

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
"ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

В. Г. Спицын, Ю.Р. Цой

**ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ
НЕЙРОННЫХ СЕТЕЙ ДЛЯ
ОБРАБОТКИ ИНФОРМАЦИИ**

Методические указания к лабораторным работам

Издательство ТПУ
Томск 2007

УДК 681.3.016

С 72

Спицын В.Г., Цой Ю.Р.

С 72 Применение искусственных нейронных сетей для обработки информации: методические указания к лабораторным работам. – Томск: Изд-во ТПУ, 2007. – 31 с.

В методических указаниях к лабораторным работам обсуждаются проблемы применения нейронных сетей для обработки информации; содержатся методические указания и задания для выполнения лабораторных работ. Пособие подготовлено на кафедре вычислительной техники, соответствует программе дисциплины и предназначено для магистрантов направления “Информатика и вычислительная техника” по магистерской программе “Компьютерный анализ и интерпретация данных”.

УДК 681.3.016

Рекомендовано к печати Редакционно-издательским Советом
Томского политехнического университета

Рецензент

Доктор технических наук, профессор, заведующий
кафедрой автоматизированных систем управления ТУСУРа
А.М. Кориков

© Томский политехнический университет, 2007

© Оформление. Издательство ТПУ, 2007

© В.Г. Спицын, Ю.Р. Цой, 2007

ВВЕДЕНИЕ

В последние десятилетия в мире активно развивается новое направление исследований, специализирующееся на искусственных нейронных сетях (ИНС). Актуальность исследований в этом направлении подтверждается широким спектром областей применения ИНС. Примерами являются: автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функций, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие. С помощью ИНС можно, например, выполнять распознавание оптических или звуковых сигналов, создавать самообучающиеся системы, способные управлять автомашиной при парковке или синтезировать речь по тексту.

1. ФОРМАЛЬНЫЙ НЕЙРОН

Искусственные нейронные сети появились в результате применения математического аппарата к исследованию функционирования нервной системы [1]. Полученные при этом результаты успешно применяются при решении проблем распознавания образов, моделирования, прогнозирования, оптимизации и управления [1, 2].

Основной структурной и функциональной частью нейронной сети является **формальный нейрон (formal neuron)**, представленный на **рис. 1**, где x_0, x_1, \dots, x_n – компоненты вектора входных сигналов, w_0, w_1, \dots, w_n – значения **весов** входных сигналов нейрона, а y – выходной сигнал нейрона.

Формальный нейрон состоит из элементов 3 типов: умножителей (синапсов), сумматора и преобразователя. Синапс характеризует силу (вес) связи между двумя нейронами. Сумматор выполняет сложение входных сигналов, предварительно помноженных на соответствующие веса. Преобразователь реализует функцию одного аргумента – выхода сумматора. Эта функция называется **функцией активации** или **передаточной функцией нейрона**. Исходя из данного описания, математическая модель нейрона может быть представлена следующим образом:

$$y = f(S),$$
$$S = \sum_{i=1}^n w_i x_i + b.$$

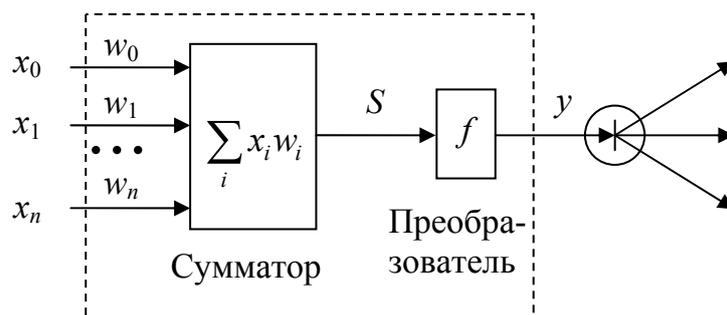


Рис. 1. Формальный нейрон (по [3])

Примеры некоторых активационных функций представлены в табл. 1 и на рис. 2.

Табл. 1. Функции активации нейронов

Название	Формула	Область значений
Пороговая	$f(S) = \begin{cases} 0, S < \Theta \\ 1, S \geq \Theta \end{cases}$	(0, 1)
Линейная	$f(S) = aS$	$(-\infty, +\infty)$
Лог-сигмоидная	$f(S) = \frac{1}{1 + e^{-aS}}$	(0, 1)
Гиперболический тангенс	$f(S) = \frac{e^{aS} - e^{-aS}}{e^{aS} + e^{-aS}}$	(-1, 1)

Описанный вычислительный элемент можно считать упрощенной математической моделью биологических нейронов. Чтобы подчеркнуть различие биологических и искусственных нейронов, вторые иногда называют **нейроподобными элементами** или **формальными нейронами**¹.

2. НЕЙРОННЫЕ СЕТИ

Описанные в п. 1 формальные нейроны можно объединять таким образом, что выходные сигналы одних нейронов являются входными

¹ Для этого в англоязычной литературе для обозначения нейрона в ИНС часто используют термин «node» – узел, вершина.

для других. Полученное множество связанных между собой нейронов называют **искусственными нейронными сетями (artificial neural networks, ANN)** или, коротко, **нейронными сетями**.

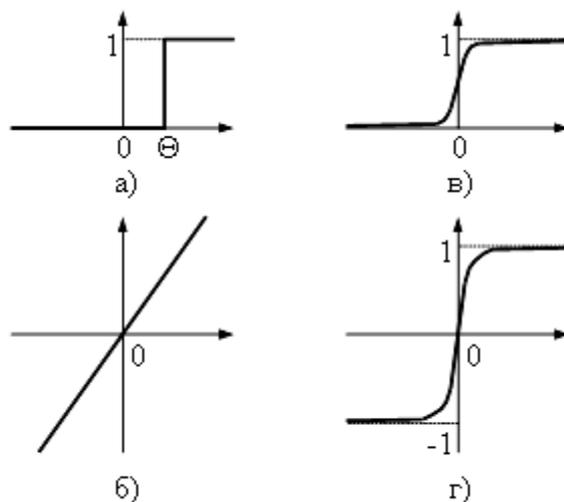


Рис. 2. Примеры активационных функций: а) пороговая; б) линейная; в) лог-сигмоидная; г) гиперболический тангенс

Пример нейронной сети с тремя входами и одним выходом представлен на рис. 3, нейроны обозначены кружками, стрелками показано направление распространения сигналов, веса межнейронных связей указаны рядом с соответствующими связями.

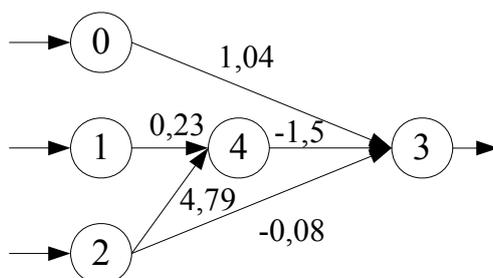


Рис. 3. Пример нейронной сети

Различают следующие три общих типа нейронов, в зависимости от их положения в нейронной сети:

- **входные нейроны (input nodes)**, на которые подаются входные для все сети сигналы. Такие нейроны имеют, как правило, один вход с единичным весом, смещение отсутствует, а значение выхода нейрона равно входному сигналу (нейроны

с индексами 0-2 на рис. 7.3);

- **выходные нейроны (output nodes)**, выходные значения которых представляют результирующие выходные сигналы нейронной сети (нейрон с индексом 3 на рис. 7.3);
- **скрытые нейроны (hidden nodes)**, не имеющие прямых связей с входными сигналами, при этом значения выходных сигналов скрытых нейронов не являются выходными сигналами ИНС (нейрон с индексом 4 на рис. 7.3).

Отметим, что структуру ИНС можно рассматривать как ориентированный граф, в котором узлы соответствуют нейронам, а ребра – межнейронным связям.

По структуре межнейронных связей различают два класса ИНС:

1. **ИНС прямого распространения (feed-forward ANNs)**, в которых сигнал распространяется только от входных нейронов к выходным. Орграф, соответствующий таким ИНС, не имеет циклов и петель. Примером ИНС прямого распространения является ИНС на рис. 3 и 4а.

2. **Рекуррентные ИНС (recurrent ANNs)** – ИНС с обратными связями. В таких ИНС сигналы могут передаваться между любыми нейронами, вне зависимости от их расположения в ИНС. Орграф, соответствующий структуре рекуррентных ИНС, может иметь петли и циклы (рис. 4б).

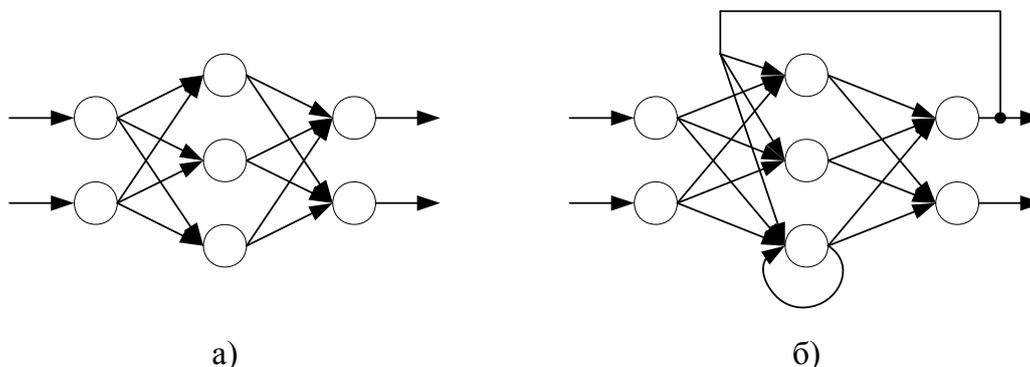


Рис. 4. Примеры структур нейронных сетей:

а) ИНС прямого распространения; б) рекуррентная ИНС

Среди различных структур ИНС наиболее известны **многослойные ИНС (multi-layered ANNs)** (рис. 5). Рассмотрим такие ИНС более подробно.

В многослойных сетях нейроны объединяются в слои таким образом, что нейроны одного слоя имеют одинаковые входные сигналы. Число нейронов в слое может быть произвольным и зависит в большей степени от решаемой задачи, чем от количества нейронов в других сло-

ях. Внешние (входные) сигналы подаются на нейроны входного слоя (его часто нумеруют как нулевой), а выходами сети являются выходные сигналы нейронов последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети может быть один или несколько скрытых слоев. Выходные сигналы нейронов слоя (q) являются входными сигналами следующего слоя ($q+1$). Одной из основных характеристик многослойных нейронных сетей является число слоев. В дальнейшем для описания структуры многослойных ИНС будем использовать количество присутствующих в ней скрытых слоев.

По структуре многослойные ИНС могут представлять как ИНС прямого распространения (рис. 4а и 5), так и рекуррентные (рис. 4б). Отметим также, что возможны многослойные сети, в которых существуют прямые связи между нейронами из несмежных слоев. Такие связи называют **перекрестными**. В данном пособии будут рассматриваться только многослойные нейронные сети без перекрестных и обратных связей.

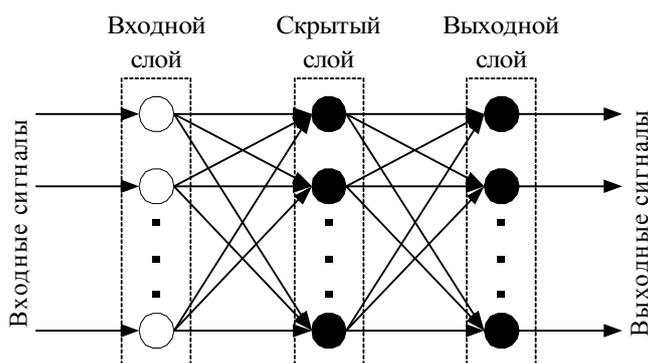


Рис. 5. Многослойная ИНС прямого распространения

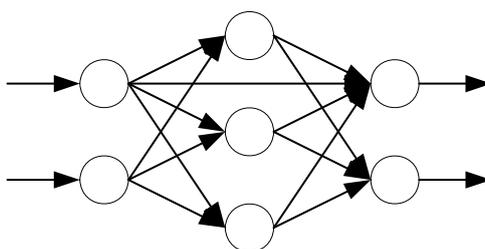


Рис. 6. Многослойная ИНС с перекрестной связью

Использование ИНС обладает следующими преимуществами:

1. Возможность решения трудноформализуемых задач, для которых трудно найти точный алгоритм решения (распознавание речи, рукописного текста). Отметим, что успешность приме-

нения ИНС существенно зависит от постановки задачи и исходных данных.

2. Массовый параллелизм в обработке информации. Данное преимущество позволяет реализовать нейросетевые алгоритмы и методы на параллельных вычислительных структурах, что особенно актуально в настоящее время в связи с распространением распределенных вычислений и массовым внедрением многоядерных центральных и графических процессоров для ПК, а также в связи с унификацией разнородных вычислений (научных, физических, графических и др.) на персональных компьютерах.
3. ИНС представляют единую концепцию для решения разнообразных задач, таких как задачи классификации, аппроксимации, моделирования, распознавания образов, принятия решений, обработки информации, кластеризации и др.
4. Возможность нестандартного решения известных задач, что расширяет и обогащает арсенал существующих средств и подходов, поскольку позволяет посмотреть на проблему и ее решение под «нестандартным» углом.

3. ОБУЧЕНИЕ ИНС

Для ИНС прямого распространения без скрытых слоев с n_I входами и n_O выходами зависимость выходных сигналов ИНС от входных можно представить следующим образом (будем считать, что одномерный вектор представлен как вектор-столбец):

$$\mathbf{Y} = G(\mathbf{X}) = F(\mathbf{W}\mathbf{X}),$$

где $\mathbf{X} = \{x_i : i = 1, \dots, n_I\}$ и $\mathbf{Y} = \{y_j : j = 1, \dots, n_O\}$ – векторы входных и выходных сигналов соответственно; $\mathbf{W} = \{w_{ij} : i = 1, \dots, n_I, j = 1, \dots, n_O\}$ – матрица весов межнейронных связей, в которой элемент w_{ij} соответствует весу связи от i -го входного нейрона к j -му выходному; $F(\cdot)$ – вектор-функция выходного сигнала слоя нейронов, вид которой зависит от выбранной активационной функции.

При добавлении скрытого слоя с n_H нейронами в структуру ИНС значение выражения для выходных сигналов ИНС изменится:

$$\mathbf{Y} = G(\mathbf{X}) = F(\mathbf{W}^{(1)}\mathbf{Y}^{(1)}) = F(\mathbf{W}^{(1)}F(\mathbf{W}^{(0)}\mathbf{X})),$$

где $\mathbf{W}^{(0)} = \{w_{ij}^{(0)} : i = 1, \dots, n_I, j = 1, \dots, n_H\}$ и $\mathbf{W}^{(1)} = \{w_{ij}^{(1)} : i = 1, \dots, n_H, j = 1, \dots, n_O\}$ – матрицы весов межнейронных связей для скрытого и выходного слоев

соответственно. Добавление дополнительных скрытых слоев соответствующим образом изменит вид функции выходного сигнала ИНС. Однако для выбранной структуры ИНС, при условии постоянного вектора входных сигналов X и фиксированной функции активации нейронов, значение выходного сигнала ИНС зависит только от значений весов связей.

Для решения практических задач важным является поиск такого набора значений весов межнейронных связей, при котором выходные сигналы ИНС изменяются в определенной зависимости от предъявляемого вектора входных сигналов. Процесс подстройки весов межнейронных связей называется **обучением нейронной сети**. От того, насколько качественно будет выполнено обучение, зависит способность нейронной сети решать поставленную задачу.

Существуют два общих подхода к обучению ИНС:

1. Обучение с учителем.
2. Обучение без учителя.

Обучение с учителем (supervised learning) подразумевает использование заранее сформированного множества обучающих примеров. Каждый пример содержит вектор входных сигналов и соответствующий вектор эталонных выходных сигналов, которые зависят от поставленной задачи. Данное множество называют **обучающей выборкой** или **обучающим множеством**. Обучение нейронной сети направлено на такое изменение весов связей ИНС, при котором значение выходных сигналов ИНС как можно меньше отличаются от требуемых значений выходных сигналов для данного вектора входных сигналов.

При **обучении без учителя (unsupervised learning)** подстройка весов связей производится либо в результате конкуренции между нейронами, либо с учетом корреляции выходных сигналов нейронов, между которыми существует связь. В случае обучения без учителя обучающая выборка не используется.

В дальнейшем будем рассматривать обучение с учителем, которое можно описать следующей последовательностью действий (практический пример представлен в п. 6):

1. Подготовка обучающей выборки.
2. Выбор структуры ИНС.
3. Настройка весов связей (обучение ИНС).

В процессе обучения на вход нейронной сети предъявляются данные из обучающей выборки и, в соответствии со значениями выходных сигналов, определяющих ошибку функционирования сети, производится коррекция весов связей. В результате обучения должна быть получена нейронная сеть, которая **без перенастройки весов связей** формирует

с требуемой погрешностью выходные сигналы \mathbf{Y} при подаче на вход сети любого набора входных сигналов из обучающего множества. Качество обученной нейронной сети проверяется с использованием данных, не участвовавших в процессе обучения.

Опишем каждый этап более подробно. Обучающая выборка состоит из множества пар векторов $(\mathbf{X}_i^{(T)}, \mathbf{Y}_i^{(T)})$, $i = 1, \dots, K$, где K – количество примеров в обучающей выборке; $\mathbf{X}_i^{(T)} = \{x_{i,j}^{(T)} : j = 1, \dots, n_I\}$ – вектор входных сигналов; $\mathbf{Y}_i^{(T)} = \{y_{i,j}^{(T)} : j = 1, \dots, n_O\}$ – вектор соответствующих $\mathbf{X}_i^{(T)}$ выходных сигналов.

Обучающая выборка формируется на основе уже известных данных по рассматриваемой проблеме. С одной стороны, чем больше и разнообразнее выборка, тем лучше будет результат обучения, но с другой – выборка может быть избыточной, что увеличивает время обучения. Отметим, что сформированная обучающая выборка может быть дополнительно обработана с использованием статистических и других методов для уменьшения мощности вектора входных сигналов путем удаления неинформативных и малоинформативных компонентов, которые не оказывают существенного влияния на результат обучения.

Выбор структуры нейронной сети оказывает влияние на характеристики функции выхода ИНС, т.к. структура ИНС определяет расположение и количество межнейронных связей и, соответственно, количество весов этих связей, которые необходимо настроить в результате обучения. Однозначной методики выбора количества скрытых слоев и нейронов в них нет, и вопрос о том, насколько успешным является тот или иной выбор, зачастую решается на основании экспериментальных результатов обучения и тестирования ИНС. Таким образом, структура сети выбирается разработчиком методом проб и ошибок, исходя из его личного опыта, а также, в ряде случаев, из характеристик обучающих данных. Отметим, что в большинстве случаев достаточно не более 2-3 скрытых слоев.

После того как определены обучающие данные и структура сети, производится настройка весов связей. Веса инициализируются случайными значениями, как правило, из диапазона $[-0,01; 0,01]$ – $[-0,1; 0,1]$, для того, чтобы впоследствии избежать возможного насыщения функций активации нейронов и повышенной чувствительности выхода ИНС к несущественным (в пределах погрешности) изменениям сигналов.

В процессе обучения происходит коррекция (подстройка) значений весов связей. При этом пары векторов $(\mathbf{X}_i^{(T)}, \mathbf{Y}_i^{(T)})$ из обучающего множества могут предъявляться многократно. Один «прогон» всех на-

боров данных из обучающей выборки вместе с коррекцией весов составляет **одну эпоху обучения**. Типичная длительность обучения может составлять от десятков до нескольких десятков тысяч эпох в зависимости от поставленной задачи, структуры ИНС, качества самих данных и выбранного алгоритма подстройки весов связей.

В результате обучения может возникнуть проблема **переобучения сети**. Дело в том, что обучающая выборка может содержать неточности, связанные, например, с погрешностями измерения, округлением или субъективностью оценок. Длительное обучение сети может привести к тому, что обученная сеть будет в процессе работы повторять эти неточности. Данная проблема возникает, если значение ошибки обучаемой сети близко к нулю. Поэтому часто обучение останавливают, когда ошибка достигает значения 0,01 – 0,001.

Рассмотрим распространенный алгоритм обратного распространения ошибки и его модификацию, использующую инерционность.

4. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

В настоящее время существует множество алгоритмов обучения. Наиболее известный из них – **алгоритм обратного распространения ошибки**. Данный алгоритм используется для минимизации отклонения реальных значений выходных сигналов нейронной сети от требуемых.

В качестве функции ошибки ИНС будем рассматривать следующую величину:

$$E(w) = \frac{1}{2} \sum_i E_i = \frac{1}{2} \sum_{i,k} (f_{i,k} - y_{i,k}^{(T)})^2, \quad (1)$$

где $f_{i,k}$ – значение выходного сигнала k -го выходного нейрона сети при подаче на её входы i -го набора обучающих данных, $y_{i,k}^{(T)}$ – требуемое значение выходного сигнала k -го выходного нейрона для i -го набора данных для обучения. Суммирование ведется по всем нейронам выходного слоя и по всем наборам данных из обучающей выборки. Обучение ИНС направлено на **минимизацию** функции $E(w)$.

Минимизация методом градиентного спуска обеспечивает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(q)} = \eta \frac{\partial E}{\partial w_{ij}}, \quad (2)$$

где $\Delta w_{ij}^{(q)}$ – величина изменения веса связи, соединяющей i -й нейрон $(q-1)$ слоя с j -м нейроном слоя q ; η – коэффициент скорости обучения,

$0 < \eta < 1$. Таким образом, вес связи изменяется пропорционально её вкладу в значение ошибки нейрона, для которого эта связь является входной, т.к. частная производная по весу $\frac{\partial E}{\partial w_{ij}}$ показывает зависимость скорости

изменения функции ошибки E от изменения этого веса.

Опустим преобразования формулы (2) и представим сразу конечный результат (3). Подробный вывод формул приведен, например, в [4] и [5]. Изменение веса связи определяется следующим образом:

$$\Delta w_{ij}^{(q)} = -\eta \delta_j x_i, \quad (3)$$

где δ_j – значение ошибки j -го нейрона в слое q , x_i – значение i -го входного сигнала для j -го нейрона слоя q . Данная формула применима и для настройки смещений нейронов, только вместо x_i необходимо подставить «1».

Отметим, что значение ошибки нейрона определяется в зависимости от его положения в сети.

Для нейронов выходного слоя

$$\delta_i = (f_{i,k}(S))' (f_{i,k} - y_{i,k}), \quad (4)$$

где $y_{i,k}$ – требуемое, а $f_{i,k}$ – фактическое значение выходного сигнала k -го нейрона для i -го набора данных из обучающей выборки, $(f_{i,k}(S))'$ – значение производной активационной функции k -го нейрона для i -го набора обучающих данных.

Если нейрон принадлежит одному из скрытых слоев, то

$$\delta_i^{(q)} = (f_i^{(q)}(S))' \sum_j w_{ij} \delta_j^{(q+1)}, \quad (5)$$

где $\delta_i^{(q)}$ – ошибка i -го нейрона в слое q , $\delta_j^{(q+1)}$ – ошибка j -го нейрона в $(q+1)$ слое, w_{ij} – вес связи, соединяющей эти нейроны, $(f_{i,k}(S))'$ – значение производной активационной функции i -го нейрона слоя q . Таким образом, значение ошибки нейрона пропорционально его «влиянию» на величины ошибок нейронов следующего слоя, а также скорости изменения его выходного сигнала для k -го набора обучающих данных.

Рассмотрим ИНС с нейронами с лог-сигмоидными функциями активации:

$$f(S) = \frac{1}{1 + e^{-aS}}, \quad (6)$$

где a – константа, S – взвешенная сумма входных сигналов нейрона, тогда

$$f'(S) = af(S)(1 - f(S)), \quad (7)$$

и формулы (4), (5) соответственно примут вид

$$\delta_i = af_{i,k}(1 - f_{i,k})(f_{i,k} - y_{i,k}), \quad (8)$$

$$\delta_i^{(q)} = af_i(1 - f_i) \sum_j w_{i,j} \delta_j^{(q+1)}. \quad (9)$$

Для реализации алгоритма обратного распространения ошибки может быть использована следующая последовательность действий:

1. Предъявление очередного набора из обучающей выборки на вход нейронной сети.
2. Вычисление выходного сигнала сети.
3. Определение величин ошибок нейронов выходного слоя по формуле (4) или (8).
4. Определение величин ошибок нейронов скрытых слоев по формулам (5) или (9).
5. Однократная коррекция весов связей.
6. Если в обучающей выборке есть неиспользованные в данной эпохе наборы данных, то переход на шаг 1.
7. Подсчет ошибки сети по формуле (1). Если ошибка меньше заданной, то конец обучения, иначе, начало новой эпохи обучения и переход на шаг 1.

Отметим, что алгоритм обратного распространения ошибки применим только для нейронных сетей, содержащих нейроны с дифференцируемой функцией активации. Т.е. рассмотренный алгоритм не подходит для настройки сетей, построенных на нейронах с пороговыми функциями активации. Для таких сетей применяются другие алгоритмы обучения, например дельта-правило Уидроу-Хоффа [2, 5]. Однако во многих случаях в нейронных сетях используются нейроны с сигмоидальными функциями активации (лог-сигмоидные функции и гиперболический тангенс), что дает возможность применять для настройки и обучения таких сетей градиентные алгоритмы.

В настоящее время создано немало модификаций алгоритма обратного распространения ошибки [5]. Одним из простых, но довольно эффективных является алгоритм, использующий инерционность обучения. В нем вес связи изменяется в соответствии со следующей формулой:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \delta_j x_i + \alpha \Delta w_{ij}(t-1),$$

где α – коэффициент инерционности от 0 до 1. Данный алгоритм сходится в среднем в 4–5 раз быстрее стандартного, но использует больший объем памяти.

5. РАБОТА НЕЙРОННОЙ СЕТИ

Для того чтобы понять возможности применения ИНС, необходимо более подробно изучить особенности функционирования отдельных нейронов и ИНС в целом.

Для начала рассмотрим один простейший нейрон с двумя входами и пороговой функцией активации (рис. 7) с порогом $\theta = 0$. Тогда выход нейрона y равен 1, если взвешенная сумма входных сигналов больше либо равна 0. В обратном случае выходной сигнал нейрона равен 0.

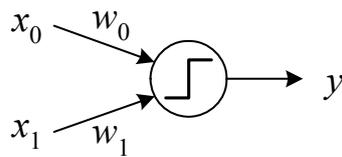


Рис. 7. Нейрон с двумя входами и пороговой функцией активации

Области на плоскости входных сигналов, в которых значение выхода нейрона постоянно, показаны на рис. 8.

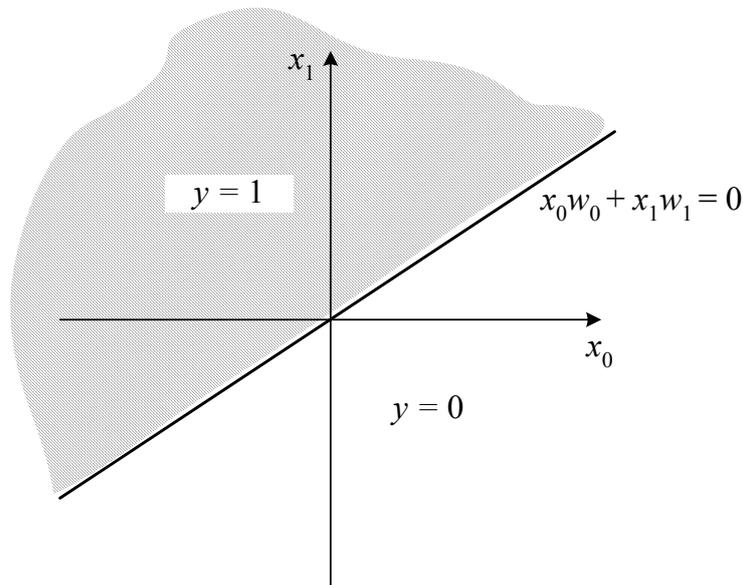


Рис. 8. Области с постоянным значением выходного сигнала для нейрона на рис. 7

Очевидно, что прямая, разделяющая эти области, имеет уравнение $x_0 w_0 + x_1 w_1 = 0$.

Изменение весов связей нейрона приведет к изменению наклона прямой относительно координатных осей, а добавление в уравнение по-

стоянного смещения вертикально смещает рассматриваемую прямую относительно оси Ox_1 . При этом уравнение разделяющей прямой примет вид:

$$x_0 w_0 + x_1 w_1 + b = 0,$$

Настройкой значений весов связей и смещения можно добиться требуемого положения разделяющей прямой, например, для реализации с помощью ИНС логической операции «ИЛИ» (рис. 9).

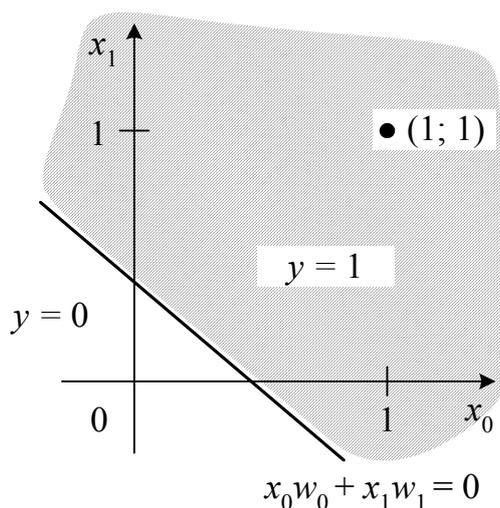


Рис. 9. Пример разделяющей поверхности для нейрона, реализующего логическую операцию «ИЛИ»

Естественно, нейрон может иметь больше двух входных сигналов и отличную от пороговой функцию активации. Однако в этом случае в силу многомерности пространства входных сигналов значительно усложняется только визуальное представление зависимости выходного сигнала нейрона от входного вектора. Принцип работы формального нейрона остается неизменным: существуют области пространства входных признаков, в которых выходной сигнал нейрона принимает определенное значение. При этом разделяющая прямая также становится многомерной (ее размерность равна $(n-1)$, где n – размерность вектора входных сигналов нейрона) и называется разделяющей гиперплоскостью.

Несколько нейронов, имеющие одинаковые входные сигналы, могут быть настроены (обучены) таким образом, чтобы реализовывать различные операции. Например, в случае использования двух нейронов с пороговой функцией активации, один из нейронов может реализовать логическую операцию «ИЛИ», а другой – инверсию «И». Разделяющие гиперплоскости таких нейронов для двумерного входного вектора показаны на рис. 10.

Тогда, в случае, если выходные сигналы этих нейронов будут по-

даны на вход нейрона, обученного логической операции «И», то такая нейронная сеть из трех нейронов будет реализовывать логическую операцию «Исключающее ИЛИ», которую невозможно получить с использованием одного нейрона, т.к. невозможно провести единственную разделяющую гиперплоскость (прямую, для случая двух входных сигналов) таким образом, чтобы отделить пару сигналов (0; 1) и (1; 0) от пары сигналов (0; 0) и (1; 1).

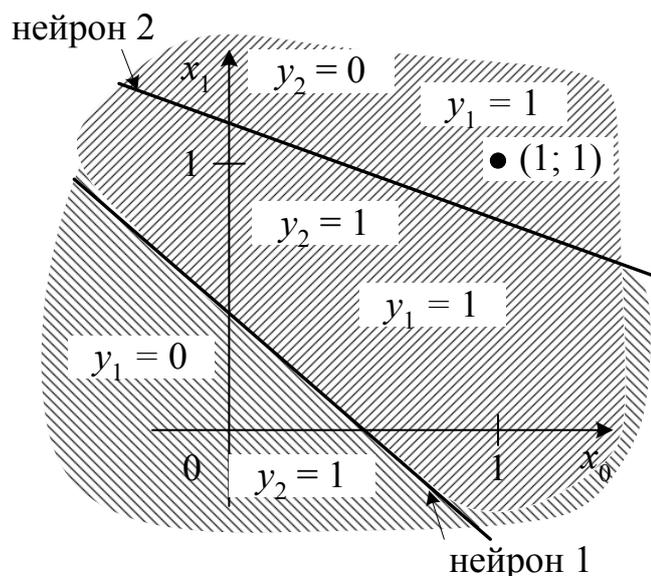


Рис. 10. Примеры разделяющих поверхностей для нейронов, реализующих логические операции «ИЛИ» (нейрон 1) и инверсия «И» (нейрон 2)

Таким образом, правильная (с точки зрения рассматриваемой задачи) настройка разделяющих гиперплоскостей, соответствующих нейронам, позволяет реализовать с помощью сети связанных нейронов достаточно сложные функции и зависимости, что актуально при решении практических задач. При этом каскадная (многослойная) организация структуры связей нейронов дает возможность выделять в пространстве входных сигналов области достаточно сложной формы (в том числе и невыпуклые, и многосвязные), причем, чем больше используется слоев, тем более сложной может быть форма подобласти (рис. 11). Так для ИНС с нейронами с пороговой функцией активации без скрытых слоев возможно лишь линейное разделение пространства входных сигналов; с одним скрытым слоем – выделение односвязных выпуклых областей; с двумя и более скрытыми слоями – выделение областей произвольной формы и связности.

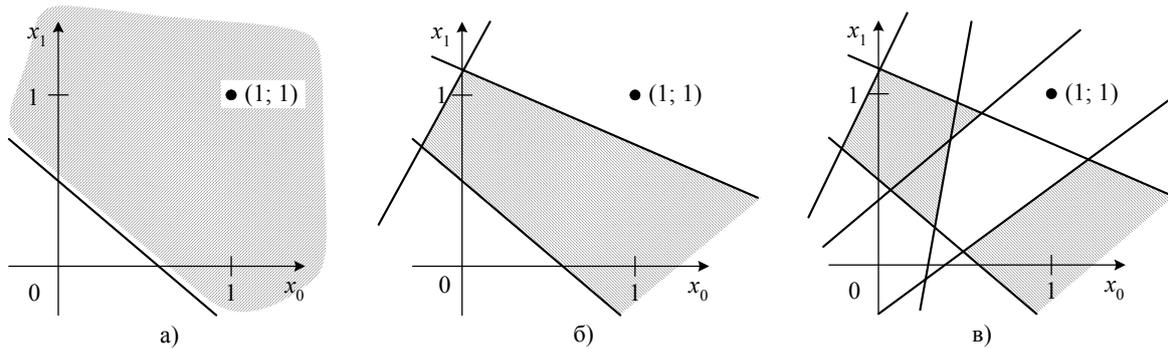


Рис. 11. Примеры зависимости формы выделяемых областей в пространстве входных сигналов от количества скрытых слоев в ИНС прямого распространения: а) нет скрытых слоев; б) 1 скрытый слой; в) 2 скрытых слоя

Использование функций активации, отличной от пороговой, например, сигмоидной, не приводит к значительному изменению характера разделяющей гиперплоскости. В этом случае в пространстве входного сигнала появляются области, соответствующие «переходным» значениям выхода нейрона между 0 и 1.

Таким образом, работа многослойной нейронной сети может быть представлена как глобальное взаимодействие распределенных нейронов, каждый из которых выполняет локальную задачу (активизируясь при определенных значениях входных сигналов). Обучение такой ИНС подразумевает решение задачи стохастической аппроксимации отображения входных сигналов ИНС в выходные часто без извлечения информации о характере и свойствах самого отображения.

Принципиально другой подход используется в ИНС с радиально-базисными функциями активации – **РБФ-сети**. Особенностью радиально-базисных функций является то, что они принимают ненулевые значения только в некоторой окрестности точки $\mathbf{c} = \{c_1, c_2, \dots, c_n\}$, называемой центром, где n -количество входных сигналов нейрона. Примером такой функции может являться многомерная функция Гаусса:

$$y = \exp\left(-\frac{\|\mathbf{c} - \mathbf{x}\|^2}{2\sigma^2}\right),$$

где $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ – вектор входных сигналов нейрона, $\|\mathbf{a}\|$ – длина вектора \mathbf{a} , σ – ширина радиальной функции. На **рис. 12** показан пример, иллюстрирующий отличия разделяющих поверхностей для нейронов с пороговой и радиальной функциями активации.

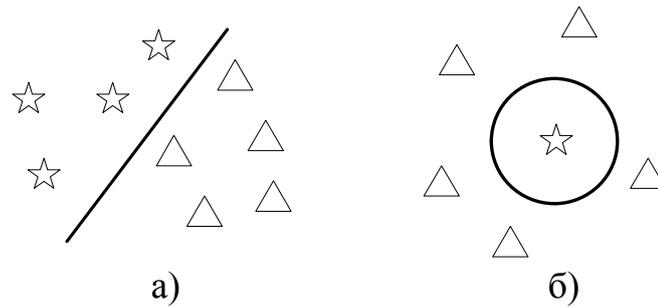


Рис. 12. Вид разделяющих поверхностей для нейронов с пороговой (а) и радиальной (б) функциями активации

Структура РБФ-сети включает один скрытый слой с радиальными функциями активации нейронов и выходной слой с единственным нейроном с линейной функцией активации. Зависимость выходных сигналов ИНС от входных представляется в виде разложения по отдельным базисным функциям (соответствующих активационным функциям нейронов), веса которых определяются как веса соответствующих связей выходного нейрона. Таким образом, обучение РБФ-сетей сводится к поиску таких векторов центров и значений ширины активационных функций РБФ-нейронов, при которых достигается аппроксимация зависимости выходных сигналов ИНС от входных с требуемой точностью.

6. ПРИМЕР РАБОТЫ И ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

Рассмотрим работу и обучение нейронной сети на примере. Дана сеть с 2 входами, двумя скрытыми слоями, в каждом из которых по 2 нейрона, и одним выходом. Функции активации нейронов – лог-сигмоидные (6), $a = 1$. Скорость обучения η примем равной 0,8. Необходимо реализовать с помощью ИНС логическую операцию «ИЛИ». Таблица истинности представлена в табл. 2, где x_1 и x_2 – входные переменные, а y – значение результата.

Табл. 2. Таблица истинности для операции «ИЛИ»

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Предположим, что на некотором этапе обучения получена сеть, соответствующая изображенной на рис. 6. Допустим также, что на неко-

тором шаге веса сети равны значениям из табл. 3. Смещения нейронов обозначены как веса с индексом «←».

Табл. 3. Веса связей ИНС

Номер слоя	Индекс нейрона	Индекс веса	Вес
1	0	-	0,02
1	0	0	0,12
1	0	1	0,35
1	1	-	-0,015
1	1	0	-0,5
1	1	1	0,24
2	0	-	-0,084
2	0	0	-0,33
2	0	1	0,27
2	1	-	0,037
2	1	0	-0,08
2	1	1	0,79
3	0	-	0,04
3	0	0	0,062
3	0	1	0,64

Роль обучающей выборки в рассматриваемом случае играют данные из табл. 2, в которой каждая строка соответствует одному набору данных. Пусть на вход нейронной сети поданы входные сигналы из 2-го обучающего набора ($x_1 = 0$, $x_2 = 1$). Тогда, в соответствии с таблицей истинности (табл. 2), выход сети должен равняться 1. Действительные значения выходов нейронов приведены в табл. 4. На рис. 13, поясняется вычисление выходных сигналов ИНС.

Табл. 4. Значения выходных сигналов нейронов

Номер слоя	Индекс нейрона	Значение выхода
1	0	0,591
1	1	0,556
2	0	0,468
2	1	0,606
3	0	0,612

Требуемое значение выхода сети «1», следовательно ошибка сети на данном этапе равна:

$$E_2 = 0,5 * (1 - 0,612)^2 = 0,075.$$

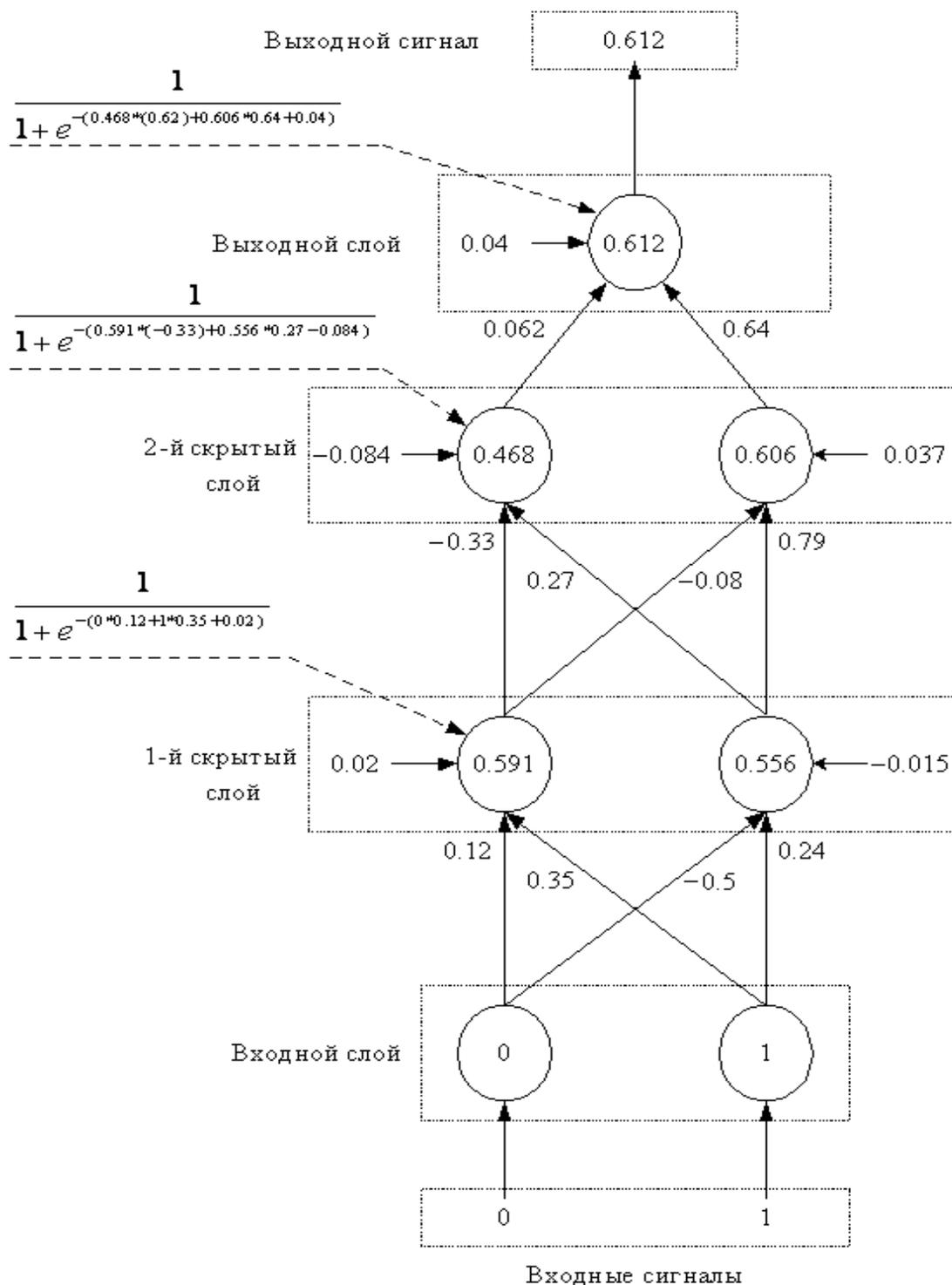


Рис. 13. Прямой проход для вычисления выхода ИНС

Ошибка выходного нейрона, согласно (7.5.8):

$$\delta_0^{(3)} = 0,612 * (1 - 0,612) * (0,612 - 1) = -0,092.$$

Ошибки нейронов 2 скрытого слоя по формуле (7.5.9):

$$\delta_0^{(2)} = 0,468 * (1 - 0,468) * (-0,092) * 0,062 = -0,001,$$

$$\delta_1^{(2)} = 0,606 * (1 - 0,606) * (-0,092) * 0,64 = -0,014.$$

Ошибки нейронов 1 скрытого слоя по формуле (7.5.9):

$$\delta_0^{(1)} = 0,591 * (1 - 0,591) * ((-0,001) * (-0,33) + (-0,014) * (-0,08)) = 0,0002,$$

$$\delta_1^{(1)} = 0,556 * (1 - 0,556) * ((-0,001) * 0,27 + (-0,014) * 0,79) = -0,003.$$

Вес первой связи выходного нейрона изменится, согласно (3), на следующую величину:

$$\Delta w_{00}^{(3)} = -0,8 \delta_0^{(3)} f_0^{(2)} = -0,8 * (-0,092) * 0,468 = 0,034.$$

Таким образом, новое значение веса будет равно:

$$w_{00}^{(3)}(t+1) = w_{00}^{(3)}(t) + \Delta w_{00}^{(3)} = 0,062 + 0,034 = 0,096.$$

Аналогично производится коррекция остальных весов связей ИНС:

$$w_{10}^{(3)}(t+1) = 0,64 - 0,8 * (-0,092) * 0,606 = 0,685;$$

$$w_{-0}^{(3)}(t+1) = 0,04 - 0,8 * (-0,092) * 1 = 0,114;$$

$$w_{00}^{(2)}(t+1) = -0,33 - 0,8 * (-0,001) * 0,591 = -0,329;$$

$$w_{10}^{(2)}(t+1) = 0,27 - 0,8 * (-0,001) * 0,556 = 0,27;$$

$$w_{-0}^{(2)}(t+1) = -0,084 - 0,8 * (-0,001) * 1 = -0,083;$$

$$w_{01}^{(2)}(t+1) = -0,08 - 0,8 * (-0,001) * 0,591 = -0,073;$$

$$w_{11}^{(2)}(t+1) = 0,79 - 0,8 * (-0,014) * 0,556 = 0,796;$$

$$w_{-1}^{(2)}(t+1) = 0,037 - 0,8 * (-0,014) * 1 = 0,048;$$

$$w_{00}^{(1)}(t+1) = 0,12 - 0,8 * 0,0002 * 0 = 0,12;$$

$$w_{10}^{(1)}(t+1) = 0,35 - 0,8 * 0,0002 * 1 = 0,35;$$

$$w_{-0}^{(1)}(t+1) = 0,02 - 0,8 * 0,0002 * 1 = 0,02;$$

$$w_{01}^{(1)}(t+1) = -0,5 - 0,8 * (-0,003) * 0 = -0,5;$$

$$w_{11}^{(1)}(t+1) = 0,24 - 0,8 * (-0,003) * 1 = 0,242;$$

$$w_{-1}^{(1)}(t+1) = -0,015 - 0,8 * (-0,003) * 1 = -0,013.$$

В случаях, когда вместо индекса начального нейрона связи стоит

символ « \leftarrow », производится настройка смещения нейрона. Т.е. запись $w_{-1}^{(2)}(t+1)$ означает новое значение смещения у нейрона с индексом 1 во втором скрытом слое. Используемые для расчета значения ошибок выходов нейронов показаны на рис. 14.

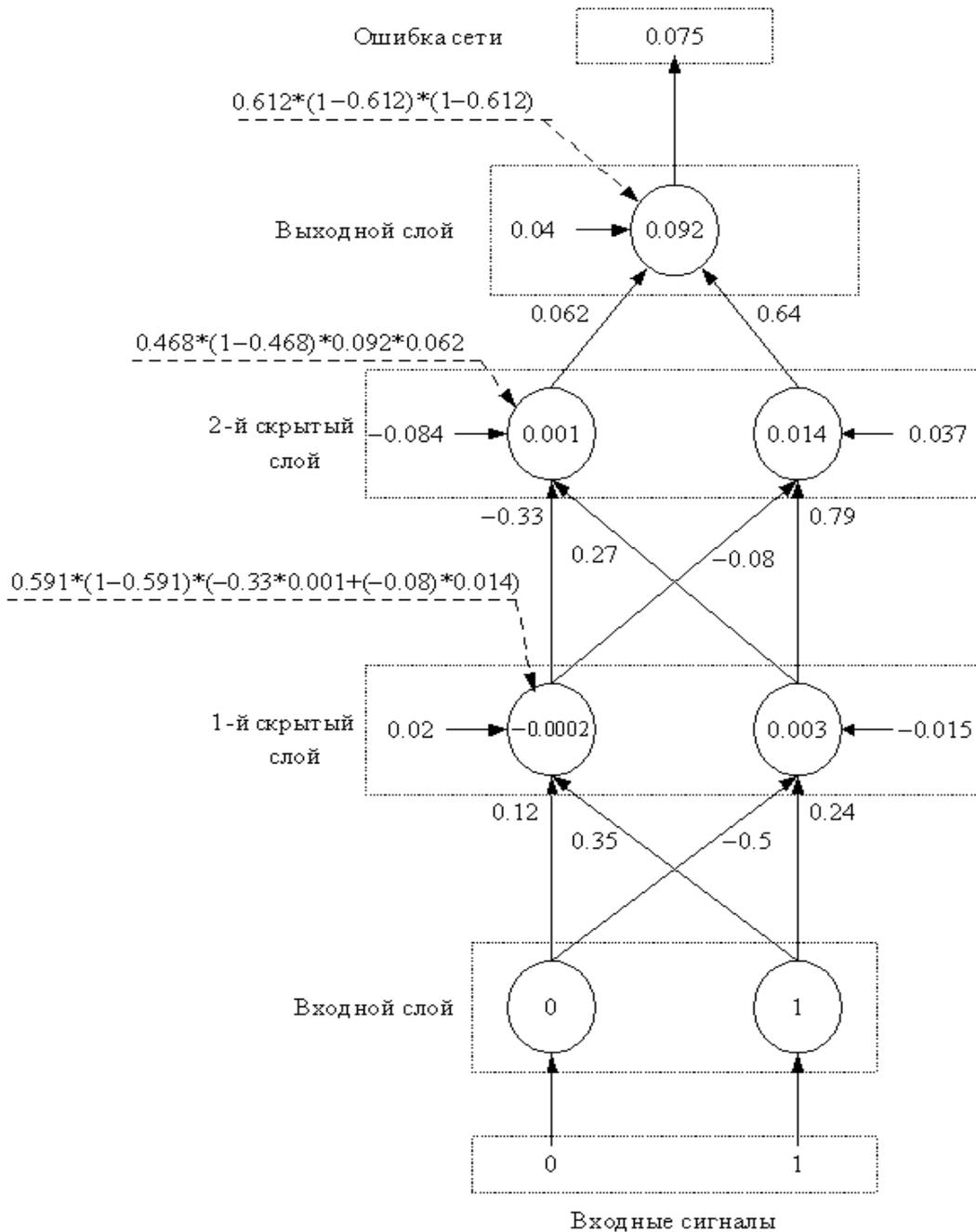


Рис. 14. Обратный проход для вычисления ошибок нейронов

Значения выходных сигналов ИНС при повторном подсчете вы-

ходов нейронов сети при использованном для коррекции весов наборе обучающих данных представлены в **табл. 5** и на **рис. 15**.

Табл. 5. Значения выходных сигналов нейронов после корректировки весов связей ИНС

Номер слоя	Индекс нейрона	Значение выхода
1	0	0,591
1	1	0,557
2	0	0,468
2	1	0,610
3	0	0,640

Новое значение ошибки ИНС для 2-го набора обучающих данных равно:

$$E_2 = 0,5 (1 - 0,641)^2 = 0,065.$$

Напомним, что значение ошибки E_2 до коррекции весов равнялось 0,075. Таким образом, в результате одного шага обучения значение ошибки уменьшилось на 0,01.

Для дальнейшего обучения ИНС на вход подается следующий вектор данных из обучающего множества и производится коррекция весов связей с учетом расхождения выходного сигнала ИНС и вектора требуемых выходных сигналов, соответствующего входному. Обучение ИНС продолжается до тех пор, пока ошибка выхода ИНС, вычисленная для **всех** наборов данных из обучающей выборки, по значению не будет меньше требуемой.

7. ОБЩИЕ РЕКОМЕНДАЦИИ К ПРОГРАММНОЙ РЕАЛИЗАЦИИ ИНС

Рассмотрим вопросы, связанные с программной реализацией многослойной нейронной сети. При написании программы необходимо знать следующие параметры сети:

- количество входов и выходов сети;
- количество скрытых слоев, а также количество и тип нейронов в них.

Т.к. нейроны входного слоя, как правило, не выполняют никаких функций, то их реализация необязательна. Т.е. достаточно считать, что

входные сигналы сети совпадают с входными сигналами нейронов первого скрытого слоя, либо, если скрытых слоев нет, со входными сигналами нейронов выходного слоя. Кроме этого, для сети прямого распространения без обратных связей верно следующее:

- все нейроны одного слоя имеют одинаковые входные сигналы;
- входные сигналы слоя $(q+1)$ являются выходными сигналами слоя q .

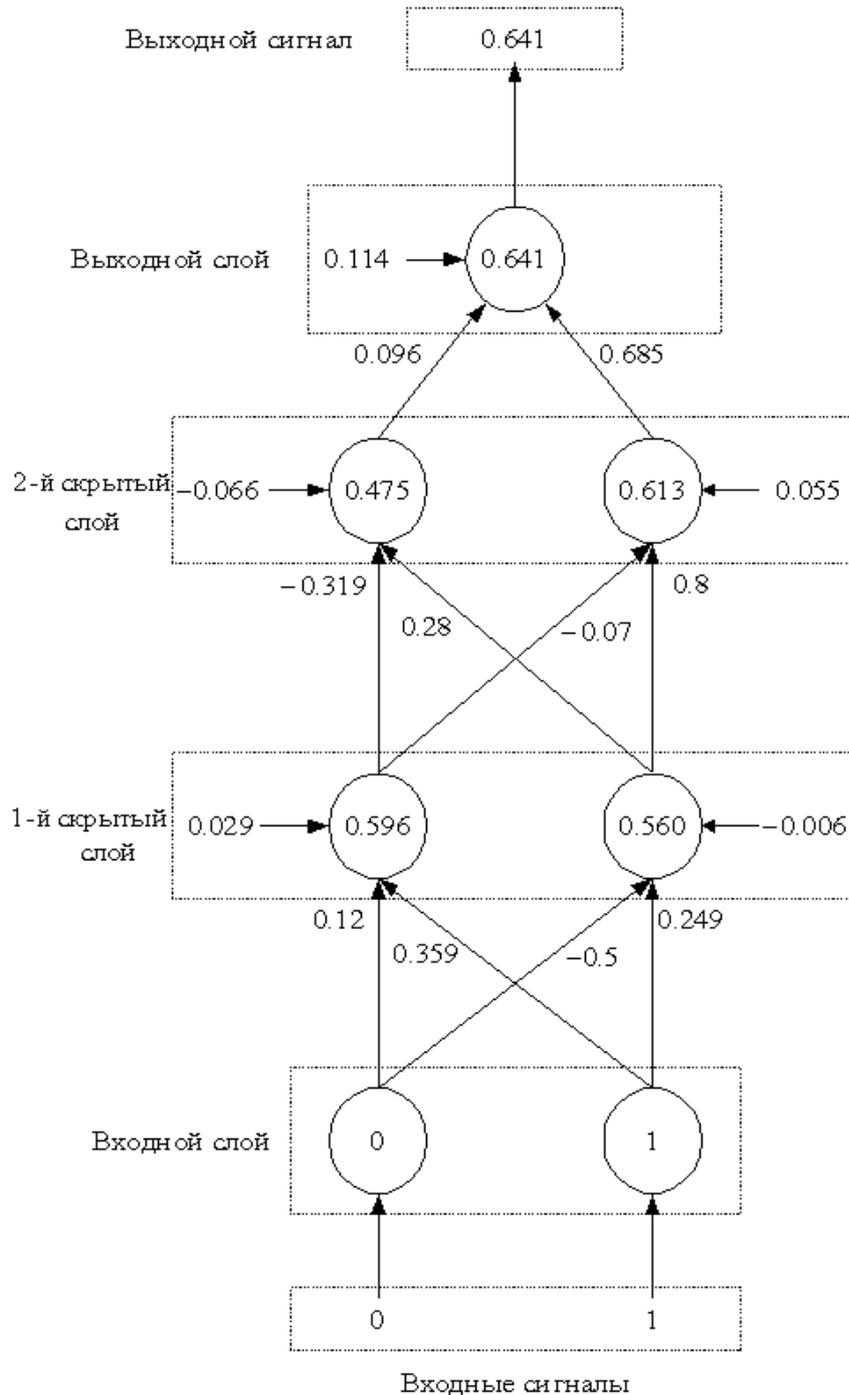


Рис. 15. Прямой проход после коррекции весов

Таким образом, каждый нейрон достаточно описать следующими параметрами:

- веса входящих связей (от нейронов предыдущего слоя к данному);
- смещение;
- коэффициент a в активационной функции;
- указатель на массив входных сигналов;
- указатель/ссылка на элемент массива выходных сигналов слоя, которому принадлежит данный нейрон.

Нейронный слой можно считать нейронной сетью без скрытых слоев, поэтому важнейшими характеристиками для него будут:

- число входов – равно числу нейронов в предыдущем слое;
- число выходов – равно числу нейронов в данном слое.

В многослойной сети два смежных слоя (q) и ($q+1$) имеют один общий массив, который будет содержать выходные сигналы слоя (q) и входные сигналы слоя ($q+1$). Поэтому целесообразно реализовать этот массив только один раз: либо как массив выходных сигналов слоя (q) (рис. 9), либо как массив входных сигналов слоя ($q+1$) (рис. 16, 17).

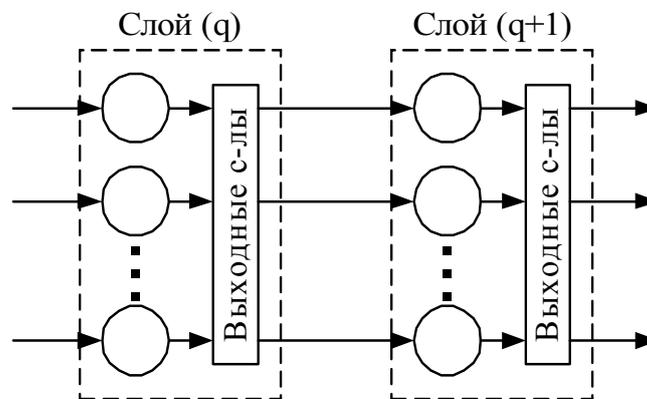


Рис. 16

В случае программной реализации искусственной нейронной сети с использованием объектно-ориентированного подхода можно выделить следующие объекты:

- нейрон;
- нейронный слой;
- нейронная сеть.

При этом класс каждого следующего объекта можно считать про-

изводным от класса предыдущего.

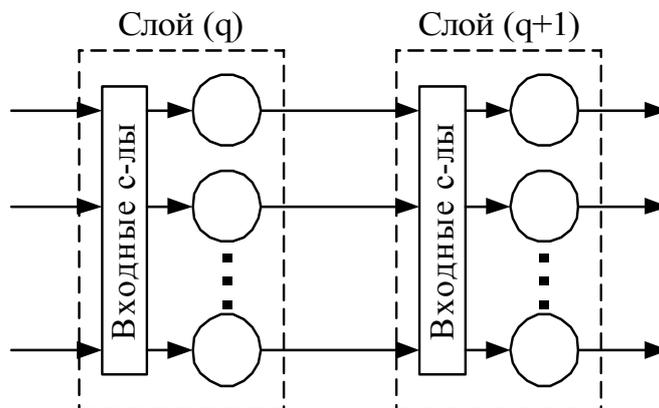


Рис. 17

При структурном подходе нейронную сеть можно, например, представить как трехмерный массив, первый индекс которого обозначает слой, второй – нейрон в этом слое, а третий – вес соответствующей связи этого нейрона. Смещения нейронов могут быть заданы в этом же массиве, либо отдельно.

Например, рассмотренная выше сеть с двумя входами, одним выходом и двумя скрытыми слоями по два нейрона в каждом может быть представлена в виде массива следующим образом (на языке программирования C):

```
double ***net; // массив весов связей сети
net = new double**[3]; // создание указателей на слои
net[0] = new double*[2]; // создание указателей
// на нейроны 1-го слоя
net[1] = new double*[2]; // создание указателей
// на нейроны 2-го слоя
net[2] = new double*[1]; // создание указателей
// на нейроны 3-го слоя
net[0][0] = new double[2]; // создание массива весов
// для 1-го нейрона 1-го слоя
net[0][1] = new double[2]; // создание массива весов
// для 2-го нейрона 1-го слоя
net[1][0] = new double[2]; // создание массива весов
// для 1-го нейрона 2-го слоя
net[1][1] = new double[2]; // создание массива весов
// для 2-го нейрона 2-го слоя
net[2][0] = new double[2]; // создание массива весов
// для 1-го нейрона 3-го слоя
```

Чтение/запись весов осуществляется обращением к массиву следующим образом:

```
double temp = net[0][1][1]; // чтение значения веса 2-го
                             // сигнала 2-го нейрона 1-го
                             // слоя
net[1][0][1] = 0.12; // запись (изменение) значения веса
                     // 2-го сигнала 1-го нейрона 2-го
                     // слоя
```

Значения смещений можно задавать аналогичным образом через двумерный массив, в котором первый индекс обозначает слой, а второй – нейрон, к которому относится это смещение.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

Целью лабораторной работы является создание студентом программы, реализующей ИНС для решения поставленной задачи.

Отчет по лабораторной работе должен содержать:

1. Цель работы.
2. Постановку задачи.
3. Метод решения задачи.
4. Структурную схему алгоритма.
5. Листинг программы.
6. Результаты работы алгоритма.
7. Выводы.

Для создания программы рекомендуется использование языков программирования: C ++, JAVA, Delphi, C#.

Ниже излагаются варианты заданий для выполнения лабораторных работ. Выбор варианта производится в соответствии с желанием студента, на основании его знаний о предметной области.

9. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

1. С помощью нейронной сети необходимо перекодировать прописные буквы в строчные (маленькие – в большие). На вход сети подается код «маленькой» буквы, с выхода «снимается» код соответствующей «большой» буквы.
2. Перевод нот из одной тональности в другую называется транспо-

нированием. С помощью нейронной сети транспонируйте ноты на один тон выше. На вход сети подается код ноты, с выхода «снимается» код ноты на тон выше, октаву учитывать не нужно.

3. Реализуйте с помощью нейронной сети преобразование градусов в радианы.
4. Реализуйте с помощью нейронной сети конвертер валют из долларов в евро.
5. Имеется сеть с двумя входами, двумя выходами и некоторым количеством скрытых нейронов. Необходимо настроить сеть таким образом, чтобы сигналы со входа менялись на выходе сети местами. Т.е., если на вход поступили числа 0,75 и 0,34, то на выходе должны быть числа 0,34 и 0,75.
6. Научите нейронную сеть осуществлять операцию сложения двух чисел.
7. С помощью нейронной сети реализуйте определение знака зодиака по числу и месяцу. Знак зодиака определяется по величине сигнала выходного нейрона сети.
8. Дан набор точек (табл. 6).
Аппроксимируйте данную зависимость полиномом второй степени с помощью нейронной сети. В отчете необходимо представить график с изображением исходных точек и кривой, полученной с помощью нейронной сети.

Табл. 6

X	Y
0	6.45
1	4.06
2	2.53
3	2.05
4	2.48
5	3.97
6	6.57
7	9.94
8	14.45

9. Реализуйте с помощью нейронной сети операцию умножения трех чисел из диапазона $[0, 1]$.
10. На основании данных из табл. 7 продолжите числовой ряд с помощью механизма предсказания на основе нейронной сети. Ряд может содержать отрицательные числа.

Табл. 7

a0	a1	A2	a3	a4	A5	a6	a7	a8	a9	a10
0.707	0.866	0.966	1	0.966	0.866	0.707	0.5	0.259	0	-0.259

Правильный ответ: $-0.5, -0.707, \dots$ (синусоида)

11. Создайте нейронную сеть, которая правильно классифицирует объекты, пользуясь данными из табл. 8.

Ответ: Если «Параметр 1» = 1, то 1-й класс, если «Параметр 2» = «Параметр 3», то 2-й класс, в противном случае – 3-й класс.

Табл. 8

Объект	Параметр 1	Параметр 2	Параметр 3	Класс
1	1	1	0	1
2	1	0	1	1
3	0	1	1	2
4	0	1	0	3
5	0	1	1	2
6	0	0	1	3
7	0	1	0	3
8	1	1	1	1
9	0	0	0	2

12. Имеется физическая система с переменными объемом и внутренним давлением. Необходимо создать нейросетевой регулятор, поддерживающий постоянную температуру внутри этой системы. На вход регулятора подается изменение давления и объема, на выходе – изменение температуры, компенсирующее действие изменяющихся параметров. Начальные условия: $V_0=15 \text{ дм}^3$, $P_0=100 \text{ Па}$, $T_0=280^\circ \text{ К}$. Расчет требуемого изменения температуры производится по формуле:

$$\Delta T = \frac{(P_0 \Delta V + V_0 \Delta P + \Delta V \Delta P) T_0}{P_0 V_0},$$

где ΔV – изменение объема, а ΔP – изменение давления.

13. Реализуйте с помощью нейронной сети сжатие бинарных изображений размером 16×16 пикселей с коэффициентом сжатия 2.
14. Научите нейронную сеть распознавать цифры от 0 до 9, заданные в матричном виде 5×7 (рис. 18).

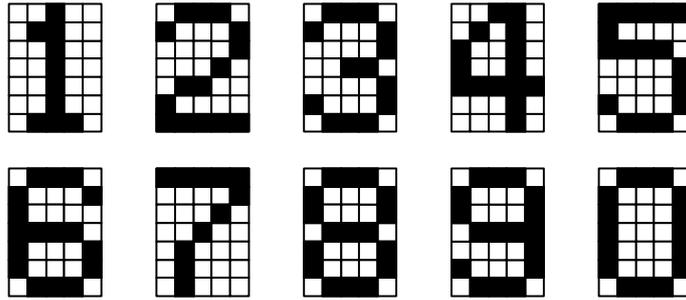


Рис. 18. Цифры для распознавания

ЛИТЕРАТУРА

1. McCulloch W.S., Pitts W.H. A logical calculus of ideas immanent in nervous activity // Bull. Math. Biophysics. – 1943. – Vol.5. – P. 115–119.
2. Хайкин С. Нейронные сети: полный курс, 2-е издание. М.: Издательский дом «Вильямс», 2006.
3. Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики.– 1998. – Т. 1, № 1. – С. 12-24.
4. Каллан Р. Основные концепции нейронных сетей: пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 288 с.
5. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. – 344 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ФОРМАЛЬНЫЙ НЕЙРОН	3
2. НЕЙРОННЫЕ СЕТИ.....	4
3. ОБУЧЕНИЕ ИНС	8
4. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ..	11
5. РАБОТА НЕЙРОННОЙ СЕТИ	14
6. ПРИМЕР РАБОТЫ И ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ	18
7. ОБЩИЕ РЕКОМЕНДАЦИИ К ПРОГРАММНОЙ РЕАЛИЗАЦИИ ИНС.....	23
8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ	27
9. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ	27
СПИСОК ЛИТЕРАТУРЫ.....	30

**Владимир Григорьевич Спицын
Юрий Робертович Цой**

**ПРЕДСТАВЛЕНИЕ ЗНАНИЙ
В ИНФОРМАЦИОННЫХ СИСТЕМАХ**

Методические указания к лабораторным работам

Редактор
О.Н. Свинцова

Подписано к печати
Формат 60x84/16. Бумага офсетная.
Печать RISO. Усл. печ. л. 3,12. Уч.-изд. л. 2,98.
Тираж 120 экз. Заказ . Цена свободная.
Издательство ТПУ. 634050, Томск, пр. Ленина, 30.