- **Большие данные (big data) –** серия математических методов, методик и алгоритмов обработки структурированных и неструктурированных данных больших объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста и распределения по многочисленным узлам вычислительной сети, сформировавшихся в конце 2000-х годов и альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

1. Горизонтальная масштабируемость. Поскольку данных может быть сколь угодно много – любая система, которая подразумевает обработку больших данных, должна быть расширяемой. В 2 раза вырос объём данных – в 2 раза увеличили количество железа в кластере и всё продолжило работать.
2. Отказоустойчивость. Принцип горизонтальной масштабируемости подразумевает, что машин в кластере может быть много. Например, Hadoop-кластер Yahoo имеет более 42000 машин. Это означает, что часть этих машин будет гарантированно выходить из строя. Методы работы с большими данными должны учитывать возможность таких сбоев и переживать их без каких-либо значимых последствий.
3. Локальность обработки данных. В больших распределённых системах данные распределены по большому количеству машин. Если данные физически находятся на одном сервере, а обрабатываются на другом – расходы на передачу данных могут превысить расходы на саму обработку. Поэтому одним из важнейших принципов проектирования BigData-решений является принцип локальности обработки данных – по возможности обрабатываем данные на той же машине, на которой их храним.

1. Медицина и биология – исследования в области медицины, биологии и генетики [Проект по расшифровке генома человека, главной целью которого было определить последовательность нуклеотидов, которые составляют ДНК и идентифицировать около 25 тыс. генов в человеческом геноме, потребовал около 10 лет и более 5 млрд. долл.]
2. Физика элементарных частиц. [Большой адронный коллайдер– ускоритель заряженных частиц на встречных пучках, предназначенный для разгона протонов и тяжёлых ионов и изучения продуктов их соударений. Столкновения частиц фиксируются в детекторах коллайдера миллионами датчиков. Детекторы должны зафиксировать «портрет» события, определив траектории частиц, их типы, заряды, энергию. В 2010 году в ходе экспериментов было произведено 13 петабайт данных]
3. Астрофизика [В рамках проекта широкомасштабного исследования изображений и спектров звёзд и галактик «Слоановский цифровой обзор неба», использующего 2,5-метровый широкоугольный телескоп в обсерватории Апачи-Пойнт, Нью-Мексико, в 2000 году был начат сбор данных, то только за первые несколько недель данных было накоплено больше, чем ранее за всю историю астрономических наблюдений. Продолжая собирать данные со скоростью около 200 Гб в сутки, к настоящему времени SDSS накопил более чем 140 терабайт информации]

Неполнота обучающих данных и информации об их природе. Неполные данные – это отсутствие некоторых значимых входов или сигналов на имеющихся входах, неточные входные сигналы, недостоверные выходы. Статистика распределения данных всегда принципиально не полна.

Противоречивость данных и другие источники информационного шума. Данные поступают от внешних сенсоров и систем накопления, передаются по неидеальным каналам, хранятся на неидеальных носителях. При росте объемов данных дополнительно вмешивается фактор времени. Часть признаков в векторах данных успевает устаревать к моменту поступления остальных фрагментов. Такие данные почти наверняка содержат противоречия.

“Проклятие” размерности. С ростом числа анализируемых признаков геометрия многомерных пространств работает против статистики. Можно было бы ожидать, что при разумном числе обучающих примеров достижимы близкие к теоретическим оценки вероятности правильной классификации вектора данных, так как в его окрестности имеется достаточное число близких примеров с известными классами . Но в многомерных пространствах это не так. Например, даже для покрытия 10% объема 10-мерного куба требуется покрыть 80% длины каждого ребра. Таким образом, локальная статистика каждой области данных автоматически оказывается грубой.

Проблема масштабирования алгоритмов. Задачу обучения классификатора для таблицы из 10,000 примеров и 10 входных признаков, по-видимому, нужно признать закрытой. Имеется большое число научных публикаций, в которых обоснованы устойчивые качественные алгоритмы обучения машин для таких масштабов. В свете практических потребностей первостепенное значение приобретают показатели роста вычислительной сложности и ресурсов требуемой памяти алгоритмов при увеличении числа содержательных обучающих примеров. Реальность такова, что даже линейный рост сложности по числу примеров и размерности задачи начинает быть неприемлемым. Например, при обучении на 1,000,000 примеров число получаемых базовых векторов памяти может достигать 10,000 для достижения сходимости валидационной ошибки. Такой размер классификатора оказывается нетехнологичным и по памяти и по скорости вычисления прогнозов.

Выбор модели. Проблема выбора модели, обладающей преимуществами в точности решения задачи, является мета-проблемой, стоящей над задачей обучения. В идеале, обе задачи – и обучения, и оптимизации структуры должны решаться машиной самостоятельно и одновременно. Это лишь частично и приближенно выполнимо на практике, поскольку требования сходящейся точности обучения и оптимальности структуры не являются строго коллинеарными, и мы имеем дело с задачей многокритериальной оптимизации. Статистическое же сравнение множества моделей с целью выбора затрудняется проблемой масштабирования.

Технология или алгоритм	Количество "голосов"	%
Деревья решений/правил	107	14%
Кластеризация	101	13%
Методы регрессии	90	11%
Статистика	80	10%
Визуализация	63	8%
Нейронные сети	61	8%
Ассоциативные правила	54	7%
Методы оценок по ближайшим соседям	34	4%
SVM (методы минимизации структурного риска)	31	4%

Совершенно очевидно, что свою силу нейронные сети черпают, во-первых, из распараллеливания обработки информации и, во-вторых, из способности самообучаться, т.е. создавать обобщения. Под термином *обобщение* (generalization) понимается способность получать обоснованный результат на основании данных, которые не встречались в процессе обучения. Эти свойства позволяют нейронным сетям решать сложные (масштабные) задачи, которые на сегодняшний день считаются трудноразрешимыми.

Минимальный шаг в переходе от *программируемых автоматов и компьютерных программ с предписанной* функциональностью к *машинам анализа больших данных* состоит во встраивании обучаемого классификатора, автоматически выбирающего одну из заранее подготовленных программ.

Классификатор в таких системах принимает решение на основе *вектора внешних наблюдаемых признаков* той ситуации, в которой сейчас находится машина.

Признак – некоторое количественное измерение свойства объекта произвольной природы. Совокупность признаков, характеризующих один объект, называется *вектором признаков*. Считается, что каждому объекту ставится в соответствие единственное значение вектора признаков и наоборот: каждому значению вектора признаков соответствует единственный объект.

Классификатором (или решающим правилом) называется правило отнесения объекта к одному из классов на основании его вектора признаков.

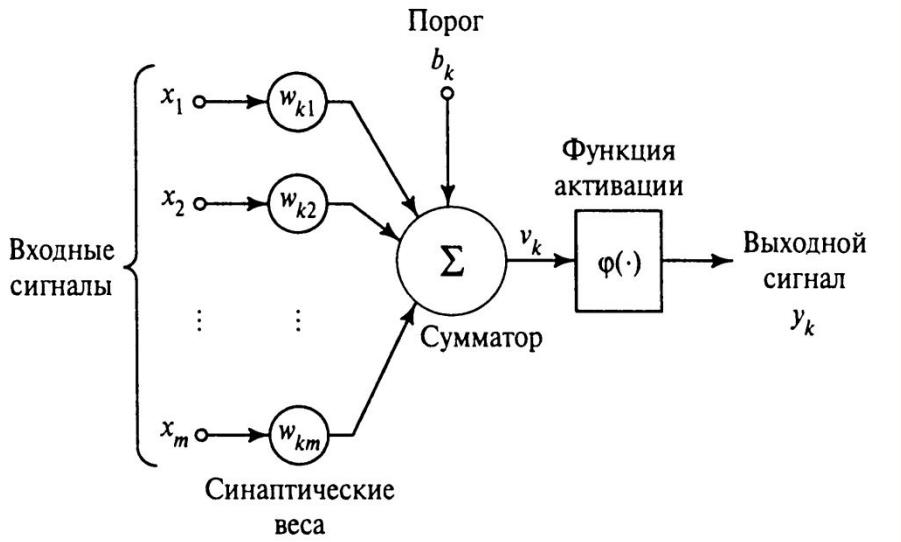
Класс – это совокупность объектов, выделенных по некоторому набору признаков.

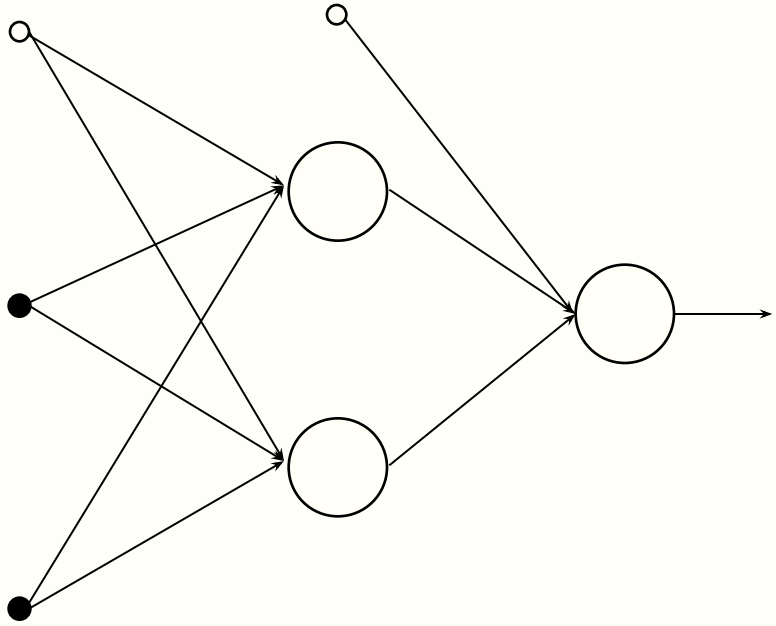
Обучающая выборка – набор объектов, для которых принадлежность к классам уже каким-то образом установлена, например, по факту свершения некоторого события с этим объектом в прошлом.

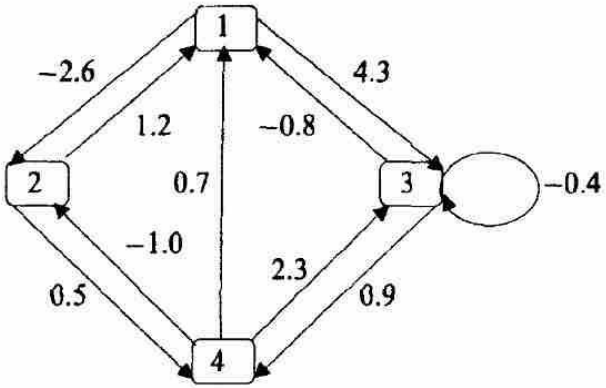
Идентификация :
- обучающая выборка – описания множеств объектов, входных стимулов и реакций на них;
- цель – полная модель системы в любых условиях ее деятельности.

Классификация (обучение с учителем):
- обучающая выборка – описания множества пар {стимул системы; ее ответная реакция};
- цель – модель прогноза реакции системы на любой входной стимул.

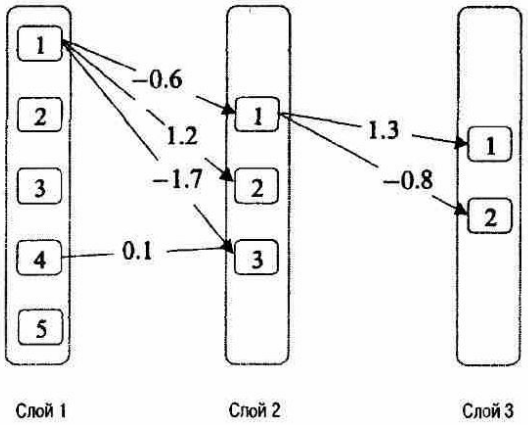
Кластеризация (обучение без учителя):
- обучающая выборка – описания множества объектов;
- цель - обнаружить внутренние взаимосвязи между объектами, зависимости, закономерности, существующие между ними, разбить на кластеры с одинаковыми свойствами.


Порог
b_k
x_1
w_{k1}
x_2
w_{k2}
x_m
w_{km}
Входные сигналы
Синаптические веса
Σ
Сумматор
v_k
Функция активации
φ(·)
Выходной сигнал
y_k



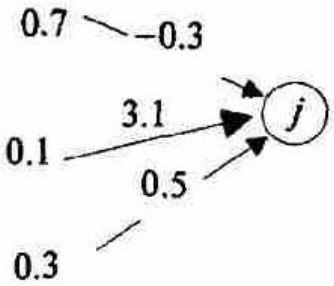


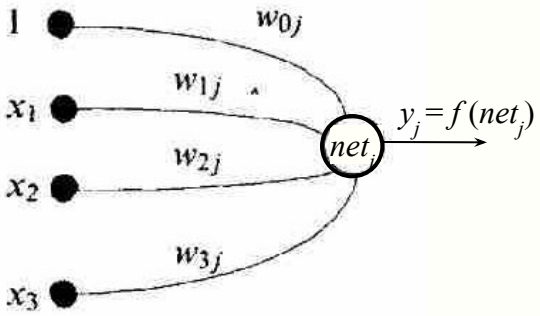
$$W = \begin{bmatrix} 0.0 & -2.6 & 4.3 & 0.0 \\ 1.2 & 0.0 & 0.0 & 0.5 \\ -0.8 & 0.0 & -0.4 & 0.9 \\ 0.7 & -1.0 & 2.3 & 0.0 \end{bmatrix}$$

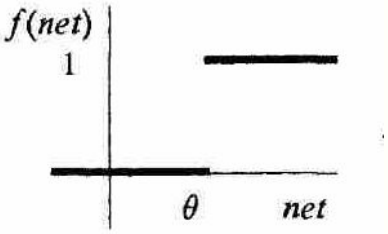


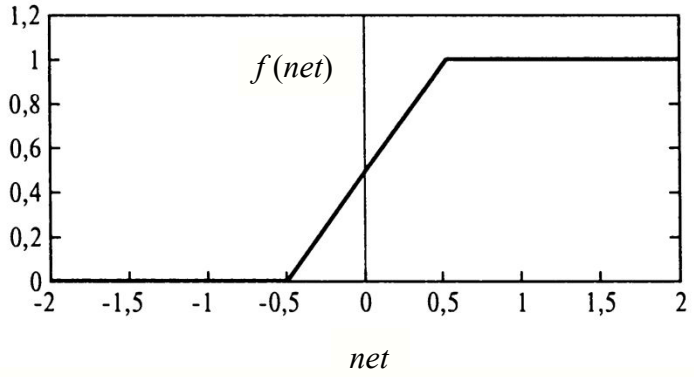
$$W_1 = \begin{bmatrix} -0.6 & 1.2 & -1.7 \\ . & . & . \\ . & . & . \\ . & . & 0.1 \\ . & . & . \end{bmatrix}$$

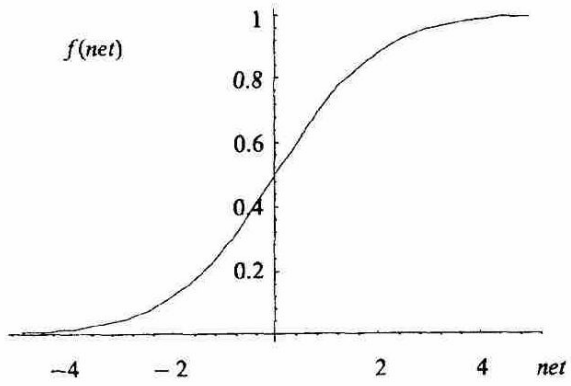
$$W_2 = \begin{bmatrix} 1.3 & -0.8 \\ . & . \\ . & . \end{bmatrix}$$




1
w_{0j}
w_{1j}
x_1
$y_j = f(net_j)$
net_j
w_{2j}
x_2
w_{3j}
x_3


$f(net)$
1
θ
net

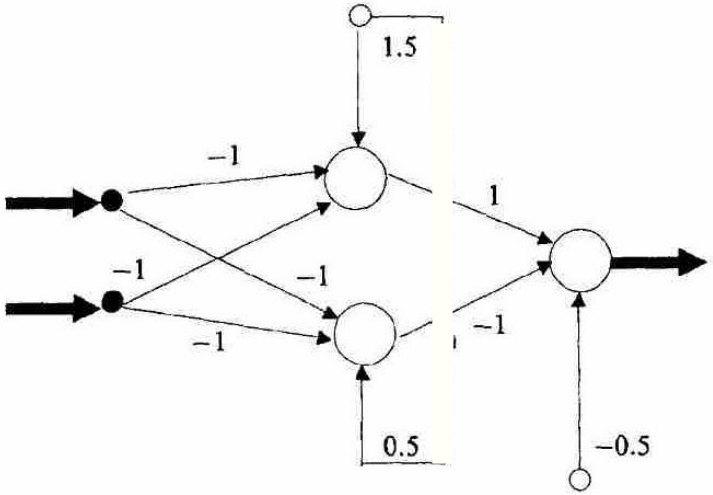

1,2
1
0,8
0,6
0,4
0,2
0
-2
-1,5
-1
-0,5
0
0,5
1
1,5
2
f(net)
net


1
0.8
0.6
0.4
0.2
f(net)
−4
−2
2
4
net

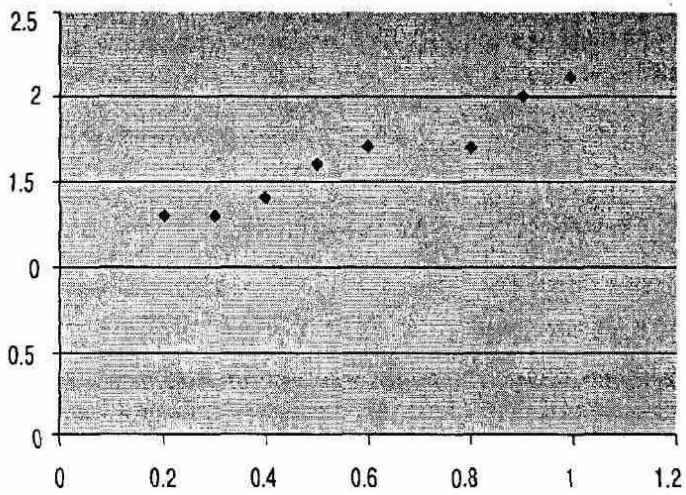
Нейронная сеть — это направленный граф, состоящий из узлов, соединенных синаптическими и активационными связями, который характеризуется следующими четырьмя свойствами.

1. *Каждый нейрон представляется множеством линейных синаптических связей, внешним порогом и, возможно, нелинейной связью активации. Порог, представляемый входной синаптической связью, считается равным* $+1$.
2. *Синаптические связи нейрона используются для взвешивания соответствующих входных сигналов.*
3. *Взвешенная сумма входных сигналов определяет индуцированное локальное поле каждого конкретного нейрона.*
4. *Активационные связи модифицируют индуцированное локальное поле нейрона, создавая выходной сигнал.*

Ввод		Вывод
x_1	x_2	
1	1	0
1	0	1
0	1	1
0	0	0




```
int XOR(int val_1, int val_2)
{
    if (val_1 == 1 && val_2 == 1)
        return 0;
    if (value_1 == 0 && val_2 == 0)
        return 0;
     if (val_1 == 1 && val_2 == 0)
         return 1;
     if (val_1 == 0 && val_2 == 1)
         return 1;
 }
```



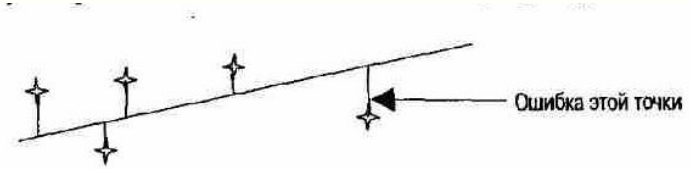


Рис. 1.14. Каждой точке соответствует своя ошибка, равная расстоянию от точки до прямой

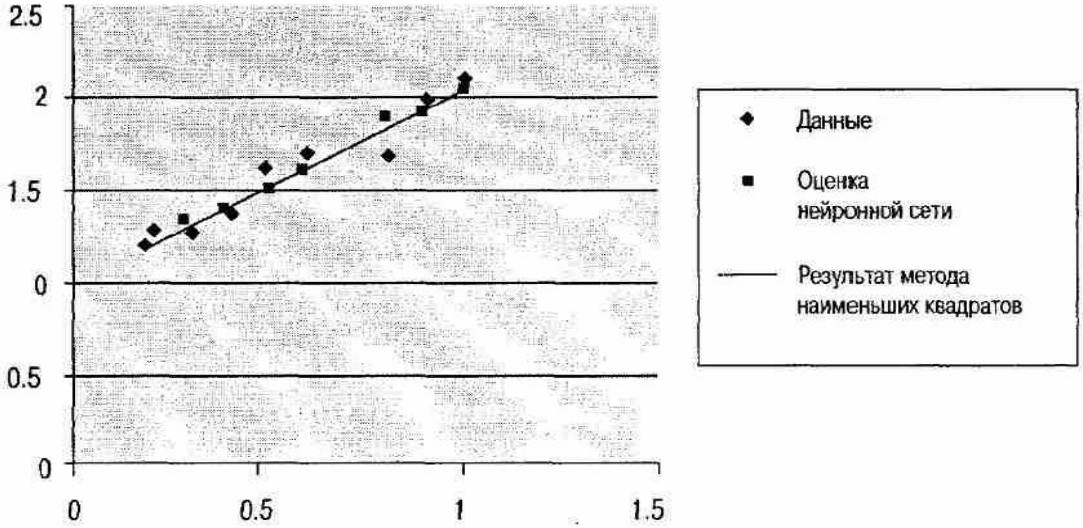


Рис. 1.16. Данные и соответствующая прямая, найденная с помощью нейронной сети, схема которой показана на рис. 1.15. Найденная нейронной сетью прямая мало отличается от прямой, получаемой при использовании метода наименьших квадратов

Задание

1. Для данных, представленных ниже, начертите на глаз несколько прямых, которые могут соответствовать этим данным. Запишите уравнения этих прямых, измерив соответствующие наклоны и координаты точек пересечения прямых с осью *y*. Для каждой прямой вычислите среднеквадратическую ошибку при условии, что для вводимых значений *x* вывод задается формулой

$$\text{вывод} = mx + c.$$

X	*Требуемый вывод*
0.30	1.60
0.35	1.40
0.40	1.40
0.50	1.60
0.60	1.70
0.80	2.00
0.95	1.70
1.10	2.10

2. Для данных из упражнения 1 найдите прямую, получаемую в результате применения метода наименьших квадратов.
3. Для данных упражнения 1 и заданных начальных весовых коэффициентов

$$\text{вывод} = 0.5x + 0.5$$

вычислите новую прямую после одного прохода через данные, используя правило обучения Видроу–Хоффа (дельта-правило) с нормой обучения, равной 0.3. (*Замечание*: после рассмотрения каждого учебного образца получается новая прямая.)
4. Наша подбирающая прямые линии нейронная сеть из раздела 1.4 имела один входной элемент, а целью было нахождение весовых коэффициентов, при которых по заданному x можно было бы оценить y.

Функция выбора решения:

ЕСЛИ вес > 0.80 И скорость < 0.55, ТО бомбардировщик.

ЕСЛИ вес < 0.90 И скорость > 0.25, ТО истребитель.

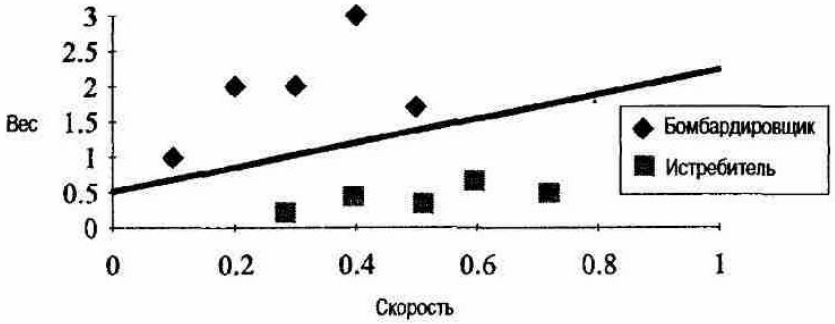


Рис. 2.1. Разделение абстрактных данных на два класса

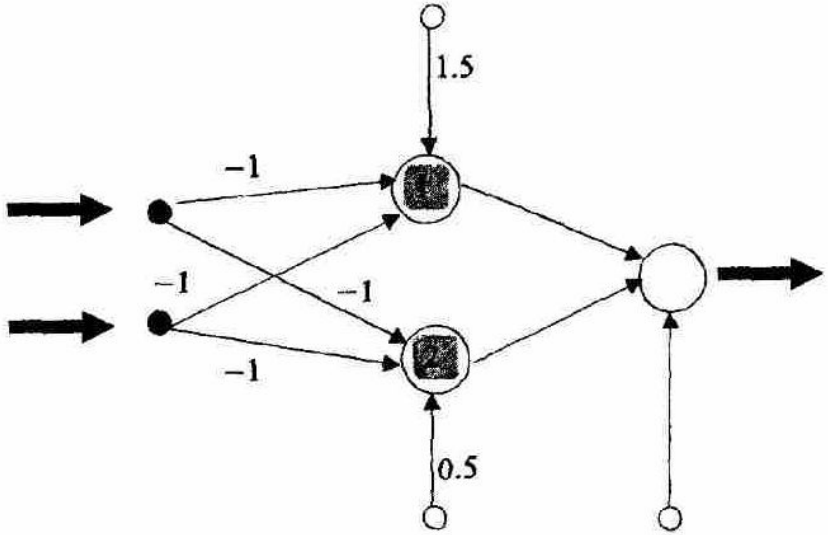
Пример 2.1

(а) Вычислите комбинированный сетевой ввод для элемента на рис. 2.2 и соответствующее выходное значение при использовании пороговой функции и входного вектора {0.7 2.5}.

(б) Вычислите выходное значение, использовав в качестве функции активности сигмоидальную функцию. Входной вектор остается таким же, как и в п. (а).

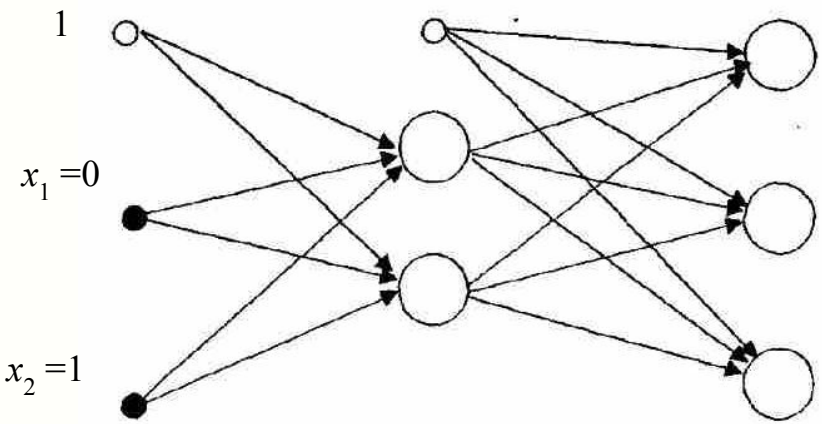
(в) Вычислите комбинированный ввод для сети с архитектурой, показанной на рис. 2.2, но с набором весовых значений {–0.2 0.03 1.2] и таким же входным вектором, как и в п. (а).

Ввод		Вывод
x_1	x_2	
1	1	0
1	0	1
0	1	1
0	0	0


1.5
−1
−1
−1
−1
0.5

Входной вектор, p	Первый слой весов	Реакция скрытого слоя		Второй слой весов	Реакция выходного слоя	
		Комбинированный ввод	Вывод		Ввод	Вывод
$\begin{bmatrix}1 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}1.5 & 0.5\\ -1.0 & -1.0\\ -1.0 & -1.0\end{bmatrix}$	$\begin{bmatrix}-0.5 & -1.5\end{bmatrix}$	$\begin{bmatrix}0 & 0\end{bmatrix}$	$\begin{bmatrix}-0.5\\ 1.0\\ -1.0\end{bmatrix}$	-0.5	0

Входной вектор, p	Первый слой весов	Реакция скрытого слоя		Второй слой весов	Реакция выходного слоя	
		Комбинированный ввод	Вывод		Ввод	Вывод
$\begin{bmatrix}1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}1.5 & 0.5\\ -1.0 & -1.0\\ -1.0 & -1.0\end{bmatrix}$	$\begin{bmatrix}0.5 & -0.5\end{bmatrix}$	$\begin{bmatrix}1 & 0\end{bmatrix}$	$\begin{bmatrix}-0.5\\ 1.0\\ -1.0\end{bmatrix}$	0.5	1
$\begin{bmatrix}1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}1.5 & 0.5\\ -1.0 & -1.0\\ -1.0 & -1.0\end{bmatrix}$	$\begin{bmatrix}0.5 & -0.5\end{bmatrix}$	$\begin{bmatrix}1 & 0\end{bmatrix}$	$\begin{bmatrix}-0.5\\ 1.0\\ -1.0\end{bmatrix}$	0.5	1
$\begin{bmatrix}1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}1.5 & 0.5\\ -1.0 & -1.0\\ -1.0 & -1.0\end{bmatrix}$	$\begin{bmatrix}1.5 & 0.5\end{bmatrix}$	$\begin{bmatrix}1 & 1\end{bmatrix}$	$\begin{bmatrix}-0.5\\ 1.0\\ -1.0\end{bmatrix}$	-0.5	0



$$\begin{bmatrix} 1.0 & -2.0 \\ 2.0 & 0.5 \\ -3.0 & 1.0 \end{bmatrix}$$

для первого слоя и

$$\begin{bmatrix} 2.0 & 1.0 & 3.0 \\ -1.0 & 5.0 & 4.0 \\ -3.0 & 1.0 & 2.0 \end{bmatrix}$$

для второго.

везде вывод равен вводу: $y_j = net_j$

(линейная тождественная функция преобразования)

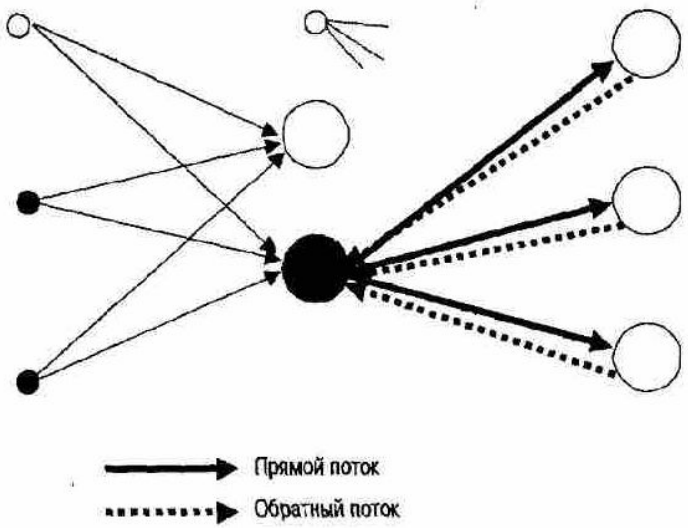
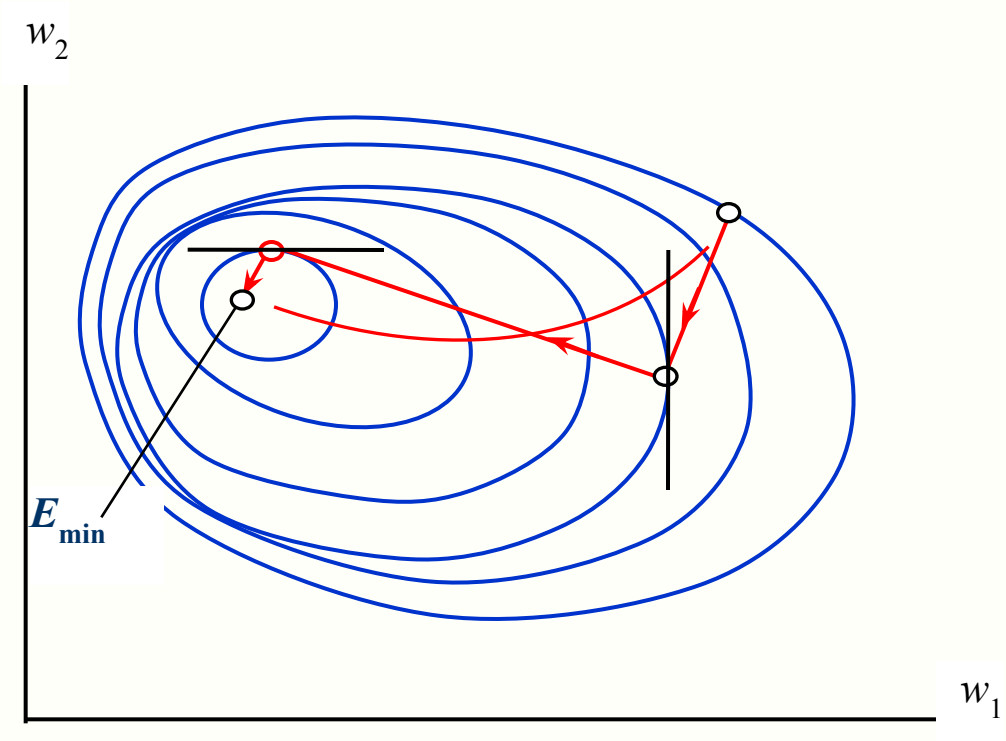


Рис. 2.10. Закрашенный скрытый элемент посылает сигнал активности каждому выходному элементу, поэтому в обратном потоке этот скрытый элемент получит сигналы ошибок от всех выходных элементов

для каждого входного вектора и связанного выходного вектора
выполнять, пока not STOP
    STOP = TRUE
    для каждого входного вектора
        выполнить прямой проход и найти реальный выход
        получить вектор ошибок путем сравнения реальных и целевых значений
        если реальный выход не попадает в допустимые рамки, установить STOP = FALSE
        выполнить обратный проход для вектора ошибок
        в результате обратного прохода определить величины изменения значений весов
        обновить значения весов

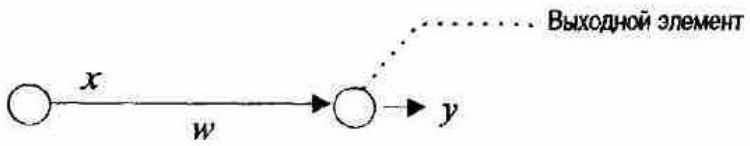

w_2
$E_{\min}$
w_1

Представьте полностью прямой и обратный проходы в сети с прямой связью, использующей алгоритм обратного распространения ошибок, для входного образца [0 1 0.9] и целевого выходного значения 0 9 в предположении, что сеть имеет архитектуру 2-2-1 (т.е. два входных, два скрытых один выходной элемент) с весовыми коэффициентами

$$y_j = f(net_j) = \frac{1}{1+\exp(-net_j)}$$

$$\left|\begin{array}{cc} 0.1 & 0.1 \\ -0.2 & -0.1 \\ 0.1 & 0.3 \end{array}\right|$$ - 1й слой

$$\left|\begin{array}{c} 0.2 \\ 0.2 \\ 0.3 \end{array}\right|$$ - для 2го слоя



$$y = mx + c \ ,$$

$m = 0.717$

$c = 1.239$

Пример 2.4

На рис. 2.12 показаны скрытый и выходной слои сети с прямой связью Вычислите ошибку для скрытого элемента U при условии, что значение его активности для обрабатываемого сетью образца равно 0 64.

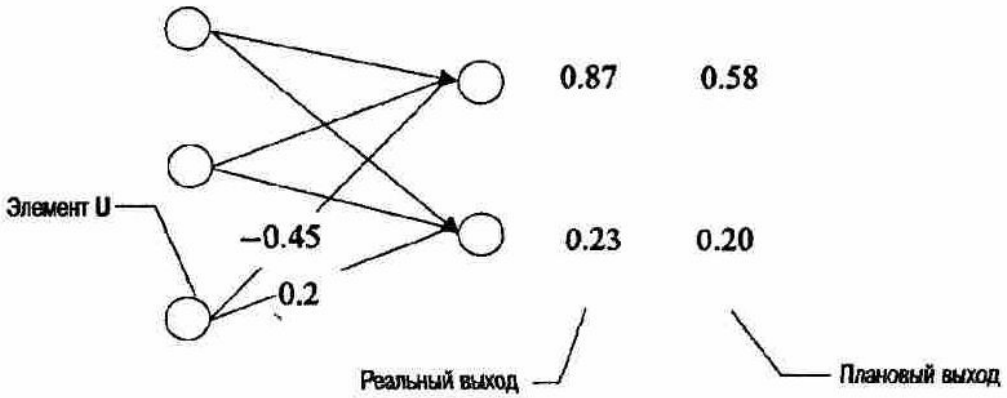


Рис. 2.12. Простая сеть Входной слой не показан

Сначала вычисляются ошибки для выходных элементов:

$$\delta_{\text{выход\_1}} = (0\,58 - 0\,87) \times 0\,87 \times (1 - 0.87)$$
$$= -0.033,$$

$$\delta_{\text{выход\_2}} = (0.20 - 0.23) \times 0.23 \times (1 - 0.23)$$
$$= -0.005.$$

Теперь ошибки распространяются обратно к элементу U:

$$\delta_U = 0.64 \times (1 - 0.64) \times [(-0.033 \times -0.45) + (-0.005 \times 0.20)]$$
$$= 0.003.$$

Пример 2.5

Сеть типа 2-2-1 с радиальными базисными функциями используется для решения проблемы XOR Первый слой весов задан матрицей

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Для каждого вводимого образца XOR вычислите значения активности для всех скрытых элементов, если функция активности имеет вид $\varphi(net) = \exp[-net^2]$, где net является евклидовой нормой

Для образца (0, 1) и первого скрытого элемента получаем

$$\varphi_1 = \exp[-[(0-1)^2 + (1-1)^2]] = \exp[-1] = 0\ 368$$

Для образца (0, 1) и второго скрытого элемента получаем

$$\varphi_2 = \exp[-[(0-0)^2 + (1-0)^2]] = \exp[-1] = 0\ 368$$

Весь набор значений активности выглядит следующим образом

Ввод	φ_1	φ_2
(0 1)	0 368	0 368
(1 0)	0 368	0 368
(0 0)	0 135	1
(1 1)	1	0 135

Если нанести значения активности невидимых элементов на плоскость, будет видно, что теперь образцы оказываются линейно отделимыми

широкого круга задач Давайте рассмотрим пример использования сети с радиальными базисными функциями для решения задачи аппроксимации кривой Кривая, которую нужно аппроксимировать, задается функцией

$$f(x) = 0\,5x + 2x^2 - x^3,$$

а ее график показан на рис 2 17

Эта кривая будет аппроксимироваться с помощью взвешенных сумм функций Гаусса, имеющих вид

$$net_j = \exp\left[-\frac{1}{2\sigma}(c - x)^2\right],$$

где с задает центр функции Пример кривой Гаусса, имеющей центр в нуле, показан на рис 2 18 От значения константы σ зависит ширина колокола кривой

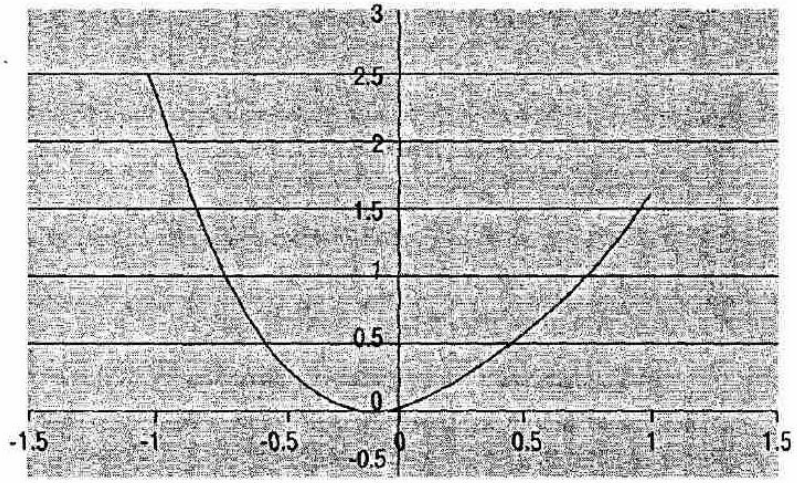


Рис. 2.17. График f(x) = 0.5x + 2x² - x³

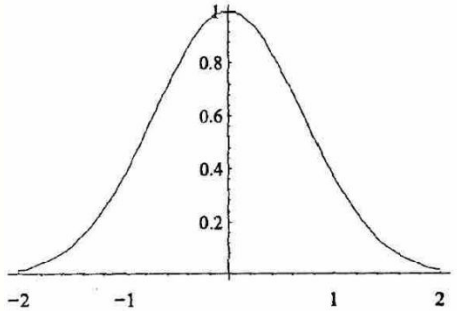
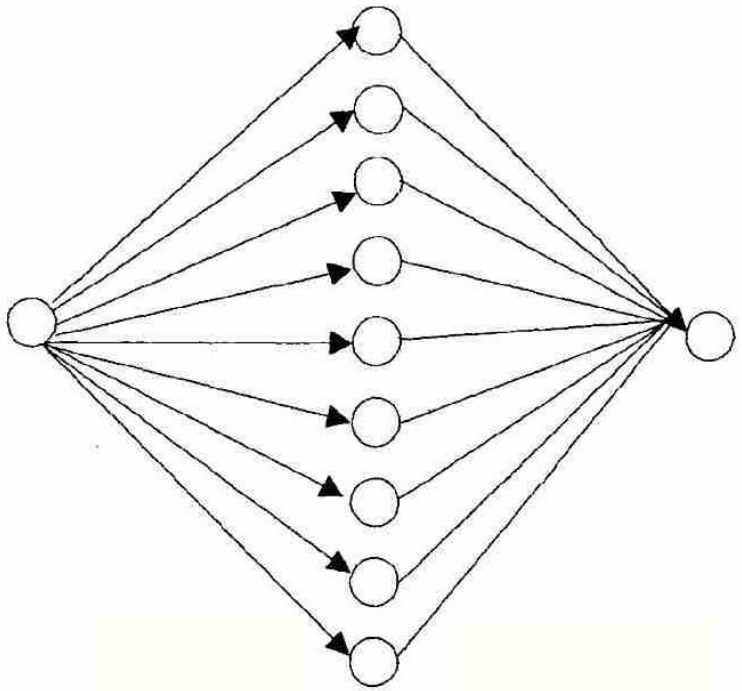


Рис. 2.18. Функция Гаусса



Число базисных функций выбирается произвольным образом. Были выбраны девять функций с центрами в точках {–0.8, –0.6, –0.4, –0.2, 0, 0.2, 0.4, 0.6, 0.8}. Ширина функций была выбрана равной 0.5. Кривая аппроксимируется взвешенной суммой базисных функций. Эти функции Гаусса представляют скрытый слой сети, поэтому скрытый слой состоит из девяти элементов. Первый слой весов, соответствующий связям, идущим от единственного входного элемента, представляет собой набор центров выбранных функций. Второй слой весов находится с помощью правила обучения Видроу–Хоффа, представленного в главе 1: для выхода o и целевого выхода t ошибка δ определяется формулой

$$\delta = t - o.$$

Суммарный сигнал, поступающий к выходному элементу, обозначен net. По закону обучения Видроу–Хоффа (дельта-правило) вносимая коррекция должна быть следующей:

$$\Delta w = \eta \delta net .$$

Архитектура сети показана на рис. 2.19. Норма обучения была выбрана равной 0.1. Чтобы найти второй слой весов, был сгенерирован 21 учебный образец путем пропускания некоторого числа (из диапазона между –1 и +1) через первый слой весов. Значения активности скрытого слоя затем давали вектор с девятью элементами для каждой учебной точки. Полученные в результате 21 вектор формировали набор учебных образцов для линейной сети, а соответствующие целевые выходные данные были значениями оригинальной кривой. Аппроксимация, построенная сетью, вместе с графиком оригинальной кривой показана на рис. 2.20.

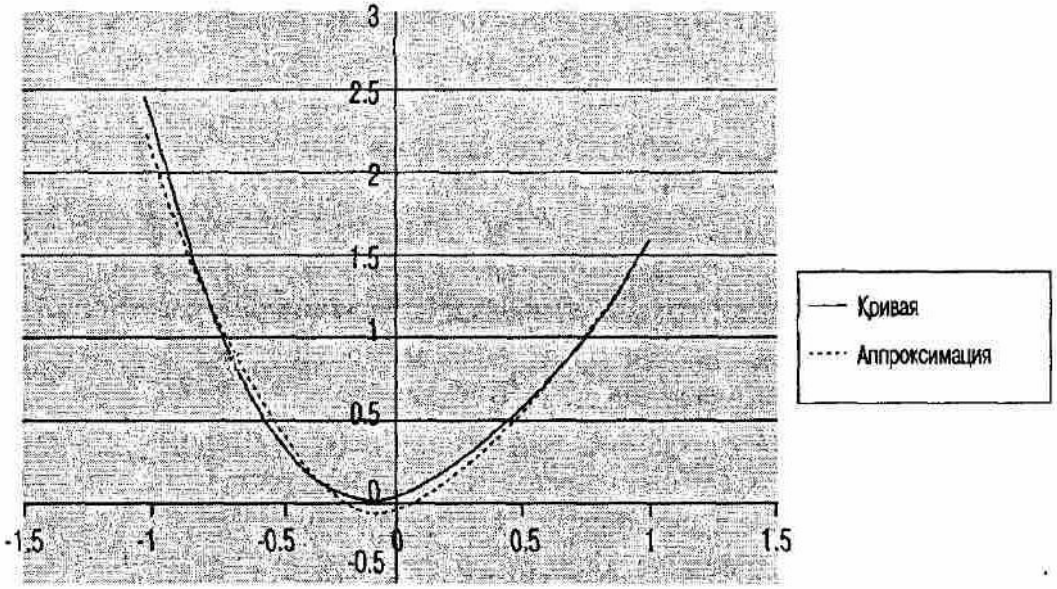


Рис. 2.20. Аппроксимация кривой с помощью сети с радиальными базисными функциями, представленной на рис. 2.19

данных. Рекомендуется выполнение следующего неравенства:

$$N > \frac{W}{\varepsilon},$$

где N обозначает число учебных образцов, W — число весовых коэффициентов в сети, а ε — долю ошибок, допустимую в ходе тестирования. Так, при допустимости 10% ошибок число учебных образцов должно быть в 10 раз больше числа имеющихся в сети весовых коэффициентов.

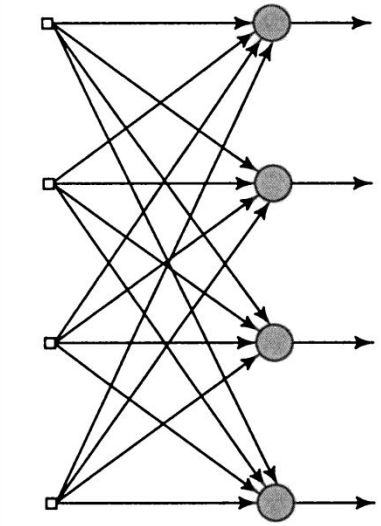
Если сеть порождает правильный вывод для большинства вводимых образцов из набора тестовых данных, говорят, что сеть обладает хорошими свойствами *обобщения*. Предполагается, что набор тестовых данных в процессе обучения не использовался.

Если сеть хорошо обучена строить гладкое нелинейное отображение, ее возможности можно интерполировать на новые образцы, которые подобны, но не повторяют в точности те образцы, которые использовались в процессе обучения. Негладкое отображение возникает тогда, когда сеть перетренирована. В такой ситуации сеть скорее будет подобна памяти, находя подходящие выходные данные среди множества учебных образцов.

*Качество обобщения сети зависит от набора учебных данных и архи*тектуры сети. Набор учебных данных должен быть характерным для решаемой проблемы, но важным также является и число скрытых элементов. Если скрытых элементов больше, чем требуется для обучения нужному отношению ввода-вывода, будет больше весовых коэффициентов, чем это необходимо, и, если процесс обучения продолжается слишком долго, в результате может наблюдаться слишком близкое отслеживание данных. Иногда в экспериментах с сетями различной архитектуры и про-

- Теоретически для моделирования многих проблем может быть достаточно одного скрытого слоя элементов, но на практике несколько скрытых слоев могут дать лучшее качество выполнения задачи.
- Сети с обратным распространением ошибок могут требовать длительного процесса обучения.

Структура нейронных сетей тесно связана с используемыми алгоритмами обучения.



Однослойные сети прямого распространения

В *многослойной* нейронной сети нейроны располагаются по слоям. В простейшем случае в такой сети существует *входной слой* (input layer) узлов источника, информация от которого передается на *выходной слой* (output layer) нейронов (вычислительные узлы), но не наоборот. Такая сеть называется сетью *прямого распространения* (feed-forward) или *ациклической* сетью (acyclic). На рис. 1.15 показана структура такой сети для случая четырех узлов в каждом из слоев (входном и выходном). Такая нейронная сеть называется *однослойной* (single-layer network), при этом под единственным слоем подразумевается слой вычислительных элементов (нейронов). При подсчете числа слоев мы не принимаем во внимание узлы источника, так как они не выполняют никаких вычислений.

Многослойные сети прямого распространения

Другой класс нейронных сетей прямого распространения характеризуется наличием одного или нескольких *скрытых слоев* (hidden layer), узлы которых называются *скрытыми нейронами* (hidden neuron), или *скрытыми элементами* (hidden unit). Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, мы можем выделить статистики высокого порядка.