МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Солдатова О. П.

Системы искусственного интеллекта

Курс лекций

Самара

2016

СОДЕРЖАНИЕ

1 Введение в нейронные сети

1.1 Основные свойства нейронных сетей

Исследования по искусственным нейронным сетям связаны с тем, что способ обработки информации человеческим мозгом принципиально отличается от методов, применяемых обычными цифровыми компьютерами. Мозг представляет собой чрезвычайно сложный, нелинейный, параллельный компьютер. Он обладает способностью организовывать свои структурные компоненты, называемые нейронами, так, чтобы они могли выполнить конкретные задачи (такие как распознавание образов, обработку сигналов органов чувств, моторные функции) во много раз быстрее, чем могут позволить самые быстродействующие компьютеры. Мозг имеет совершенную структуру, позволяющую строить собственные правила на основе опыта. Опыт накапливается с течением времени.

Понятие развития нейронов мозга связано с понятием пластичности мозга – способностью настройки нервной системы в соответствии с окружающей средой. Аналогично в искусственных нейронных сетях производится настройка искусственных нейронов и формируется структура нейронной сети. В общем случае нейронная сеть представляет машину, моделирующую способ обработки мозгом конкретной задачи. Эта сеть обычно реализуется с помощью электронных компонентов или моделируется программой.

Таким образом, можно дать следующее определение нейронных сетей, выступающих в роли адаптивной машины [3]:

нейронная сеть – это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и представляющих их для последующей обработки. Нейронная сеть сходна с мозгом с двух точек зрения:

- *знания поступают в нейронную сеть из окружающей среды и используются в процессе обучения;*
- *для накопления знаний применяются связи между нейронами, называемые синаптическими весами.*

Процедура настройки синаптических весов называется *алгоритмом обучения.*Эта процедура выстраивает в определённом порядке веса нейронов сети для обеспечения необходимой взаимосвязи между ними.

Наиболее существенными свойствами нейронных сетей являются:

1. *Нелинейность*.Поскольку искусственные нейроны могут быть линейными и нелинейными, то нейронные сети позволяют воспроизводить сложные зависимости, как линейные, так и нелинейные. Нейронные сети реализуют нелинейность особого вида, так как она распределена по сети. Кроме того, нейронные сети справляются с "проклятием размерности", которое не позволяет моделировать нелинейные зависимости в случае большого числа переменных.

2. *Параллельная обработка информации.* Благодаря этой способности при большом количестве межнейронных связей достигается значительное ускорение процесса обработки информации. Во многих ситуациях становится возможной обработка сигналов в реальном масштабе времени.

3. *Обучение на примерах*. Одной из популярных парадигм обучения является *обучение с учителем*. Такой способ обучения предполагает изменение синаптических весов на основе набора *учебных примеров*. Каждый пример состоит из входного сигнала и соответствующего ему *ожидаемого* выходного сигнала. Нейронная сеть модифицирует синаптические веса для минимизации разности ожидаемого выходного сигнала и *реального* выходного сигнала, формируемого нейронной сетью. Таким образом, нейронная сеть обучается на примерах, представляющих собой таблицу соответствий вход-выход для конкретной задачи. Ранее использованные примеры могут быть использованы для обучения снова в таком же или ином порядке.

4. *Адаптивность* (adaptivity). Нейронные сети обладают способностью адаптировать свои синаптические веса к изменениям окружающей среды. Нейронные сети могут быть легко переучены для работы в *нестационарной* среде. Для того, чтобы использовать все достоинства адаптивности, основные параметры системы должны быть достаточно стабильными, чтобы не учитывать внешние помехи, и достаточно гибкими, чтобы обеспечить реакцию на существенные изменения среды.

5. *Нечувствительность к ошибкам* (faulttolerance). Очень большое количество межнейронных соединений приводит к тому, что сеть становится нечувствительной к ошибкам, возникающим в отдельных контактах. Функции поврежденных соединений принимают на себя другие элементы, в результате в деятельности сети не наблюдаются заметные нарушения. Только серьезные повреждения структуры нейронных сети существенно влияют на ее работоспособность.

6. *Способность к обобщению полученных знаний*. Сеть обладает чертами искусственного интеллекта. Натренированная на ограниченном множестве обучающих примеров, она обобщает накопленную информацию и вырабатывает ожидаемую реакцию применительно к данным, не обрабатывавшимся в процессе обучения.

7. *Единообразие анализа и проектирования.* Нейронные сети являются универсальным механизмом обработки информации. Одно и тоже проектное решение нейронной сети может быть использовано в разных предметных областях. Это свойство проявляется из-за нескольких причин:

- нейроны являются стандартными составными частями любой нейронной сети;

- можно использовать одни и те же алгоритмы обучения в различных нейросетевых приложениях;
- на основе интеграции целых модулей могут быть построены модульные сети.

*8. Аналогия с нейробиологией.*Строение нейронных стетй определяется аналогией с живым мозгом, являющимся доказательством возможности существования отказоустойчивых вычислительных параллельных систем, эффективно решающих поставленные задачи.

Наличие перечисленных свойств вызвало в последние годы огромный рост интереса к нейронным сетям и существенный прогресс в их исследовании. Искусственные нейронные сети используются для аппроксимации функций, сжатия данных, классификации и распознавания образов, прогнозирования, идентификации, оценивания и ассоциативного управления.

1.2 Биологические основы нейронных сетей

Искусственные нейронные сети возникли на основе знаний о функционировании нервной системы живых существ. Невную систему человека можно представить в виде трёхступенчатой системы.

Центром этой системы является *мозг,* представляемый сетью нервных клеток,то есть нейронной сетью. Он получает информацию, анализирует ее и выдает соответствующие решения. *Рецепторы* получают информацию от тела и окружающей среды и преобразуют её в электрический импульс, передаваемый в мозг. *Эффекторы* преобразовывают электрические импульсы, вырабатываемы мозгом в выходные сигналы.

Нервная клетка, сокращенно называемая нейроном, является основным элементом нервной системы. У нейрона есть тело, называемое сомой, внутри которого располагается ядро. Из сомы нейрона выходят отростки двух видов: многочисленные тонкие, густо ветвящиеся дендриты и более толстый, расщепляющийся на многочисленные нервные окончания – колатералы,аксон (рис.1.1)[4].

Выходной сигнал клетки передается через аксон при помощи колатералов. Колатералы контактируют с сомой и дендритами других нейронов, образуя каналы связи выходных сигналов клетки с входами других клеток, которые называются синапсами. Синапсы могут находиться как на дендритах, так и непосредственно в теле клетки. Самым распространённым типом синапсов является химический синапс, который работает следующим образом.

Передача сигналов внутри нервной системы – это очень сложный электрохимический процесс. С большим упрощением можно считать, что передача нервного импульса между двумя клетками основана на выделении особых химических субстанций, называемых нейромедиаторами, которые формируются под влиянием поступающих от синапсов раздражителей. Предсинаптический процесс формирует субстанцию, которая методом диффузии передаётся по синаптическим связям и влияет на

постсинаптический процесс. Синапс преобразует пресинаптический электрический сигнал в химический, а затем в постсинаптический - электрический.

Данные субстанции воздействуют на клеточную мембрану, вызывая изменение ее энергетического потенциала, причем величина этого изменения пропорциональна количеству нейромедиатора, попадающего на мембрану.

Синапсы отличаются друг от друга размерами и возможностями концентрации нейромедиатора.

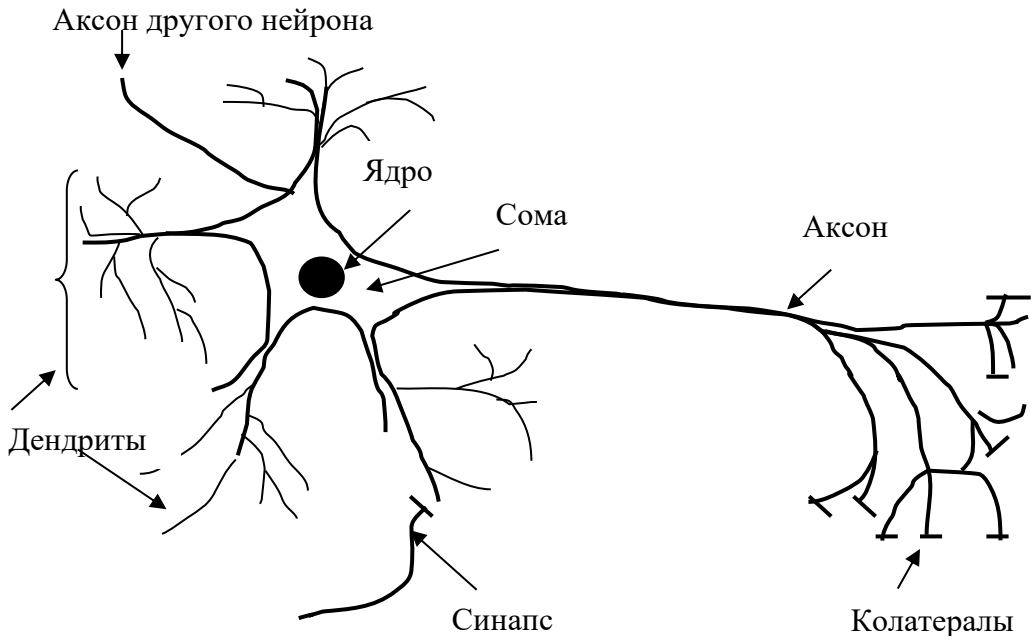

Рис. 1.1 Упрощенная структура биологической нервной клетки

Поэтому импульсы одинаковой величины, поступающие на входы нервной клетки через различные синапсы, могут в разной степени изменять ее энергетический потенциал. Мерой изменения потенциала считается уровень поляризации мембраны, зависящий от суммарного количества нейромедиатора, выделенного на всех синапсах.

В результате поступления входных импульсов на конкретные синапсы происходит изменение электрического потенциала клетки. Если отклонение от состояния электрического равновесия невелико, клетка возвращается в исходное состояние и на ее выходе сигнал не регистрируется. В этом случае считается, что уровень изменения потенциала ниже порога ее срабатывания. Если суммарное изменение потенциала превысило порог активации клетки, значение выходного сигнала начинает нарастать, приобретая характер нервного импульса, пересылаемого аксоном на другие нейроны, подключенные к данной клетке (рис.1.2). Величина этого сигнала не зависит от степени превышения порога срабатывания.

Количество взаимодействующих друг с другом нервных клеток в человеческом мозге оценивается, как 10^{11}- 10^{14}. Каждая нервная клетка выполняет функцию суммирования весовых коэффициентов входных

сигналов и сравнивает полученную сумму с пороговым значением. Каждый нейрон имеет свои веса и свои пороговые значения. Громадное количество нейронов и межнейронных связей (до 1000 входов в каждый нейрон) приводит к тому, что ошибка в срабатывании отдельного нейрона остается незаметной в общей массе взаимодействующих клеток.

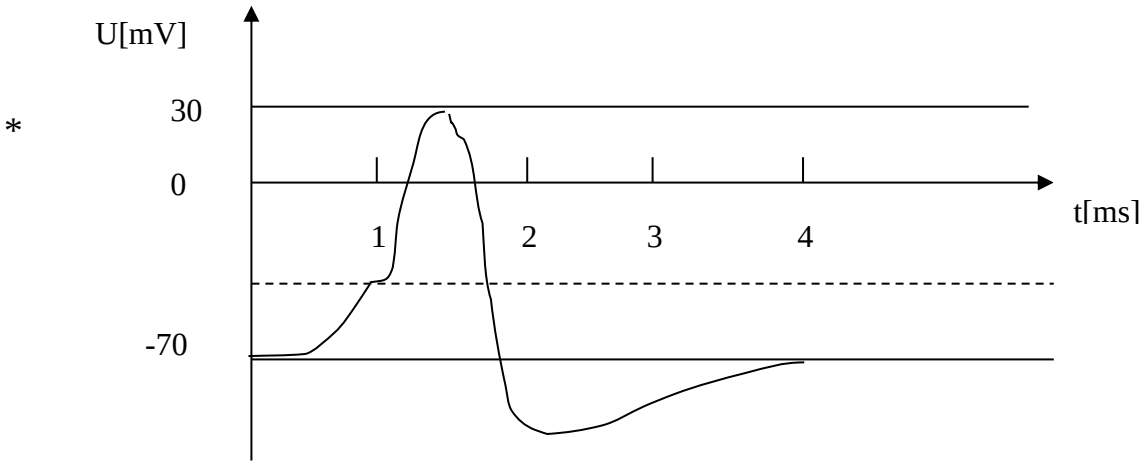

Рис. 1.2 Типичная форма нервного импульса

Существует огромное количество форм и размеров нейронов в зависимости от того, в какой части мозга он находятся. Самыми распространёнными нейронами коры головного мозга являются пирамидальные нейроны.

Следует отметить, что ни одна современная технология не позволяет построить искусственную нейронную сеть, близкую по масштабам к нейронной сети мозга. Однако изучение и копирование биологических нервных систем, позволяет надеяться на создание нового поколения электронных устройств, имеющих аналогичные характеристики.

1.3 Модель МакКаллока - Питса

Нейрон является единицей обработки информации в нейронной сети. Из приведенных выше рассуждений следует, что каждый нейрон суммирует с соответствующими весами сигналы, приходящие от других нейронов, выполняет нелинейную решающую функцию и передает результат связанным с ним другим нейронам. В простейших моделях нейронов выходной сигнал принимает двоичные значения: 0 или 1. Значение 1 соответствует превышению порогового уровня, значение 0 – в противном случае. Одна из первых моделей нейрона была предложена Дж. МакКаллоком и У. Питсом в 1943 году [4]. Структурная схема этой модели представлена на рис. 1.3.

Сигналы x_j на входе синапсов j $(j = 1,2,...,N)$, связанные с нейроном i, суммируются с учетом соответствующих синаптических весов w_{ij} (первый индекс относится к нейрону, а второй к синапсу), после чего результат сравнивается с пороговым значением w_{i0}.

Пороговое значение отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации, которая ограничивает амплитуду выходного сигнала. Выходной сигнал нейрона y_i определяется при этом зависимостью

$$y_j = f\left(\sum_{J=1}^{N} w_{ij} x_j(t) + w_{i0}\right) \qquad (1.1)$$

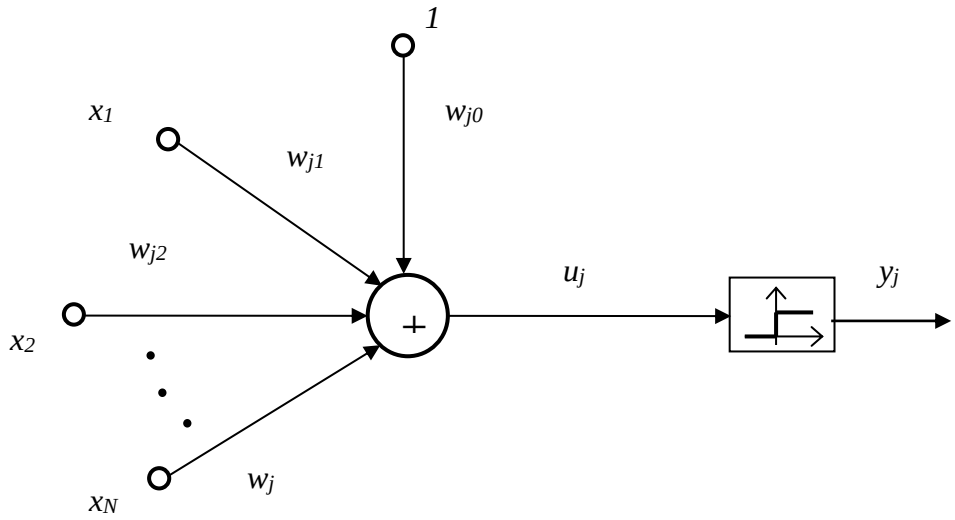

Рис. 1.3 Модель нейрона МакКаллока-Питса

Аргументом функции выступает суммарный сигнал, формируемый сумматором искусственного нейрона.

$$u_i = \sum_{j=1}^{N} w_{ij} x_j(t) + w_{i0}. \qquad (1.2)$$

Коэффициенты w_{ij} в формуле (1.1) представляют веса синапсов. Положительные значения w_{ij} соответствует синапсам, повышающим потенциал, отрицательные значения – синапсам, понижающим потенциал, $w_{ij} = 0$ свидетельствует об отсутствии связи между *i*-м и *j*-м нейронами.

Использование порогового сигнала w_{i0} обеспечивает эффект аффинного преобразования выхода линейного сумматора u_i .

Модель МакКаллока – Питса – это дискретная модель, в которой состояние нейрона в момент *(t+1)* рассчитывается по значению его входных сигналов в момент времени *t*.

Функция $f(u_i)$ называется функцией активации. В модели МакКаллока – Питса это пороговая функция вида:

$$f(u) = \begin{cases} 1 \ для\ u > 0 \\ 0 \ для\ u \le 0 \end{cases}. \qquad (1.3)$$

В общем случае эта функция активации описывается следующим выражением:

$$\mathrm{sgn}(x)=\begin{cases} b, \textit{ если } x>0; \\ c, \textit{ если } x\le 0, \end{cases} \qquad (1.4)$$

где b и c – некоторые постоянные. На практике чаще всего используют две пары постоянных b и c: первая (-1,1); вторая – (0,1). Первая пара коэффициентов определяет так называемую симметричную пороговую функцию, вторая – смещенную.

1.4 Персептрон

Ф. Розенблатт в 1958 году ввел понятие персептрона как первой модели обучения с учителем [4]. Обучение персептрона требует наличие учителя и состоит в таком подборе весов w_{ij} ,чтобывыходной сигнал y_i был наиболее близок к заданному значению d_i. При таком способе обучения, каждой обучающей выборке, представленной вектором $\boldsymbol{x}$ поставлено в соответствии ожидаемое значение d_i на выходе i-го нейрона.

Наиболее популярный метод обучения персептрона, называемый правилом персептрона, состоит в подборе весовых коэффициентов по следующему алгоритму:

- при первоначально выбранных (как правило, случайным образом) значениях весов w_{ij} на вход нейрона подается обучающий вектор x и рассчитывается значение выходного сигнала y_i. По результатам сравнения значения y_i с заданным значением d_i уточняются значения весов;
- если y_i совпадает с ожидаемым значением d_i, то весовые коэффициенты w_{ij} не изменяются;
- если $y_i=0$, а соответствующее значение $d_i=1$, то значения весов уточняются по формуле $w_{ij}(t+1)=w_{ij}(t)+x_j$, где $(t+1)$– это номер текущего цикла, а t – номер предыдущего цикла;
- если $y_i=1$, а соответствующее значение $d_i=0$, то значения весов уточняются по формуле $w_{ij}(t+1)=w_{ij}(t)-x_j$, *где* $(t+1)$– это номер текущего цикла, а t – номер предыдущего цикла;

По завершении уточнения весов предоставляются очередной обучающий вектор x и связанное с ним значение d_i, и значения весов уточняются заново. Этот процесс повторяется для всех обучающих выборок, пока не будут минимизированы различия между всеми значениями y_i и соответствующими им значениями d_i.

1.5 Сигмоидальный нейрон

Нейрон сигмоидального типа имеет структуру, подобную модели МакКаллока–Питса, с той разницей, что функция активации является непрерывной и может быть выражена в виде сигмоидальной униполярной или биполярной функции [4]. Структура нейрона представлена на рис. 1.4.

Входные сигналы $x_j (j=1,2,...,N)$ суммируются с учетом соответствующих весов w_{ij} (сигнал поступает в направлении от узла j к узлу i) в сумматоре, после чего результат сравнивается с пороговым значением w_{i0}. Выходной сигнал нейрона y_i определяется при этом зависимостью

$$y_i = f\left(\sum_{j=1}^{N} w_{ij} x_j(t) + w_{i0} \right). \qquad (1.9)$$

Аргументом функции выступает суммарный сигнал $u_i = \sum_{j=1}^{N} w_{ij} x_j(t) + w_{i0}$.

Функция $f(u_i)$, называемая функцией активации, относится к классу непрерывных, монотонно возрастающих и дифференцируемых функций. Нейрон сигмоидального типа использует сигмоидальную униполярную (логистическую) или сигмоидальную биполярную (гиперболический тангенс) функцию активации.

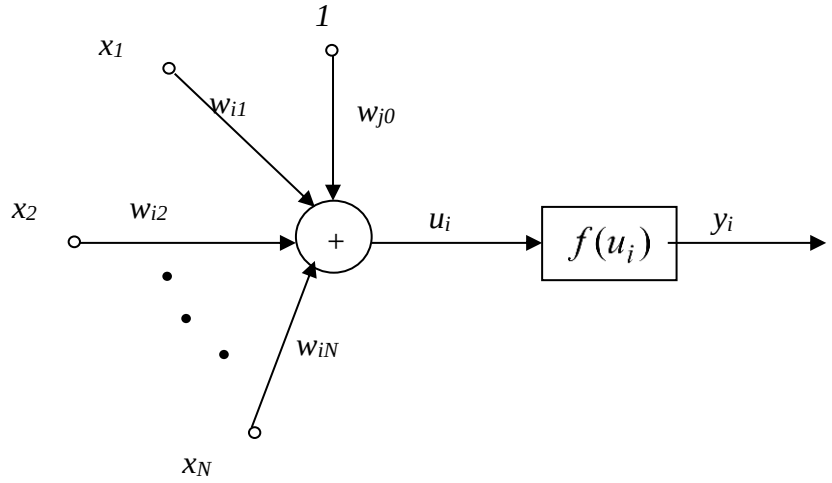

Рис. 1.4 Модель сигмоидального нейрона

Униполярная функция, как правило, представляется формулой

$$f(x) = \frac{1}{1+e^{-kx}}, \qquad (1.10)$$

тогда как биполярная функция задается в виде (1.11) или (1.12):

$$f(x) = \tanh(kx). \qquad (1.11)$$

$$f(x) = \frac{e^{kx} - e^{-kx}}{e^{kx} + e^{-kx}}. \qquad (1.12)$$

Графики сигмоидальных функций при k=1 представлены на рис. 1.5.

Отметим, что, как правило, современные компьютеры вычисляют функцию гиперболического тангенса быстрее, чем логистическую. Другое преимущество функции гиперболического тангенса состоит в том, что она изменяется в диапазоне от –1 до +1. Часто бывает необходимо нормировать обучающий набор данных таким образом, чтобы среднее значение было равно 0 при единичном стандартном отклонении.

Такая нормировка возможна только с функцией активации, которая способна принимать отрицательные значения. И наконец, нечетная функция, такая, как гиперболический тангенс, обеспечивает более быстрое обучение, чем несимметричная логистическая функция.

В этих формулах параметр k подбирается пользователем. Его значение влияет на форму функции активации. При малых значениях k график функции достаточно пологий, по мере роста значения kкрутизна графика увеличивается.

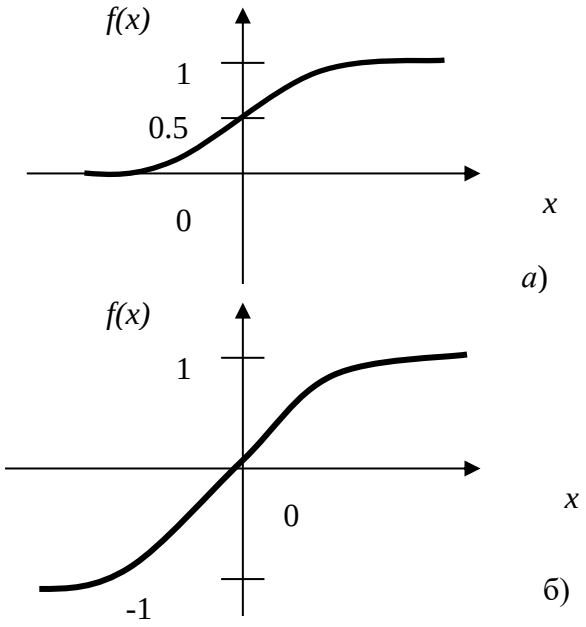

Рис. 1.5 Графики сигмоидальных функций:
а – логистическая; б – гиперболический тангенс

При $k \to \infty$ сигмоидальная функция превращается в пороговую функцию, идентичную функции активации персептрона. На практике чаще всего для упрощения используется значение k =1.

Важным свойством сигмоидальной функции является ее дифференцируемость. Для униполярной функции имеем

$$\frac{df(x)}{dx} = kf(x)(1 - f(x)), \qquad (1.13)$$

тогда как для биполярной функции

$$\frac{df(x)}{dx} = k(1 - f^2(x)). \qquad (1.14)$$

И в первом, и во втором случае график изменения производной относительно переменной x имеет колоколообразную форму, а его максимум соответствует значению $x=0$.

Сигмоидальный нейрон, как правило, обучается с учителем.

При обучении с учителем предполагается, что помимо входных сигналов, составляющих вектор x, известны также и ожидаемые выходные

сигналы нейрона d_i, составляющие вектор d. В подобной ситуации подбор весовых коэффициентов должен быть организован так, чтобы фактические выходные сигналы нейрона y_i принимали бы значения, как можно более близкие к ожидаемым значениям d_i. Ключевым элементом процесса обучения с учителем является знание ожидаемых значений d_i выходного сигнала нейрона.

При обучении с учителем производится минимизация целевой функции, которая для единичного обучающего кортежа <x, d>i-го нейрона определяется в виде

$$E = \frac{1}{2}(y_i - d_i)^2, \tag{1.15}$$

где

$$y_i = f(u_i) = f\left(\sum_{j=0}^{N} w_{ij} x_j\right). \tag{1.16}$$

Применение непрерывной функции активации позволяет использовать при обучении градиентные алгоритмы.

1.6 Нейрон типа WTA

В соответствии с принципами функционирования биологических нейронов созданы различные математические модели, которыми в большей или меньшей степени реализуются свойства природной нервной клетки. Обобщенная схема, составляющая основу большинства таких моделей, восходит к представленной на рисунке 1.3 модели МакКаллока-Питса, содержащий сумматор взвешенных входных сигналов и нелинейный блок выработки выходного сигнала нейрона, функционально зависящего от выходного сигнала сумматора. Однако, существуют и другие модели нейронов существенно отличающиеся от модели МакКаллока–Питса.

Рассмотрим более подробно нейроны типа *WTA*(*winnertakesall* – победитель получает все) [4]. Эти нейроны имеют входной модуль в виде стандартного сумматора, рассчитывающего сумму входных сигналов с соответствующими весами w_{ij}. Выходной сигнал i-го сумматора определяется согласно формуле:

$$u_i = \sum_{j=0}^{N} w_{ij} x_j \tag{1.17}$$

Группа конкурирующих между собой нейронов получает одни и те же входные сигналы x_j. Выходные сигналы нейронов u_i сравниваются между собой, и по результатам сравнения победителем признается нейрон, значение выходного сигнала у которого оказалось наибольшим. Нейрон-победитель вырабатывает на своем выходе состояние 1, а остальные нейроны переходят в состояние 0. Для обучения нейронов типа *WTA* не требуется учитель. На начальном этапе случайным образом выбираются весовые коэффициенты

каждого нейрона, нормализуемые относительно 1. После подачи первого входного вектора ***x*** определяется победитель этапа. Победивший нейрон переходит в состояние 1, что позволяет провести уточнение весов его входных линий по следующему правилу:

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \dot{\eta}\left(x_j - w_{ij}(t)\right) \qquad (1.18)$$

Проигравшие нейроны не изменяют свои весовые коэффициенты.

Схема соединения нейронов типа *WTA* изображена на рис.1.6.

На функционирование нейронов типа *WTA* оказывает существенное влияние нормализация входных векторов и весовых коэффициентов. Выходной сигнал *i*-го нейрона может быть описан векторным отношением:

$$u_i = w^T x = \|w\|\|x\|\cos\varphi_i \qquad (1.19)$$

Поскольку $\|w\| = \|x\| = 1$, значение выходного сигнала определяется углом между векторами *x* и *w*. Поэтому победителем оказывается нейрон, вектор весов которого оказывается наиболее близким текущему обучаемому вектору.

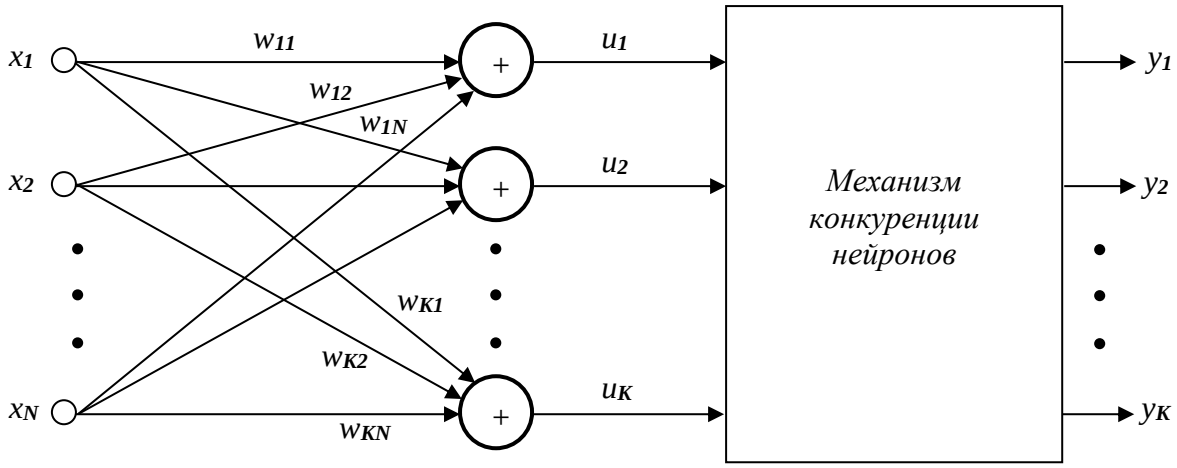

Рис. 1.6 Схема соединения нейронов типа WTA

В результате победы нейрона уточняются его весовые коэффициенты, значения которых приближаются к значениям вектора *x*. Проигравшие нейроны не изменяют свои веса. Следствием такой конкуренции становится самоорганизация процесса обучения.

1.7 Функции активации нейронов

Вид функции активации во многом определяет функциональные возможности нейронной сети и метод обучения этой сети. Перечень наиболее известных функций активации представлен в таблице 1.1.

Табл. 1.1

Примеры функций активации

Название	Формула	Область значений
Линейная	$f(x)=kx$	$(-\infty,\infty)$
Полулинейная	$f(x)=\begin{cases} kx, & x>0 \\ 0, & x\le 0 \end{cases}$	$(0,\infty)$
Линейная с насыщением	$f(x)=\begin{cases} -1, x\le -1 \\ x, -1<x<1 \\ 1, x\ge 1 \end{cases}$	$(-1,1)$
Логистическая	$f(x)=\frac{1}{1+e^{-kx}}$	$(0,1)$
Гиперболический тангенс	$f(x)=\frac{e^{kx}-e^{-kx}}{e^{kx}+e^{-kx}}$	$(-1,1)$
Рациональная	$f(x)=\frac{x}{k+\lvert x\rvert}$	$(-1,1)$
Синусоидальная	$f(x)=\sin(x)$	$(-1,1)$
Экспоненциальная	$f(x)=e^{-kx}$	$(0,\infty)$
Гаусса	$f(x)=\exp\left(-\frac{\lVert x-c_i\rVert^2}{2\sigma_i^2}\right)$	$(-\infty,\infty)$
Пороговая	$f(x)=\begin{cases} 0, x<0 \\ 1, x\ge 0 \end{cases}$	$(0,1)$
Модульная	$f(x)=\lvert x\rvert$	$(0,\infty)$
Знаковая (сигнатурная)	$f(s)=\begin{cases} 1, x>0 \\ -1, x\le 0 \end{cases}$	$(-1,1)$
Квадратичная	$f(x)=x^2$	$(0,\infty)$

2 Многослойные сети прямого распространения сигнала

2.1 Однослойный персептрон

Однослойный персептрон образуют нейроны, расположенные в одной плоскости (рисунок 2.1). Каждый нейрон имеет поляризатор, то есть единичный сигнал, который с весом w_{i0} поступает на вход нейрона, а также множество связей с весами w_{ij}, по которым поступают входные сигналы x_j.

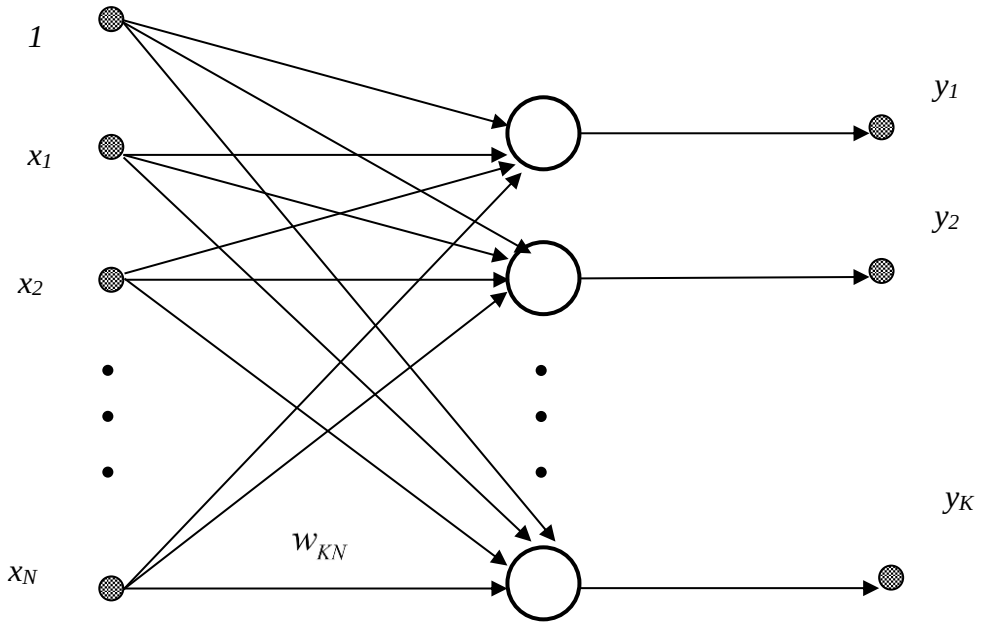

Рисунок 2.1 Структура однослойного персептрона

Значения весов подбираются в процессе обучения сети путём приближения реальных выходных значений y_i к эталонным значениям d_i. Мерой близости является значение целевой функции.При использовании p обучающих векторов $\langle x, d \rangle$ для сети из K нейронов, целевую функцию можно определить следующим образом:

$$E = \frac{1}{2} \sum_{m=1}^{p} \sum_{i=1}^{K} \left(y_i^{(m)} - d_i^{(m)} \right)^2 \qquad (2.1)$$

Выходные сигналы y_i являются функциями весов сети w_{ij} и уточняются в процессе обучения в соответствии с критерием минимизации целевой функции (2.1).

Расположенные на одном уровне нейроны функционируют независимо друг от друга, поэтому возможности однослойного персептрона определяются возможностями отдельных нейронов.Каждый нейрон реализует функциональное отображение $y_i = f\left(\sum_{j=0}^{N} w_{ij} x_j \right)$, где f -

сигмоидальная функция, поэтому значение выходного сигнала будет зависеть от знака выражения $\sum_{j=0}^{N} w_{ij} x_j$.

Выходной сигнал y_i при фиксированных значениях весов зависит от входного вектора x, который определяет гиперплоскость, разделяющую многомерное входное пространство на два подпространства. Вследствие этого, задача классификации (приписывание выходному сигналу значения 1 или 0), может быть решена одним нейроном, если это задача линейной разделимости классов. Добавление нейронов в единственный слой сети не позволяет улучшить её функциональные возможности. Вследствие этого, однослойная сеть имеет небольшое практическое значение. Добавление ещё одного слоя нейронов, даже состоящего из единственного нейрона позволяет существенно расширить возможнойти сети.

2.2 Многослойный персептрон

В настоящее время наиболее часто используемой архитектурой нейросети является *многослойный персептрон (MLP),* который представляет собой обобщение однослойного персептрона.

Обычно сеть состоит из множества входных узлов, которые образуют *входной слой*; одного или нескольких *скрытых слоев* вычислительных нейронов и одного *выходного слоя*. Входной сигнал распространяется по сети в прямом направлении от слоя к слою. Многослойные персептроны успешно применяются для решения разнообразных сложных задач. При этом обучение с учителем выполняется с помощью такого популярного алгоритма, как *алгоритм обратного распространения ошибки*.

Многослойный персептрон имеет три отличительных признака:

1. Каждый нейрон имеет *нелинейную функцию активации*. Данная функция должна быть гладкой (то есть всюду *дифференцируемой*). Самой популярной гладкой функцией активации является сигмоидальная функция.
2. Сеть содержит один или несколько слоев *скрытых нейронов*. Эти нейроны позволяют сети обучаться решению сложных задач, последовательно извлекая наиболее важные признаки из входного вектора.
3. Сеть обладает высокой степенью *связности*, реализуемой посредством синаптических соединений.

Структура многослойного персептрона с двумя скрытыми слоями изображена на рис. 2.2 [4]. Показанная на рисунке сеть является *полносвязной*, что характерно для многослойного персептрона. Это значит, что каждый нейрон любого слоя связан со всеми нейронами предыдущего слоя. Сигнал передается по сети в прямом направлении слева направо.

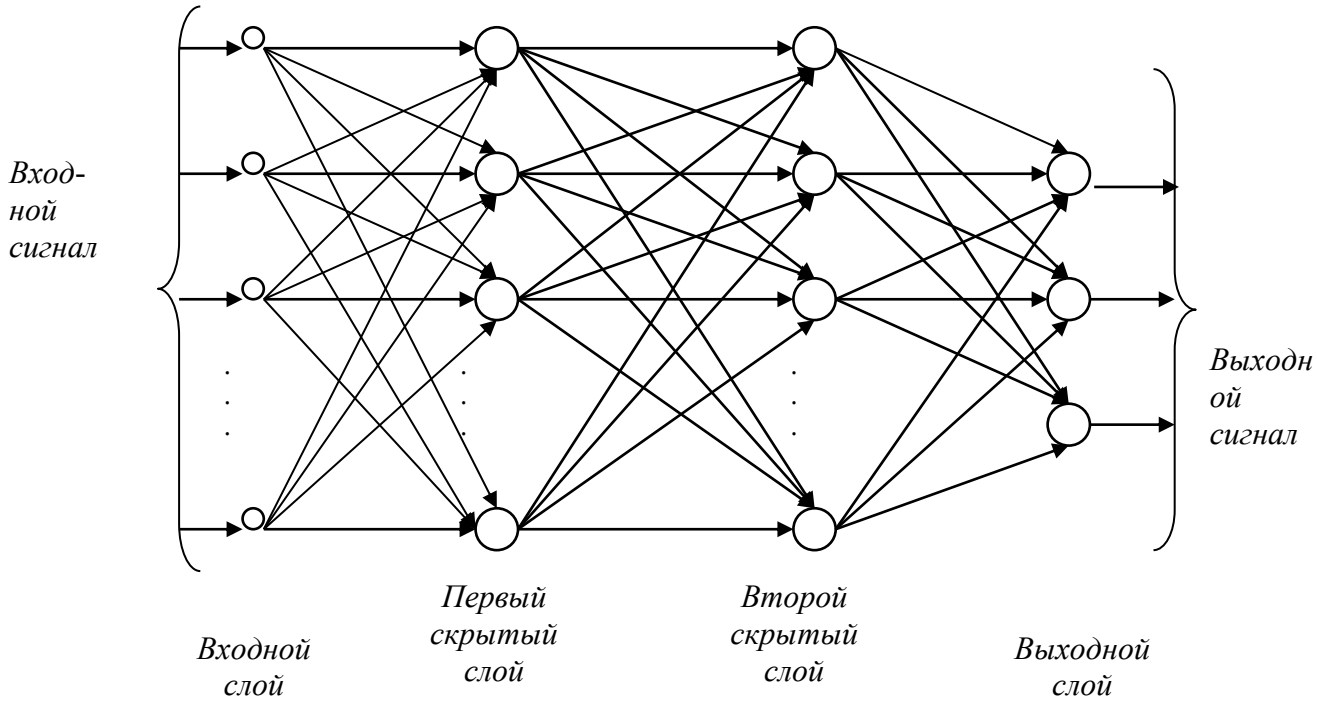

Рисунок 2.2 Структура многослойного персептрона с двумя скрытыми слоями

Для многослойного персептрона выделяют два типа сигналов:

1. *Функциональный сигнал* – это входной сигнал сети, передаваемый в прямом направлении по всей сети. Такой сигнал достигает конца сети в виде выходного сигнала. В каждом нейроне, через который передается этот сигнал, вычисляется некоторая функция от взвешенной суммы его входов с поправкой в виде порогового элемента - единичного сигнала с весовым коэффициентом w_{io}.
2. *Сигнал ошибки* – берет своё начало на выходе сети и распространяется в обратном направлении от слоя к слою. Вычисляется каждым нейроном на основе функции ошибки, представленной в той или иной форме.

Выходные нейроны составляют выходной слой сети. Остальные нейроны относятся к скрытым слоям. Первый скрытый слой получает данные из входного слоя. Результирующий сигнал первого скрытого слоя, в свою очередь, поступает на следующий скрытый слой и так далее, до самого конца сети.

В сетях подобного типа используются, в основном, сигмоидальные нейроны. Такую сеть легко можно интерпретировать как модель вход-выход, в которой веса и пороговые значения являются свободными параметрами модели. Такая сеть может моделировать функцию практически любой степени сложности, причем число слоев и число нейронов в каждом слое определяют сложность функции.

В многослойных сетях эталонные значения выходных сигналов известны, как правило, только для нейронов выходного слоя, поэтому сеть

невозможно обучить, руководствуясь только величинами ошибок на выходе нейросети.

Один из вариантов решения этой проблемы – разработка учебных примеров для каждого слоя нейросети, что является очень трудоемкой операцией и не всегда осуществимо.

Второй вариант – динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод "тыка", несмотря на свою кажущуюся простоту, требует громоздких рутинных вычислений.

И, наконец, третий, более приемлемый вариант – распространение сигналов ошибки от выходов нейросети к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения нейросети получил название процедуры обратного распространения. Разработка алгоритма обратного распространения для определения весов в многослойном персептроне сделала эти сети наиболее популярными у исследователей и пользователей нейронных сетей.

Основными достоинствами многослойного персептрона являются простота в использовании, гарантированное получение ответа после прохода данных по слоям, хорошо апробированные и широко применяемые алгоритмы обучения, способность моделирования функции любой степени сложности.

С другой стороны, существует множество спорных вопросов при проектировании сетей прямого распространения. Например, сколько скрытых слоев необходимо для решения данной задачи, сколько следует выбрать нейронов в каждом слое, как сеть будет реагировать на данные, не включенные в обучающую выборку, и какой размер обучающей выборки необходим для достижения "хорошей" обобщающей способности сети.

2.3 Структура двухслойной сигмоидальной нейронной сети

На рис. 2.3 представлена сеть с одним скрытым слоем. Все последующие рассуждения относятся к сетям именно такого типа. Обозначения сигналов и весов также будут соответствовать этому рисунку. Веса нейронов скрытого слоя пометим верхним индексом (1), а выходного слоя – верхним индексом (2). Выходные сигналы нейронов скрытого слоя обозначим v_i (i=0, 1, 2,…, K), а выходного слоя y_s (s=1, 2, …, M).

Пусть функция активации нейронов задана в сигмоидальной униполярной или биполярной форме. Для упрощения описания будем использовать расширенное обозначение входного вектора сети в виде $x = [x_0, x_1, ..., x_N]^T$, где $x_0 = 1$ соответствует единичному сигналу порогового элемента.

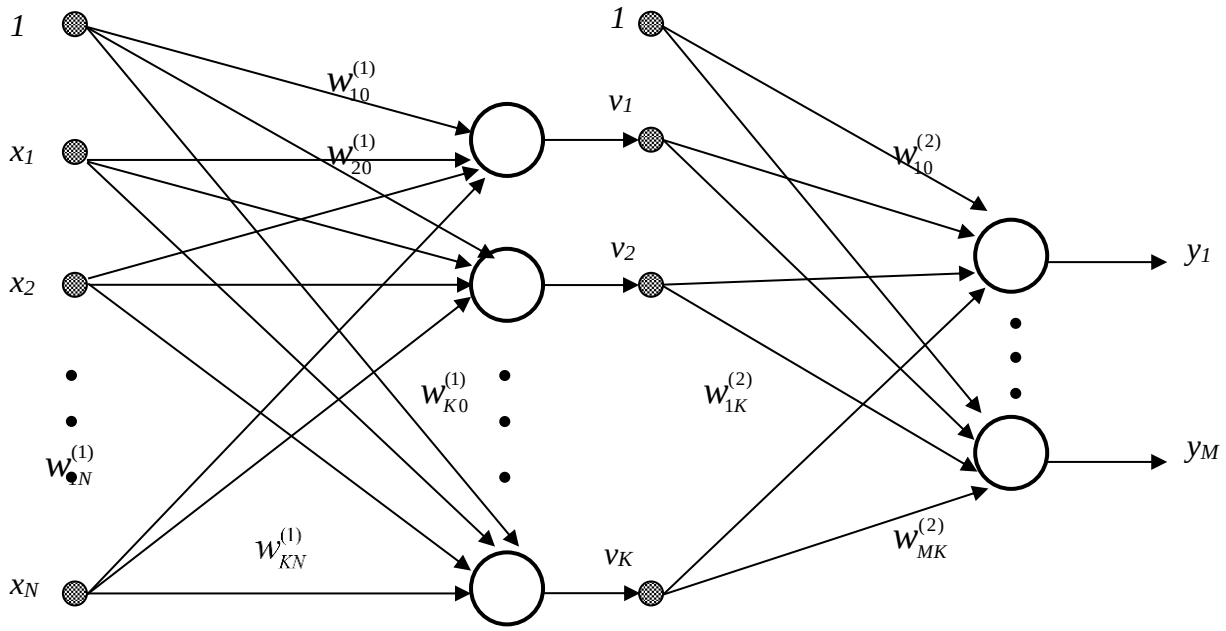

Рисунок 2.3 Обобщенная структура двухслойной сигмоидальной нейронной сети

С вектором x связаны два выходных вектора сети: вектор фактических выходных сигналов $y = \left[y_{0,} y_1,...,y_M\right]^T$ и вектор ожидаемых выходных сигналов $d = \left[d_0, d_1,...,d_M\right]^T$.

Цель обучения состоит в подборе таких значений весов $w_{ij}^{(1)}$ и $w_{si}^{(2)}$ для двух слоев сети, чтобы при заданном входном векторе x получить на выходе значения сигналов y_s, которые с требуемой точностью будут совпадать с ожидаемыми значениями d_s для s=1, 2, …, M.

Если рассматривать единичный сигнал порогового элемента как один из компонентов входного вектора *x*, то веса пороговых элементов можно добавить в векторы весов соответствующих нейронов обоих слоев. При таком подходе выходной сигнал i-го нейрона скрытого слоя удается описать функцией

$$v_i = f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right), \qquad (2.2)$$

в которой индекс 0 соответствует сигналу и весам пороговых элементов, причем $v_0 \equiv 1,\ x_0 \equiv 1$. В выходном слое *s*-ый нейрон вырабатывает выходной сигнал, определяемый как

$$y_s = f\left(\sum_{i=0}^{K} w_{si}^{(2)} v_i\right) = f\left(\sum_{i=0}^{K} w_{si}^{(2)} f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right)\right). \qquad (2.3)$$

Из формулы (2.3) следует, что на значение выходного сигнала влияют веса обоих слоев, тогда как сигналы, вырабатываемые в скрытом слое, не зависят от весов выходного слоя.

Для того чтобы сеть можно было применять в дальнейшем, ее прежде надо обучить на полученных ранее данных, для которых известны и значения входных параметров, и правильные ответы на них. Это обучение состоит в подборе весов межнейронных связей, обеспечивающих наибольшую близость ответов сети к известным правильным ответам.

Определение числа промежуточных слоев и числа элементов в них является важным вопросом при конструировании многослойного персептрона.

2.4 Градиентные методы обучения многослойного персептрона

2.4.1 Основные положения градиентных алгоритмов обучения сети

Задачу обучения нейронной сети будем рассматривать, как требование минимизировать априори определенную целевую функцию *E(w)*. При таком подходе можно применять для обучения алгоритмы, которые в теории оптимизации считаются наиболее эффективными. К ним, без сомнения, относятся градиентные алгоритмы, чью основу составляет выявление градиента целевой функции. Они связаны с разложением целевой функции *E(w)* в ряд Тейлора в ближайшей окрестности точки имеющегося решения *w*. В случае целевой функции от многих переменных ($w = [w_1, w_2, ..., w_n]^T$) такое представление связывается с окрестностью ранее определенной точки (в частности, при старте алгоритма это исходная точка w_0) в направлении *p*. Подобное разложение описывается универсальной формулой вида

$$E(w+p) = E(w) + [g(w)]^T p + \frac{1}{2} p^T H(w) p + ..., \qquad (2.4)$$

где $g(w) = \nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_n} \right]^T$ - это вектор градиента, а симметричная квадратная матрица

$$H(w) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1 \partial w_1} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & \cdots & \frac{\partial^2 E}{\partial w_n \partial w_n} \end{bmatrix}$$

является матрицей производных второго порядка, называемой гессианом [4].

В выражении (2.4) *p* играет роль направляющего вектора, зависящего от фактических значений вектора *w*. На практике чаще всего рассчитываются три первых члена ряда (2.4), а последующие члены ряда просто игнорируются. При этом зависимость (2.4) может считаться квадратичным приближением целевой функции *E(w)* в ближайшей окрестности найденной точки *w* с точностью, равной локальной погрешности отсеченной части $O(h^3)$, где *h*= ||*p*||. Для упрощения описания значения переменных,

полученных в *t*-ом цикле, будем записывать с нижним индексом *t*. Точкой решения $w = w_t$ будем считать точку, в которой достигается минимум целевой функции *E(w)* и $g(w_t) = 0$, а гессиан $H(w_t))$ является положительно определенным.

В процессе поиска минимального значения целевой функции направление поиска *p* и шаг *h* подбираются таким образом, чтобы для каждой очередной точки $w_{t+1} = w_t + \eta_t p_t$ выполнялось условие $E(w_{t+1}) < E(w_t)$. Поиск минимума продолжается, пока норма градиента не упадет ниже априори заданного значения допустимой погрешности либо пока не будет превышено максимальное время вычислений (количество итераций).

Универсальный оптимизационный алгоритм обучения нейронной сети можно представить в следующем виде (будем считать, что начальное значение оптимизируемого вектора известно и составляет $w_t = w_0$) [4]:

1. Проверка сходимости и оптимальности текущего решения w_t. Если точка w_t отвечает градиентным условиям остановки процесса – завершение вычислений. В противном случае перейти к п.2.
2. Определение вектора направления оптимизации p_t для точки w_t.
3. Выбор величины шага η_t в направлении p_t, при котором выполняется условие $E(w_t + \eta_t p_t) < E(w_t)$.
4. Определение нового решения $w_{t+1} = w_t + \eta_t p_t$, а также соответствующих ему значений $E(w_t)$ и $g(w_t)$, а если требуется – то и $H(w_t)$, и возврат к п.1.

2.4.2 Подбор коэффициента обучения

После правильно выбранного направления p_t, в градиентных алгоритмах обучения, следует определить новую точку решения w_{t+1}, в которой будет выполняться условие $E(w_{t+1}) < E(w_t)$. Необходимо подобрать такое значение η_t, чтобы новое решение $w_{t+1} = w_t + \eta_t p_t$ лежало как можно ближе к минимуму функции *E(w)* в направлении p_t. Грамотный подбор коэффициента η_t оказывает огромное влияние на сходимость алгоритма оптимизации к минимуму целевой функции. Чем сильнее величина η_t отличается от значения, при котором *E(w)* достигает минимума в выбранном направлении p_t, тем большее количество итераций потребуется для поиска оптимального решения. Слишком малое значение η_t не позволяет минимизировать целевую функцию за один шаг и вызывает необходимость повторно двигаться в том же направлении. Слишком большой шаг приводит к «перепрыгиванию» через минимум функции и фактически заставляет возвращаться к нему.

Существуют различные способы подбора значений η_t, называемого в теории нейронных сетей коэффициентом обучения. Простейший из них

основан на фиксации постоянного значения η_t на весь период оптимизации. Этот способ практически используется только совместно с методом наискорейшего спуска. Он имеет низкую эффективность, поскольку значение коэффициента обучения никак не зависит от вектора фактического градиента и, следовательно, от направления *p* на данной итерации. Величина η подбирается, как правило, раздельно для каждого слоя сети с использованием различных эмпирических зависимостей. Один из походов состоит в определении минимального значения ηдля каждого слоя по формуле

$$\eta \le \min\left(\frac{1}{n_i}\right), \tag{2.5}$$

где n_i обозначает количество входов i-го нейрона в слое.

В [4] предложена следующая адаптивная формула для корректировки значений коэффициента обучения в зависимости от итерации обучения:

$$\eta(t) = \frac{\eta_0}{1+\left(\frac{t}{\tau}\right)}, \tag{2.6}$$

где η_0 - начальное значение коэффициента обучения, τ - константа времени поиска, задаваемые пользователем, а t - номер итерации обучения.

При достаточно больших значениях η_0 и при большом числе итераций, значительно превосходящих константу τ, алгоритм обучения будет вести себя как стохастический алгоритм аппроксимации, а веса будут сходиться к своим оптимальным значениям.

2.4.3 Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки определяет стратегию подбора весов многослойной сети с применением градиентных методов оптимизации. В настоящее время считается одним из наиболее эффективных алгоритмов обучения многослойной сети. При обучении ставится задача минимизации целевой функции, формируемой, как правило, в виде квадратичной суммы разностей между фактическими и ожидаемыми значениями выходных сигналов, которая для P обучающих выборок определяется по формуле:

$$E(w) = \frac{1}{2}\sum_{t=1}^{P}\sum_{s=1}^{M}\left(y_s^{(t)} - d_s^{(t)}\right)^2 \tag{2.7}$$

В случае единичной обучающей выборки *(x,d)* целевая функция имеет вид:

$$E(w) = \frac{1}{2}\sum_{s=1}^{M}\left(y_s - d_s\right)^2 \tag{2.8}$$

Уточнение весов может проводиться после предъявления каждой обучающей выборки (так называемый режим «онлайн»), при этом используется целевая функция вида (2.8), либо однократно после предъявления всех обучающих выборок (режим «оффлайн»), при этом

используется целевая функция вида (2.7). В последующем изложении используется целевая функция вида (2.8).

Для упрощения можно считать, что цель обучения состоит в таком определении значений весов нейронов каждого слоя сети, чтобы при заданном входном векторе получить на выходе значения сигналов y_s, совпадающие с требуемой точностью с ожидаемыми значениями d_s при s= 1, 2,…,M.

Обучение сети с использованием алгоритма обратного распространения ошибки проводится в несколько этапов.

На первом из них предъявляется обучающая выборка *x* и рассчитываются значения сигналов соответствующих нейронов сети. При заданном векторе x определяются вначале значения выходных сигналов v_i скрытого слоя, а затем значения y_s выходного слоя. Для расчета применяются формулы (2.1) и (2.2). После получения значений выходных сигналов y_s становится возможным рассчитать фактическое значение целевой функции ошибки E(w).

На втором этапе минимизируется значение этой функции.

Так как целевая функция непрерывна, то наиболее эффективными методами обучения оказываются градиентные алгоритмы, согласно которым уточнение вектора весов (обучение) производится по формуле:

$$w(t+1) = w(t) + \Delta w, \qquad (2.9)$$

где $\Delta w = \eta\, p(w), \qquad (2.10)$

η - коэффициент обучения, а *p(w)* – направление в многомерном пространстве *w*. В алгоритме обратного распространения ошибки *p(w)* определяется как частная производная $\frac{\partial E}{\partial w_{ij}}$, взятая со знаком минус.

Обучение многослойной сети с применением градиентных методов требует определения вектора градиента относительно весов всех слоев сети, что необходимо для правильного выбора направления *p(w)*. Эта задача имеет очевидное решение только для весов выходного слоя. Для других слоев используется алгоритм обратного распространения ошибки, который определяется следующим образом [4]:

1. Подать на вход сети вектор *x* и рассчитать значения выходных сигналов нейронов скрытых слоев и выходного слоя, а также соответствующие производные $\frac{df(u_i^{(1)})}{du_i^{(1)}}, \frac{df(u_i^{(2)})}{du_i^{(2)}}, \ldots, \frac{df(u_i^{(m)})}{du_i^{(m)}}$ функций активации каждого слоя (m – количество слоев).
2. Создать сеть обратного распространения ошибок путем изменения направления передачи сигналов, замены функций активации их производными и подачи на бывший выход сети в качестве входного сигнала разности между фактическими y_s и ожидаемыми d_s значениями.

3. Уточнить веса по формулам (2.9) и (2.10) на основе результатов, полученных в п.1 и п.2 для исходной сети и для сети обратного распространения ошибки.
4. Пункты 1, 2, 3 повторить для всех обучающих выборок, вплоть до выполнения условия остановки: норма градиента станет меньше заданного значения ε, характеризующего точность обучения.

Рассмотрим основные расчетные формулы для сети с одним скрытым слоем, представленной на рисунке 2.2. Используется сигмоидальная функция активации, при этом в случае гиперболического тангенса производная функции активации равна

$$\frac{df(u)}{du} = 1 - f^2(u) \tag{2.11}$$

В случае логистической функции производная равна

$$\frac{df(u)}{du} = f(u) \cdot (1 - f(u)) \tag{2.12}$$

В формулах (2.11) и (2.12) под переменной *u* будем понимать выходные сигналы сумматоров нейронов скрытого или выходного слоя, представленных формулами (2.13) и (2.14).

$$u_i^{(1)} = \sum_{j=0}^{N} w_{ij}^{(1)} x_j \tag{2.13}$$

$$u_s^{(2)} = \sum_{i=0}^{K} w_{si}^{(2)} v_i \tag{2.14}$$

Уточнение весовых коэффициентов будем проводить после предъявления каждой обучающей выборки. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(m)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \tag{2.15}$$

$$w_{ij}^{(m)}(t+1) = w_{ij}^{(m)}(t) - \eta \cdot \frac{\partial E}{\partial w_{ij}^{(m)}} \tag{2.16}$$

Здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей *i*-ый нейрон слоя *m* с *j*-ым нейроном слоя *m-1*, η – коэффициент обучения, $0<\eta<1$.

С учетом принятых на рисунке 2.3 обозначений целевая функция для выходного слоя нейронов определяется следующим образом:

$$E = \frac{1}{2}\sum_{s=1}^{M}\left[f\left(\sum_{i=o}^{K} w_{si}^{(2)} v_i\right) - d_s \right]^2 = \frac{1}{2}\sum_{s=1}^{M}\left[f\left(\sum_{i=0}^{K} w_{si}^{(2)} f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_i\right)\right) - d_s \right]^2 \tag{2.17}$$

$$\frac{\partial E}{\partial w_{si}^{(2)}} = (y_s - d_s) \cdot \frac{df(u_s^{(2)})}{du_s^{(2)}} \cdot v_i \tag{2.18}$$

Здесь под y_s, как и раньше, подразумевается выход s –го нейрона.

Компоненты градиента относительно нейронов скрытого слоя описываются более сложной зависимостью:

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = \sum_{s=1}^{M} (y_s - d_s) \cdot \frac{dy_s}{dv_i} \cdot \frac{dv_i}{dw_{ij}^{(1)}} \qquad (2.19)$$

В другом виде эта зависимость может быть выражена формулой:

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = \sum_{s=1}^{M} (y_s - d_s) \cdot \frac{df(u_s^{(2)})}{du_s^{(2)}} \cdot w_{si}^{(2)} \frac{df(u_s^{(1)})}{du_i^{(1)}} x_j \qquad (2.20)$$

2.4.4 Алгоритм наискорейшего спуска

Если при разложении целевой функции E(w) в ряд Тейлора ограничиться ее линейным приближением, то мы получим алгоритм наискорейшего спуска. Для выполнения соотношения $E(w_{t+1}) < E(w_t)$ достаточно подобрать $g(w_t)^T p < 0$. Условию уменьшения значения целевой функции отвечает выбор вектора направления

$$p_t = -g(w_t). \qquad (2.21)$$

В этом случае коррекция весовых коэффициентов производится по формуле:

$$w_{ij}(t+1) = w_{ij}(t) + \eta p_t \qquad (2.22)$$

В другом виде формулу коррекции весов по методу наискорейшего спуска можно представить следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}(t)} \qquad (2.23)$$

Ограничение слагаемым первого порядка при разложении функции в ряд Тейлора, не позволяет использовать информацию о ее кривизне. Это обуславливает линейную сходимость метода. Указанный недостаток, а также резкое замедление минимизации в ближайшей окрестности точки оптимального решения, когда градиент принимает очень малые значения, делают алгоритм наискорейшего спуска низкоэффективным. Тем не менее, простота, невысокие требования к объему памяти и относительно невысокая вычислительная сложность, обуславливают широкое использование алгоритма. Повысить эффективность удается путем эвристической модификации выражения, определяющего направление градиента.

Одна из модификаций получила название алгоритма обучения с моментом. При этом подходе уточнение весов сети производится по формуле:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha(w_{ij}(t) - w_{ij}(t-1)) \qquad (2.24)$$

где α - это коэффициент момента, принимающий значения в интервале [0, 1].

Первое слагаемое в формуле (2.24) соответствует алгоритму наискорейшего спуска, а второе слагаемое учитывает последнее изменение весов и не зависит от фактического значения градиента. Чем больше значение коэффициента α, тем большее значение оказывает показатель момента на подбор весов. При постоянном значении коэффициента обучения $\eta(t)=\eta$ приращение весов остается примерно одинаковым, то есть $\Delta w_{ij}(t)=\eta p(t)+\alpha\,\Delta w_{ij}(t)$, поэтому эффективное приращение весов можно писать формулой:

$$\Delta w_{ij}(t)=\frac{\eta}{1-\alpha}\,p(t) \qquad (2.25)$$

При значении α=0,9 это соответствует десятикратному увеличению значения коэффициента обучения и, следовательно, десятикратному ускорению процесса обучения. При малых значениях градиента показатель момента начинает доминировать, что приводит к такому приращению весов, которое соответствует увеличению значения целевой функции, позволяющему выйти из зоны локального минимума. Однако показатель момента, не должен доминировать на протяжении всего процесса обучения, поскольку это приводит к нестабильности алгоритма. На практике, увеличение целевой функции не допускается больше, чем на 4%. В противном случае, $\Delta w_{ij}(t)=0$. При этом показатель градиента начинает доминировать над показателем момента и процесс развивается в направлении минимизации, заданном вектором градиента [2].

3 Сети с самоорганизацией на основе конкуренции

3.1 Сеть Кохонена

Основу самоорганизации нейронных сетей составляет подмеченная закономерность, что глобальное упорядочение сети становится возможным в результате *самоорганизующих* операций, независимо друг от друга проводящихся в различных локальных сегментах сети. В соответствии с поданными сигналами осуществляется *активация* нейронов, в результате чего активным оказывается один нейрон в сети (или в группе). Выходной нейрон, который выиграл соревнование, называется *нейроном-победителем*.

Нейроны в ходе конкурентного процесса, вследствие изменения значений синаптических весов, избирательно настраиваются на различные входные векторы или классы входных векторов. В процессе обучения наблюдается тенденция к росту значений весов, из-за которой создается своеобразная положительная обратная связь: более мощные возбуждающие импульсы – более высокие значения весов – большая активность нейронов.

При этом происходит естественное расслоение нейронов на различные группы, отдельные нейроны или их группы сотрудничают между собой и активизируются в ответ на возбуждение, создаваемое конкретными обучающими векторами, подавляя своей активностью другие нейроны. Можно говорить как о сотрудничестве между нейронами внутри группы, так и о конкуренции между нейронами внутри группы и между различными группами.

Среди механизмов самоорганизации можно выделить два основных класса: самоорганизация, основанная на ассоциативном правиле Хебба, и механизм конкуренции нейронов на базе обобщенного правила Кохонена. В дальнейшем будем рассматривать механизм конкуренции нейронов.

Нейроны помещаются в узлах решетки, обычно одно- или двумерной. Сети более высокой размерности также возможны, но используются достаточно редко. Как правило, это однослойные сети прямого распространения, в которых нейрон соединен со всеми компонентами N-мерного входного вектора x так, как это схематично изображено для $N = 2$ на рис. 3.1 [4].

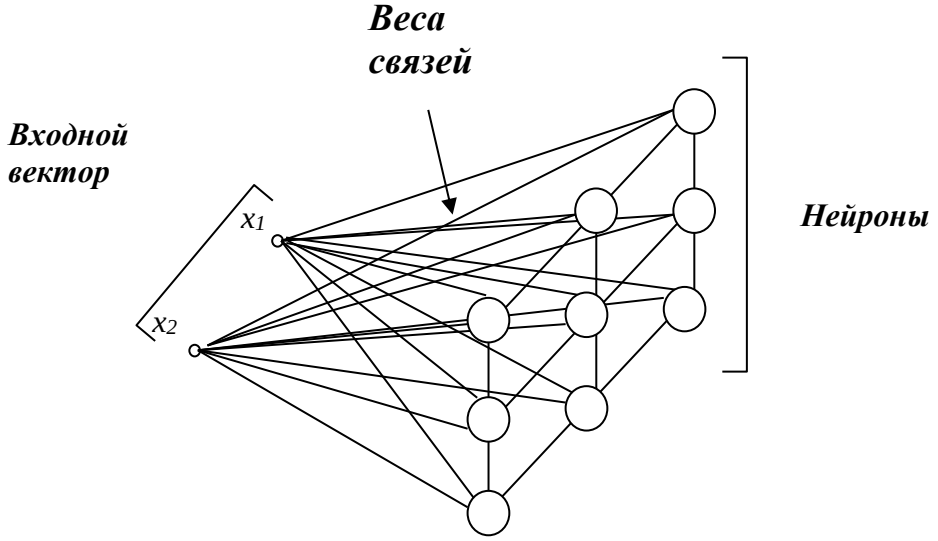

Рис. 3.1 Структура самоорганизующейся сети Кохонена.

Формирование самоорганизующихся сетей начинается с инициализации синаптических весов сети. Обычно синаптическим весам присваиваются малые значения, которые формируются генератором случайных чисел. При такой инициализации сеть изначально не имеет какого-либо порядка признаков входных векторов. После инициализации сети реализуются три основных процесса [4]:

1. *Конкуренция.* Для каждого входного вектора нейроны сети вычисляют относительные значения дискриминантной функции.
2. *Кооперация.* Победивший нейрон определяет топологическую окрестность группы нейронов, обеспечивая базис для кооперации между ними.
3. *Синаптическая адаптация*. Корректировка синаптических весов возбужденных нейронов позволяет увеличить их собственные значения дискриминантных функций по отношению к входным векторам. Корректировка производится таким образом, чтобы выходной сигнал нейрона-победителя усиливался при последующем применении аналогичных входных векторов.

Веса синаптических связей нейронов образуют вектор $w_i = [w_{i1}, w_{i2}, ..., w_{iN}]^T$. После нормализации входных векторов при активации сети вектором x в конкурентной борьбе побеждает тот нейрон, веса которого в наименьшей степени отличаются от соответствующих компонентов этого вектора. Для w-го нейрона-победителя выполняется соотношение

$$d(x, w_w) = \min_{1 \le i \le K} d(x, w_i) \qquad (3.1)$$

где $d(x, w)$ обозначает расстояние между векторами x и w, а K - количество нейронов. Вокруг нейрона-победителя образуется топологическая окрестность $S_w(t)$ с определенной энергетикой, уменьшающейся с течением времени. Нейрон-победитель и все нейроны, лежащие в пределах его окрестности, подвергаются адаптации, в ходе

которой их векторы весов изменяются в направлении вектора x по правилу Кохонена:

$$w_i(t+1) = w_i(t) + \eta_i(t)(x - w_i(t)) \qquad (3.2)$$

для $i \in S_w(t)$, где $\eta_i(t)$ - коэффициент обучения i-го нейрона на окрестности $S_w(t)$ в t-й момент времени. Значение $\eta_i(t)$ уменьшается с увеличением расстояния между i-м нейроном и победителем. Веса нейронов, находящихся вне окрестности $S_w(t)$, не изменяются. Размер окрестности и коэффициенты обучения нейронов являются функциями, значения которых уменьшаются с течением времени. В [10] доказано, что адаптация по формуле (3.2) эквивалентна градиентному методу обучения, основанному на минимизации целевой функции

$$E(w) = \frac{1}{2}\sum_{i,j,t} S_i(x(t))[x_j(t) - w_{ij}(t)]^2 \quad , \qquad (3.3)$$

а $S_i(x(t))$представляет собой функцию определения окрестности, изменяющуюся в процессе обучения.

После предъявления двух различных векторов x_1 и x_2 активизируются два нейрона сети, веса которых наиболее близки к координатам соответствующих векторов. Эти веса, обозначенные w_1 и w_2, могут отображаться в пространстве как две точки. Сближение векторов x_1 и x_2 вызывает соответствующее изменение в расположении векторов w_1 и w_2 В пределе равенство $w_1 = w_2$ выполняетсятогда и только тогда, когда x_1 и x_2 совпадают или практически неотличимы друг от друга. Сеть, в которой эти условия выполняются, называется топографической картой или картой Кохонена.

3.2 Меры расстояния между векторами и нормализация векторов

Процесс самоорганизации предполагает определение победителя каждого этапа, то есть нейрона, вектор весов которого в наименьшей степени отличается от поданного на вход сети вектора x. В этой ситуации важной проблемой становится выбор метрики, в которой будет измеряться расстояние между векторами x и w_i. Чаще всего в качестве меры расстояния используются:

- эвклидова мера

$$d(x, w_i) = \|x - w_i\| = \sqrt{\sum_{j=1}^{N}(x_j - w_{ij})^2}\,; \qquad (3.4)$$

- скалярное произведение

$$d(x, w_i) = 1 - x \cdot w = 1 - \|x\| \cdot \|w_i\| \cdot \cos(x, w_i)\,; \qquad (3.5)$$

- мера относительно нормы L1 (Манхэттен)

$$d(x, w_i) = \sqrt{\sum_{j=1}^{N}\left|x_j - w_{ij}\right|}\,; \qquad (3.6)$$

- мера относительно нормы L_∞

$$d(x, w_i) = \max_j \left| x_j - w_{ij} \right|. \quad (3.7)$$

При использовании эвклидовой меры разбиение пространства на зоны доминирования нейронов равносильно разбиению на области Вороного. При использовании другой меры формируется другое разделение областей влияния нейронов. Использование скалярного произведения требует нормализации входных векторов, так как в противном случае возможно неравномерное распределение нейронов: в одной области может находиться несколько нейронов, а в другой – ни одного нейрона.

При нормализованных входных обучающих векторах стремящиеся к ним векторы весов нормализуются автоматически. Однако нормализация векторов приводит к тому, что если $\|w_i\| = const$, то для всех нейронов при фиксированном значении x произведение $\|x\|\|w_i\|$ также становится постоянной величиной. Поэтому активация нейрона определяется значением $\cos(x, w_i)$, которое становится общей мерой для всей сети. Следует отметить, что при нормализации вектора весов эвклидова мера и скалярное произведение равнозначны друг другу, так как $\|x - w_i\|^2 = \|x\|^2 + \|w_i\|^2 - 2x^T w_i$. Поэтому $\min\|x - w\|^2 = \max\left(x^T w_i\right) \text{при} \|w_i\| = const$. Экспериментальные исследования подтвердили необходимость применения нормализации векторов при малой размерности пространства [2]. Такая нормализация проводится двумя способами:

1. Переопределение компонентов вектора в соответствии с формулой

$$x_i \leftarrow \frac{x_i}{\sqrt{\sum_{i=1}^{N} x_i^2}}. \quad (3.8)$$

2. Увеличением размерности пространства на одну координату с таким выбором $(N+1)$-го компонента вектора, чтобы выполнялось условие:

$$\sum_{i=1}^{N+1} x_i^2 = 1. \quad (3.9)$$

При использовании второго способа возникает необходимость предварительного масштабирования компонентов вектора x.

С увеличением размерности входного вектора эффект нормализации становится менее заметным, а при размерности $N > 200$ вообще сходит на нет.

3.3 Проблема мертвых нейронов

При инициализации весов сети случайным способом часть нейронов может оказаться в области пространства, в которой отсутствуют данные или их количество ничтожно мало. Эти нейроны имеют мало шансов на победу и адаптацию своих весов, поэтому они остаются мертвыми. Таким образом,

входные данные будут интерпретироваться меньшим количеством нейронов, а погрешность интерпретации данных увеличится. Поэтому важной проблемой становится активация всех нейронов сети, которую можно осуществить, если в алгоритме обучения предусмотреть учет количества побед каждого нейрона, а процесс обучения организовать так, чтобы дать шанс победить и менее активным нейронам.

Существуют различные механизмы учета активности нейронов в процессе обучения [4]. Часто используется метод подсчета потенциала p_i каждого нейрона, значение которого модифицируется всякий раз после предъявления очередной реализации входного вектора x в соответствии со следующей формулой (в ней предполагается, что победителем стал w-й нейрон):

$$p_i(t+1) = \begin{cases} p_i(t) + \dfrac{1}{K}; (i \neq w) \\ p_i(t) - p_{\min}; (i = w) \end{cases}. \qquad (3.10)$$

Значение коэффициента $p_{\min}$ определяет минимальный потенциал, разрешающий участие в конкурентной борьбе. Если фактическое значение потенциала p_i падает ниже $p_{\min}$, то *i-й* нейрон «отдыхает», а победитель ищется среди нейронов, для которых выполняется соотношение

$$d(x, w_w) = \min\{d(x, w_i)\}, для 1 \le i \le K \; и \; p_i \ge p_{\min} \qquad (3.11)$$

Максимальное значение потенциала ограничивается на уровне, равном 1. Выбор конкретного значения $p_{\min}$ позволяет установить порог готовности нейрона к конкурентной борьбе. При $p_{\min}$ =0 утомляемость нейронов не возникает, и каждый из них сразу после победы будет готов к продолжению соперничества. При $p_{\min}$ =1 возникает другая крайность, вследствие которой нейроны побеждают по очереди, так как в каждый момент только один из них оказывается готовым к соперничеству. На практике хорошие результаты достигаются при $p_{\min} \approx 0{,}74$.

В другом очень удачном алгоритме обучения количество побед нейрона учитывается при подсчете эффективного расстояния между вектором весов и реализацией обучающего вектора x. Это расстояние модифицируется пропорционально количеству побед данного нейрона в прошлом. Если обозначить количество побед i-го нейрона N_i, такую модификацию можно представить в виде

$$d(x, w_i) \leftarrow N_i d(x, w_i). \qquad (3.12)$$

Активные нейроны с большим значением N_i штрафуются искусственным завышением этого расстояния. Отметим, что модификация расстояния производится только при выявлении победителя. В момент уточнения весов учитывается фактическое расстояние. Обычно после двух или трех циклов обучения модификация прекращается, что позволяет продолжить «честную» конкуренцию нейронов.

3.4 Алгоритмы обучения без учителя

3.4.1 Алгоритм WTA

Алгоритмы обучения, используемые для обучения нейронных сетей Кохонена, называются алгоритмами обучения без учителя. Подобные алгоритмы применяются в тех случаях, когда нет эталонных выходных значений для входных векторов.

Целью обучения сети с самоорганизацией на основе конкуренции, считается такое упорядочение нейронов, которое минимизирует значение отклонения вектора весов от входного вектора x. При p входных векторах x эта погрешность в эвклидовой метрике может быть выражена в виде:

$$E_q = \frac{1}{2}\sum_{t=1}^{p}\left\| x_i - w_{w(t)} \right\|^2, \qquad (3.13)$$

где $w_{w(t)}$ - это вес нейрона-победителя при предъявлении вектора x_i.

Этот подход также называется *векторным квантованием* (*VQ*). Номера нейронов-победителей при последовательном предъявлении векторов *x*образуют так называемую кодовую таблицу. При классическом решении задачи кодирования применяется алгоритм *K-усреднений*, носящий имя обобщенного алгоритма Ллойда.

Для нейронных сетей аналогом алгоритма Ллойда считается алгоритм *WTA(WinnerTakesAll* – победитель получает все). В соответствии с ним после предъявления вектора *x* рассчитывается активность каждого нейрона. Победителем признается нейрон с самым сильным выходным сигналом, то есть тот, для которого скалярное произведение $\left(x^T w\right)$ оказывается наибольшим, что для нормализованных векторов равнозначно наименьшему эвклидову расстоянию между входным вектором и вектором весов нейронов. Победитель получает право уточнить свои веса в направлении вектора *x* согласно правилу

$$w_w(t+1) = w_w(t) + \eta(x - w_w(t)) \qquad (3.14)$$

Веса остальных нейронов уточнению не подлежат. Алгоритм позволяет учитывать усталость нейронов путем подсчета количества побед каждого из них и поощрять элементы с наименьшей активностью для выравнивания их шансов.

Помимо алгоритмов *WTA,* в которых в каждой итерации может обучаться только один нейрон, для обучения сетей с самоорганизацией широко применяется алгоритмы типа *WTM* (*WinnerTakesMost* – победитель получает больше), в которых, кроме победителя, уточняют значения своих весов и нейроны из его ближайшего окружения. При этом, чем дальше какой-либо нейрон находится от победителя, тем меньше изменяются его веса. Процесс уточнения вектора весов может быть определен в виде

$$w_i(t+1) = w_i(t) + \eta_i S(i,x)(x - w_i(t)) \qquad (3.15)$$

для всех *i* нейронов, расположенных в окрестности победителя. В приведенной формуле коэффициент обучения η_i каждого нейрона отделен от

его расстояния до предъявленного вектора x функцией $S(i,x)$. Если $S(i,x)$ определяется в форме

$$S(i,x)=\begin{cases}1, для\ i=w\\0, для\ i\neq w\end{cases}, \tag{3.16}$$

для wобозначает номер победителя, то мы получаем классический алгоритм *WTA*. Существует множество вариантов алгоритма *WTM*, отличающихся ,прежде всего формой функции $S(i,x)$. Для дальнейшего обсуждения выберем классический алгоритм Кохонена.

3.4.2 Алгоритм Кохонена

Алгоритм Кохонена относится к наиболее старым алгоритмам обучения сетей с самоорганизацией на основе конкуренции, и в настоящее время существуют различные его версии [4]. В классическом алгоритме Кохонена сеть инициализируется путем приписывания нейронам определенных позиций в пространстве и связывании их с соседями на постоянной основе. В момент выбора победителя уточняются не только его веса, но также веса и его соседей, находящихся в ближайшей окрестности. Таким образом, нейрон-победитель подвергается адаптации вместе со своими соседями.

$$S(i,x)=\begin{cases}1, для\ -K<d(i,w)<K\\0, иначе\end{cases}, \tag{3.17}$$

В этом выражении $d(i,w)$ может обозначать как эвклидово расстояние между векторами весов нейрона-победителя w и i-го нейрона, так и расстояние, измеряемое количеством нейронов.

Другой тип соседства в картах Кохонена- это соседство гауссовского типа, при котором функция $S(i,x)$ определяется формулой

$$S(i,x)=\exp\left(-\frac{d^2(i,w)}{2\lambda^2}\right). \tag{3.18}$$

Уточнение весов нейронов происходит по правилу:

$$w_i(t)=w_i(t-1)+\eta(t)\cdot S(i,w)\cdot(x-w_i). \tag{3.19}$$

Степень адаптации нейронов-соседей определяется не только эвклидовым расстоянием между i-м нейроном и нейроном-победителем (w-м нейроном) $d(i,w)$, но также уровнем соседства λ. Как правило, гауссовское соседство дает лучшие результаты обучения и обеспечивает лучшую организацию сети, чем прямоугольное соседство.

4 Сети на основе радиально-базисных функций

4.1 Математическое обоснование радиально-базисных сетей

Многослойные нейронные сети, с точки зрения математики, выполняют аппроксимацию стохастической функции нескольких переменных путем преобразования множества входных переменных $x \in R^N$ во множество выходных переменных $y \in R^M$. Вследствие характера сигмоидальной функции активации осуществляется аппроксимация глобального типа, так как преобразование значения функции в произвольной точке пространства выполняется объединенными усилиями многих нейронов.

Другой способ отображения входного множества в выходное множество заключается в преобразовании путем адаптации нескольких одиночных аппроксимирующих функций к ожидаемым значениям, причем эта адаптация проводится только в локальной области многомерного пространства. При таком подходе отображение всего множества данных представляет собой сумму локальных преобразований, а скрытые нейроны составляют множество базисных функций локального типа.

Особое семейство образуют радиальные сети, в которых скрытые нейроны реализуют функции, радиально изменяющиеся вокруг выбранного центра и принимающие ненулевые значения только в окрестности этого центра. Подобные функции, определяемые в виде $\varphi(x) = \varphi(\|x - c\|)$, называются *радиальными базисными функциями*. В таких сетях роль скрытого нейрона заключается в отображении радиального пространства вокруг одиночной заданной точки либо вокруг группы таких точек, образующих кластер. Суперпозиция сигналов, поступающих от всех скрытых нейронов, которая выполняется выходным нейроном, позволяет получить отображение всего многомерного пространства.

Сети радиального типа представляют собой естественное дополнение сигмоидальных сетей. Сигмоидальный нейрон представляется в многомерном пространстве гиперплоскостью, которая разделяет это пространство на два класса, в которых выполняется одно из двух условий: либо $\sum_j w_{ij} x_j > 0$, либо $\sum_j w_{ij} x_j < 0$. Такой подход продемонстрирован на рис. 4.1а. В свою очередь радиальный нейрон представляет собой гиперсферу, которая осуществляет шаровое разделение пространства вокруг центральной точки (рис. 4.1б) [4].

Именно с этой точки зрения радиальный нейрон является естественным дополнением сигмоидального нейрона, поскольку в случае круговой симметрии данных позволяет заметно уменьшить количество нейронов, необходимых для разделения различных классов.

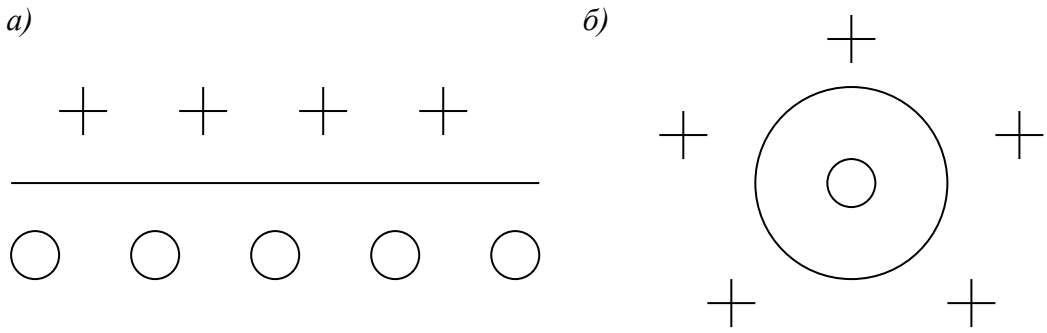

Рис. 4.1 Иллюстрация способов разделения пространства данных:
а) сигмоидальным нейроном; б) радиальным нейроном

Так как нейроны могут выполнять различные базисные функции, в радиальных сетях отсутствует необходимость использования большого количества скрытых слоев. Структура типичной радиальной сети включает входной слой, на который подаются сигналы, описываемые входным вектором *x*, скрытый слой с нейронами радиального типа и выходной слой, состоящий, как правило, из одного или нескольких линейных нейронов. Функция выходного нейрона сводится исключительно к взвешенному суммированию сигналов, генерируемых скрытыми нейронами.

Математическую основу функционирования радиальных сетей составляет теорема Т. Ковера о разделимости образов, которая базируется на двух моментах [4]:

1. Определение нелинейной скрытой функции $\varphi_i(x)$, где *x*– входной вектор, а *i=1,2,…,K, K* – размерность скрытого пространства.
2. Высокая размерность скрытого пространства по сравнению с размерностью входного. Эта размерность определяется количеством скрытых нейронов.

Если вектор радиальных функций $\varphi(x) = [\varphi_1(x), \varphi_2(x), ..., \varphi_K(x)]^T$ в N-мерном входном пространстве обозначить *φ(x),* то это пространство является нелинейно φ -разделяемым на два пространственных класса X^+и X^- тогда, когда существует такой вектор весов *w*, что

$$\begin{aligned} w^T\varphi(x) > 0 \quad x \in X^+, \\ w^T\varphi(x) < 0 \quad x \in X^-. \end{aligned} \tag{4.1}$$

Граница между этими классами определяется уравнением $w^T\varphi(x) = 0$.

Ковер доказал, что каждое множество образов, случайным образом размещенных в многомерном пространстве, является φ-разделяемым с вероятностью 1 при условии большой размерности *K* этого пространства. На практике это означает, что применение достаточно большого количества скрытых нейронов, реализующих радиальные функции $\varphi_i(x)$, гарантирует решение задачи классификации при построении всего лишь двухслойной сети. При этом скрытый слой должен реализовать вектор *φ(x)*, а выходной

слой может состоять из единственного линейного нейрона, выполняющего суммирование выходных сигналов от скрытых нейронов с весовыми коэффициентами, заданными вектором *w*.

Простейшая нейронная сеть радиального типа функционирует по принципу многомерной интерполяции, состоящей в отображении *p* различных входных векторов x_t *(t=1,2,…,p)* из входного *N*-мерного пространства во множество из *p* рациональных чисел d_t *(t=1,2,…,p).* Для реализации этого процесса необходимо использовать *p* скрытых нейронов радиального типа и задать такую функцию отображения *F(x)*, для которой выполняется условие интерполяции:

$$F(x_t) = d_t . \tag{4.2}$$

С практической же точки зрения использование в разложении большого числа *p* базисных функций недопустимо, поскольку если число обучающих выборок велико и равно числу радиальных функций, то в результате вычислительная сложность обучающего алгоритма становится чрезмерной, а сама сеть адаптируется к разного рода шумам и нерегулярностям, сопровождающим обучающие выборки. Поэтому необходимо редуцировать количество весов, что приводит к уменьшению количества базисных функций. В этом случае ищется субоптимальное решение в пространстве меньшей размерности, которое с достаточной точностью аппроксимирует точное решение. Если ограничиться *K* базисными функциями, то аппроксимирующее решение можно представить в виде

$$F(x) = \sum_{i=1}^{K} w_i \phi\left(\left\| x - c_i \right\|\right), \tag{4.3}$$

где *K<p*, а c_i *(i=1,2,…,K)* – множество центров, которые необходимо определить. В особом случае, если принять *K=p*, то можно получить точное решение $c_i = x_t$.

Задача аппроксимации состоит в подборе соответствующего количества радиальных функций $\varphi(\| x - c_i \|)$ и их параметров, а также в таком подборе весов w_i *(i=1,2,…,K)*, чтобы решение уравнения (4.3) было наиболее близким к точному. Поэтому проблему подбора параметров радиальных функций и значений весов w_i сети можно свести к минимизации целевой функции, которая при использовании метрики Эвклида записывается в форме

$$E = \sum_{t=1}^{p} \left[\sum_{i=1}^{K} w_i \varphi(\| x_t - c_i \|) - d_t \right]^2 . \tag{4.4}$$

В этом уравнении *K* представляет количество радиальных нейронов, а *p*– количество обучающих пар (x_t, d_t), где x_t – это входной вектор, а d_t – соответствующий ему ожидаемый выходной вектор.

4.2 Структура радиально-базисной сети

На рис. 4.2 представлена обобщенная структура радиально-базисной сети. В качестве радиальной функции чаще всего применяется функция

Гаусса. При размещении ее центра в точке c_i она может быть определена в сокращенной форме как:

$$\varphi(x) = \varphi(\| x - c_i \|) = \exp\left(- \frac{\| x - c_i \|^2}{2\sigma_i^2} \right). \qquad (4.5)$$

В этом выражении σ_i – параметр, от значения которого зависит ширина функции.

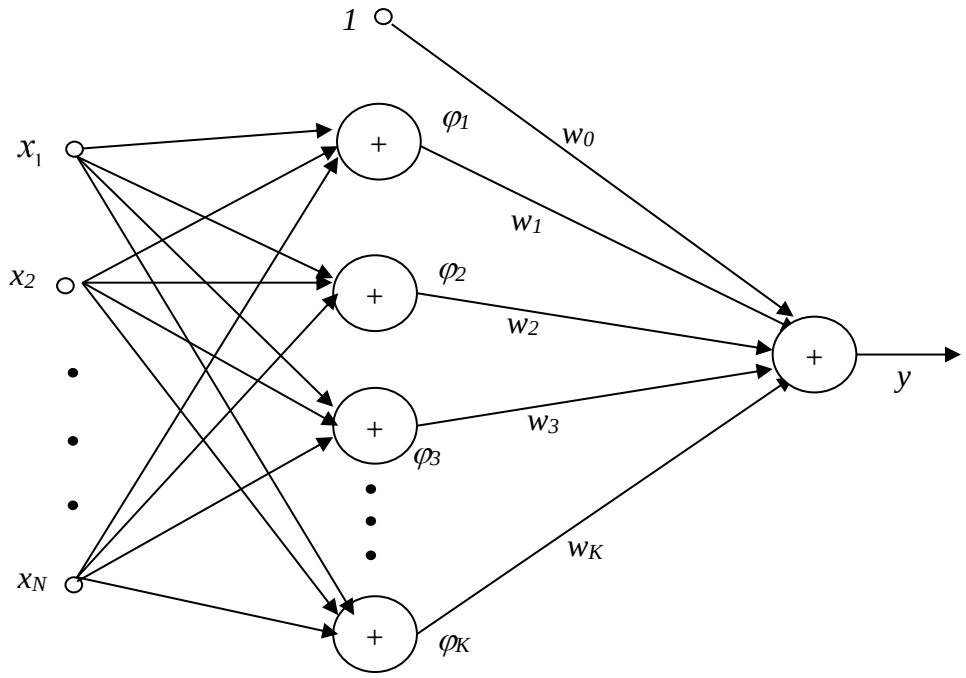

Рис. 4.2 Обобщенная структура радиальной сети RBF

Полученное решение, представляющее аппроксимирующую функцию в многомерном пространстве в виде взвешенной суммы локальных базисных радиальных функций (выражение (4.3)), может быть интерпретировано радиальной нейронной сетью, представленной на рис. 4.2, в которой φ_i определяется зависимостью (4.5).

Это сеть с двухслойной структурой, в которой только скрытый слой выполняет нелинейное отображение, реализуемое нейронами с базисными радиальными функциями, параметры которых (центры c_i и коэффициенты σ_i) уточняются в процессе обучения. Скрытый слой не содержит линейных весов, аналогичных весам сигмоидальной сети. Выходной нейрон, как правило, линеен, а его роль сводится к взвешенному суммированию сигналов, поступающих от нейронов скрытого слоя. Вес w_0, как и при использовании сигмоидальных функций, представляет пороговый элемент, определяющий показатель постоянного смещения функции.

Полученная архитектура радиальных сетей аналогична структуре многослойной сети с одним скрытым слоем. Роль скрытых нейронов в ней играют базисные радиальные функции. Однако, в отличие от сигмоидальной сети, радиальная сеть имеет фиксированную структуру с одним скрытым слоем и линейными выходными нейронами. Используемые радиальные

функции скрытых нейронов могут иметь разнообразную структуру. Нелинейная радиальная функция каждого скрытого нейрона имеет свои значения параметров c_i, тогда как в сигмоидальной сети применяются, как правило, стандартные функции активации с одним и тем же параметром. Аргументом радиальной функции является эвклидово расстояние вектора x от центра c_i и σ_i, а в сигмоидальной сети это скалярное произведение векторов $w^T x$.На рисунке 4.3 приведена детальная структура *RBF*-сети с радиальной функцией вида (4.5).

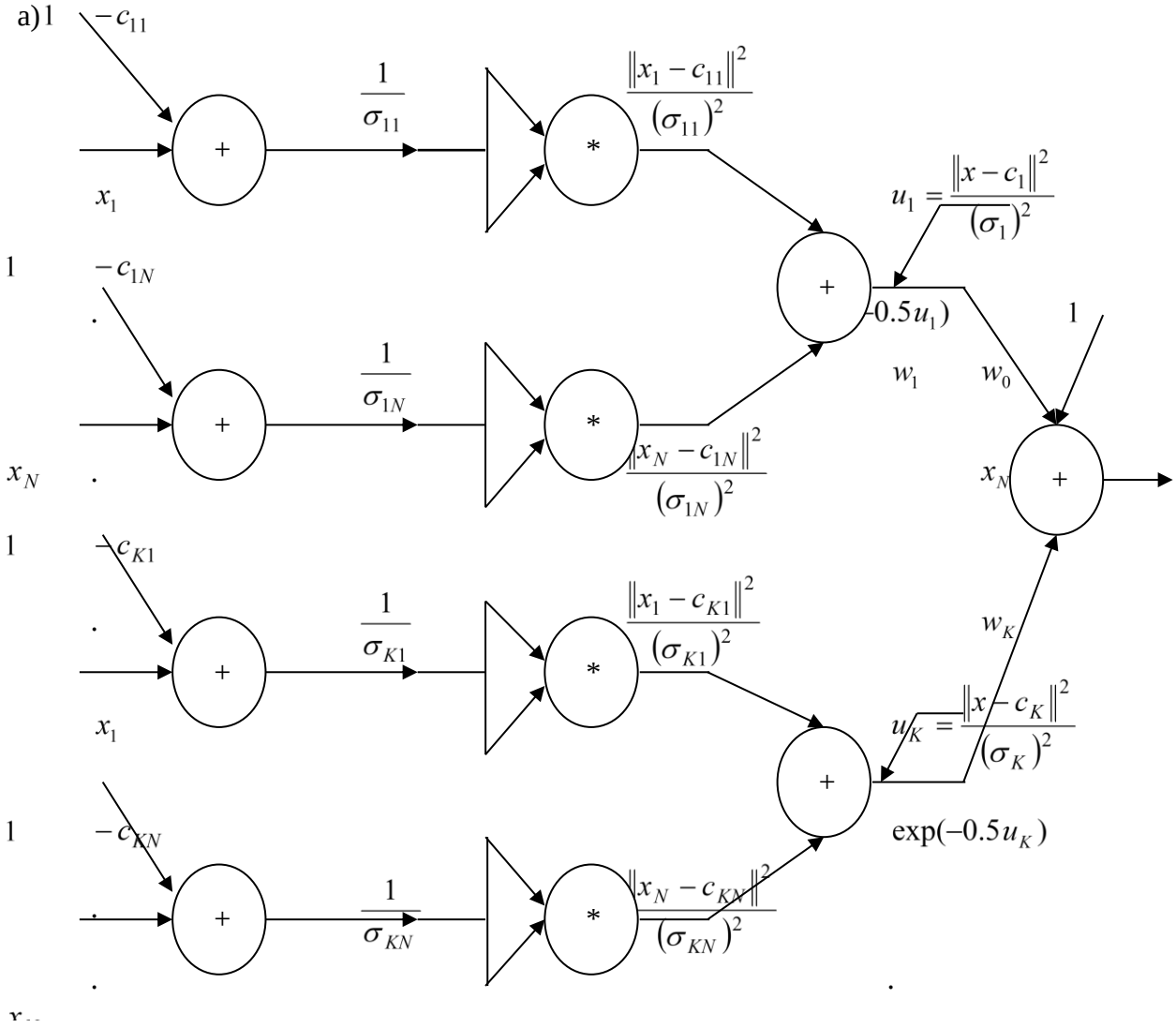

Рис. 4.3 Детальная структура *RBF*-сети.

4.3 Основные алгоритмы обучения радиальных сетей

4.3.1. Алгоритм самоорганизации для уточнения параметров радиальных функций

Процесс обучения сети *RBF* с учетом выбранного типа радиальной базисной функции сводится:

- к подбору центров c_i и параметров σ_i формы базисных функций (обычно используются алгоритмы обучения без учителя);
- к подбору весов нейронов выходного слоя (обычно используются алгоритмы обучения с учителем).

Подбор количества базисных функций, каждой из которых соответствует один скрытый нейрон, считается основной проблемой, возникающей при корректном решении задачи аппроксимации. Как и при использовании сигмоидальных сетей, слишком малое количество нейронов не позволяет уменьшить в достаточной степени погрешность обобщения множества обучающих данных, тогда как слишком большое их число увеличивает погрешность выводимого решения на множестве тестирующих данных. Подбор необходимого и достаточного количества нейронов зависит от многих факторов. Как правило, количество базисных функций *K* составляет определенную долю от объема обучающих данных *p*, причем фактическая величина этой доли зависит от размерности вектора $\boldsymbol{x}$ и от разброса ожидаемых значений d_t, соответствующих входным векторам x_t, для $t=1,2,...,p$.

Процесс самоорганизации обучающих данных автоматически разделяет пространство на так называемые области Вороного, определяющие различающиеся группы данных. Данные, сгруппированные внутри кластера, представляются центральной точкой, определяющей среднее значение всех его элементов. Центр кластера отождествляется с центром соответствующей радиальной функции.

Разделение данных на кластеры можно выполнить с использованием алгоритма *K-усреднений*.

Согласно этому алгоритму центры радиальных базисных функций размещаются только в тех областях входного пространства, в которых имеются информативные данные. Если обучающие данные представляют непрерывную функцию, начальные значения центров в первую очередь размещают в точках, соответствующих всем максимальным и минимальным значениям функции.

Пусть N - число нейронов скрытого слоя, *t*– номер итерации алгоритма. Тогда алгоритм *K-усреднений* можно описать следующим образом [4]:

1. *Инициализация*. Случайным образом выбираем начальные значения центров $c_i(0)$, которые должны быть различны. При этом значения эвклидовой нормы по возможности должны быть небольшими.
2. *Выборка*. Выбираем вектор x_t из входного пространства.
3. *Определение центра-победителя.* Выбираем центр c_w, ближайший к x_t, для которого выполняется соотношение: $w = \arg\min_i \| x_t - c_i(t) \|, i = 1,2,...,N.$
4. *Уточнение.* Центр-победитель подвергается уточнению в соответствии с формулой (4.6):

$$c_i(t+1) = c_i(t) + \eta(x_t - c_i(t)), \quad (4.6)$$

где η - коэффициент обучения, имеющий малое значение (обычно η<<1), причем уменьшающееся во времени. Остальные центры не изменяются.

5. *Продолжение.* Увеличиваем на единицу значение t и возвращаемся к шагу 2, пока положение центров не стабилизируется.

Также применяется разновидность алгоритма, в соответствии с которой значение центра-победителя уточняется в соответствии с формулой (4.6), а один или несколько ближайших к нему центров отодвигаются в противоположном направлении, и этот процесс реализуется согласно выражению

$$c_i(t+1) = c_i(t) - \eta_1(x_t - c_i(t)). \quad (4.7)$$

Такая модификация алгоритма позволяет отдалить центры, расположенные близко друг к другу, что обеспечивает лучшее обследование всего пространства данных *($\eta_1 < \eta$).*

После фиксации местоположения центров проводится подбор значений параметров σ_i, соответствующих конкретным базисным функциям. Параметр σ_i радиальной функции влияет на форму функции и величину области ее охвата, в которой значение этой функции не равно нулю. Подбор σ_i должен проводится таким образом, чтобы области охвата всех радиальных функций накрывали все пространство входных данных, причем любые две зоны могут перекрываться только в незначительной степени. При такой организации подбора значения σ_i, реализуемое радиальной сетью отображение функции будет относительно монотонным.

Для расчета σ_i может быть применен алгоритм, при котором на значение σ_i влияет на расстояние между *i*-м центром c_i и его *R* ближайшими соседями. В этом случае значение σ_i определяется по формуле (4.8):

$$\sigma_i = \sqrt{\frac{1}{R}\sum_{k=1}^{R} \| c_i - c_k \|^2} \; . \quad (4.8)$$

На практике значение *R* обычно лежит в интервале [3; 5].

Данный алгоритм обеспечивает только локальную оптимизацию, зависящую от начальных условий и параметров процесса обучения.

При неудачно выбранных начальных условиях, некоторые центры могут застрять в области, где количество обучающих данных ничтожно мало, либо они вообще отсутствуют. Следовательно, процесс модификации центров затормозится или остановится.

Для решения данной проблемы могут быть применены два различных подхода:

1. Задать фиксированные значения η для каждого центра. Центр, наиболее близкий к текущему вектору *x,* модифицируется сильнее,

остальные - обратно пропорционально их расстоянию до этого текущего вектора *x*.

2. Использовать взвешенную меру расстояния от каждого центра до вектора *x*. Весовая норма делает «фаворитами» те центры, которые реже всего побеждают.

Оба подхода не гарантируют 100% оптимальность решения.

Подбор коэффициента ηтоже является проблемой. Если ηимеет постоянное значение, то оно должно быть мало, чтобы обеспечить сходимость алгоритма, следовательно, увеличивается время обучения.

Адаптивные методы позволяют уменьшать значение η по мере роста времени *t*. Наиболее известным адаптивным методом является алгоритм Даркена-Муди:

$$\eta(t) = \frac{\eta_0}{1 + \frac{t}{T}}, \tag{4.9}$$

где T – постоянная времени, подбираемая для каждой задачи. При $t<T$ ηне изменяется, при $t>T$ – уменьшается до нуля.

4.3.2. Применение метода обратного распространения ошибки для радиально-базисных сетей

Обособленный класс алгоритмов обучения радиальных сетей составляют градиентные алгоритмы обучения с учителем, в которых используется алгоритм обратного распространения ошибки. Их основу составляет целевая функция, которая для одного обучающегопримера имеет вид:

$$E = \frac{1}{2}\left[\sum_{i=1}^{K} w_i \varphi_i(x) - d\right]^2 \tag{4.10}$$

Предположим, что применяется гауссовская радиальная функция вида:

$$\varphi_i(x(t)) = \exp\left(-\frac{1}{2}u_i(t)\right) \tag{4.11}$$

$$u_i(t) = \sum_{j=1}^{N} \frac{(x_j(t) - c_{ij}(t))^2}{\sigma_{ij}^2(t)} \tag{4.12},$$

где *i*– индекс нейрона скрытого слоя, *j*–индекс компонента входного вектора, *t* – индекс обучающего примера в выборке.

Обучение сети с использованием алгоритма обратного распространения ошибки проводится в два этапа. На первом этапе предъявляется обучающий пример и рассчитываются значения сигналов выходных нейронов сети и значение целевой функции, заданной выражением (4.10). На втором этапе минимизируется значение этой функции.

Подбор значений параметров можно осуществлять, используя градиентные методы оптимизации независимо от объекта обучения – будь то вес или центр. Независимо от выбираемого метода градиентной

оптимизации, необходимо, прежде всего, получить вектор градиента целевой функции относительно всех параметров сети. В результате дифференцирования этой функции получим:

$$\frac{\partial E(t)}{\partial w_0(t)} = y(t) - d(t) \tag{4.13}$$

$$\frac{\partial E(t)}{\partial w_i(t)} = \exp\left(-\frac{1}{2}u_i(t)\right)(y(t) - d(t)) \tag{4.14}$$

$$\frac{\partial E(t)}{\partial c_{ij}(t)} = (y(t) - d(t))w_i(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^2(t)} \tag{4.15}$$

$$\frac{\partial E(t)}{\partial \sigma_{ij}(t)} = (y(t) - d(t))w_i(t)\exp\left(-\frac{1}{2}u_i(t)\right)\frac{(x_j(t) - c_{ij}(t))}{\sigma_{ij}^3(t)} \tag{4.16}$$

При использовании метода наискорейшего спуска формулы для корректировки параметров радиально-базисной сети примут следующий вид:

$$w_i(t+1) = w_i(t) - \eta\frac{\partial E(t)}{\partial w_i(t)}, \tag{4.17}$$

$$c_{ij}(t+1) = c_{ij}(t) - \eta\frac{\partial E(t)}{\partial c_{ij}(t)}, \tag{4.18}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta\frac{\partial E(t)}{\partial \sigma_{ij}(t)}. \tag{4.19}$$

Список используемых источников

1. Адаменко А.Н., Кучуков А. Логическое программирование и Visual Prolog – Спб.: БХВ – Петербург, 2003.
2. Андрейчиков А.В., Андрейчикова О.Н.. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2006. – 424 с., ил.
3. Хайкин С. Нейронные сети: Полный курс: Пер. с англ. - 2-е изд. – М.: Вильямс, 2006. – 1104 с.: ил.
4. Оссовский С. Нейронные сети для обработки информации / Пер. с пол. И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.: ил.
5. Уоссерман Ф. Нейрокомпьютерная техника: теория и практика / Пер. с англ. Ю.А. Зуев. – М.: Мир, 1992.