

## 11. НЕЙРОННЫЕ СЕТИ

### 11.1. Распознавание образов

Распознавание образов — это научное направление, связанное с разработкой принципов и построением систем, предназначенных для определения принадлежности объекта к одному из классов объектов. Классы объектов могут быть заранее выделены (задача классификации) или необходимо их выявить в процессе решения задачи (задача кластеризации). Под объектами в распознавании образов понимают: различные предметы и явления, процессы и ситуации, сигналы и т.п. Помимо смыслового (семантического) отличия, вкладываемого в понятие образ, они отличаются друг от друга также по способу представления (синтаксису):

- в классических моделях образ обычно описывается набором признаков, каждый из которых характеризует определенное свойство объекта;
- в структурных моделях в качестве образа выступает некоторое высказывание, порождаемое грамматикой, характеризующей класс;
- в задачах обработки текста роль образа выполняет некоторая цепочка символов или шаблонное представление этой цепочки (например, регулярные выражения).

Задачи распознавания образов представляют собой, по существу, дискретные аналоги задач поиска оптимальных решений (дискретного программирования). К ним относится широкий класс задач, в которых по некоторой, обычно весьма разнородной, неполной, нечеткой, искаженной и косвенной информации требуется установить, обладают ли изучаемые объекты, ситуации или явления фиксированным конечным набором свойств, позволяющим отнести их к определенному классу.

Другой из важных областей применения теории распознавания образов является решение задач прогнозирования поведения объектов или развития ситуации. К задачам этого вида относятся задачи технической и медицинской диагностики, геологического прогнозирования, прогнозирования свойств химических соединений, сплавов и новых материалов, прогнозирования урожая и хода строительства крупных объектов, обнаружения лесных пожаров, управления производственными процессами и т.д.

Задача распознавания образов возникает и в системах искусственного интеллекта. Например, в понимании естественного языка, символьной обработке алгебраических выражений, экспертных системах и др. Вообще, любую задачу можно рассматривать как задачу распознавания образа. Основные типы задач распознавания образов приведены в табл. 11.1.

Таблица 11.1

**Типы задач распознавания образов**

Тип задачи	Примечания
Классификация	Отнесение предъявленного объекта (ситуации) по его формализованному описанию к одному из заданных классов
Кластеризация	Разбиение множества объектов (ситуаций) по их формализованным описаниям на систему непересекающихся подмножеств (классов)
Прогнозирование	Предсказание значений характеристик или поведения системы в будущем на основании предшествующих и текущих наблюдений
Аппроксимация функций	Поиск функции, наиболее близко соответствующей набору экспериментальных данных
Оптимизация	Поиск эффективного или оптимального решения целевой функции в условиях действия ограничений

Одним из наиболее эффективных и распространенных способов представления и решения перечисленных выше задач являются искусственные нейронные сети (ИНС). Основными отличительными особенностями ИНС от логических ИИС являются:

- Искусственные нейронные сети подобны структуре мозга. В логических ИИС используются абстрактные структуры (правила, фреймы, сценарии и т.д.), не имеющие аналогов в живой природе.
- Для решения задач в логических ИИС надо заранее сформулировать весь набор закономерностей, описывающих предметную область. В ИНС используется иной подход, который основан на концепции обучения на примерах.

В этом случае при построении ИИС не требуется заранее знать обо всех закономерностях исследуемой области, но необходимо располагать достаточным количеством примеров для настройки разрабатываемой системы, которая после обучения будет способна получать требуемые результаты с определенной степенью достоверности.

## 11.2. Краткий исторический обзор

Исследования в области ИНС пережили три периода активизации. Первый пик в 40-х годах обусловлен пионерской работой Мак-Каллоха и Питтса. Второй был в 60-х годах прошлого века благодаря перцептрон Розенблатта. В 1958 г. Фрэнк Розенблатт продемонстрировал компьютерную модель электронного устройства, названную им перцептроном, а в 1960 г. - первую действующую машину «Марк-1», моделирующую совместную работу человеческого глаза и мозга. Машина могла распознавать некоторые из букв, написанные на карточках, подносимых к кинокамерам («глазам» машины). Она различала буквы алфавита, но была чувствительна к их написанию.



Рис. 11.1. Фрэнк Розенблатт с датчиком изображения Mark-1

В своей работе (1969 г.) Минский и Пейперт указали на ограниченные возможности простейшего перцептрона. Результаты этой работы надолго погасили энтузиазм большинства исследователей в области ИНС. Возникшее в исследованиях по нейронным сетям затишье продлилось почти 15 лет.

С начала 80-х годов нейронные сети вновь привлекли интерес исследователей, что связано с энергетическим подходом Хопфилда (1982 г.) и алгоритмом обратного распространения ошибок для обучения многослойного перцептрона (многослойные сети прямого распространения), впервые предложенного Вербосом.

### 11.3. Биологический нейрон

Стандартный способ решения любой задачи на компьютере заключается в том, что составляется алгоритм, который реализуется в виде программы. Если мы хотим создать робота для игры в баскетбол, то составим дифференциальные уравнения полета мяча с разных позиций, внесем поправки на сопротивление воздуха, параллакс датчиков, а затем все это запрограммируем. В то же время вряд ли кто-нибудь предположит, что Шакил О'Нил так хорошо знает математику. Человеческий мозг, очевидно, решает эту задачу принципиально другим способом.

Нервная система и мозг человека состоят из нейронов, соединенных между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от нашей кожи, ушей и глаз к мозгу, процессы мышления и управления действиями – все это реализовано в живом организме как передача электрических импульсов между нейронами.

Рассмотрим строение биологического нейрона (рис. 11.2). Каждый нейрон имеет отростки нервных волокон двух типов – дендриты, по которым принимаются импульсы, и единственный аксон, по которому нейрон может передавать импульс. Аксон в конце разветвляется на волокна, контактирует с дендритами других нейронов через специальные образования – синапсы, которые влияют на силу импульса.

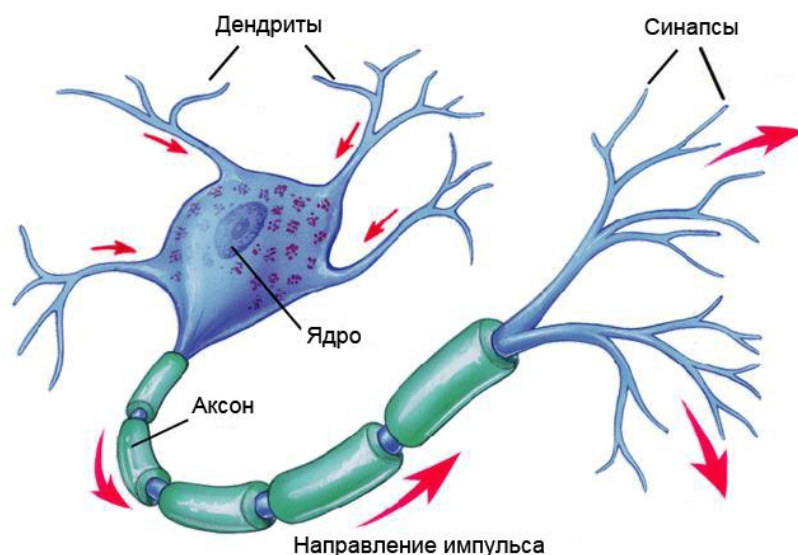


Рис. 11.2. Биологический нейрон

Можно считать, что при прохождении синапса сила импульса меняется в определенное число раз, которое мы будем называть весом синапса. Импульсы, поступившие к нейрону одновременно по нескольким дендритам, суммируются. Если суммарный импульс превышает некоторый порог, нейрон возбуждается, формирует собственный импульс и передает его далее по аксону. Важно отметить, что веса синапсов могут изменяться со временем, а значит, может меняться и поведение соответствующего нейрона.

Кора головного мозга человека является протяженной, образованной нейронами поверхностью толщиной от 2 до 3 мм с площадью около  $2200 \text{ см}^2$ , что вдвое превышает площадь поверхности стандартной клавиатуры. Общее количество нейронов в мозге человека около  $10^{11}$ , что приблизительно равно числу звезд Млечного пути. Каждый нейрон связан с  $10^3 - 10^4$  другими нейронами. В целом мозг человека содержит приблизительно от  $10^{14}$  до  $10^{15}$  взаимосвязей.

## 11.4. Математическая модель нейрона

Обобщенная математическая модель нейрона выглядит следующим образом [2,3]:

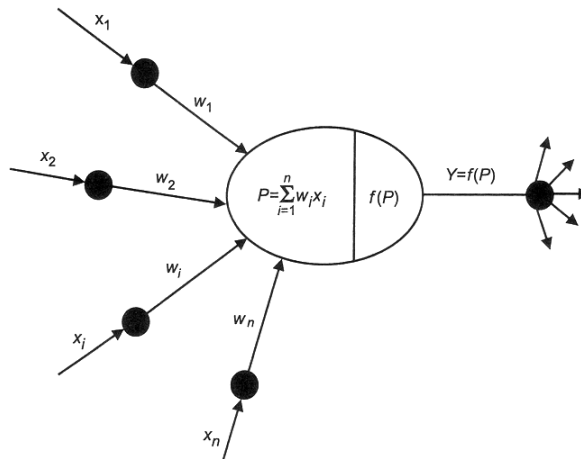


Рис. 11.3. Математическая модель нейрона

- входные сигналы  $x_i$  – данные, поступающие из окружающей среды или от других активных нейронов. Входные значения могут быть дискретными из множеств  $[0, 1]$  или  $[-1, 1]$  либо принимать любые вещественные значения.
- весовые коэффициенты  $w_i$  – определяют силу связи между нейронами;
- уровень активации (потенциал) нейрона  $P = \sum w_i x_i$ ;
- функция активации  $Y = f(P)$  – используют для вычисления выходного значения сигнала, передаваемого другим нейронам.

Входные сигналы  $x_i$  взвешиваются (умножаются на коэффициенты  $w_i$ , называемые синаптическими весами), и полученная взвешенная сумма  $P = \sum w_i x_i$  подвергается изменению функцией  $f(P)$ , называемой функцией активации. Выходной сигнал  $Y$  также может подвергаться взвешиванию (масштабированию). В качестве функции активации используют разные функции, но чаще сигмоидную функцию  $Y = 1 / (1 + \exp(-\lambda P))$ , а также гиперболический тангенс, логарифмическая функция, линейная и другие. Основное требование к таким функциям – монотонность.

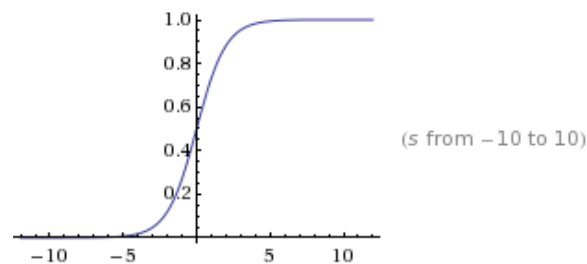


Рис. 11.4. Функция логистического сигмоида

Одна из причин, по которой сигмоид используется в нейронных сетях, это простое выражение его производной через саму функцию  $S'(x) = S(x) * (1 - S(x))$ , что позволяет существенно сократить вычислительную сложность метода обратного распространения ошибки, сделав его применимым на практике.

Интеллект одиночного нейрона невысок. Можно считать, что он реализует простую регрессионную модель для  $N$  независимых переменных. Если объединить множество нейронов, в сетевые структуры, то и реализуемая функция может быть сколь угодно сложной. Нейронная сеть – это совокупность вычислительных элементов (нейронов), каждый из которых имеет несколько входов-синапсов и один выход-аксон.

## 11.5. Обучение нейронной сети

Аналогия с мозгом не заканчивается на структуре нейрона и их сети. Из природы позаимствована также идея обучения нейронных сетей [4]. Известно, что человеческий мозг способен к самообучению, причем достигает успехов зачастую, не зная природы процессов, лежащих в основе выполняемых действий. Например, чтобы попасть мячом в баскетбольное кольцо, робот должен измерить расстояние до кольца и направление, рассчитать траекторию, и совершить бросок с учетом массы мяча и сопротивления воздуха. Человек же обходится без этого только через тренировки. Многократно совершая броски и наблюдая результаты, он корректирует свои действия, постепенно совершенствуя свою технику. В его мозгу формируются соответствующие структуры нейронов, отвечающие за технику бросков. Таким образом, непременным атрибутом обучения является многократное повторение и возможность немедленной оценки полученного результата. Для нейронных сетей это процесс имеет вид (рис. 11.5):

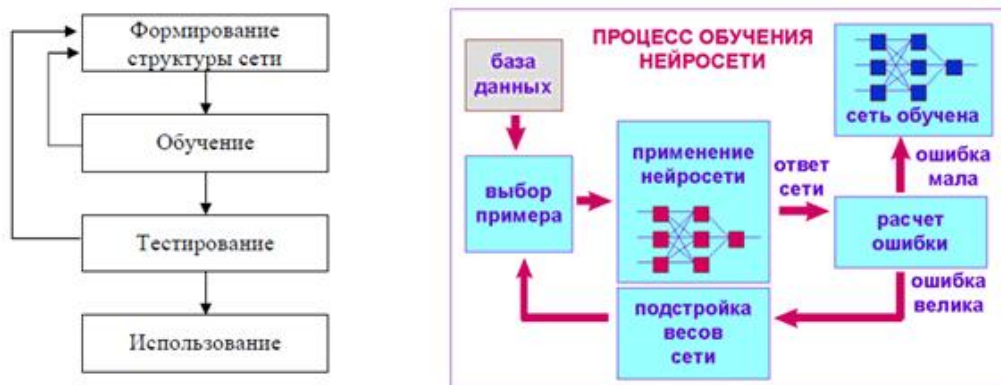


Рис. 11.5. Процесс создания и использования нейронной сети

Выбор структуры ИНС – это отдельная задача, которая включает в себя выбор топологии сети и функций активации каждого нейрона. Вначале параметры нейронов устанавливаются произвольно. Обучение заключается в том, что на вход сети подаются тренировочные данные, то есть такие, выходной результат для которых известен. На выходе результаты сравниваются с ожидаемыми данными, и вычисляется значение ошибки. После этого выполняется коррекция параметров ИНС с целью минимизации функции ошибки. Если удовлетворительной точности достигнуть не удастся, то надо изменить структуру сети и повторить обучение на множестве тренировочных данных.

После того, как сеть обучена, выполняется тестирование – контроль точности на специальных тестовых данных. Это означает, что все данные следует разбить на два подмножества: на одном выполнить обучение сети, а на втором – тестирование. Это разбиение может быть случайным или регулярным. Отличие тестирования от обучения в том, что на тестовых данных только проверяется точность, а, поскольку эти данные не используются для подбора параметров сети, они могут служить критерием качества обучения. По аналогии с обучением человека тестирование можно уподобить экзамену.

## 11.6. Иллюстрация функционирования ИНС

Получив представление о нейроне, которые иногда называют персептронами, и о возможности построения сети на них, можно было бы переходить к более подробному знакомству с ИНС. Однако для того, чтобы было бы более понятно, о чем идет речь, рассмотрим ряд примеров по использованию и функционированию ИНС.

Дальнейшее изложение этого раздела и приводимые в тексте скриншоты основаны на материале обучающей программе EASYDEMO, которая разработана компанией

«НейроПроект» и находится (<http://www.neuroproject.ru/download.htm#dnwsgdemo>) в свободном доступе Интернет. Эта программа позволяет получить представление о том, как устроены нейронные сети и демонстрирует :

- каким образом нейронные сети используют данные,
- как выглядят нейронные сети
- и как они обучаются.

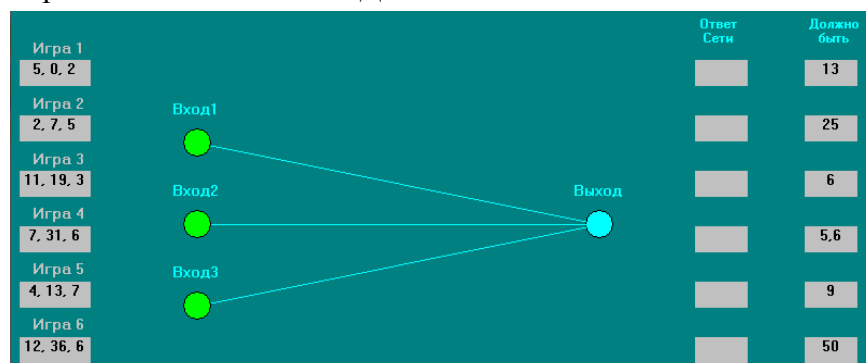
Одной из областей использования ИНС является прогнозирование на основе массива данных за предыдущий период. Эти данные могут быть представлены в виде таблицы, каждая строка которой представляет собой отдельный пример. В столбцах таблицы содержатся параметры или переменные, от которых зависит предсказание, а также переменная, которую пытаемся предсказать. Предположим, фирма занимается продажей подержанных автомобилей и желает прогнозировать их реальную стоимость в зависимости от года выпуска, величины пробега и технического состояния. В этом случае каждый пример состоит из четырех чисел: трех параметров автомобиля и цены, за которую удалось его продать.

	Год	Пробег	Состояние	Цена
Машина 1	1996	78000	5	13000
Машина 2	1970	245000	2	2500
Машина 3	1992	30000	3	6000
Машина 4	1980	56000	5	4500
Машина 5	1994	13500	4	9000
Машина 6	1990	64250	3	5000

Или, организовав продажу кваса на стадионе, коммерсант хочет прогнозировать количество раскупленных бутылок кваса в зависимости от числа проданных билетов (тыс. штук), температура воздуха и дня недели.

	Зрители	Температура	День недели	Квас
Игра 1	5	0	2	13
Игра 2	2	7	5	25
Игра 3	11	19	3	6
Игра 4	7	31	6	5,6
Игра 5	4	13	7	9
Игра 6	12	36	6	50

Для этих примеров хотелось бы построить модель для предсказания величины, стоящей в последнем столбце, в зависимости от значений в других столбцах. С этой целью, необходимо все данные разделить на входные и выходные. Первые определяют требуемое количество **ВХОДНЫХ НЕЙРОНОВ** – по одному для каждой входной переменной. Вторые количество **ВЫХОДНЫХ НЕЙРОНОВ**.

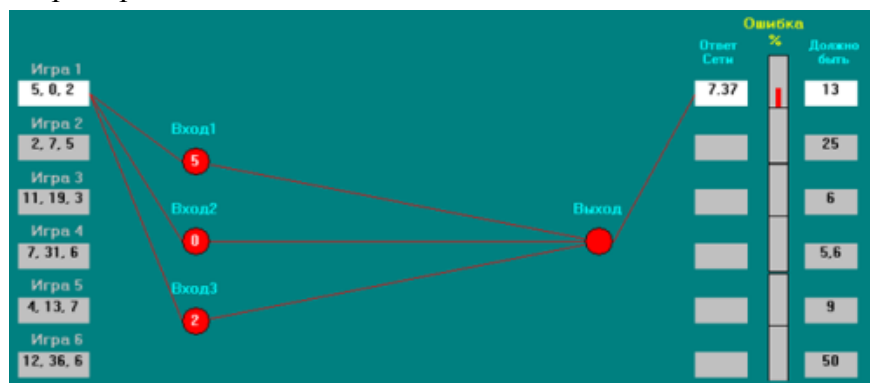


В рассматриваемом примере входных нейронов будет три, и они определяют входы по зрителям, температуре и дню недели. Выходной нейрон один, и определяет он

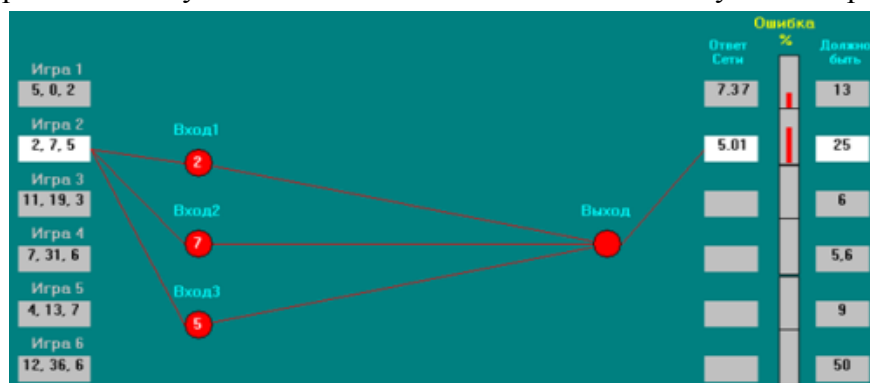
количество проданных бутылок. Если на входной нейрон подать значение входной переменной, нейрон станет АКТИВНЫМ, и сигнал активации распространится по всем его СВЯЗЯМ с другими нейронами. На данной стадии создания нейронной сети входные нейроны непосредственно связаны с выходным нейроном.

Каждая связь имеет свой собственный ВЕС, который определяет силу связи. В дальнейшем веса могут изменяться. Величина активации выходного нейрона является функцией суммы значений, поступающих на входные нейроны, умноженных на веса соответствующих связей с выходным нейроном. Мы хотим, чтобы выходное значение ИНС было как можно ближе к желаемому результату.

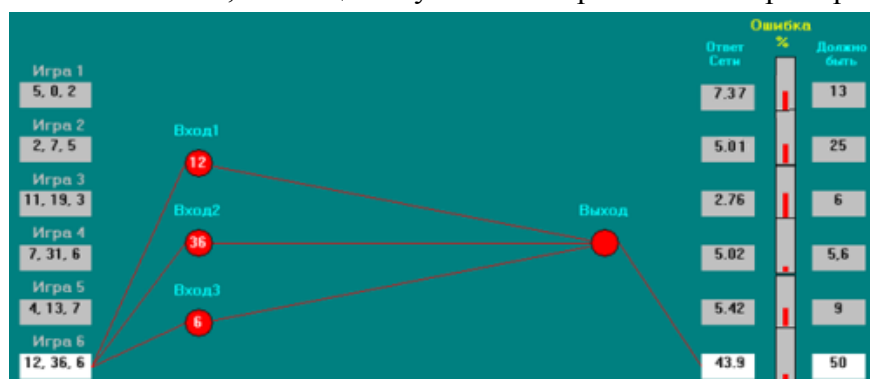
Поскольку изначально веса связей установлены случайным образом, для некоторых примеров ответ сети может очень сильно отличаться от желаемого ответа. Абсолютное значение разности между желаемым ответом и ответом сети называется ОШИБКОЙ сети. Чем меньше ошибка, тем лучше будут предсказания. Подадим на вход сети данные первого примера:



Высота красного столбика в столбце ошибок показывает величину ошибки для примера Игра 1. Для многих примеров вначале ошибка может быть достаточно велика. Но в процессе обучения ИНС подберет и запомнит, какие веса дают наилучшие результаты. Продолжим обучение и на вход подадим данные следующего примера:

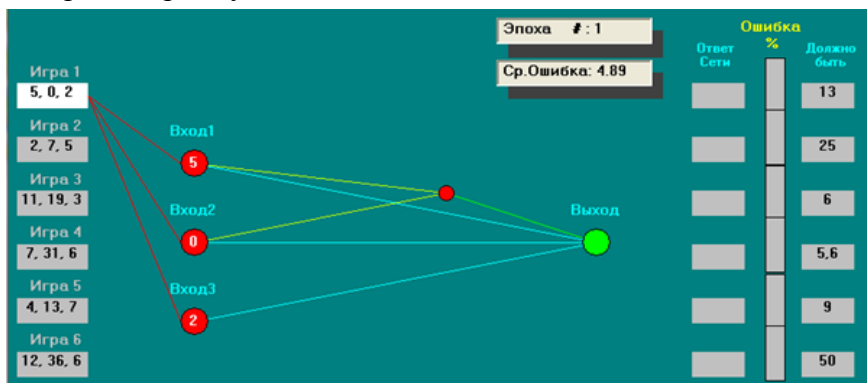


А потом и всех остальных, имеющихся у нас в наборе тестовых примеров:

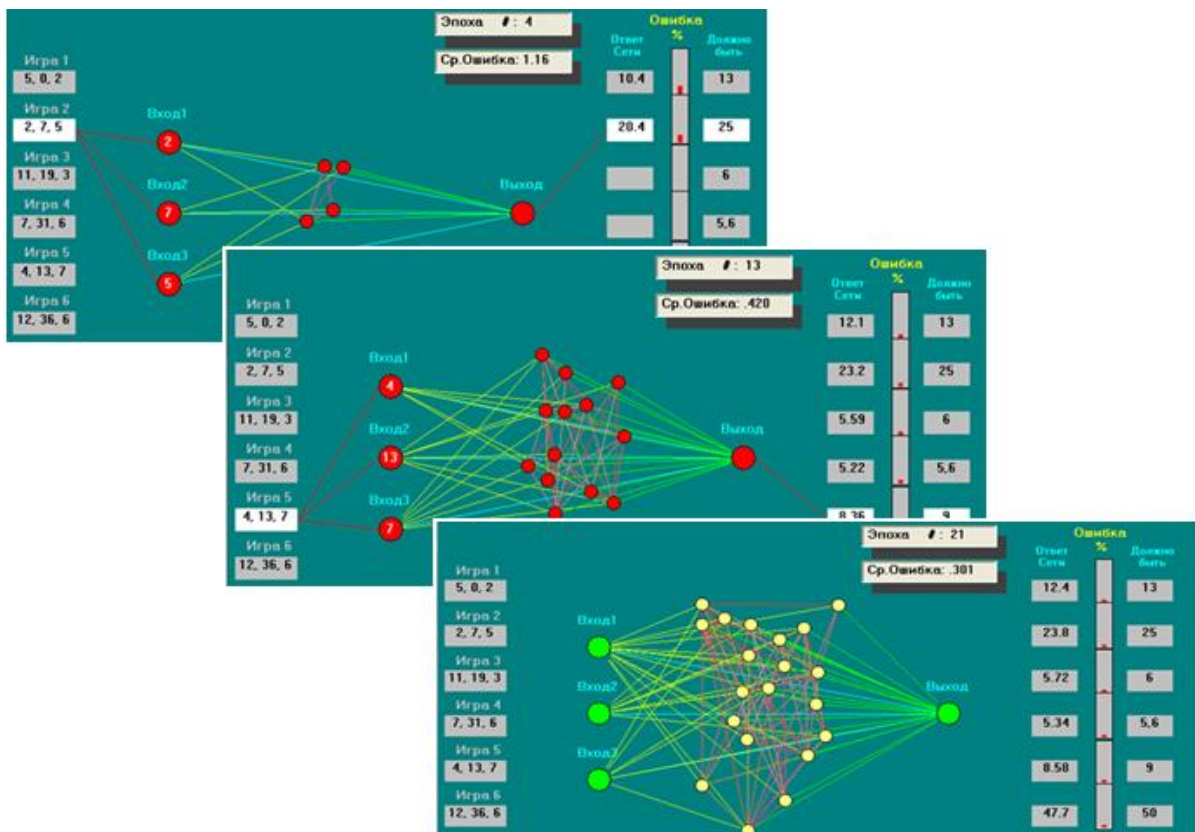


После того, как данные всех примеров будут предъявлены нейронной сети, величина ошибки используется для такого изменения весов всех связей, чтобы ответы ИНС стали как можно ближе к желаемым ответам.

Один цикл предъявления всех примеров называется ЭПОХОЙ. Если суммарная ошибка по всем примерам достаточно велика, то между входными и выходными нейронами добавляется СКРЫТЫЙ НЕЙРОН. Впоследствии может быть добавлено необходимое количество скрытых нейронов. Скрытые нейроны имеют связи с одним или несколькими входными нейронами, другими скрытыми нейронами и с выходным нейроном. Процесс обучения повторяется до тех пор, пока средняя ошибка не упадет до приемлемого уровня. Давайте добавим один скрытый нейрон и пропустим все данные снова через нейронную сеть.



Далее можно повторить этот процесс с двумя скрытыми нейронами, тремя и т.д. Обратите внимание, что уровень ошибки падает от эпохи к эпохе при добавлении новых скрытых нейронов: эпоха 1 – 4.89, ..., эпоха 4 – 1.16, ..., эпоха 13 – 0.420, ..., и, наконец, эпоха 21 – 0.301.



Теперь нейронная сеть достаточно хорошо выучила данные, и готова при предъявлении новых данных делать предсказания интересующих нас величин.



## 11.7. Классификация искусственных нейронных сетей

Схематично ИНС можно представить следующим образом [2,3].

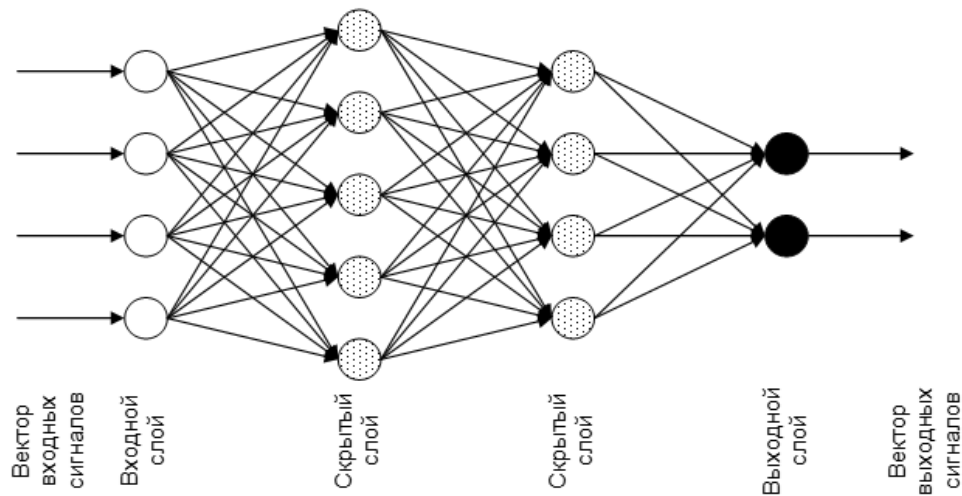


Рис.11.6. Искусственная нейронная сеть

На нейроны входного слоя, которые соответствует одной из характеристик объекта, подаются измеренные значения этих характеристик. Сигналы, поступившие к нейронам входного слоя, передаются на следующий слой (скрытый или выходной) без преобразования: к ним не применяется функция активации. Скрытые слои и выходной слой, отражающие специфику знаний, преобразуют входные данные. Выходной слой генерируют скаляр или вектор, которые являются решением задачи. В частности, они определяют распознаваемый образ.

В зависимости от количества преобразующих слоев (скрытых и выходного) различают однослойные и многослойные сети. В однослойных отсутствуют скрытые слои. Определение числа нейронов в каждом слое и количества скрытых слоев для каждой конкретной задачи является неформальной проблемой, при решении которой можно использовать эвристическое правило: число нейронов в следующем слое должно быть в два раза меньше, чем в предыдущем.

В зависимости от **типа межнейронных связей** различают ИНС:

- с прямыми связями (рис. 11.6);
- с перекрестными связями (рис. 11.7);

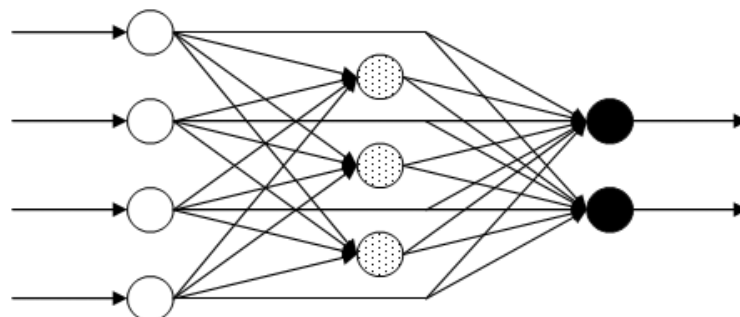


Рис.11.7. ИНС с перекрестными связями

- с обратными (рекуррентными) связями (рис. 11.8). В таких сетях нейрон может посылать сигналы сам себе, нейронам того же слоя или нейронам предыдущих слоев.

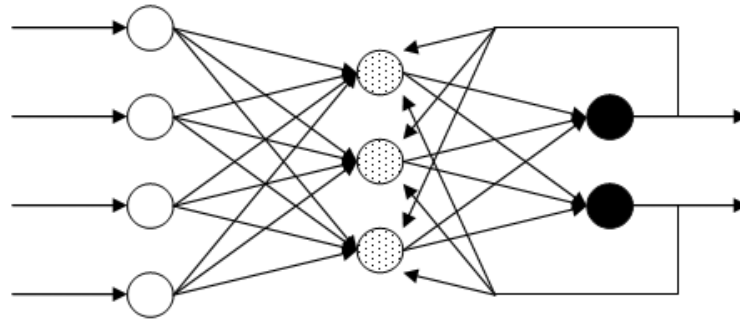


Рис.11.8. ИНС с обратными связями

Перед непосредственным использованием, т.е. перед решением конкретной задачи распознавания образов, необходимо настроить (обучить) сеть. Процесс обучения сети заключается в определении набора связей и коэффициентов связей между нейронами. В зависимости от **способа обучения** различают следующие типы ИНС (рис. 11.9).

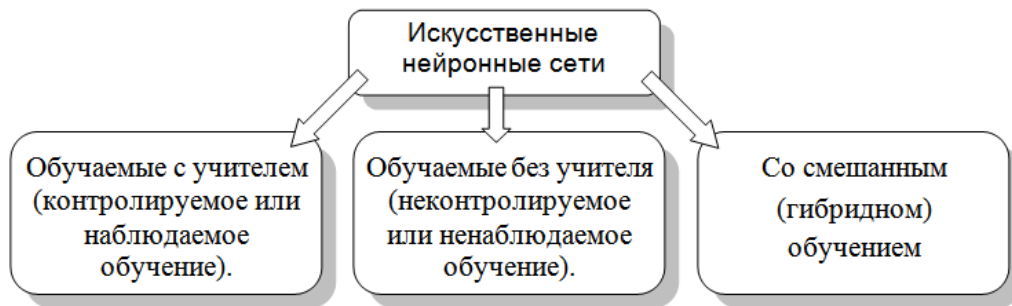


Рис.11.9. Классификация ИНС по способу обучения.

При обучении с учителем все примеры обучающей выборки содержат правильные ответы (выходы), соответствующие исходным данным (входам). В процессе обучения веса настраиваются так, чтобы сеть находила ответы, наиболее близкие к правильным.

Обучение без учителя используется, когда не для всех данных обучающей выборки известны правильные ответы. В этом случае предпринимаются попытки определения коэффициентов сети с целью определения категорий (классов) образов и дальнейшего их распределения по категориям. Используется, в частности, для решения задач кластеризации. При смешанном обучении часть весов определяется посредством обучения с учителем, а другая часть получается с помощью алгоритмов самообучения.

В зависимости от используемого алгоритма коррекции коэффициентов – **правила коррекции коэффициентов ( $w_i$ )** искусственные нейронные сети подразделяют на следующие категории:

- Правило коррекции по ошибке (дельта-правило). Корректировка заключается в небольшом (обычно менее 1%) увеличении весов тех связей, которые усиливают правильные реакции, и уменьшении весов тех связей, которые способствуют ошибочным. Обычно используется для однослойных сетей;
- Правило обратного распространения ошибки. При обучении корректировка распространяется обратно по сети на все весовые коэффициенты. Используется в многослойных сетях. В случае, когда корректировка весовых коэффициентов выполняется после прогона одного образа, говорят о последовательном режиме обучения. Режим, когда корректировка выполняется не сразу, а после прогона нескольких обучающих образов, называется пакетным;
- Синхронное обучение (правило Хебба). Это правило опирается на следующие нейрофизиологическое наблюдение: если нейроны с обеих сторон синапса

активизируются одновременно и регулярно, то сила синаптической связи возрастает. Таким образом, корректировке подлежат коэффициенты только тех связей, выход которых отличен от нуля;

- Конкурентное обучение (правило Кохонена, «победитель забирает все»). В каждом слое корректируются весовые коэффициенты только одного нейрона, у которого выход наиболее точно соответствует предъявленному образцу;
- Правило Больцмана. Алгоритм обучения основан на идее моделирования отжига – способа выжигания дефектов в кристаллической решетке. Атомы, занимающие в ней неправильное место, при низкой температуре не могут сместиться в нужное положение - им не хватает кинетической энергии для преодоления потенциального барьера. При этом система в целом находится в состоянии локального энергетического минимума. Для выхода из него металл нагревают до высокой температуры, а затем медленно охлаждают, позволяя атомам занять правильные положения в решетке, соответствующее глобальному минимуму энергии. Имитация отжига в нейронной сети выполняется по следующей процедуре:
  - ✓ на вход сети подается обучающий образ и вычисляется выход;
  - ✓ вычисляется значение средней квадратичной ошибки между желаемым и полученным выходными векторами;
  - ✓ весовые коэффициенты изменяются случайным образом, затем вычисляются новый выход и результирующая ошибка. Если ошибка уменьшилась, оставляют измененные веса; если ошибка увеличилась, оставляют измененные веса с вероятностью, определяемой распределением Больцмана. Если ошибка осталась неизменной, то весовые коэффициенты возвращают к его предыдущему значению.

## 11.8. Процедура построения искусственных нейронных сетей

*Первым этапом* построения нейросетевой модели является тщательный отбор входных данных, влияющих на ожидаемый результат. Из исходной информации необходимо исключить все сведения, не относящиеся к исследуемой проблеме. В то же время следует располагать достаточным количеством примеров для обучения ИНС. Существует эмпирическое правило, которое устанавливает соотношение  $\lambda$  между количеством примеров, содержащих входные данные и правильные ответы, и числом соединений в нейронной сети:  $\lambda \leq 10$  [2].

Для факторов, которые включаются в обучающую выборку, целесообразно предварительно оценить их значимость, проведя корреляционный и регрессионный анализ, и проанализировать диапазоны их возможных изменений.

*На втором этапе* осуществляется преобразование исходных данных с учетом характера и типа проблемы, отображаемой нейросетевой моделью, и выбираются способы представления информации. Эффективность нейросетевой модели повышается, если диапазоны изменения входных и выходных величин приведены к некоторому стандарту, например  $[0,1]$  или  $[-1,1]$ .

*Третий этап* заключается в проектировании архитектуры ИНС (определение числа слоев и числа нейронов в каждом слое), а также в выборе функции активации нейронов, учитывающей характер решаемой задачи. Структура ИНС формируется до начала обучения, поэтому успешное решение этой проблемы во многом определяется опытом и искусством аналитика, проводящего исследования.

*Четвертый этап* связан с обучением сети - уточнением значений весовых коэффициентов (синаптических весов) связей на основе многократного прогона через сеть эталонных примеров.

Третий и четвертый этапы могут совместно применяться для уточнения структуры сети на основе конструктивного или деструктивного подхода.

- В соответствии с первым подходом обучение ИНС начинается на сети небольшого размера, который постепенно увеличивается до достижения требуемой точности по результатам тестирования.
- Деструктивный подход базируется на принципе «прореживания дерева», в соответствии с которым из сети с заведомо избыточным объемом постепенно удаляют «лишние» нейроны и примыкающие к ним связи. Этот подход дает возможность исследовать влияние удаленных связей на точность сети.

*На пятом этапе* проводится тестирование полученной модели ИНС на независимой выборке примеров.

## 11.9. Алгоритмы обучения искусственных нейронных сетей

Способность к обучению является уникальным свойством мозга. Для ИНС под обучением понимается процесс настройки (корректировки) весовых коэффициентов связей ( $w$ ) и порогов нейронов ( $P^*$ ) для эффективного решения поставленной задачи. К сожалению, нет универсального алгоритма обучения, подходящего для всех архитектур нейронных сетей. Рассмотрим один из самых распространенных алгоритмов обучения многослойных сетей с прямыми связями – алгоритмом обратного распространения ошибки (*back propagation*). Основные этапы реализации этого алгоритма:

- На первом шаге все параметры сети устанавливаются произвольно.
- Через сеть прогоняются тренировочные данные, и вычисляется суммарная функция ошибки  $E = \sum(E_i^2)$ , где  $E_i = Y_i - y_i$ ,  $Y_i$  – вычисленное значений выходной величины,  $y_i$  – ожидаемое значение.
- Вычисляется значение производных функции ошибки по каждому параметру, а на их основе – расчет поправок к параметрам нейронной сети.
- Параметры сети корректируются на величину поправок. Шаги 2 и 3 повторяется с начала до тех пор, пока функция ошибки не снизится до заданного уровня.

Этот алгоритм был изложен в диссертации Пола Вербоса (Paul Werbos) 1974 г., но тогда не привлек к себе должного внимания. Рождение алгоритма для широкой публики связано с работой группы PDP (Parallel Distributed Processing), освещенной в двухтомном труде 1986г. Именно там, в статье Румельхарта, Хинтона и Уильямса, была изложена теория обучения многослойного персептрона.

### Исходные предпосылки и данные алгоритма.

Для реализации этого алгоритма, как и большинство других алгоритмов, надо четко определить исходные предпосылки и соответствующие набора данных. Рассмотрим, реализацию этого алгоритма на примере задачи классификации цифр от 0 до 9. Для решения этой задачи должны быть определены и известны:

- Структура сети. – Это количество слоев и количество нейронов в каждом слое.  
Пусть исходный образ цифры 0 будет представляться матрицей  $7 \times 9$  (рис. 11.10), аналогично будут представляться и другие цифры. Таким образом, количество элементов входного слоя будет равно  $7 \times 9 = 63$ . На вход этим элементам будут поступать значения: либо 0 – белая клетка, либо 1 – черная клетка.

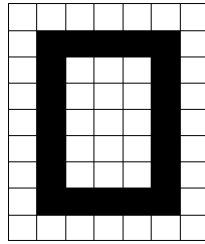


Рис. 11.10. Представление цифры 0

Количество элементов выходного слоя будет равно количеству распознаваемых образов, то есть 10 (цифры 0..9). Тогда выходной вектор  $[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$  означает, что распознана цифра 2. Количество скрытых слоев и элементов в этих слоях будет устанавливаться опытным путем.

- Вид функции активации. – Пусть, это будет сигмоидальная функция вида  $Y = 1 / (1 + e^{-aP})$ , где  $P = \sum w_i x_i$  – уровень активации (потенциал) нейрона.
- Начальные значения весовых коэффициентов. – Все весовые коэффициенты  $w_{ij}$  инициализируются случайными величинами, например, в диапазоне  $[-0.3, 0.3]$ . Значения  $i$  и  $j$  – это уникальные номера элементов из которого направлена связь и в который направлена связь, соответственно.
- Эталонные образы. – Это наборы из двух векторов: с эталонными входными и выходных значениями. В данной задаче, например можно сформировать 10 образцов на 10 цифр.
- Количество эпох. – То есть, повторов обучения всем эталонным образцам. Можно использовать и другие способы прекращения обучения. Например, по заданной погрешности между текущей эпохой и предыдущей.
- Норма обучения  $\alpha$ . – Это коэффициент, от которого зависит величина изменения веса (скорость обучения). Выбирается в диапазоне  $0 < \alpha < 1$ .

### Математическая основа алгоритма.

В основе алгоритма лежит математический метод градиентного спуска, который осуществляется в пространстве весовых коэффициентов в целях уменьшения значения ошибки между ожидаемым выходным значением  $y'$  и получаемым в результате прогона эталонного образца  $y$ .

Изменение весового коэффициента  $w_{ij}$  от нейрона  $i$  слоя  $k-1$  к нейрону  $j$  слоя  $k$  определяется по формуле

$$\Delta w_{ij} = \alpha \delta_j x_i,$$

где  $\delta_j$  – локальный градиент, определяемый выражением

$$\delta_j = e_j Y'(P_j),$$

где  $e_j$  – сигнал ошибки для нейрона  $j$  выходного слоя  $e_j = y'_j - y_j$ ;

$e_j$  – для нейронов скрытых слоев  $e_j = \sum \delta_s w_{js}$ ,  $s$  – индекс нейрона слоя  $k+1$ ;

$Y'(P_j)$  – значение первой производной функции активации для нейрона  $j$  с потенциалом  $P_j = \sum w_{ij} x_i$ .

Наличие первой производной (дифференцируемости функции активации) обуславливает применимость алгоритма обратного распространения ошибки. Первая производная сигмоидальной функции имеет вид:

$$Y' = (a e^{-aP}) / (1 + e^{-aP})^2 = a y_j (1 - y_j)$$

Новое значение весового коэффициента  $w'_{ij}$  определяется по формуле

$$w'_{ij} = w_{ij} + \Delta w_{ij}.$$

Алгоритм обратного распространения ошибки подразумевает два потока сигналов: прямое (от входного до выходного слоя) распространение функционального сигнала при распознавании эталонного образа и обратное (от выходного до входного слоя) распространение сигнала ошибки для корректировки весовых коэффициентов.

### Алгоритм обучения (на примере сигмоидальной функции активации при $a = 1$ ).

Шаг 1. Инициализировать значения всех весовых коэффициентов  $w_{ij}$  случайными значениями в диапазоне  $[-0.3, 0.3]$ .

Шаг 2. Для каждой эпохи повторить следующие шаги.

Шаг 2.1. Для одного эталонного образца повторить следующие шаги.

Шаг 2.1.1. Для каждого элемента входного слоя установить значение вывода  $x_i$  равным  $i$ -ому значению входного вектора.

Шаг 2.1.2. Для каждого следующего слоя (в прямом порядке) повторить следующие шаги.

Шаг 2.1.2.1. Для каждого  $j$ -ого элемента слоя вычислить совокупный ввод и вывод по формулам:

- совокупный ввод:  $P_j = \sum x_i w_{ij}$ , где  $i$  – номера нейронов предыдущего слоя;
- совокупный вывод:  $y_j = x_j = 1 / (1 + e^{-P})$ .

Шаг 2.1.3. Для каждого выходного элемента вычислить его ошибку по формуле

$$\delta_j = (y'_j - y_j) * y_j * (1 - y_j),$$

где  $y'_j$  – ожидаемое значение из эталонного выходного вектора.

Шаг 2.1.4. Для каждого предыдущего скрытого слоя (в обратном порядке) повторить следующие шаги.

Шаг 2.1.4.1. Для каждого  $i$ -ого элемента слоя вычислить его ошибку по формуле

$$\delta_i = y_i * (1 - y_i) * \sum \delta_j w_{ij},$$

где  $j$  – номера элементов следующего слоя.

Шаг 2.1.5. Для всех весовых коэффициентов выполнить перерасчет по формуле

$$w_{ij} = w_{ij} + \alpha * \delta_j * y_i.$$

### Тестирование обученной нейронной сети.

На вход подается, как правило, искаженный образец. Искусственная нейронная сеть на основе определенных при обучении итоговых весовых коэффициентов определяет совокупный вывод элементов выходного слоя. Элемент с максимальным совокупным выводом и есть распознанный образ. Например, если выходной вектор для задачи классификации цифр равен  $[0 \ 0.11 \ 0.23 \ \mathbf{0.76} \ 0.01 \ 0.05 \ 0 \ 0 \ 0 \ 0.25]$ , то предъявленный для распознавания образ соответствует цифре 3.

## 11.10. Иллюстрация принципа работы алгоритма обратного распространения

Рассмотрим, приведенное в [5,6] описание функционирования алгоритма обратного распространения ошибки (back propagation). Иллюстрировать этот процесс будем на базе нейронной сети, состоящей из трёх слоёв, имеющей два входа и один выход:

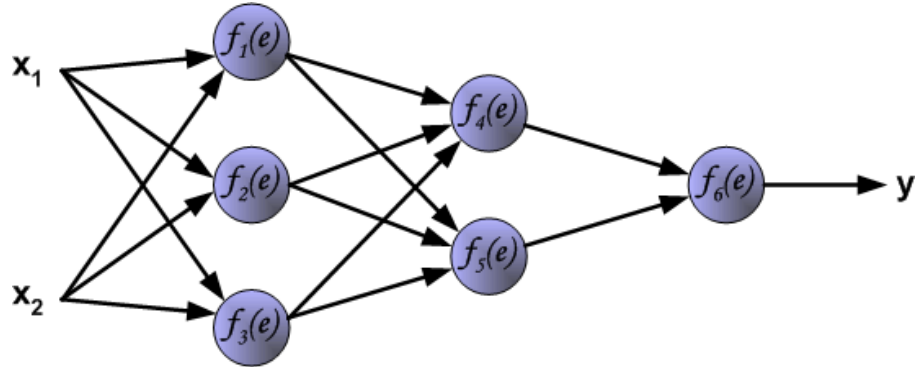


Рис. 11.11. ИНС из трёх слоёв, с двумя входами и одним выходом

Каждый шаг обучения начинается с воздействия входных сигналов из набора тренировочных примеров. После этого можно определить значения выходных сигналов для всех нейронов в каждом слое сети. Иллюстрации, приведенные ниже, показывают, как сигнал распространяется по первому слою сети.

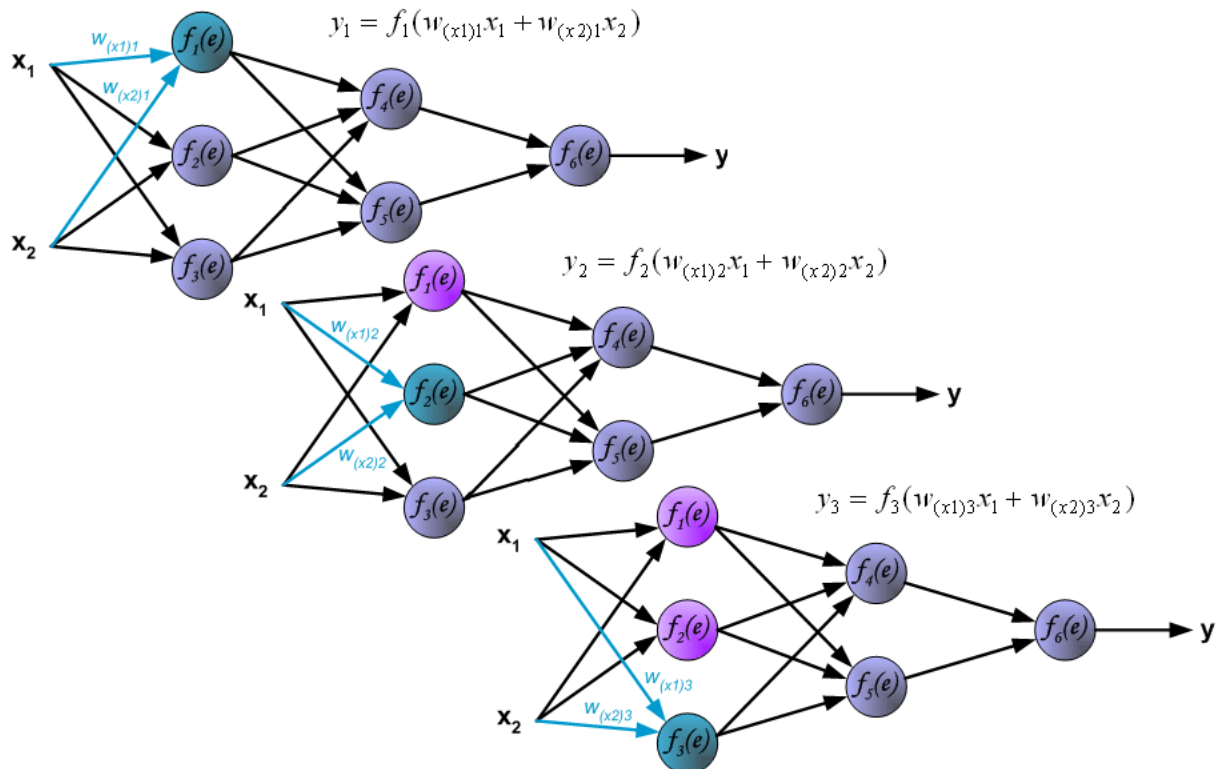


Рис. 11.12. Прямое распространение на входном слое.

Обозначение введенные на схеме:

- $W_{(X_m)_n}$  представляют вес связи между сетевым входом  $X_m$  и нейроном  $n$  во входном слое.
- $y_n$  представляют выходной сигнал нейрона  $n$ .

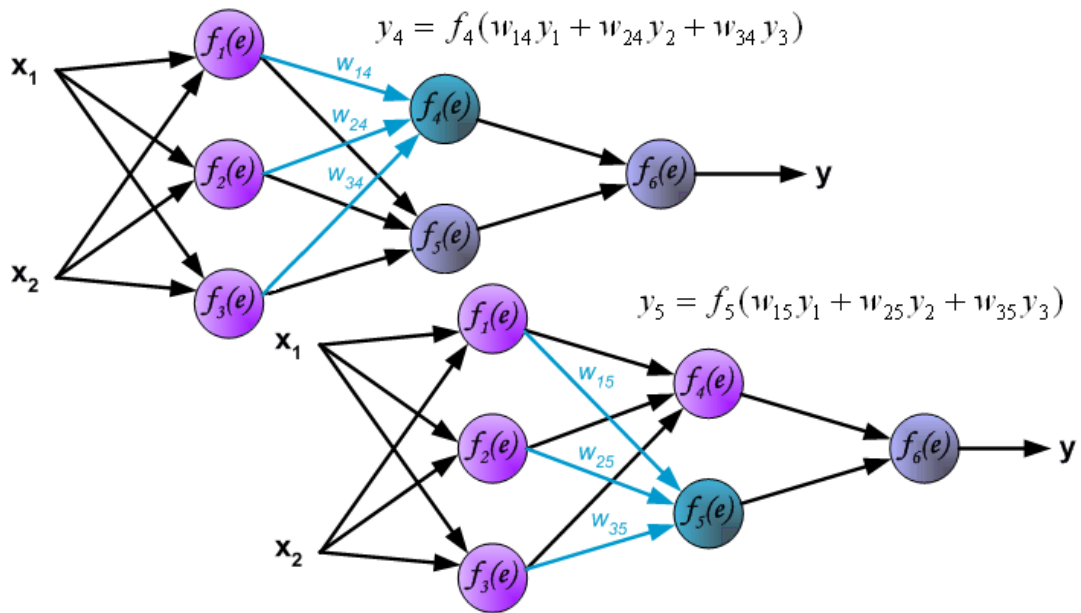


Рис. 11.13. Распространение сигнала через скрытый слой

На рис. 11.13 приведено распространение сигнала через скрытый слой. Символы  $w_{mn}$  представляют весовые множители связей между выходом нейрона  $m$  и входом нейрона  $n$  в следующем слое. На следующем шаге производится распространение сигнала через выходной слой.

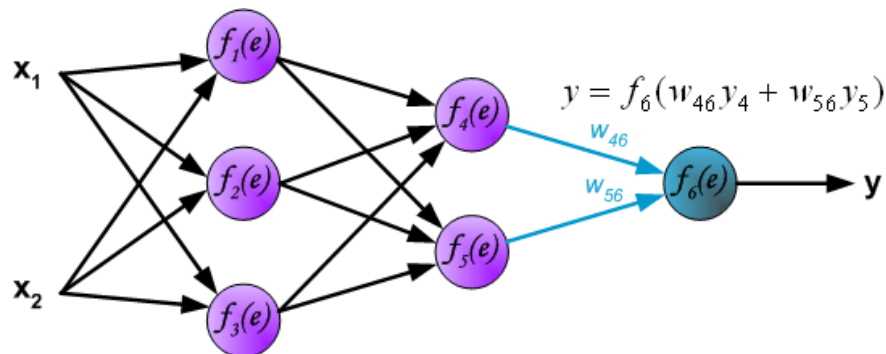


Рис. 11.14. Распространение сигнала через выходной слой.

На следующем шаге алгоритма, выходной сигнала сети  $y$  сравнивается с желаемым выходным сигналом  $z$ , который хранится в тренировочных данных. Разница между этими двумя сигналами называется ошибкой  $\delta$  выходного слоя сети.

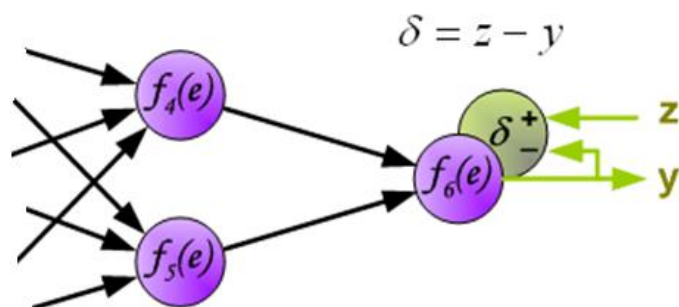


Рис. 11.15. Вычисление ошибки распространения тестового сигнала.

Невозможно непосредственно вычислить сигнал ошибки для внутренних нейронов, потому что выходные значения этих нейронов, неизвестны. На протяжении многих лет



был неизвестен эффективный метод для обучения многослойной сети. Только в середине восьмидесятых годов был разработан алгоритм обратного распространения ошибки. Идея заключается в распространении сигнала ошибки  $\delta$  (вычисленного в шаге обучения) обратно на все нейроны, чьи выходные сигналы были входящими для последнего нейрона.

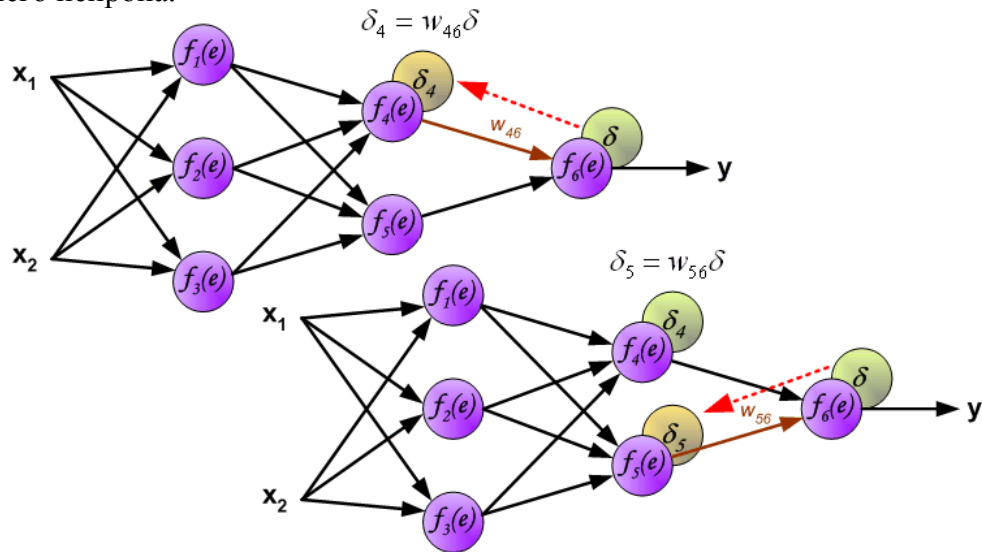
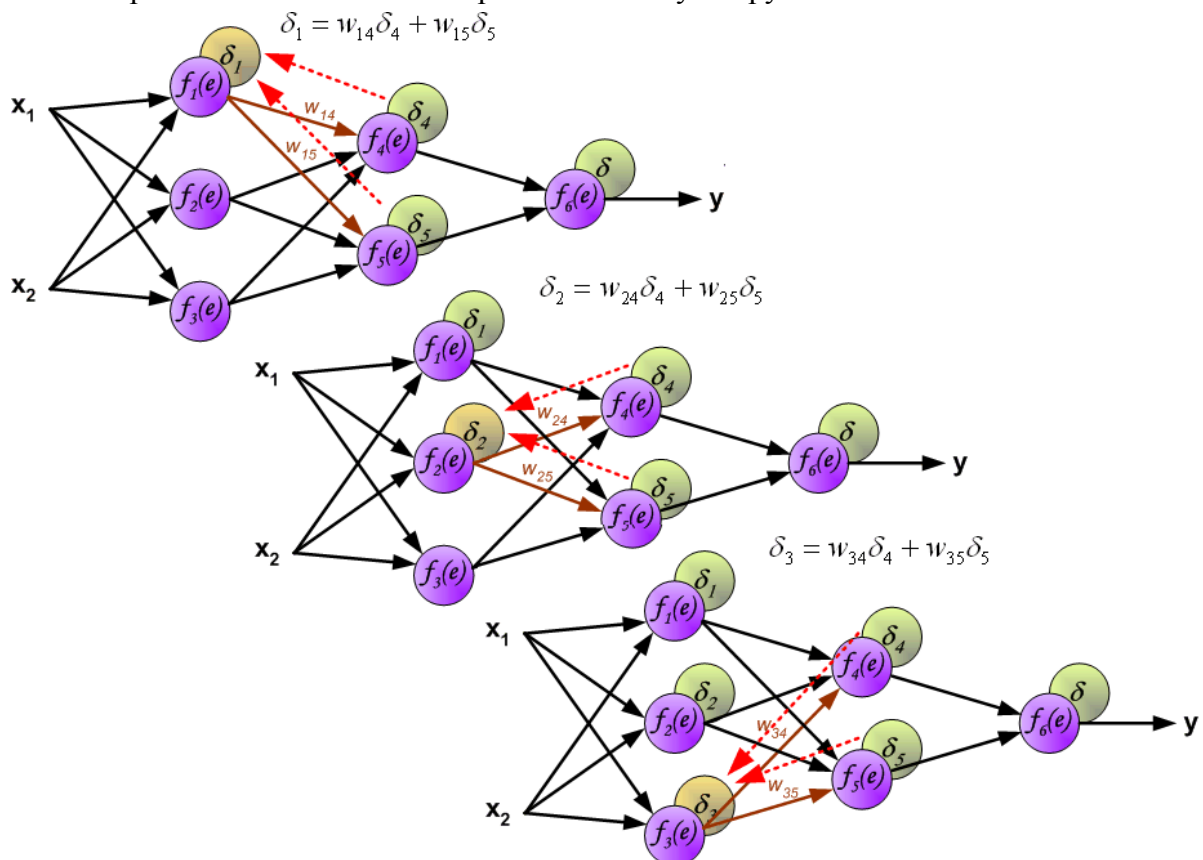


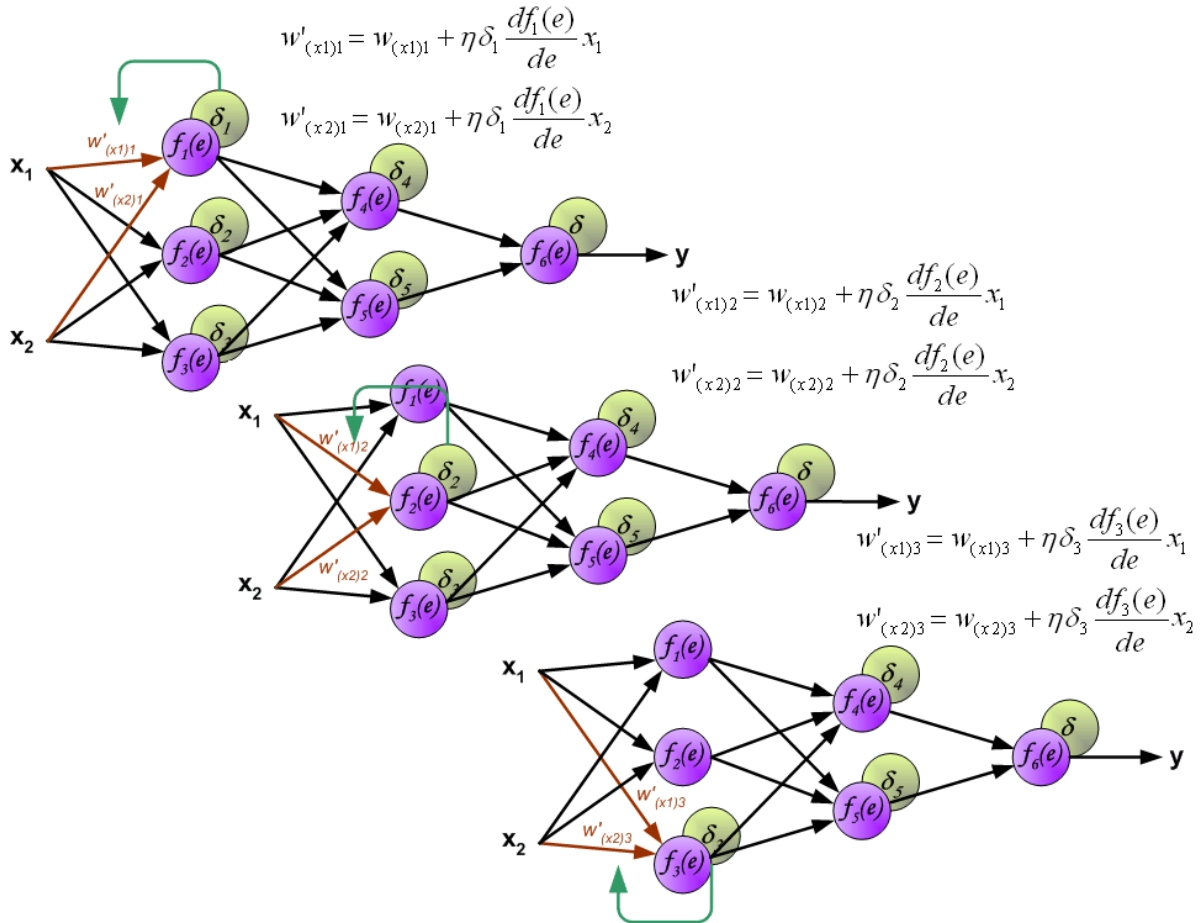
Рис. 11.15. Пересчет ошибок на скрытый слой.

Весовые коэффициенты  $W_{mn}$ , используемые для обратного распространения ошибки, равны тем же коэффициентам, что использовались во время вычисления выходного сигнала. Только изменяется направление потока данных. Теперь сигналы передаются от выхода к входу. Этот процесс повторяется для всех слоёв сети. Если ошибка пришла от нескольких нейронов — она суммируется:



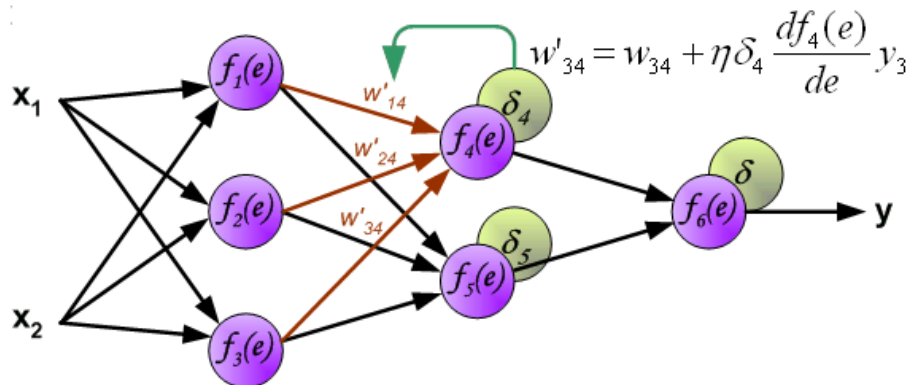
Когда вычисляется величина ошибки сигнала для каждого нейрона – можно скорректировать весовые коэффициенты каждого узла ввода (дендрита) нейрона.

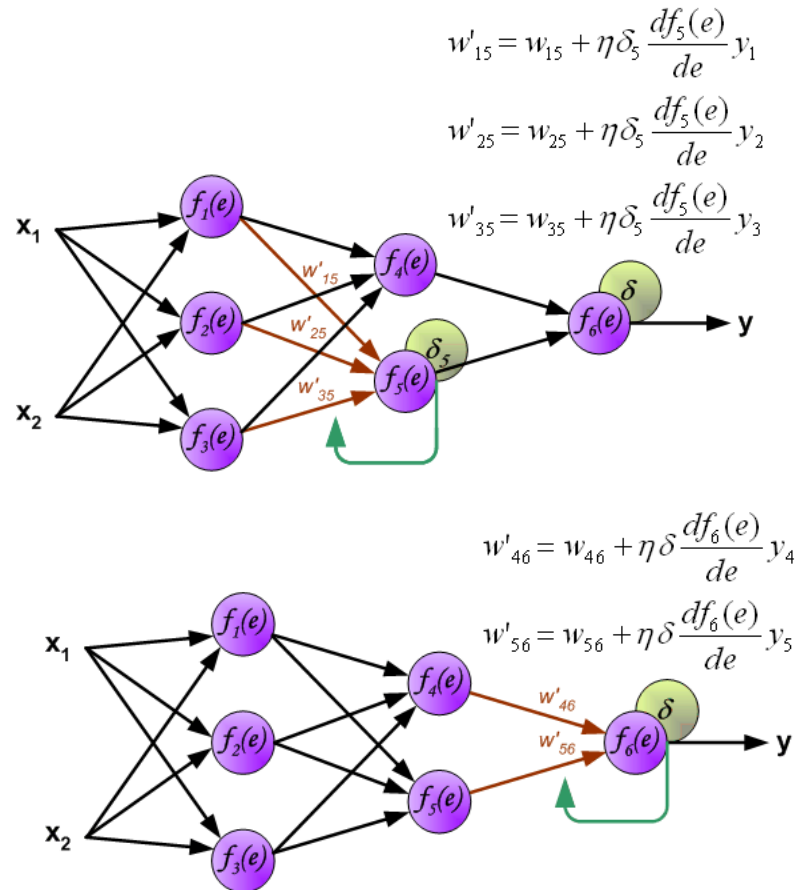
В формулах ниже  $df(e)/de$  — является производной от функции активации нейрона, чьи весовые коэффициенты корректируются.



$$w'_{14} = w_{14} + \eta \delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta \delta_4 \frac{df_4(e)}{de} y_2$$





Коэффициент  $h$  влияет на скорость обучения сети. Есть несколько методов для выбора этого параметра. Первый способ — начать процесс обучения с большим значением параметра  $h$ . Во время коррекции весовых коэффициентов, параметр постепенно уменьшают. Второй — более сложный метод обучения, начинается с малым значением параметра  $h$ . В процессе обучения параметр увеличивается, а затем вновь уменьшается на завершающей стадии обучения. Начало процесса обучения с низким значением параметра  $h$  позволяет определить знак весовых коэффициентов.

### 11.11. Применение искусственных нейронных сетей для решения задач

В следующей таблице (табл. 11.2) приведены рекомендации по использованию наиболее распространенных архитектур ИНС для решения задач.

Таблица 11.2

Рекомендации по использованию ИНС

Тип обучения	Правило коррекции коэффициентов	Архитектура	Задачи
С учителем	Коррекция по ошибке и обратное распространение ошибки	Однослойные и многослойные сети с прямыми и обратными связями	Классификация. Аппроксимация функций. Прогнозирование. Управление.
	Правило Хебба	Многослойные сети с прямыми связями	Анализ данных. Классификация.
	Соревнование	Сеть ART (адаптивный резонанс)	Классификация.

Без учителя	Коррекция по ошибке и обратное распространение ошибки	Многослойные сети с прямыми связями	Кластеризация. Анализ данных.
	Правило Хебба	Сети Хопфилда	Ассоциативная память. Оптимизация.
	Соревнование	Сети SOF (самоорганизующиеся карты, сети Кохонена)	Кластеризация. Анализ данных.
		Сети ART (сети на базе теории адаптивного резонанса)	Кластеризация.
Смешанная	Коррекция по ошибке, обратное распространение ошибки и соревнование	Сеть RBF (сети с радиальными базисными функциями) - частный случай двухслойной сети с прямыми связями	Классификация. Аппроксимация функций. Прогнозирование. Управление.

### Нейросетевая классификация

Отметим, что задачи классификации (типа распознавания букв) очень плохо алгоритмируются. Если в случае распознавания букв верный ответ очевиден для нас заранее, то в более сложных практических задачах обученная нейронная сеть выступает как эксперт, обладающий большим опытом и способный дать ответ на трудный вопрос.

Примером такой задачи служит медицинская диагностика, где нейронная сеть может учитывать большое количество числовых параметров (энцефалограмма, давление, вес и т.д.). Конечно, "мнение" нейронной сети в этом случае нельзя считать окончательным. Классификация предприятий по степени их перспективности - это уже привычный способ использования нейронных сетей в практике западных компаний. При этом нейронная сеть также использует множество экономических показателей, сложным образом связанных между собой.

Нейросетевой подход особенно эффективен в задачах экспертной оценки по той причине, что он сочетает в себе способность компьютера к обработке чисел и способность мозга к обобщению и распознаванию. Говорят, что у хорошего врача способность к распознаванию в своей области столь велика, что он может провести приблизительную диагностику уже по внешнему виду пациента. Можно согласиться также, что опытный трейдер чувствует направление движения рынка по виду графика. Однако в первом случае все факторы наглядны, то есть характеристики пациента мгновенно воспринимаются мозгом как "бледное лицо", "блеск в глазах" и т.д. Во втором же случае учитывается только один фактор, показанный на графике - курс за определенный период времени. Нейронная сеть позволяет обрабатывать огромное количество факторов (до нескольких тысяч), независимо от их наглядности - это универсальный "хороший врач", который может поставить свой диагноз в любой области.

### Кластеризация с помощью нейронных сетей и поиск зависимостей

Помимо задач классификации, нейронные сети широко используются для поиска зависимостей в данных и кластеризации. Например, нейронная сеть на основе методики МГУА (метод группового учета аргументов) позволяет на основе обучающей выборки построить зависимость одного параметра от других в виде полинома. Такая нейронная сеть может не только мгновенно выучить таблицу умножения, но и найти

сложные скрытые зависимости в данных (финансовых), которые не обнаруживаются стандартными статистическими методами

Кластеризация - это разбиение данных на несколько компактных областей (кластеров), причем число кластеров заранее неизвестно. Кластеризация позволяет представить неоднородные данные в более наглядном виде и использовать далее для исследования каждого кластера различные методы. Например, так можно быстро выявить фальсифицированные страховые случаи или недобросовестные предприятия.

### **Применение нейронных сетей в задачах прогнозирования**

Задачи прогнозирования особенно важны для практики, в частности, для финансовых приложений, поэтому поясним способы применения нейронных сетей в этой области более подробно.

Рассмотрим практическую задачу, ответ в которой неочевиден - задачу прогнозирования курса акций на 1 день вперед. Пусть у нас имеется база данных, содержащая значения курса за последние 300 дней. Простейший вариант в данном случае - попытаться построить прогноз завтрашней цены на основе курсов за последние несколько дней. Понятно, что прогнозирующая нейронная сеть должна иметь всего один выход и столько входов, сколько предыдущих значений мы хотим использовать для прогноза - например, 4 последних значения. Составить обучающий пример очень просто - входными значениями нейронной сети будут курсы за 4 последовательные дня, а желаемым выходом нейронной сети - известный нам курс в следующий день за этими четырьмя.

### **Вопросы для самопроверки**

- Дайте семантическую и синтаксическую классификация понятия «образ».
- Опишите строение биологического нейрона.
- Опишите математическую модель нейрона.
- Приведите классификацию ИНС по типу нейронных связей.
- Опишите процедуру построения искусственных нейронных сетей.

### **Библиографические ссылки:**

1. Старков Ф.А., Старков Е.Ф. Распознавание образов. Учеб. пособие – Курск: Изд-во Курск. гуманит.-техн. ин-т., 2000. – 140 с.
2. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2004. – 424 с.
3. Люггер Джордж Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Издательский дом «Вильямс», 2004. – 864 с.
4. Бессмертный И.А. Искусственный интеллект – СПб:СПбГУ ИТМО, 2010.– 132с.
5. Нейронная сеть – обучение ИНС с помощью алгоритма обратного распространения. – <http://robocraft.ru/blog/algorithm/560.html>
6. Principles of training multi-layer neural network using backpropagation – [http://galaxy.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html)