

$t > t'$ . В табл. 2 приведены координаты точек кривой (19) на рисунке, которая выделяет подобласть допустимых значений  $x$ ,  $y$ , где выполняется доста-

точное условие оптимальности МР (в силу симметрии выражений относительно  $x$  и  $y$ :  $y = x_2$  при  $x = x_1$  и  $y = x_1$  при  $x = x_2$ ).

## Литература

1. **Смирнов Ю. М.** Состояние и перспективы развития методов системного моделирования // Методы кибернетики и информационные технологии: Сб. научных трудов. — Вып. 1., — Саратов: Изд-во Саратовского университета, 1994. — С. 34–40.
2. **Смирнов Ю. М.** Направления развития методологии системного проектирования // Вычислительная техника, автоматика и радиоэлектроника. Труды СПбГТУ. — СПб.: Изд-во СПбГТУ, — 1998. № 472. — С. 109–123.
3. **Смирнов Ю. М.** Системное проектирование комплексов управления летательными аппаратами: Уч. пособие. — СПб.: Изд-во СПбГТУ, 1996. — 120 с.
4. **Поляков А. О., Смирнов Ю. М., Турчак А. А.** Информационно-динамические основы организации управления крупными предприятиями и холдингами. — СПб.: Изд-во СПбГПУ, 2002. — 192 с.
5. **Смирнов Ю. М., Швырков В. Г.** Математические методы внешнего проектирования систем // Вычислительная техника, автоматика и радиоэлектроника. Труды СПбГТУ. — СПб.: Изд-во СПбГТУ, 2002. — № 488 — С. 32–58.
6. **Смирнов Ю. М.** Планирование испытаний в условиях неопределенности // Управление в условиях неопределенности: Сб. Под ред. А. Е. Городецкого. — СПб.: Изд-во СПбГТУ, 2002. — С. 339–360.

УДК 621-52:004.52; 629.78; 681.3

# МНОГОСЛОЙНАЯ ПЕРСЕПТРОННАЯ НЕЙРОННАЯ СЕТЬ В ЗАДАЧЕ МОДЕЛИРОВАНИЯ РЕЧЕВЫХ СИГНАЛОВ

**Ю. Я. Изилов,**

канд. техн. наук

Санкт-Петербургский государственный политехнический университет

*Рассматриваются возможности применения искусственной нейронной сети перцептронного типа в задаче моделирования речевых сигналов. Приводятся структуры и алгоритм, позволяющий осуществить функционирование многослойной перцептронной нейросети на персональном компьютере. Обсуждаются проблемы, возникающие при практической реализации, а также достоинства и недостатки использования данного подхода.*

*This article considers the opportunities of application artificial neural network based on perceptron type for speech signals modeling task. Structures and the algorithm, which allows to carry out functioning of multilayer perceptron neural network on personal computer, are resulted. The problems arising at practical realization, and also limits and advantages of the given approach usage are discussed.*

## Введение

Речь представляет собой важнейшее и самое удобное средство взаимодействия между людьми. В этой связи, в условиях компьютеризированного общества, понятно стремление специалистов ос-

настить автоматизированные системы различного назначения средствами речевого ввода—вывода информации.

Речевой сигнал (РС) представляет собой многоуровневую структуру и характеризуется определенной иерархией: фонемы, слова, фразы и т. д.

Основу такой иерархии составляют акустические сигналы, моделирование которых вызывает наибольшую трудность.

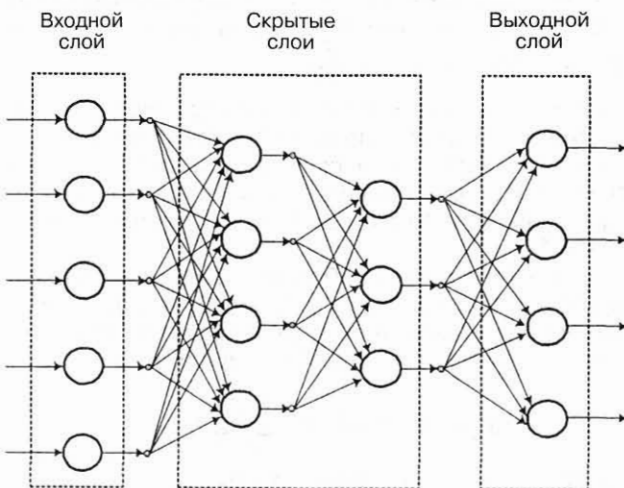
С точки зрения акустики, РС состоит из быстрых и очень неустойчивых колебаний воздушного давления [3], что, в свою очередь, порождает значительную неустойчивость параметрического описания.

Неустойчивость РС учитывают путем слежения за его вероятностными характеристиками (такими, как функция распределения вероятностей и/или моментные функции), для оценки которых необходимо задать модель сигнала. На практике апробированные сведения о вероятностном характере РС не являются полными для того, чтобы можно было явно задать его вероятностную модель. Поэтому поведение РС приходится изучать по опытным данным. С учетом сказанного, в качестве альтернативной модели может быть выбрана искусственная нейронная сеть.

Нейронные сети (НС) являются важным дополнением традиционных методов обработки данных [1, 2, 4, 6, 13]. Наибольшее распространение среди них получили так называемые многослойные персептронные НС, или многослойные персептроны. Они могут использоваться при решении различного рода задач, таких, как прогнозирование, сжатие данных, аппроксимация функций, распознавание и восстановление образов.

Достоинства многослойных персептронов (МП) — возможность построения сложных разделяющих поверхностей, возможность осуществления любого отображения входных векторов в выходные, способность к обобщению входных данных. При этом не требуется знания закона распределения входных векторов. Вместе с тем, МП сводит задачу моделирования РС к параметрической идентификации входных векторов. Рассмотрим модель МП более подробно.

Многослойный персептрон представляет собой модель НС с прямыми связями [6, 14, 17, 18]. Она включает в себя входной слой, выходной слой и



■ Рис. 1. Многослойный персептрон

находящиеся между ними промежуточные, так называемые скрытые слои нейронов (рис. 1).

Входной слой нейронов не производит никаких вычислений. Он осуществляет распределение данных и работает в качестве разветвителя. Скрытые слои, минимальное число которых должно быть не менее одного, предназначены непосредственно для обработки входной информации и внутреннего представления данных. Выходной слой предназначен для обработки информации, поступающей от предыдущего слоя, и выдачи соответствующих результатов.

Топология МП предусматривает соединение выхода каждого нейрона предыдущего слоя с входом каждого нейрона последующего слоя.

Особенностью МП является возможность модификации формы областей решений посредством изменения количества слоев нейронов. При этом двухслойная НС (с одним слоем нейронных связей) формирует области решений в виде гиперплоскостей; трехслойная НС (с двумя слоями нейронных связей) позволяет формировать любые выпуклые области в пространстве решений; четырехслойная НС (с тремя слоями нейронных связей) позволяет получать область решений произвольной формы и сложности [2, 13].

МП предполагает выполнение процесса обучения. Наиболее распространенный метод обучения — это алгоритм обратного распространения ошибки [15, 20]. Его можно применить к МП с любым числом слоев нейронных связей. Однако для того, чтобы понять идею алгоритма, достаточно рассмотреть его работу на примере МП с двумя слоями нейронных связей (рис. 2).

Алгоритм обратного распространения ошибки производит настройку весовых коэффициентов в многослойной НС на основе использования метода градиентного спуска. Он предполагает, что обучение происходит с учителем, и требует предъявления обучающей выборки.

Обучающая выборка представляет собой множество пар входных и выходных векторов  $\{(X_m, G_m) | m = 1, 2, \dots, M\}$ , между которыми нужно установить соответствие.

Множество нейронных связей в данном случае (см. рис. 2) образуется двумя матрицами весовых коэффициентов:  $W^{(1)}, W^{(2)}$  с элементами  $w_{ik}^{(1)}$  и  $w_{kj}^{(2)}$  соответственно. Порог возбуждения  $k$ -го нейрона в скрытом слое формируется связью с весовым коэффициентом  $w_{0k}^{(1)}$ , по которой поступает сигнал  $u_0 = 1$ . Порог возбуждения  $j$ -го нейрона в выходном слое формируется связью с весовым коэффициентом  $w_{0j}^{(2)}$ , по которой поступает сигнал  $h_0 = 1$ .

Эффективность функционирования НС для некоторого конечного набора обучающих пар можно определить с помощью общей ошибки сравнения фактических и требуемых выходных векторов для каждого случая. Для этого вначале вычислим ошибку сравнения каждой пары векторов типа «вход—выход»:

$$E_m = \frac{1}{2} \sum_{j=1}^J (g_{j,m} - y_{j,m})^2, \quad (1)$$

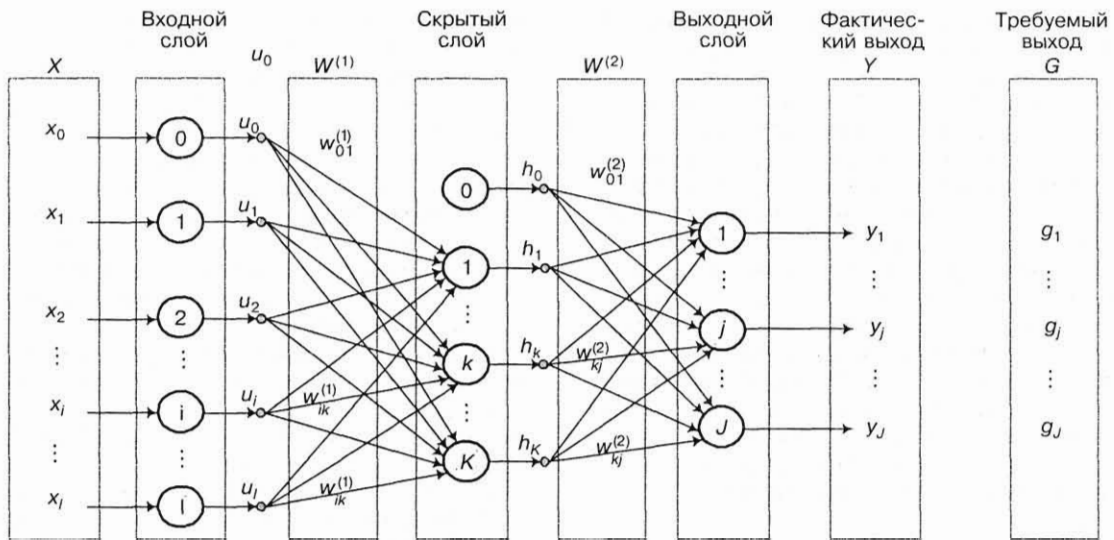


Рис. 2. Многослойный перцептрон с двумя слоями нейронных связей

где  $g_{j,m}$  — требуемое значение выходного сигнала  $j$ -го выходного нейрона для  $m$ -й обучающей пары;  $y_{j,m}$  — фактическое значение выходного сигнала  $j$ -го выходного нейрона для  $m$ -й обучающей пары.

Теперь общую ошибку функционирования НС, с учетом (1), можно вычислить по формуле

$$E = \sum_{m=1}^M E_m = \frac{1}{2} \sum_{m=1}^M \sum_{j=1}^J (g_{j,m} - y_{j,m})^2. \quad (2)$$

Целью обучения НС является такая настройка ее весовых коэффициентов, при которой для каждого входного вектора найдется выходной вектор, сгенерированный НС и представляющий собой требуемый выходной вектор или достаточно близкий к нему.

Другими словами, процесс обучения НС сводится к нахождению таких матриц весовых коэффициентов  $W^{(1)}, W^{(2)}$ , которые минимизируют общую ошибку функционирования нейросети  $E$ . Для решения данной задачи алгоритм обратного распространения ошибки использует метод градиентного спуска. При этом поиск минимального значения производится по всему пространству весовых коэффициентов НС.

Метод градиентного или наискорейшего спуска предполагает, что коррекция весовых коэффициентов происходит итерационным способом: малым перемещением и в том направлении, при котором ошибка функционирования НС минимизируется наискорейшим образом. Обычно используется следующая форма:

$$W(t+1) = W(t) + \Delta W(t), \quad (3)$$

где  $t$  — шаг итерации;  $\Delta W(t)$  — коррекция весовых коэффициентов;  $W(t)$  — значения весовых коэффициентов до коррекции;  $W(t+1)$  — значения весовых коэффициентов после коррекции.

Согласно этому методу, частная производная меры ошибки относительно каждого веса пропорциональна изменению соответствующего веса с отрицательным коэффициентом пропорциональности:

$$\Delta W = -\eta \frac{\partial E}{\partial W}, \quad (4)$$

где  $\Delta W$  — коррекция весового коэффициента;  $\eta$  — коэффициент скорости обучения.

Коэффициент  $\eta$  в формуле (4) обозначает небольшое положительно определенное число, которое обычно находится в пределах от 0,01 до 1. Он влияет на шаг итерации и, соответственно, позволяет управлять скоростью сходимости алгоритма обучения.

Результаты вычислений согласно выражениям (3), (4) будут соответствовать линии, проходящей по поверхности ошибок в пространстве весовых коэффициентов, которая ведет к наискорейшему спуску. Следовательно, задача коррекции весовых коэффициентов эквивалентна задаче вычисления частных производных  $\frac{\partial E}{\partial W}$ .

Задача алгоритма обучения сводится к минимизации функции ошибки НС относительно всего множества нейронных связей. При этом функцию ошибки НС, как правило, задают формулой (2), а уровень допустимых ошибок ограничивают величиной  $\xi$ .

В соответствии с обозначениями сигналов и весовых коэффициентов (см. рис. 2) выходное значение  $j$ -го нейрона в выходном слое для  $m$ -й обучающей пары определяется по формуле

$$y_{j,m} = F_j(S_{j,m}) = F_j \left( \sum_{k=0}^K w_{kj}^{(2)} h_{k,m} \right), \quad (5)$$

где  $S_{j,m}$  — значение уровня возбуждения  $j$ -го нейрона выходного слоя для  $m$ -й обучающей пары;

$F_j(\cdot)$  — функция активации  $j$ -го нейрона выходного слоя;  $w_{kj}^{(2)}$  — весовой коэффициент для связи, идущей от  $k$ -го нейрона скрытого слоя к  $j$ -му нейрону выходного слоя;  $h_{k,m}$  — выходное значение сигнала  $k$ -го нейрона скрытого слоя для  $m$ -й обучающей пары,  $h_{0,m} = 1$  соответствует сигналу, формирующему пороговое значение.

Выходное значение  $k$ -го нейрона скрытого слоя для  $m$ -й обучающей пары формируется следующим образом:

$$h_{k,m} = F_k(S_{k,m}) = F_k\left(\sum_{i=0}^l w_{ik}^{(1)} u_{i,m}\right) = F_k\left(\sum_{i=0}^l w_{ik}^{(1)} x_{i,m}\right), \quad (6)$$

где  $S_{k,m}$  — значение уровня возбуждения  $k$ -го нейрона скрытого слоя для  $m$ -й обучающей пары;  $F_k(\cdot)$  — функция активации  $k$ -го нейрона скрытого слоя;  $w_{ik}^{(1)}$  — весовой коэффициент для связи, идущей от  $i$ -го нейрона входного слоя к  $k$ -му нейрону скрытого слоя;  $u_{i,m}$  — значение сигнала, поступающего от  $i$ -го нейрона входного слоя для  $m$ -й обучающей пары,  $u_{i,m} = x_{i,m}$ ;  $x_0 = 1$  соответствует сигналу, формирующему пороговое значение.

В случаях, когда используется нелинейная функция активации (ФА) и для минимизации общей ошибки функционирования НС применяется метод градиентного спуска, для реализации процесса обучения существует важное требование. Оно предъявляется к самой функции активации. Необходимо, чтобы она была неубывающей и имела ограниченную производную. На практике это требование часто удовлетворяют посредством применения нелинейной ФА сигмоидного вида:

$$F(S) = \frac{1}{1 + e^{-S}}. \quad (7)$$

Производная такой функции имеет следующий вид:

$$F'(S) = F(S)(1 - F(S)). \quad (8)$$

Несложно определить, что производная максимальна при  $F(S) = 0,5$  и равна нулю при  $F(S) = 0$  или  $F(S) = 1$ . Дополнительное преимущество функции (7) состоит в автоматическом контроле степени усиления. Это значит, что для слабых сигналов (близких к нулю) кривая «вход—выход» имеет сильный наклон, который обеспечивает большое усиление. С увеличением сигнала степень усиления падает. Следовательно, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без особого ослабления.

Процесс обучения включает в себя два этапа вычислений: прямого и обратного распространения сигнала по нейросети.

На первом этапе происходит определение ошибки НС при распространении входного сигнала по сети в прямом направлении, т. е. от входа к выходу. Суть этого этапа заключается в следующем. Для заданного входного вектора вычисляют фактический выход НС с помощью выражений (5) и (6). Затем производят расчет ошибки между фактическим и требуемым выходом НС по формуле (2).

На втором этапе выполняют минимизацию полученной ошибки, которая сводится к последовательной настройке всех весовых коэффициентов НС. С этой целью вычисленный сигнал ошибки распространяется по сети в обратном направлении, т. е. от выхода к входу, и используется для подстройки весовых коэффициентов. Для определения величины, на которую необходимо скорректировать весовые коэффициенты, применяется метод градиентного спуска.

Вычисления повторяют до тех пор, пока общая ошибка функционирования НС не достигнет заданной величины  $\xi$ , устанавливающей точность процесса обучения.

Рассмотрим второй этап вычислений более подробно. Настройка весовых коэффициентов  $w_{kj}^{(2)}$  между  $k$ -м нейроном скрытого слоя и  $j$ -м нейроном выходного слоя производится на основе использования выражений (3) и (4). При этом

$$\Delta w_{kj}^{(2)} = -\eta \frac{\partial E_m}{\partial w_{kj}^{(2)}}, \quad (9)$$

где  $w_{kj}^{(2)}$  — коррекция весового коэффициента для связи, идущей от  $k$ -го нейрона скрытого слоя к  $j$ -му нейрону выходного слоя для  $m$ -й обучающей пары;  $\eta$  — коэффициент скорости обучения.

Используя правила частных производных, представим производную ошибки НС относительно выходных нейронов, принимая во внимание выражение (1):

$$\begin{aligned} \frac{\partial E_m}{\partial w_{kj}^{(2)}} &= \frac{1}{2} \sum_j \frac{\partial (g_{j,m} - y_{j,m})^2}{\partial w_{kj}^{(2)}} = \\ &= \frac{1}{2} \frac{\partial (g_{j,m} - y_{j,m})^2}{\partial w_{kj}^{(2)}} = -(g_{j,m} - y_{j,m}) \frac{\partial y_{j,m}}{\partial w_{kj}^{(2)}}. \end{aligned} \quad (10)$$

Принимая во внимание выражение (5), определим частные производные  $\frac{\partial y_{j,m}}{\partial w_{kj}^{(2)}}$ :

$$\begin{aligned} \frac{\partial y_{j,m}}{\partial w_{kj}^{(2)}} &= \frac{\partial F_j(S_{j,m})}{\partial S_{j,m}} \frac{\partial S_{j,m}}{\partial w_{kj}^{(2)}} = \\ &= F'_j(S_{j,m}) \frac{\partial}{\partial w_{kj}^{(2)}} \sum_k w_{kj}^{(2)} h_{k,m} = F'_j(S_{j,m}) h_{k,m}, \end{aligned} \quad (11)$$

где  $F'_j$  — производная ФА  $j$ -го нейрона выходного слоя.

С учетом (10) и (11), производную ошибки относительно выходных нейронов можно представить как

$$\frac{\partial E_m}{\partial w_{kj}^{(2)}} = -(g_{j,m} - y_{j,m}) F'_j(S_{j,m}) h_{k,m}. \quad (12)$$

Введем обозначение  $\delta_{j,m}^{(2)}$ :

$$\delta_{j,m}^{(2)} = (g_{j,m} - y_{j,m}) F'_j(S_{j,m}). \quad (13)$$

Переменная  $\delta_{j,m}^{(2)}$  в выражении (13) представляет собой ошибку  $j$ -го нейрона выходного слоя для  $m$ -й обучающей пары.

Теперь формулу (9) можно записать в сжатом виде:

$$\Delta w_{kj,m}^{(2)} = \eta \delta_{j,m}^{(2)} h_{k,m}. \quad (14)$$

Настройка весовых коэффициентов  $w_{ik}^{(1)}$  между  $i$ -м нейроном входного слоя и  $k$ -м нейроном скрытого слоя производится также на основе использования выражений (3) и (4):

$$\Delta w_{ik,m}^{(1)} = -\eta \frac{\partial E_m}{\partial w_{ik}^{(1)}}, \quad (15)$$

где  $\Delta w_{ik,m}^{(1)}$  — коррекция весового коэффициента для связи, идущей от  $i$ -го нейрона входного слоя к  $k$ -му нейрону скрытого слоя для  $m$ -ой обучающей пары;  $\eta$  — коэффициент скорости обучения.

Используя правила частных производных, представим производную ошибки НС относительно скрытых нейронов, учитывая выражение (1):

$$\frac{\partial E_m}{\partial w_{ik}^{(1)}} = \frac{1}{2} \sum_j \frac{\partial (g_{j,m} - y_{j,m})^2}{\partial w_{ik}^{(1)}} = \sum_j (g_{j,m} - y_{j,m}) \frac{\partial y_{j,m}}{\partial w_{ik}^{(1)}}. \quad (16)$$

Определим в формуле (16) частные производные  $\frac{\partial y_{j,m}}{\partial w_{ik}^{(1)}}$ :

$$\begin{aligned} \frac{\partial y_{j,m}}{\partial w_{ik}^{(1)}} &= \frac{\partial F_j(S_{j,m})}{\partial S_{j,m}} \frac{\partial S_{j,m}}{\partial w_{ik}^{(1)}} = \\ &= F'_j(S_{j,m}) \frac{\partial}{\partial w_{ik}^{(1)}} \sum_k w_{kj}^{(1)} u_{k,m} = F'_j(S_{j,m}) w_{kj}^{(2)} \frac{\partial h_{k,m}}{\partial w_{ik}^{(1)}}. \end{aligned} \quad (17)$$

Вычислим в формуле (17) оставшиеся частные производные  $\frac{\partial h_{k,m}}{\partial w_{ik}^{(1)}}$ :

$$\begin{aligned} \frac{\partial h_{k,m}}{\partial w_{ik}^{(1)}} &= \frac{\partial F_k(S_{k,m})}{\partial S_{k,m}} \frac{\partial S_{k,m}}{\partial w_{ik}^{(1)}} = \\ &= F'_k(S_{k,m}) \frac{\partial}{\partial w_{ik}^{(1)}} \sum_i w_{ik}^{(1)} x_{i,m} = F'_k(S_{k,m}) x_{i,m}, \end{aligned} \quad (18)$$

где  $F'_k(\ )$  — производная ФА  $k$ -го нейрона скрытого слоя.

Подставим результаты преобразований, полученные в выражениях (16), (17), (18), в формулу (15):

$$\Delta w_{ik,m}^{(1)} = \eta \sum_j (g_{j,m} - y_{j,m}) = F'_j(S_{j,m}) w_{kj}^{(2)} F'_k(S_{k,m}) x_{i,m}. \quad (19)$$

Упростим формулу (19) с учетом зависимости (13):

$$\Delta w_{ik,m}^{(1)} = \eta \{F'_k(S_{k,m}) \sum_j w_{kj}^{(2)} \delta_{j,m}^{(2)}\} x_{i,m}. \quad (20)$$

Введем новое обозначение  $\delta_{k,m}^{(1)}$ :

$$\delta_{k,m}^{(1)} = F'_k(S_{k,m}) \sum_j w_{kj}^{(2)} \delta_{j,m}^{(2)}. \quad (21)$$

Переменная  $\delta_{k,m}^{(1)}$  в соотношении (21) представляет собой ошибку  $k$ -го нейрона скрытого слоя для  $m$ -й обучающей пары.

Теперь формулу (20) можно представить в компактном виде:

$$\Delta w_{ik,m}^{(1)} = \eta \delta_{k,m}^{(1)} x_{i,m}. \quad (22)$$

Таким образом, формула (14) представляет собой дельта-правило, которое позволяет произвести настройку весовых коэффициентов между нейронами выходного и скрытого слоев, а формула (22) — между нейронами скрытого и входного слоев, соответственно.

Более подробное описание алгоритма обратного распространения ошибки для рассмотренного примера приводится ниже.

**Шаг 1. Инициализация.** Вводим обозначения и начальные данные:

$m$  — номер обучающего примера,  $1 \leq m \leq M$ , где  $M$  — количество примеров обучения;

$j$  — номер выходного нейрона,  $1 \leq j \leq J$ , где  $J$  — количество выходных нейронов;

$k$  — номер скрытого нейрона,  $0 \leq k \leq K$ , где  $K$  — количество скрытых нейронов;

$i$  — номер входного нейрона,  $0 \leq i \leq I$ , где  $I$  — количество входных нейронов;

$W^{(1)}$ ,  $W^{(2)}$  — матрицы весовых коэффициентов скрытого и выходного слоев, соответственно;

$X = (x_0, x_1, \dots, x_j, \dots, x_I)^T$  — вектор входного сигнала, где  $x_0 = 1$  соответствует сигналу, формирующему пороговое значение;

$H = (h_0, h_1, \dots, h_k, \dots, h_K)^T$  — вектор выходного сигнала скрытого слоя нейронов, где  $h_0 = 1$  соответствует сигналу, формирующему пороговое значение;

$Y = (y_1, y_2, \dots, y_j, \dots, y_J)^T$  — вектор выходного сигнала;

$T$  — символ транспонирования;

Всем весовым коэффициентам нейросети —  $w_{ik}^{(1)}$  и  $w_{kj}^{(2)}$  присваиваем небольшие начальные значения, которые выбираем случайным образом. Устанавливаем коэффициент скорости обучения  $\eta$  в пределах от 0,01 до 1 и задаем минимальное значение ошибки функционирования нейросети  $\xi$ .

**Шаг 2. Прямое распространение сигнала.** Вводим  $m$ -й обучающий пример и вычисляем для него фактическое значение выходного сигнала  $j$ -го нейрона:

$$y_{j,m} = F_j \left( \sum_{k=0}^K w_{kj}^{(2)} h_{k,m} \right), \quad h_{k,m} = F_k \left( \sum_{i=0}^I w_{ik}^{(1)} x_{i,m} \right).$$

**Шаг 3. Расчет ошибки нейросети.** Производим расчет ошибки между требуемым  $g_{j,m}$  и фактическим  $y_{j,m}$  значениями выходного сигнала  $j$ -го нейрона для  $m$ -го обучающего примера:

$$E_m = \frac{1}{2} \sum_{j=1}^J (g_{j,m} - y_{j,m})^2.$$

Затем вычисляем общую ошибку функционирования НС:

$$E = \sum_{m=1}^M E_m.$$

**Шаг 4. Проверка окончания обучения.** Если  $E \leq \xi$ , то вычисления завершаются. Переходим к шагу 6, в противном случае — переходим к шагу 5.

**Шаг 5. Обратное распространение сигнала.** Вычисляем ошибки нейронов выходного слоя и корректируем связи между выходным и скрытым слоями нейронов для  $m$ -го обучающего примера:

$$\begin{aligned} \delta_{j,m}^{(2)} &= (g_{j,m} - y_{j,m}) F'_j(S_{j,m}); \\ \Delta w_{kj,m}^{(2)}(t) &= \eta \delta_{j,m}^{(2)} h_{k,m}; \\ w_{kj,m}^{(2)}(t+1) &= w_{kj,m}^{(2)}(t) + \Delta w_{kj,m}^{(2)}(t). \end{aligned} \quad (23)$$

Вычисляем ошибки нейронов скрытого слоя и корректируем связи между скрытым и входным слоями нейронов для  $m$ -го обучающего примера:

$$\begin{aligned} \delta_{k,m}^{(1)} &= F'_k(S_{mk}) \sum_j w_{kj}^{(2)} \delta_{j,m}^{(2)}; \\ \Delta w_{ik,m}^{(1)}(t) &= \eta \delta_{k,m}^{(1)} x_{i,m}; \\ w_{ik,m}^{(1)}(t+1) &= w_{ik,m}^{(1)}(t) + \Delta w_{ik,m}^{(1)}(t), \end{aligned} \quad (24)$$

где  $t$  — шаг итерации;  $\Delta w_{kj,m}^{(2)}(t)$ ,  $\Delta w_{ik,m}^{(1)}(t)$  — коррекция весовых коэффициентов;  $w_{kj,m}^{(2)}(t)$ ,  $w_{ik,m}^{(1)}(t)$  — значения весовых коэффициентов до коррекции;  $w_{kj,m}^{(2)}(t+1)$ ,  $w_{ik,m}^{(1)}(t+1)$  — значения весовых коэффициентов после коррекции.

После вычислений согласно (23) и (24) переходим к шагу 2.

**Шаг 6. Завершение обучения.** Условие  $E \leq \xi$  выполнено. Процесс обучения завершен.

Представленный алгоритм, основу которого составляет метод градиентного спуска, имеет ряд проблем при его практическом применении.

1. Незнание выбора числа нейронных слоев и количества нейронов для каждого слоя.

2. Влияние случайной инициализации весовых коэффициентов НС на поиск минимума функции ошибки. Данная проблема отражает то, что при различной инициализации весовых коэффициентов НС могут получаться различные решения. Это характеризует неустойчивость алгоритма обучения, когда НС в одних случаях может обучаться до требуемой ошибки, а в других нет.

3. Сложность выбора подходящего коэффициента скорости обучения  $\eta$ . Так, слишком малое значение коэффициента  $\eta$  медленно изменяет весовые коэффициенты, что увеличивает время процесса обучения и приводит к скатыванию НС в локальный минимум. Большое значение  $\eta$  быстрее изменяет весовые коэффициенты. С одной стороны, это позволяет ускорить процесс обучения, с другой стороны — вызывает резкие скачки при перемещении по поверхности ошибок НС. Это может привести к пропуску глобального минимума, а в случае сложной формы поверхности ошибок — к выбору неправильного направления минимизации ошибки.

4. Невозможность определения точек локального и глобального минимумов, так как метод градиентного спуска их не различает.

Рассмотрим эти проблемы более подробно. Первая проблема, которая появляется при практическом применении МП, связана с его конфигурированием. Совершенно ясно, что размерность входного и выходного слоев нейронов определяется из условия задачи. Неясность возникает с выбором количества скрытых нейронных слоев и количества нейронов в каждом из них.

Ранее отмечалось, что НС с двумя слоями нейронных связей позволяет аппроксимировать любые выпуклые области в пространстве решений. При этом точность аппроксимации зависит от числа нейронов в скрытом слое. Чем больше число нейронов в скрытом слое, тем больше точность. Однако при слишком большой размерности скрытого слоя может наступить явление, называемое «переобучением» сети, что приводит к ухудшению обобщающей способности НС. При слишком малой размерности скрытого слоя нейросеть может либо попасть в нежелательный локальный минимум, либо существенно увеличить время процесса обучения.

Обеспечить требуемую точность и повысить обобщающую способность можно также с помощью увеличения количества скрытых слоев нейронов. Однако НС с несколькими скрытыми слоями обучаются еще медленнее.

Вместе с этим, существуют методы подбора конфигурации нейронной сети [6, 7, 9, 11, 12]. Применение таких методов приводит к минимизации количества весовых коэффициентов и нейронов в сети. Этим уменьшается сложность НС и повышается ее способность к обобщению. Однако решение по выбору окончательной конфигурации сети может быть принято только после полноценного обучения (с уменьшением ошибки функционирования до уровня, признаваемого удовлетворительным) различных ее вариантов.

Вторая проблема возникает при определении начальных значений (инициализации) весовых коэффициентов. К сожалению, не существует универсального метода выбора начальных значений весовых коэффициентов, который гарантировал бы нахождение наилучшей начальной точки для любой решаемой задачи. Поэтому без какой-либо априорной информации они часто выбираются случайным образом, согласно критерию минимальной энтропии [15, 16, 18]. При этом важен размер случайно инициализируемых весовых коэффициентов. Например, для сигмоидной ФА нейронов, если весовые коэффициенты будут иметь большие значения (положительные или отрицательные), выходная активность нейронов, учитывая формулы (8), (5), (6), будет близка к единице или нулю. В данном случае весовые коэффициенты будут изменяться незначительно. Это приведет к тому, что процесс обучения остановится в ближайшем от начальных значений локальном минимуме. В работах [15, 16, 18, 20] начальные значения весовых коэффициентов предлагается случайно выбирать в диапазонах  $[-0,05, 0,05]$ ,  $[-0,1, 0,1]$ ,  $[-0,3, 0,3]$ . В работе [8] рекомендуется случайно выбирать значения весовых коэффициентов следующим образом:

$$w_{ij} \approx \frac{1}{\sqrt{n(i)}}$$

где  $n(i)$  — число нейронов в слое  $i$ .

Не менее важная проблема, связанная с применением МП, возникает вследствие тенденции к стабилизации метода градиентного спуска в локальных минимумах. Метод градиентного спуска требует выполнения бесконечно малых шагов при перемещении по поверхности ошибок. Это неосуществимо на практике, так как выполнение данного условия ведет к бесконечно длительному времени процесса обучения МП. По этой причине в целях ускорения процесса обучения значение коэффициента скорости обучения  $\eta$  в определенной степени увеличивают.

Однако увеличение  $\eta$  вызывает резкие скачки при перемещении по поверхности ошибок НС. В свою очередь, перемещение с увеличенным коэффициентом скорости обучения может привести к пропуску глобального минимума, а в случае сложной формы поверхности ошибок — к выбору неправильного направления минимизации ошибки функционирования НС.

Для того чтобы сгладить резкие скачки в процессе коррекции весовых коэффициентов при перемещении по поверхности функции ошибок НС, правило обновления весовых коэффициентов дополняют значением коррекции весового коэффициента, выполненной на предыдущем шаге итерации:

$$\Delta w_{ij,m}^{(s)}(t) = \eta \delta_{j,m}^{(s)} u_{i,m}^{(s)} + \mu \Delta w_{ij,m}^{(s)}(t-1), \quad (25)$$

где  $s$  — номер слоя нейронных связей,  $1 \leq s \leq S$ ;  $t$  — шаг итерации;  $\Delta w_{ij,m}^{(s)}(t)$  — текущая коррекция весовых коэффициентов в  $s$ -м слое нейронных связей для  $m$ -й обучающей пары;  $\Delta w_{ij,m}^{(s)}(t-1)$  — коррекция весовых коэффициентов, выполненная на предыдущем шаге итерации;  $\delta_{j,m}^{(s)}$  — ошибка  $j$ -го нейрона в  $s$ -м слое нейронных связей для  $m$ -й обучающей пары;  $u_{i,m}^{(s)}$  — значение выходного сигнала нейрона в  $s$ -м слое;  $\eta$  — коэффициент скорости обучения;  $\mu$  — коэффициент момента.

В формуле (25) коэффициент  $\mu$  называют *моментным параметром*, а последнее слагаемое — *моментным термом* (momentum term) [8, 15]. Его значение выбирают в пределах от 0 до 1. Введение его позволяет выходить из небольших локальных минимумов.

Для обучения многослойной НС, содержащей более одного скрытого слоя нейронов, алгоритм обратного распространения ошибки требует определения производной ошибки НС относительно весовых коэффициентов всех слоев сети. Вычисления, связанные с этим процессом, становятся более сложными и менее эффективными в практическом применении. Поэтому на практике очень часто ограничиваются НС с двумя слоями нейронных связей.

Существуют адаптивные методы подбора коэффициента скорости обучения  $\eta$  [5, 10, 19], которые более прогрессивны по сравнению с постоянным значением. Однако они тоже не могут считаться наилучшим решением, особенно при моделирова-

нии динамических процессов. Учитывая эти особенности, выбор коэффициента  $\eta$  на практике чаще всего осуществляют опытным путем.

Таким образом, решить большинство проблем применения МП представляется возможным в процессе экспериментов — путем поиска компромисса между длительностью обучения и ошибкой функционирования НС. Тем не менее, возможности МП не позволяют в полной мере осуществить моделирование изменяющейся во времени последовательности векторов данных. В этой связи можно заключить, что МП имеет смысл использовать для кодирования псевдостационарных сегментов РС.

## Литература

1. **Минский М. Л., Пейперт С.** Перцептроны. — М.: Мир, 1971.
2. **Скурихин А. Н.** Нейронные сети: определения, концепции, применение. — М.: ЦНИИ управления экономики и информатики, 1991.
3. **Фланеган Дж.** Анализ, синтез и восприятие речи. — М.: Связь, 1968.
4. **Bishop C.** Neural Networks for Pattern Recognition. — Oxford University Press, 1995.
5. **Gill P., Murray W., Wright M.** Practical Optimization. — N. Y.: Academic Press, 1981.
6. **Haykin S.** Neural Networks: A comprehensive Foundation. — N. Y.: Macmillan College Publishing Company, 1994.
7. **Hassibi B., Stork D.** Second order derivatives for network pruning: Optimal brain surgeon // Advances in NIPS2 / Ed. D. Touretzky. — San Mateo: Morgan Kaufmann, 1993. — P. 164–171.
8. **Horz J., Krogh A., Palmer R.** Introduction to the theory of neural computation. — Addison Wesley Publishing Company, 1991.
9. **Hush D., Horne B.** Progress in supervised neural networks // IEEE Signal Processing Magazine, 1993, January. — P. 8–39.
10. **Klimauskas G.** Neural Ware — User manual. Natick, USA: Neural Ware Inc., 1992.
11. **Le Cun Y., Denker J., Solla S.** Optimal brain damage // Advances in NIPS2 / Ed. D. Touretzky. — San Mateo: Morgan Kaufmann, 1990. — P. 598–605.
12. **Li Q., Tufts D.** Synthesizing neural networks by sequential addition of hidden nodes // IEEE Proc. ICNN, Orlando, 1994. — P. 708–713.
13. **Lippmann R. P.** An Introduction to Computing with Neural Nets // IEEE Transactions ASSP Magazine, April 1987. — N 2. — L4. — P. 4–22.
14. **Rosenblatt F.** Principle of neurodynamics. — N. Y.: Spartan, 1992.
15. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning representation by Back-Propagating Errors // Nature, October 1986. — N 323 (9). — P. 533–536.
16. **Rumelhart D. E., McClelland J. L.** Parallel Distributed Processing. — Cambridge, MA: MIT Press, 1986.
17. **Sankar K. P., Sushmita M.** Multilayer Perceptron, Fuzzy Sets and Classification // IEEE Transactions on Neural Networks, 3(5). — September 1992. — P. 683–697.
18. **Thimm G., Fiesler E.** High Order and Multilayer Perceptron Initialization // IEEE Transactions on Neural Networks, 8(2). — 1997. — P. 1045–9227.
19. **Wasserman P. D.** Experiments in translating Chinese characters using backpropagation // Proceedings of the Thirty-Third IEEE Computer Society International Conference. — Washington, D. C.: Computer Society Press of the IEEE, 1988.
20. **Werbos P. J.** Backpropagation through time: what it does and how to do it // Proceedings of the IEEE, 1990, 78(10). — P. 1550–1560.