ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

НЕЙРОСЕТЕВЫЕ СТРУКТУРЫ И ТЕХНОЛОГИИ

Часть 1
Электрические и математические модели нейронов. НС прямого распространения

Учебное пособие для вузов

Составители:
В.И. Клюкин
Ю.К. Николаенков

Издательско-полиграфический центр
Воронежского государственного университета
2008

Утверждено научно-методическими советами физического факультета 14 февраля 2008 г., протокол № 6 и факультета компьютерных наук 4 декабря 2008 г., протокол № 4.

Рецензент доктор технических наук, профессор кафедры информационных систем факультета компьютерных наук ВГУ А.А. Сирота.

Предлагаемое пособие содержит материал по основным принципам построения, функционирования и применения искусственных нейронных сетей. В первой части рассмотрены биологические основы функционирования нервных клеток, электрические и математические модели нейронов, а также основные структуры и методы обучения многослойных НС прямого распространения. Пособие подготовлено на кафедре физики полупроводников и микроэлектроники физического факультета Воронежского государственного университета.

Рекомендуется для самостоятельной работы студентов 3, 4 курсов факультета компьютерных наук и 4, 5 курсов физического факультета.

Для специальностей: 010300 – Математика. Компьютерные науки
230201 (071900) – Информационные системы и технологии
010803 (010400) – Микроэлектроника и полупроводниковые приборы
202100 – Нанотехнологии в электронике

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Основные тенденции развития кибернетики начала XIX века – это биологизация и гибридизация. Под первым направлением чаще всего понимается создание моделей и устройств, имитирующих механизмы, реализованные в процессе эволюции в живых существах, второе состоит в совместном применении различных методов для обработки информации об одном и том же объекте, поскольку только многоаспектное изучение проблемы позволяет получить ее оптимальное решение. Обе названные тенденции весьма удачно иллюстрирует наиболее динамично развивающаяся область современной теории интеллектуальных вычислений, связанная с построением и применением искусственных нейронных сетей (далее – ИНС, НС), которые все более серьезно рассматриваются в качестве методологической основы для создания сверхмощных вычислительных систем с параллельной обработкой информации.

Широкую популярность ИНС приобрели благодаря способности сравнительно легко адаптироваться к требованиям различных практических приложений. Они реализуют одну из парадигм искусственного интеллекта – коннекционистскую, когда возможности сети полностью определяются ее топологией, а вместо характерного для традиционных ЭВМ программирования используется обучение НС, сводящееся к настройке весовых коэффициентов межнейронных связей с целью оптимизации заданного критерия качества функционирования сети. Присущие ИНС нелинейность, адаптивность, потенциальная отказоустойчивость делают их универсальным средством обработки информации, особенно эффективным при решении трудноформализуемых задач распознавания образов, построения ассоциативной памяти, динамического управления и т. п.

Характерная особенность ИНС состоит также в возможности их реализации с применением технологий СБИС и наноэлектроники. Все это вызывает в последние годы огромный рост интереса к нейронным сетям и существенный прогресс в их исследовании. Практически создана база для выработки новых приемов восприятия, распознавания и обобщения визуальной информации, управления сложными системами, обработки речевых и биологических сигналов, решения задач аппроксимации, классификации и прогнозирования.

В связи с вышесказанным в предлагаемом пособии предпринята попытка в сжатой форме изложить основные концепции теории нейронных сетей, возможные методы их программной и аппаратной реализации, а также использования в практических задачах и приложениях. Считаем, что представленный материал окажется полезным для студентов и научных работников, специализирующихся в областях компьютерных наук, микро- и наноэлектроники.

1. ЭЛЕКТРИЧЕСКИЕ МОДЕЛИ НЕЙРОНОВ

Искусственные нейронные сети представляют собой набор математических и алгоритмических методов для решения широкого круга задач. Существуют два подхода к созданию ИНС – информационный и биологический. В первом подходе безразлично, какие механизмы лежат в основе функционирования ИНС, достаточно, чтобы процессы обработки информации были аналогичны биологическим, во втором – важно полное биоподобие, но в любом варианте необходимо детальное изучение работы биологических нервных клеток и сетей с точки зрения химии, физики, теории информации и синергетики. При этом желательно знать ответы на следующие основные вопросы:

– как работает биологический нейрон, какие его свойства важны при моделировании;

– каким образом нейроны объединяются в сеть, как передается информация между ними;

– каковы механизмы обучения бионейронных сетей, как оценивается «правильность» выходных сигналов.

1.1. Биологические основы функционирования нервных клеток

ИНС представляют собой попытку использования процессов, происходящих в нервных системах живых существ для создания новых информационных технологий. Основным элементом нервной системы является нервная клетка, сокращенно называемая *нейроном*. Как и у любой другой клетки, у нейрона имеется тело, называемое *сомой*, внутри которого располагается ядро, а наружу выходят многочисленные отростки – тонкие, густо ветвящиеся *дендриты*, и более толстый, расщепляющийся на конце *аксон* (рис. 1.1). Входные сигналы поступают в клетку через *синапсы*, выходной сигнал передается аксоном через его нервные окончания (*коллатералы*) к синапсам других нейронов, которые могут находиться как на дендритах, так и непосредственно на теле клетки.

Передача сигналов внутри нервной системы – очень сложный электрохимический процесс, основанный на выделении особых химических веществ – нейромедиаторов, которые образуются под действием поступающих от синапсов раздражителей и воздействуют на клеточную мембрану, вызывая изменение ее энергетического потенциала пропорционально количеству попавшего нейромедиатора. Поскольку синапсы отличаются размерами и концентрацией выделяемого нейромедиатора, то импульсы одинаковой величины, поступающие по различным синапсам, могут возбуждать нервную клетку в разной степени. Мерой возбуждения клетки считается уровень поляризации ее мембраны, зависящий от суммарного количества нейромедиатора по всем синапсам, причем синапсы могут оказывать как возбуждающее, так и тормозящее действие. Если баланс возбуждений и торможений невелик или отрицателен, то есть ниже порога срабатывания клетки, то выходной сигнал не образуется, в про-

тивном случае значение выходного сигнала лавинообразно нарастает, принимая характерный вид нервного импульса (рис. 1.2), передаваемого аксоном на подключенные к нему нейроны. Амплитуда этого импульса не зависит от степени превышения порога, то есть клетка действует по принципу «всё или ничего». После выполнения своей функции нейромедиатор удаляется путем либо всасывания или разложения клеткой, либо выбросом за пределы синапса.

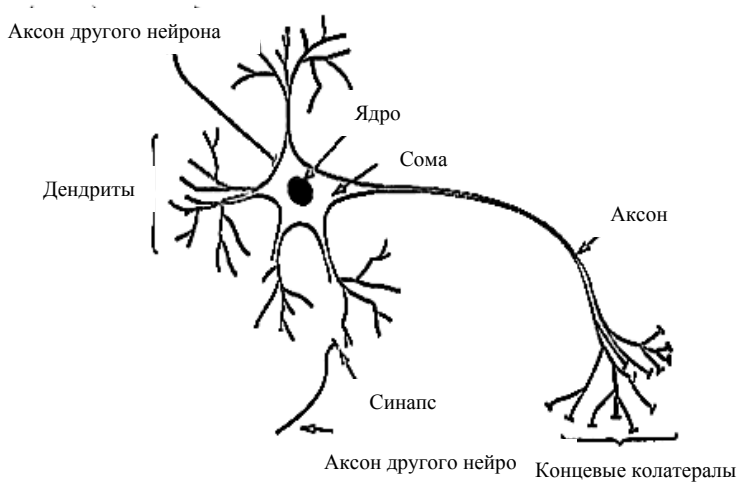

Рис. 1.1. Упрощенная структура биологической нервной клетки

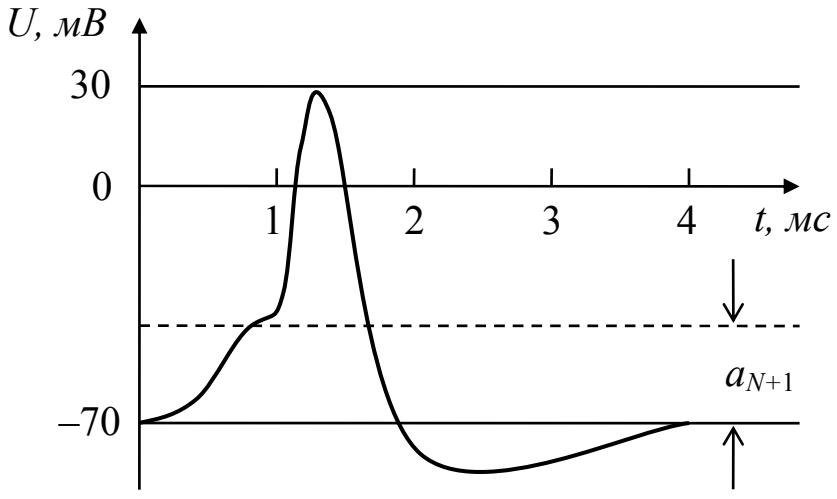

Рис. 1.2. Типичная форма нервного импульса

После генерации нервного импульса в клетке наступает период абсолютной рефрактерности Δt_p, когда нейрон теряет способность вырабатывать очередной сигнал даже при сильном возбуждении. По окончании Δt_p, наступает период относительной рефракции Δt_0, за который

порог срабатывания возвращается к равновесному значению. В это время клетку можно активировать, но только прикладывая более сильные возбуждения. В естественных процессах, как правило, выполняется $\Delta t_0 >> \Delta t_p$.

Количество взаимодействующих друг с другом нервных клеток чрезвычайно велико. Считается, что человеческий мозг содержит порядка 10^{11} нейронов, соединенных между собой примерно 10^{15} связями (до 10^4 на один нейрон). Каждый нейрон имеет свои веса связей и свое пороговое значение, определяемые его расположением и выполняемой функцией. Громадное число нейронов и межнейронных связей приводит к тому, что ошибки в срабатывании отдельных нейронов (до 10 % от общего числа) слабо влияют на конечный результат, несколько ухудшая, может быть, его точность. Вторая важная особенность нервных систем – высокая скорость их функционирования, несмотря на невысокое быстродействие ($\sim 10^{-3}$ с) отдельных клеток, что достигается благодаря параллельной обработке информации лавинообразно нарастающим количеством элементарных вычислительных ячеек – нейронов. Если бы удалось создать вычислительную систему с аналогичной степенью параллельности независимых операций при существующих в СБИС скоростях их выполнения ($\sim 10^{-9}$ с на такт), ее потенциальные возможности трудно даже представить.

1.2. Аналоговая модель Ходжкина–Хаксли

Моделирование электрохимических процессов, составляющих основу функционирования нервных клеток, издавна привлекало внимание исследователей. В 50-е годы прошлого века А. Ходжкин и А. Хаксли на основе экспериментов с мембраной гигантского аксона кальмара (диаметр 0,5–1 мм, длина – до нескольких сантиметров) установили, что ток I_m, текущий через мембрану, может быть представлен суммой нескольких компонентов – током смещения через эквивалентную емкость $C_Э$ мембраны, ионными токами I_{Na}, I_K проводимости каналов для Na^+ и K^+, а также током утечки I_y

$$I_m = C_Э \frac{\partial U}{\partial t} + I_{Na} + I_K + I_y, \qquad (1.1)$$

где U – электрический потенциал мембраны, то есть эквивалентная схема единичного отрезка нервного волокна, может быть представлена в виде схемы, изображенной на рисунке 1.3. Ионные батареи E_{Na} и E_K отражают тенденцию ионов натрия диффундировать внутрь мембраны, а ионов калия – наружу. Переменные сопротивления отражают нелинейную зависимость проводимостей g_{Na}, g_K ионных каналов от потенциала мембраны U.

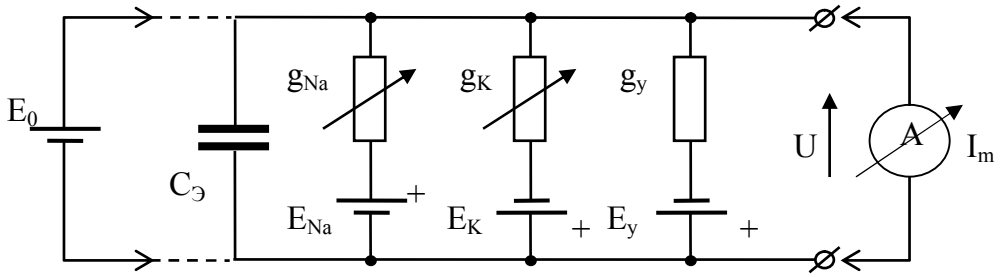

Рис. 1.3. Эквивалентная схема единичной площади мембраны

В результате ионные токи и ток утечки были записаны следующим образом

$$\begin{cases} I_{Na} = g_{Na}(U,t)(U - E_{Na}); \\ I_K = g_K(U,t)(U - E_K); \\ I_y = g_y(U,t)(U - E_y), \end{cases} \tag{1.2}$$

где равновесные параметры исследуемой модели имели значения: $E_{Na} \approx$ 115 мВ; $E_K \approx 12$ мВ; $E_y \approx 10$ мВ; $g_y \approx 0{,}3$ мСм/см^2). Для описания зависимостей g_K, g_{Na} от мембранного потенциала и времени Ходжкин и Хаксли на основе экспериментальных измерений ввели следующие функции

$$g_K(U,t) = g_K^0 \cdot n^4; \quad g_{Na}(U,t) = g_{Na}^0 \cdot m^3 p, \tag{1.3}$$

где $g_K^0 \approx 36 \frac{мСм}{см^2}$; $g_{Na}(U,t) \approx 120 \frac{мСм}{см^2}$ – максимальные проводимости каналов, а имеющие смысл вероятностей n, m, p удовлетворяют кинетическим уравнениям

$$\begin{aligned} \frac{\partial n}{\partial t} &= \alpha_n(U) - \beta_n(U)n; \\ \frac{\partial m}{\partial t} &= \alpha_m(U) - \beta_m(U)m; \\ \frac{\partial p}{\partial t} &= \alpha_p(U) - \beta_p(U)p, \end{aligned} \tag{1.4}$$

коэффициенты α_i, β_i которых определяются эмпирически для каждого конкретного случая.

Система нелинейных дифференциальных уравнений 4-го порядка (1.1–1.4), дополненная учетом влияния синаптического тока, представ-

ляет собой каноническую модель электрогенеза нервной клетки. Решение этой системы имеет вид скачкообразного изменения мембранного потенциала $U(t)$ на достаточно большую (~100 мВ) величину (рис. 1.4), которое распространяется по поверхности мембраны. Значение полученных результатов состояло в том, что для других биообъектов аппроксимация кинетики ионных токов (со своим набором эмпирических функций и констант) оказалась аналогичной, то есть система уравнений Ходжкина–Хаксли стала образцовой моделью, по которой проверяются нелинейные эффекты в нервных клетках.

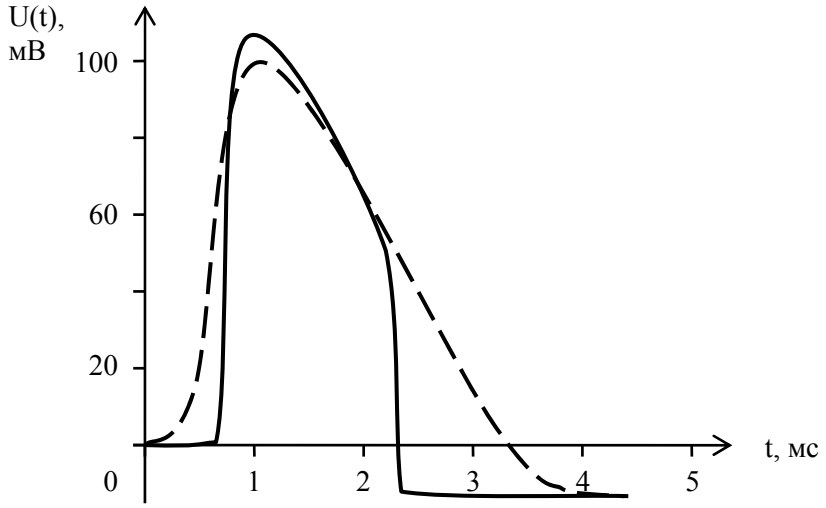

Рис. 1.4. Форма нервного импульса – спайка
(—— – из модели Ходжкина–Хаксли; – – – – по результатам измерений)

Необходимо отметить, однако, что рассмотренная модель описывает только электрическую активность нейронов. Более сложные модели, включающие химические взаимодействия и содержащие до 150 параметров, требуют для своего расчета супермощных компьютеров, хотя получаемые на их основе результаты качественно практически не отличаются от модели Ходжкина–Хаксли.

1.3. Оптоэлектронная модель нейрона

Электрические модели нейронов типа Ходжкина–Хаксли не слишком удобны для построения в элементной базе ИС ввиду необходимости реализации заметной емкостной составляющей и большого числа управляемых источников тока (напряжения). Более того, огромное число синаптических связей, осуществляемых с помощью электрических межсоединений, в любых технологиях СБИС приводит к значительному росту размеров и энергопотребления НС. В этом плане наиболее перспективно использование оптики, где в условиях однонаправленности сигналов (от источника света к приемнику) и отсутствии их взаимного влияния возможно построение даже трехмерных структур ИНС. И здесь может оказаться целесообразным использование

оптоэлектронной модели нейрона, предложенной на кафедре физики полупроводников и микроэлектроники ВГУ.

Рассматриваемая модель основана на эффекте «замороженной» фотопроводимости в полупроводниках, имеющих локальные уровни прилипания с аномально большим временем жизни носителей (для определенности будем считать, что это – дырки). Такое свойство материала позволяет реализовать структуру фоторезистора с памятью (ФРП, рис. 1.5), при попадании на которую фотонов с энергией $h\nu$, достаточной для образования электронно-дырочных пар, электроны переходят в зону проводимости и устремляются к аноду, а дырки захватываются ловушками. В результате в объеме полупроводника накапливается положительный заряд, что приводит к поступлению из катода дополнительных электронов, то есть количество дырок на ловушках пропорционально потоку фотонов, а ток через полупроводник пропорционален количеству захваченных ловушками дырок. Память описанной структуры заключается в хранении на уровнях прилипания положительного заряда, причем, чтобы в отсутствие света ток через полупроводник отсутствовал, последовательно в цепь катода включается фотодиод с малым темновым током, образуя совместно с ФРП прибор фотопамяти (ПФП).

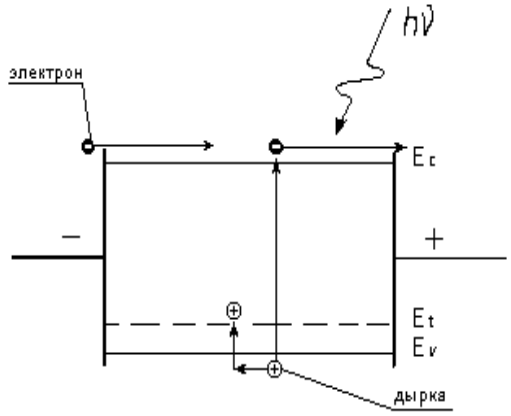

Рис. 1.5. Фоторезистор с памятью

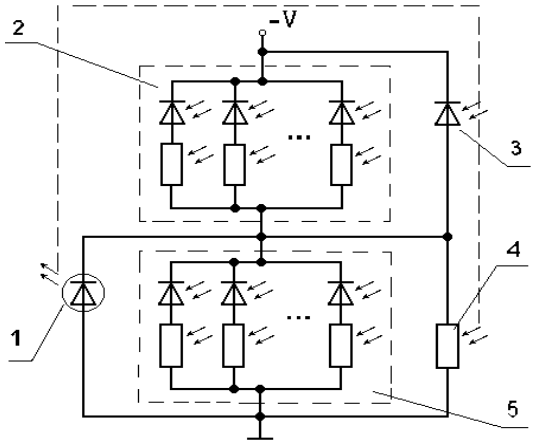

Рис. 1.6. Оптоэлектронный нейрон

Параллельным соединением таких приборов можно смоделировать множество входных синапсов нейрона, принимающих от своих соседей световые сигналы, которые преобразуются в электрические, одновременно взвешиваясь благодаря пропорциональности тока через ПФП накопленному в нем заряду. Далее токи от ПФП-синапсов объединяются и подаются на пороговое устройство, в качестве которого можно использовать полупроводниковый инжекционный лазер. Пока ток через лазер не превышает критического значения, он работает в светодиодном режиме, его излучение имеет широкую спектральную полосу, низкую мощность и большую угловую апертуру. При достижении током порога мощность излучения резко увеличивается, уменьшается угловая апертура, свет становится монохроматичным. Один из вари-

антов предлагаемой структуры нейрона представлен на рисунке 1.6. При отсутствии возбуждения через приемники излучения ПФП 2 и ПФП 5 протекают малые темновые токи, которые с приходом светового импульса резко возрастают. Усиление тока зависит от накопленных на ловушках дырок и увеличивается с каждым новым возбуждением. Усиленные токи от ПФП 2, выполняющего роль возбуждающих связей, идут на полупроводниковый лазер, который, если суммарный ток превышает порог, излучает монохроматичный свет высокой интенсивности. Для расширения функциональных возможностей модели в состав нейрона вводятся тормозящие связи, образованные ПФП 5, которые шунтируют источник излучения.

За образование аналога выходного нервного импульса (спайка) в оптоэлектронном нейроне отвечают фотодиод 3 и фоторезистор с памятью 4, оптически связанные с лазером 1. В момент включения лазера фотодиод переходит в проводящее состояние, делая излучение лазера более мощным и независимым от продолжительности возбуждения. Из-за большей инерционности фоторезистора по сравнению с фотодиодом только через 1 мкс ток фоторезистора возрастет до величины, способной, шунтировав лазер, погасить его. Благодаря наличию у фоторезистора 4 памяти (порядка нескольких микросекунд) в рассматриваемом нейроне после прекращения излучения лазера наступает состояние рефрактерности, так как лазер остается зашунтированным, пока не рекомбинируют накопленные на ловушках ФРП 4 дырки. Если в биологическом нейроне длительность нервных импульсов порядка миллисекунды, то в нашей модели время «забывания» изменения веса связи (время жизни дырок на ловушках) в тысячи раз меньше, то есть быстродействие рассматриваемого аналога биопрототипа будет примерно на три порядка выше.

Анализ временных характеристик модели проводился для входного узла нейрона на основе уравнения квазинейтральности и двух (для электронов и дырок) уравнений непрерывности. В предположении равномерной генерации носителей во всем объеме полупроводникового материала соответствующая система уравнений может быть представлена в виде (1.5)

$$\begin{cases} \dfrac{dn}{dt} = g - \gamma_r\left(np - n_0 p_0\right) - S_n \mathrm{v}_t N_t\left[\, n\left(1-f\right) - n_1 f\right]; \\ \dfrac{dp}{dt} = g - \gamma_r\left(np - n_0 p_0\right) - S_p \mathrm{v}_t N_t\left[\, pf - p_1\left(1-f\right)\right]; \\ \dfrac{d\left(p_t - n_t\right)}{dt} = S_p \mathrm{v}_t N_t\left[\, pf - p_1\left(1-f\right)\right] - S_n \mathrm{v}_t N_t\left[\, n\left(1-f\right) - n_1 f\right]; \end{cases} \qquad (1.5)$$

$$n_1 = N_c \exp\left(-\frac{E_c - E_t}{kT} \right);$$

$$p_1 = N_{\mathrm{V}} \exp\left(-\frac{E_t - E_{\mathrm{V}}}{kT} \right),$$

где n, p – концентрации электронов и дырок в объеме полупроводника; n_t, p_t – концентрации электронов и дырок на ловушках; n_0, p_0 – равновесные концентрации электронов и дырок; N_t – концентрация ловушек; $g_n = g_p = g$ – скорость генерации электронов и дырок в объеме полупроводника; S_n, S_p – эффективное сечение захвата электронов и дырок; v_t – среднее значение тепловой скорости электронов; γ_r – постоянная рекомбинации; f – вероятность захвата электрона; E_t – энергия ловушечного уровня, отсчитанная от дна зоны проводимости.

Для кремния n–типа с ловушками для дырок, то есть при $n_t = 0$, $p_t = N_t f_p = N_t(1 - f)$ систему (1.5) можно переписать в виде

$$\begin{cases} \dfrac{dn}{dt} = g - \gamma_r \left(np - n_0 p_0 \right) - S_n \mathrm{v}_t N_t \left[\, nf_p - \left(1 - f\right) n_1 \right]; \\ \dfrac{dp}{dt} = g - \gamma_r \left(np - n_0 p_0 \right) - S_p \mathrm{v}_t N_t \left[\, p\left(1 - f_p\right) - f_p p_1 \right]; \\ \dfrac{df_p}{dt} = S_p \mathrm{v}_t \left[\, p\left(1 - f_p\right) - f_p p_1 \right] - S_n \mathrm{v}_t \left[\, nf_p - \left(1 - f\right) n_1 \right], \end{cases} \qquad (1.6)$$

по форме схожим с системой (1.4) Ходжкина–Хаксли.

Численное решение этой системы (методом Рунге–Кутта 4-го порядка) для различных значений электрофизических параметров показало возможность получения скачка концентрации электронов $n(t)$ (спайка) при подаче возбуждающих импульсов длительностью ~10^{-6} с. Наиболее приемлемый результат (с формой импульса типа, изображенного на рис. 1.4) обеспечивался при следующих значениях параметров: $n_0 = 10^{11}$ см$^{-3}$, $N_t = 5 \cdot 10^{16}$ см$^{-3}$; E_t = = 1,12 В; $g = 10^{15}$ см$^{-3}$с$^{-1}$; $\gamma_r = 10^{-10}$ см^{3}с$^{-1}$; $S_n = S_p = 10^{-18}$ см$^{-2}$, $\mathrm{v}_t = 10^7$ см/с, довольно типичных для полупроводниковых материалов, кроме концентрации ловушек N_t, которая в обычных условиях работы составляет величину 10^{11}…10^{12} см$^{-3}$. Более высокие значения N_t возможны в сильно дефектных полупроводниках или при наличии мощного нейтронного облучения.

Оценка характерных размеров оптоэлектронного нейрона при реализации в рамках интегральной технологии с площадью инжекционного лазера 300 × 300 мкм^2 и светочувствительных площадок 10 × 10 мкм^2 показывает, что на кристалле 50 × 50 мм^2 можно разместить до 10^4 нейронов с 10^7…10^8 связями, то есть получить достаточно мощную вычислительную

структуру. Более подробное обсуждение возможностей оптических НС будет рассмотрено позднее, в разделе, посвященном вопросам практической реализации НС.

Контрольные вопросы

1. Расскажите об основных тенденциях развития кибернетики начала XXI века.
2. Какую из основных парадигм искусственного интеллекта реализуют ИНС? В чем она заключается?
3. При решении каких задач наиболее полно проявляется преимущество ИНС перед традиционными ЭВМ?
4. Опишите биологическую структуру и основы функционирования нервных клеток живых организмов.
5. Приведите типичную форму нервного импульса (спайка), расскажите об условиях его возникновения.
6. Что такое процесс рефракции, и как он реализуется в нервной клетке?
7. Где хранится информация, и как производится ее обработка в системе нервных клеток?
8. Что послужило основой для разработки аналоговой модели нейрона Ходжкина–Хаксли? Какова эквивалентная схема единичного отрезка мембраны?
9. Приведите систему дифференциальных уравнений, описывающих модель Ходжкина–Хаксли, и охарактеризуйте результат ее решения.
10. Какие сложности возникают при реализации нейрона Ходжкина–Хаксли в элементной базе ИС?
11. На каком эффекте основана оптоэлектронная модель нейрона? В чем он заключается?
12. Опишите структуру и функционирование прибора фотопамяти (ПФП) в модели оптонейрона.
13. Как реализуются основные элементы биопрототипа нейрона в его оптоэлектронной модели?
14. Приведите систему дифференциальных уравнений для описания временных характеристик оптонейрона и охарактеризуйте результат ее решения.
15. Что показывает сравнительный анализ характеристик электрической и оптоэлектронной моделей биопрототипа нейрона?

2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ НЕЙРОНОВ

Из анализа материала предыдущего раздела следует, что с точки зрения обработки информации каждый нейрон можно считать своеобразным процессором, который суммирует с соответствующими весами сигналы от других нейронов, выполняет нелинейную обработку полученной суммы и формирует результирующий сигнал для передачи связанным с ним нейронам. На основе принципов функционирования биологических нейронов были созданы различные математические модели, реализующие (в большей или меньшей степени) свойства природной нервной клетки. Основу большинства таких моделей составляет структура формального нейрона (ФН) МакКаллока–Питтса (1943), представленная на рисунке 2.1, где компоненты входного вектора $\vec{x}$ $(x_1, x_2, \ldots, x_N)$ суммируются с учетом весов w_{ij} и сравниваются с пороговым значением w_{i0}. Выходной сигнал ФН y_i определяется как

$$y_i(t) = f(u_i) = f\left(\sum_{j=1}^{N} w_{ij} x_j(t) + w_{i0} \right), \tag{2.1}$$

где в общем случае нелинейная функция преобразования $f(u_i)$ называется функцией активации. Коэффициенты w_{ij} соответствуют весам синаптических связей: положительное значение w_{ij} – возбуждающим, отрицательное w_{ij} – тормозящим синапсам, $w_{ij} = 0$ означает отсутствие связи между i-м и j-м нейронами.

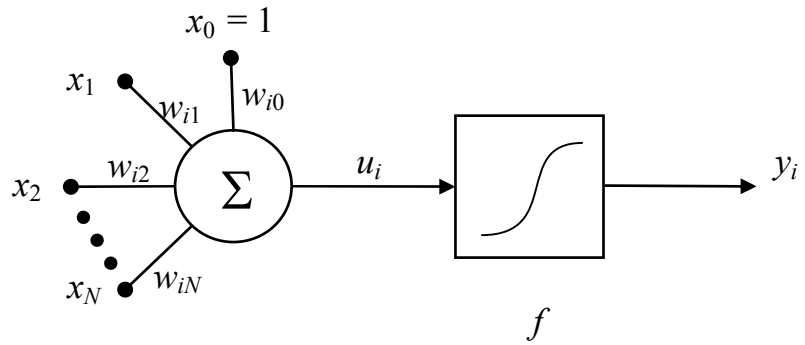

Рис. 2.1. Структура модели формального нейрона МакКаллока–Питтса

Функция активации ФН – это пороговая функция вида

$$f(u) = \begin{cases} 1, & \text{если } u > 0; \\ 0, & \text{если } u \le 0, \end{cases} \tag{2.2}$$

хотя в принципе набор используемых в моделях нейронов $f(u)$ достаточно разнообразен (табл. 2.1), поскольку их свойства, особенно непрерывность, оказывают значительное влияние на выбор способа обучения нейрона (подбор w_{ij}). Наиболее распространенными функциями активации являются пороговая, линейная (в том числе с насыщением) и сигмоидальные – логистическая и гиперболический тангенс (рис. 2.2). Заметим, что с уменьшением α

сигмоиды становятся более пологими, а при $\alpha \to \infty$ превращаются в пороговую и сигнатурную функции соответственно. В числе их достоинств следует также упомянуть относительную простоту и непрерывность производных и свойство усиливать слабые сигналы лучше, чем большие.

Таблица 2.1

Функции активации нейронов

Название	Формула	Область значений
Линейная	$f(u) = ku$	$(-\infty, \infty)$
Полулинейная	$f(u) = \begin{cases} ku, & u > 0 \\ 0, & u \le 0 \end{cases}$	$(0, \infty)$
Логистическая (сигмоидальная)	$f(u) = \dfrac{1}{1 + e^{-\alpha u}}$	(0, 1)
Гиперболический тангенс (сигмоидальная)	$f(u) = \dfrac{e^{\alpha u} - e^{-\alpha u}}{e^{\alpha u} + e^{-\alpha u}} \equiv th(\alpha u)$	(–1, 1)
Экспоненциальная	$f(u) = e^{-\alpha u}$	$(0, \infty)$
Синусоидальная	$f(u) = \sin(u)$	(–1, 1)
Сигмоидальная (рациональная)	$f(u) = \dfrac{u}{\alpha + \lvert u \rvert}$	(–1, 1)
Линейная с насыщением	$f(u) = \begin{cases} -1, & u \le -1 \\ u, & -1 < u < 1 \\ 1, & u \ge 1 \end{cases}$	(–1, 1)
Пороговая	$f(u) = \begin{cases} 0, & u < 0 \\ 1, & u \ge 0 \end{cases}$	(0, 1)
Модульная	$f(u) = \lvert u \rvert$	$(0, \infty)$
Сигнатурная	$f(u) = \begin{cases} 1, & u > 0 \\ -1, & u \le 0 \end{cases}$	(–1, 1)
Квадратичная	$f(u) = u^2$	$(0, \infty)$

Помимо выбора $f(u)$ важным фактором является выбор стратегии обучения. При обучении *с учителем* для каждого входного $\vec{x}^{(k)}$ должны быть известны ожидаемые выходные сигналы $\vec{d}^{(k)}$, а подбор w_{ij} должен быть организован так, чтобы фактические значения $y_i^{(k)}$ были наиболее близки к $d_i^{(k)}$. При обучении *без учителя* подбор весовых коэффициентов проводится на основании либо конкуренции нейронов между собой, либо с учетом корреляции обучающих и выходных сигналов. В этом случае (в отличие от обучения с учителем) прогнозирование выходных сигналов нейрона на этапе адаптации невозможно. Наиболее распространенные модели нейронов, реализующие каждый из указанных подходов, представлены на рисунке 2.2.

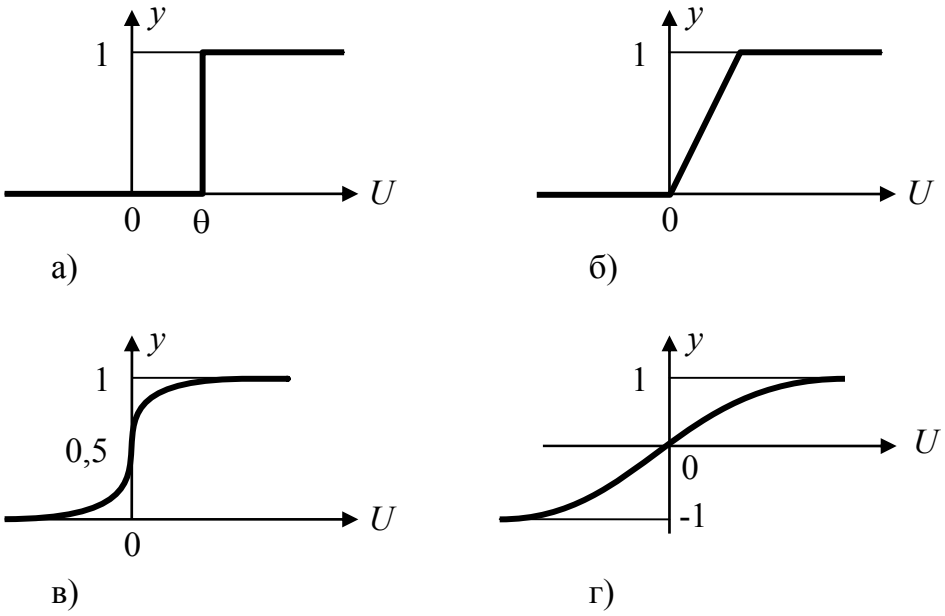

Рис. 2.2. Примеры активационных функций:
а – пороговая; б – полулинейная с насыщением; в – логистическая; г – гиперболический

2.1. Персептрон

Простой персептрон – это ФН МакКаллока–Питтса со *структурой*, представленной на рисунке 2.1, и соответствующей стратегией обучения. *Функция активации* – пороговая, вследствие чего выходные сигналы могут принимать только два значения

$$y_i(u_i) = \begin{cases} 1, & \text{при } u_i \geq 0; \\ 0, & \text{при } u_i < 0, \end{cases} \tag{2.3}$$

где для выходного сигнала сумматора

$$U_i = \sum_{j=0}^{N} w_{ij} x_j \tag{2.4}$$

входной вектор дополнен нулевым членом $x_0 = 1$, формирующим сигнал поляризации, то есть $\vec{x} = (x_0, x_1, x_2, \ldots, x_N)$.

Обучение – с учителем по правилу персептрона в соответствии с алгоритмом:

1) при начальных значениях w_{ij} (выбранных, как правило, случайным образом) на вход подается обучающий $\vec{x}$, рассчитывается y_i и по результатам сравнения y_i с известным d_i уточняются значения весов;
2) если $y_i = d_i$, то $w_{ij} = const$;
3) если $y_i = 0$, а $d_i = 1$, то $w_{ij}(t+1) = w_{ij}(t)+x_j$, где t – номер итерации;
4) если $y_i = 1$, а $d_i = 0$, то $w_{ij}(t+1) = w_{ij}(t)–x_j$.

После уточнения весовых коэффициентов подается следующая обучающая пара $\vec{x} \Leftrightarrow d_i$ и значения w_{ij} уточняются заново. Процесс повторяется многократно на всех обучающих выборках до минимизации разницы между всеми y_i и d_i. Вообще говоря, правило персептрона является частным случаем предложенного позднее правила Видроу–Хоффа

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij}; \\ \Delta w_{ij} &= x_j \left(d_i - y_i \right), \end{aligned} \tag{2.5}$$

где d_i, y_i могут принимать любые значения.

Минимизация различий между фактическими y_i и ожидаемыми d_i выходными сигналами нейрона может быть представлена как минимизация некоторой (целевой) функции погрешности $E(w)$, чаще всего определяемой, как

$$E(w) = \frac{1}{2} \sum_{k=1}^{p} \left[y_i^{(k)} - d_i^{(k)} \right]^2, \tag{2.6}$$

где p – количество обучающих выборок. Оптимизация $E(w)$ по правилу персептрона является безградиентной, при большом p количество циклов обучения и его длительность быстро возрастают без всякой гарантии достижения минимума целевой функции. Устранить эти недостатки можно только при использовании непрерывных $f(u)$ и $E(w)$.

2.2. Сигмоидальный нейрон

Структура – ФН МакКаллока–Питтса (рис. 2.1).

Функции активации – униполярный f_1 (табл. 2.1, рис. 2.2, в) или биполярный f_2 (табл. 2.1, рис. 2.2, г) сигмоиды, непрерывно дифференцируемые во всей области определения, причем как $f_1'(u) = \alpha f_1(u)\left[1 - f_1(u)\right]$, так и $f_2'(u) = \alpha\left[1 - f_2^2(u) \right]$ имеют практически одинаковую колоколообразную форму с максимумом при $u = 0$ (рис. 2.3).

Обучение – с учителем путем минимизации целевой функции (2.6) с использованием градиентных методов оптимизации, чаще всего алгоритма наискорейшего спуска (АНС). Для одной обучающей пары ($p = 1$) j-я составляющая градиента согласно (2.4), (2.6) имеет вид

$$\nabla_j E(w) = \frac{dE}{dw_{ij}} = e_i x_j \frac{df(u_i)}{du_i} = \delta_i x_j, \tag{2.7}$$

где $\delta_i = e_i \dfrac{df(u_i)}{du_i}$; $e_i = y_i - d_i$. При этом значения w_{ij} уточняются либо дискретным

$$w_{ij}(t+1) = w_{ij}(t) - \eta\delta_i x_j, \tag{2.8}$$

либо аналоговым способом из решения разностного уравнения

$$\frac{dw_{ij}}{dt} = -\mu\delta_i x_j, \tag{2.9}$$

где η, $\mu \in (0,1)$ играют роль коэффициентов обучения, от которых сильно зависит его эффективность. Наиболее быстрым (но одновременно наиболее трудоемким) считается метод направленной минимизации с адаптивным выбором значений η, μ.

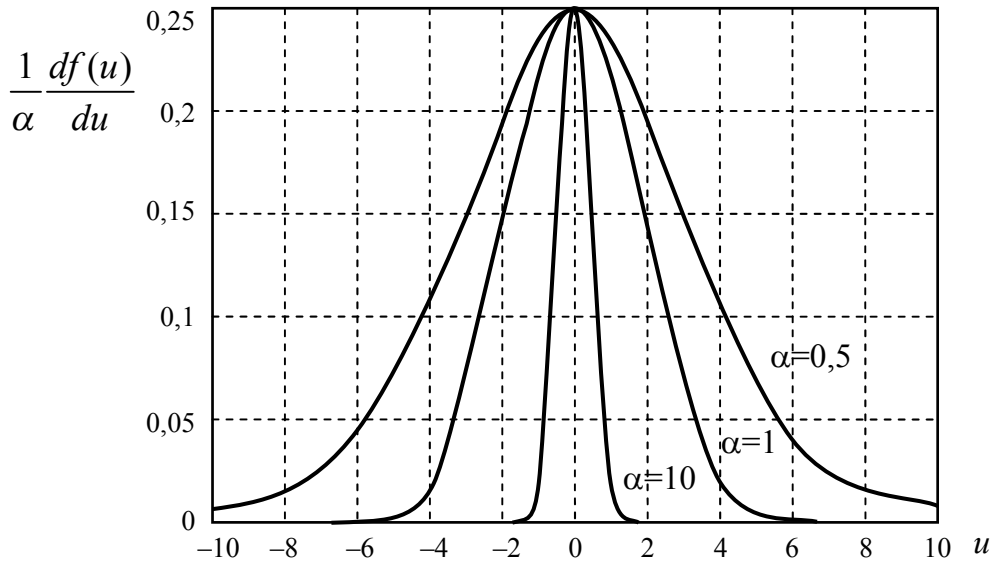

Рис. 2.3. График производной от сигмоидальной функции при различных α

Следует отметить, что применение градиентных методов обучения нейрона гарантирует достижение только локального экстремума, который для полимодальной $E(w)$ может быть достаточно далек от глобального минимума. В этом случае результативным может оказаться обучение с моментом (ММ)

$$\Delta w_{ij}(t+1) = -\eta\delta_i x_j + \alpha\Delta w_{ij}(t), \tag{2.10}$$

где $0 < \alpha < 1$ – коэффициент момента, или использование стохастических методов оптимизации.

2.3. Адаптивный линейный нейрон

Структура предложенного Б. Видроу нейрона «ADALINE» (ADAptive LInear NEuron) – ФН МакКаллока–Питтса (рис. 2.1) с *функцией активации* типа signum (табл. 2.1), то есть

$$y_i(u_i) = \begin{cases} 1, & \text{при } u_i > 0; \\ -1, & \text{при } u_i \le 0. \end{cases} \qquad (2.11)$$

Обучение – с учителем путем подбора w_{ij} в процессе минимизации целевой функции

$$E(w) = \frac{1}{2}e_i^2 = \frac{1}{2}\left[d_i - \sum_{j=0}^{N} w_{ij}x_j \right]^2 \qquad (2.12)$$

с использованием градиентных методов, поскольку в $E(w)$ входят только линейные члены. Уточнение w_{ij} – либо дискретно согласно

$$w_{ij}(t+1) = w_{ij}(t) + \eta e_i x_j, \qquad (2.13)$$

либо аналогово – путем решения разностного уравнения

$$\frac{dw_{ij}}{dt} = -\mu e_i x_j. \qquad (2.14)$$

Нейроны типа «ADALINE» имеют относительно простую схемную реализацию, включающую интеграторы, сумматоры и элементы задержки. В практических приложениях эти нейроны всегда используются группами, образуя слои, называемые «MADALINE» (Many ADALINE), где каждый нейрон обучается по правилам (2.13), (2.14).

2.4. «Instar» и «Outstar» Гроссберга

Структуры «Instar» и «Outstar», предложенные С. Гроссбергом (рис. 2.4, а,б), представляют собой взаимодополняющие элементы: «Instar» адаптирует веса связей нейрона к входным сигналам (компонентам $\vec{x} = [x_1, x_2, \ldots, x_N]$), а «Outstar» – к выходным (компонентам $\vec{y} = [y_1, y_2, \ldots, y_M]$).

Функции активации – чаще всего линейные (табл. 2.1).

Обучение – по правилам Гроссберга: для «Instar» (рис. 2.4, а) –

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_i \left[x_j - w_{ij}(t) \right], \qquad (2.15)$$

для «Outstar» (рис. 2.4, б) –

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_i \left[y_j - w_{ij}(t) \right]. \qquad (2.16)$$

Входные данные для обучения (компоненты $\vec{x}$), как правило, выражаются в нормализованной форме, когда $\|\vec{x}\| = 1$, а $\hat{x}_j = \dfrac{x_j}{\sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}}$.

«Instar» и «Outstar» существенно отличаются от предыдущих типов нейронов прежде всего тем, что могут обучаться как *с учителем* (в этом случае $y_i = d_i$), так и *без него*.

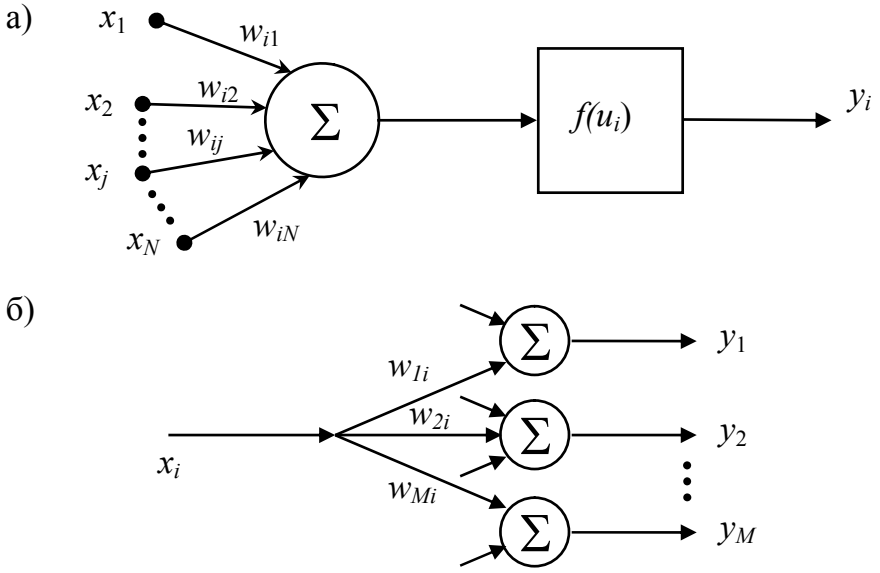

Рис. 2.4. Структурные схемы нейронов Гроссберга:
а – «Instar»; б – «Outstar»

2.5. Модель нейрона Хебба

В процессе исследования свойств нервных клеток Д. Хебб заметил, что связь между двумя клетками усиливается, если обе клетки активируются одновременно, и предложил формальное правило обучения, в соответствии с которым вес w_{ij} нейрона изменяется пропорционально произведению его входного и выходного сигналов. Правило Хебба может применяться для НС различных типов с любыми функциями активации отдельных нейронов.

Структурная схема нейрона Хебба аналогична стандартной структуре ФН (рис. 2.1), *обучение* – по правилу Хебба

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij},$$

где для *обучения с учителем*

$$\Delta w_{ij} = \eta x_j d_i, \tag{2.17}$$

а для *обучения без учителя*

$$\Delta w_{ij} = \eta x_j y_i. \tag{2.18}$$

При обучении по Хеббу веса w_{ij} могут принимать сколь угодно большие значения, поскольку на каждой итерации текущее значение $w_{ij}(t)$

суммируется с его приращением Δw_{ij}. Обеспечить сходимость процесса обучения возможно: 1) введением коэффициента забывания γ

$$w_{ij}(t+1) = w_{ij}(t)\left(1-\gamma\right) + \Delta w_{ij}, \tag{2.19}$$

где при рекомендуемом γ < 0,1 нейрон сохраняет бóльшую часть информации, накопленной в процессе обучения, и получает возможность стабилизировать w_{ij} на определенном уровне; 2) использованием для обучения линейных нейронов, где стабилизации не происходит даже при введении γ, модифицированного правила Хебба–Ойя, согласно которому

$$\Delta w_{ij} = \eta y_i \left(x_j - y_i w_{ij} \right), \tag{2.20}$$

что приводит к ограничению $|w|=1$, обеспечивающему конечность значений весовых коэффициентов.

2.6. Нейроны типа WTA

Нейроны типа WTA (Winner Takes All – Победитель получает все) представляют группу конкурирующих между собой нейронов, получающих одни и те же входные сигналы x_j (рис. 2.5). Сравнением выходных значений сумматоров (2.4) определяется нейрон-победитель с максимальной величиной u_i, на его выходе устанавливается сигнал $y_i = 1$, остальные (проигравшие) нейроны переходят в состояние 0, что блокирует процесс уточнения их весовых коэффициентов. Веса же победившего нейрона уточняются по упрощенному (ввиду бинарности значений выходных сигналов) правилу Гроссберга

$$w_{ij}(t+1) = w_{ij}(t) + \eta\left[x_j - w_{ij}(t) \right] \tag{2.21}$$

с нормализацией x_j и w_{ij}.

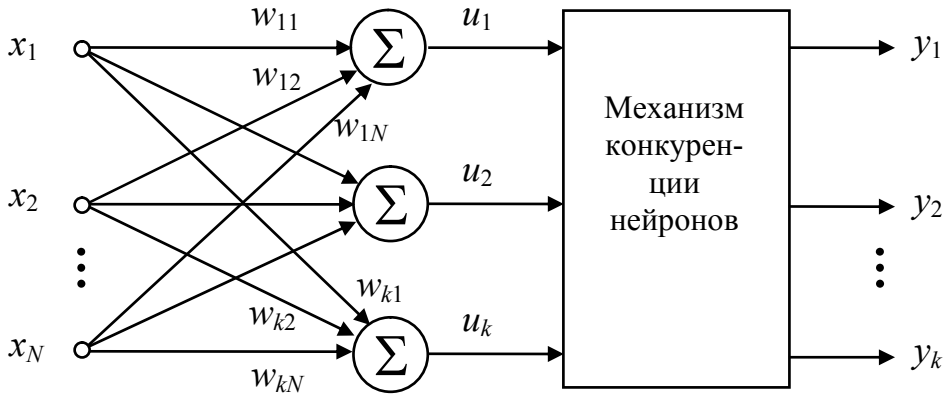

Рис. 2.5. Схема соединения нейронов типа WTA

Следствием этой конкуренции становится самоорганизация процесса обучения, ибо уточнение весов происходит таким образом, что для каждой группы близких по значениям обучающих $\vec{x}^{(k)}$ побеждает свой нейрон, который при функционировании распознает именно эту категорию $\vec{x}$. Серьезной проблемой при обучении WTA остается проблема «мертвых» нейронов, которые после инициализации ни разу не победили в конкурентной борьбе, ибо их наличие уменьшает число прошедших обучение нейронов, соответственно увеличивая общую погрешность распознавания данных. Для решения проблемы используют модифицированное обучение, основанное на штрафовании (временной дисквалификации) наиболее активных нейронов.

2.7. Стохастическая модель нейрона

В отличие от детерминированных моделей (2.1)…(2.6), в стохастической модели выходное состояние нейрона зависит не только от взвешенной суммы u_i, но и от некоторой случайной переменной, выбираемой при каждой реализации из интервала (0,1). Это означает, что y_i структуры ФН (рис. 2.1) принимает значения ±1 с вероятностью

$$P(y_i = \pm 1) = \left[\, 1 + \exp\left(\mp 2\beta u_i\right)\right]^{-1}, \tag{2.22}$$

где u_i определяется (2.4), а $\beta > 0$ (чаще всего $\beta = 1$).

Процесс обучения нейрона стохастической модели состоит из следующих этапов:

1) расчет u_i (2.4) для каждого нейрона сети;
2) расчет вероятности $P(y_i = \pm 1)$ по формуле (2.22);
3) генерация значения случайной переменной $R \in (0,1)$ и формирование выходных сигналов y_i, если $P(y_i) > R$, или $-y_i$, если $P(y_i) < R$;
4) адаптация весовых коэффициентов w_{ij} (при фиксированных y_i) по используемым правилам, например, при обучении с учителем – по правилу Видроу–Хоффа

$$\Delta w_{ij} = \eta x_j \left(d_i - y_i\right). \tag{2.23}$$

Доказано, что такой подбор w_{ij} минимизирует целевую функцию

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{n}\left[d_i^{(k)} - y_i^{(k)}\right]^2, \tag{2.24}$$

где n – число нейронов, p – количество обучающих выборок.

Контрольные вопросы и задачи

1. Какие процессы в нервной клетке отражает структура ФН МакКаллока–Питтса?
2. Какие особенности имеют сигмоидальные функции активации?
3. Какие стратегии обучения НС вы знаете? Каковы их особенности?
4. Приведите структуру и алгоритм обучения персептрона.
5. Почему проблему обучения нейрона можно свести к минимизации некоторой функции?
6. Опишите достоинства и недостатки градиентных методов оптимизации.
7. Почему «ADALINE» с функцией активации типа signum называют линейным нейроном?
8. Каковы основные особенности структур нейронов «Instar» и «Outstar» Гроссберга?
9. Расскажите о структуре и правилах обучения нейрона Хебба.
10. Какие методы преодоления расходимостей весов при обучении по Хеббу вы знаете?
11. В чем заключается механизм конкуренции и правила обучения нейронов типа WTA?
12. В чем заключается и как решается проблема «мертвых» нейронов при обучении структур WTA?
13. Укажите принципиальное отличие стохастической модели нейрона от остальных моделей и что это дает?
14. Приведите алгоритм обучения стохастической модели нейрона?
15. В чем заключается правило обучения Видроу–Хоффа?
16. Определите область значений, выражение для производной и ее значение в начале координат для функции активации типа алгебраической сигмоиды $f(u) = \dfrac{u}{\sqrt{1+u^2}}$.
17. Пусть $x_1, x_2, \ldots x_N$ – компоненты вектора входных сигналов, подаваемых на вход нейрона с порогом w_0 и логистической функцией активации (табл. 2.1), где α – произволен. Как нужно изменить компоненты $x_1, x_2, \ldots x_N$, чтобы получить на выходе прежний сигнал при $\alpha = 1$?
18. Нейрон j получает входной сигнал от четырех других нейронов, уровни возбуждения которых равны 10; –20; 4; –2, а соответствующие веса связей – 0,8; 0,2; –1,0; –0,9. Вычислите выходной сигнал нейрона, если его функция активации:

а) пороговая; б) линейная (с $k = 1$); в) логистическая (с $\alpha = 1$).

19. Покажите, в каких случаях ФН МакКаллока–Питтса можно аппроксимировать сигмоидальным нейроном?
20. При каких условиях нейрон с сигмоидальной функцией активации может аппроксимировать линейный нейрон?

3. АРХИТЕКТУРА, ПРОБЛЕМЫ ПОСТРОЕНИЯ И ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

3.1. Основные конфигурации ИНС и их свойства

Искусственная нейронная сеть (ИНС) представляет собой совокупность определенным образом соединенных между собой и с внешней средой нейронов трех типов – входных, выходных и промежуточных. С точки зрения топологии ИНС различают:

1) *полносвязные НС* (рис. 3.1, а), где каждый нейрон передает свой выходной сигнал всем остальным, в том числе и самому себе, все входные сигналы подаются всем нейронам, а выходными сигналами могут быть отклики всех или некоторых нейронов после нескольких тактов функционирования сети;

2) *слоистые* или *многослойные НС*, в которых нейроны расположены в несколько слоев. Нейроны нулевого слоя служат для приема входных сигналов и передачи их через точки ветвления всем нейронам следующего (скрытого) слоя без обработки, 1-й слой осуществляет первичную обработку входных сигналов и формирует сигналы для 2-го слоя, который таким же образом формирует сигналы для 3-го и т. д. вплоть до последнего слоя, который образует выход НС. Число нейронов в каждом слое может быть любым и никак не связанным с количеством нейронов в других слоях. Если не оговорено особо, то каждый выходной сигнал i-го слоя подается на входы всех нейронов $(i + 1)$-го.

Среди многослойных НС выделяют следующие типы:

а) НС прямого распространения, в которых отсутствуют обратные связи (ОС), то есть подача выходных сигналов любого слоя на входы нейронов этого же или любого предыдущего слоя.

б) рекуррентные НС, где указанные ОС присутствуют в том или ином варианте.

Наиболее часто используются трехслойные НС прямого распространения с одним скрытым слоем (рис. 3.1, б), которые иногда называют двухслойными из-за отсутствия обработки информации нейронами входного слоя;

3) с*лабосвязные НС,* где нейроны располагаются в узлах прямоугольной или гексагональной решетки. При этом каждый нейрон может быть связан с четырьмя (окрестность фон Неймана, рис. 3.1, в), шестью (окрестность Голея) или восемью (окрестность Мура, рис. 3.1, г) ближайшими соседями.

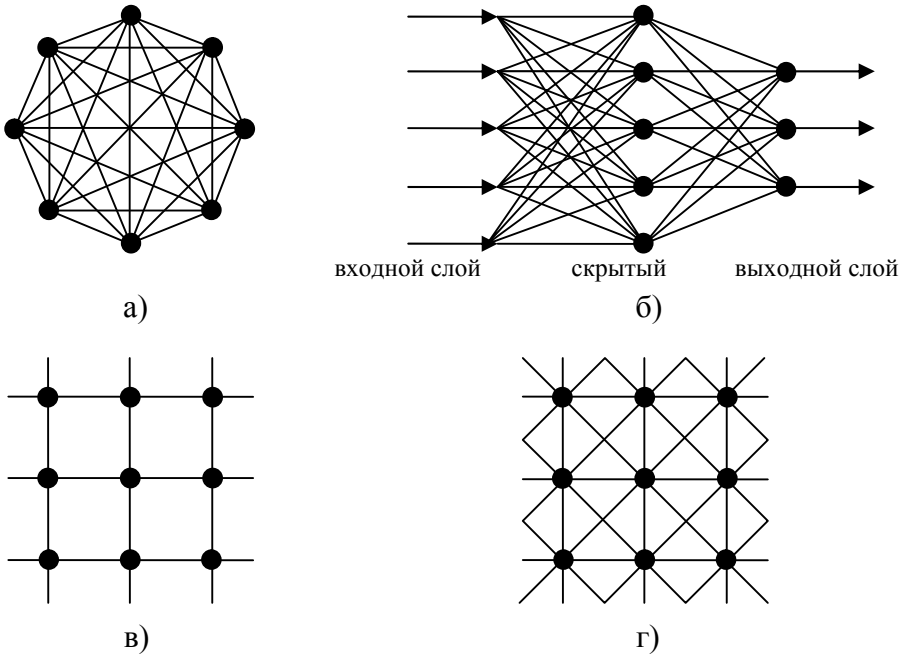

Рис. 3.1. Архитектуры нейронных сетей:
а – полносвязная НС; б – многослойная НС прямого распространения; в – слабосвязная НС с окрестностью фон Неймана; г – слабосвязная НС с окрестностью Мура

Выбор структуры НС обусловлен спецификой решаемой задачи и подчиняется следующим правилам:

– информационная мощность НС возрастает с увеличением числа слоев, нейронов, связей, усилению мощности НС способствует также использование в ее составе различных типов нейронов;

– возможности НС увеличивает введение ОС, однако при этом возникает проблема обеспечения динамической устойчивости сети.

Вопрос о необходимых и достаточных свойствах НС для решения тех или иных задач представляет собой целое направление нейрокомпьютерной науки. Подробные рекомендации здесь практически отсутствуют и в большинстве случаев оптимальный вариант получается на основе интуитивного подбора, хотя в принципе для любого алгоритма существует реализующая его НС.

Подавляющая часть прикладных задач может быть сведена к реализации некоторого многомерного функционального преобразования (вход) $X \rightarrow Y$ (выход), где правильность выходных сигналов необходимо обеспечить в соответствии:

– со всеми примерами обучающей выборки;

– со всеми возможными входными сигналами, не вошедшими в обучающую выборку, что в значительной степени осложняет задачу формирования последней.

Вообще говоря, построить многомерное отображение $X \rightarrow Y$ – это значит представить его с помощью математических операций над не более чем двумя переменными. В результате многолетней научной полемики между

А.Н. Колмогоровым и В.В. Арнольдом в 1957 году была доказана теорема о представимости непрерывных функций нескольких переменных суперпозицией непрерывных функций одной переменной, которая в 1987 году была переложена Хехт–Нильсеном для нейронных сетей: любая функция нескольких переменных может быть представлена двухслойной НС с прямыми полными связями с N нейронами входного слоя, $(2N+1)$ нейронами скрытого слоя с ограниченными функциями активации (например, сигмоидальными) и M нейронами выходного слоя с неизвестными функциями активации.

Из теоремы Колмогорова–Арнольда–Хехт–Нильсена (КАХН) следует, что для любой функции многих переменных существует отображающая ее НС фиксированной размерности, при настройке (обучении) которой могут использоваться три степени свободы:

– область значений сигмоидальных функций активации нейронов скрытого слоя;

– наклон сигмоид нейронов этого слоя;

– вид функций активации нейронов выходного слоя.

Точной оценки числа нейронов K в скрытом слое для каждой конкретной выборки с p элементами нет, однако можно использовать одно из наиболее простых приближенных соотношений:

$$\frac{p}{10} - N - M \le K \le \frac{p}{2} - N - M. \qquad (3.1)$$

Иногда целесообразно использовать НС с бóльшим числом слоев, имеющие (при решении тех же задач) меньшие размерности матриц [W] нейронов скрытых слоев, однако строгой методики построения таких НС пока нет.

3.2. Методы обучения нейронных сетей

Чтобы ИНС с предварительно выбранной начальной архитектурой могла эффективно функционировать, ее необходимо обучить, то есть определить оптимальные значения величин связей w_{ij}, обычно путем минимизации некоторого функционала качества (функции ошибки) $E(\vec{w})$ в процессе итерационной процедуры, где количество итераций t может быть весьма значительным ($t = 10^3 \ldots 10^8$). Функция ошибки $E(\vec{w})$ может быть произвольной, однако наиболее часто используется ее представление в виде (2.6) или (2.24). После выбора совокупности обучающих примеров и способа вычисления $E(\vec{w})$ обучение ИНС превращается в задачу многомерной оптимизации, для решения которой могут быть использованы следующие методы:

– *локальной оптимизации* с вычислением частных производных 1 и 2-го порядков (градиентные методы);

– *глобальной (стохастической) оптимизации* (методы случайного поиска и алгоритмы искусственного отбора).

Основным критерием для сравнения эффективности различных методов обучения ИНС являются вычислительные затраты, то есть количество циклов (время) плюс количество операций.

3.2.1. Градиентные методы

Согласно теории среди детерминированных методов оптимизации наиболее эффективными считаются градиентные методы, связанные с разложением целевой функции $E(\vec{w})$ в ряд Тейлора в окрестности $\vec{p}$ решения $\vec{w}$

$$E(\vec{w}+\vec{p}) = E(\vec{w}) + \left[\vec{g}(\vec{w})\right]^T \vec{p} + \frac{1}{2}\vec{p}^T H(\vec{w})\vec{p} + O(h^3), \qquad (3.2)$$

где $\vec{g}(\vec{w}) = \nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n}\right]^T$ – вектор градиента, $h \equiv \left\|\vec{p}\right\|$, а симметричная квадратная матрица $H(\vec{w})$ производных 2-го порядка

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

Выражение (3.2) можно считать квадратичным приближением $E(\vec{w})$ в ближайшей окрестности *w*. Точкой решения $\vec{w}_p$ будем считать точку, где достигается минимум $E(\vec{w})$ с точностью $O(h^3)$, то есть $\vec{g}(\vec{w}_p) = 0$, а гессиан $H(\vec{w})$ – положительно определен.

В процессе нахождения минимума $E(\vec{w})$ направление поиска $\vec{p}$ и шаг *h* подбираются таким образом, чтобы для каждой очередной точки $\vec{w}(t+1) \equiv \vec{w}_{t+1} = \vec{w}_t + \eta_t \vec{p}_t$ выполнялось условие $E(\vec{w}_{t+1}) < E(\vec{w}_t)$. Поиск продолжается, пока $\left\|\vec{g}(\vec{p})\right\|$ не станет меньше наперед заданной погрешности ε, или не будет превышено максимальное время вычислений (количество итераций). В соответствии с этим универсальный оптимизационный алгоритм обучения ИНС можно представить в следующем виде (считаем, что начальное значение $\vec{w}_{t=0} \equiv \vec{w}_0$ известно):

1. Проверка оптимальности текущего значения $\vec{w}_t$, если «ДА», то «STOP», если «НЕТ», то переход к пункту 2.
2. Определение вектора направления оптимизации $\vec{p}_t$ для точки w_t.
3. Выбор шага η_t в направлении $\vec{p}_t$, при котором выполняется условие $E(w_{t+1}) < E(w_t)$.

4. Определение нового решения $\vec{w}_{t+1} = \vec{w}_t + \eta_t \vec{p}_t$ и соответствующих ему $E(\vec{w}_{t+1}), \vec{g}(w_{t+1}), H(w_{t+1})$ и возврат к пункту 1.

3.2.1.1. Алгоритм наискорейшего спуска (АНС)

Если в разложении (3.2) ограничиться линейным приближением, то для выполнения соотношения $E(\vec{w}_{t+1}) < E(\vec{w}_t)$ достаточно подобрать $\vec{g}(\vec{w}_t)^T \vec{p} < 0$, чему однозначно удовлетворяет выбор

$$\vec{p}_t = -\vec{g}(\vec{w}_t) \tag{3.3}$$

в методе наискорейшего спуска. Ограничение линейным приближением в АНС не позволяет использовать информацию о кривизне $E(\vec{w})$, что обусловливает медленную сходимость метода (она остается линейной). Более того, вблизи точки решения, когда градиент принимает малые значения, процесс минимизации $E(\vec{w})$ резко замедляется. Несмотря на указанные недостатки, простота и небольшие вычислительные затраты АНС сделали его одним из основных способов обучения многослойных ИНС. Повысить эффективность АНС удается путем модификации (как правило, эвристической) выражения (3.3).

Достаточно удачной разновидностью АНС является метод обучения с так называемым *моментом,* где приращение

$$\Delta\vec{w}_t = \eta_t \vec{p}_t + \alpha\left(\vec{w}_t - \vec{w}_{t-1}\right) \tag{3.4}$$

записывается с учетом коэффициента момента $\alpha \in [0,1]$. Первое слагаемое (3.4) соответствует обычному АНС, второе учитывает предыдущее изменение весов и не зависит от величины $\nabla E(\vec{w})$. Влияние $\alpha\left(\vec{w}_t - \vec{w}_{t-1}\right)$ резко возрастает на плоских участках $E(\vec{w})$, а также вблизи точек минимума, где значения градиента близки к нулю. Например, для плоских участков $E(\vec{w})$, где при постоянном $\eta_t \equiv \eta$ приращение весов $\Delta\vec{w}_t \simeq \text{const}$, можно записать $\Delta\vec{w}_t = \eta\vec{p}_t + \alpha\Delta\vec{w}_t$, откуда $\Delta\vec{w}_t = \dfrac{\eta}{1-\alpha}\vec{p}_t$, что при $\alpha = 0{,}9$ соответствует ускорению процесса обучения на порядок. Аналогично, вблизи локальных минимумов второе слагаемое (3.4) ввиду малости $\vec{p}_t$ начинает доминировать над первым, приводя к увеличению $E(\vec{w})$ и даже к уходу из окрестности данного локального минимума, что может быть использовано для целей глобальной оптимизации. Однако для предотвращения нестабильности алгоритма временные возрастания $E(\vec{w})$ не должны превышать (4–5) %.

3.2.1.2. Алгоритм переменной метрики (АПМ)

В основе АПМ лежит ньютоновский алгоритм оптимизации с использованием вторых производных оценки, то есть трех первых слагаемых

в разложении (3.2). Для достижения минимума $E(\vec{w})$ необходимо, чтобы $\frac{dE(\vec{w}_t + \vec{p}_t)}{d\vec{p}_t} \approx 0$, то есть дифференцированием (3.2) условие оптимальности можно получить в виде $\vec{g}(\vec{w}_t) + H(\vec{w}_t)\vec{p}_t = 0$ с очевидным решением

$$\vec{p}_t = -\left[H(\vec{w}_t)\right]^{-1}\vec{g}(\vec{w}_t), \tag{3.5}$$

однозначно указывающим направление, гарантирующее достижение на данном шаге минимальной $E(\vec{w})$.

Применение (3.5) требует положительной определенности $H(\vec{w})$ на каждом шаге, что в общем случае практически неосуществимо, поэтому в известных реализациях алгоритма, как правило, вместо точного $H(\vec{w})$ используется его приближение $G(\vec{w})$, при котором гессиан или обратная ему величина модифицируется на величину некоторой поправки, вычис–ляемой по формулам Бройдена–Флетчера–Гольдфарба–Шенно (BFGS – метод) или Дэвидона–Флетчера–Пауэлла (DFP – метод). Если обозначить $\vec{w}_t - \vec{w}_{t-1} \equiv s_t; \vec{g}(\vec{w}_t) - \vec{g}(\vec{w}_{t-1}) \equiv \vec{r}_t; \left[G(\vec{w}_t)\right]^{-1} \equiv V_t$, то процесс уточнения матрицы V можно описать рекуррентными зависимостями: для BFGS – метода –

$$V_t = V_{t-1} + \left[1 + \frac{\vec{r}_t^{\,T} V_{t-1} \vec{r}_t}{\vec{s}_t^{\,T}\vec{r}_t}\right]\frac{\vec{s}_t\vec{s}_t^{\,T}}{\vec{s}_t^{\,T}\vec{r}_t} - \frac{\vec{s}_t\vec{r}_t^{\,T}V_{t-1}\vec{r}_t\vec{s}_t^{\,T}}{\vec{s}_t^{\,T}\vec{r}_t}, \tag{3.6}$$

а для DFP – метода –

$$V_t = V_{t-1} + \frac{\vec{s}_t\vec{s}_t^{\,T}}{\vec{s}_t^{\,T}\vec{r}_t} - \frac{V_{t-1}\vec{r}_t\vec{r}_t^{\,T}V_{t-1}}{\vec{r}_t^{\,T}V_{t-1}\vec{r}_t}, \tag{3.7}$$

где в качестве начального значения обычно принимается $V_0 = 1$, а первая итерация выполняется в соответствии с АНС. Показано, что обеспечение с помощью (3.5), (3.6) положительной определенности гессиана на каждом шаге итерации действительно гарантирует решение проблемы оптимизации, причем метод BFGS менее чувствителен к различным погрешностям вычислительного процесса.

АПМ характеризуется более быстрой сходимостью, чем АНС, и именно он в настоящее время считается одним из наиболее эффективных методов оптимизации функций нескольких переменных, а следовательно, и обучения ИНС. Его недостаток – это большие вычислительные затраты, связанные с необходимостью расчета и хранения в памяти n^2 элементов гессиана в каждом цикле, что при оптимизации функции с большим количеством переменных может стать серьезной проблемой. По этой причине метод применяется для не очень больших НС, имеющих не более тысячи взвешенных связей.

3.2.1.3. Алгоритм Левенберга–Марквардта (АЛМ)

Как и АПМ, АЛМ относится к ньютоновским методам оптимизации с заменой $H(\vec{w})$ приближенным $G(\vec{w})$, рассчитываемым на основе имеющейся информации о $\vec{g}(\vec{w})$ с учетом некоторого фактора регуляризации. Обозначая

$$\vec{e}\left(\vec{w}\right) \equiv \begin{bmatrix} e_1\left(\vec{w}\right) \\ e_2\left(\vec{w}\right) \\ \vdots \\ e_p\left(\vec{w}\right) \end{bmatrix}; \quad J\left(w\right) \equiv \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_n} \\ \vdots & & \vdots \\ \frac{\partial e_p}{\partial w_1} & \cdots & \frac{\partial e_p}{\partial w_n} \end{bmatrix}, \tag{3.8}$$

где $e_i\left(\vec{w}\right) \equiv \left[y_i\left(\vec{w}\right) - d_i \right]$, вектор градиента $\vec{g}(\vec{w})$ и матрицу $G(\vec{w})$ можно представить в виде

$$\begin{aligned} \vec{g}\left(\vec{w}\right) &= \left[J\left(\vec{w}\right) \right]^T \vec{e}\left(\vec{w}\right); \\ G\left(\vec{w}\right) &= \left[J\left(\vec{w}\right) \right]^T J\left(\vec{w}\right) + R\left(\vec{w}\right), \end{aligned} \tag{3.9}$$

где $R(\vec{w})$ – компоненты $H(\vec{w})$ с высшими производными относительно $\vec{w}$, которые в АЛМ аппроксимируются с помощью скалярного параметра Левенберга–Марквардта υ, изменяющегося в процессе оптимизации таким образом, что

$$G\left(\vec{w}_t\right) = \left[J\left(\vec{w}_t\right) \right]^T J\left(\vec{w}_t\right) + \upsilon_t \cdot \mathbf{1}. \tag{3.10}$$

В начале обучения, когда значения $\vec{w}_t$ далеки от решения, используют $\upsilon_t >> \left\| \left[J\left(\vec{w}\right) \right]^T J\left(\vec{w}\right) \right\|$, то есть $G\left(\vec{w}_i\right) \approx \upsilon_t \cdot \mathbf{1}$ и $\vec{p}_t = -\dfrac{\vec{g}(\vec{w}_t)}{\upsilon_t}$, однако по мере уменьшения погрешности $e_i\left(\vec{w}\right)$ первое слагаемое в (3.10) начинает играть все более важную роль. Эффективность метода сильно зависит от выбора υ_t. Существуют различные способы подбора этого параметра, однако наиболее известна методика Д. Марквардта:

– если $E\left(\dfrac{\upsilon_{t-1}}{r}\right) \le E_t$, то $\upsilon_t = \dfrac{\upsilon_{t-1}}{r}$, где r>1 – коэффициент уменьшения υ;

– если $E\left(\dfrac{\upsilon_{t-1}}{r}\right) > E_t$, а $E\left(\upsilon_{t-1}\right) < E_t$, то $\upsilon_t = \upsilon_{t-1}$;

– если $E\left(\dfrac{\upsilon_{t-1}}{r}\right) > E_t$ и $E\left(\upsilon_{t-1}\right) > E_t$, то $\upsilon_t = \upsilon_{t-1} r^m$ до достижения $E\left(\upsilon_{t-1} r^m\right) \le E_t$.

Заметим, что в непосредственной близости к точке решения $\upsilon = 0$, процесс определения $G(\vec{w})$ сводится к аппроксимации 1-го порядка, а АЛМ превращается в алгоритм Гаусса–Ньютона, характеризующийся квадратичной сходимостью к оптимальному решению.

3.2.1.4. Алгоритм сопряженных градиентов (АСГ)

Этот метод не использует информацию о $H(\vec{w})$, а направление поиска $\vec{p}_t$ выбирается ортогональным и сопряженным всем предыдущим направлениям $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_{t-1}$. Показано, что этим условиям удовлетворяет

$$\vec{p}_t = -\vec{g}_t + \beta_{t-1}\vec{p}_{t-1}, \tag{3.11}$$

где коэффициент сопряжения β_{t-1} играет важную роль, аккумулируя информацию о предыдущих направлениях поиска. Наиболее известны следующие правила определения β_{t-1}

$$\beta_{t-1} = \frac{\vec{g}_t\left(\vec{g}_t - \vec{g}_{t-1}\right)}{\vec{g}_{t-1}^T\vec{g}_{t-1}}; \quad \beta_{t-1} = \frac{\vec{g}_t^T\left(\vec{g}_t - \vec{g}_{t-1}\right)}{-\vec{p}_{t-1}\vec{g}_{t-1}}. \tag{3.12}$$

Метод сопряженных градиентов имеет сходимость, близкую к линейной, он менее эффективен, чем АПМ, но заметно быстрее АНС. Благодаря невысоким требованиям к памяти и относительно низкой вычислительной сложности, АСГ широко применяется как единственно эффективный алгоритм оптимизации при значительном числе переменных (до нескольких десятков тысяч весов связей при обучении НС).

3.2.2. Эвристические методы обучения НС

Помимо алгоритмов обучения, использующих апробированные методы оптимизации нелинейной целевой функции, создано огромное количество алгоритмов эвристического типа, представляющих собой, в основном, модификацию АНС или АСГ. Подобные модификации связаны с внесением в них некоторых изменений, ускоряющих (по мнению авторов) процесс обучения ИНС. Как правило, эти методы не имеют серьезного теоретического обоснования, однако в них реализуется личный опыт работы авторов с нейронными сетями. К наиболее известным и эффективным эвристическим алгоритмам относятся:

– *алгоритм Quickprop Фальмана, с*одержащий элементы, предотвращающие зацикливание в точках неглубоких локальных минимумов. Изменение весов на шаге *t* алгоритма осуществляется согласно

$$\Delta w_{ij}(t) = -\eta_t\left[\frac{\partial E}{\partial w_{ij}} + \gamma w_{ij}(t)\right] + \alpha_{ij}(t)\Delta w_{ij}(t-1), \tag{3.13}$$

где первое слагаемое соответствует АНС, последнее – методу моментов, а средний член $\gamma w_{ij}(t)$ предназначен для минимизации ($\gamma \sim 10^{-4}$) абсолютных значений весов вплоть до возможного разрыва соответствующих связей (при $w_{ij} \approx 0$). Важную роль в алгоритме Quickprop играет фактор момента $\alpha_{ij}(t)$, который подбирается индивидуально для каждого веса w_{ij} и адаптируется к текущим результатам обучения;

– *алгоритм RPROP Ридмиллера–Брауна*, где при уточнении весов учитывается только знак градиентной составляющей, а ее значение отбрасывается, то есть

$$\Delta w_{ij}(t) = -\eta_{ij}(t)\, sign\left(\frac{\partial E}{\partial w_{ij}}\right). \qquad (3.14)$$

Коэффициент обучения $\eta_{ij}(t)$ также подбирается индивидуально для каждого w_{ij} с учетом изменения градиента на каждом шаге обучения. Предельные значения $\eta_{ij}(t)$ для алгоритма RPROP составляют $\eta_{\min} = 10^{-6}$ и $\eta_{\max} = 50$ соответственно. Заметим, что этот алгоритм позволяет значительно ускорить процесс обучения в тех случаях, когда угол наклона целевой функции невелик.

3.2.3. Подбор коэффициентов и сравнение эффективности детерминированных алгоритмов обучения НС

Все рассмотренные алгоритмы обучения НС связаны только с определением направления $\vec{p}_t$ на каждом шаге, но ничего не говорят о выборе коэффициента обучения $\eta(t)$, хотя он оказывает огромное влияние на скорость сходимости: слишком малое значение $\eta(t)$ не позволяет минимизировать $E(\vec{w})$ за один шаг в заданном направлении и требует повторных итераций, слишком большой шаг приводит к «перепрыгиванию» через минимум целевой функции и фактически заставляет возвращаться к нему. Существуют различные способы подбора η. Простейший из них основан на фиксации η = const на весь период оптимизации, практически используется только в АНС при обучении в режиме «online» и имеет низкую эффективность, поскольку никак не связан с величиной и направлением $\vec{p}_t$ на данной итерации. Обычно величина η подбирается отдельно для каждого слоя НС, чаще всего с использованием соотношения

$$\eta \le \min\left(\frac{1}{n_i}\right), \qquad (3.15)$$

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

Более эффективный способ – *адаптивный выбор* η с учетом динамики изменения $E(\vec{w})$ в процессе обучения, когда тенденция к непрерывному увеличению η сочетается с контролем суммарной погрешности $\varepsilon_t = \sqrt{\sum_{j=1}^{M}\left(y_j - d_j\right)^2}$ на каждой итерации. При этом

$$\begin{aligned} &\eta_{t+1} = k_y \eta_t, \quad \textit{при } \varepsilon_t > k_n \varepsilon_{t-1}; \\ &\eta_{t+1} = k_в \eta_t, \quad \textit{при } \varepsilon_t \le k_n \varepsilon_{t-1}, \end{aligned} \tag{3.16}$$

где k_y, $k_в$ – коэффициенты уменьшения и увеличения η_t соответственно, k_n – коэффициент допустимого прироста погрешности ε. Заметим, что реализация этой стратегии выбора η в NNT MATLAB 6,5 при k_n = 1,41, k_y = 0,7 , $k_в$ = 1,05 позволила в несколько раз ускорить обучение многослойных НС при решении задач аппроксимации нелинейных функций.

Наиболее эффективный, хотя и наиболее сложный, метод подбора η связан *с направленной минимизацией* $E(\vec{w})$ в заранее выбранном направлении $\vec{p}_t$, когда значение η_t подбирается так, чтобы новое решение $\vec{w}_{t+1} = \vec{w}_t + \eta_t \vec{p}_t$ соответствовало минимуму $E(\vec{w})$ в направлении $\vec{p}_t$. Чаще всего определение оптимальной величины η связано с представлением $E(\vec{w})$ полиномом 2 или 3-го порядка от η

$$\begin{aligned} &E\left(\vec{w}\right) \Rightarrow P_2\left(\eta\right) = a_2\eta^2 + a_1\eta + a_0; \\ &E\left(\vec{w}\right) \Rightarrow P_3\left(\eta\right) = a_3\eta^3 + a_2\eta^2 + a_1\eta + a_0, \end{aligned} \tag{3.17}$$

где для определения коэффициентов a_i используют информацию о величине $E(\vec{w})$ и ее производной в направлении $\vec{p}_t$, а значения $\eta_{опт}$ получают из условия минимума $P_2(\eta)$ или $P_3(\eta)$ согласно $\eta_{опт} = -\dfrac{a_1}{2a_2}$ для $P_2(\eta)$ или $\eta_{опт} = \dfrac{-a_2 + \sqrt{a_2^2 - 3a_2a_1}}{3a_3}$ для $P_3(\eta)$.

Эффективность алгоритмов обучения проверяется на стандартных тестах, к которым относятся задачи логистики (предсказания последующего значения x_{n+1} случайной последовательности по предыдущему значению x_n), кодирования и декодирования двоичных данных, аппроксимации нелинейных функций определенного вида, комбинаторной оптимизации («задача коммивояжера») и т. п. Сравнение идет по количеству циклов обучения, количеству расчетов $E(\vec{w})$, чувствительности к локальным минимумам и т. д. Поскольку эти характеристики могут существенно отличаться в зависимости от характера

тестовой задачи, то однозначный ответ на вопрос, какой алгоритм считать абсолютно лучшим, дать невозможно.

В качестве возможного примера сравнения эффективности рассмотренных методов обучения в таблице 3.1 представлены результаты обучения многослойного персептрона со структурой 1–10–1, предназначенного для аппроксимации одномерной функции на основе обучающей выборки из 41 элемента. Все алгоритмы обучения были реализованы в пакете дополнений NNT MATLAB, что послужило основой для получения объективных оценок. Видно, что наибольшую эффективность продемонстрировал АЛМ, за ним идут АПМ (BFGS) и АСГ. Наихудшие результаты (по всем параметрам) показал АНС, а эвристический алгоритм RPROP в этом примере был сравним с АПМ и АСГ. Заметим, однако, что на основании более общих тестов был сделан вывод, что доминирующая роль АЛМ и АПМ снижается по мере увеличения размеров НС, и при числе связей больше 10^3 наиболее эффективным становится АСГ.

Таблица 3.1

Обучение многослойного персептрона

Алгоритм	Время, с	Кол-во циклов	Кол-во операций, ($\times 10^{-6}$)
АНС с адаптируемым η	57,7	980	2,50
Сопряженных градиентов	19,2	89	0,75
АПМ типа BFGS	10,9	44	1,02
Левенберга–Марквардта	1,9	6	0,46
RPROP	13,0	185	0,56

3.2.4. Методы глобальной оптимизации

При обучении НС с нелинейными функциями активации даже при решении относительно простых технических задач необходимо учитывать возможность появления большого количества локальных минимумов целевой функции. Например, если для одного нейрона с входным весом w_1 и весом поляризатора w_0 при линейной функции активации график зависимости $E(\vec{w})$ от w_0, w_1 имеет вид выпуклой поверхности, единственный минимум которой легко определить при любых начальных условиях, то при использовании в качестве функции активации гиперболического тангенса форма $E(\vec{w})$ принципиально меняется, изобилуя плоскими участками и множеством локальных минимумов. Увеличение размеров НС только осложняет проблему, поскольку число минимумов также возрастает, каждый из которых представляет собой ловушку на пути к глобальному минимуму, в котором $E(\vec{w})$ принимает наименьшее значение.

Все рассмотренные до сих пор детерминированные методы обучения являются локальными, поскольку ведут к одному из локальных минимумов $E(\vec{w})$, лежащему в окрестности точки начала обучения. При этом оценить оптимальность найденного решения можно лишь в тех случаях, когда значение глобального минимума известно. Если локальное решение считается неудов-

летворительным, то процесс обучения можно повторить, используя новые (как правило, случайные) начальные значения $\vec{w}_0$ или изменяя случайным образом найденное ранее решение («встряхивание» весов). Подобный прием – применение стохастических алгоритмов к детерминированным методам обучения – связан с определенной вероятностью того, что новый поиск будет покидать «зоны притяжения» найденных ранее локальных минимумов. При решении реальных задач даже приблизительная оценка глобального минимума оказывается неизвестной, что требует, в общем, применения методов *глобальной оптимизации*, среди которых наиболее разработаны метод имитации отжига, генетические алгоритмы и метод виртуальных частиц.

3.2.4.1. Метод имитации отжига (ИО)

Предложенный Н. Метрополисом в 1953 году, метод ИО представляет собой алгоритмический аналог физического процесса управляемого охлаждения, при котором кристаллизация расплава сопровождается глобальным уменьшением его энергии, причем допускаются ситуации кратковременного повышения энергетического уровня (например, при небольшом подогреве), способствующие выводу из ловушек локальных минимумов энергии, возникающих при реализации процесса. Классический алгоритм ИО можно представить следующим образом:

1. Определяем некоторую начальную переменную («температуру») $T = T_{max}$ и запускаем процесс обучения НС из некоторой начальной точки $\vec{w}_0$.

2. Пока $T > 0$, повторяем L раз следующие шаги:

– выбираем новое решение $\vec{w}_t$ из окрестности $\vec{w}_0$;

– рассчитываем изменение целевой функции $\Delta = E(\vec{w}_t) - E(\vec{w}_0)$;

– если $\Delta \le 0$, принимаем $\vec{w}_{t+1} = \vec{w}_t$;

– если $\Delta > 0$, то вычисляем вероятность $P = \exp\left(-\Delta / T\right)$, выбираем случайное число $R \in (0,1)$ и, если $R \le P$, то $\vec{w}_{t+1} = \vec{w}_t$, в противном случае ($R>P$) $\vec{w}_{t+1} = \vec{w}_0$.

3. Уменьшаем температуру ($T_k = rT_{k-1}$) с использованием коэффициента уменьшения $r \in (0,1)$ и повторяем п. 2.

4. При снижении T до нуля обучаем НС любым из представленных выше детерминированных методов.

Эффективность метода ИО сильно зависит от выбора T_{max}, L и r. Величина T_{max} определяется из предварительных имитационных экспериментов таким образом, чтобы обеспечить реализацию не менее 50 % последующих случайных изменений решения. Выбор максимальных L и r для конкретных температурных уровней менее однозначен и должен учитывать динамику изменения $E(\vec{w})$ в зависимости от количества выполненных циклов обучения. Общие рекомендации, вытекающие из компьютерных экспериментов, таковы: если время обучения ограничено, его лучше потратить на один процесс

ИО с соответствующим удлинением циклов, если же моделирование может быть более длительным, статистически лучшие результаты достигаются при многократной реализации процесса ИО с большими (близкими к 1) значениями *r* и последующим выбором оптимального решения.

Таким образом, метод ИО наиболее эффективен для полимодальных комбинаторных проблем с большим числом возможных решений, например, для машины Больцмана, в которой каждое состояние системы (с различной вероятностью) считается допустимым. В общем же случае при решении наиболее распространенных задач обучения многослойных НС наилучшие результаты достигаются применением стохастических методов совместно с детерминированными алгоритмами локальной оптимизации.

3.2.4.2. Генетические алгоритмы (ГА)

Генетические алгоритмы, первоначально предложенные Дж. Холландом и использованные Д. Гольдбергом для численных оптимизационных расчетов в 70-х годах прошлого века, имитируют процессы наследования свойств живыми организмами и генерируют последовательности новых векторов $\vec{w}$, содержащие оптимизированные переменные $\vec{w} = \left[w_1, w_2, \ldots, w_n \right]^T$. При этом выполняются операции трех видов: *селекция, скрещивание* и *мутация.*

На исходной стадии ГА случайным образом инициализируется определенная популяция хромосом (векторов $\vec{w}$). Размер популяции постоянен и обычно пропорционален количеству оптимизируемых параметров, поскольку слишком малая или слишком большая популяции приводят либо к замыканию в локальных минимумах, либо чрезмерно увеличивают вычислительные затраты без гарантии достижения глобального минимума.

Селекция (отбор) хромосом для создания нового поколения может производится разными способами, однако самым распространенным считается *принцип элитарности*, при котором наиболее приспособленные (в смысле $E(\vec{w})$) хромосомы сохраняются, а наихудшие отбрасываются и заменяются вновь созданным потомством, полученным в результате скрещивания пар родителей.

Количество методов *скрещивания* достаточно велико, от полностью случайного до турнирного. Чисто случайное спаривание осуществляется среди наиболее приспособленных хромосом, взвешенно-случайные методы используют информацию о текущем значении $E(\vec{w})$, например, при отборе по *принципу рулетки* вероятность скрещивания конкретной хромосомы пропорциональна величине ее *функции приспособленности* $F(\vec{w}) = -E(\vec{w})$. Процесс скрещивания основан на рассечении пары хромосом на 2 части с последующим обменом этих частей в хромосомах родителей (рис. 3.2). Место рассечения выбирается случайным образом, количество новых потомков равно количеству отбракованных в результате селекции, допуска-

ется перенос в очередное поколение некоторых хромосом (из числа хорошо приспособленных) вообще без скрещивания.

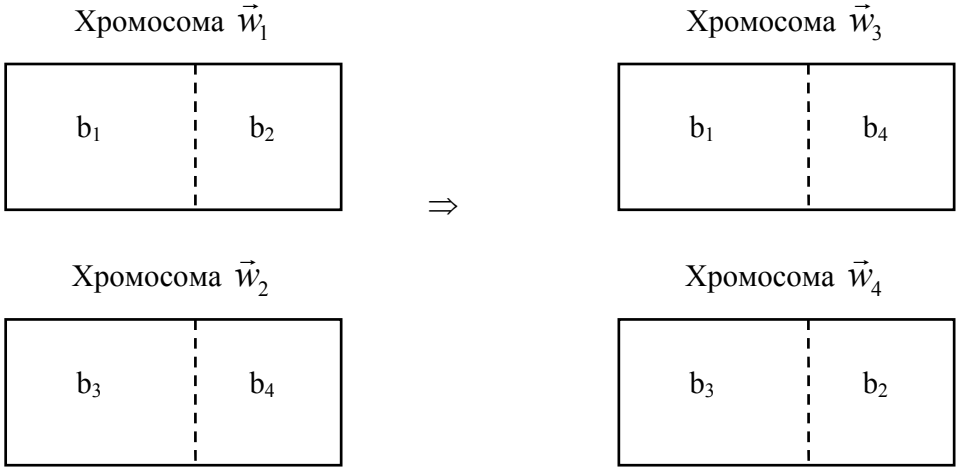

Рис. 3.2. Вариант операции скрещивания, применяемой в ГА

Последняя генетическая операция – *мутация* – обеспечивает защиту от слишком быстрого завершения алгоритма в точке, далекой от глобального экстремума. При двоичном кодировании $\vec{w}$ мутация состоит в инверсии случайно выбранных битов, при использовании десятичных цифр – в замене некоторых компонентов $\vec{w}$ случайно выбранными значениями. Необходимо помнить, что случайные мутации приводят к повреждению уже частично приспособленных векторов, поэтому обычно мутации подвергается не более (1…5) % элементов $\vec{w}$ всей популяции хромосом.

Доказано, что каждое последующее поколение, сформированное селекцией, скрещиванием и мутацией, имеет статистически лучшие средние показатели приспособленности (меньшие значения $E(\vec{w})$). В качестве конечного решения принимается наиболее приспособленная хромосома $\vec{w}$, имеющая минимальное значение $E(\vec{w})$. Генетический процесс завершается либо при достижении приемлемого решения, либо по превышении максимально допустимого количества итераций. При реализации ГА отслеживается не только минимальное значение $E(\vec{w})$, но и ее среднее значение по всей популяции хромосом, так что решение об остановке алгоритма может приниматься и в случае отсутствия прогресса минимизации указанных характеристик.

3.2.4.3. Метод виртуальных частиц (ВЧ)

Метод виртуальных (случайных) частиц может использоваться практически с любым методом оптимизации для повышения устойчивости обученных НС и вывода НС из локальных минимумов $E(\vec{w})$. Основная идея метода – усреднение значений $E(\vec{w})$ для случайных сдвигов аргумента с целью уменьшения влияния рельефа функции $E(\vec{w})$ на процесс ее минимизации. Реализация метода ВЧ состоит в том, что к оптимизируемой точке

(частице) $\vec{w}^0$ добавляется несколько других $\vec{w}_i$, траектории которых получаются из траектории исходной частицы сдвигом на случайные векторы $\vec{r}_i$, то есть $\vec{w}_i = \vec{w}^0 + \vec{r}_i$, где координаты $\vec{r}_i$ независимо и равномерно распределены в заданных интервалах случайных сдвигов. Далее минимизируется функция $E(\vec{w}) = E(\vec{w}^0) + E(\vec{w}^0 + \vec{r}_1) + \cdots + E(\vec{w}^0 + \vec{r}_n)$ с помощью любого метода локальной оптимизации.

Поскольку «виртуальные» частицы время от времени уничтожаются и рождаются новые, естественно возникает вопрос: когда это делать? Наиболее перспективен подход, в котором порождение новых частиц производится при рестартах после каждого цикла основного алгоритма обучения, поскольку в этом случае не разрушается базовая структура обучения, а многократное порождение виртуальных частиц позволяет приблизиться к глобальному оптимуму.

3.3. Проблемы практической реализации ИНС

Для решения какой-либо задачи с применением ИНС нужно, прежде всего, создать структуру сети, адекватную поставленной задаче (аналогично составлению соответствующей программы для универсальной ЭВМ). Это предполагает выбор количества слоев НС и числа нейронов в каждом слое, а также определение необходимых связей между ними.

3.3.1. Выбор оптимальной архитектуры

Как уже упоминалось (п. 3.1, теорема КАХН), количество нейронов входного слоя НС определяется размерностью N входного вектора $\vec{x} = [x_1, x_2, \ldots, x_N]^T$, количество нейронов выходного слоя – размерностью M ожидаемого вектора $\vec{d} = [d_1, d_2, \ldots, d_M]^T$. Определение минимального числа скрытых слоев основано на использовании свойств аппроксимирующих функций. Для непрерывного преобразования X→Y (см. теорему КАХН) достаточно иметь один скрытый слой с $K = (2N + 1)$ нейронами, в случае дискретного преобразования необходимое число скрытых слоев возрастает до двух [1]. В практических реализациях ИНС как количество скрытых слоев, так и число нейронов в них могут отличаться от теоретически предлагаемых. За немногими исключениями, чаще всего используются НС, имеющие один (максимум – 2) скрытый слой, в котором $K = N \ldots 3N$.

Определение оптимального числа K основано на способности ИНС к обобщению полученных знаний, то есть выдаче правильных результатов при подаче на ее вход данных, не входящих непосредственно в обучающую выборку. Пример разделения множества данных, подчиняющихся правилу R, на обучающее L, контрольное V и тестовое G подмножества приведен на рисунке 3.3. Элементы L и G должны быть типичными элементами множества R. Способность отображения сетью эле-

ментов L является показателем ее обученности и характеризуется погрешностью обучения $E_L(\vec{w})$, способность распознавания данных подмножества G показывает ее возможности обобщения знаний и описывается погрешностью обобщения $E_G(\vec{w})$. Для верификации качества обучения НС в составе L выделяется определенное подмножество контрольных данных V.

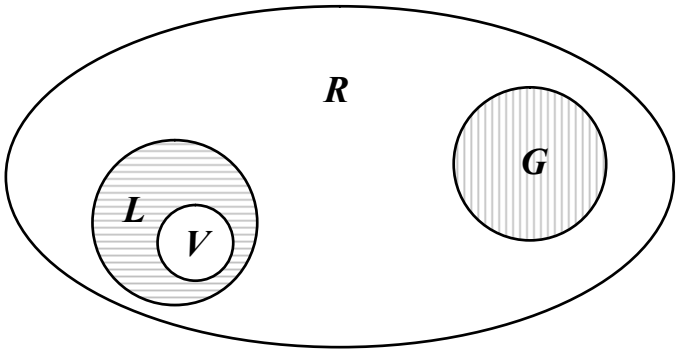

Рис. 3.3. Разделение множества данных R на обучающее L, контрольное V и тестовое G подмножества

При обучении НС оказывается, что количество весов T_w сети (число степеней свободы) и число обучающих выборок p тесно связаны. Например, если бы целью обучения НС было только запоминание $\vec{x}^{(k)}$, тогда достаточно было бы $p = T_w$, однако такая сеть не будет обладать свойством обобщения и сможет только восстанавливать данные. Для обретения обобщающих свойств НС необходимо выполнение $p > T_w$, чтобы веса сети адаптировались не к уникальным выборкам, а к их статистически усредненным совокупностям. Наглядная графическая иллюстрация способности НС к обобщению показана на рисунке 3.4 на примере аппроксимации одномерной функции двухслойной НС. Видно, как при избыточном числе нейронов и весов проявляется эффект гиперразмерности НС, когда минимизация $E_L(\vec{w})$ на относительно малом числе обучающих выборок спровоцировала случайный характер значений многих весов, что при переходе к тестовым сигналам обусловило значительное отклонение фактических значений y_i от ожидаемых d_i (рис. 3.4, а). Уменьшение количества скрытых нейронов до оптимального значения (рис. 3.4, б) обеспечило и малую погрешность обучения, и высокую степень обобщения (малую $E_G(\vec{w})$). Дальнейшее уменьшение K привело к потере НС способности восстанавливать обучающие данные (рис. 3.4, в).

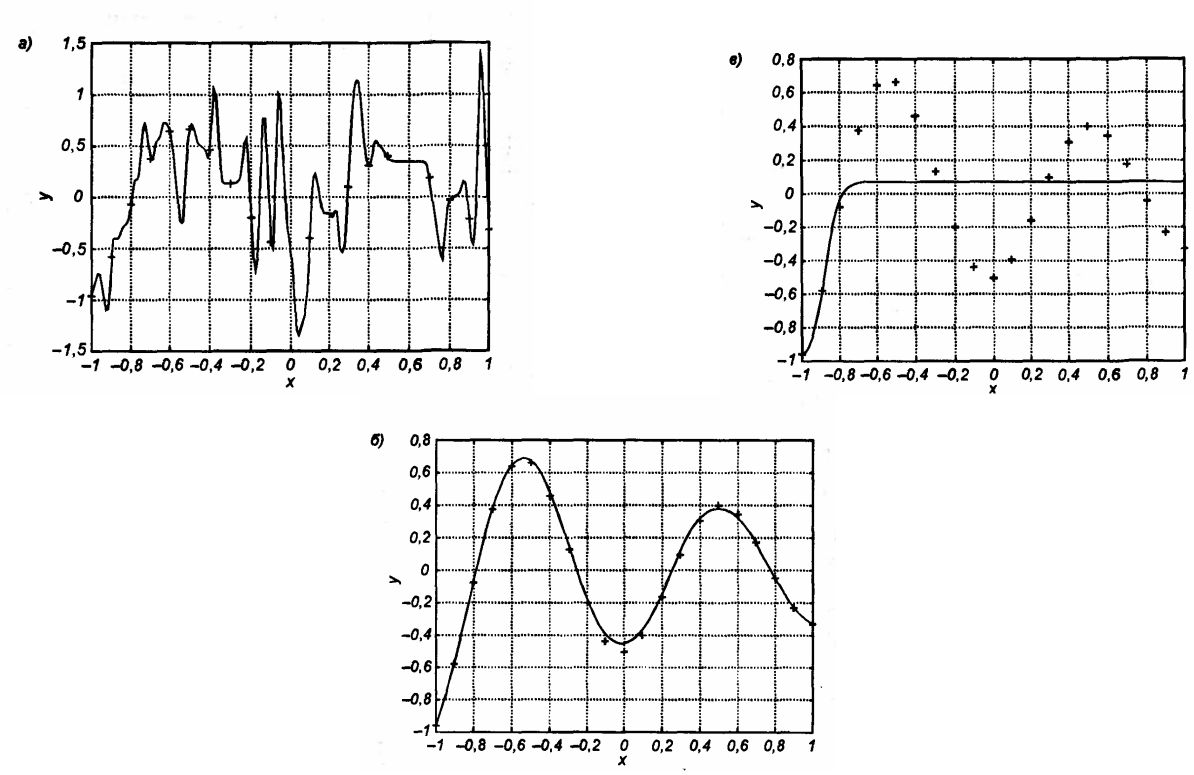

Рис. 3.4. Аппроксимация одномерной функции y = f(x), заданной в 21 точке, двухслойной НС, содержащей К нейронов скрытого слоя:
а – К = 80; б – К = 5; в – К = 1

Следует отметить, что длительность обучения по-разному влияет на значения $E_L(\vec{w})$ и $E_G(\vec{w})$. Если погрешность $E_L(\vec{w})$ монотонно уменьшается с увеличением числа итераций *t*, то снижение $E_G(\vec{w})$ происходит только до определенного момента, после чего она начинает расти (рис. 3.5).

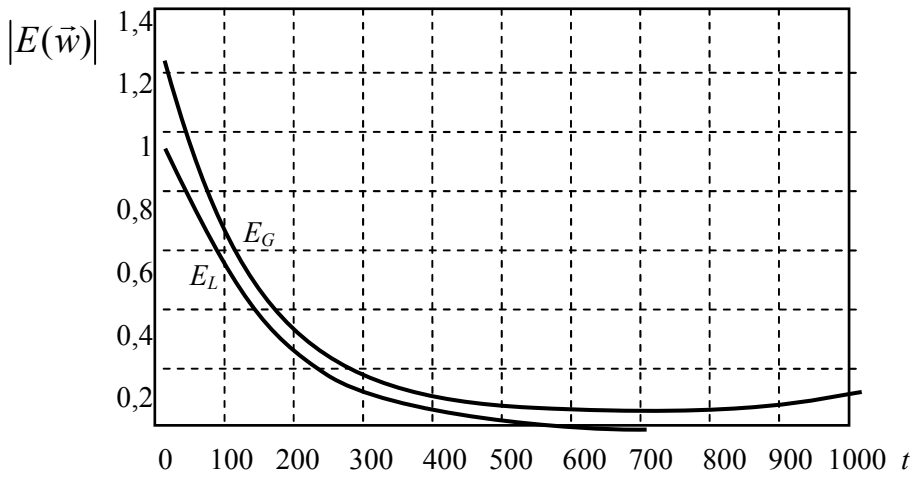

Рис. 3.5. Влияние длительности обучения на погрешности $E_L(\vec{w})$

Это означает, что слишком долгое обучение может привести к «переобучению» НС, которое выражается в слишком детальной адаптации весов к

несущественным флуктуациям обучающих данных. Такая ситуация особенно заметна при использовании сети с излишним количеством весов. Для предотвращения перетренированности НС служит контрольное подмножество V, с помощью которого осуществляется оперативная проверка фактически набранного уровня обобщения $E_G(\vec{w})$.

3.3.2. Методы редукции и наращивания НС

Редукция ИНС производится для сокращения количества скрытых нейронов и межнейронных связей, что усиливает способность сети к обобщению. Большинство методов сокращения размерности НС можно разделить на три группы:

1. *Редукция НС с учетом величины весов* предполагает отсечение весов, значительно меньших средних значений, поскольку они оказывают небольшое влияние на уровень выходного сигнала связанных с ними нейронов. Однако это не всегда справедливо и может быть связано, например, с неудачным выбором $\vec{w}_0$, то есть в случае отсечения могут произойти значительные изменения в функционировании НС. Поэтому этим методом целесообразно пользоваться в редких и простейших случаях (например, отсечение одного или нескольких весов).

2. *Редукция НС с учетом чувствительности* основана на разложении $E(\vec{w})$ в ряд Тейлора (3.2) с использованием в качестве показателя важности конкретных весов вторых производных целевой функции. Одним из лучших методов регуляризации НС считается метод OBD (Optimal Brain Damage), в котором для упрощения задачи автор [Ле Кун] исходит из положительной определенности гессиана $H(\vec{w})$, когда в качестве меры значимости используется коэффициент асимметрии

$$S_{ij} = \frac{1}{2}\frac{\partial^2 E}{\partial w_{ij}^2} w_{ij}^2, \tag{3.18}$$

содержащий только диагональные элементы $H(\vec{w})$. Алгоритм OBD выглядит следующим образом:

– полное предварительное обучение НС (любым способом);

– определение элементов S_{ij};

– сортировка w_{ij} в порядке убывания S_{ij} и отсечение наименее значимых (с минимальными S_{ij}) w_{ij};

– возврат к началу и повторение процедуры с редуцированной НС.

Развитием метода OBD считается метод OBS (Optimal Brain Surgeon), предложенный Б. Хассиби и Д. Шторком, где величина

$$S_i = \frac{1}{2}\frac{w_i^2}{\left[H^{-1}\right]_{ii}} \tag{3.19}$$

определяется всеми компонентами гессиана, а после отсечения веса с минимальной S_i уточнение оставшихся происходит согласно

$$\Delta w = \frac{w_i}{\left[H^{-1}\right]_{ii}} H^{-1} \cdot \vec{1}_i, \tag{3.20}$$

где $\vec{1}_i = \left[0, 0, \ldots, 1, \ldots 0\right]^T$ – вектор с единичной компонентой в *i*-й позиции. Коррекция производится после отсечения каждого очередного веса и заменяет повторное обучение НС.

3. *Редукция НС с использованием штрафной функции* состоит в такой организации обучения, которая провоцирует самостоятельное уменьшение значений весов с исключением тех, величина которых опускается ниже определенного порога. Для этого целевая функция $E(\vec{w})$ модифицируется таким образом, чтобы в процессе обучения значения w_{ij} минимизировались автоматически вплоть до некоторого порога, после достижения которого они приравниваются к нулю. В простейшем варианте

$$E\left(\vec{w}\right) = E^{(0)}\left(\vec{w}\right) + \gamma \sum_{i,j} w_{ij}^2, \tag{3.21}$$

где $E^{(0)}\left(\vec{w}\right)$ – стандартная целевая функция, γ – коэффициент штрафа. Каждый цикл обучения складывается из двух этапов: минимизации $E^{(0)}\left(\vec{w}\right)$ любым стандартным методом и коррекции значений весов согласно формуле

$$w_{ij} = w_{ij}^{(0)}\left(1 - \eta\gamma\right), \tag{3.22}$$

где $w_{ij}^{(0)}$ – значения весов после первого этапа, η – коэффициент обучения. Следует отметить, что при такой функции штрафа происходит уменьшение всех весов и выбор порога отсечения должен производится весьма осторожно.

Более приемлемые результаты получаются при модификации $E(\vec{w})$ в виде

$$E\left(\vec{w}\right) = E^{(0)}\left(\vec{w}\right) + \frac{1}{2}\gamma \sum_{i,j} \frac{w_{ij}^2}{\left(1 + \sum_k w_{ik}^2\right)}, \tag{3.23}$$

когда осуществляется не только редукция межнейронных связей, но и исключаются те нейроны, для которых $\sum_k \left|w_{ik}\right| \approx 0$. Правило коррекции весов в этом случае выглядит следующим образом

$$w_{ij} = w_{ij}^{(0)} \left(1 - \eta\gamma \frac{1 + 2\sum_{k \neq i} \left(w_{ik}^{(0)} \right)^2}{\left[1 + \sum_{k} \left(w_{ik}^{(0)} \right)^2 \right]^2} \right). \qquad (3.24)$$

При малых w_{ik}, подходящих к i-му нейрону, происходит дальнейшее их уменьшение, при больших – коррекционная составляющая невелика и слабо влияет на процесс редукции сети.

Еще один способ минимизации НС основан на модификации $E(\vec{w})$, позволяющей исключить в процессе обучения скрытые нейроны с наименьшей активностью, то есть предполагается, что, если при любых обучающих выборках выходной сигнал какого-либо нейрона остается неизменным, то его присутствие в сети излишне. Целевая функция в этом случае записывается как

$$E\left(\vec{w}\right) = E^{(0)}\left(\vec{w}\right) + \mu \sum_{i=1}^{K} \sum_{j=1}^{p} e\left(\Delta_{ij}^2\right), \qquad (3.25)$$

где $e\left(\Delta_{ij}^2\right)$ – корректирующий фактор, зависящий от активности всех K скрытых нейронов для всех p обучающих выборок, Δ_{ij} – изменение значения i-го нейрона для j-й обучающей пары, μ – коэффициент коррекции. Вид $e\left(\Delta^2\right)$ подбирается так, чтобы при высокой активности скрытого нейрона величина ΔE была малой, при низкой активности – большой. Один из вариантов реализации

$$e\left(\Delta_{ij}^2\right) = \left[1 + \Delta_i^2\right]^{-1}. \qquad (3.26)$$

Следует отметить, все методы редукции НС ведут к улучшению их обобщающих свойств, причем в целом методы с использованием штрафных функций несколько уступают методам с учетом чувствительности.

В алгоритмах редукции в качестве исходной точки используется избыточная архитектура НС. Противоположный подход заключается в первоначальном включении в НС небольшого числа скрытых нейронов (часто они вообще отсутствуют), а по мере развития процесса обучения их число постепенно возрастает. Большинство известных *методов наращивания* НС имеют относительно низкую эффективность при большой размерности $\vec{x}$ и не составляют серьезной конкуренции методам редукции. Наиболее известным методом расширения является алгоритм каскадной корреляции Фальмана, но слоистая структура получаемой НС весьма специфична и не является полносвязной, так что ее реализация будет рассмотрена позднее при анализе специализированных структур НС.

3.3.3. Методы инициализации весов и подбора обучающих данных

Обучение НС, даже при использовании самых эффективных алгоритмов, – достаточно трудоемкий процесс, зависящий от многих факторов. Один из них – выбор начальных значений весов сети. Идеальными считаются начальные значения w_{ij}, достаточно близкие к оптимальны, когда не только устраняются задержки в точках локальных минимумов, но и значительно ускоряется процесс обучения. Универсального метода выбора $\vec{w}_0$ нет, поэтому в большинстве практических реализаций используется случайная инициализация $\vec{w}_0$ с равномерным распределением значений w_{ij} в заданном интервале. Чтобы стартовая точка активации нейронов лежала достаточно далеко от зоны насыщения, в качестве такого интервала чаще всего выбирают (0,1). Хорошие результаты дает равномерное распределение весов, нормализованное для каждого нейрона по амплитуде $w_{in} = 2\left[N_{in}\right]^{-1/2}$, где N_{in} – количество входов нейрона. Веса порогов скрытых нейронов должны принимать случайные значения из интервала ($-1/w_{in}$, $1/w_{in}$), а выходных нейронов – нулевые значения.

Достаточно серьезным фактором, влияющим на качество обучения НС, является подбор обучающих данных. С точки зрения цели функциони-

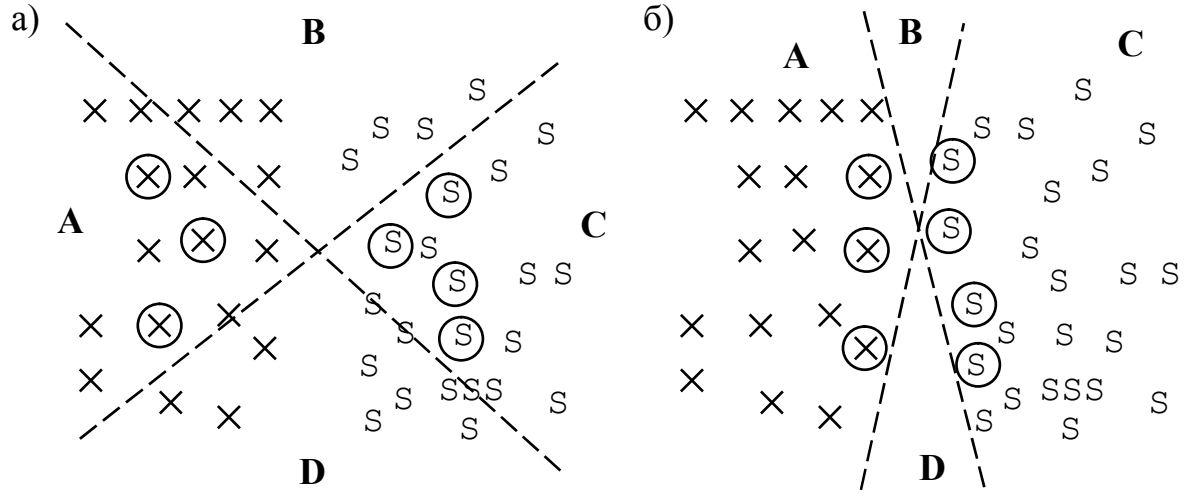

Рис. 3.6. Примеры выбора обучающих данных:
а – некорректный выбор; б – оптимальный выбор

рования НС можно рассматривать как векторный классификатор, определяющий принадлежность каждого входного вектора $\vec{x}$ к конкретной группе. Нейроны первого скрытого слоя образуют гиперплоскости, разделяющие N-мерное пространство входных данных на *кластеры,* нейроны следующего (чаще выходного) слоя идентифицируют кластеры. При ограниченном числе обучающих выборок их размещение относительно конкретных гиперплоскостей становится очень важным. Наилучшие результаты

достигаются в тех случаях, когда они располагаются с разных сторон границ гиперплоскостей, разделяющих пространство данных. Примеры выбора обучающих точек для разделения универсального множества показаны на рисунке 3.6. Неудачный выбор обучающих данных потребует использования 3-х скрытых нейронов (рис. 3.6, а), при корректном выборе будет достаточно одного (рис. 3.6, б), поскольку подмножества B и D оказываются пустыми. Видно, что оптимальному случаю соответствует выбор обучающих данных по границам областей, поэтому весьма важной является любая предварительная информация о количестве областей, по которым распределены эти данные.

3.3.4. Обеспечение устойчивости функционирования НС

После определения оптимальной архитектуры ИНС, выбора начальных значений параметров, подготовки обучающих данных и хорошего обучения актуальной становится задача обеспечения стабильности выходных сигналов, то есть устойчивости функционирования НС. Разработчики нейрокомпьютеров (НК) выделяют четыре типа устойчивости:

1) к случайным возмущениям входных сигналов;
2) к флуктуациям параметров сети;
3) к разрушению части элементов НС;
4) к обучению новым примерам.

Для выработки устойчивости первых трех типов целесообразно использовать генераторы случайных искажений, которые для устойчивости 1-го типа производят возмущение входных сигналов (преобразуют обучающий пример), для устойчивости 2-го типа – случайным образом меняют параметры сети в заданных пределах, а для устойчивости 3-го типа – удаляют случайно выбранную часть НС, состоящую из заданного количества элементов (нейронов, связей).

Средствами обучения устойчивости 4-го типа, вообще говоря, являются выработка устойчивости либо 1-го, когда возмущение состоит в изменении процесса обучения, либо 2-го типа, когда определяющую роль играет случайный сдвиг параметров. Опыт показывает, что обучение позволяет выработать устойчивость к весьма сильным возмущениям. Так, в задачах распознавания образов уровень шума мог в несколько раз превосходить полезный сигнал, случайный сдвиг параметров – достигать 0,5–0,7 их идеального значения, разрушение – 30–50 % элементов. И, тем не менее, обученная сеть делала не более 10 % ошибок!

Контрольные вопросы

1. Дайте определение и приведите классификацию НС.
2. Приведите теорему Колмогорова–Арнольда–Хехт–Нильсена и следствия из нее.

3. Какие критерии используются для сравнения методов обучения НС?

4. На чем основано и в чем заключается обучение ИНС?

5. Что лежит в основе градиентных методов обучения?

6. Опишите универсальный оптимизационный алгоритм обучения НС.

7. Дайте характеристику градиентных методов 1-го порядка.

8. В чем заключается основная идея оптимизации с помощью АПМ?

9. На чем основано обучение ИНС с помощью АЛМ?

10. Какова основная особенность оптимизации функций в методе АСГ?

11. Охарактеризуйте эвристические методы обучения НС.

12. Как осуществляется и на что влияет подбор коэффициентов обучения в детерминированных алгоритмах оптимизации?

13. Что показывает сравнение детерминированных методов обучения НС?

14. Почему при обучении ИНС наибольшую эффективность обеспечивают глобальные методы оптимизации?

15. Поясните основные этапы метода имитации отжига и его зависимость от выбора параметров T, L, r.

16. Охарактеризуйте основные операции ГА и методы их реализации.

17. Как осуществляется глобальная оптимизация в методе виртуальных частиц?

18. Определите основные этапы практического построения и обучения НС.

19. Чем определяется и как выбирается оптимальная архитектура НС?

20. На какие подмножества разбивается область входных данных при решении задач с помощью НС?

21. В чем заключается и как проявляется эффект гиперразмерности НС?

22. Что такое погрешность обучения и погрешность обобщения? Как они изменяются при обучении НС?

23. Какие алгоритмы сокращения НС Вы знаете? В чем их отличие?

24. Как осуществляется редукция НС с учетом чувствительности $E(w)$ к весу w_{ij}?

25. В чем заключается использование штрафных функций при редукции НС?

26. Как осуществляется начальная инициализация весов при обучении НС? На что она влияет?

27. Каким образом сказывается на эффективности обучения НС выбор обучающих данных?

28. Расскажите о методах обеспечения стабильности функционирования ИНС.

4. МНОГОСЛОЙНЫЕ НС ПРЯМОГО РАСПРОСТРАНЕНИЯ

В зависимости от способа объединения нейронов выделяют два основных типа НС – сети с прямым распространением сигнала и рекуррентные НС (с обратными связями). Среди НС прямого распространения наиболее известны многослойные структуры с прямыми полными связями между отдельными слоями. Передача сигнала в таких НС происходит только от входа к выходу, их математическое описание и методы обучения (как правило, с учителем) достаточно просты и практически несложны.

Исторически первой НС была однослойная сеть (рис. 4.1), состоящая из персептронов или сигмоидальных нейронов, каждый из которых реализует функциональное отображение (2.1). Обучение сети состоит в подборе весов w_{ij} в процессе минимизации целевой функции

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{p}\left\|\vec{y}^{(i)} - \vec{d}^{(i)}\right\|^2 = \frac{1}{2}\sum_{i=1}^{p}\sum_{k=1}^{M}\left[y_k^{(i)} - d_k^{(i)}\right]^2 \qquad (4.1)$$

и является точной копией обучения одиночного нейрона. Расположенные на одном уровне нейроны НС (рис. 4.1) функционируют независимо друг от друга, поэтому возможности такой сети определяются свойствами отдельных нейронов. Показано, например, что однослойный персептрон не в состоянии реализовать даже такую несложную функцию, как «Исключительное ИЛИ» (XOR), поэтому его применимость ограничена классом задач линейной сепарации. Однако положение кардинально меняется при добавлении еще хотя бы одного слоя.

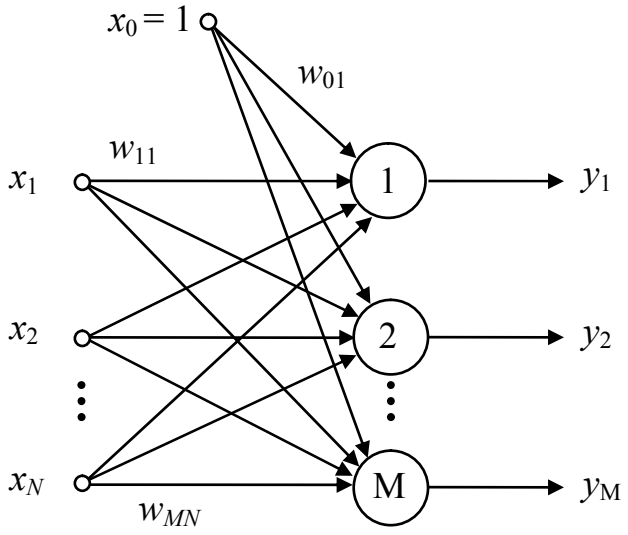

Рис. 4.1. Схема однослойного персептрона

4.1. Многослойный персептрон (МСП)

Многослойная сеть состоит из нейронов, расположенных на разных уровнях, когда, помимо входного и выходного слоев, имеется еще, как минимум, один промежуточный (скрытый) слой. Обобщенная структура двухслойной НС приведена на рис. 4.2. Выходной сигнал *i*–го нейрона скрытого слоя можно записать как

$$u_i = f\left[\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right], \quad i = 1,\ 2,\ \ldots K, \tag{4.2}$$

а выходные сигналы

$$y_l = f\left[\sum_{i=0}^{K} w_{li}^{(2)} u_i\right] = f\left[\sum_{i=0}^{K} w_{li}^{(2)} f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right)\right], \quad l = 1,\ 2,\ \ldots M, \tag{4.3}$$

где $f(\ldots)$, как правило, сигмоидальная функция активации. Из выражений (4.2), (4.3) следует, что на значение выходного сигнала влияют веса обоих слоев, тогда как сигналы от скрытого слоя не зависят от выходных весов $w_{li}^{(2)}$.

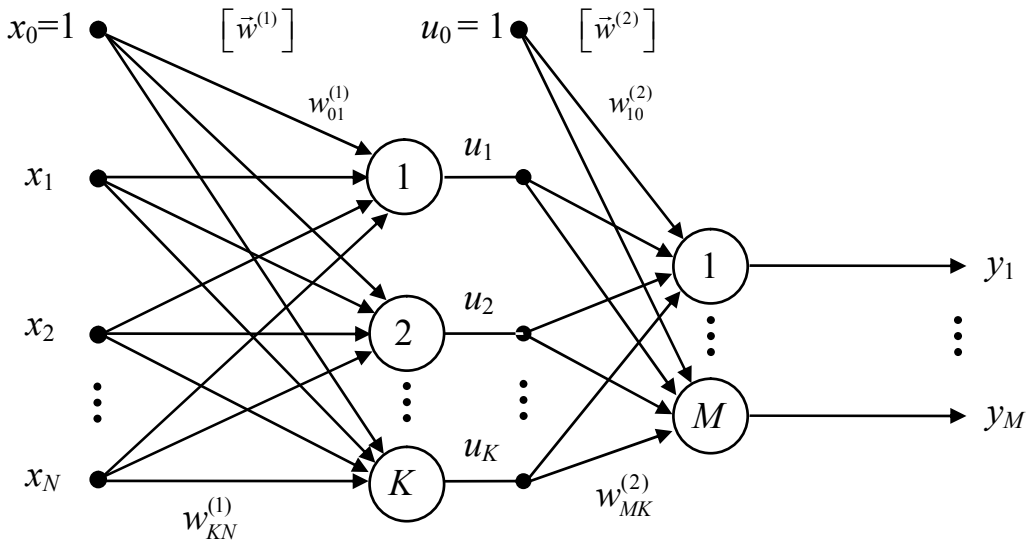

Рис. 4.2. Обобщенная структура двухслойной НС (с одним скрытым слоем)

4.2. Алгоритм обратного распространения ошибки (ОРО)

Алгоритм ОРО определяет стратегию подбора весов МСП с применением градиентных методов оптимизации и считается одним из наиболее эффективных алгоритмов обучения НС. Его основу составляет целевая функция вида (4.1). Уточнение весов может проводиться многократно после предъявления каждой обучающей пары $\left(\vec{x},\ \vec{d}\right)$ (режим «online») либо однократно после предъявления всех выборок, составляющих цикл обучения (режим «offline»). Формула для уточнения вектора весов имеет вид

$$\vec{w}_{t+1} = \vec{w}_t + \Delta\vec{w}_t = \vec{w}_t + \eta\vec{p}\left(\vec{w}_t\right), \tag{4.4}$$

где $\vec{p}\left(\vec{w}\right)$ – направление в многомерном пространстве $\vec{w}$. Для правильного выбора $\vec{p}\left(\vec{w}\right)$ необходимо определение вектора градиента относительно весов всех слоев сети, однако эта задача имеет очевидное решение только для весов выходного слоя. Для других слоев используется алгоритм ОРО, в соответствии с которым каждый цикл обучения состоит из следующих этапов:

1. По значениям компонент x_j входного вектора $\vec{x}$ расчет выходных сигналов $u_i^{(m)}$ всех слоев сети, а также соответствующих производных $\dfrac{df\left(u_i^{(m)}\right)}{du_i^{(m)}}$ функций активации каждого слоя (m – число слоев НС).

2. Создание сети ОРО путем замены выхода на вход, функций активации – их производными, входного сигнала $\vec{x}$ – разностью $\vec{y} - \vec{d}$.

3. Уточнение весов на основе результатов п. 1, 2 для оригинальной НС и сети ОРО.

4. Повторение процесса для всех обучающих выборок вплоть до выполнения условия остановки обучения (снижения нормы градиента до заданного ε, выполнения определенного количества шагов и т. п.).

Рассмотрим метод ОРО более подробно, полагая НС с одним скрытым слоем и режим обучения «online», когда $E\left(\vec{w}\right)$ определяется только одной обучающей парой. С учетом обозначений рисунка 4.2

$$E\left(\vec{w}\right) = \frac{1}{2}\sum_{k=1}^{M}\left[f\left(\sum_{i=0}^{K} w_{ki}^{(2)} u_i \right) - d_k \right]^2 = \frac{1}{2}\sum_{k=1}^{M}\left[f\left\{ \sum_{i=0}^{K} w_{ki}^{(2)} f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j \right) \right\} - d_k \right]^2, \tag{4.5}$$

откуда для компонентов градиента относительно выходного слоя получаем

$$\frac{\partial E}{\partial w_{ij}^{(2)}} = \left(y_i - d_i\right)\frac{df\left(u_i^{(2)}\right)}{du_i^{(2)}} u_j = \delta_i^{(2)} u_j. \tag{4.6}$$

Аналогично, для нейронов скрытого слоя

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = \sum_{k=1}^{M}\left(y_k - d_k\right)\frac{dy_k}{du_i}\frac{du_i}{dw_{ij}^{(1)}} = \sum_{k=1}^{M}\left(y_k - d_k\right)\frac{df\left(u_k^{(2)}\right)}{du_k^{(2)}} w_{ki}^{(2)} \frac{df\left(u_k^{(1)}\right)}{du_k^{(1)}} x_j = \delta_i^{(1)} x_j. \tag{4.7}$$

Обе полученные формулы имеют аналогичную структуру, дающую описание градиента в виде произведения двух сигналов: первый соответствует

начальному узлу данной взвешенной связи, второй – величине погрешности, перенесенной на тот узел, с которым эта связь установлена. В классическом алгоритме ОРО фактор $\vec{p}(\vec{w})$ задает направление отрицательного градиента, поэтому в выражении (4.4)

$$\Delta\vec{w} = -\eta\nabla E(\vec{w}). \qquad (4.8)$$

4.3. Радиальные нейронные сети (RBF–НС)

Рассмотренные выше многослойные сигмоидальные НС ввиду характера своей функции активации осуществляют аппроксимацию *глобального типа.* Однако возможен и другой подход – путем адаптации одиночных аппроксимирующих функций к ожидаемым значениям, когда отображение всего входного множества представляет собой сумму локальных преобразований с помощью функций, принимающих ненулевые значения в ограниченной области пространства данных, то есть *локальную аппроксимацию*. При этом особое семейство образуют НС, в которых скрытые нейроны описываются радиальными базисными функциями (RBF – Radial Basic Function) $\varphi(x) = \varphi(\|\vec{x} - \vec{c}\|)$, принимающие ненулевые значения только в окрестности выбранного центра $\vec{c}$. Эти сети представляют собой естественное дополнение сигмоидальных НС. Действительно, если сигмоидальный нейрон образует в многомерном пространстве гиперплоскость (рис. 4.3, а), то радиальный – гиперсферу (рис. 4.3, б), осуществляющую шаровое разделение пространства вокруг центральной точки, что в случае круговой симметрии входных данных позволяет значительно сократить число скрытых нейронов, необходимых для разделения различных классов.

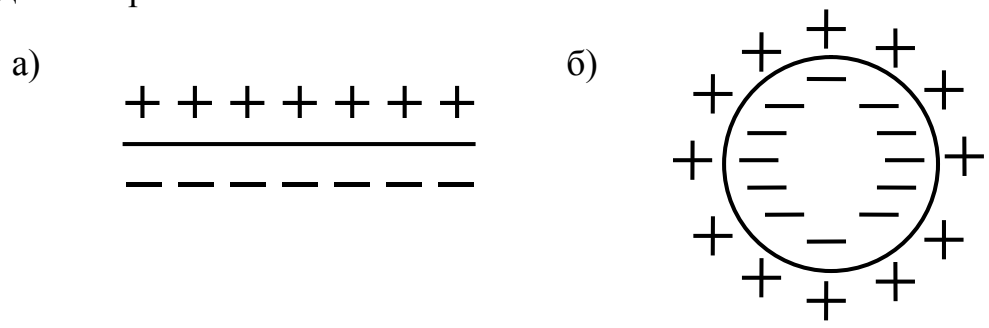

Рис. 4.3. Разделение пространства данных:
а – сигмоидальным нейроном; б – радиальным нейроном

Математическую основу функционирования RBF–НС составляет теорема Т. Ковера, согласно которой *N*-мерное входное пространство является φ-разделяемым на два пространственных класса X^+ и X^-, если существует такой вектор весов $\vec{w}$, что

$$\begin{aligned} &\vec{w}^T\vec{\varphi}(\vec{x}) > 0, \quad \textit{если } \vec{x} \in X^+; \\ &\vec{w}^T\vec{\varphi}(\vec{x}) < 0, \quad \textit{если } \vec{x} \in X^-. \end{aligned} \qquad (4.9)$$

Граница между этими классами определяется уравнением $\vec{w}^T\vec{\varphi}(\vec{x})=0$. Показано, что при достаточно большом числе скрытых нейронов *K*, реализующих радиальные функции $\varphi_i(\vec{x})$, решение задачи классификации гарантирует двухслойная НС, где скрытый слой реализует $\vec{\varphi}(\vec{x})$, а выходной слой состоит из одного или нескольких линейных нейронов, осуществляющих взвешенное суммирование сигналов, генерируемых скрытыми нейронами (рис. 4.4). Сеть функционирует по принципу многомерной интерполяции, состоящей в отображении *p* входных векторов $\vec{x}_i$ в множество из *p* рациональных чисел d_i (i = 1, 2, …, p), что возможно при *p* нейронах скрытого слоя и функции отображения $F(\vec{x}_i)=d_i$. Для RBF–НС с одним выходом (рис. 4.4) зависимость между входными и выходным сигналами может быть определена системой уравнений

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1p} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2p} \\ \vdots & \vdots & & \vdots \\ \varphi_{p1} & \varphi_{p2} & \cdots & \varphi_{pp} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix}, \qquad (4.10)$$

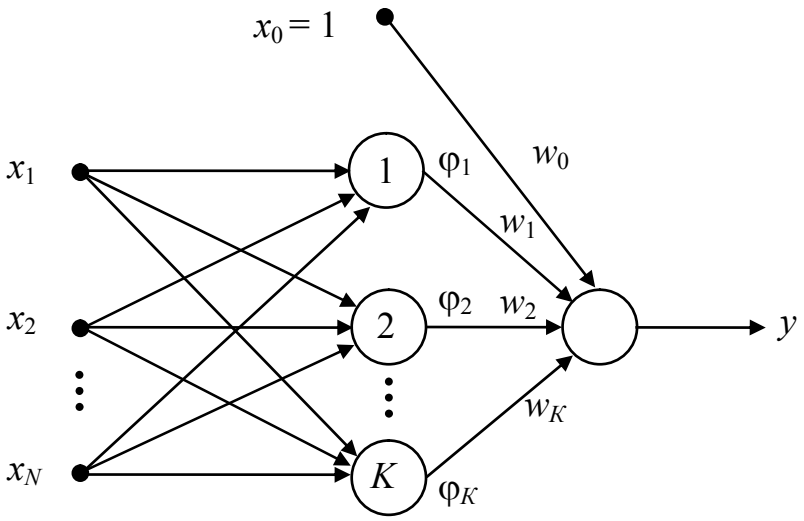

Рис. 4.4. Обобщенная структура радиальной НС

где $\varphi_{ji}=\varphi\left(\left\|\vec{x}_j-\vec{x}_i\right\|\right)$ определяет радиальную функцию с центром в точке $\vec{x}_i$ с вынужденным вектором $\vec{x}_j$. В сокращенной матричной форме система уравнений (4.10) может быть представлена как

$$[\Phi]\vec{w}=\vec{d}, \qquad (4.11)$$

откуда для ряда радиальных функций с $\vec{x}_1 \neq \vec{x}_2 \neq \ldots \neq \vec{x}_p$ может быть получено решение для вектора весов выходного слоя

$$\vec{w} = \left[\Phi\right]^{-1} \vec{d}. \tag{4.12}$$

Математически точное решение (4.12) системы (4.10) при $K = p$ совершенно неприемлемо с практической точки зрения по двум причинам:

1) наличие большого числа скрытых нейронов вызовет адаптацию НС к разного рода шумам или нерегулярностям, сопровождающим обучающие выборки, то есть обобщающие свойства НС окажутся весьма слабыми;

2) при большой величине p вычислительная сложность обучающего алгоритма становится чрезмерной.

Поэтому чаще всего отыскивается субоптимальное решение в пространстве меньшей размерности $K < p$, которое с достаточной степенью точности аппроксимирует точное, то есть

$$F\left(\vec{x}\right) = \sum_{i=1}^{K} w_i \varphi\left(\left\|\vec{x} - \vec{c}_i\right\|\right), \tag{4.13}$$

где c_i – множество центров RBF, которые необходимо определить (заметим, что при $K = p$ можно положить $\vec{c}_i = \vec{x}_i$).

Таким образом, задача обучения RBF–НС состоит в подборе определенного количества радиальных базисных функций, их параметров и весов w_i таким образом, чтобы решение уравнения (4.13) оказалось наиболее близким к точному. Эту проблему можно свести к минимизации некоторой целевой функции, которую при использовании метрики Эвклида можно записать как

$$E\left(\vec{w}\right) = \sum_{i=1}^{p} \left[\sum_{j=1}^{K} w_j \varphi\left(\left\|\vec{x}_i - \vec{c}_i\right\|\right) - \vec{d}_i \right]^2. \tag{4.14}$$

Чаще всего в качестве RBF применяется функция Гаусса

$$\varphi\left(\vec{x}\right) = \varphi\left(\left\|\vec{x} - \vec{c}_i\right\|\right) = \exp\left[-\frac{\left\|\vec{x} - \vec{c}_i\right\|^2}{2\sigma_i^2} \right], \tag{4.15}$$

где $\vec{c}_i$ означает расположение центра $\varphi\left(\vec{x}\right)$, а дисперсия σ_i определяет ширину радиальной функции, то есть процесс обучения при $K << p$ сводится к:

– подбору центров $\vec{c}_i$ и дисперсий σ_i радиальных функций (4.15);

– подбору весов w_i нейронов выходного слоя.

Поскольку значения w_i можно определить решением матрично-векторного уравнения типа (4.12), то главной проблемой обучения остается выбор $\vec{c}_i$ и σ_i, особенно центров RBF $\vec{c}_i$. Одним из простейших (хотя и не самых эффективных) методов является случайный выбор $\vec{c}_i$ на основе равномерного распределения при $\sigma = \frac{d}{\sqrt{2K}}$, где d – максимальное расстояние между $\vec{c}_i$. Очевидно, что ширина радиальных функций пропорциональна максимальному разбросу центров и уменьшается с ростом их количества.

Среди специализированных методов выбора центров RBF прежде всего следует выделить алгоритмы самоорганизации, когда множество входных обучающих данных разделяется на кластеры, которые в дальнейшем представляются центральными точками, определяющими усредненные значения всех их элементов. Эти точки в дальнейшем выбираются в качестве центров соответствующих радиальных функций, то есть количество RBF равно количеству кластеров. Для разделения данных на кластеры чаще всего используют алгоритм *K*-усреднений Линде–Бузо–Грея в прямом («online») или накопительном («offline») варианте. При этом начальные положения центров $\vec{c}_i$ выбираются случайным образом на основе равномерного распределения, а затем производится их уточнение либо после предъявления каждого очередного $\vec{x}$ (online), либо после предъявления всех элементов обучающего множества (offline). Если обучающие данные представляют непрерывную функцию, начальные значения $\vec{c}_i$ в первую очередь размещают в точках экстремумов (максимумов и минимумов) функции, а оставшиеся центры распределяют равномерно среди незадействованных элементов обучающего множества.

В прямой версии («online») после подачи каждого обучающего вектора $\vec{x}_k$ выбирается ближайший к $\vec{x}_k$ центр $\vec{c}_i$ и подвергается уточнению в соответствии с алгоритмом WTA

$$\vec{c}_i(t+1) = \vec{c}_i(t) + \eta\left[\vec{x}_k - \vec{c}_i(t)\right], \tag{4.16}$$

где $\eta << 1$ – коэффициент обучения, уменьшающийся с ростом t, а остальные центры не изменяются. Все обучающие $\vec{x}_k$ предъявляются случайным образом по несколько раз, вплоть до стабилизации положения $\vec{c}_i$. В режиме «offline» уточнение положения всех $\vec{c}_i$ происходит параллельно после подачи всех обучающих векторов $\vec{x}_k$ согласно

$$\vec{c}_i(t+1) = \frac{1}{N_i}\sum_{j=1}^{N_i}\vec{x}_j(t), \tag{4.17}$$

где N_i – количество $\vec{x}_k$, приписанных к $\vec{c}_i$ в цикле t. На практике чаще применяется прямой алгоритм, имеющий несколько лучшую сходимость.

Основная трудность алгоритмов самоорганизации – выбор коэффициента обучения η. При η = const он должен быть очень малым, что, гарантируя сходимость алгоритма, непомерно увеличивает время обучения. Из адаптивных методов подбора η наиболее известен алгоритм Даркена–Муди, согласно которому

$$\eta(t) = \frac{\eta_0}{1 + \frac{t}{T}}, \tag{4.18}$$

где T – постоянная времени, индивидуальная для каждой задачи. Несмотря на то, что адаптивные методы выбора η более прогрессивны по сравнению с η = const, они также не могут считаться наилучшим решением, особенно при моделировании динамических процессов.

После фиксации положения $\vec{c}_i$ производится подбор значений σ_i таким образом, чтобы области охвата всех RBF накрывали все пространство входных данных, лишь незначительно перекрываясь друг с другом. Проще всего в качестве σ_i выбрать эвклидово расстояние между $\vec{c}_i$ и его ближайшим соседом $\vec{c}_j$, то есть $\sigma_i = \sqrt{\left\|\vec{c}_i - \vec{c}_j\right\|^2}$, но можно учитывать и более широкое соседство с помощью

$$\sigma_i = \sqrt{\frac{1}{p}\sum_{j=1}^{p}\left\|\vec{c}_i - \vec{c}_j\right\|^2}, \tag{4.19}$$

где обычно $p \in [3,5]$. Заметим, что существуют и другие алгоритмы обучения НС–RBF (вероятностный, гибридный, на основе ОРО), однако ни один из них не гарантирует 100 %-й оптимальности результата.

Поскольку RBF–НС используются для решения тех же задач (классификация, аппроксимация, прогнозирование), что и сигмоидальные НС, основной проблемой их корректного построения является оптимальный выбор количества скрытых нейронов, то есть числа RBF. Как правило, величина K зависит от многих факторов, прежде всего от размерности $\vec{x}$, объема обучающих данных p и разброса $\vec{d}_i \Leftrightarrow \vec{x}_i$, то есть пространственной структуры аппроксимируемой функции. Для подбора K используют:

1) *эвристические методы,* использующие алгоритмы увеличения или уменьшения числа RBF по оценке динамики изменения $E(\vec{w})$;

2) *метод ортогонализации Грэма–Шмидта*, когда при начальной фиксации $K = p$ количество скрытых нейронов постепенно уменьшается путем выделения оптимального числа RBF, дающих наибольший вклад в энергетическую функцию $E(\vec{w})$, то есть путем определения необходимой размерности $\vec{w}$, гарантирующей наилучшие результаты обучения.

Радиальные НС относятся к той же категории сетей, обучаемых с учителем, что и сигмоидальные НС (например, МСП), однако обнаруживают значительные отличия:

– RBF–НС имеют фиксированную структуру с одним скрытым слоем и линейными выходными нейронами;

– обобщающие способности радиальных НС несколько хуже ввиду глобального характера сигмоидальных функций активации;

– в отличие от сигмоид RBF могут быть весьма разнообразны, что увеличивает вероятность достижения успеха с их помощью;

– RBF–НС имеют более простой (и более быстрый) алгоритм обучения, поскольку этапы определения $\vec{c}_i$, σ_i и $\vec{w}$ можно разделить;

– возможность лучшего выбора начальных условий обучения радиальных НС увеличивает вероятность достижения глобального минимума $E(\vec{w})$;

– радиальные НС обеспечивают лучшее решение классификационных задач.

4.4. Специализированные структуры НС

Специализированные НС обеспечивают оптимальный выбор архитектуры, сочетая определение структуры НС с ее обучением. К их числу относятся сеть каскадной корреляции Фальмана и сеть Вольтерри.

4.4.1. НС каскадной корреляции Фальмана

Эта сеть представляет собой многослойную конструкцию, в которой формирование структуры НС происходит параллельно с ее обучением путем добавления на каждом этапе обучения одного скрытого нейрона. Архитектура сети каскадной корреляции представляет собой объединение нейронов взвешенными связями в виде развивающегося каскада (рис. 4.5), где каждый очередной добавляемый нейрон подключается ко всем уже существующим нейронам (входным, скрытым, выходным), причем входные узлы НС напрямую подключаются также и к выходным нейронам.

Начальный этап включает формирование структуры НС только из входных и выходных нейронов, количество которых определяется спецификой решаемой задачи и не подлежит модификации. Каждый вход соединен со всеми выходными нейронами, функция активации которых может быть любой. Обучение состоит в подборе весов связей любым методом обучения (в оригинале Quickprop Фальмана) на основе минимизации целевой функции $E(\vec{w})$. Если результат обучения удовлетворителен с точки зрения допустимой погрешности, процесс формирования структуры НС считается законченным. В противном случае в структуру НС добавляется один скрытый нейрон, образующий одноэлементный скрытый слой, в котором веса входных связей фиксированы, а обучению (коррекции) подлежат только веса его связей с выходными нейронами.

Формирование каждого скрытого слоя начинают с подготовки нейронов-кандидатов (обычно 5…10), подбор входных весов которых (фиксируемых при включении в НС) осуществляется по значению максимума функции корреляции *S*, зависящей от выходного сигнала кандидата при подаче всех обучающих выборок. Каждый нейрон-кандидат представляет собой обособленный элемент, соединенный со всеми входами сети и с выходами ранее введенных нейронов. Начальные веса нейронов-кандидатов выбирают случайным образом, после обучения в каскадную НС ставится лучший из претендентов, что уменьшает вероятность попадания НС в точку локального минимума $E(\vec{w})$ из-за ввода в сеть нейрона с плохо подобранными входными весами, которые уже невозможно будет откорректировать на последующих этапах обучения.

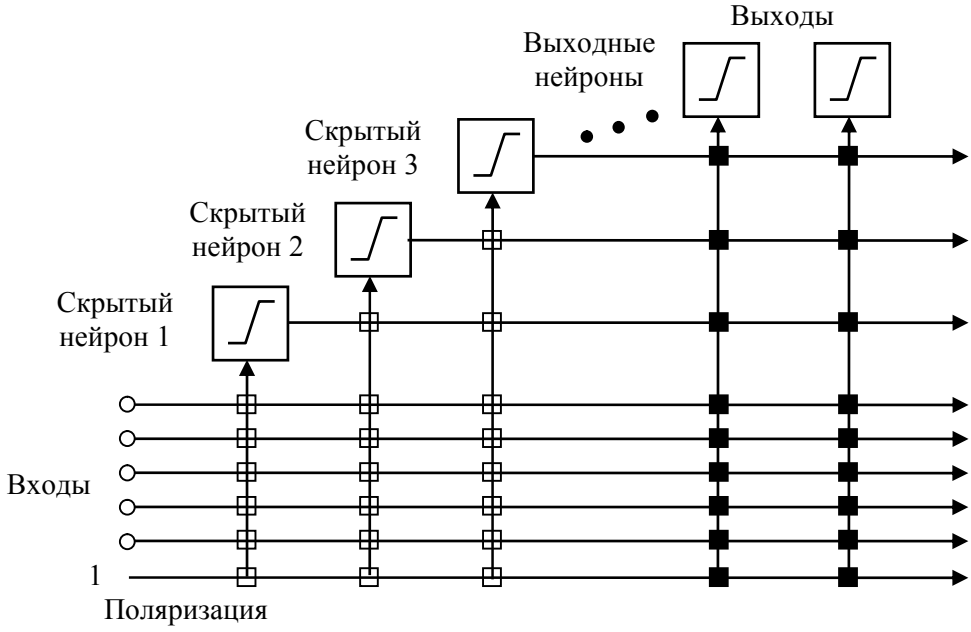

Рис. 4.5. Структура сети каскадной корреляции Фальмана

Каждый нейрон, претендующий на включение в сетевую структуру, может иметь свою функцию активации – сигмоидальную, гауссовскую, радиальную и т. п. Поскольку побеждают те нейроны, которые лучше приспосабливаются к условиям, созданным множеством обучающих данных, то сеть Фальмана может объединять нейроны с различными функциями активации. Заметим, что, несмотря на большое количество слоев, НС Фальмана не требует применения алгоритма ОРО при обучении, поскольку в процессе минимизации $E(\vec{w})$ задействованы весовые коэффициенты только выходного слоя, для которых погрешность рассчитывается непосредственно.

Для проверки обобщающих свойств НС Фальмана был рассмотрен пример аппроксимации функции двух переменных $f(x,y)=0{,}5\sin(\pi x^2)\sin(2\pi y)$ для значений $x,y \in [-1, 1]$. В качестве обучающих данных использовались 500 зна-

чений этой функции, равномерно распределенных по всему диапазону, сеть обучалась из условия $\left|E\left(\vec{w}\right)\right| < 0{,}01$. График изменения $E\left(\vec{w}\right)$ в зависимости от номера итерации представлен на рисунке 4.6.

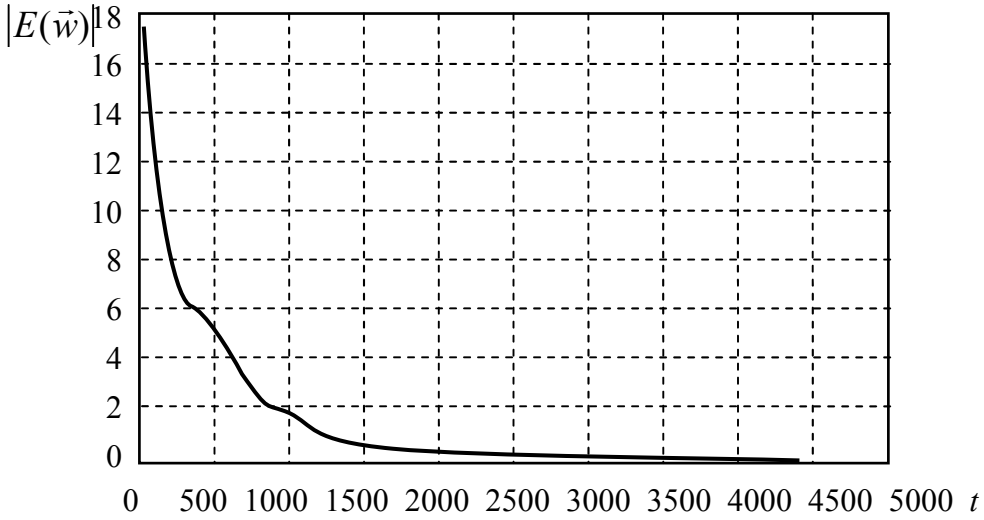

Рис. 4.6. График кривой обучения НС каскадной корреляции на примере трехмерной функции

Ожидаемое значение погрешности обучения было получено на выходе НС при введении в ее структуру 41-го скрытого нейрона. В качестве тестирующих данных были сгенерированы 1000 значений функции в других точках того же диапазона. Результаты тестирования подтвердили хорошие обобщающие способности сети.

4.4.2. НС Вольтерри

Сеть Вольтерри – это динамическая сеть для нелинейной обработки последовательности сигналов, задержанных относительно друг друга. Входным вектором сети в момент *t* служит $\vec{x} = \left[x_t,\ x_{t-1},\ \ldots,\ x_{t-L}\right]^T$, где *L* – количество единичных задержек, а (*L*+ 1) – длина вектора. Выходной сигнал *y* в соответствии с определением ряда Волтерри описывается формулой

$$y(t) = \sum_{i=1}^{L} w_i x\left(t-i\right) + \sum_{i=1}^{L}\sum_{j=1}^{L} w_{ij} x\left(t-i\right)x\left(t-j\right) + \ldots, \qquad (4.20)$$

где веса w_i, w_{ij}, …, называемые ядрами Вольтерри, соответствуют реакциям высших порядков. Для адаптации НС Вольтерри к заданной последовательности значений *d*(*t*) формируется целевая функция $E\left(\vec{w}\right) = \frac{1}{2}\left[y(t) - d(t)\right]^2$ и производится ее минимизация на основе решения системы дифференциальных уравнений

$$\frac{d\vec{w}}{dt} = -\mu\frac{dE}{d\vec{w}}. \qquad (4.21)$$

Нетрудно показать, что при степени ряда Вольтерри $K = 2$ система (4.21) может быть записана в виде

$$\begin{aligned} \frac{dw_i(t)}{dt} &= -\mu\left[y(t) - d(t)\right]x\left(t - i\right); \\ \frac{dw_{ij}(t)}{dt} &= -\mu\left[y(t) - d(t)\right]x\left(t - i\right)x\left(t - j\right) \quad i, j = 1,\ 2,\ \ldots,\ L. \end{aligned} \tag{4.22}$$

Для упрощения структуры сети и уменьшения ее вычислительной сложности разложение Вольтерри (4.20) можно представить следующим образом:

$$y_t = \sum_{i=0}^{L} x_{t-i}\left[w_i + \sum_{j=0}^{L} x_{t-j}\left[w_{ij} + \sum_{k=0}^{L} x_{t-k}\left(w_{ijk} + \ldots \right) \right] \right], \tag{4.23}$$

где введены обозначения $y_t \equiv y(t)$; $x_{t-i} = x(t-i)$ и т. д. Пример структуры, реализующий распространение сигнала по сети с зависимостью (4.23) и числом уровней $K = 2$ (рис. 4.7), показывает, что система является типичной многослойной однонаправленной динамической НС с полиномиальной нелинейностью. Подбор весов НС производится последовательно слой за слоем, причем эти процессы независимы друг от друга, что позволяет существенно увеличивать длину L и порядок K системы при ее практической реализации.

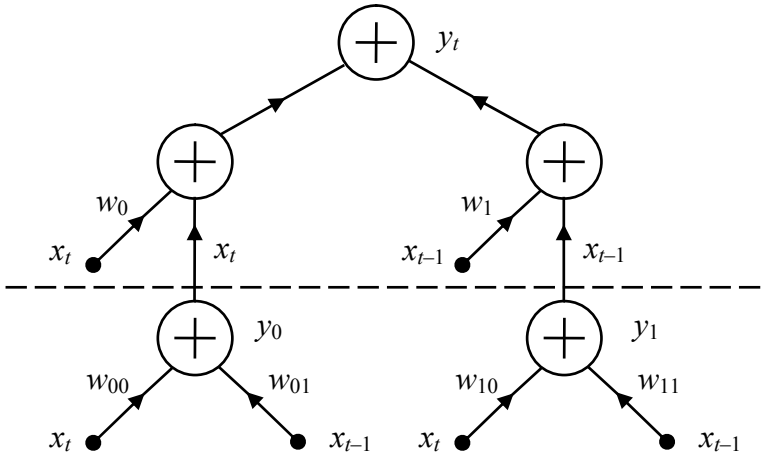

Рис. 4.7. Структура двухслойной НС Вольтерри

Нелинейность НС Вольтерри позволяет успешно использовать ее для решения таких задач, как:

1. Идентификация нелинейного объекта (рис. 4.8), когда одна и та же последовательность входных сигналов подается на объект и его модель в виде НС Вольтерри, управляемой адаптивным алгоритмом таким образом, чтобы сигнал рассогласования $\varepsilon(t) = y(t) - d(t)$ в процессе адаптации параметров сети снижался до нуля. Присущая НС Вольтерри нелинейность позволила значительно расширить класс идентифицируемых объектов.

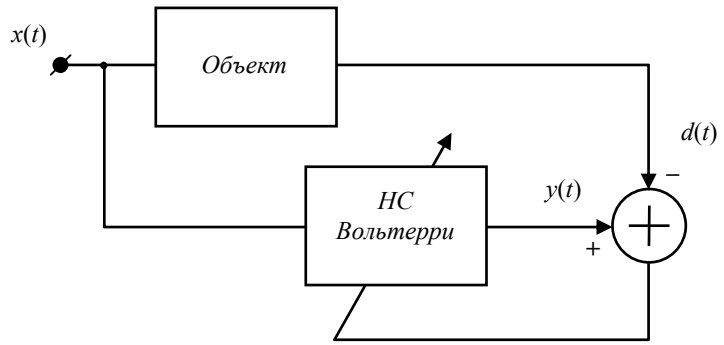

Рис. 4.8. Адаптивная система идентификации динамического объекта на основе НС Вольтерри

2. Подавление интерференционных шумов (рис. 4.9), где в качестве обрабатываемого сигнала используется смесь полезного сигнала S с некоррелируемым с ним шумом n_0, то есть $x = S + n_0$, а входным сигналом НС – установочный сигнал n, который не коррелирует с S, однако неизвестным образом коррелирует с сигналом помехи n_0. Задача НС состоит в такой обработке сигнала n, чтобы сигнал y был наиболее близок к n_0. Поскольку сигнал погрешности на выходе сумматора $\varepsilon = S + n_0 - y$, то целевую функцию $E(\vec{w})$ можно представить как

$$E(\vec{w}) = 0{,}5E\left[\ \varepsilon^2\right] = 0{,}5E\left[\ S^2 + (n_0 - y)^2 + 2S(n_0 - y)\right]\ , \qquad (4.24)$$

а если принять во внимание, что S не коррелирует с сигналами помехи, то

$$E(\vec{w}) = 0{,}5\left\{E(S^2) + E\left[\ (n_0 - y)^2\right]\right\}\ . \qquad (4.25)$$

Таким образом, минимизация $E(\vec{w})$ означает наилучшую адаптацию y к помехе $n_0\left[E(\vec{w})_{\min} = E(S^2)\right]$, поскольку в этом случае выходной сигнал ε соответствует полностью очищенному от шума полезному сигналу S.

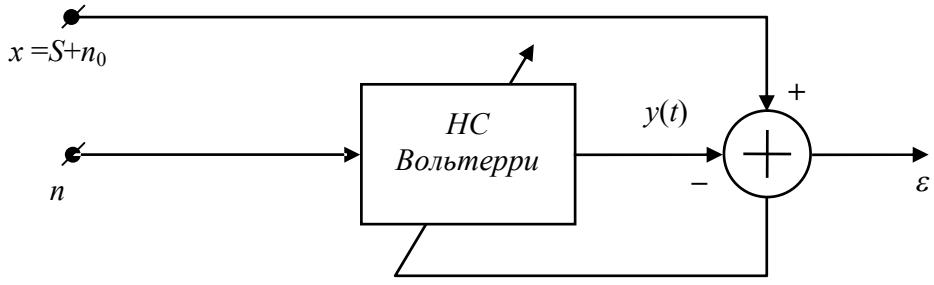

Рис. 4.9. Применение НС Вольтерри в адаптивной системе подавления интерференционных шумов

3. Прогнозирование нестационарных сигналов (рис. 4.10), когда выходной сигнал НС Вольтерри описывается формулой (4.23) с заменой x_{t-}

$_i \Rightarrow h_{t-i}$, где $h_t \equiv h(t)$ обозначает задержанный сигнал x_t, а решение задачи адаптации весов находится из системы дифференциальных уравнений типа (4.22) с той же заменой. Компьютерный анализ показывает, что учет нелинейности фильтра Вольтерри значительно повышает качество прогнозирования.

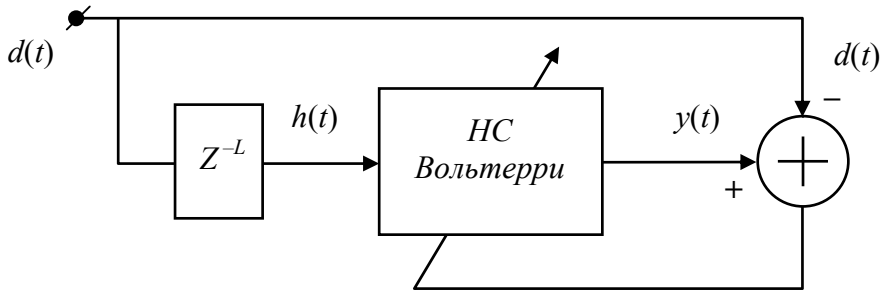

Рис. 4.10. Использование НС Вольтерри в качестве прогнозирующей системы

Контрольные вопросы

1. В чем причина ограниченности возможностей однослойного персептрона?
2. Приведите структуру и опишите функционирование МСП.
3. В чем сходство и в чем отличие режимов обучения «online» и «offline» нейронной сети?
4. Каковы основные этапы обучения НС методом обратного распространения ошибки?
5. Чем отличаются методы аппроксимации многомерных функций с помощью сигмоидальных и радиальных НС?
6. Что составляет математическую основу функционирования RBF–НС?
7. При каких условиях возможно точное решение матричного уравнения для определения весов w_i RBF–НС?
8. Почему математически точное решение матричной системы уравнений для определения w_i неприемлемо с практической точки зрения?
9. В чем заключается задача обучения радиальных НС?
10. Какие алгоритмы обучения RBF–НС Вы знаете?
11. Как располагаются центры RBF, если множество обучающих данных представляет собой непрерывную функцию?
12. Расскажите о характере подбора коэффициента дисперсии σ_i RBF при обучении радиальных НС.
13. Как осуществляется оптимальный выбор количества скрытых нейронов RBF–НС?
14. Что показывает сравнительный анализ характеристик сигмоидальных и радиальных НС?

15. Как осуществляется формирование структуры НС каскадной корреляции Фальмана на начальном этапе?

16. Зачем и каким образом производится подготовка нейронов-кандидатов для НС Фальмана?

17. Какие функции активации используются при формировании скрытых слоев НС Фальмана?

18. Расскажите о процессе обучения НС каскадной корреляции. Почему обучение НС Фальмана не требует применения алгоритма ОРО?

19. Что из себя представляет НС Вольтерри? Как формируется выходной сигнал и производится обучение этой сети?

20. Расскажите об идентификации нелинейных объектов с помощью сети Вольтерри.

21. Как осуществляется подавление интерференционных шумов на основе нелинейного фильтра (НС) Вольтерри?

22. Можно ли использовать НС Вольтерри для прогнозирования нестационарных сигналов? Если «да», то каким образом?

ЛИТЕРАТУРА

1. *Хайкин С.* Нейронные сети : полный курс / С. Хайкин. – М. : ИД «Вильямс», 2006. – 1104 с.

2. *Осовский С.* Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2004. – 344 с.

3. *Круглов В.В.* Искусственные нейронные сети : теория и практика / В.В. Круглов, В.В. Борисов. – М. : Горячая линия–Телеком, 2002. – 382 с.

4. *Тарков М.С.* Нейрокомпьютерные системы : учеб. пособие / М.С. Тарков. – М. : Интернет–Университет информационных технологий ; БИНОМ. Лаборатория знаний, 2006. – 142 с.

5. *Медведев В.С.* Нейронные сети. MATLAB 6 / В.С. Медведев, В.Г. Потемкин. – М. : ДИАЛОГ–МИФИ, 2002. – 496 с.

6. *Круг П.Г.* Нейронные сети и нейрокомпьютеры : учеб. пособие / П.Г. Круг. – М. : Изд-во МЭИ, 2002. – 176 с.

7. *Галушкин А.И.* Нейрокомпьютеры : учеб. пособие для вузов / А.И. Галушкин. – М. : ИПРЖР, 2000. – 528 с.

8. *Уоссермен Ф.* Нейрокомпьютерная техника : теория и практика / Ф. Уоссермен. – М. : Мир, 1992. – 240 с.

9. *Заенцев И.В.* Нейронные сети : основные модели / И.В. Заенцев. – Воронеж : ВГУ, 1999. – 78 с.

10. *Суровцев И.С.* Нейронные сети. Введение в современную информационную технологию / И.С. Суровцев, В.И. Клюкин, Р.П. Пивоварова. – Воронеж : ВГУ, 1994. – 224 с.

11. *Клюкин В.И.* Моделирование нейронных сетей в среде MATLAB: учеб. пособие / В.И. Клюкин, Ю.К. Николаенков. – Воронеж : ВГУ, 2007. – 60 с.

12. *Каллан Р.* Основные концепции нейронных сетей / Р. Каллан. – М. : ИД «Вильямс», 2001. – 288 с.

Учебное издание

НЕЙРОСЕТЕВЫЕ СТРУКТУРЫ И ТЕХНОЛОГИИ
Часть 1
Электрические и математические модели нейронов. НС прямого распространения

Учебное пособие для вузов

Составители:
Клюкин Владимир Иванович
Николаенков Юрий Кимович

Редактор Л.М. Носилова

Подписано в печать 20.03.09. Формат 60×84/16. Усл. печ. л. 3,7.
Тираж 50 экз. Заказ 451.

Издательско-полиграфический центр
Воронежского государственного университета.
394000, г. Воронеж, пл. им. Ленина, 10. Тел. 208-298, 598-026 (факс)
http://www.ppc.vsu.ru; e-mail: pp_center@ppc.vsu.ru

Отпечатано в типографии Издательско-полиграфического центра
Воронежского государственного университета.
394000, г. Воронеж, ул. Пушкинская, 3. Тел. 204-133