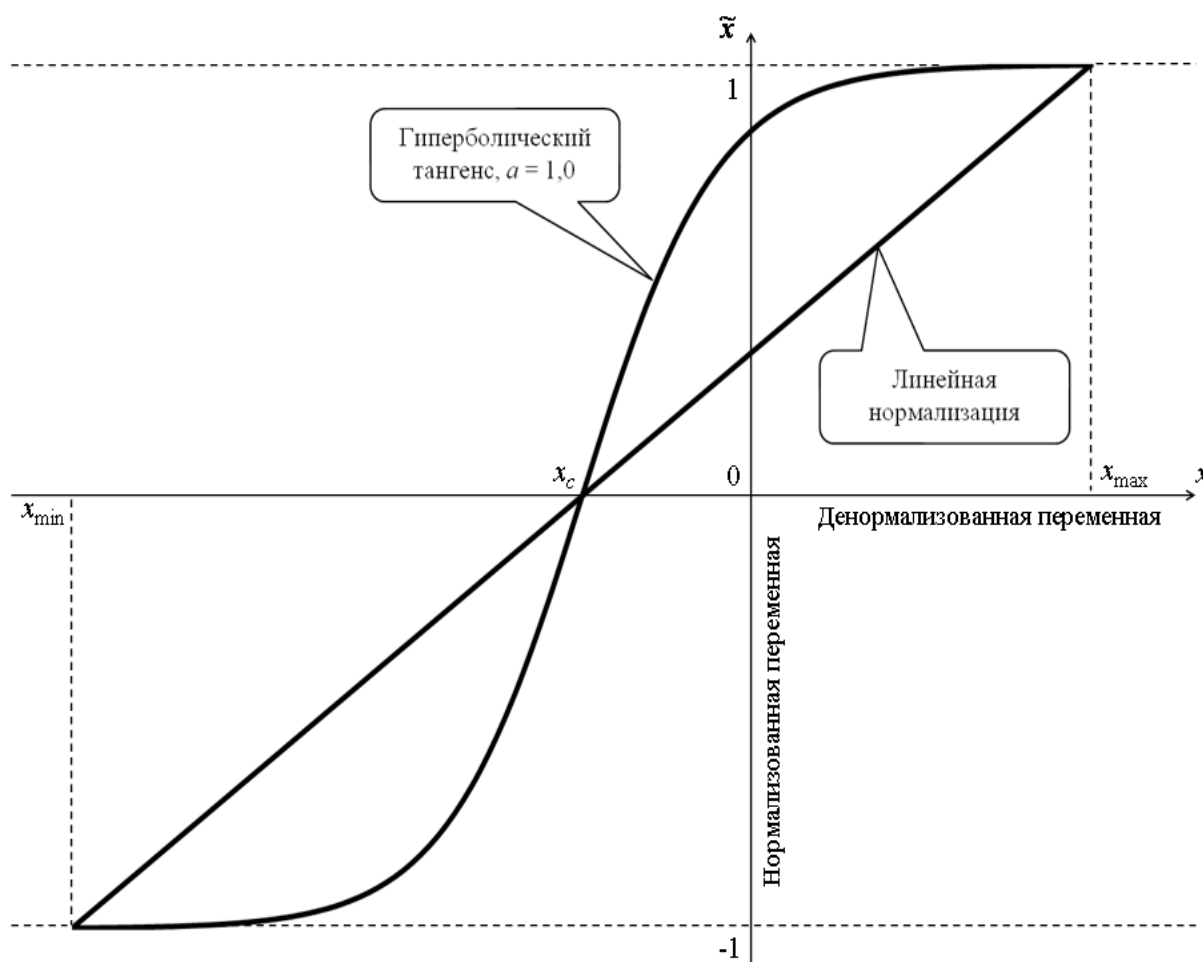


Рис. 3.1. Сравнение линейной и нелинейной функций нормализации

На рис. 3.2 показаны случаи нелинейной нормализации в пределах  $[0; 1]$  с использованием функции гиперболического тангенса с параметра-

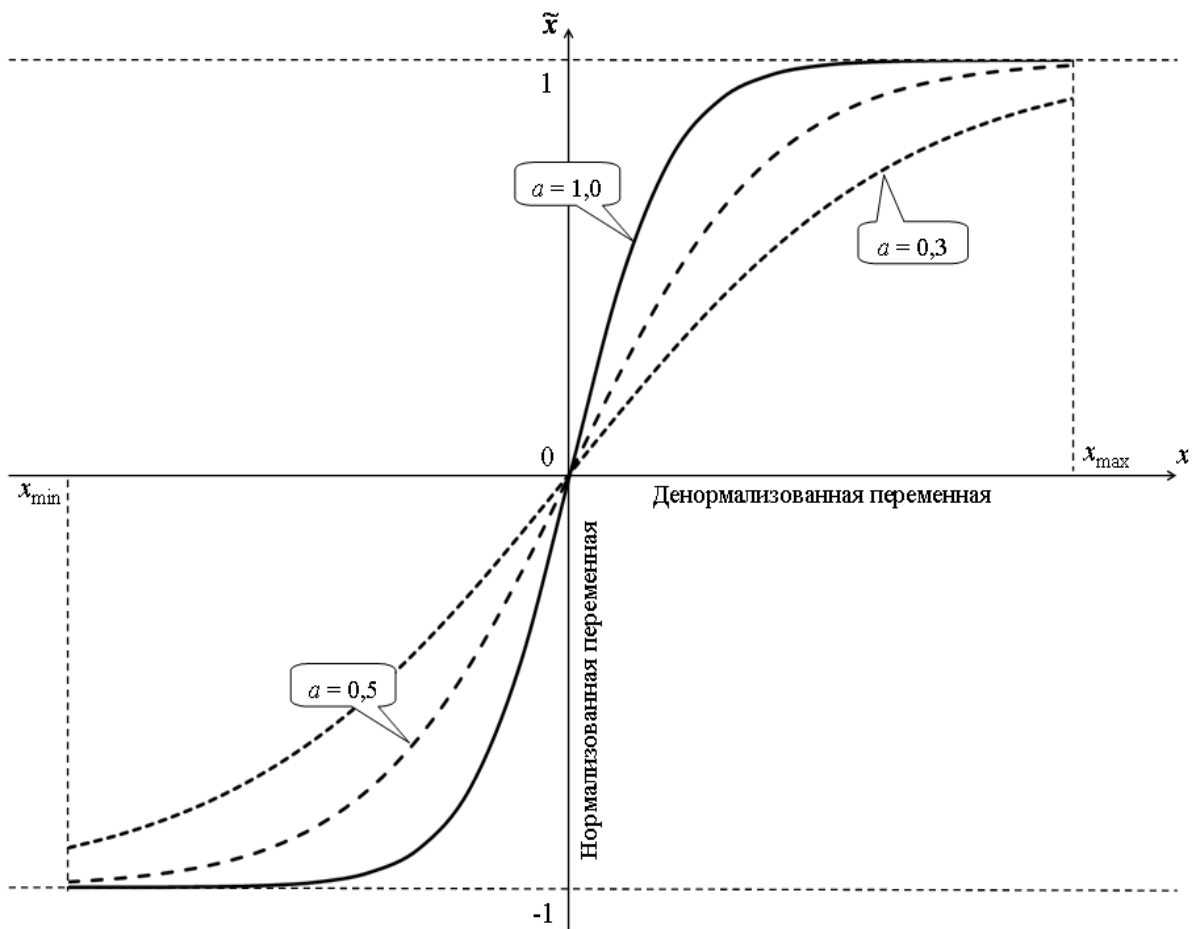


Рис. 3.2. Влияние параметра на график функции нелинейной нормализации

ми  $a$ , равными, соответственно, 0,3, 0,5, 1,0. Очевидно, что чем меньше значение параметра  $a$ , тем более полого выглядит нормализованная зависимость и больше ширина коридора экстраполяции.

#### 4. Активационные функции и их свойства

К активационным функциям типа **гистерезиса** относятся:

– понижающегося единичного скачка (рис. 4.1, а):

$$f(s) = \begin{cases} 1, & s \leq T; \\ 0, & s > T; \end{cases}$$

– повышающегося единичного скачка (рис. 4.1, б):

$$f(s) = \begin{cases} 0, & s \leq T; \\ 1, & s > T; \end{cases}$$

– линейная пороговая (рис. 4.1, в, г):

$$f(s) = \begin{cases} 1, & s \leq T_1; \\ b_1 s + b_0, & T_1 < s < T_2, \quad b_1 < 0; \\ 0, & s \geq T_2; \end{cases} \quad \text{или} \quad f(s) = \begin{cases} 0, & s \leq T_1; \\ b_1 s + b_0, & T_1 < s < T_2, \quad b_1 > 0; \\ 1, & s \geq T_2; \end{cases}$$

– сигмоидная логистическая (рис. 4.1, д):

$$f_1(s) = \frac{1}{e^{-\alpha s} + 1}; \quad (4.1)$$

– сигмоидная – гиперболический тангенс (рис. 4.1, е):

$$f_2(s) = 2f_1(s) - 1 = \frac{e^{\alpha s} - 1}{e^{\alpha s} + 1}.$$

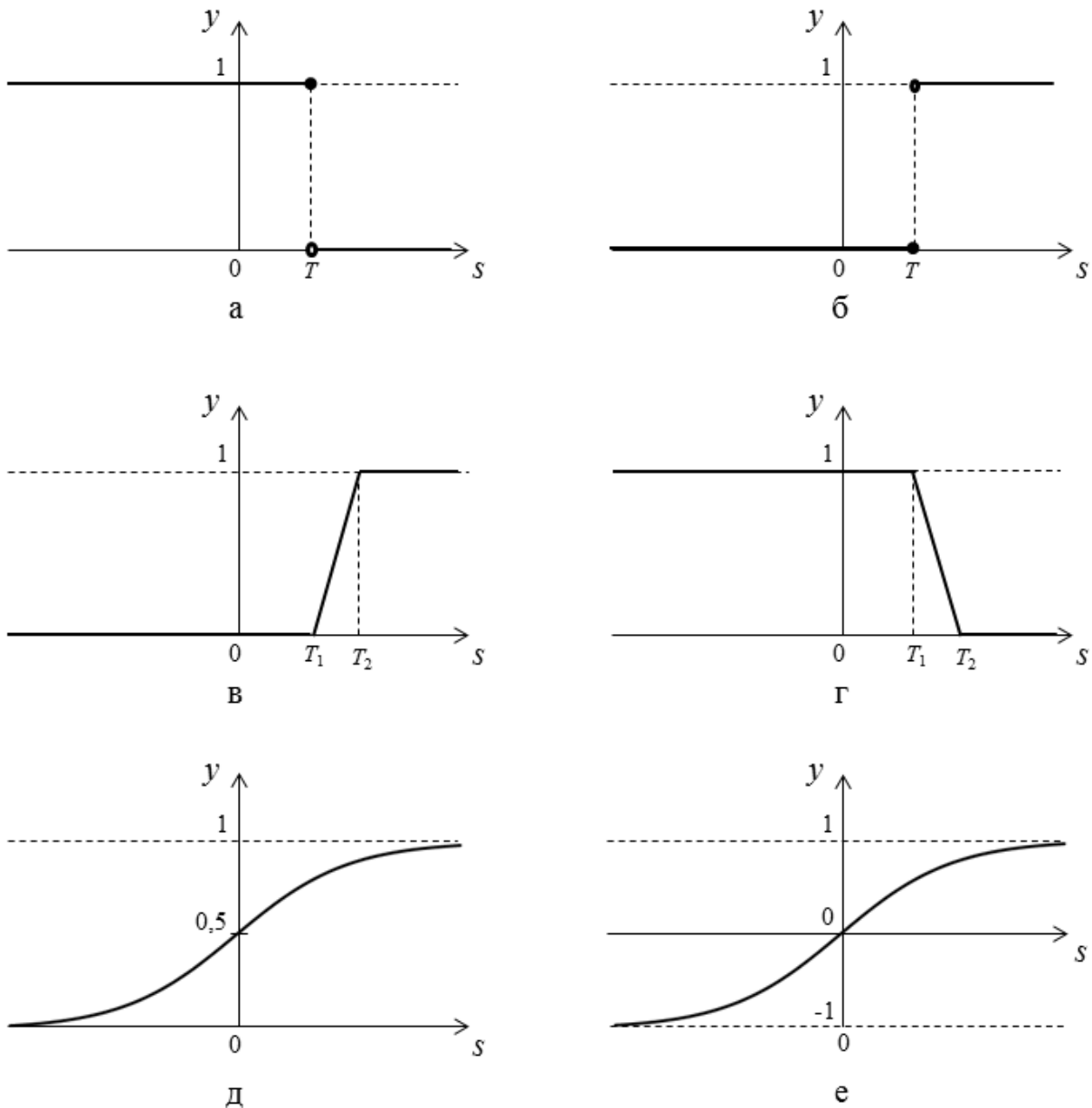


Рис. 4.1. Примеры функций активации типа гистерезиса

Варьируя параметр насыщения  $\alpha$  логистической функции или гиперболического тангенса, можно добиться изменения вида их графика. В предельных значениях, близких по абсолютному значению к 0 или к бесконечности, графики обеих функций приближаются к форме горизонтальной прямой на уровне середины интервала изменения нормированной переменной или мгновенного скачка (таблица).

### Свойства сигмоидных функций

Параметр $\alpha$	Состояние нейрона $s$	Значение функции	Вид графика
$\rightarrow -\infty$	–	$\approx 1-$	Мгновенный понижающийся скачок при $s = 0$
	+	$\approx 0+$	
$\rightarrow +\infty$	–	$\approx 0+$	Мгновенный повышающийся скачок при $s = 0$
	+	$\approx 1-$	
$\rightarrow 0-$	–	$\approx 0,5+$	Медленно убывающая зависимость
	+	$\approx 0,5-$	
$\rightarrow 0+$	–	$\approx 0,5-$	Медленно возрастающая зависимость
	+	$\approx 0,5+$	

Еще одно очень важное свойство сигмоидных функций – возможность точного расчета значений их производных только на основе численных значений самих функций. Данное свойство выгодно используется во многих алгоритмах обучения нейронных сетей.

В соответствии с описанным свойством для сигмоидной логистической функции производную в точке  $s$  можно рассчитать по соотношению:

$$f_1'(s) = \alpha \cdot f_1(s) \cdot (1 - f_1(s)), \quad (4.2)$$

а для функции гиперболического тангенса:

$$f_2'(s) = \frac{\alpha}{2} \cdot (1 - f_2(s)) \cdot (1 + f_2(s)) = \frac{\alpha}{2} \cdot (1 - f_2^2(s))$$

В качестве функции **импульсного** типа обычно используется функция Гаусса (рис. 4.2, а):

$$f_3(s) = e^{-\alpha(s-T)^2}, \quad (4.3)$$

где  $\alpha > 0$ , причем, чем меньше значение  $\alpha$ , тем более широкий импульс на графике функции;  $T$  – координата вертикальной оси симметрии функции Гаусса. В пределе при  $\alpha = 0$  функция вырождается в горизонтальную пря-

мую на уровне 1.

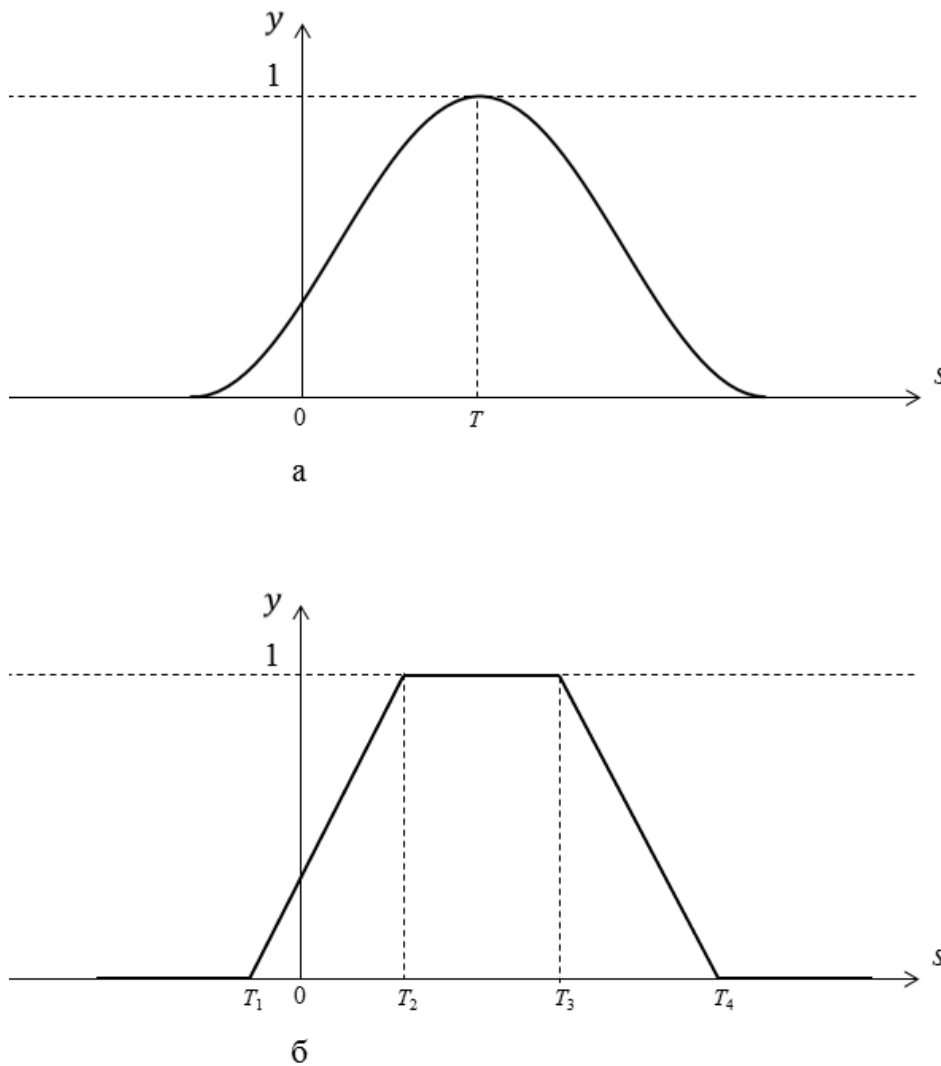


Рис. 4.2. Примеры функций активации импульсного типа

Производная данной функции в точке  $s$  также легко может быть рассчитана по ее выходному значению:

$$f_3'(s) = -2\alpha \cdot (s - T) \cdot f_3(s).$$

Кроме того, в качестве функции импульсного типа может быть использована кусочно-линейная П-зависимость (рис. 4.2, б):

$$f(s) = \begin{cases} 0, (s \leq T_1) \cup (s \geq T_4); \\ b_{11}s + b_{10}, T_1 < s < T_2, b_{11} > 0; \\ 1, T_2 < s < T_3 \\ b_{21}s + b_{20}, T_3 < s < T_4, b_{21} < 0. \end{cases}$$

## 5. Методы обработки выборки исходных данных

Для использования в нейросетевом моделировании выборка исходных данных должна удовлетворять ряду требований, а именно соответствовать используемой структуре нейронной сети, содержать уникальные (неповторяющиеся) примеры, быть непротиворечивой и репрезентативной.

**Соответствие используемой структуре** сети, как правило, определяется имеющимся составом входных и выходных переменных, а также соотношением объема обучающей выборки и количества рассчитываемых весовых коэффициентов.

Обеспечение **уникальности** примеров исходной выборки позволяет исключить ситуацию, при которой по сути один и тот же пример участвует в настройке весовых коэффициентов в несколько раз чаще, чем все остальные.

**Непротиворечивость** выборки достигается за счет того, что в ней исключается присутствие примеров с одинаковыми (или очень близкими по величине) наборами значений входных переменных и серьезно различающимися значениями выходных переменных.

Анализ обучающей выборки на наличие повторяющихся и противоречивых примеров проводится следующим образом.

1. При настройке нейронной сети задается предельное значение параметра сходства входных переменных двух примеров  $dx_{\text{пр}}$ .

2. Для каждой возможной пары примеров рассчитывается расстояние между входными векторами  $dx_{ij}$ :

$$dx_{ij} = \sqrt{\sum_{q=1}^m (x_q^{(i)} - x_q^{(j)})^2}.$$

3. Если найденное расстояние  $dx_{ij}$  меньше предельного значения  $dx_{\text{пр}}$ , считается, что два данных примера имеют практически идентичный набор входных значений. При наличии хотя бы у одного из таких примеров аналогичной близости с третьим считается, что практически идентичный набор входных значений имеется у группы из трех примеров. Аналогичным образом может быть выделена группа из любого другого количества практически идентичных примеров.

4. Если в результате проведенного анализа была выделена группа из практически идентичных примеров, у них оценивается близость выходных векторов. Для этого задается предельное значение параметра сходства выходных переменных двух примеров  $dy_{\text{пр}}$  и для каждой возможной пары примеров рассчитывается расстояние между выходными векторами  $dy_{ij}$ :

$$dy_{ij} = \sqrt{\sum_{q=1}^m (y_q^{(i)} - y_q^{(j)})^2}.$$

5. Аналогично принципу, изложенному в п. 3 данного алгоритма, анализируется близость между всеми выходными векторами группы примеров. Если все примеры оказались близки по выходному набору значений переменных, данные примеры считаются повторяющимися. Если хотя бы для одного примера все рассчитанные расстояния до остальных примеров группы превысили предельное значение, считается, что вся группа содержит противоречивые примеры.

6. Группа повторяющихся примеров заменяется одним единственным примером, в котором значения элементов входного и выходного векторов представляют собой средние арифметические соответствующих значений переменных примеров этой группы. При обнаружении противоречий в группе выполняется одно из следующих действий:

- исключаются все примеры, имеющие одинаковый вектор входных переменных, если хотя бы у двух из них обнаруживается одно или несколько отличающихся значений выходных переменных (из выборки полностью исключается вся группа);

- оставляется единственный из группы противоречивых примеров, если он встречается наибольшее количество раз по сравнению с остальными (оценивается по количеству раз выполнения условия:  $dy_{ij} < dy_{\text{пр}}$ );

- формируется и включается в выборку единственный пример, каждый элемент выходного вектора которого повторяется наибольшее количество раз в исходной группе противоречивых примеров;

- формируется и включается в выборку единственный пример, каждый элемент выходного вектора которого есть среднее арифметическое соответствующих элементов в исходной группе противоречивых элементов.

**Репрезентативность** выборки характеризуется наличием в ней ши-

рокого спектра значений входных переменных и их различных комбинаций, достаточных для получения нейросетевой модели, корректно и с требуемым уровнем погрешности решающей поставленную задачу также при любых комбинациях входных значений.

Практика применения искусственных нейронных сетей показывает, что в отдельных случаях имеющийся набор входных обучающих данных не может быть напрямую использован для обучения даже после обеспечения требований уникальности и непротиворечивости. Природа некоторых входных параметров нейросетевых моделей такова, что соответствующие им значения переменных изменяются в очень широком интервале, границы которого различаются на несколько порядков, причем пик распределения количества примеров оказывается практически полностью смещен к одной из границ интервала изменения переменной: минимальной или максимальной. Таким образом, в силу вполне объективных причин оказывается нарушенным требование репрезентативности.

Сделать выборку репрезентативной в этом случае можно, применив масштабирование одной или нескольких переменных. Применяв его, мы добиваемся более равномерного распределения примеров в области допустимых значений переменных. В качестве метода масштабирования входных данных в данном случае может быть использовано их логарифмирование натуральным или десятичным логарифмом. При логарифмировании и последующем возвращении к исходному масштабу следует учитывать поправку значения переменной, выводящее ее из отрицательного интервала исходных значений.

Рассмотрим еще один способ предварительного масштабирования данных. Пусть одной из входных переменных нейросетевой модели является угловая градусная мера, которая может принимать целое значение в пределах от 0 до  $359^\circ$ . В данном случае значения переменной 0 и  $359^\circ$  по сути идентичны, однако разница между числами настолько высока, что при дальнейшей обработке они окажутся на противоположных границах пределов нормализации. Масштабирование градусной меры предусматривает замену одной входной переменной (угловой градусной меры) двумя другими (ее синусом и косинусом). Тогда между парами соответствующих значений, выраженных тригонометрическими функциями, будут иметь ме-



сто минимальные отличия, что отвечает физическому смыслу исходной переменной и выразится в более корректной работе нейросетевой модели.

При оценке репрезентативности выборки анализируется распределение ее примеров по диапазонам внутри используемых пределов нормализации.

1. Задается количество диапазонов  $n_d$  по каждой переменной, определяются длина и границы каждого диапазона, рассчитывается количество примеров выборки, попавших в каждый диапазон.

2. Для каждого отдельно взятого диапазона:

– задается уровень допустимого отклонения от равномерного распределения  $d_p$  (обычно 10–20 %);

– рассчитывается среднее количество примеров в диапазоне:

$$N_d = \frac{N}{n_d};$$

– рассчитывается показатель репрезентативности (распределения примеров в каждом диапазоне  $D_d$ ) как отношение их фактического числа в нем  $N_d$  к среднему показателю по всем диапазонам:

$$D_d = \frac{N_d}{N_d}.$$

Характеристики категорий значений показателя репрезентативности и соответствующие им рекомендации по работе с выборками данных представлены в таблице.

#### Характеристики и рекомендации на основе показателя репрезентативности

Категория $D_d$	Характеристика категории и рекомендации
$D_d < (1 - d_p)$	Диапазон описан недостаточно полно по сравнению с другими. Целесообразно дополнить выборку новыми примерами
$(1 - d_p) \leq D_d \leq (1 + d_p)$	Диапазон описан достаточно полно
$D_d > (1 + d_p)$	Имеется избыток примеров выборки для данного диапазона. Целесообразно исключить часть примеров

Для принятия окончательного решения по работе с выборкой делается интегрированная оценка для каждой переменной по всем ее диапазонам.

## **6. Особенности постановки и решения задач с использованием искусственных нейронных сетей**

**Аппроксимация и интерполирование данных** – наверно наиболее часто решаемый с помощью искусственных нейронных сетей класс задач. И это неудивительно, потому что даже классические методы их решения зачастую используют функциональные зависимости, находящиеся вне связи с физикой описываемых ими процессов или явлений. И в том, и в другом случаях на первое место выходят статистическое соответствие и адекватность полученной модели экспериментальным данным.

Среди нейросетевых архитектур для решения задач аппроксимации и интерполирования данных могут быть использованы однослойные и многослойные перцептроны, а также сети на основе радиально-симметричных функций. Входы и выходы таких сетей – нормализованные аналоговые сигналы.

Общая постановка задачи следующая. Имеется выборка экспериментальных данных, представляющая собой множество примеров (опытов). Каждый пример – это вектор значений входных переменных и соответствующих им значений выходов. Требуется с использованием одной из архитектур нейронных сетей получить нейросетевую модель, позволяющую с нулевой или наименьшей из возможных погрешностью описать многосвязную зависимость выходов от входов в пределах изменения независимых переменных модели, соответствующих диапазонам их изменения в экспериментальной выборке.

Качество полученной модели оценивается как традиционными статистическими методами, например, проверкой ее адекватности по критерию Фишера, так и путем расчета средней относительной ошибки в нормализованном и исходном масштабах для обучающей и тестовой выборок.

Главное преимущество нейросетевого подхода при решении задач интерполирования и аппроксимации данных заключается в возможности

получения единственной модели, полностью описывающей множественные взаимные связи между ее выходными и входными переменными, в том числе и в случае высокой нелинейности этих связей.

**Прогнозирование временных рядов** – задача во многом похожая на аппроксимацию. В результате ее решения должно быть получено изменение выходных переменных модели во времени. Тем не менее, сама переменная времени может как напрямую фигурировать среди входных переменных (1-й вид моделей), так и быть задана в неявном виде, являясь специфическим свойством, различающим однотипные входы и выходы (2-й вид моделей). Примером второго вида могут быть модели, включающие входные и/или выходные вектора концентрации вещества, измеренной в различные (как правило, последовательные и равноотстоящие) моменты времени.

Для решения данного класса задач также используются однослойные и многослойные перцептроны и нейронные сети на основе радиально-симметричных функций. Входы и выходы таких сетей – как правило, нормализованные аналоговые сигналы.

Задача получения временного ряда с использованием нейросетевой модели ставится следующим образом. Дана выборка с результатами наблюдений изменения одной или нескольких целевых характеристик во времени. Требуется получить нейросетевую модель, позволяющую спрогнозировать значения целевых характеристик в один или несколько заданных моментов времени вперед. Помимо динамики целевых переменных для обучения нейронных сетей зачастую используются дополнительные переменные, характеризующие внешние условия, влияющие на эту динамику.

Во многих случаях в нейросетевые модели добавляются обратные связи, позволяющие подавать в качестве входных переменных значения, полученные на выходе в результате прохождения сигналов по нейронной сети на предыдущем такте расчета.

Специально для прогнозирования временных рядов исследователями были предложены архитектуры нейронных сетей с обратными связями на основе многослойных перцептронов, такие, как искусственные нейронные сети Элмана и Джордана.

Задачи **распознавания и ассоциации образов** сводятся к выделению некоторого эталонного сигнала, хранящегося в памяти нейронной сети из зашумленного входного образа. Общая постановка таких задач следующая. Дано множество эталонных образов либо пар соответствия входных и выходных эталонных образов в виде векторов бинарных и/или аналоговых сигналов. Нужно получить нейросетевую модель, позволяющую:

- удалить из входного сигнала шум (ошибки) и получить на выходе чистый (эталонный) сигнал либо
- сопоставить зашумленный входной образ с соответствующим ему эталонным выходным образом.

Таким образом, формат сигнала на входе может совпадать, но может и отличаться от формата выходного сигнала. В первом случае для решения задач, как правило, применяют нейронную сеть Хопфилда, имитирующую работу автоассоциативной памяти. Во втором – нейронную сеть Коско – нейросетевой аналог гетероассоциативной памяти. Входы и выходы таких сетей – бинарные.

Также распознавание образов можно реализовать с использованием однослойных или многослойных перцептронов. В этом случае входы и выходы могут быть как аналоговыми, так и бинарными. Во втором случае, так как все вычисления также производятся в аналоговой форме, чтобы получить бинарные выходы модели, требуется использование активационных функций единичного скачка в выходных нейронах.

Однако выбор перцептрона для решения задачи распознавания образов связан с определенными проблемами. Дело в том, что эталонных образов, как правило, не очень много, а количество используемых входных и выходных переменных, наоборот, достаточно большое. Все это приводит к тому, что в процессе обучения структурно насыщенного перцептрона приходится настраивать достаточно большое количество весовых коэффициентов при ограниченном объеме обучающей выборки. Сама нейронная сеть в этом случае обучается на различных возможных зашумленных образах, которым каждый раз ставятся в соответствие заранее известные эталоны. На стадии практического использования обученная нейронная сеть в идеале должна уметь выделять эталонный образ в том числе из вариантов зашумленных образов, которые ей ранее никогда на входы не предъявля-

лись.

В отличие от предыдущего случая при решении задачи **классификации образов** требуется подаваемый на входы сети зашумленный образ отнести к одному из известных классов, а не восстановить или получить сам эталон. Характерно, что практически любую задачу распознавания или ассоциации образов можно привести к задаче классификации, но обратное утверждение неверно.

Задача классификации ставится следующим образом. Имеется множество векторов входных образов, переменные которых представлены в форме аналоговых и/или бинарных сигналов. Для каждого из входных образов известен класс соответствия. Требуется получить нейросетевую модель, позволяющую выполнить одно из следующих действий:

- однозначно указать класс образов, к которому может быть отнесен входной вектор;
- ограничить круг возможных классов, к которым может быть отнесен зашумленный входной образ;
- сделать заключение о невозможности отнесения входного образа к одному из известных сети классов.

В зависимости от выбранной архитектуры нейронная сеть может обучаться классификации только на эталонных образах (сети Хэмминга или Коско) или как на эталонных, так и на зашумленных образах (однослойные и многослойные перцептроны, сети на основе радиально-симметричных функций). При решении задач данного класса входные переменные нейросетевой модели могут быть представлены как в бинарной, так и в аналоговой формах. Выходные – строго в бинарной форме. Таким образом, в выходных нейронах однослойных и многослойных перцептронов должны использоваться активационные функции единичного скачка.

Бинарный выходной вектор для нейросетевой модели с известным количеством возможных классов  $N_C$  может быть представлен двумя способами:

1. Количество выходов нейронной сети  $K$  определяется как значение  $\log_2 N_C$ , округленное в большую сторону. Каждому классу образов должен соответствовать выходной вектор, являющийся уникальной бинарной последовательностью.

2. Количество выходов нейронной сети равно количеству возможных классов образов. Каждому классу должен соответствовать уникальный выходной вектор, включающий единственное единичное (положительное) и все остальные нулевые (отрицательные) значения.

Первый способ лучше всего использовать в задачах с большим количеством классов образов. Это позволяет значительно упростить структуру нейронной сети. Второй же способ представления выходов позволяет получать более точные результаты классификации.

Задача **кластеризации** данных немного похожа на классификацию. Ее принципиальное отличие заключается в том, что при кластеризации изначально неизвестно количество будущих кластеров. Оно определяется количественным и качественным составом обучающей выборки и настройками нейронной сети.

Постановка задачи кластеризации следующая. Дана экспериментальная выборка, представляющая собой множество векторов переменных, характеризующих известные свойства объектов, событий, процессов или явлений. Для каждого из примеров выборки могут быть известны дополнительные данные или описание, относящиеся лишь к самим отдельно взятым примерам, но не участвующие непосредственно в обучении нейронной сети. Требуется получить нейронную сеть, которая на протяжении всего своего жизненного цикла будет выполнять одну или несколько функций из списка:

- относить подаваемые на ее входы вектора к одному из уже существующих кластеров, наиболее близкому к входному вектору;
- создавать новый кластер, если не обнаружено достаточно близкого соответствия существующим кластерам;
- удалять кластеры, оказавшиеся неиспользованными в результате последней эпохи обучения.

Среди архитектур нейронных сетей для решения задач кластеризации чаще всего рекомендуются сети Кохонена и сети адаптивного резонанса.

Входные переменные таких сетей могут принимать как бинарные, так и аналоговые значения. Их выходными значениями можно условно считать расстояние до центра каждого из существующих кластеров. Фак-

тически же выходов в понимании, привычном для других нейросетевых архитектур, у них нет, так как отсутствует четкая заранее известная цель обучения – выходной образ или класс образов.

## **7. Этапы жизненного цикла нейронной сети**

Использование искусственной нейронной сети для решения прикладных задач – трудоемкий, а в некоторых случаях и длительный процесс. В этой связи можно выделить и описать несколько этапов ее жизненного цикла.

**1. Постановка задачи и выбор архитектуры нейронной сети.** На данном этапе необходимо получить четкое понимание о решаемой задаче, причинах использования для этого метода нейросетевого моделирования и преимуществах, которые он должен дать. Под конкретную постановку задачи подбирается архитектура нейронной сети. Подробнее о возможных задачах и соответствующих им архитектурах искусственных нейронных сетей можно узнать из предыдущей главы.

**2. Определение количественного и качественного составов входов и выходов.** Качественный состав входных и выходных переменных нейросетевой модели определяется, главным образом, поставленной на предыдущем этапе задачей. Количественный состав может зависеть еще от природы входных и выходных переменных и описываемых ими процессов и свойств. Так, при необходимости масштабирования некоторых переменных число фактических входов и выходов нейронной сети может увеличиться по сравнению с количеством входов и выходов модели.

**3. Формирование исходной выборки данных.** Исходная выборка данных может быть сформирована на основе практического или вычислительного эксперимента. Чем больше объем проведенного эксперимента, тем более точную нейросетевую модель можно получить и использовать на практике. Следует также учитывать и требования к выборке исходных данных, сформулированные в предыдущей главе.

**4. Предварительная обработка и нормализация исходной выборки.** Данный этап определяется выбранной архитектурой нейронной сети,

составом входных и выходных переменных и, собственно, фактическими данными, полученными в результате эксперимента.

**5. Разделение исходной выборки на обучающую и тестовую составляющие.** Имеющаяся исходная выборка делится на обучающую и тестовую подвыборки с учетом используемой архитектуры нейронной сети. Обычно объем обучающей подвыборки в несколько раз больше, чем тестовой. Первая используется строго для настройки весовых коэффициентов. Вторая – для проверки корректности настроенной нейросетевой модели. В случае ограниченного объема исходной выборки она вся может быть задействована для обучения, а оценка корректности работы сети проверяется на этапе практической эксплуатации.

**6. Определение структуры нейронной сети.** Помимо ранее установленного состава входных и выходных переменных на данном этапе задается количество скрытых слоев и нейронов в каждом слое. Для многих архитектур нейронных сетей выбор структуры зависит, в том числе, от объема обучающих данных.

**7. Настройка параметров нейронной сети и алгоритма ее обучения.** На данном этапе задаются вид и параметры активационных функций скрытых и выходных нейронов. Для итерационного алгоритма выбираются условия окончания обучения, коэффициент скорости, порядок предъявления примеров обучающей выборки.

**8. Обучение нейронной сети.** Для точных алгоритмов обычно осуществляется однократный расчет весовых коэффициентов. Для итерационных – многократное повторение эпох обучения. Для обучения могут быть использованы только примеры обучающей подвыборки.

**9. Контрастирование нейронной сети.** Если алгоритм обучения нейронной сети подразумевал итерационную подстройку весовых коэффициентов на протяжении всех эпох обучения, нередки случаи, когда некоторые из весов по абсолютному значению настолько близки к нулю, что не оказывают никакого практического влияния на результаты расчетов. Тем не менее, на такие расчёты тратится довольно много вычислительных ресурсов, что заметно замедляет процесс обучения. Для решения этой проблемы такие весовые коэффициенты полностью обнуляются, то есть соответствующая связь между входами и нейронами или между двумя нейро-



нами исключается из структуры сети.

**10. Тестирование нейронной сети** производится с использованием примеров тестовой выборки. По результатам тестирования оценивается ошибка работы нейросетевой модели.

Этапы 7–9 могут выполняться в цикле неоднократно до тех пор, пока не будет получена настроенная нейронная сеть, с помощью которой можно решать задачу с требуемым уровнем ошибки.

**11. Практическое использование.** Обученная и протестированная нейронная сеть используется для практического решения задачи по мере поступления новых исходных данных (значений входных переменных). Результаты по обученной сети могут быть получены практически мгновенно и интерпретированы пользователем для последующего принятия решений.

**12. Дообучение нейронной сети.** Получаемые в ходе практической эксплуатации нейронной сети пары входных и выходных векторов могут использоваться для дальнейшей подстройки весовых коэффициентов. Это особенно важно, если изначально объем выборки данных был небольшим. Часть архитектур нейронных сетей – использующих принцип самоорганизации – в ходе дообучения могут не только изменять значения весовых коэффициентов, но и менять свою структуру.

Следует заметить, что перечисленная в данной главе последовательность этапов характерна в общем для работы с искусственными нейронными сетями. Однако для ряда архитектур могут иметь место собственные только им отличия. Подробно особенности реализации и функционирования отдельных архитектур искусственных нейронных сетей приводятся далее в соответствующих главах.

## **8. Однослойные перцептроны**

В современном понимании **перцептроны** представляют собой однослойные или многослойные искусственные нейронные сети прямого распространения с бинарными или аналоговыми выходными сигналами, обучающиеся с учителем. Они хорошо подходят для решения нескольких типов задач: аппроксимации данных, прогнозирования состояния на основе

временного ряда, распознавания образов и классификации, а также могут быть использованы в других задачах сами по себе или совместно с другими методами моделирования.

На рис. 8.1 представлена структура однослойного перцептрона с  $M$  входами и  $K$  выходами. Очевидно, что каждый выход соответствует своему нейрону единственного слоя. Кроме того, ясно, что сложность структуры однослойной сети не может варьироваться ввиду отсутствия скрытых слоев нейронов.

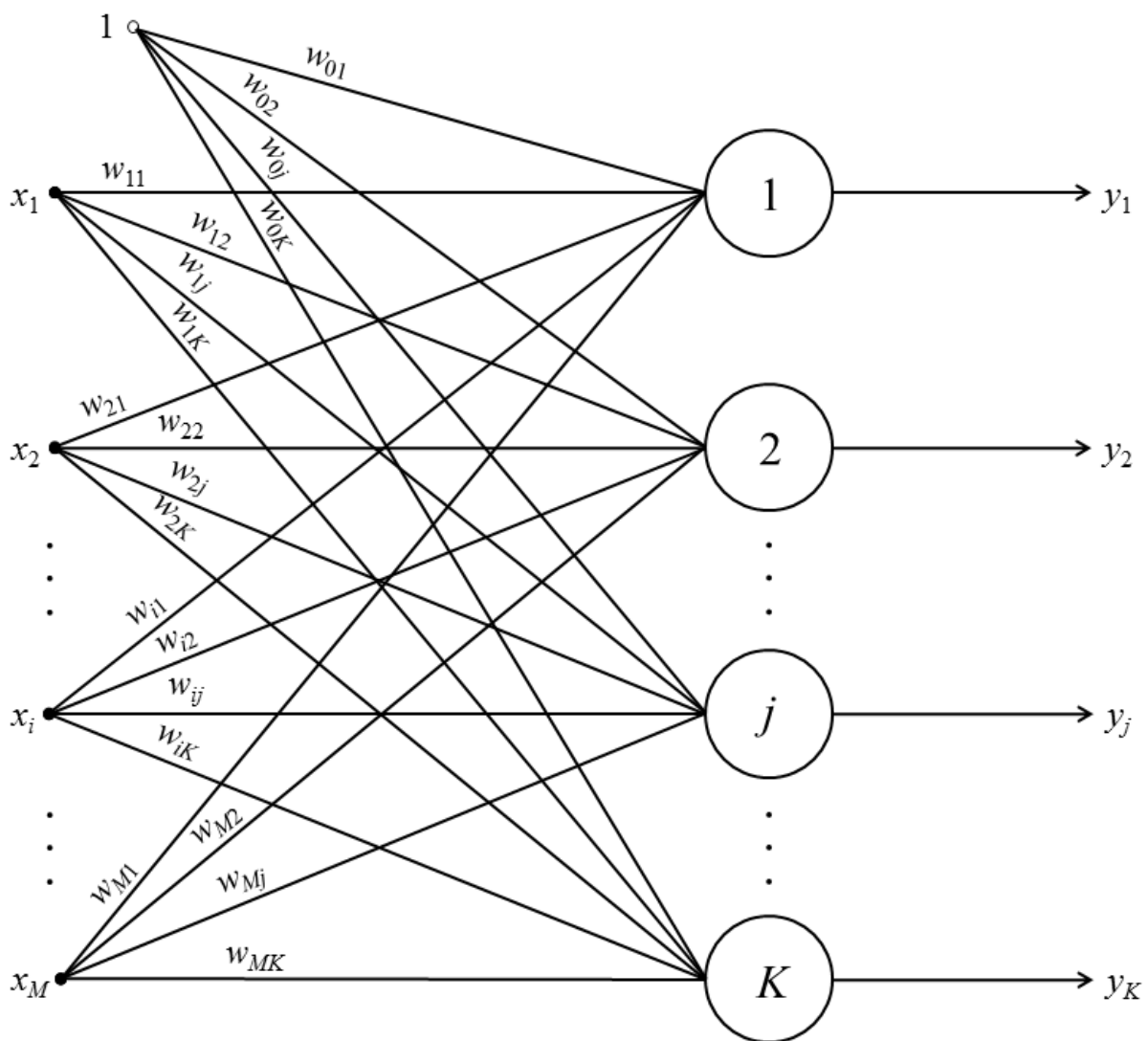


Рис. 8.1. Структура однослойного перцептрона

Количество весовых коэффициентов  $N_w$ , настраиваемых в процессе обучения, рассчитывается следующим образом:

$$N_w = (M + 1) \cdot K.$$

Для каждого нейрона сети помимо синаптических связей с элементами входного вектора настраивается связь с фиктивным единичным входом (коэффициент смещения).

Так как выходные переменные могут принимать как бинарные, так и аналоговые значения, выбор вида активационных функций ограничен только областью допустимых значений выходных сигналов, принятой для нормализации.

Однослойные перцептроны обучаются на основе итерационного **метода Уидроу–Хоффа**, иначе называемого дельта-правилом. Алгоритм данного метода следующий.

1. Весовые коэффициенты однослойного перцептрона выбранной структуры инициализируются небольшими по абсолютной величине (не более  $M^{-1}$ ) случайными значениями.

2. На входы перцептрона подается входной вектор одного из примеров обучающей выборки. Производится прямое распространение сигналов по сети с расчетом значений выходных переменных  $\tilde{y}_j^p$ .

3. Для каждого рассчитанного значения выходной переменной вычисляется погрешность по сравнению со значениями элементов выходного вектора взятого обучающего примера  $\tilde{y}_j^n$ :

$$\Delta_j = \tilde{y}_j^n - \tilde{y}_j^p. \quad (8.1)$$

4. Выполняется коррекция старых значений весовых коэффициентов каждого нейрона  $w_{ij}^{(q)}$  на основе погрешности соответствующей выходной переменной:

$$w_{ij}^{(q+1)} = w_{ij}^{(q)} + v \Delta_j \tilde{x}_i, \quad (8.2)$$

где  $v$  – коэффициент скорости обучения.

5. Цикл повторяется с шага 2 до выполнения одного или нескольких условий окончания:

- исчерпано заданное предельное количество эпох обучения;
- достигнут удовлетворительный уровень ошибки по всей обучающей выборке;
- не происходит уменьшения ошибки обучающей выборки на протя-

жении заданного предельного количества эпох обучения;

– исчерпано заданное предельное физическое время обучения.

На протяжении одной эпохи обучения последовательно или в случайном порядке предъявляются все примеры обучающей выборки, причем каждый пример должен быть предъявлен однократно.

Коэффициент скорости обучения задается положительной константой или переменной величиной ( $0 < \nu \leq 1$ ), постепенно уменьшающейся в процессе обучения нейронной сети.

Общая ошибка работы нейронной сети с обучающей выборкой оценивается по соотношению:

$$\tilde{E}_{\text{об.}} = \sqrt{\frac{1}{NK} \sum_{k=1}^N \sum_{j=1}^K (\tilde{y}_{jk}^{\text{п}} - \tilde{y}_{jk}^{\text{р}})^2}. \quad (8.3)$$

Также полезно оценивать ошибку работы нейронной сети в отдельности по каждому ее выходу:

$$\tilde{E}_j = \sqrt{\frac{1}{N} \sum_{k=1}^N (\tilde{y}_{jk}^{\text{п}} - \tilde{y}_{jk}^{\text{р}})^2}. \quad (8.4)$$

Порядок величины полученных в последнем случае ошибок сопоставим со средней абсолютной ошибкой нормализованных выходных переменных. Как следствие, с ее помощью можно оценить порядок величины абсолютной ошибки в исходных единицах:

$$E_j = \tilde{E}_j \cdot (y_{\max j} - y_{\min j}) \quad (8.5)$$

или относительной ошибки в процентах:

$$E_{\text{отн}} = \tilde{E}_j \cdot 100 \%. \quad (8.6)$$

Рассмотрим пример обучения и оценки работы однослойной нейронной сети (рис. 8.2), описывающей зависимость двух выходных переменных  $y_1, y_2$  от двух входных  $x_1, x_2$ . Данная сеть имеет 6 синаптических связей и, соответственно, в процессе ее обучения будет настраиваться такое же количество весовых коэффициентов.

Исходная выборка, состоящая из 10 примеров, приведена в табл. 8.1.

При обучении и практическом использовании однослойного перцептрона необходимо нормализовать значения переменных. Воспользуемся методом линейной нормализации в пределах  $[0, 1]$  по соотношению (3.1).

Результаты нормализации выборки представлены в табл. 8.2.

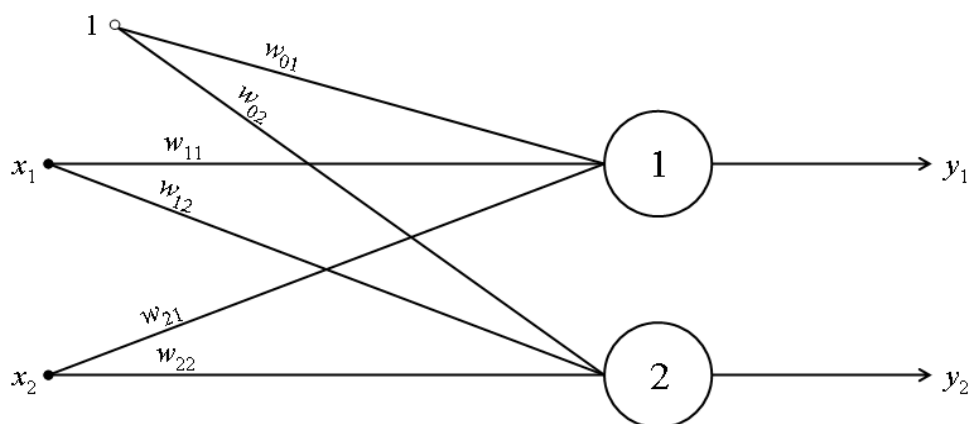


Рис. 8.2. Однослойный перцептрон с 2 входами и 2 выходами

Таблица 8.1

**Исходная выборка данных для однослойного перцептрона**

№ примера	$x_1$	$x_2$	$y_1$	$y_2$
1	-5	2	-18	-1
2	0	5	-9	16
3	2	-4	15	-18
4	-3	1	-10	-3
5	5	0	16	1
6	1	-5	14	-23
7	-3	-1	-6	-11
8	2	5	-3	18
9	4	3	7	12
10	0	-2	5	-12
Минимальное значение	-5	-5	-18	-23
Максимальное значение	5	5	16	18

В качестве активационной функции используем сигмоидную логистическую (4.1) с параметром насыщения  $\alpha = 1,0$ . Коэффициент скорости обучения примем равным 0,9.

В соответствии с алгоритмом метода Уидроу–Хоффа зададимся следующими начальными значениями весовых коэффициентов:  $w_{01} = 0,0$ ,  $w_{02} = 0,2$ ,  $w_{11} = -0,4$ ,  $w_{12} = -0,1$ ,  $w_{21} = 0,3$ ,  $w_{22} = 0,2$ .

Предъявим сети элементы входного вектора первого обучающего примера:  $\tilde{x}_1 = 0,0$ ,  $\tilde{x}_2 = 0,7$ . В результате прямого распространения сигнала

по сети по соотношению (2.1) получаются состояния нейронов  $s_1 = 0,21$  и  $s_2 = 0,34$ . В результате применения активационной функции к этим состояниям рассчитываются нормализованные выходные значения:  $\tilde{y}_1^p = 0,552$ ,  $\tilde{y}_2^p = 0,584$ .

Таблица 8.2

**Нормализованная исходная выборка для обучения и практического использования однослойного перцептрона**

№ примера	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{y}_1$	$\tilde{y}_2$
1	0,000	0,700	0,000	0,537
2	0,500	1,000	0,265	0,951
3	0,700	0,100	0,971	0,122
4	0,200	0,600	0,235	0,488
5	1,000	0,500	1,000	0,585
6	0,600	0,000	0,941	0,000
7	0,200	0,400	0,353	0,293
8	0,700	1,000	0,441	1,000
9	0,900	0,800	0,735	0,854
10	0,500	0,300	0,676	0,268

С использованием выражения (8.1) рассчитаем погрешности выходных значений:  $\Delta_1 = -0,552$ ,  $\Delta_2 = -0,048$ . Данные значения погрешностей далее берем для пересчета новых значений весовых коэффициентов по выражению (8.2). Результаты пересчета весовых коэффициентов после предъявления первого и всех последующих обучающих примеров в течение первой эпохи обучения представлены в табл. 8.3. В последней строке таблицы приведены весовые коэффициенты по завершении десятой эпохи обучения.

С использованием весовых коэффициентов, представленных в последней строке табл. 8.3, были рассчитаны нормализованные выходные значения всех примеров обучающей выборки (табл. 8.4) и получены результаты оценки ошибок обучения.

Общая ошибка работы полученной нейронной сети с обучающей выборкой составила  $\tilde{E}_{об.} = 0,0773$ . Относительные ошибки каждого выхода:  $\varepsilon_1 = 6,66 \%$ ,  $\varepsilon_2 = 8,67 \%$ . Денормализованные средние абсолютные ошибки:

$E_1 = 2,27$ ,  $E_2 = 3,55$ . Для однослойного перцептрона такой простой структуры, какая была использована, это очень хороший уровень ошибок.

Таблица 8.3

**Ход коррекции весовых коэффициентов однослойного перцептрона**

Номер эпохи	Номер примера	$w_{01}$	$w_{02}$	$w_{11}$	$w_{12}$	$w_{21}$	$w_{22}$
1	1	-0,497	0,157	-0,400	-0,100	-0,048	0,170
1	2	-0,549	0,501	-0,426	0,072	-0,099	0,514
1	3	0,057	0,029	-0,002	-0,258	-0,039	0,467
1	4	-0,189	-0,039	-0,051	-0,272	-0,186	0,426
1	5	0,335	0,060	0,473	-0,173	0,076	0,475
1	6	0,597	-0,380	0,630	-0,437	0,076	0,475
1	7	0,303	-0,505	0,571	-0,462	-0,042	0,425
1	8	0,107	0,035	0,434	-0,084	-0,238	0,965
1	9	0,250	0,195	0,563	0,060	-0,124	1,094
1	10	0,300	-0,135	0,588	-0,105	-0,109	0,995
...	...	...	...	...	...	...	...
10	10	-0,190	-2,110	3,513	0,460	-2,474	3,826

Таблица 8.4

**Сравнение значений выходных переменных обучающих выборок и рассчитанных по обученному однослойному перцептрону**

№ примера	Выходы выборки		Выходы перцептрона	
	$\tilde{y}_1^n$	$\tilde{y}_2^n$	$\tilde{y}_1^p$	$\tilde{y}_2^p$
1	0,000	0,537	0,128	0,638
2	0,265	0,951	0,288	0,875
3	0,971	0,122	0,883	0,197
4	0,235	0,488	0,275	0,569
5	1,000	0,585	0,890	0,565
6	0,941	0,000	0,872	0,138
7	0,353	0,293	0,383	0,380
8	0,441	1,000	0,449	0,885
9	0,735	0,854	0,730	0,797
10	0,676	0,268	0,695	0,325

Многократное повторение эпох обучающей процедуры с различным случайным порядком предъявления обучающих примеров, как правило,

способствует значительному снижению ошибок (рис. 8.3).

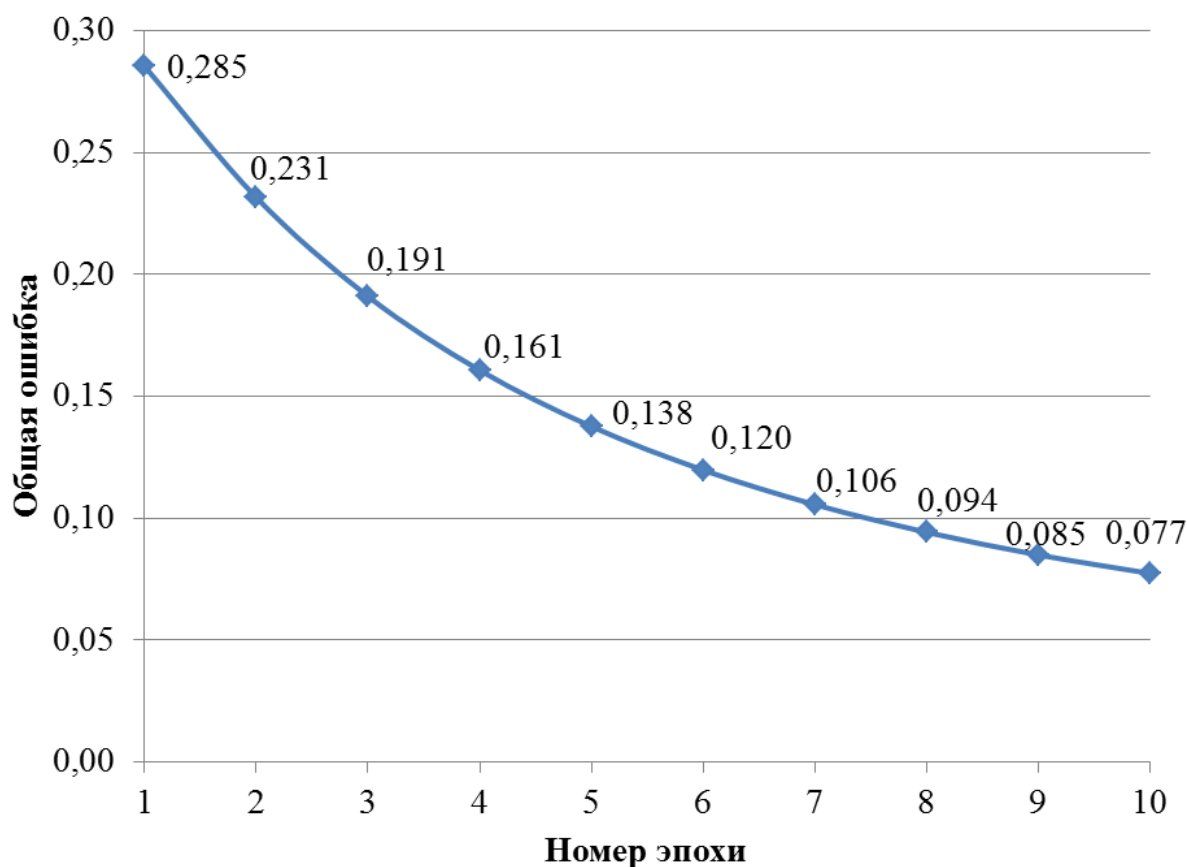


Рис. 8.3. Изменение общей ошибки по эпохам обучения

Как видно из рисунка, дальнейшее продолжение обучения, скорее всего, способствовало бы еще большему снижению общей ошибки. Для уточнения оптимальных значений весовых коэффициентов последние эпохи проводятся с минимальным коэффициентом скорости обучения.

## 9. Многослойные перцептроны

Многослойные перцептроны эффективны при решении тех же самых задач, что и однослойные перцептроны, но обладают значительно большей вычислительной способностью в сравнении с однослойными перцептронами. Благодаря этой своей способности они могут гораздо точнее описывать многомерные зависимости с большой степенью нелинейности и высо-



ким уровнем перекрестного и группового влияния входных переменных на выходные.

Пример двухслойного перцептрона с  $M$  входами и  $K$  выходами был ранее приведен на рис. 2.2. При необходимости использования сетей более сложной структуры добавляются новые скрытые слои или наращивается количество нейронов в скрытых слоях.

Количество весовых коэффициентов, настраиваемых в процессе обучения многослойного перцептрона с  $L$  скрытыми слоями по  $m_l$  нейронов в каждом, рассчитывается следующим образом:

$$N_w = (M + 1)m_1 + \sum_{l=2}^L (m_{l-1} + 1)m_l + (m_L + 1)K .$$

Для каждого нейрона сети помимо синаптических связей с элементами входного вектора настраивается связь с фиктивным единичным входом (коэффициент смещения).

Для большинства задач, решаемых с помощью многослойных перцептронов, выбор структуры нейронной сети должен осуществляться на основе следующего правила («Правила 2–5»): *количество настраиваемых в процессе обучения весовых коэффициентов должно быть в 2–5 раз меньше, чем количество примеров обучающей выборки*. Если это соотношение меньше 2, сеть теряет способность к обобщению обучающей информации, а при достижении 1 и меньше просто запоминает ответы для каждого обучающего примера. Если же количество обучающих примеров слишком велико для выбранной структуры сети, нейросетевая модель во многих случаях просто усредняет выходные значения для различных комбинаций входных векторов, теряя способность к корректному отклику в отдельных частных случаях и повышая величину максимальной выборочной ошибки.

Кроме того, при выборе структуры многослойного перцептрона следует задавать количество нейронов в скрытом слое, предшествующем выходному слою, не меньшим, чем количество самих выходов.

Метод Уидроу–Хоффа, рассмотренный в предыдущей главе, не может быть использован для настройки весовых коэффициентов многослойных перцептронов, поскольку он позволяет сделать прямую коррекцию по величине ошибки только для нейронов выходного слоя, тогда как синапти-

ческие коэффициенты скрытых нейронов также требуют изменения.

Задача обучения многослойных перцептронов может быть сформулирована как оптимизационная, в которой целевой функцией (критерием оптимизации) является общая ошибка, рассчитанная по обучающей выборке (8.3). Соответственно, и решать данную задачу можно как любую задачу многомерной оптимизации с использованием методов детерминированного, градиентного или стохастического поиска.

Наиболее распространенный метод обучения многослойного перцептрона – **метод обратного распространения ошибки**. Суть данного метода заключается в том, что сигнал ошибки каждого выходного значения, рассчитанный на текущем такте обучения, распространяется по слоям в обратном направлении (от выходного к первому) с учетом тех же весовых коэффициентов, которые использовались при прямом прохождении входных сигналов по нейронной сети и расчете выходных значений.

Алгоритм метода обратного распространения ошибки включает следующие этапы.

1. Весовые коэффициенты многослойного перцептрона выбранной структуры инициализируются небольшими по абсолютной величине (не более  $M^{-1}(L + 1)^{-1}$ ) случайными значениями, где  $L$  – количество скрытых слоев.

2. На входы нейронной сети подается входной вектор одного из примеров обучающей выборки. Производится прямое распространение сигналов по сети с расчетом значений выходных переменных  $\tilde{y}_j^p$ .

3. Для каждого рассчитанного значения выходной переменной по соотношению (8.1) вычисляется погрешность по сравнению со значениями элементов выходного вектора взятого обучающего примера  $\tilde{y}_j^n$ .

4. По полученным погрешностям рассчитываются невязки нейронов выходного слоя с учетом производных соответствующих активационных функций:

$$\delta_j = \Delta_j \cdot f'(s_j). \quad (9.1)$$

Если активационная функция имеет вид сигмоиды или функции Гаусса, следует воспользоваться соотношениями для аналитического расчета ее производной из главы 4.

5. В обратной последовательности (от последнего скрытого слоя к первому) рассчитываются невязки нейронов остальных слоев с учетом связывающих слои синаптических связей:

$$\delta_i = \left( \sum_{j=1}^{m_l} w_{ij} \delta_j \right) \cdot f'(s_i). \quad (9.2)$$

6. Для всех слоев нейронов, кроме первого, пересчитываются значения весовых коэффициентов по следующим соотношениям:

$$w_{0j}^{(q+1)} = w_{0j}^{(q)} + v \delta_j, \quad (9.3)$$

$$w_{ij}^{(q+1)} = w_{ij}^{(q)} + v \tilde{y}_i^p \delta_j, \quad (9.4)$$

где  $v$  – коэффициент скорости обучения.

Для первого скрытого слоя нейронов коррекция весовых коэффициентов происходит не по выходам предыдущего слоя, а по нормализованным значениям входных переменных нейронной сети:

$$w_{ij}^{(q+1)} = w_{ij}^{(q)} + v \tilde{x}_i \delta_j. \quad (9.5)$$

Для расчета коэффициентов смещения нейронов первого скрытого слоя, как и для всех других слоев, используется соотношение (9.3).

7. Цикл повторяется с шага 2 до выполнения одного или нескольких условий окончания:

- исчерпано заданное предельное количество эпох обучения;
- достигнут удовлетворительный уровень ошибки по всей обучающей выборке;
- не происходит уменьшения ошибки обучающей выборки на протяжении заданного предельного количества эпох обучения;
- исчерпано заданное предельное физическое время обучения.

На протяжении одной эпохи обучения последовательно или в случайном порядке предъявляются все примеры обучающей выборки, причем каждый пример должен быть предъявлен однократно.

Коэффициент скорости обучения задается положительной константой или переменной величиной ( $0 < v \leq 1$ ), постепенно уменьшающейся в процессе обучения нейронной сети.

Одна из особенностей метода обратного распространения ошибки заключается в том, что средние абсолютные значения невязок уменьшают-

ся от последнего слоя нейронов к первому на порядки. Следствием этого становится практически незначительная величина изменения весовых коэффициентов первых скрытых слоев и, соответственно, требуется очень большое количество эпох обучения для значимой коррекции весов. Для устранения данного недостатка в многослойных перцептронах с более чем одним скрытым слоем можно использовать коэффициент скорости обучения, увеличивающийся от последнего скрытого слоя к первому в пределах одной эпохи обучения, в том числе, значения  $\nu_l > 1$ .

Ошибки работы всей нейронной сети по обучающей и тестовой выборкам могут быть рассчитаны по соотношениям (8.3)–(8.6).

Метод обратного распространения ошибки базовый для обучения многослойных перцептронов, но не единственно возможный. С одной стороны, поскольку задача обучения любых перцептронов, по сути своей, оптимизационная, с этой целью могут использоваться практически любые методы оптимизации многомерных функций. С другой стороны, оптимизируемая функция ошибки (8.3) не просто многомерная, а в большинстве случаев имеет очень высокую размерность в несколько десятков или даже сотен переменных – синаптических коэффициентов, и, к тому же, может обладать очень сложным рельефом поверхности, что делает невозможным использование на практике большинства традиционных методов. В этой связи реальной заменой методов градиентного поиска, адаптированных для обучения нейронных сетей (например, метода обратного распространения ошибки или метода Левенберга–Марквардта), могут быть лишь эволюционные или многоагентные методы (генетические алгоритмы, искусственные иммунные системы и другие), относящиеся к группе методов стохастического поиска. Зачастую такие методы могут быть даже более полезны, поскольку позволяют на любой эпохе обучения оперировать значениями переменных из всей области их определения, а не останавливаются в локальных оптимумах функции ошибки как методы градиентного поиска.

Если сравнивать архитектуры многослойных и однослойных перцептронов, можно выделить несколько ее отличительных особенностей:

— для одного и того же состава входных и выходных переменных в качестве нейросетевой модели могут быть использованы многослойные

перцептроны различной структуры (с различным количеством скрытых слоев и нейронов в них), тогда как при использовании однослойных перцептронов возможен лишь один вариант структуры сети;

– многослойные перцептроны позволяют получить более корректное математическое описание многосвязных функциональных зависимостей с ярко выраженной нелинейностью;

– для обучения многослойных перцептронов требуется выбор более сложного алгоритма, чем для обучения однослойных перцептронов;

– обучение многослойных перцептронов занимает большее время и требует больший объем вычислительных ресурсов компьютера.

Рассмотрим пример решения задачи классификации образов с использованием двухслойного перцептрона. Пусть дана выборка данных (табл. 9.1), включающая 48 примеров. Используем первые 40 примеров для обучения перцептрона, а остальные 8 (выделены курсивом) – для тестирования обученной нейронной сети.

Таблица 9.1

**Исходная выборка данных для обучения и практического  
использования двухслойного перцептрона**

№	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$y_3$	$y_4$
1	-0,12	0,42	0,52	0,17	-0,64	-1,89	0	1
2	0,97	0,66	0,47	0,37	0,39	0,51	1	0
3	-2,60	-1,08	-0,24	-0,10	-0,65	-1,90	0	1
4	-1,45	-0,61	-0,41	-0,86	-1,96	-3,70	0	1
5	0,55	0,97	1,11	0,97	0,55	-0,15	0	1
6	0,54	0,53	0,57	0,64	0,75	0,90	1	0
7	0,93	0,93	1,11	1,47	2,00	2,72	1	0
8	-4,07	-2,36	-1,46	-1,36	-2,06	-3,56	0	1
9	3,66	2,25	1,73	2,08	3,31	5,42	1	0
10	4,87	2,25	0,29	-1,02	-1,67	-1,67	1	0
11	0,82	0,79	0,76	0,75	0,74	0,75	1	0
12	4,29	2,19	0,62	-0,41	-0,91	-0,88	1	0
13	-1,06	0,91	1,89	1,86	0,83	-1,20	0	1
14	-1,11	-0,49	-0,46	-1,03	-2,20	-3,97	0	1
15	0,71	-0,31	-0,85	-0,92	-0,51	0,37	1	0
16	-0,93	-0,63	-0,43	-0,35	-0,36	-0,48	0	1
17	-4,78	-2,71	-1,26	-0,46	-0,29	-0,75	0	1
18	-0,42	-0,41	-1,16	-2,67	-4,94	-7,97	0	1

Окончание табл. 9.1

№	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$y_3$	$y_4$
19	3,25	2,21	1,60	1,42	1,67	2,35	1	0
20	-2,07	-1,40	-0,95	-0,72	-0,71	-0,92	0	1
21	4,35	1,59	-0,26	-1,20	-1,22	-0,32	1	0
22	0,68	0,13	0,04	0,42	1,27	2,58	1	0
23	-1,63	-1,58	-2,42	-4,16	-6,79	-10,31	0	1
24	0,97	0,42	0,07	-0,09	-0,05	0,18	1	0
25	-0,38	-0,47	-0,12	0,65	1,85	3,48	1	0
26	-2,19	-1,51	-1,44	-2,00	-3,16	-4,95	0	1
27	-1,16	-1,12	-1,16	-1,29	-1,49	-1,77	0	1
28	-0,80	-0,31	0,02	0,19	0,20	0,05	0	1
29	-1,97	-0,74	-0,04	0,13	-0,22	-1,10	0	1
30	1,67	1,86	1,74	1,31	0,58	-0,46	0	1
31	7,55	4,56	2,35	0,93	0,30	0,46	1	0
32	-0,79	-0,70	-0,69	-0,77	-0,93	-1,17	0	1
33	4,14	1,60	-0,24	-1,38	-1,83	-1,58	1	0
34	-0,91	-1,69	-1,64	-0,75	0,97	3,52	1	0
35	-4,79	-2,04	-0,08	1,09	1,48	1,07	0	1
36	-3,02	-2,26	-1,75	-1,51	-1,51	-1,77	0	1
37	4,36	2,67	1,82	1,82	2,66	4,35	1	0
38	2,23	1,91	2,14	2,92	4,25	6,13	1	0
39	2,47	0,50	-0,56	-0,70	0,06	1,73	1	0
40	0,36	-0,39	-0,59	-0,25	0,63	2,05	1	0
41	-2,04	-0,29	0,53	0,42	-0,61	-2,57	0	1
42	0,00	0,16	-0,13	-0,88	-2,09	-3,76	0	1
43	-2,75	-1,95	-1,40	-1,08	-0,99	-1,15	0	1
44	0,64	0,17	-0,03	0,04	0,37	0,97	1	0
45	-0,78	-1,06	-0,87	-0,22	0,89	2,48	1	0
46	-0,90	-1,00	-0,98	-0,82	-0,54	-0,13	1	0
47	0,33	1,14	1,60	1,71	1,46	0,86	0	1
48	2,20	1,84	1,98	2,62	3,75	5,37	1	0
<b>Мин. зн.</b>	<b>-4,79</b>	<b>-2,71</b>	<b>-2,42</b>	<b>-4,16</b>	<b>-6,79</b>	<b>-10,31</b>	<b>0</b>	<b>0</b>
<b>Макс. зн.</b>	<b>7,55</b>	<b>4,56</b>	<b>2,35</b>	<b>2,92</b>	<b>4,25</b>	<b>6,13</b>	<b>1</b>	<b>1</b>

Требуется разработать нейросетевую модель на основе двухслойного перцептрона, позволяющую классифицировать тип оптимума некоторой функции (минимум или максимум) по последовательно взятым 6 значениям этой функции на произвольном интервале в произвольных точках.

Соответственно, рассматриваемая нейронная сеть должна иметь 6 входных переменных, кодирующих входную последовательность значений

функции, и 2 выходные переменные, позволяющие идентифицировать класс оптимума ( $y_3$  – минимум,  $y_4$  – максимум): единица указывает на принадлежность классу, 0 – на ее отсутствие. При использовании «Правила 2–5» двухслойный перцептрон с указанным количеством входов и выходов, обучающийся на 40 примерах, может содержать один или два скрытых нейрона. Рассмотрим процедуру обучения нейронной сети с двумя нейронами в скрытом слое (рис. 9.1).

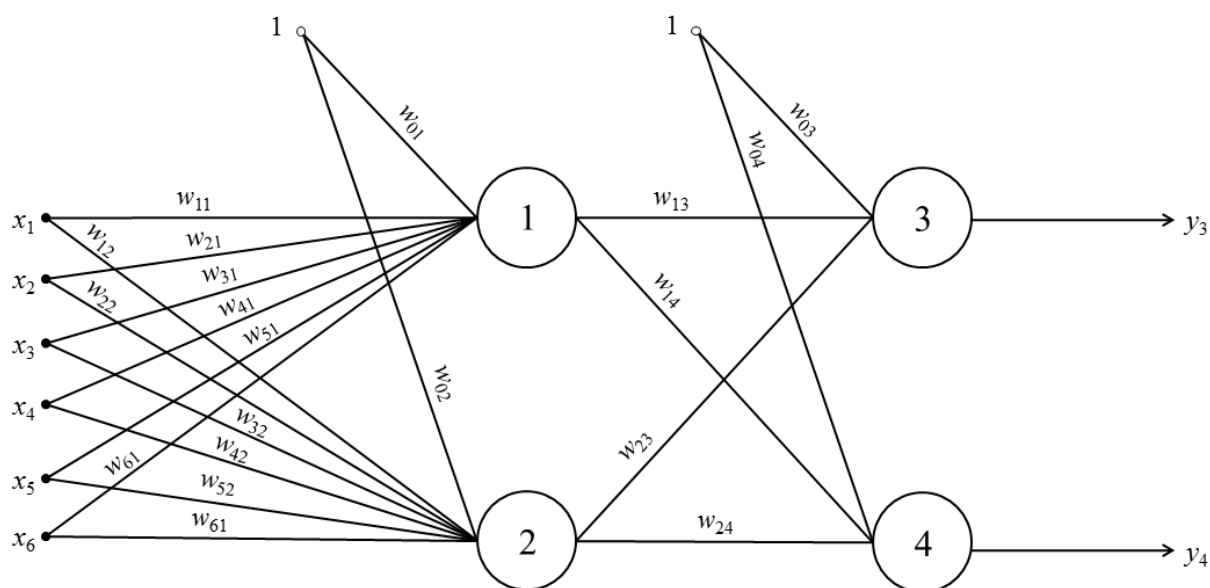


Рис. 9.1. Структура двухслойного перцептрона для решения задачи классификации типа оптимального значения

Выполним линейную нормализацию входных и выходных данных исходной выборки в пределах  $[0, 1]$  с использованием соотношения (3.1). Нормализованные значения, используемые далее для обучения и тестирования двухслойного перцептрона, представлены в табл. 9.2. Курсивом, как и раньше, выделены тестовые примеры.

Выходные значения в обучающей выборке имеют бинарную форму, поэтому в качестве активационной функции выходных нейронов может выступать как функция единичного скачка (рис. 4.1, а, б), так и сигмоидная логистическая функция (рис. 4.1, д) с достаточно большой величиной параметра  $\alpha$ , приближающей ее к скачкообразной форме, как показано на рис. 3.2.

Таблица 9.2

**Нормализованная исходная выборка данных для обучения  
и практического использования двухслойного перцептрона**

№	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$	$\tilde{y}_3^n$	$\tilde{y}_4^n$
1	0,378	0,431	0,616	0,612	0,557	0,512	0	1
2	0,467	0,464	0,606	0,640	0,650	0,658	1	0
3	0,177	0,224	0,457	0,573	0,556	0,512	0	1
4	0,271	0,289	0,421	0,466	0,438	0,402	0	1
5	0,433	0,506	0,740	0,725	0,665	0,618	0	1
6	0,432	0,446	0,627	0,678	0,683	0,682	1	0
7	0,464	0,501	0,740	0,795	0,796	0,793	1	0
8	0,058	0,048	0,201	0,395	0,428	0,411	0	1
9	0,685	0,682	0,870	0,881	0,915	0,957	1	0
10	0,783	0,682	0,568	0,444	0,464	0,526	1	0
11	0,455	0,481	0,667	0,694	0,682	0,673	1	0
12	0,736	0,674	0,637	0,530	0,533	0,574	1	0
13	0,302	0,498	0,904	0,850	0,690	0,554	0	1
14	0,298	0,305	0,411	0,442	0,416	0,386	0	1
15	0,446	0,330	0,329	0,458	0,569	0,650	1	0
16	0,313	0,286	0,417	0,538	0,582	0,598	0	1
17	0,001	0,000	0,243	0,523	0,589	0,582	0	1
18	0,354	0,316	0,264	0,210	0,168	0,142	0	1
19	0,652	0,677	0,843	0,788	0,766	0,770	1	0
20	0,220	0,180	0,308	0,486	0,551	0,571	0	1
21	0,741	0,591	0,453	0,418	0,505	0,608	1	0
22	0,443	0,391	0,516	0,647	0,730	0,784	1	0
23	0,256	0,155	0,000	0,000	0,000	0,000	0	1
24	0,467	0,431	0,522	0,575	0,611	0,638	1	0
25	0,357	0,308	0,482	0,679	0,783	0,839	1	0
26	0,211	0,165	0,205	0,305	0,329	0,326	0	1
27	0,294	0,219	0,264	0,405	0,480	0,519	0	1
28	0,323	0,330	0,512	0,614	0,633	0,630	0	1
29	0,229	0,271	0,499	0,606	0,595	0,560	0	1
30	0,524	0,629	0,872	0,773	0,668	0,599	0	1
31	1,000	1,000	1,000	0,719	0,642	0,655	1	0
32	0,324	0,276	0,363	0,479	0,531	0,556	0	1
33	0,724	0,593	0,457	0,393	0,449	0,531	1	0
34	0,314	0,140	0,164	0,482	0,703	0,841	1	0
35	0,000	0,092	0,491	0,742	0,749	0,692	0	1
36	0,143	0,062	0,140	0,374	0,478	0,519	0	1
37	0,741	0,740	0,889	0,845	0,856	0,892	1	0
38	0,569	0,635	0,956	1,000	1,000	1,000	1	0
39	0,588	0,442	0,390	0,489	0,620	0,732	1	0



Окончание табл. 9.2

40	0,417	0,319	0,384	0,552	0,672	0,752	1	0
41	0,223	0,333	0,618	0,647	0,560	0,471	0	1
42	0,388	0,395	0,480	0,463	0,426	0,398	0	1
43	0,165	0,105	0,214	0,435	0,525	0,557	0	1
44	0,440	0,396	0,501	0,593	0,649	0,686	1	0
45	0,325	0,227	0,325	0,556	0,696	0,778	1	0
46	0,315	0,235	0,302	0,472	0,566	0,619	1	0
47	0,415	0,530	0,843	0,829	0,747	0,679	0	1
48	0,566	0,626	0,922	0,958	0,955	0,954	1	0

Для обучения двухслойного перцептрона выберем метод обратного распространения ошибки, требующий дифференцируемость активационных функций для всей области определения. Следовательно, остановим выбор активационной функции для всех нейронов, включая скрытый, на сигмоидной логистической функции с  $\alpha = 2,0$ , а получаемый аналоговый результат расчета по нейронной сети будем округлять для корректного решения задачи классификации. Коэффициент скорости обучения примем равным 0,5.

В соответствии с алгоритмом метода обратного распространения ошибки зададимся следующими начальными значениями весовых коэффициентов:  $w_{01} = 0,03$ ,  $w_{11} = -0,08$ ,  $w_{21} = -0,06$ ,  $w_{31} = 0,02$ ,  $w_{41} = 0,02$ ,  $w_{51} = -0,08$ ,  $w_{61} = -0,07$ ,  $w_{02} = 0,02$ ,  $w_{12} = -0,05$ ,  $w_{22} = -0,01$ ,  $w_{32} = 0,05$ ,  $w_{42} = -0,04$ ,  $w_{52} = -0,01$ ,  $w_{62} = 0,01$ ,  $w_{03} = 0,04$ ,  $w_{13} = 0,01$ ,  $w_{23} = 0,04$ ,  $w_{04} = 0,07$ ,  $w_{14} = 0,02$ ,  $w_{24} = 0,01$ .

Предъявим сети элементы нормализованного входного вектора первого обучающего примера из табл. 9.2. Расчет состояний и выходных значений нейронов производится с использованием выражений (2.1) и (4.1). В результате получим:  $s_1 = -0,082$ ,  $s_2 = -0,0027$ ,  $\tilde{y}_1^p = 0,459$ ,  $\tilde{y}_2^p = 0,501$ ,  $s_3 = 0,065$ ,  $s_4 = 0,084$ ,  $\tilde{y}_3^p = 0,532$ ,  $\tilde{y}_4^p = 0,542$ .

С использованием выражения (8.1) получим погрешности выходных значений:  $\Delta_3 = -0,532$ ,  $\Delta_4 = 0,458$ .

Зная выходы всех нейронов, вычислим производные активационной функции в точках, соответствующих состояниям каждого нейрона по соотношению (4.2):  $y'_1 = 0,497$ ,  $y'_2 = 0,500$ ,  $y'_3 = 0,498$ ,  $y'_4 = 0,496$ .

Таблица 9.3

## Изменение весовых коэффициентов двухслойного перцептрона

№	W <sub>01</sub>	W <sub>11</sub>	W <sub>21</sub>	W <sub>31</sub>	W <sub>41</sub>	W <sub>51</sub>	W <sub>61</sub>	W <sub>02</sub>	W <sub>12</sub>	W <sub>22</sub>	W <sub>32</sub>	W <sub>42</sub>	W <sub>52</sub>	W <sub>62</sub>	W <sub>03</sub>	W <sub>13</sub>	W <sub>23</sub>	W <sub>04</sub>	W <sub>14</sub>	W <sub>24</sub>
1	0,030	-0,080	-0,060	0,020	0,020	-0,080	-0,070	0,018	-0,051	-0,011	0,049	-0,041	-0,011	0,009	-0,093	-0,056	-0,026	0,184	0,076	0,067
2	0,021	-0,084	-0,064	0,015	0,014	-0,086	-0,076	0,011	-0,054	-0,014	0,045	-0,046	-0,016	0,005	0,046	0,013	0,043	0,037	0,004	-0,006
3	0,021	-0,084	-0,064	0,014	0,014	-0,086	-0,076	0,008	-0,054	-0,015	0,043	-0,047	-0,017	0,003	-0,087	-0,053	-0,024	0,158	0,064	0,054
4	0,026	-0,083	-0,063	0,017	0,017	-0,084	-0,074	0,012	-0,053	-0,014	0,045	-0,046	-0,016	0,004	-0,195	-0,107	-0,078	0,252	0,111	0,101
5	0,035	-0,079	-0,058	0,023	0,023	-0,078	-0,069	0,019	-0,050	-0,010	0,050	-0,041	-0,011	0,009	-0,279	-0,149	-0,120	0,326	0,147	0,138
6	0,013	-0,088	-0,068	0,010	0,008	-0,093	-0,083	0,000	-0,059	-0,019	0,038	-0,053	-0,024	-0,004	-0,132	-0,076	-0,046	0,180	0,075	0,065
7	0,003	-0,093	-0,073	0,002	0,000	-0,101	-0,092	-0,008	-0,062	-0,022	0,032	-0,060	-0,030	-0,010	0,011	-0,007	0,025	0,034	0,004	-0,008
8	0,004	-0,093	-0,073	0,002	0,000	-0,101	-0,091	-0,010	-0,062	-0,023	0,032	-0,061	-0,031	-0,011	-0,116	-0,070	-0,038	0,155	0,064	0,053
9	-0,005	-0,099	-0,079	-0,006	-0,008	-0,109	-0,100	-0,017	-0,067	-0,027	0,026	-0,066	-0,037	-0,017	0,025	-0,005	0,031	0,011	-0,002	-0,018
10	-0,005	-0,099	-0,079	-0,006	-0,008	-0,109	-0,100	-0,014	-0,064	-0,025	0,028	-0,065	-0,036	-0,016	0,145	0,052	0,090	-0,114	-0,061	-0,080
11	0,000	-0,097	-0,077	-0,002	-0,004	-0,105	-0,096	-0,005	-0,061	-0,021	0,033	-0,059	-0,030	-0,010	0,241	0,097	0,137	-0,215	-0,108	-0,129
12	0,007	-0,091	-0,072	0,002	0,000	-0,101	-0,092	0,005	-0,053	-0,014	0,040	-0,054	-0,025	-0,005	0,316	0,132	0,174	-0,294	-0,146	-0,168
13	-0,012	-0,097	-0,081	-0,015	-0,017	-0,115	-0,103	-0,020	-0,061	-0,027	0,017	-0,075	-0,042	-0,018	0,170	0,062	0,102	-0,147	-0,075	-0,095
14	-0,022	-0,100	-0,084	-0,019	-0,021	-0,119	-0,107	-0,034	-0,065	-0,031	0,011	-0,081	-0,048	-0,024	0,024	-0,009	0,029	-0,002	-0,004	-0,023
15	-0,022	-0,100	-0,085	-0,019	-0,021	-0,119	-0,107	-0,031	-0,064	-0,030	0,012	-0,080	-0,046	-0,022	0,145	0,048	0,088	-0,125	-0,062	-0,084
16	-0,030	-0,103	-0,087	-0,023	-0,025	-0,123	-0,111	-0,043	-0,067	-0,033	0,007	-0,086	-0,053	-0,029	0,001	-0,021	0,018	0,018	0,006	-0,013
17	-0,028	-0,103	-0,087	-0,022	-0,024	-0,122	-0,110	-0,045	-0,067	-0,033	0,007	-0,087	-0,054	-0,030	-0,124	-0,081	-0,044	0,141	0,065	0,047
18	-0,021	-0,100	-0,084	-0,020	-0,023	-0,121	-0,109	-0,041	-0,066	-0,032	0,008	-0,086	-0,053	-0,030	-0,224	-0,130	-0,093	0,239	0,113	0,095
19	-0,037	-0,111	-0,095	-0,034	-0,035	-0,133	-0,122	-0,054	-0,074	-0,041	-0,003	-0,097	-0,064	-0,040	-0,076	-0,065	-0,023	0,091	0,048	0,025
20	-0,031	-0,109	-0,094	-0,032	-0,033	-0,130	-0,118	-0,052	-0,074	-0,040	-0,002	-0,096	-0,062	-0,038	-0,186	-0,116	-0,076	0,199	0,099	0,078
21	-0,045	-0,120	-0,102	-0,038	-0,039	-0,137	-0,127	-0,062	-0,082	-0,047	-0,007	-0,100	-0,068	-0,045	-0,040	-0,051	-0,006	0,053	0,033	0,008
22	-0,050	-0,122	-0,104	-0,041	-0,042	-0,141	-0,131	-0,063	-0,082	-0,047	-0,008	-0,101	-0,068	-0,045	0,092	0,006	0,055	-0,080	-0,024	-0,054
23	-0,052	-0,122	-0,105	-0,041	-0,042	-0,141	-0,131	-0,071	-0,084	-0,048	-0,008	-0,101	-0,068	-0,045	-0,046	-0,062	-0,013	0,058	0,044	0,014
24	-0,058	-0,125	-0,107	-0,044	-0,045	-0,145	-0,135	-0,072	-0,085	-0,049	-0,008	-0,102	-0,069	-0,047	0,087	-0,004	0,049	-0,076	-0,015	-0,048
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Итог	5,133	-6,131	-1,962	3,254	2,118	-2,005	-5,730	4,251	-5,088	-1,598	2,685	1,606	-1,733	-4,832	1,886	-2,721	-2,494	-1,886	2,712	2,506

По погрешностям и производным активационной функции нейронов выходного слоя рассчитаем их невязки по соотношению (9.1):  $\delta_3 = -0,265$ ,  $\delta_4 = 0,227$ , а с учетом этих значений по соотношению (9.2) – невязки скрытых нейронов:  $\delta_1 = 0,0009$ ,  $\delta_2 = -0,0042$ .

Зная все невязки, выполним коррекцию весовых коэффициентов выходных нейронов по соотношениям (9.3) и (9.4) и скрытых нейронов по соотношениям (9.3) и (9.5). Новые значения весов записаны в первой строке данных в табл. 9.3.

В целом, в табл. 9.3 показано изменение весовых коэффициентов в ходе их коррекции при последовательном предъявлении 24 обучающих примеров первой эпохи обучения. В последней строке показаны значения весовых коэффициентов по окончании 100-й эпохи обучения.

Для обученной нейронной сети проведено тестирование на обучающей и тестовой выборках. Из 40 примеров обучающей выборки правильно оказались распознаны 38 примеров, что соответствует 95 % случаев верной классификации. Ошибки классификации выявлены для примеров №№ 6 и 11. Анализ входных векторов этих примеров показывает, что оптимальное значение в них выражено крайне слабо (отличается на 0,01 от одной из соседних точек) и находится близко к левой или правой границам. Учитывая эти особенности примеров с ошибками, а также ограниченный объем использованной выборки и простоту структуры двухслойного перцептрона, следует признать полученный результат тестирования очень хорошим.

Для тестовой выборки получены правильные результаты классификации в 7 случаях из 8 (табл. 9.4), что составляет 87,5 %.

Таблица 9.4

**Результаты классификации тестовых примеров**

№ примера	$\tilde{y}_3^п$	$\tilde{y}_4^п$	$\tilde{y}_3^р$	$\tilde{y}_4^р$	Корректность классификации
41	0	1	0	1	+
42	0	1	0	1	+
43	0	1	0	1	+
44	1	0	1	0	+
45	1	0	1	0	+
46	1	0	0	1	–
47	0	1	0	1	+
48	1	0	1	0	+

Единственная ошибка для примера № 46 не может объясняться теми же причинами, что для ошибок обучающей выборки. В данном случае очевидная причина – отсутствие примеров с близким характером и диапазоном изменения значений входных переменных в обучающей выборке. Будь обучающая выборка более объемной, этой ошибки, скорее всего, удалось бы избежать.

## 10. Нейронные сети на основе радиально-симметричных функций

Искусственные нейронные сети на основе радиально-симметричных (радиально-базисных) функций могут использоваться для решения широкого круга задач, среди которых наиболее частые – аппроксимация, классификация и кластеризация данных.

Основное свойство радиально-симметричных функций – это монотонное и симметричное относительно некоторой вертикальной оси симметрии изменение (убывание или возрастание) их откликов.

В качестве примера такой функции может служить рассмотренное ранее выражение функции Гаусса (4.3). Именно эта функция наиболее часто используется в рассматриваемой архитектуре нейронных сетей, однако главным образом, в ее многомерном случае:

$$\vec{h}(x) = \exp\left(-\alpha \cdot \|x - \vec{c}\|^2\right), \quad (10.1)$$

где  $\vec{c}$  – вектор центров (координат вертикальных осей симметрии) множества радиально-симметричных функций;  $\|x - \vec{c}\|$  – норма вектора отклонений входной переменной от центров радиально-симметричных функций. Параметр  $\alpha$  связан с радиусом рассеяния входных переменных  $r$  и может быть заменен в выражении (10.1) на соответствующее отношение:

$$\alpha = \frac{1}{2r^2}.$$

Норма разности векторов рассчитывается как евклидово расстояние:

$$\|x - \vec{c}\| = \sqrt{(x - c_1)^2 + (x - c_2)^2 + \dots + (x - c_m)^2}.$$

На рис. 10.1 приведена типовая структура искусственной нейронной сети на основе радиально-симметричных функций.

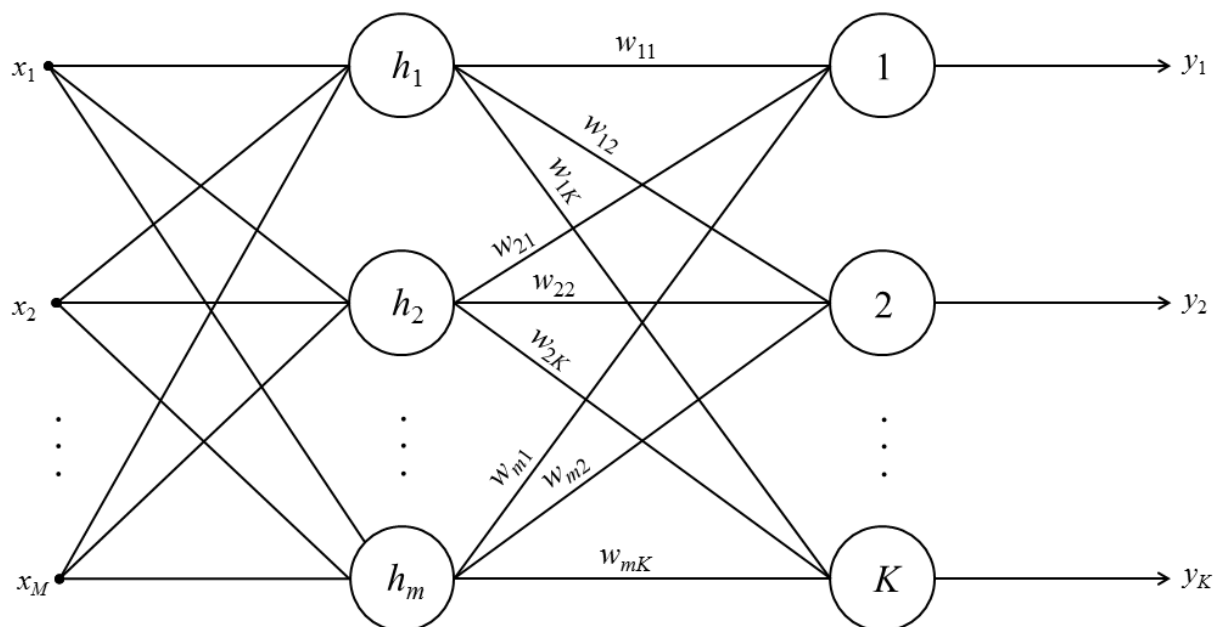


Рис. 10.1. Структура нейронной сети на основе радиально-симметричных функций

Данная структура содержит два слоя нейронов. Выходы первого слоя активируются множеством радиально-симметричных функций (10.1). Фактически они обрабатывают вектор входных значений, определяя степень близости каждого из них к центрам радиально-симметричных функций. Выходы нейронов второго слоя (т. е. выходы всей нейронной сети) – это линейные комбинации выходов первого слоя.

Состав и количество входов и выходов определяются классом решаемой задачи. При аппроксимации данных входы – это аргументы аппроксимирующей зависимости, а выходы – возвращаемые ею значения. При кластеризации или классификации данных входы – это характеристические признаки, по которым различаются объекты, относимые к кластерам или классам, а выходы указывают на соответствующий входам кластер или класс.

Количество скрытых элементов также зависит от решаемой задачи. Если это аппроксимация данных, оно может быть любым. В случае кластеризации или классификации данных должно соответствовать количеству

кластеров или эталонных образов классов.

Жизненный цикл искусственных нейронных сетей на основе радиально-симметричных функций, как и для большинства других архитектур, включает две стадии: обучения и практического использования. В свою очередь, на стадии **обучения** можно выделить также два этапа: настройка нейронной сети и оптимизация синаптических коэффициентов линейного выходного слоя.

На этапе **настройки** рассматриваемой нейронной сети необходимо определить центры  $c$  и радиусы  $r$  радиальных элементов (нейронов скрытого слоя).

1. При наличии небольшого количества эталонных образцов для обучения в качестве центров радиально-симметричных функций следует выбирать соответствующие им вектора. Если объем обучающей выборки достаточно велик, в качестве центров могут быть использованы:

- центры потенциальных кластеров, по которым можно распределить все примеры обучающей выборки вручную или с использованием дополнительных алгоритмов кластеризации, в том числе других архитектур нейронных сетей;

- отдельные случайные примеры обучающей выборки.

Следует заметить, что второй вариант лучше применять при большом количестве нейронов в скрытом слое.

2. Выбор радиусов радиальных элементов определяется требуемым видом радиально-симметричной функции. При больших значениях параметра  $a$  график функции слишком острый, а это значит, что сеть не будет корректно интерполировать данные между известными точками на достаточно большом удалении от них, так как теряет способность к обобщению обучающих данных. Наоборот, при чрезмерно малых значениях параметра  $a$  сеть становится невосприимчивой к отдельным деталям.

С учетом вышесказанного радиусы могут задаваться следующими способами:

- пользователем нейронной сети в явном виде на основе эвристического подбора;

- рассчитываться автоматически по среднему расстоянию до нескольких (в зависимости от общего объема обучающей выборки и количе-

ства скрытых нейронов) ближайших примеров.

На этапе **оптимизации весовых коэффициентов** линейного выходного слоя последовательно выполняются следующие действия.

1. Рассчитывается характеристическая матрица значений радиально-симметричных элементов всех обучающих примеров:

$$\overline{\overline{H}} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_m(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_m(x_2) \\ \cdots & \cdots & \cdots & \cdots \\ h_1(x_N) & h_2(x_N) & \cdots & h_m(x_N) \end{bmatrix}.$$

Количество строк данной матрицы равно количеству примеров обучающей выборки. Количество столбцов – количеству радиальных элементов.

2. Методами линейной алгебры рассчитывается матрица весовых коэффициентов выходного слоя нейронов:

$$\overline{\overline{W}} = \left( \overline{\overline{H}}^T \cdot \overline{\overline{H}} \right)^{-1} \cdot \overline{\overline{H}}^T \cdot \overline{\overline{Y}}, \quad (10.2)$$

где матрица выходов обучающих примеров содержит столбцы в количестве, равном числу обучающих примеров, и строки в количестве, соответствующем числу выходов нейронной сети:

$$\overline{\overline{Y}} = \begin{bmatrix} y_{11} & y_{21} & \cdots & y_{K1} \\ y_{12} & y_{22} & \cdots & y_{K2} \\ \cdots & \cdots & \cdots & \cdots \\ y_{1N} & y_{2N} & \cdots & y_{KN} \end{bmatrix}.$$

Среди преимуществ рассматриваемой архитектуры нейронных сетей выделяют:

- наличие единственного скрытого слоя, достаточного для моделирования ярко выраженных нелинейных зависимостей;
- простота алгоритма оптимизации весовых коэффициентов;
- гарантированное нахождение глобального оптимума функции ошибки при нахождении весовых коэффициентов нейронов выходного слоя;
- высокая скорость обучения.

К ограничениям или недостаткам нейронных сетей на основе ради-

ально-симметричных функций можно отнести:

- необходимость специальной настройки параметров радиально-симметричных функций, сложность настройки при большом количестве скрытых радиальных элементов;

- невозможность экстраполирования модели за пределами исходного интервала изменения входных значений обучающей выборки.

Рассмотрим пример решения задачи аппроксимации данных с использованием искусственной нейронной сети на основе радиально-симметричных функций. Дана экспериментальная зависимость в виде набора из 9 пар точек: значений независимой переменной  $x$  и соответствующих им значений функции отклика  $y$ , представленных в таблице.

**Экспериментальная выборка данных**

№ примера	$x$	$y$
1	-2,0	-0,48
2	-1,5	-0,78
3	-1,0	-0,83
4	-0,5	-0,67
5	0,0	-0,20
6	0,5	0,70
7	1,0	1,48
8	1,5	1,17
9	2,0	0,20

Требуется, используя данную выборку в качестве обучающей, получить аппроксимирующую модель в виде нейронной сети на основе радиально-симметричных функций. Единственный вход данной сети – значение независимой переменной, единственный выход – соответствующее ей значение функции.

Зададимся структурой сети (рис. 10.2), включающей 5 скрытых нейронов (радиальных элементов). В соответствии с имеющимся алгоритмом требуется указать центры и радиусы скрытых радиальных элементов. Используем в качестве центров радиальных элементов значения независимой переменной в опытах 1, 3, 5, 7 и 9, а для каждого элемента примем  $r = 1,5$ , тогда параметр  $\alpha$  будет равен 0,22.



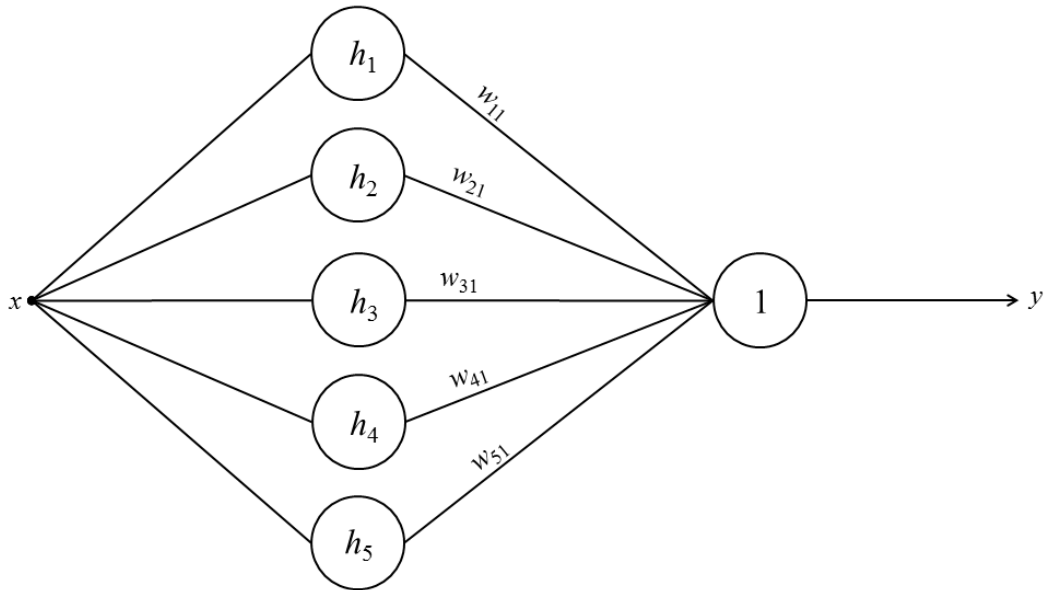


Рис. 10.2. Структура аппроксимирующей нейронной сети

В результате применения принятых настроек нейронов скрытого слоя и соотношения (10.1) получим характеристическую матрицу:

$$\overline{\overline{H}} = \begin{bmatrix} 1,00 & 0,80 & 0,64 & 0,51 & 0,41 \\ 0,89 & 0,89 & 0,72 & 0,57 & 0,46 \\ 0,80 & 1,00 & 0,80 & 0,64 & 0,51 \\ 0,72 & 0,89 & 0,89 & 0,72 & 0,57 \\ 0,64 & 0,80 & 1,00 & 0,80 & 0,64 \\ 0,57 & 0,72 & 0,89 & 0,89 & 0,72 \\ 0,51 & 0,64 & 0,80 & 1,00 & 0,80 \\ 0,46 & 0,57 & 0,72 & 0,89 & 0,89 \\ 0,41 & 0,51 & 0,64 & 0,80 & 1,00 \end{bmatrix}$$

Используем соотношение (10.2) для определения синаптических коэффициентов выходного нейрона. В результате получим следующий вектор-столбец:

$$\overline{\overline{W}} = \begin{bmatrix} 0,59 \\ -2,48 \\ -2,63 \\ 7,17 \\ -2,72 \end{bmatrix}$$

На рис. 10.3 показано положение исходных экспериментальных точек относительно графика полученной нейросетевой аппроксимирующей

зависимости. Как видно, на всем исследуемом диапазоне точки находятся очень близко к линии графика. При использованных настройках нейронной сети средняя относительная ошибка аппроксимации, рассчитанная для экспериментальных точек, составила 3,9 %.

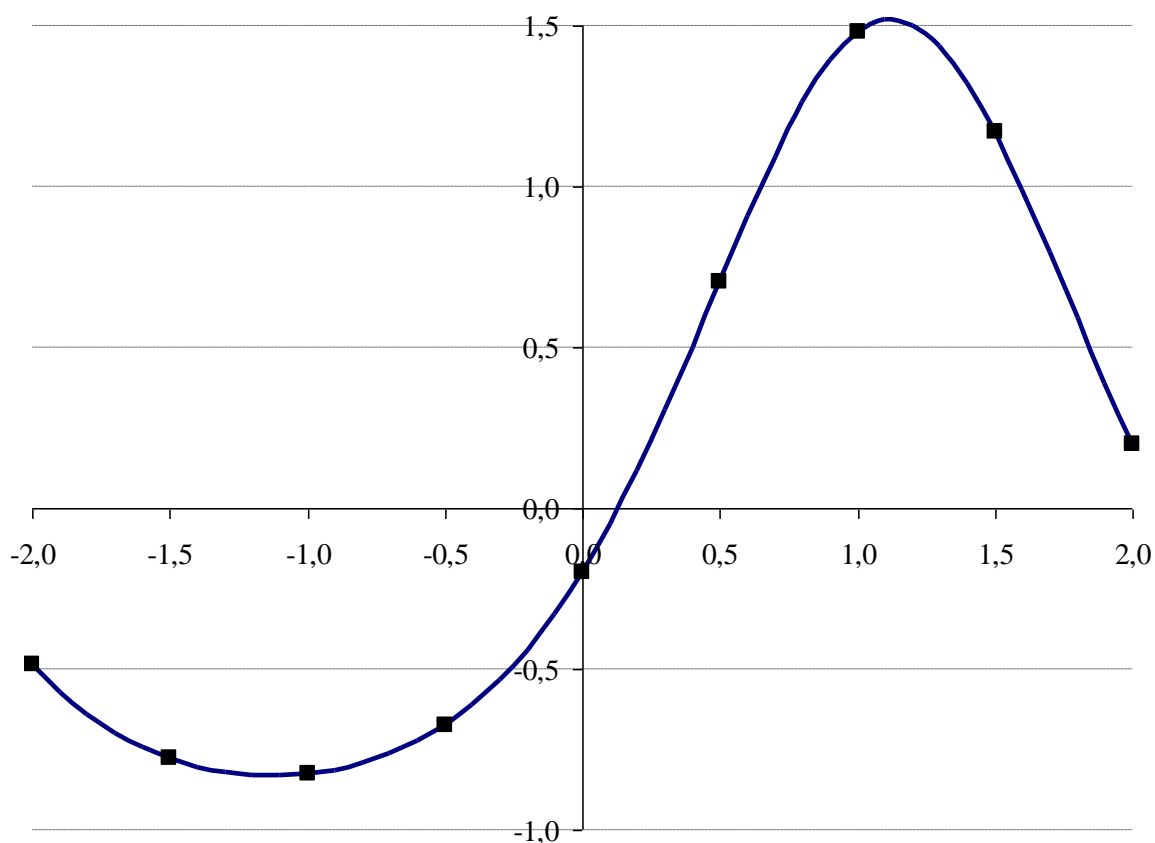


Рис. 10.3. График нейросетевой аппроксимирующей зависимости

Повысить точность решения задачи аппроксимации с использованием искусственной нейронной сети на основе радиально-симметричных функций можно за счет добавления новых радиальных элементов или изменения настроек имеющихся скрытых нейронов. В пределе количество радиальных элементов может совпадать с количеством экспериментальных точек. В этом случае задача сводится к интерполированию экспериментальных данных, в результате чего значения, рассчитываемые нейронной сетью, будут в точности повторять результаты эксперимента в соответствующих точках.

## 11. Нейронные сети Кохонена

Нейронные сети Кохонена – типичный пример нейросетевой архитектуры, обучающейся без учителя. Отсюда и перечень решаемых ими задач: кластеризация данных или прогнозирование свойств. Кроме того, сети Кохонена могут использоваться с целью уменьшения размерности данных с минимальной потерей информации.

Ранее рассмотренные архитектуры нейронных сетей обучались с учителем на выборках данных, включающих множество примеров, состоящих из соответствующих друг другу пар входных и выходных векторов. При этом выходные значения принимали самое непосредственное участие в настройке весовых коэффициентов. В нейронных сетях Кохонена выходные вектора в обучающей выборке могут быть, но могут и отсутствовать, и, в любом случае, они не принимают участия в процессе обучения. То есть выходы не используются в качестве ориентиров при коррекции синопсов. Именно поэтому данный принцип настройки нейронной сети называется самообучением.

В рассматриваемой архитектуре сигнал распространяется от входов к выходам в прямом направлении. Структура нейронной сети содержит единственный слой нейронов (слой Кохонена) без коэффициентов смещения (рис. 11.1). Общее количество весовых коэффициентов рассчитывается как произведение:

$$N_w = MK.$$

Количество нейронов равно количеству кластеров, среди которых происходит начальное распределение и последующее перераспределение обучающих примеров. Количество входных переменных нейронной сети равно числу признаков, характеризующих объект исследования и на основе которых происходит отнесение его к одному из кластеров.

Следует различать собственно самообучение и самоорганизацию нейронной сети Кохонена. При обычном **самообучении** сеть имеет строго фиксированную структуру, т. е. количество нейронов, не изменяющееся на протяжении всего жизненного цикла. При **самоорганизации** сеть, напротив, не имеет постоянной структуры. В зависимости от найденного рассто-

яния до нейрона-победителя либо этот нейрон используется для кластеризации примера, либо для поданного на входы примера создается новый кластер с соответствующими ему весовыми коэффициентами. Кроме того, в процессе самоорганизации структуры сети Кохонена отдельные нейроны могут исключаться из нее.

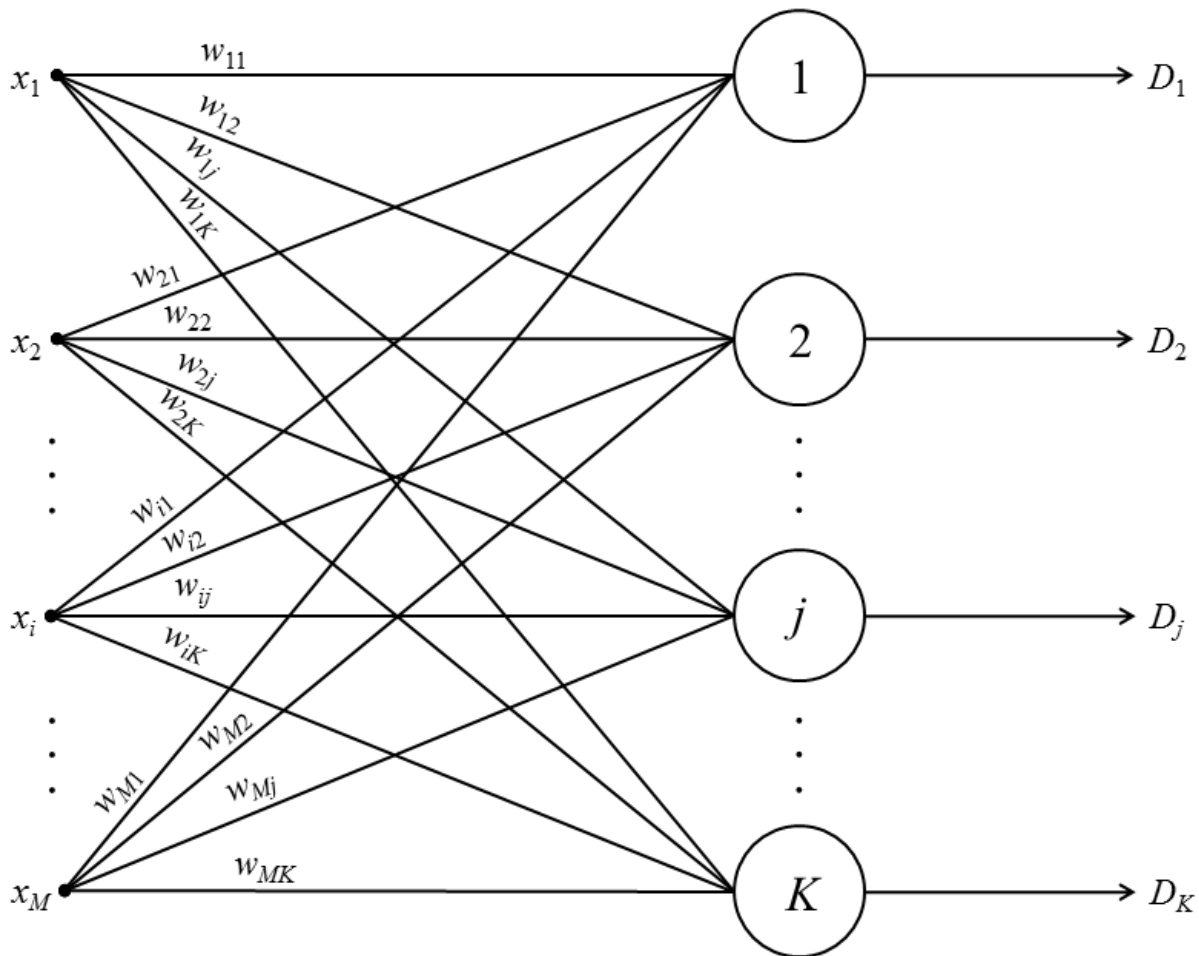


Рис. 11.1. Общая структура нейронной сети Кохонена

Нормализация входных переменных выполняется в пределах  $[-1, 1]$  или  $[0, 1]$ .

Для жизненного цикла нейронных сетей данной архитектуры характерны три основных стадии жизненного цикла: обучение, кластерный анализ и практическое использование.

Алгоритм **обучения** сети Кохонена включает этапы, состав которых зависит от типа структуры: постоянной (самообучающаяся сеть) или переменной (самоорганизующаяся сеть). Для **самообучения** последовательно

выполняются:

1. Задание структуры сети (количества нейронов слоя Кохонена) ( $K$ ).
2. Случайная инициализация весовых коэффициентов значениями, удовлетворяющими одному из следующих ограничений:

– при нормализации исходной выборки в пределах  $[-1, 1]$ :

$$|w_{ij}| \leq \frac{1}{\sqrt{M}}; \quad (11.1)$$

– при нормализации исходной выборки в пределах  $[0, 1]$ :

$$0,5 - \frac{1}{\sqrt{M}} \leq w_{ij} \leq 0,5 + \frac{1}{\sqrt{M}}, \quad (11.2)$$

где  $M$  – количество входных переменных сети – характеристических признаков объекта исследования.

3. Подача на входы сети случайного обучающего примера текущей эпохи обучения и расчет евклидовых расстояний от входного вектора до центров всех кластеров:

$$R_j = \sqrt{\sum_{i=1}^M (\tilde{x}_i - w_{ij})^2}. \quad (11.3)$$

4. По наименьшему из значений  $R_j$  выбирается нейрон-победитель  $j$ , в наибольшей степени близкий по значениям с входным вектором. Для выбранного нейрона (и только для него) выполняется коррекция весовых коэффициентов:

$$w_{ij}^{(q+1)} = w_{ij}^{(q)} + v(\tilde{x}_i - w_{ij}^{(q)}), \quad (11.4)$$

где  $v$  – коэффициент скорости обучения.

5. Цикл повторяется с шага 3 до выполнения одного или нескольких условий окончания:

- исчерпано заданное предельное количество эпох обучения;
- не произошло значимого изменения весовых коэффициентов в пределах заданной точности на протяжении последней эпохи обучения;
- исчерпано заданное предельное физическое время обучения.

Коэффициент скорости обучения может задаваться постоянным из пределов  $(0, 1]$  или переменным значением, постепенно уменьшающимся от эпохи к эпохе.

В случае **самоорганизации** сети Кохонена алгоритм претерпевает

определенные изменения:

1. Задается критическое расстояние  $R_{кр}$ , соответствующее максимально допустимому евклидову расстоянию между входами примера и весами нейрона-победителя. Начальная структура не содержит нейронов. При подаче на входы сети самого первого примера обучающей выборки создается первый нейрон с весовыми коэффициентами, равными поданным входным значениям.

2. На входы сети подается новый случайно выбранный пример текущей эпохи обучения, рассчитываются евклидовы расстояния от примера до центра каждого кластера по соотношению (11.3) и определяется нейрон-победитель с наименьшим из них  $R_{min}$ .

3. Если выполняется условие  $R_{min} \leq R_{кр}$ , производится коррекция весовых коэффициентов соответствующего нейрона-победителя по соотношению (11.4), в противном случае в структуру сети добавляется новый нейрон, весовые коэффициенты которого принимаются численно равными входным значениям поданного примера.

4. Процедура повторяется с п. 2. Если на протяжении последней эпохи обучения какие-либо кластеры остались не задействованными, соответствующие нейроны исключаются из структуры сети Кохонена.

5. Вычисления заканчиваются, если выполняется одно из условий, прописанных в алгоритме самообучения сети фиксированной структуры.

Еще одна модификация алгоритмов самообучения и самоорганизации предусматривает коррекцию весовых коэффициентов не только нейрона-победителя, но и всех остальных нейронов. Для этого следует использовать коэффициент скорости обучения, убывающий с увеличением расстояния до центра кластера  $R_j$ :

$$v_j = v_0 \left( 1 - \frac{1}{1 + e^{-\beta(R_j - R_{кр})}} \right),$$

где  $R_{кр}$  – критическое значение расстояния: чем оно меньше, тем более значительны будут корректировки весов самых близких к обучающему примеру кластеров и практически незначимы – весов более-менее удаленных от него;  $\beta$  – параметр, устанавливающий степень нелинейности влияния расстояния на коэффициент скорости;  $v_0$  – базовое (максимально возможное) значение коэффициента скорости на текущей эпохе обучения.

В качестве значения  $R_{кр}$  можно рассчитывать среднее расстояние для каждого кластера при текущем предъявлении обучающего примера. Параметр  $\beta$  рекомендуется выбирать равным  $3,0 \pm 0,5$ .

Как правило, на практике при использовании самоорганизации нейронной сети Кохонена приходится сталкиваться еще с одной проблемой. С одной стороны, какие-то кластеры могут содержать слишком маленькое количество примеров, что приводит к сложностям в последующем обобщении информации. С другой стороны, некоторые кластеры могут оказаться слишком большими, то есть содержать очень много примеров. В этом случае для регулирования размера кластера и решения проблемы его переполненности можно задать в качестве дополнительного параметра предельное число примеров, формирующих кластер ( $N_{пр}$ ). Если в какой-то момент оказывается, что новый пример должен быть отнесен к кластеру, размер которого уже максимален, принимается решение о создании другого кластера, центр которого будет представлять собой вектор переменных одного из  $(N_{пр} + 1)$  примеров кластера (включая новый) наиболее удаленного от центра данного кластера.

К обученной нейронной сети применяется процедура **кластерного анализа** – процедуры описания свойств кластера на основе анализа количественного и качественного составов примеров, сформировавших его. Следует учитывать, что описание кластеров может базироваться не только на значениях входных переменных обучающей выборки, но и на значениях переменных, не участвовавших в формировании кластеров. В частности, в описание могут входить данные о средних значениях таких переменных среди всех примеров, сформировавших кластер. Кроме того, целесообразно для каждого кластера иметь данные о среднеквадратическом отклонении или дисперсии по каждой переменной.

При **практическом использовании** нейронной сети Кохонена новый пример подается на ее вход и относится к одному из существующих кластеров, либо делается вывод о невозможности такого отнесения (при большом расстоянии до центра ближайшего кластера). Если выбор кластера состоялся, его описание, полученное в результате кластерного анализа, и соответствующие кластеру решения должны распространяться в том числе на поданный пример.

Практическое использование сети Кохонена облегчается за счет визуализации результатов кластеризации. В результате самообучения (самоорганизации) сети получается набор кластеров, каждый из которых характеризуется своим центром (значениями весовых коэффициентов соответствующего нейрона) и количеством обучающих примеров, сформировавших его. Не составляет никакого труда определить евклидово расстояние между центрами всех возможных пар кластеров и графически изобразить их на так называемой **карте Кохонена** – двумерной графической структуре, позволяющей судить не только о размерах и положении каждого отдельно взятого кластера, но и о близости друг к другу и взаимном расположении отдельных кластеров.

Приведем пример самообучения нейронной сети Кохонена для решения задачи кластеризации данных об успеваемости студентов одной из студенческих учебных групп. Исходная выборка представлена в табл. 11.1.

Таблица 11.1

**Исходная выборка данных об успеваемости студентов**

№ примера	Фамилия	Пол $x_1$	Получ. все зачеты $x_2$	Рейтинг по дисциплинам:					Коэфф. стипендии $x_8$
				история $x_3$	инж. графика $x_4$	матем. $x_5$	орг. химия $x_6$	физика $x_7$	
1	Варданын	М	Да	60	79	60	72	63	1,00
2	Горбунов	М	Нет	60	61	30	5	17	0,00
3	Гуменюк	Ж	Нет	60	61	30	66	58	0,00
4	Егоров	М	Да	85	78	72	70	85	1,25
5	Захарова	Ж	Да	65	78	60	67	65	1,00
6	Иванова	Ж	Да	60	78	77	81	60	1,25
7	Ишонина	Ж	Да	55	79	56	69	72	0,00
8	Климчук	М	Нет	55	56	50	56	60	0,00
9	Лисовский	М	Нет	55	60	21	64	50	0,00
10	Нетребя	М	Нет	60	56	30	16	17	0,00
11	Остапова	Ж	Да	85	89	85	92	85	1,75
12	Пашкова	Ж	Да	60	88	76	66	60	1,25
13	Попов	М	Нет	55	64	0	9	50	0,00
14	Сазон	Ж	Да	80	83	62	72	72	1,25
15	Степonenко	М	Нет	55	10	3	8	50	0,00
16	Терентьева	Ж	Да	60	67	57	64	50	0,00
17	Титов	М	Да	75	98	86	82	85	1,50
18	Чернова	Ж	Да	85	85	81	85	72	1,25
19	Четкин	М	Да	80	56	50	69	50	0,00
20	Шевченко	М	Нет	55	60	30	8	60	0,00



Распределение примеров должно осуществляться строго по 4 кластерам. В качестве входных переменных используем  $x_1 - x_7$ , переменная  $x_8$  не будет использоваться для обучения, однако информация о ее значениях будет задействована в ходе кластерного анализа. Таким образом, структурно сеть будет состоять из единственного слоя нейронов, имеющего 7 входов и 4 выхода (рис. 11.2).

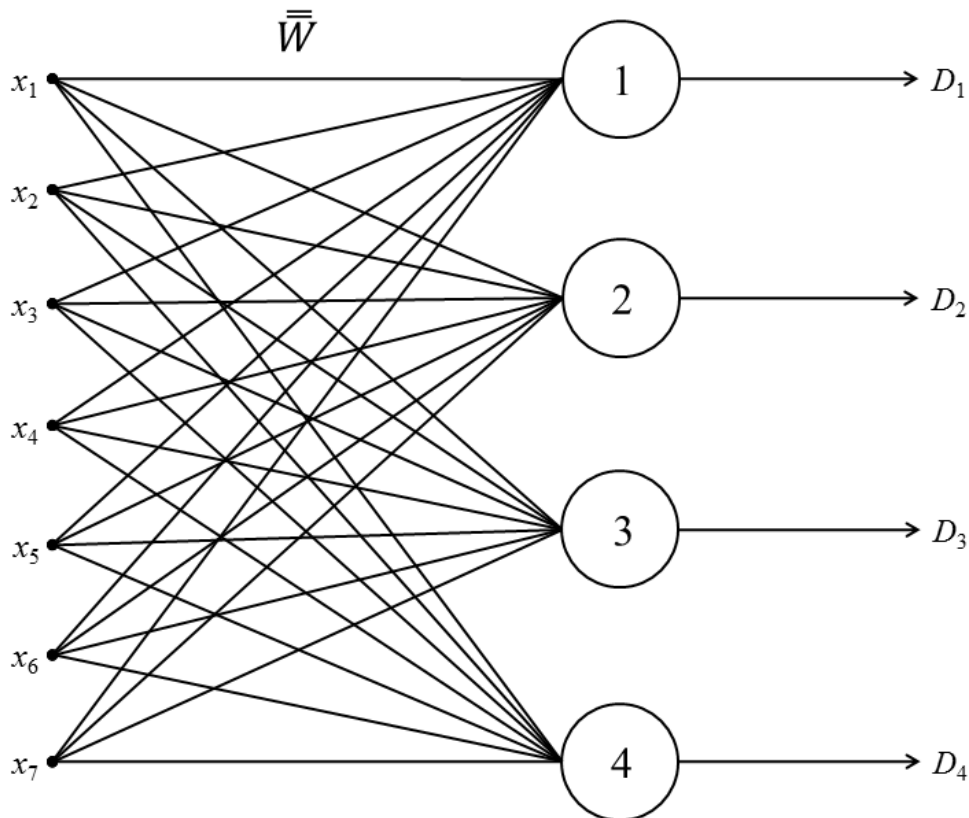


Рис. 11.2. Структура нейронной сети Кохонена для кластеризации студентов

Выполним линейную нормализацию аналоговых значений входных переменных выборки в пределах  $[0, 1]$  по соотношению (3.1). Дискретные значения опишем следующим образом:

- пол студента: 0 – женский, 1 – мужской;
- наличие всех зачетов: 0 – нет, 1 – да.

Результаты нормализации представлены в табл. 11.2.

Проинициализируем все 28 весовых коэффициентов нейронной сети значениями, представленными в табл. 11.3, с учетом ограничения (11.2).

Таблица 11.2

**Нормализованная выборка данных об успеваемости студентов**

№ примера	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$	$\tilde{x}_7$
1	1,00	1,00	0,17	0,78	0,70	0,77	0,68
2	1,00	0,00	0,17	0,58	0,35	0,00	0,00
3	0,00	0,00	0,17	0,58	0,35	0,70	0,60
4	1,00	1,00	1,00	0,77	0,84	0,75	1,00
5	0,00	1,00	0,33	0,77	0,70	0,71	0,71
6	0,00	1,00	0,17	0,77	0,90	0,87	0,63
7	0,00	1,00	0,00	0,78	0,65	0,74	0,81
8	1,00	0,00	0,00	0,52	0,58	0,59	0,63
9	1,00	0,00	0,00	0,57	0,24	0,68	0,49
10	1,00	0,00	0,17	0,52	0,35	0,13	0,00
11	0,00	1,00	1,00	0,90	0,99	1,00	1,00
12	0,00	1,00	0,17	0,89	0,88	0,70	0,63
13	1,00	0,00	0,00	0,61	0,00	0,05	0,49
14	0,00	1,00	0,83	0,83	0,72	0,77	0,81
15	1,00	0,00	0,00	0,00	0,03	0,03	0,49
16	0,00	1,00	0,17	0,65	0,66	0,68	0,49
17	1,00	1,00	0,67	1,00	1,00	0,89	1,00
18	0,00	1,00	1,00	0,85	0,94	0,92	0,81
19	1,00	1,00	0,83	0,52	0,58	0,74	0,49
20	1,00	0,00	0,00	0,57	0,35	0,03	0,63

Таблица 11.3

**Начальные значения весовых коэффициентов сети Кохонена**

№ кластера $j$	Весовые коэффициенты $w_{ij}$						
	$w_{1j}$	$w_{2j}$	$w_{3j}$	$w_{4j}$	$w_{5j}$	$w_{6j}$	$w_{7j}$
1	0,20	0,20	0,30	0,40	0,40	0,20	0,50
2	0,20	0,80	0,70	0,80	0,70	0,70	0,80
3	0,80	0,20	0,50	0,50	0,40	0,40	0,40
4	0,80	0,80	0,60	0,70	0,70	0,60	0,70

Алгоритм самообучения настроим следующим образом. Выберем начальный коэффициент скорости обучения, равный 0,30, уменьшающийся с каждой эпохой на 0,05. Таким образом, будет выполнено 6 эпох обучения с различным коэффициентом скорости, на каждой из которых будет 20 корректировок весов одного из нейронов.

Подадим на вход нейронной сети случайно выбранный нормализованный пример № 10. В соответствии с выражением (11.3) рассчитаем расстояния до текущих центров четырех кластеров. Они соответственно будут равны 0,98, 1,65, 0,65 и 1,32. Наименьшее расстояние соответствует третьему кластеру, из чего делаем вывод, что третий нейрон – нейрон-победитель и именно его веса должны быть скорректированы по соотношению (11.4). Новые значения весовых коэффициентов составят:  $w_{13} = 0,86$ ,  $w_{23} = 0,14$ ,  $w_{33} = 0,40$ ,  $w_{43} = 0,51$ ,  $w_{53} = 0,39$ ,  $w_{63} = 0,32$ ,  $w_{73} = 0,28$ . Веса остальных нейронов при этом не изменяются.

Далее аналогичным образом на входы сети предъявляются остальные примеры выборки в случайной последовательности. После предъявления всех 20 примеров начинается следующая эпоха обучения, при этом коэффициент скорости уменьшаем на 0,05. В результате полного цикла обучения сети Кохонена получаем итоговые значения весов, приведенные в табл. 11.4.

Таблица 11.4

**Итоговые значения весовых коэффициентов нейронной сети Кохонена**

№ кластера $j$	Весовые коэффициенты $w_{ij}$						
	$w_{1j}$	$w_{2j}$	$w_{3j}$	$w_{4j}$	$w_{5j}$	$w_{6j}$	$w_{7j}$
1	0,06	0,06	0,21	0,52	0,36	0,55	0,57
2	0,00	1,00	0,50	0,80	0,80	0,80	0,73
3	1,00	0,00	0,04	0,48	0,26	0,22	0,42
4	1,00	0,99	0,69	0,77	0,79	0,78	0,81

Для обученной нейронной сети выполним процедуру кластерного анализа. Все четыре кластера оказались заполнены. Однако количественный состав их разный. В 1-й кластер попал единственный пример – № 3. Во 2-м, самом объемном кластере оказались примеры №№ 5–7, 11, 12, 14, 16, 18. В 3-й кластер вошли примеры №№ 2, 8–10, 13, 15, 20. В 4-й – №№ 1, 4, 17, 19. Качественные данные и описания кластеров, полученные в результате анализа, представлены в табл. 11.5.

Выводы о факте получения стипендии в описаниях кластеров сделаны на основе анализа значений переменной  $x_8$ , не участвовавшей в процессе обучения.

Таблица 11.5

**Результаты кластерного анализа**

№ кластера	Размер кластера	Пол	Получ. все зачеты	Средний рейтинг	Коэфф. стипендии	Описание
1	1	Ж	Нет	55	0,00	Удовлетворительно успевающие студенты-девушки, не имеющие одного или нескольких зачетов и не получающие стипендию
2	8	Ж	Да	72	0,97	Хорошо успевающие студенты-девушки, имеющие все зачеты и в большинстве своем получающие стипендию
3	7	М	Нет	40	0,00	Неуспевающие студенты-юноши, не имеющие одного или нескольких зачетов и не получающие стипендию
4	4	М	Да	73	0,94	Хорошо успевающие студенты-юноши, имеющие все зачеты и в большинстве своем получающие стипендию

Полученные описания и обученную нейронную сеть можно использовать для получения характеристики любого другого студента, для которого имеется тот же набор входных данных. Следует заметить, что на стадии практического использования могут встречаться данные, выходящие за ранее использованные пределы нормализации, что, однако, не должно повлиять на выбор нейрона-победителя и, как следствие, на отнесение нового примера к соответствующему кластеру.

## 12. Нейронные сети адаптивного резонанса

Человеческий мозг непрерывно занимается обработкой информации, поступающей от органов чувств: звуки, запахи, изображения и т. д. Большая часть такой информации не представляет интереса или неважна и

поэтому игнорируется. Однако та небольшая ее составляющая, что имеет ценность, должна быть не только обработана, но и сохранена в долговременной памяти. Важно понимать, что при этом не только новые образы должны запоминаться, но также должны идентифицироваться ранее встречавшиеся. В свою очередь, запоминание новых образов не должно влиять на образы, хранящиеся в памяти.

Приведенные особенности работы человеческого мозга отражены в теории адаптивного резонанса. Несмотря на существование в настоящее время сравнительно большого количества вариантов архитектур искусственных нейронных сетей адаптивного резонанса, основными остаются две:

- АРТ-1 (от англ. ART – Adaptive Resonance Theory) – для кластеризации, хранения и идентификации образов в форме двоичных сигналов;
- АРТ-2 – для кластеризации, хранения и идентификации образов, представленных как в форме двоичных сигналов, так и в форме аналоговых сигналов, в том числе с использованием обоих типов сигналов в одной структуре.

В отличие от большинства существующих архитектур нейронных сетей АРТ-сети не предусматривают строгого деления жизненного цикла на стадии обучения и практического использования. Они продолжают обучаться на протяжении всего времени их практического использования, включая этап практической эксплуатации.

Структура любой АРТ-сети содержит единственный слой нейронов. Количество входных переменных равно количеству двоичных или вещественных признаков, характеризующих объект. Количество выходов постоянно. В момент начала функционирования сети выходов нет вовсе. Постепенно их количество возрастает с каждым новым незнакомым входным образом, образуя, по сути, новый кластер. Таким образом, очевидно, что АРТ-сети – самоорганизующиеся.

Рассмотрим структуру и алгоритм работы **нейронной сети АРТ-1** (рис. 12.1).

В отличие от большинства структур других нейронных сетей входы АРТ-1 связаны с выходами не одним, а парой синапсов:  $w_{ij}$  – синапсы кратковременной памяти;  $t_{ij}$  – синапсы долговременной памяти. Назначе-

ние кратковременной памяти – установление кластеров, к которым может быть отнесен входной образ. Назначение долговременной памяти – установление степени соответствия входного образа кластерам, определение нейрона-победителя или принятие решения о создании нового кластера.

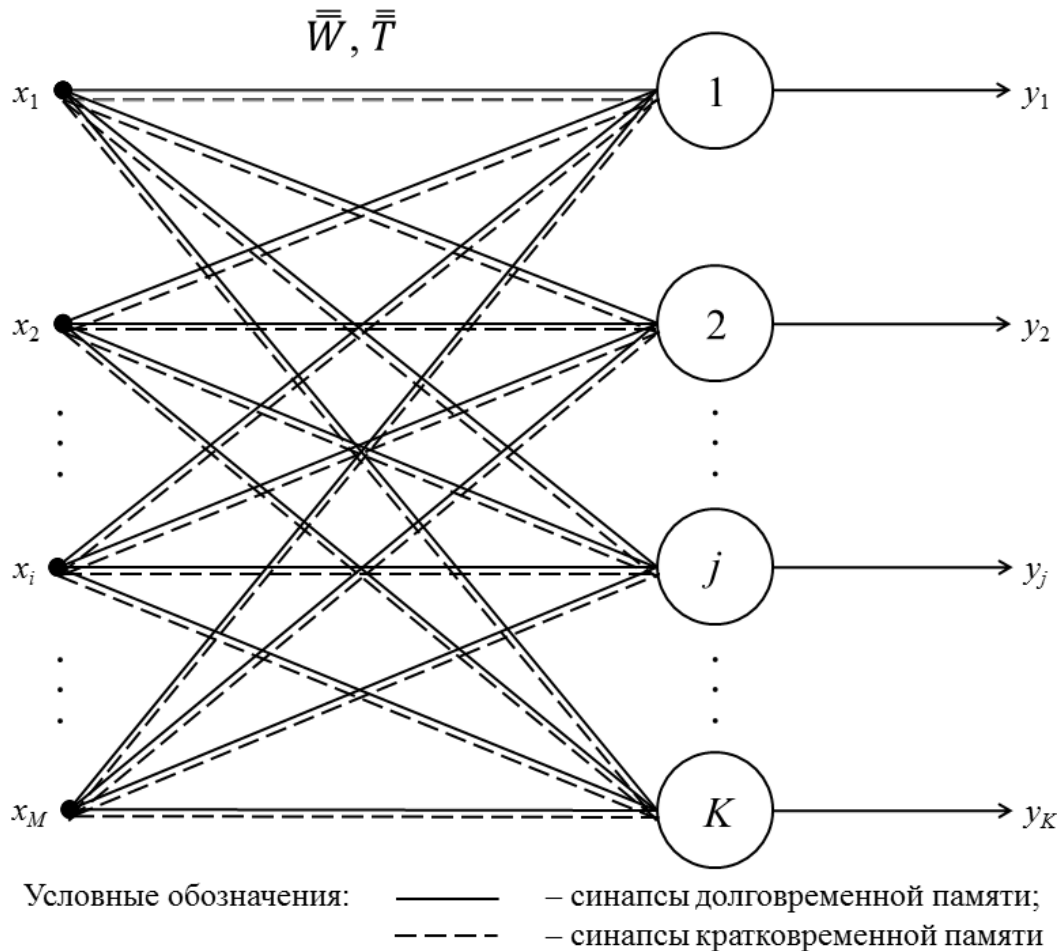


Рис. 12.1. Структура нейронной сети АРТ-1

Поскольку АРТ-1 работает с бинарными входами (0 или 1), никакая дополнительная нормализация или масштабирование входных данных не требуются.

Алгоритм функционирования АРТ-1 включает 3 стадии жизненного цикла: инициализацию сети, распознавание образа, кластеризацию образа (сравнение). На стадии **инициализации** последовательно выполняются следующие этапы:

1.1. Устанавливается параметр сходства  $R_{кр}$  ( $0 < R_{кр} < 1$ ), причем чем

он больше, тем выше должно быть сходство образа и кластера для отнесения к последнему.

1.2. Для первого поданного на входы сети образа создается первый нейрон (кластер), значения синаптических связей которого устанавливаются согласно следующим соотношениям:

$$w_{i1} = \frac{\lambda \cdot x_i}{\lambda - 1 + \sum_{p=1}^M x_p}, \quad (12.1)$$

$$t_{i1} = x_i, \quad (12.2)$$

где  $\lambda$  – положительная ( $> 1,0$ ) константа, определяющая степень влияния нового входного образа на кратковременную память. Чаще всего  $\lambda$  принимается равной 2,0.

Стадия **распознавания образа** включает следующие этапы:

2.1. На входы сети подается новый входной образ. Для каждого нейрона рассчитывается значение его выхода с учетом коэффициентов кратковременной памяти:

$$y_j = \sum_{i=1}^n w_{ij} x_i. \quad (12.3)$$

2.2. Положительные выходы нейронов указывают на кластеры, имеющие качественное сходство с входным образом. Если же все выходы оказались нулевыми, входной образ не соответствует ни одному из кластеров, создается новый нейрон с синаптическими коэффициентами, рассчитываемыми по соотношениям (12.1) и (12.2), после чего алгоритм продолжает работу с п. 2.1.

Количественное сходство входного образа с кластерами определяется на стадии **кластеризации образа (сравнения)**:

3.1. Рассчитывается количественная мера сходства входного образа с кластером, имеющим наибольшее значение выхода нейрона в п. 2.1:

$$R_j = \frac{\sum_{i=1}^M t_{ij} x_i}{\sum_{i=1}^M x_i}. \quad (12.4)$$

3.2. Если выполняется условие  $R_j > R_{кр}$ ,  $j$ -й нейрон считается нейрон-победителем, а входной образ – соответствующим  $j$ -му кластеру. В

этом случае для нейрона выполняется пересчет весовых коэффициентов по соотношениям:

$$w_{ij}^{(q+1)} = (1 - \nu) \cdot w_{ij}^{(q)} + \nu \frac{\lambda \cdot x_i}{\lambda - 1 + \sum_{p=1}^M x_p}, \quad (12.5)$$

$$t_{ij}^{(q+1)} = (1 - \nu) \cdot t_{ij}^{(q)} + \nu \cdot x_i, \quad (12.6)$$

где  $\nu$  – коэффициент скорости адаптации ( $0 < \nu < 1$ ). Большее значение  $\nu$  соответствует быстрой адаптации, меньшее – медленной.

3.3. Если условие  $R_j > R_{кр}$  не выполняется, алгоритм повторяется с п. 3.1 для остальных кластеров, следующие в порядке убывания их выходных значений, рассчитанных в п. 2.1, до тех пор, пока не будет получен нейрон-победитель или не будут использованы все кластеры с положительным выходом нейрона.

3.4. Если нейрон-победитель не найден, создается новый кластер с синаптическими коэффициентами нейрона, рассчитанными по соотношениям (12.1) и (12.2), после чего алгоритм продолжает работу с п. 2.1.

Рассмотрим пример работы сети АРТ-1 на протяжении всего жизненного цикла. Последовательно подадим на ее входы 4 бинарных вектора, состоящих из 9 элементов, кодирующих различные образы. Графическая и цифровая интерпретации образов приведены на рис. 12.2.

Очевидно, что структура нейронной сети для кластеризации указанных образов будет включать 9 входов. Выберем следующие параметры настройки нейронной сети:  $R_{кр} = 0,7$ ,  $\lambda = 2,0$ ,  $\nu = 0,6$ . Проведем расчеты в соответствии с алгоритмом работы АРТ-1.

На стадии инициализации подаем на входы сети образ 1. Он формирует первый нейрон, весовые коэффициенты которого рассчитываются по соотношениям (12.1), (12.2):  $w_{11} = w_{31} = w_{51} = w_{71} = w_{91} = 0,33$ ;  $w_{21} = w_{41} = w_{61} = w_{81} = 0,00$ ;  $t_{11} = t_{31} = t_{51} = t_{71} = t_{91} = 1,00$ ;  $t_{21} = t_{41} = t_{61} = t_{81} = 0,00$ .

Далее работа с каждым образом начинается со стадии распознавания образа. Подаем на входы сети образ 2. Выходное значение единственного существующего на данный момент нейрона, рассчитанное по соотношению (12.3), составит:  $y_1 = 0,00$ . Таким образом, образ 2 оказался не соот-



ветствующим первому кластеру, и для него создается новый нейрон с весовыми коэффициентами, рассчитанными по соотношениям (12.1), (12.2):

$$w_{12} = w_{32} = w_{52} = w_{72} = w_{92} = 0,00; \quad w_{22} = w_{42} = w_{62} = w_{82} = 0,40;$$

$$t_{12} = t_{32} = t_{52} = t_{72} = t_{92} = 0,00; \quad t_{22} = t_{42} = t_{62} = t_{82} = 1,00.$$

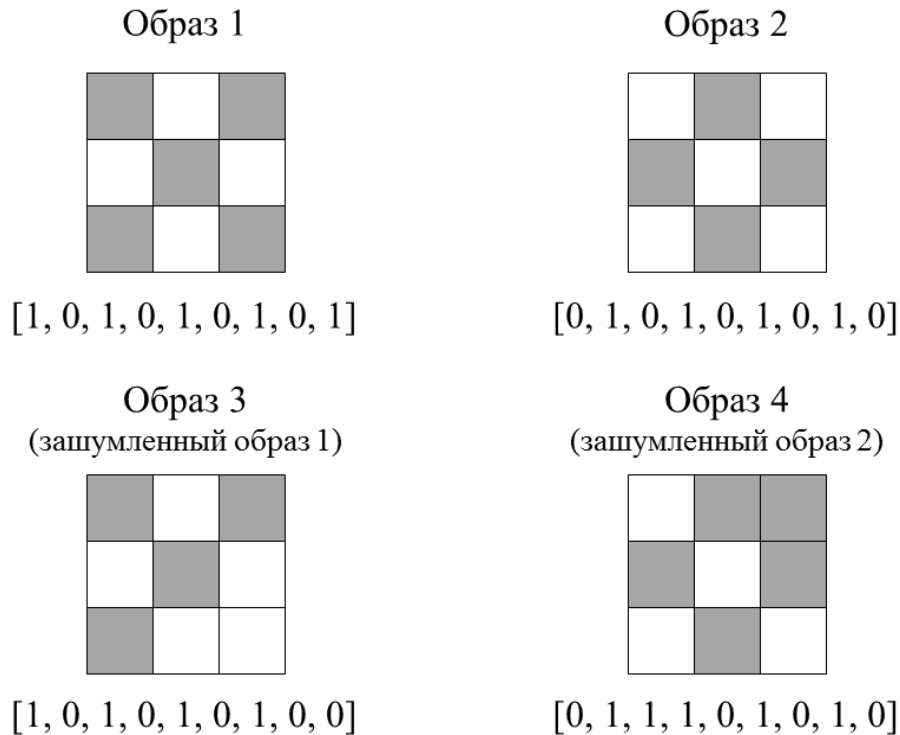


Рис. 12.2. Набор входных образов для кластеризации с помощью сети АРТ-1

Подает на входы нейронной сети образ 3. Выходные значения имеющих двух нейронов равняются соответственно:  $y_1 = 1,33$ ,  $y_2 = 0,00$ . Очевидно наличие возможного соответствия только образу, ранее сформировавшему первый кластер. Проверим сходство по критерию (12.4). В результате расчета  $R_1 = 1,0$ , что больше заданного значения  $R_{кр}$ . Таким образом, соответствие образа 3 первому кластеру считаем установленным. В соответствии с алгоритмом стадии кластеризации образа далее по соотношениям (12.5) и (12.6) пересчитываются коэффициенты кратковременной и долговременной памяти первого нейрона. Их новые значения составят:

$$w_{11} = w_{31} = w_{51} = w_{71} = 0,37; \quad w_{91} = 0,13; \quad w_{21} = w_{41} = w_{61} = w_{81} = 0,00;$$

$$t_{11} = t_{31} = t_{51} = t_{71} = 1,00; \quad t_{91} = 0,40; \quad t_{21} = t_{41} = t_{61} = t_{81} = 0,00.$$

Подает на входы нейронной сети образ 4. Выходные значения нейронов составят соответственно:  $y_1 = 0,37$ ,  $y_2 = 1,60$ . Это говорит о том,

что в разной степени возможно сходство с обоими образами, однако со вторым оно гораздо вероятнее. Следовательно, по соотношению (12.4) рассчитываем значение критерия:  $R_2 = 0,8$ . Оно больше заданного критического значения, а значит образ 4 должен быть отнесен ко второму кластеру. По соотношениям (12.5) и (12.6) пересчитываются синаптические коэффициенты второго нейрона, в результате чего получим:  $w_{12} = w_{52} = w_{72} = w_{92} = 0,00$ ;  $w_{32} = 0,20$ ;  $w_{22} = w_{42} = w_{62} = w_{82} = 0,36$ ;  $t_{12} = t_{52} = t_{72} = t_{92} = 0,00$ ;  $t_{32} = 0,60$ ;  $t_{22} = t_{42} = t_{62} = t_{82} = 1,00$ .

Жизненный цикл нейронной сети продолжается далее в соответствии с алгоритмами стадий распознавания и кластеризации образа. При наличии множества тестовых образов можно в любой момент оценить их сходство с существующими кластерами и выполнить кластерный анализ полученных групп.

В отличие от предыдущей разновидности архитектуры **нейронная сеть АРТ-2** позволяет использовать как бинарные, так и аналоговые входные сигналы. Алгоритм ее функционирования включает две стадии жизненного цикла: инициализацию и кластеризацию (сравнение) образов.

Стадия **инициализации** состоит из следующих этапов:

1.1. Устанавливается параметр сходства  $R_{кр}$ , аналогичный по физическому смыслу и области допустимых значений такому же параметру архитектуры АРТ-1.

1.2. Для имеющейся выборки данных выполняется нормализация значений переменных в пределах  $[0, 1]$ .

1.3. Производится нормирование элементов каждого входного вектора таким образом, чтобы сумма квадратов этих элементов равнялась 1:

$$\hat{x}_i = \frac{\tilde{x}_i}{\sqrt{\sum_{p=1}^M x_p^2}}. \quad (12.7)$$

1.4. Создается первый нейрон (кластер) с весовыми коэффициентами, численно равными нормированным входным значениям первого примера (образа):

$$w_{i1} = \hat{x}_i.$$

Стадия **кластеризации образа (сравнения)** включает следующие

этапы:

2.1. На входы нейронной сети подается очередной образ и определяется количественная мера его сходства с каждым из имеющихся кластеров:

$$R_j = \sum_{i=1}^M w_{ij} \hat{x}_i. \quad (12.8)$$

2.2. Выбирается кластер с номером  $J$  с максимальным значением меры сходства:  $R_{\max} = \max_{j=1, \overline{K}} (R_j)$ .

2.3. Если условие  $R_{\max} < R_{\text{кр}}$  выполняется, считается, что сходство входного образа ни с одним из кластеров не установлено. В этом случае создается новый нейрон (кластер) с весовыми коэффициентами, равными элементам соответствующего нормированного вектора  $\hat{x}$ .

2.4. Если условие  $R_{\max} \geq R_{\text{кр}}$  не выполняется, считается, что установлено наибольшее сходство входного образа с кластером  $J$ . Тогда весовые коэффициенты соответствующего нейрона пересчитываются по соотношению:

$$w_{ij}^{(q+1)} = (1 - \nu) \cdot w_{ij}^{(q)} + \nu \cdot \hat{x}_i. \quad (12.9)$$

Далее алгоритм продолжает работу на стадии кластеризации с п. 2.1.

Рассмотрим практическое использование нейронной сети АРТ-2 на примере задачи кластеризации данных об успеваемости учебной группы студентов, решенной ранее с помощью нейронной сети Кохонена в главе 11. Нормализованная обучающая выборка приведена в табл. 11.2.

Примем критическое значение параметра сходства  $R_{\text{кр}} = 0,8$  и коэффициент скорости адаптации  $\nu = 0,5$ .

В соответствии с алгоритмом данные в строках обучающей выборки должны быть пронормированы по соотношению (12.7).

Поскольку в момент инициализации сеть не содержит нейронов, самый первый поданный на нее входной образ (пример 1) сформирует этот нейрон и передаст ему значения весовых коэффициентов, равные значениям элементов входного вектора:  $w_{11} = 0,489$ ,  $w_{21} = 0,489$ ,  $w_{31} = 0,082$ ,  $w_{41} = 0,384$ ,  $w_{51} = 0,341$ ,  $w_{61} = 0,377$ ,  $w_{71} = 0,331$ .

Подает на входы пример 2 и по соотношению (12.8) определяем значение критерия сходства с единственным пока кластером. Получаем  $R_1 = 0,693$ . Эта величина меньше критического, следовательно, сходство

оказалось недостаточным для отнесения примера 2 к первому кластеру. Формируем на основе второго примера новый кластер со следующими весовыми коэффициентами:  $w_{12} = 0,821$ ,  $w_{22} = 0,000$ ,  $w_{32} = 0,137$ ,  $w_{42} = 0,476$ ,  $w_{52} = 0,286$ ,  $w_{62} = 0,000$ ,  $w_{72} = 0,000$ .

Подает на входы пример 3 и определяем критерии сходства с обоими созданными ранее кластерами. Получаем  $R_1 = 0,707$ ,  $R_2 = 0,344$ . Таким образом, для примера 3 также не удалось подобрать подходящий кластер, поэтому создаем для него новый с весовыми коэффициентами:  $w_{13} = 0,000$ ,  $w_{23} = 0,000$ ,  $w_{33} = 0,144$ ,  $w_{43} = 0,501$ ,  $w_{53} = 0,301$ ,  $w_{63} = 0,606$ ,  $w_{73} = 0,521$ .

Подает на входы пример 4 и определяем критерии сходства со всеми имеющимися кластерами. Получаем  $R_1 = 0,931$ ,  $R_2 = 0,646$ ,  $R_3 = 0,726$ . Так как  $R_1 > R_{кр}$ , данный пример должен быть отнесен к первому кластеру, для которого по соотношению (12.9) пересчитываем весовые коэффициенты. Получаются следующие их новые значения:  $w_{11} = 0,451$ ,  $w_{21} = 0,451$ ,  $w_{31} = 0,247$ ,  $w_{41} = 0,351$ ,  $w_{51} = 0,344$ ,  $w_{61} = 0,343$ ,  $w_{71} = 0,372$ .

Аналогично выполняется кластеризация для всех остальных примеров обучающей выборки. Результирующие значения весовых коэффициентов представлены в таблице.

### Итоговые значения весовых коэффициентов нейронной сети АРТ-2

№ кластера $j$	Весовые коэффициенты $w_{ij}$						
	$w_{1j}$	$w_{2j}$	$w_{3j}$	$w_{4j}$	$w_{5j}$	$w_{6j}$	$w_{7j}$
1	0,298	0,478	0,375	0,325	0,349	0,379	0,303
2	0,788	0,000	0,011	0,322	0,170	0,052	0,406
3	0,000	0,000	0,144	0,501	0,301	0,606	0,521

В результате полного перебора всех примеров обучающей выборки в нейронной сети самоорганизуются 3 кластера, различающиеся количественным и качественным составами студентов. Первый кластер включает хорошо и отлично успевающих студентов и студентов, не имеющих академических задолженностей (примеры №№ 1, 4–7, 11, 12, 14, 16–19). Второй кластер – неуспевающих студентов с низкими баллами, имеющих академические задолженности и не получающих стипендию (примеры №№ 2, 8–10, 13, 15, 20). Третий кластер содержит единственный пример (№ 3), который сеть АРТ-2 не смогла отнести ни к одному из двух больших кла-

стеров.

Как видно из полученных результатов, сеть не разделяет по отдельным группам студентов, имеющих разный пол. И по другим входным признакам границы более размытые. Отчасти это связано с величиной выбранного критического значения параметра сходства. Но не меньшее влияние оказывает и использование нормирования данных внутри каждого примера.

Следует заметить, что возможно провести настройку нейронной сети путем многократной подачи примеров обучающей выборки на входы с уменьшающимся от эпохи к эпохе коэффициентом скорости адаптации. В этом случае целесообразно каждый раз использовать новую случайную последовательность примеров.

### 13. Автоассоциативная память

Человеческая нервная система получает и обрабатывает огромное количество сигналов, поступающих из окружающего пространства. Если эти сигналы могут быть осмыслены человеком и им могут быть сопоставлены соответствующие образы, принято говорить о работе **ассоциативной памяти**.

Получаемые сигналы имеют различное происхождение и различную форму. Часть их свидетельствует о явлениях или событиях, непосредственно влияющих на жизнедеятельность людей. В таких случаях они, как правило, представляют собой информационный массив, который необходимо правильно интерпретировать, очистив от шума. Таким образом, по сути, решается задача распознавания образа, закодированного массивом, имеющего те же размерность, размеры и физический смысл переменных. Соответствующую категорию ассоциативной памяти называют **автоассоциативной**, и речь о ней пойдет в данной главе.

Значительная часть информации, получаемой человеком в виде сигналов из окружающей среды, не несет для него какого-либо смысла, оставаясь информационным фоном, и, соответственно, человек не придает ей какого-либо значения. Однако возможно совпадение по времени получе-

ния этой и более ценной, жизненно необходимой информации. Так, определенные запахи или музыкальный мотив могут напомнить о событии в жизни человека или о другом человеке. То есть, один конкретный информационный образ, возможно, и зашумленный, непосредственно ассоциируется с другим образом. В этом случае говорят о **гетероассоциативной памяти**, которой посвящена следующая глава.

Типичный пример имитации автоассоциативной памяти в нейроинформатике – нейронная сеть Хопфилда. Структура данной сети содержит единственный слой нейронов. Количества входных и выходных переменных совпадают ( $M = K$ ), а сами переменные должны соответствовать друг другу по физическому смыслу и представляют собой бинарные признаки образов, хранимых в памяти сети. Значения входов и выходов: +1 или –1. Сигналы выходных нейронов направляются по обратным связям ко всем остальным нейронам, поэтому данную сеть можно считать полносвязной. Общая структура нейронной сети Хопфилда изображена на рис. 13.1.

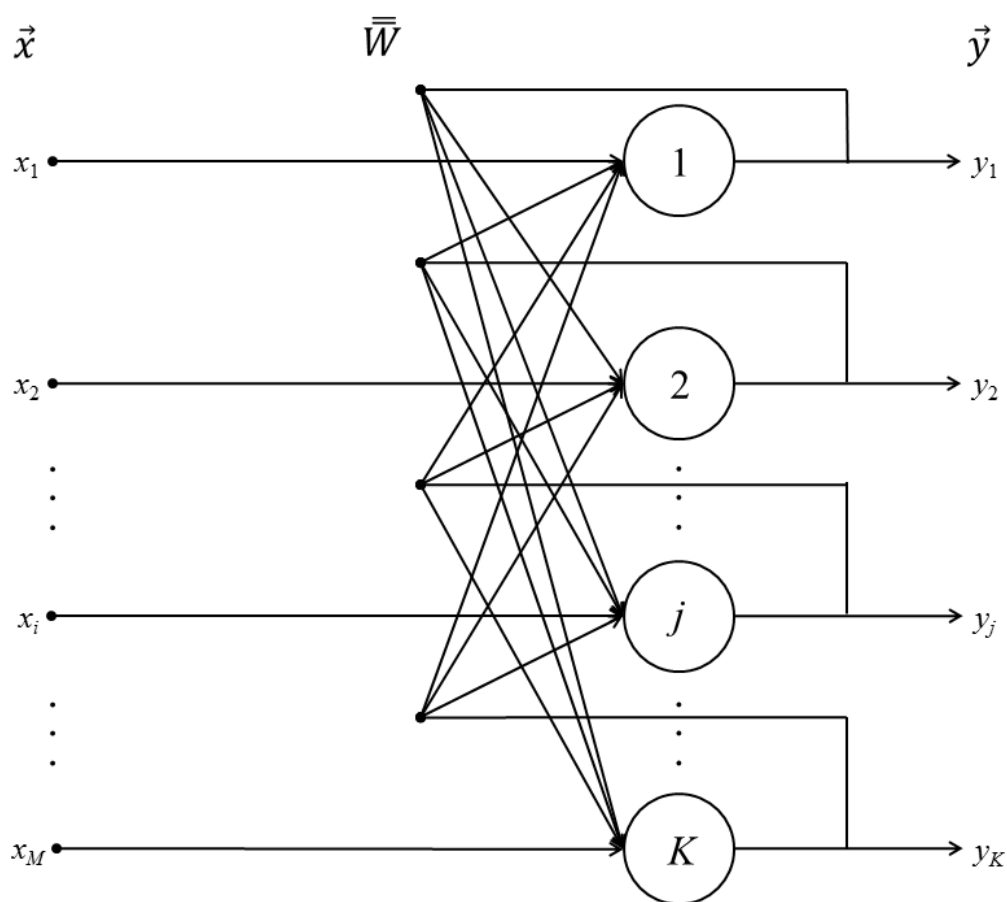


Рис. 13.1. Структура нейронной сети Хопфилда

Общая постановка задачи, решаемой с помощью нейронной сети Хопфилда, следующая. Известен набор из  $N$  эталонных образов, которые должны храниться в памяти сети и, при необходимости, правильно распознаваться ею. Каждый образ – это вектор, состоящий из  $M$  идеальных входных сигналов. После обучения сети Хопфилда она должна уметь из поданного на ее вход зашумленного вектора выделить (распознать) один из заложенных в нее эталонных образов или дать заключение, что входные данные не соответствуют ни одному из них.

Алгоритм жизненного цикла нейронной сети Хопфилда включает две стадии: обучения и практического использования.

На стадии **обучения** выполняется следующая последовательность действий:

1.1. Составляется матрица эталонных образов  $\overline{X}$  размера  $N \times M$  (табл. 13.1). Поскольку архитектура сети предусматривает использование лишь бинарных входов, никаких дополнительных нормализации и нормировки данных не требуется.

Таблица 13.1

**Матрица эталонных образов нейронной сети Хопфилда**

№ образа	№ входной бинарной переменной					
	1	2	...	$i$	...	$M$
1	$x_{11}$	$x_{12}$	...	$x_{1i}$	...	$x_{1M}$
2	$x_{21}$	$x_{22}$	...	$x_{2i}$	...	$x_{2M}$
...	...	...	...	...	...	...
$k$	$x_{k1}$	$x_{k2}$	...	$x_{ki}$	...	$x_{kM}$
...	...	...	...	...	...	...
$N$	$x_{N1}$	$x_{N2}$	...	$x_{Ni}$	...	$x_{NM}$

1.2. Рассчитываются элементы  $w_{ji}$  матрицы весовых коэффициентов  $\overline{W}$ :

$$w_{ij} = \begin{cases} \sum_{k=1}^N x_{ik} x_{kj}, & i \neq j; \\ 0, & i = j \end{cases} \quad (13.1)$$

или в матричной форме записи:

$$\overline{\overline{W}} = \begin{cases} \overline{\overline{X}}^T \overline{\overline{X}} - \text{кроме диагональных элементов;} \\ 0 - \text{для диагональных элементов.} \end{cases}$$

Очевидно, что квадратная матрица  $\overline{\overline{W}}$  симметрична относительно нулевой главной диагонали.

На стадии **практического использования**:

2.1. На входы сети подается неизвестный, в общем случае, зашумленный вектор-столбец сигналов  $\vec{x}^*$ . Выходам нейронов присваиваются соответствующие значения элементов входного вектора:

$$\vec{y}^{(0)} = \vec{x}^*. \quad (13.2)$$

2.2. Пересчитываются состояния нейронов для следующей  $(q+1)$  итерации:

$$s_j^{(q+1)} = \sum_{i=1}^M w_{ij} y_j^{(q)} \quad (13.3)$$

или в матричной форме:

$$\vec{s}^{(q+1)} = \overline{\overline{W}} \vec{y}^{(q)}.$$

2.3. Рассчитываются новые значения элементов выходного вектора:

$$y_j^{(q+1)} = f(s_j^{(q+1)}) \quad (13.4)$$

или в матричной форме:

$$\vec{y}^{(q+1)} = f(\vec{s}^{(q+1)}).$$

где  $f(s)$  – активационная функция в виде повышающегося единичного скачка с бинарным множеством допустимых значений  $\{-1; 1\}$ :

$$f(s) = \begin{cases} -1, & s \leq T; \\ 1, & s > T. \end{cases} \quad (13.5)$$

Порог активации  $T$  обычно принимается равным нулю.

2.4. Цикл повторяется с п. 2.2 до тех пор, пока все значения элементов выходного вектора не стабилизируются, т. е. пока не будет выполнено условие:

$$\|\vec{y}^{(q)} - \vec{y}^{(q-1)}\| = 0 \quad (13.6)$$

либо пока выходные вектора не начнут чередоваться:



$$\begin{cases} \|\vec{y}^{(q)} - \vec{y}^{(q-2)}\| = 0, \\ \|\vec{y}^{(q-1)} - \vec{y}^{(q-3)}\| = 0. \end{cases} \quad (13.7)$$

2.5. После стабилизации выходов должен получиться вектор:

– соответствующий одному из эталонных образов – сеть проассоциировала и распознала его (условие (13.6));

– не соответствующий ни одному из эталонов – сеть не смогла распознать сигнал, что означает либо его сильную зашумленность, либо несоответствие ни одному из эталонных образов в памяти сети Хопфилда (условия (13.6) или (13.7)).

При использовании данной архитектуры для решения задачи распознавания образов следует иметь в виду, что это возможно только при соблюдении определенного соотношения между количеством бинарных признаков, характеризующих образы, и количеством самих запоминаемых образов. В первом приближении должно соблюдаться условие:  $N \leq 0,15 \cdot M$ . В предельно строгом выражении:

$$N \leq \frac{M}{4 \log_2 M}. \quad (13.8)$$

Если среди запоминаемых образов нет сильно похожих друг на друга, достаточно воспользоваться первым условием. В противном случае схожие образы могут вызывать так называемые перекрестные ассоциации и необходимо учитывать второе ограничение.

Рассмотрим пример распознавания двух графических образов, представленных бинарными элементами в квадратной матрице (рис. 13.2). С учетом ограничения сети Хопфилда размер входного вектора для двух эталонных образов должен быть больше 13 элементов, а поскольку графический образ кодируется квадратной матрицей, минимально возможное число входов будет равно 16. Формат (количество и последовательность) выходов идентичен формату входов.

Значение элемента входного вектора, равное +1, соответствует наличию изображения в соответствующем участке. В противном случае элемент равен –1.

В соответствии с алгоритмом для данных образов составляется матрица эталонных образов (табл. 13.2), содержащая 2 строки эталонных

Таблица 13.2

**Матрица эталонных образов для настройки нейронной сети Хопфилда**

№ образа	№ переменной															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
2	1	1	1	1	-1	1	1	-1	-1	1	1	-1	1	1	1	1

Таблица 13.3

**Матрица весовых коэффициентов нейронной сети Хопфилда**

№ строки	№ столбца															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	2	2	2	0	0	0	0	0	0	0	0	2	0	0	2
2	2	0	2	2	0	0	0	0	0	0	0	0	2	0	0	2
3	2	2	0	2	0	0	0	0	0	0	0	0	2	0	0	2
4	2	2	2	0	0	0	0	0	0	0	0	0	2	0	0	2
5	0	0	0	0	0	-2	-2	2	2	-2	-2	2	0	-2	-2	0
6	0	0	0	0	-2	0	2	-2	-2	2	2	-2	0	2	2	0
7	0	0	0	0	-2	2	0	-2	-2	2	2	-2	0	2	2	0
8	0	0	0	0	2	-2	-2	0	2	-2	-2	2	0	-2	-2	0
9	0	0	0	0	2	-2	-2	2	0	-2	-2	2	0	-2	-2	0
10	0	0	0	0	-2	2	2	-2	-2	0	2	-2	0	2	2	0
11	0	0	0	0	-2	2	2	-2	-2	2	0	-2	0	2	2	0
12	0	0	0	0	2	-2	-2	2	2	-2	-2	0	0	-2	-2	0
13	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	2
14	0	0	0	0	-2	2	2	-2	-2	2	2	-2	0	0	2	0
15	0	0	0	0	-2	2	2	-2	-2	2	2	-2	0	2	0	0
16	2	2	2	2	0	0	0	0	0	0	0	0	2	0	0	0

векторов и 16 столбцов бинарных данных. Далее с использованием соотношения (13.1) определяются весовые коэффициенты нейронной сети Хопфилда (табл. 13.3).

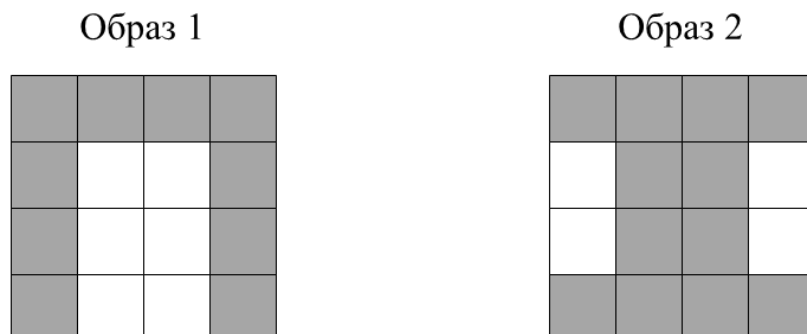


Рис. 13.2. Эталонные образы для обучения сети Хопфилда

На стадии практического использования попытаемся распознать зашумленные образы, представленные на рис. 13.3, обученной нейронной сетью.

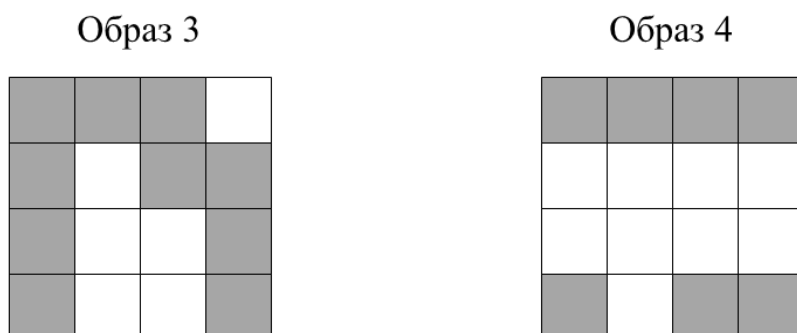


Рис. 13.3. Зашумленные образы для распознавания сетью Хопфилда

Подаем на входы сети образ 3. По соотношениям (13.2)–(13.4) последовательно определяем начальные выходные значения и серии состояний нейронов, изменяющихся при синхронной активизации обратных связей, и соответствующих им новых выходных значений.

Процесс изменения сигналов в работающей сети Хопфилда при подаче образа 3 продемонстрирован в табл. 13.4.

Последний столбец табл. 13.4 идентичен первому эталонному образцу, а значит распознавание следует признать выполненным успешно.

Таблица 13.4

**Изменение сигналов в сети Хопфилда при подаче образа 3**

№ входа или выхода $i, j$	$x_i$	$y_j^{(0)}$	Итерация 1		Итерация 2	
			$s_j^{(1)}$	$y_j^{(1)}$	$s_j^{(2)}$	$y_j^{(2)}$
1	1	1	6	1	10	1
2	1	1	6	1	10	1
3	1	1	6	1	10	1
4	-1	-1	10	1	10	1
5	1	1	14	1	18	1
6	-1	-1	-14	-1	-18	-1
7	1	1	-18	-1	-18	-1
8	1	1	14	1	18	1
9	1	1	14	1	18	1
10	-1	-1	-14	-1	-18	-1
11	-1	-1	-14	-1	-18	-1
12	1	1	14	1	18	1
13	1	1	6	1	10	1
14	-1	-1	-14	-1	-18	-1
15	-1	-1	-14	-1	-18	-1
16	1	1	6	1	10	1
$\ \vec{y}^{(q)} - \vec{y}^{(q-1)}\ $	—	—	—	8	—	0

Таблица 13.5

**Изменение сигналов в сети Хопфилда при подаче образа 4**

№ входа или выхода $i, j$	$x_i$	$y_j^{(0)}$	Итерация 1		Итерация 2		Итерация 3	
			$s_j^{(1)}$	$y_j^{(1)}$	$s_j^{(2)}$	$y_j^{(2)}$	$s_j^{(3)}$	$y_j^{(3)}$
1	1	1	10	1	10	1	10	1
2	1	1	10	1	10	1	10	1
3	1	1	10	1	10	1	10	1
4	1	1	10	1	10	1	10	1
5	-1	-1	2	1	-2	-1	2	1
6	-1	-1	2	1	-2	-1	2	1
7	-1	-1	2	1	-2	-1	2	1
8	-1	-1	2	1	-2	-1	2	1
9	-1	-1	2	1	-2	-1	2	1
10	-1	-1	2	1	-2	-1	2	1
11	-1	-1	2	1	-2	-1	2	1
12	-1	-1	2	1	-2	-1	2	1
13	1	1	10	1	10	1	10	1
14	-1	-1	2	1	-2	-1	2	1
15	1	1	-2	-1	2	1	-2	-1
16	1	1	10	1	10	1	10	1
$\ \vec{y}^{(q)} - \vec{y}^{(q-1)}\ $	—	—	—	40	—	40	—	40
$\ \vec{y}^{(q)} - \vec{y}^{(q-2)}\ $	—	—	—	—	—	0	—	0
$\ \vec{y}^{(q-1)} - \vec{y}^{(q-3)}\ $	—	—	—	—	—	—	—	0

Подает теперь на входы образ 4 и выполняем все операции в соответствии с алгоритмом стадии практического использования нейронной сети Хопфилда. Процесс изменения сигналов в этом случае представлен в табл. 13.5.

Очевидно, что в данном случае сеть не может отнести зашумленный образ к какому-либо из эталонных, так как происходит постоянное чередование двух различных выходных векторов, причем не соответствующих ни одному из исходных эталонных образов. Это позволяет сделать вывод, что значит распознавание зашумленного образа 4 данной нейронной сетью Хопфилда невозможно.

## 14. Гетероассоциативная память

Как уже отмечалось в предыдущей главе, одна из характерных особенностей человеческой памяти – возможность ассоциации каких-либо объектов (образов) с другими, непохожими на них или даже с образами-антагонистами. Так, например, в первом случае понятие «зима» ассоциируется с понятием «холод», «праздник» – с «подарком». В случае противоположностей мозгом легко выстраиваются пары: «черный» – «белый», «большой» – «маленький».

На имитацию этой особенности человеческой памяти ориентированы нейронные сети гетероассоциативной памяти – сети Коско, иначе называемые еще сетями двунаправленной ассоциативной памяти. В структуре этой сети (рис. 14.1) два слоя нейронов. Количество нейронов первого слоя  $K$  равно количеству бинарных выходных переменных. Число нейронов второго слоя  $M$  равно числу бинарных переменных, которые кодируют входной образ, соответствующий выходному. Сами значения входных и выходных переменных принадлежат бинарному множеству  $\{-1; 1\}$ .

В общем случае, задача, решаемая с помощью нейронной сети Коско, ставится следующим образом. Известен набор из  $N$  эталонных входных образов и соответствующих им  $N$  идеальных выходных образов. После обучения сеть Коско должна уметь из поданного на ее вход зашумленного входного вектора выделить один из заложенных в нее выходных иде-

альных образов или дать заключение, что входные данные не соответствуют ни одному из них. Таким образом, в отличие от нейронной сети Хопфилда, сеть двунаправленной ассоциативной памяти должна не только избавиться от шума во входном векторе, но и сопоставить соответствующий ему выходной вектор.

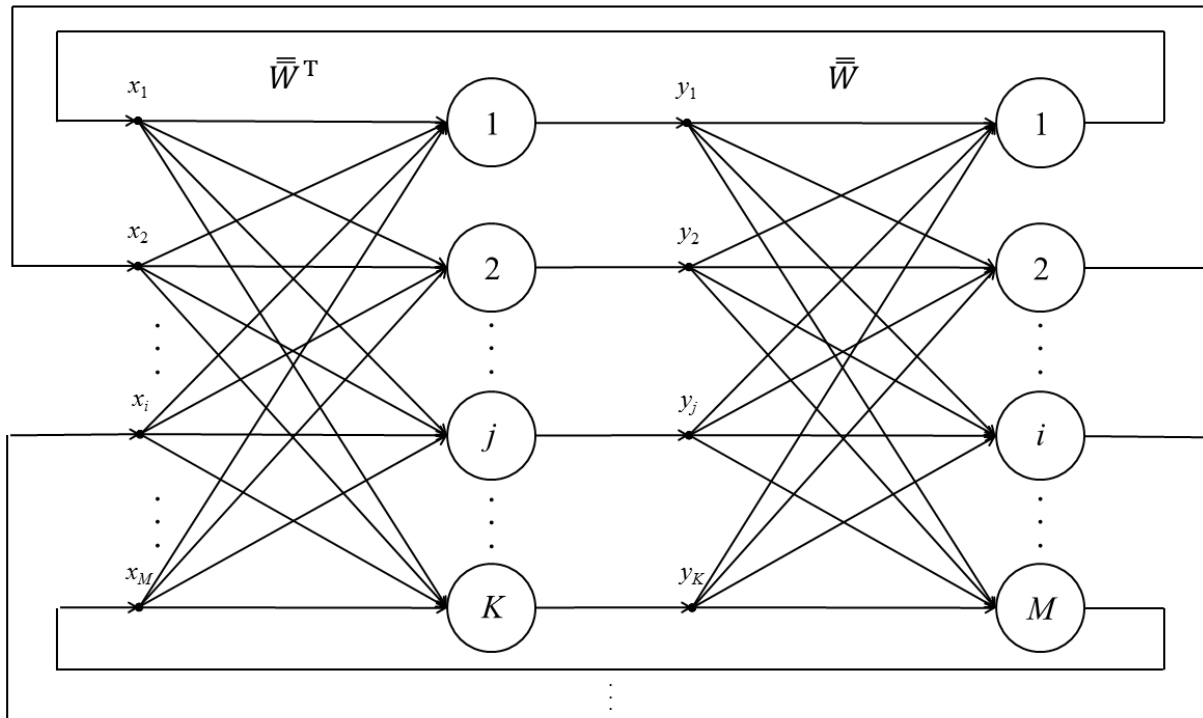


Рис. 14.1. Структура нейронной сети Коско

Алгоритм жизненного цикла нейронной сети Коско включает две стадии: обучения и практического использования.

Стадия **обучения** содержит следующую последовательность действий:

1.1. Формируются матрицы ассоциированных входных и выходных образов. Матрица входных образов  $\bar{X}$  размера  $N \times M$  аналогична матрице эталонных входных образов нейронной сети Хопфилда (табл. 13.1). Структура матрицы выходных образов  $\bar{Y}$  размера  $N \times K$  приведена в табл. 14.1.

1.2. Рассчитываются элементы  $w_{ij}$  матрицы весовых коэффициентов  $\bar{W}$ :

$$w_{ij} = \sum_{k=1}^N x_{ik} y_{kj} \quad (14.1)$$

или в матричной форме записи:

$$\overline{\overline{W}} = \overline{\overline{X}}^T \overline{\overline{Y}}.$$

Таблица 14.1

**Матрица выходных образов нейронной сети Коско**

№ образа	№ выходной бинарной переменной					
	1	2	...	$j$	...	$K$
1	$y_{11}$	$y_{12}$	...	$y_{1j}$	...	$y_{1K}$
2	$y_{21}$	$y_{22}$	...	$y_{2j}$	...	$y_{2K}$
...	...	...	...	...	...	...
$k$	$y_{k1}$	$y_{k2}$	...	$y_{kj}$	...	$y_{kK}$
...	...	...	...	...	...	...
$N$	$y_{N1}$	$y_{N2}$	...	$y_{Nj}$	...	$y_{NK}$

На стадии **практического использования** выполняются следующие операции:

2.1. Неизвестный, возможно, зашумленный образ  $\vec{x}^{(0)}$  подается в виде значений элементов входного вектора-столбца в нейронную сеть. Для него или любого следующего  $(q+1)$ -го приближения эталонного входного вектора рассчитываются состояния нейронов первого слоя:

$$s_{1j}^{(q+1)} = \sum_{i=1}^M w_{ji} x_i^{(q)} \quad (14.2)$$

или в матричной форме:

$$\vec{s}_1^{(q+1)} = \overline{\overline{W}}^T \vec{x}^{(q)}.$$

2.2. К рассчитанным на предыдущем шаге состояниям нейронов применяется пороговая активационная функция вида (13.5), в результате чего получается новый вектор выходных значений  $\vec{y}^{(q+1)}$ .

Если значение порога  $T$  одинаково для всех нейронов, например,  $T = 0$ , такая нейронная сеть Коско называется **гомогенной**. Если имеются различающиеся значения порогов, сеть называется **негомогенной**. Негомогенные сети Коско имеет смысл использовать в тех случаях, когда входные или выходные вектора содержат различные по своему происхождению

элементы.

2.3. Для всех итераций, кроме первой ( $q \neq 1$ ), проверяется условие окончания – стабилизация элементов выходного вектора – равенство (13.6). Если оно не выполняется, цикл расчетов продолжается с п. 2.4, иначе – переход к п. 2.6.

2.4. Полученные значения элементов выходного образа подаются на входы нейронов второго слоя, для которого, в свою очередь, рассчитываются новые состояния нейронов:

$$s_{2i}^{(q+1)} = \sum_{j=1}^K w_{ij} y_j^{(q+1)} \quad (14.3)$$

или в матричной форме:

$$\vec{s}_2^{(q+1)} = \overline{\overline{W}} \vec{y}^{(q+1)}.$$

2.5. К рассчитанным состояниям нейронов второго слоя также применяется пороговая активационная функция (13.5) и вычисляются элементы нового вектора входного образа  $\vec{x}^{(q+1)}$ .

2.6. Проверяется соответствие стабилизировавшегося вектора выходных значений одному из идеальных образов матрицы  $\overline{\overline{Y}}$ . Если соответствие установлено, можно сделать вывод, что нейронной сети удалось найти ассоциацию между зашумленным входом и одним из идеальных выходов. В противном случае можно сказать, что входной вектор был слишком сильно зашумлен и соответствие не может быть установлено.

Искусственная нейронная сеть Коско имеет ограничения, связанные с количеством запоминаемых образов. Если все они достаточно сильно друг от друга отличаются, в первом приближении достаточно выполнения условия:  $N \leq n$ , где  $n$  – минимальное из двух слоев количество нейронов. Предельно строгое условие аналогично неравенству (13.8) для сети Хопфилда.

Задачу ассоциации неоднородных образов с помощью нейронной сети Коско можно свести к задаче их классификации. В этом случае количество переменных выходного образа должно соответствовать количеству возможных классов. Для каждого эталонного образа класс кодируется так: в выходном векторе все значения элементов, кроме одного, принимаются равными  $-1$ . Единственное положительное значение элемента выходного



вектора располагается в позиции, соответствующей нужному классу.

Рассмотрим пример работы гетероассоциативной памяти для двух графических образов, представленных бинарными элементами в квадратной матрице (рис. 14.2). Входной вектор содержит 16 элементов, выходной – 9 элементов. Соответственно, структура нейронной сети будет включать 9 нейронов в первом слое и 16 – во втором.

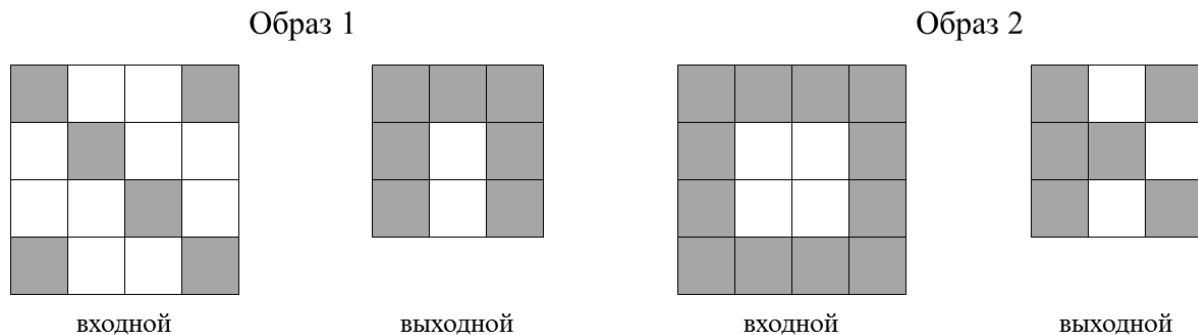


Рис. 14.2. Эталонные входные и выходные образы для обучения сети Коско

Значение элемента входного вектора, равное +1, соответствует закрашенному фрагменту изображения входного образа, –1 – незакрашенному.

В соответствии с алгоритмом стадии обучения нейронной сети Коско составляются матрицы входных и выходных эталонных образов (табл. 14.2, 14.3).

Таблица 14.2

**Матрица входных образов для настройки нейронной сети Коско**

№ образа	№ входной переменной							
	1	2	3	4	5	6	7	8
1	1	–1	–1	1	–1	1	–1	–1
2	1	1	1	1	1	–1	–1	1
№ образа	№ входной переменной							
	9	10	11	12	13	14	15	16
1	–1	–1	1	–1	1	–1	–1	1
2	1	–1	–1	1	1	1	1	1

Обе матрицы содержат по 2 строки бинарных данных, кодирующие соответствующие друг другу пары образов, однако размер этих образов

разный: в первом случае – 16 входов, во втором – 9 выходов. Далее с использованием соотношения (14.1) определяются весовые коэффициенты сети Коско (табл. 14.4).

Таблица 14.3

**Матрица выходных образов для настройки нейронной сети Коско**

№ образа	№ выходной переменной								
	1	2	3	4	5	6	7	8	9
1	1	1	1	1	–1	1	1	–1	1
2	1	–1	1	1	1	–1	1	–1	1

Для тестирования сети Коско на стадии практического использования воспользуемся зашумленным входным графическим образом, показанным на рис. 14.3.

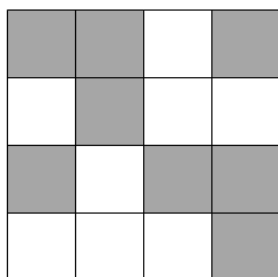


Рис. 14.3. Зашумленный входной образ для распознавания сетью Коско

Бинарный входной вектор, соответствующий тестовому образу, подается на входы нейронной сети. По соотношениям (14.2) и (13.5) производится ассоциация данного вектора с выходным образом. По соотношениям (14.3) и (13.5) выполняется обратная ассоциация вновь полученного вектора выходов с входным вектором. Ход последовательных ассоциаций представлен в табл. 14.5.

Как видно из таблицы, стабилизация выходного вектора наступила уже на второй итерации, и она оказалась идентичной второму выходному эталонному образу. Следовательно, нейронная сеть, моделирующая работу гетероассоциативной памяти, успешно решила задачу для зашумленного набора входных значений.

Таблица 14.4

**Матрица весовых коэффициентов нейронной сети Коско**

№ входного нейрона	№ выходного нейрона								
	1	2	3	4	5	6	7	8	9
1	2	0	2	2	0	0	2	-2	2
2	0	-2	0	0	2	-2	0	0	0
3	0	-2	0	0	2	-2	0	0	0
4	2	0	2	2	0	0	2	-2	2
5	0	-2	0	0	2	-2	0	0	0
6	0	2	0	0	-2	2	0	0	0
7	-2	0	-2	-2	0	0	-2	2	-2
8	0	-2	0	0	2	-2	0	0	0
9	0	-2	0	0	2	-2	0	0	0
10	-2	0	-2	-2	0	0	-2	2	-2
11	0	2	0	0	-2	2	0	0	0
12	0	-2	0	0	2	-2	0	0	0
13	2	0	2	2	0	0	2	-2	2
14	0	-2	0	0	2	-2	0	0	0
15	0	-2	0	0	2	-2	0	0	0
16	2	0	2	2	0	0	2	-2	2

Таблица 14.5

**Изменение сигналов в сети Коско при подаче тестового образа**

№ входа или выхо- да	$x_i^{(0)}$	Итерация 1				Итерация 2			
		расчет выходов		расчет входов		расчет выходов		расчет входов	
		$s_{1j}^{(1)}$	$y_j^{(1)}$	$s_{2i}^{(1)}$	$x_i^{(1)}$	$s_{1j}^{(2)}$	$y_j^{(2)}$	$s_{2i}^{(2)}$	$x_i^{(2)}$
1	1	8	1	12	1	12	1	12	1
2	1	8	1	-6	-1	20	1	-6	-1
3	-1	8	1	-6	-1	12	1	-6	-1
4	1	8	1	12	1	12	1	12	1
5	-1	-8	-1	-6	-1	-20	-1	-6	-1
6	1	8	1	6	1	20	1	6	1
7	-1	8	1	-12	-1	12	1	-12	-1
8	-1	-8	-1	-6	-1	-12	-1	-6	-1
9	1	8	1	-6	-1	12	1	-6	-1
10	-1	—	—	-12	-1	—	—	-12	-1
11	1	—	—	6	1	—	—	6	1
12	1	—	—	-6	-1	—	—	-6	-1
13	-1	—	—	12	1	—	—	12	1
14	-1	—	—	-6	-1	—	—	-6	-1
15	-1	—	—	-6	-1	—	—	-6	-1
16	1	—	—	12	1	—	—	12	1
$\ \bar{y}^{(q)} - \bar{y}^{(q-1)}\ $	—	—	—	—	—	—	0	—	—

В отличие от искусственной нейронной сети Хопфилда, в сетях Коско весьма сложно столкнуться с ситуацией зацикленного чередования двух различных образов в выходном векторе. Именно по этой причине обычно бывает достаточно проверки одного условия окончания вычислительной процедуры – условия (13.6).

## 15. Нейронные сети Хэмминга

Искусственная нейронная сеть Хэмминга используется для решения задач классификации бинарных входных векторов. В основе ее работы лежат процедуры, направленные на выбор в качестве решения задачи классификации одного из эталонных образов, наиболее близкого к поданному на вход сети зашумленному входному образу, и отнесение данного образа к соответствующему классу. Для оценки меры близости к каждому классу используется критерий, учитывающий расстояние Хэмминга – количество различающихся переменных у зашумленного и эталонного входных образов.

Структурно нейронная сеть Хэмминга включает два слоя (рис. 15.1), количество нейронов в которых  $K$  равно количеству классов ( $K = N$ ). Число входов  $M$  соответствует числу бинарных признаков, по которым различаются образы. Значения входных переменных принадлежат множеству  $\{-1; 1\}$ . Выходные значения подаются по обратным связям на входы нейронов второго слоя, в том числе свой собственный.

Общая постановка задачи, которая решается с помощью нейронной сети Хэмминга, следующая. Имеется исходный набор эталонных образов, представленных в виде бинарных векторов. Каждому из них соответствует свой класс. Требуется для поданного на входы сети неизвестного образа произвести его сопоставление со всеми известными эталонными образами и отнесение к соответствующему классу либо сделать заключение о несоответствии ни одному из классов.

Алгоритм жизненного цикла нейронной сети Хэмминга состоит из двух основных стадий: обучения и практического использования.

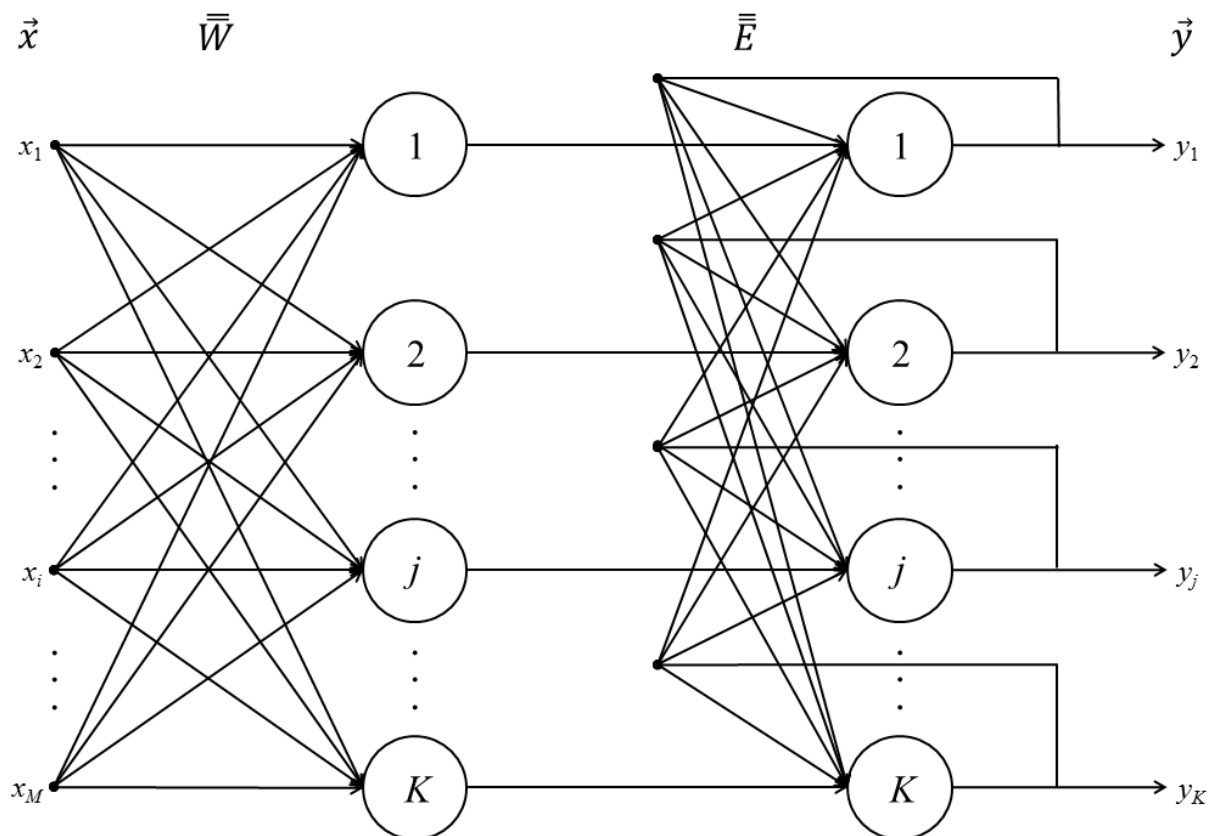


Рис. 15.1. Структура нейронной сети Хэмминга

На стадии **обучения** выполняется следующая последовательность действий:

1.1. Формируется матрица эталонных образов  $\bar{X}$  размера  $K \times M$  (табл. 15.1)

Таблица 15.1

**Матрица эталонных образов нейронной сети Хэмминга**

№ образа	№ входной бинарной переменной					
	1	2	...	$i$	...	$M$
1	$x_{11}$	$x_{12}$	...	$x_{1i}$	...	$x_{1M}$
2	$x_{21}$	$x_{22}$	...	$x_{2i}$	...	$x_{2M}$
...	...	...	...	...	...	...
$j$	$x_{j1}$	$x_{j2}$	...	$x_{ji}$	...	$x_{jM}$
...	...	...	...	...	...	...
$K$	$x_{K1}$	$x_{K2}$	...	$x_{Ki}$	...	$x_{KM}$

1.2. Рассчитывается матрица весовых коэффициентов нейронов пер-

вого слоя:

$$w_{ij} = \frac{1}{2} x_{ij} \quad (15.1)$$

или в матричной форме записи:

$$\overline{\overline{W}} = \frac{1}{2} \overline{\overline{X}}.$$

1.3. Определяются настройки активационной функции:

– вид – линейная пороговая функция:

$$f(s) = \begin{cases} 0, & s \leq 0; \\ s, & 0 < s \leq T; \\ T, & s \geq T; \end{cases} \quad (15.2)$$

– параметр:

$$T = \frac{M}{2}. \quad (15.3)$$

Таким образом, очевидно, что выходы нейронной сети могут принимать любые значения в пределах  $[0, T]$ .

1.4. Задаются значения синапсов обратных связей нейронов второго слоя в виде элементов квадратной матрицы размера  $K \times K$ :

$$\varepsilon_{jp} = \begin{cases} 1, & j = p; \\ -\varepsilon, & j \neq p, \end{cases} \quad (15.4)$$

где  $\varepsilon \in \left(0, \frac{1}{K}\right]$

или в матричной форме:

$$\overline{\overline{E}} = \begin{cases} 1 - \text{для диагональных элементов;} \\ -\varepsilon - \text{кроме диагональных элементов.} \end{cases}$$

Синапсы обратных связей нейронной сети Хэмминга, имеющие отрицательный вес, называются **ингибиторными**, или **тормозящими**.

1.5. Устанавливается максимально допустимое значение нормы разности выходных векторов на двух последовательных итерациях  $E_{\max}$ , требующееся для оценки стабилизации решения. Обычно достаточно принимать  $E_{\max} = 0,1$ .

На стадии **практического использования** выполняются следующие действия:

2.1. На входы сети подается неизвестный, в общем случае, зашумленный вектор сигналов  $\vec{x}^*$ .

2.2. Рассчитываются состояния и выходные значения нейронов первого слоя.

Для расчета состояний нейронов используется соотношение:

$$s_{1j} = \sum_{i=1}^M w_{ji} x_i^* + T \quad (15.5)$$

или в матричной форме:

$$\vec{s}_1 = \overline{W} \vec{x}^* + \vec{T}.$$

Для расчета выходов нейронов первого слоя  $\vec{y}_1$  к полученным значениям состояний применяется активационная функция (15.2).

2.3. Выходам нейронов второго слоя в качестве начальных величин присваиваются значения выходов нейронов первого слоя, полученные на предыдущем шаге:

$$\vec{y}_2^{(0)} = \vec{y}_1.$$

Далее первый слой нейронов на стадии практического использования больше не задействуется.

2.4. Для каждой итерации  $q$  рассчитываются новые значения состояний и выходов нейронов второго слоя.

Состояния нейронов определяются по соотношению:

$$s_{2j}^{(q+1)} = y_{2j}^{(q)} - \varepsilon \sum_{\substack{p=1, \\ p \neq j}}^K y_{2p}^{(q)} \quad (15.6)$$

или в матричной форме записи:

$$\vec{s}_2^{(q+1)} = \overline{E} \vec{y}_2^{(q)}.$$

Новые выходные значения  $\vec{y}_2^{(q+1)}$  определяются в результате применения линейной пороговой активационной функции (15.2) к соответствующим состояниям нейронов  $\vec{s}_2^{(q+1)}$ .

2.5. Цикл в п. 2.4 повторяется до стабилизации выходного вектора в соответствии с условием:

$$\|\vec{y}^{(q+1)} - \vec{y}^{(q)}\| \leq E_{\max}. \quad (15.7)$$

В идеальном случае после стабилизации должен получиться выход-

ной вектор с одним положительным и всеми остальными нулевыми элементами. Индекс единственного положительного элемента непосредственно указывает на класс неизвестного входного образа.

Если данные входного образа сильно зашумлены или в обучающей выборке отсутствовал подходящий эталон, в результате останова цикла в п. 2.4 могут быть получены несколько положительных выходов, причем значение любого из них окажется меньше, чем  $E_{\max}$ . В этом случае делается заключение о невозможности отнесения входного образа к определенному классу, однако индексы положительных выходов указывают на наиболее схожие с ним эталоны.

Рассмотрим пример использования искусственной нейронной сети Хэмминга для классификации по трем эталонным графическим образам, представленным бинарными элементами в квадратной матрице (рис. 15.2). Таким образом, входной вектор содержит 9 элементов, выходной – 3 элемента.

Наличие рисунка в любом из 9 элементов кодируется значением 1, отсутствие – значением  $-1$ .

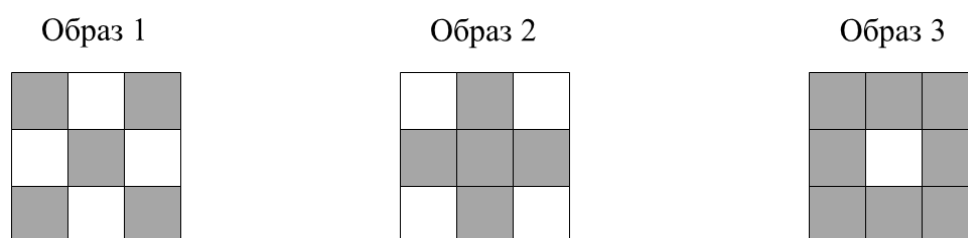


Рис. 15.2. Эталонные образы для обучения сети Хэмминга

Создаваемая нейронная сеть будет включать 9 входных переменных и по 3 нейрона в первом и втором (выходном) слоях (рис. 15.3).

В соответствии с алгоритмом стадии обучения искусственной нейронной сети Хэмминга сформируем матрицу эталонных образов (табл. 15.2). Так как имеют место 3 эталонных образа с 9 входами у каждого, матрица будет содержать 3 строки и 9 столбцов.

На основе матрицы эталонных образов с использованием соотношения (15.1) рассчитываем весовые коэффициенты нейронов первого слоя. Результаты расчета приведены в табл. 15.3.



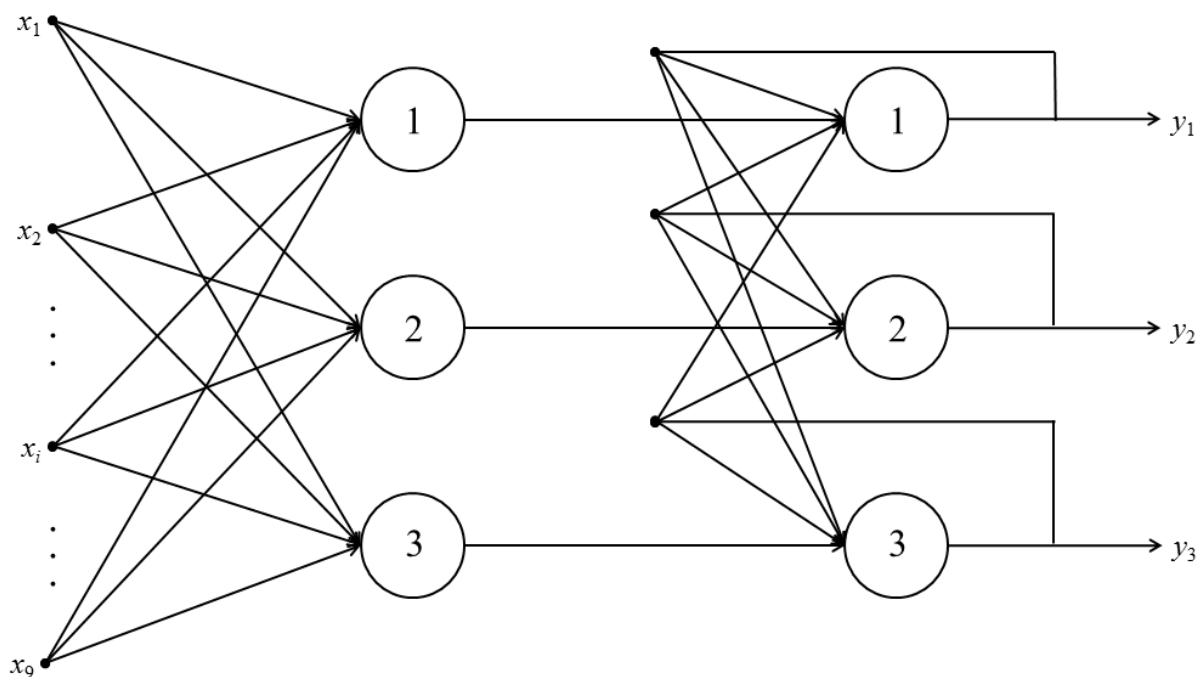


Рис. 15.3. Структура сети Хэмминга для классификации трех образов

Таблица 15.2

**Матрица эталонных образов для настройки нейронной сети Хэмминга**

№ образа	№ входной переменной								
	1	2	3	4	5	6	7	8	9
1	1	-1	1	-1	1	-1	1	-1	1
2	-1	1	-1	1	1	1	-1	1	-1
3	1	1	1	1	-1	1	1	1	1

Таблица 15.3

**Матрица весовых коэффициентов нейронной сети Хэмминга**

№ нейрона первого слоя	№ входной переменной								
	1	2	3	4	5	6	7	8	9
1	0,5	-0,5	0,5	-0,5	0,5	-0,5	0,5	-0,5	0,5
2	-0,5	0,5	-0,5	0,5	0,5	0,5	-0,5	0,5	-0,5
3	0,5	0,5	0,5	0,5	-0,5	0,5	0,5	0,5	0,5

По формуле (15.3) определяем порог активационной функции  $T = 4,5$ . С учетом ограничения  $\varepsilon \in \left(0, \frac{1}{3}\right]$  абсолютное значение веса каждо-

го ингибиторного синапса примем  $\varepsilon = 0,3$ . Также зададимся предельной величиной критерия стабилизации выходного вектора  $E_{\max} = 0,1$ .

На основе выражения (15.4) составим матрицу весовых коэффициентов обратных синапсов (табл. 15.4).

Таблица 15.4

**Матрица весов обратных связей нейронной сети Хэмминга**

№ выходного нейрона	№ нейрона первого слоя		
	<b>1</b>	<b>2</b>	<b>3</b>
1	1,0	−0,3	−0,3
2	−0,3	1,0	−0,3
3	−0,3	−0,3	1,0

Для тестирования настроенной сети используем два зашумленных графических образа, показанных на рис. 15.4.



Рис. 15.4. Тестовые образы для обученной сети Хэмминга

Подаем на входы сети бинарный вектор, соответствующий образу 4:  $\vec{x}^T = [1, -1, -1, -1, 1, -1, 1, -1, 1]$ . Применение расчетного соотношения (15.5) дает вектор-столбец состояний нейронов первого слоя, а активационной функции (15.2) к состояниям – вектор-столбец выходных значений нейронов первого слоя:

$$\vec{s}_1 = \begin{bmatrix} 8,00 \\ 2,00 \\ 3,00 \end{bmatrix}, \quad \vec{y}_1 = \begin{bmatrix} 4,50 \\ 2,00 \\ 3,00 \end{bmatrix}.$$

В соответствии с п. 2.3 алгоритма стадии практического использования нейронной сети Хэмминга выходам нейронной сети присваиваются соответствующие выходные значения нейронов первого слоя. Далее с ис-

пользованием соотношений (15.6) и (15.2) итерационно рассчитывается серия выходных векторов до выполнения условия стабилизации (15.7). Сигналы нейронной сети Хэмминга, получаемые на протяжении полного цикла расчета при подаче тестового образа 4 на ее входы, представлены в табл. 15.5.

Таблица 15.5

**Изменение сигналов в сети Хэмминга при подаче на входы образа 4**

Номер итерации	Вектор состояний			Вектор выходов			$\ \vec{y}^{(q+1)} - \vec{y}^{(q)}\ $
	$s_{21}^{(q)}$	$s_{22}^{(q)}$	$s_{23}^{(q)}$	$y_{21}^{(q)}$	$y_{22}^{(q)}$	$y_{23}^{(q)}$	
1	8,00	2,00	3,00	4,50	2,00	3,00	—
2	3,00	−0,25	1,05	3,00	0,00	1,05	10,05
3	2,69	−1,22	0,15	2,69	0,00	0,15	0,91
4	2,64	−0,85	−0,66	2,64	0,00	0,00	0,02

Как видно из таблицы, критерий остановки цикла возврата сигнала по обратным связям выполнен после 4-й итерации. Положительное выходное значение 1-го нейрона указывает на то, что зашумленный входной образ следует отнести к 1-му классу.

Аналогичные расчеты выполняются для тестового образа 5 с бинарным входным вектором вида:  $\vec{x}^T = [1, 1, 1, -1, -1, -1, 1, 1, 1]$ . Изменения сигналов нейронной сети для этого случая приведены в табл. 15.6.

Таблица 15.6

**Изменение сигналов в сети Хэмминга при подаче на входы образа 5**

Номер итерации	Вектор состояний			Вектор выходов			$\ \vec{y}^{(q+1)} - \vec{y}^{(q)}\ $
	$s_{21}^{(q)}$	$s_{22}^{(q)}$	$s_{23}^{(q)}$	$y_{21}^{(q)}$	$y_{22}^{(q)}$	$y_{23}^{(q)}$	
1	6,00	2,00	7,00	4,50	2,00	4,50	—
2	2,55	−0,70	2,55	2,55	0,00	2,55	11,61
3	1,79	−1,53	1,79	1,79	0,00	1,79	1,17
4	1,25	−1,07	1,25	1,25	0,00	1,25	0,57
5	0,87	−0,75	0,87	0,87	0,00	0,87	0,28
6	0,61	−0,52	0,61	0,61	0,00	0,61	0,14
7	0,43	−0,37	0,43	0,43	0,00	0,43	0,07

В этом случае критерий остановки был выполнен после 7-й итера-

ции, однако уже на 2-й итерации стало понятно, что сеть Хэмминга не может отдать предпочтение 1-му и 3-му классам при отнесении входного зашумленного образа 5. В условиях малого количества входных характеристик следует сделать вывод, скорее, о том, что сеть вовсе не смогла классифицировать образ, чем о том, что она в равной степени отнесла его к двум классам.

## **Заключение**

Приведенными в учебном пособии архитектурами искусственных нейронных сетей отнюдь не ограничивается их многообразие в нейроинформатике. Точно также приведенные примеры использования описанных здесь архитектур – лишь маленькая толика тех задач, которые грамотный специалист в области методов искусственного интеллекта может решать с помощью нейросетевого подхода.

Определенно, их использование для серьезной практической деятельности или в науке не может быть всегда оправдано, поэтому важно не только уметь применять искусственные нейронные сети на практике, но и понимать, когда это делать целесообразно, а когда стоит воспользоваться более простыми и менее ресурсоемкими методами.

Тем не менее, уже в течение нескольких десятилетий очевидно, что на современном уровне развития математики, кибернетики и информационных технологий квалифицированные специалисты этих отраслей науки должны знать и владеть нейросетевыми методами как очень эффективным инструментом математического моделирования сложных и многомерных явлений и процессов. В этой связи авторы учебного пособия искренне надеются, что оно окажется хорошим подспорьем в освоении студентами высших профессиональных учебных заведений такой сложной науки, как нейроинформатика.

## Предметный указатель

- Автоассоциативная память, 77  
Адаптивный резонанс, 68  
Аппроксимация данных, 26  
Архитектура нейронной сети, 9  
Ассоциативная память, 77  
Ассоциация образов, 28  
Весовой коэффициент, 10  
Входной слой, 12  
Выходной слой, 12  
Гетероассоциативная память, 78, 85  
Гиперболический тангенс, 19  
Двухслойная сеть, 12  
Дообучение нейронной сети, 33  
Жизненный цикл нейронной сети, 31  
Импульсные функции, 20  
Ингибиторные синапсы, 94  
Интерполирование данных, 26  
Искусственная нейронная сеть, 9  
Искусственный нейрон, 9  
Карта Кохонена, 64  
Классификация образов, 29  
Кластеризация данных, 30  
Кластерный анализ, 63  
Контрастирование нейронной сети, 32  
Коэффициент смещения, 10  
Линейная нормализация, 15  
Логистическая функция, 19  
Метод обратного распространения ошибки, 42  
Метод Уидроу–Хоффа, 35  
Многослойный перцептрон, 40  
Нейроинформатика, 9  
Нелинейная нормализация, 16  
Непротиворечивость выборки, 22  
Обучение нейронной сети, 10, 32  
Однослойная сеть, 12  
Однослойный перцептрон, 34  
Ошибки нейронных сетей, 36  
Перцептрон, 33  
Полносвязная сеть, 14  
«Правило 2–5», 41  
Прогнозирование временных рядов, 27  
Радиально-симметричная функция, 52  
Распознавание образов, 28  
Репрезентативность выборки, 23  
Самообучение, 59  
Самоорганизация, 59, 61  
Сигмоидная функция, 19  
Синаптический коэффициент, 10  
Скрытый слой, 12  
Слой нейронов, 12  
Состояние нейрона, 11  
Структура нейронной сети, 10  
Тестирование нейронной сети, 33  
Уникальность примеров, 22  
Функции гистерезиса, 18  
Функция активации, 11  
Эпоха обучения, 11

## Оглавление

Перечень принятых обозначений .....	3
Введение.....	5
1. История развития искусственных нейронных сетей .....	6
2. Основные положения и классификация искусственных нейронных сетей.....	9
3. Способы нормализации переменных .....	15
4. Активационные функции и их свойства .....	18
5. Методы обработки выборки исходных данных .....	22
6. Особенности постановки и решения задач с использованием искусственных нейронных сетей.....	26
7. Этапы жизненного цикла нейронной сети .....	31
8. Однослойные перцептроны.....	33
9. Многослойные перцептроны .....	40
10. Нейронные сети на основе радиально-симметричных функций .....	52
11. Нейронные сети Кохонена .....	59
12. Нейронные сети адаптивного резонанса .....	68
13. Автоассоциативная память.....	77
14. Гетероассоциативная память .....	85
15. Нейронные сети Хэмминга .....	92
Заключение .....	101
Предметный указатель.....	102

Учебное издание

ДУДАРОВ Сергей Павлович  
ПАПАЕВ Павел Леонидович

**Теоретические основы и практическое применение  
искусственных нейронных сетей**

Редактор: Р. Г. Чиркова

Подписано в печать 20.10.14. Формат 60×84 1/16.  
Усл. печ. л. 6,05. Уч. - изд. л. 5,92. Тираж 150 экз.  
Заказ

Российский химико-технологический университет им. Д. И. Менделеева  
Издательский центр  
Адрес университета и издательского центра:  
125047 Москва, Миусская пл., 9