

Теоретический материал к лабораторной работе №5

ДВУХСЛОЙНЫЙ ПЕРСЕПТРОН

1. Ограниченность однослойного персептрона

Как уже отмечалось ранее, Ф.Розенблатту удалось обучить свой персептрон распознавать буквы алфавита. Это был колоссальный успех: *Электронное устройство, созданное по образу и подобию человеческого мозга, обученное подобно человеку, успешно моделировало интеллектуальные функции человека.* Это был успех в познании самой природы человеческого мышления. Мозг начал раскрывать свои тайны. Появилась возможность исследовать мозг методами моделирования, не прибегая к сложнейшим антигуманным и мало что дающим натурным экспериментам. Это была сенсация, приковавшая к себе внимание мыслящих людей всего мира. Казалось, что ключ к интеллекту был найден и полное воспроизведение человеческого мозга и всех его функций – всего лишь вопрос времени. Писателям-фантастам, ученым, инженерам, бизнесменам, политикам виделись самые радужные перспективы практического применения идей искусственного интеллекта. Правительство Соединенных Штатов Америки выделило крупные субсидии на развитие нового перспективного научного направления.

Класс решаемых нейросетями задач расширялся. Но по мере расширения фронта научных исследований появлялись трудности. Неожиданно оказалось, что многие новые задачи персептрон решить не мог. Причем эти новые задачи, внешне ничем не отличались от тех, с которыми персептрон успешно справлялся ранее. Возникла необходимость объяснения парадоксов, глубокого анализа и создания теоретической базы нейроинформатики.

Следующий период истории искусственного интеллекта начался с появления в 1969 г. книги двух известных американских математиков *М.Минского* и *С.Пайперта* «Персептроны». Авторы этой книги математически строго доказали, что использовавшиеся в то время однослойные персептроны в принципе не способны решать многие простые задачи. Одну из таких задач, вошедшую в историю нейроинформатики под названием проблемы «Исключающего ИЛИ», мы рассмотрим подробно.

«Исключающее ИЛИ» – это логическая функция двух аргументов, каждый из которых может иметь значение «истинно» либо «ложно». Сама она принимает значение «истинно», когда только один из аргументов имеет значение «истинно». Во всех остальных случаях эта функция принимает значение «ложно». Если закодировать значение «истинно» единицей, а значение

«ложно» – нулем, то требуемое соответствие между аргументами x_1 , x_2 и самой функцией y можно представить в виде табл. 1, называемой таблицей истинности логической функции.

Т а б л и ц а 1

Таблица истинности логической функции «Исключающее ИЛИ»

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

Задача состоит в том, чтобы научиться моделировать функцию «Исключающее ИЛИ» с помощью однопейронного персептрона с двумя входами x_1 и x_2 и одним выходом y (рис. 1). При выполнении лабораторной работы №1 Вы уже пытались решить эту задачу путем подбора значений синаптических весов w_1 , w_2 и порога θ , однако сделать это Вам не удалось. Вам не удалось это сделать, хотя в других случаях, при моделировании логических функций «И» и «ИЛИ», у Вас проблем не возникало.

Внешне функции «И», «ИЛИ» и «Исключающее ИЛИ» мало чем отличаются друг от друга, и Вам было не понятно, почему Ваш однопейронный персептрон успешно справлялся с моделированием двух первых функций, а с моделированием третьей функции он справиться не мог.

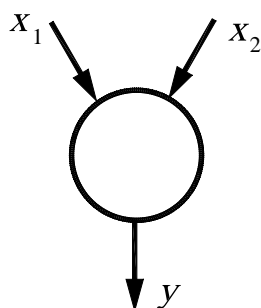


Рис. 1. Однопейронный персептрон с двумя входами и одним выходом

Для объяснения этого парадокса американскими математиками М.Минским и С.Пайпертом была предложена геометрическая интерпретация, состоящая в следующем. Они предложили изобразить на координатной плоскости x_1 , x_2 все возможные комбинации входных сигналов в виде четырех

точек: A, B, C, D , как показано на рис. 2. Точка A имеет координаты $x_1 = 0, x_2 = 0$; точка B имеет координаты $x_1 = 0, x_2 = 1$ и т.д. согласно табл. 2.

Т а б л и ц а 2

**Таблица истинности логической функции «Исключающее ИЛИ»,
дополненная точками A, B, C, D**

Точки	x_1	x_2	y
A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

Тогда в точке A выход персептрона y должен быть равен нулю, в точке B – единице, в точке C – единице и в точке D – нулю.

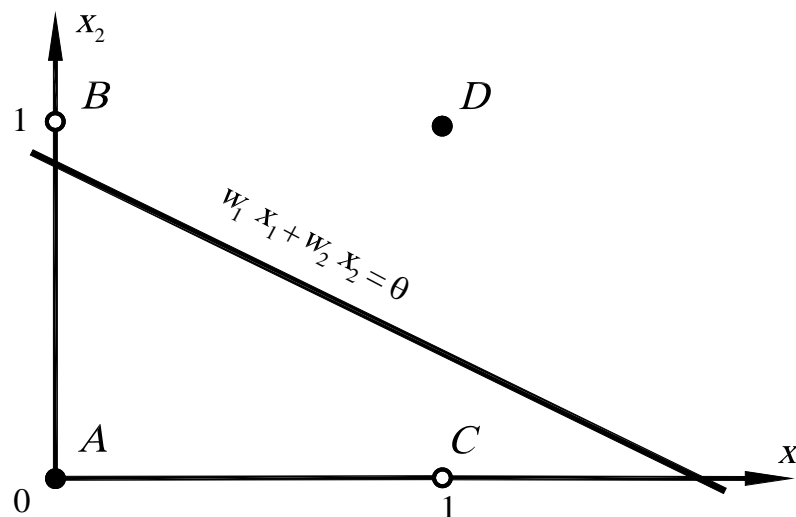


Рис. 2. Графическая интерпретация к объяснению проблемы
«Исключающего ИЛИ»

Как нам известно (см. например лабораторную работу №1), математический нейрон Мак-Каллока – Питтса, изображенный на рис. 1, осуществляет преобразование

$$S = w_1 x_1 + w_2 x_2; \quad (1)$$

$$y = \begin{cases} 1, & \text{если } S \geq \theta; \\ 0, & \text{если } S < \theta, \end{cases} \quad (2)$$

Заменим в уравнении (1) S на θ :

$$w_1x_1 + w_2x_2 = \theta. \quad (3)$$

Если в этом уравнении величины x_1 и x_2 считать переменными, а θ , w_1 и w_2 — константами, то на координатной плоскости x_1 , x_2 рассматриваемое уравнение изобразится в виде прямой линии, положение и наклон которой определяются значениями коэффициентов w_1 , w_2 и порога θ . Для всех точек плоскости x_1 , x_2 , лежащих на этой линии, выполняется равенство $S = \theta$ и поэтому, согласно формуле (2), выход персептрона равен единице. Для точек, лежащих выше указанной линии сумма $w_1x_1 + w_2x_2$ больше чем θ , и поэтому по формулам (1)-(2) выход персептрона также равен единице, а для точек, лежащих ниже этой линии, сумма $w_1x_1 + w_2x_2$ меньше чем θ , и выход персептрона равен нулю. Поэтому линию, изображающую уравнение (3), называют *пороговой прямой*.

А теперь посмотрим на табл. 2. Согласно этой таблице в точках A и D выход персептрона должен быть нулевым, а в точках B и C — единичным. Но для этого надо расположить пороговую прямую так, чтобы точки A и D лежали ниже этой линии, а точки B и C — выше, что невозможно. Это значит, что, сколько бы персептрон ни обучали, какие бы значения ни придавали его синаптическим весам и порогу, персептрон в принципе не способен воспроизвести соотношение между входами и выходом, требуемое таблицей истинности функции «Исключающее ИЛИ».

Помимо проблемы «Исключающего ИЛИ» М.Минский и С.Пайперт привели ряд других задач, в которых точки, изображающие входные сигналы, не могут быть разделены пороговой прямой (в многомерных случаях — плоскостью, гиперплоскостью). Такие задачи получили название *линейно неразделимых*.

После выхода в свет книги М.Минского и С.Пайперта «Персептроны» всем стало ясно, что активно предпринимавшиеся в то время попытки обучать персептроны решению многих задач, которые, как оказалось, относятся к классу линейно неразделимых, с самого начала были обречены на провал. Это была пустая трата времени, сил и финансовых ресурсов.

----- Коротко о главном -----

Однонеуронный персептрон в принципе не позволяет моделировать логическую функцию «Исключающее ИЛИ» и решать другие линейно неразделимые задачи.

2. Решение проблемы «Исключающего ИЛИ»

Появление книги М.Минского и С.Пайперта «Персептроны» вызвало шок в научном мире. Строгие математические доказательства М.Минского и С.Пайперта были неуязвимы. Всеобщий энтузиазм сменился не менее всеобщим пессимизмом. В газетах стали появляться критические статьи с сообщениями о том, что ученые мужи в своих исследованиях зашли в тупик, впустую израсходовав огромные государственные деньги. Правительство США немедленно прекратило финансирование нейропроектов и приступило к поискам виновных. Бизнесмены, потерявшие надежду вернуть вложенные капиталы, отвернулись от ученых и нейроинформатика была предана забвению, длившемуся более 20 лет.

Тем не менее, работы в области нейросетевых и нейрокомпьютерных технологий продолжались отдельными энтузиастами. Работы продолжались в засекреченных научно-исследовательских институтах Советского Союза, отделенного в то время от Запада «железным занавесом». Не имея информации о настроениях зарубежных коллег, советские ученые спокойно продолжали заниматься захватившей их умы темой и к началу 80-х гг. удивили мир ракетами и самолетами, управлявшимися компьютерами нового поколения – нейрокомпьютерами. Советские компьютеры, в отличие от американских, стойко переносили довольно серьезные повреждения, продолжая работать в сложных условиях, что было особенно важно для объектов военного назначения. Выявилось еще одно свойство нейрокомпьютеров, унаследованное от мозга – свойство *живучести*.

Советским ученым *С.О.Мкртчяном* было показано, что с помощью многослойных персептронов может быть смоделирована любая логическая функция, если только известна ее логическая формула. Более того, им был разработан специальный математический аппарат, позволяющий конструировать такие персептроны. Оказалось, что проблема «Исключающего ИЛИ», явившаяся камнем преткновения для однейронного персептрона, может быть разрешена с помощью нейронной сети, состоящей из трех нейронов – технейронного персептрона, изображенного на рис. 3.

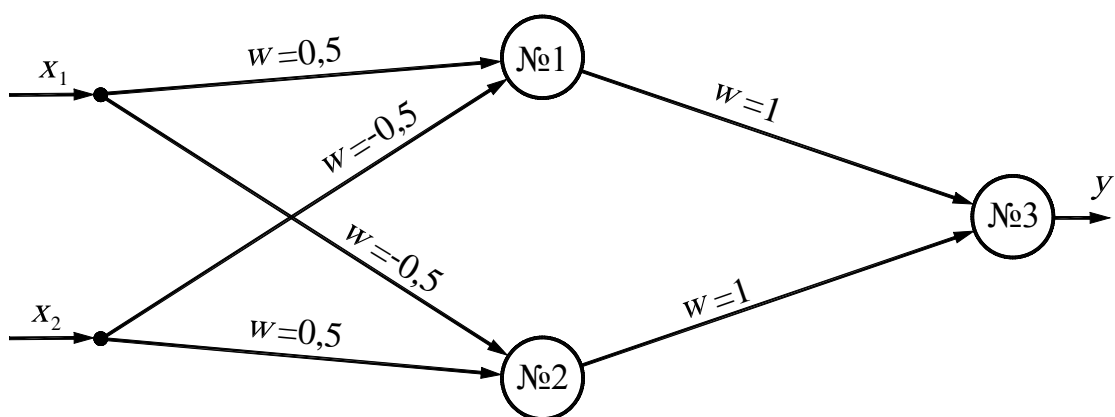


Рис. 3. Нейронная сеть, моделирующая функцию
«Исключающее ИЛИ»

Работа этого персептрона происходит по следующему алгоритму.

Нейрон № 1:

$$S_1 = 0,5 \times x_1 + (-0,5) \times x_2;$$

$$y_1 = \begin{cases} 1, & \text{если } S_1 \geq \theta; \\ 0, & \text{если } S_1 < \theta. \end{cases}$$

Нейрон № 2:

$$S_2 = (-0,5) \times x_1 + 0,5 \times x_2;$$

$$y_2 = \begin{cases} 1, & \text{если } S_2 \geq \theta; \\ 0, & \text{если } S_2 < \theta. \end{cases}$$

Нейрон № 3:

$$S_3 = 1 \times y_1 + 1 \times y_2;$$

$$y_3 = \begin{cases} 1, & \text{если } S_3 \geq \theta; \\ 0, & \text{если } S_3 < \theta. \end{cases}$$

Задавшись значением порога $\theta = 0,5$ и заполнив с помощью этих формул табл. 3, легко убедиться, что трехнейронный персептрон успешно моделирует функцию «Исключающее ИЛИ».

Т а б л и ц а 3

Процесс формирования сигналов в трехнейронном персептроне

x_1	x_2	S_1	S_2	y_1	y_2	S_3	y_3	y
0	0	0	0	0	0	0	0	0
0	1	-0,5	0,5	0	1	1	1	1
1	0	0,5	-0,5	1	0	1	1	1

1	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Впоследствии было показано, что и другие линейно неразделимые задачи, приведенные в книге М.Минского и С.Пайперта, могут быть решены с помощью нейросетей, содержащих один или несколько *скрытых* нейронных слоев, т.е. слоев нейронов, расположенных между входным и выходным слоями.

Многие исследователи понимали, что нужно создавать нейросети более сложной архитектуры, содержащие скрытые слои нейронов, но не представляли, как такие сети обучать. Правила Хебба и их обобщение – дельта-правило, годились только для корректировки синаптических весов нейронов выходного слоя, тогда как вопрос о настройке параметров скрытых нейронных слоев оставался открытым.

----- Коротко о главном -----

Логическую функцию «Исключающее ИЛИ» может моделировать нейронная сеть, состоящая из трех нейронов, изображенная на рис. 3.

3. Алгоритм обратного распространения ошибки

Эффективный алгоритм обучения многослойных персептронов, открывший путь их широкому практическому применению, стал известен только в 1986 г. благодаря публикациям *Д.Румельхарта*, *Г.Хилтона* и *Р.Вильямса*. Идея этого алгоритма заключается в том, что ошибки нейронов выходного слоя $\varepsilon_i = d_i - y_i$ используются для вычисления ошибок нейронов, расположенных в скрытых слоях. Значения ошибок как бы распространяются от выходного слоя нейронов вовнутрь сети от последующих нейронных слоев к предыдущим. Отсюда название метода *алгоритмом обратного распространения ошибки (back propagation)*.

Интересно отметить, что алгоритм обратного распространения ошибки был предложен на один год ранее в работах *А.Паркера* и *А.Ле-Кана*, изданных независимо одна от другой. Более того, еще в 1974 г. этот простой и изящный алгоритм был защищен *П.Вербосом* в его докторской диссертации. Однако тогда он остался незамеченным, и только спустя более десяти лет был «переоткрыт» заново и получил всеобщее признание и применение. Ос-

тались незамеченными и работы советских ученых, еще раньше разрабатывавших подобные алгоритмы в своих засекреченных институтах и успешно применявших их при построении систем управления объектами военного назначения.

Рассмотрим идею алгоритма обратного распространения ошибки, попытавшись обобщить дельта-правило на случай обучения *двухслойного* персептрона, имеющего N входов, I выходов и *скрытый* слой из J нейронов (рис. 4). Этот персептрон на самом деле имеет три слоя, однако в литературе его называют двухслойным, поскольку нейроны входного слоя имеют всего один вход, не имеют синаптических весов и не выполняют суммирования входных сигналов, а лишь передают один единственный входной сигнал нейронам следующего слоя.

Алгоритм корректировки синаптических весов нейронов выходного слоя оставим таким же, как для однослойного персептрона (см. обобщенное дельта-правило: теоретический материал к лабораторной работе №4), заменив x_j на y_j :

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}; \quad (4)$$

$$\Delta w_{ij} = \eta \delta_i y_j; \quad (5)$$

$$\delta_i = y_i(1 - y_i)(d_i - y_i). \quad (6)$$

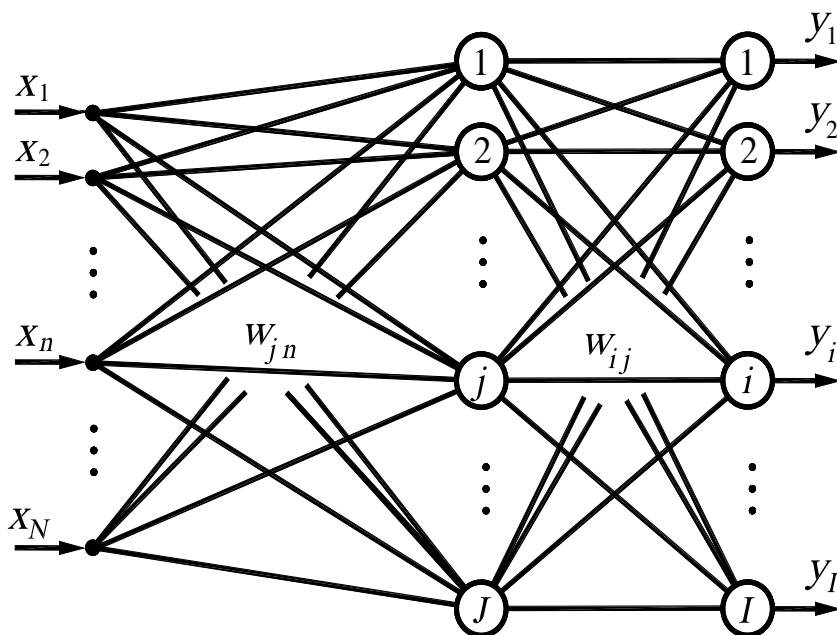


Рис. 4. Двухслойный персептрон с N входами, I выходами и скрытым слоем из J нейронов

Синаптические веса нейронов скрытого слоя попытаемся корректировать с помощью все тех же формул, в которых индекс i заменим на j , а индекс j заменим на индекс n :

$$\Delta w_{jn} = \eta \delta_j x_n; \quad (7)$$

$$\delta_j = y_j(1 - y_j)(d_j - y_j). \quad (8)$$

При использовании этих формул возникает вопрос о вычислении нейронной ошибки $(d_j - y_j)$, которая для скрытого слоя неизвестна. Идея авторов рассматриваемого алгоритма состояла в том, чтобы в качестве этой ошибки использовать суммарные нейронные ошибки с выходного слоя, помноженные на силы соответствующих синаптических связей, т.е.

$$(d_j - y_j) = \sum_{i=1}^I \delta_i w_{ij}. \quad (9)$$

Итак, для скрытого слоя окончательно имеем

$$\Delta w_{jn} = \eta \delta_j x_n; \quad (10)$$

$$\delta_j = y_j(1 - y_j) \sum_{i=1}^I \delta_i w_{ij}. \quad (11)$$

Используя эту идею, несложно расписать алгоритм обратного распространения ошибки для обучения персептрона, имеющего произвольное количество скрытых слоев. Однако прежде отметим, что мы будем использовать нейроны, имеющие сигмоидную активационную функцию (см. теоретический материал к лабораторной работе №4), и выполняющие операцию суммирования с учетом нейронного смещения (см. теоретический материал к лабораторной работе №1):

$$S_i = \sum_{j=0}^J w_{ij} x_j. \quad (12)$$

Здесь w_{i0} – вес дополнительного входа i -го нейрона, имитирующий его смещение b_i , а $x_0 = 1$ – величина сигнала дополнительного входа.

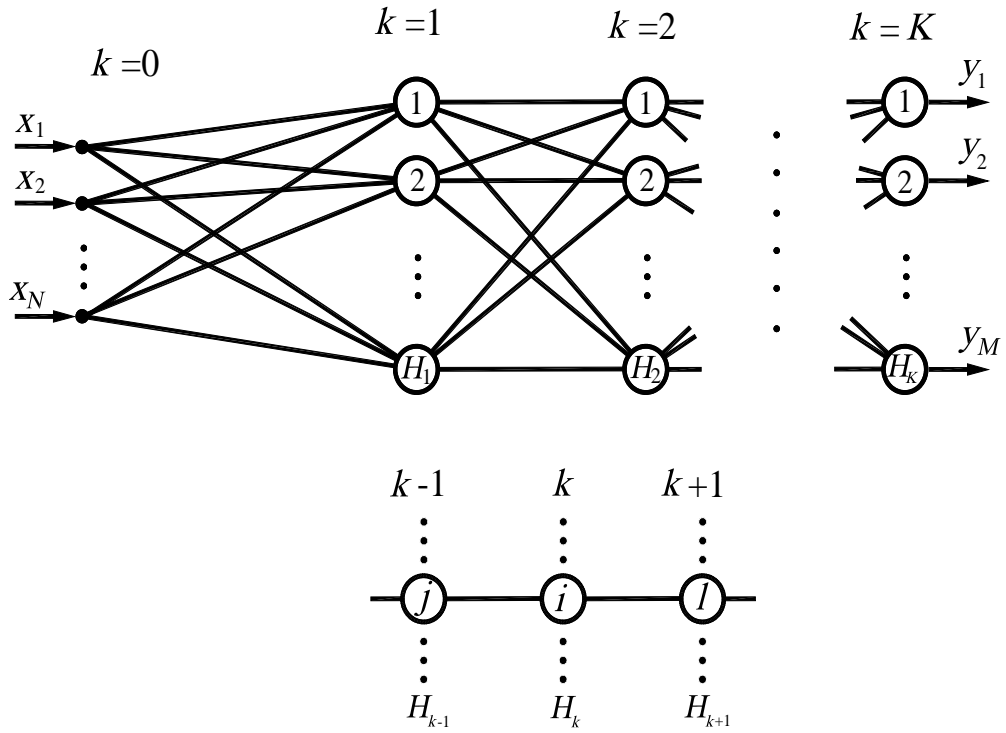


Рис. 5. Многослойный персептрон (MLP – MultiLayerPerseptron)

Алгоритм обратного распространения ошибки распишем для многослойного персептрона, имеющего входной слой $k=0$, несколько скрытых слоев $k=1, 2, \dots, K-1$ и выходной слой $k=K$ (рис. 5).

Нейроны входного слоя не выполняют математических преобразований, а лишь передают входные сигналы нейронам первого слоя. Будем полагать, что каждый k -й слой содержит H_k нейронов. Таким образом, персептрон имеет $N = H_0$ входов и $M = H_K$ выходов. В алгоритме будем использовать следующие обозначения: i – порядковый номер нейрона k -го слоя; j – порядковый номер нейрона $(k-1)$ -го слоя; l – порядковый номер нейрона $(k+1)$ -го слоя (см. рис. 5, внизу).

Шаг 1. Инициализация синаптических весов и смещений.

В циклах по $k=1, 2, \dots, K$; $i=1, 2, \dots, H_k$; $j=0, 1, 2, \dots, H_{k-1}$ синаптическим весам и смещениям $w_{ij}^{(k)}$ датчиком случайных чисел присваиваются малые величины, например, из интервала от -1 до 1 .

Шаг 2. Открытие цикла по $q=1, 2, \dots, Q$. Представление из обучающего множества примеров очередного входного вектора $\mathbf{X}_q = (x_1, x_2, \dots, x_N)_q$ и соответствующего ему желаемого выходного вектора $\mathbf{D}_q = (d_1, d_2, \dots, d_M)_q$, где q – номер примера в обучающем множестве.

Шаг 3. Прямой проход.

В циклах по $k = 1, 2, \dots, K$; $i = 1, 2, \dots, H_k$ вычисляются выходные сигналы i -го нейрона в k -м слое

$$y_i^{(k)} = f_{\sigma} \left(\sum_{j=0}^{H_{k-1}} w_{ij}^{(k)} y_j^{(k-1)} \right), \quad (13)$$

где $y_j^{(0)} = x_j$; $x_0 = 1$; $y_0^{(k-1)} = 1$; выходные сигналы персептрона $y_i = y_i^{(K)}$.

Шаг 4. Обратный проход.

В циклах по $k = K, K-1, \dots, 1$; $i = 1, 2, \dots, H_k$; $j = 0, 1, 2, \dots, H_{k-1}$ вычисляются синаптические веса на новой эпохе

$$w_{ij}^{(k)}(t+1) = w_{ij}^{(k)}(t) + \Delta w_{ij}^{(k)}, \quad (14)$$

где

$$\Delta w_{ij}^{(k)} = \eta \delta_i^{(k)} y_j^{(k-1)}, \quad (15)$$

причем для выходного слоя $k = K$ согласно (8)

$$\delta_i^{(K)} = y_i(1 - y_i)(d_i - y_i),$$

а для всех других скрытых слоев согласно (11)

$$\delta_i^{(k)} = y_i^{(k)}(1 - y_i^{(k)}) \sum_{l=1}^{H_{k+1}} \delta_l^{(k+1)} w_{li}^{(k+1)}.$$

Шаг 5. Закрытие цикла по q .

Шаг 6. Повторение шагов 2 – 5 необходимое количество раз.

Векторы обучающих примеров \mathbf{X}_q и \mathbf{D}_q на шаге 2 алгоритма обычно представляются последовательно от первого до последнего, т.е. $q = 1, 2, \dots, Q$, где Q – общее количество примеров. Например, в случае распознавания букв русского алфавита $Q = 33$. После того, как для каждого обучающего примера будут скорректированы весовые коэффициенты персептрона, т.е. шаги 2–4 будут повторены 33 раза, на шаге 6 алгоритма вычисляется средне-квадратичная ошибка, усредненная по всем обучающим примерам:

$$\varepsilon = \frac{1}{QM} \sum_{q=1}^Q \sum_{i=1}^M ((d_i - y_i)_q)^2. \quad (16)$$

Помимо среднеквадратичной ошибки может быть также оценена максимальная разность между желаемым и прогнозным (то, что вычислил персептрон) выходами персептрона:

$$\varepsilon = \max(|d_i - y_i|_q); \quad i = 1, 2, \dots, M; \quad q = 1, 2, \dots, Q. \quad (17)$$

Итерационный процесс, задаваемый *шагом 6*, заканчивается после того, как ошибка ε , вычисляемая по формулам (16) или (17), достигнет заданной величины, либо когда будет достигнуто предельное количество эпох обучения. В результате персептрон обучится выполнять нужное отображение любого входного вектора X_q на выходной вектор Y_q , отличающийся от желаемого вектора D_q на некоторую малую величину.

----- Коротко о главном -----

Первым алгоритмом обучения нейронной сети были правила Хебба, предназначенные для обучения однослойного персептрона с нейронами, имеющими ступенчатые активационные функции. Затем было введено понятие нейронной ошибки как разницы между требуемым выходом нейрона d_i и его реальным значением y_i . В результате алгоритм обучения персептрона с помощью правил Хебба был обобщен в виде алгоритма дельта-правила. В итерационных формулах алгоритма дельта-правила появился коэффициент скорости обучения η , позволяющий влиять на величину итерационного шага. Затем была предложена сигмоидная активационная функция и было введено понятие квадратичной ошибки обучения персептрона. В результате появилось обобщенное дельта-правило, реализующее метод градиентного спуска и позволяющий работать не только с бинарными, но и с непрерывными сигналами. Алгоритм обратного распространения ошибки является следующим обобщением обобщенного дельта-правила и позволяет обучать не только однослойные, но и многослойные персептроны.

4. Виды активационных функций

В современных нейронных сетях и нейропакетах наиболее часто применяются следующие виды активационных функций.

Пороговые активационные функции (функции-ступеньки). Пороговые активационные функции-ступеньки могут иметь как несимметричный (рис. 6, *а*), так и симметричный (рис. 6, *б*) относительно начала координат вид.

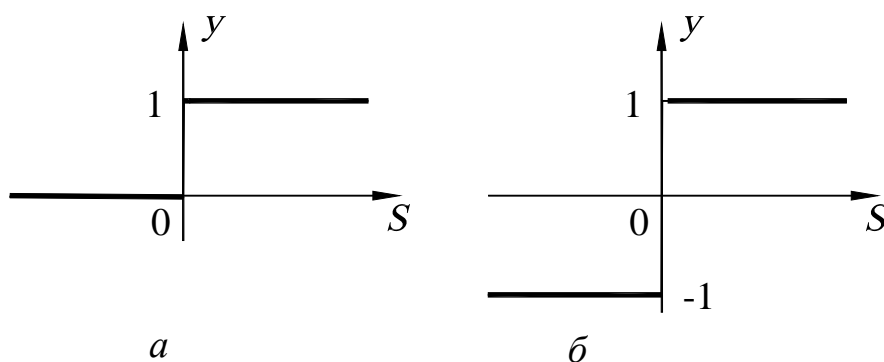


Рис. 6. Пороговые активационные функции-ступеньки

Их аналитическое представление соответственно для рис. 6,*а* и рис. 6,*б*:

$$y = \begin{cases} 1, & \text{если } S \geq 0; \\ 0, & \text{если } S < 0; \end{cases} \quad y = \begin{cases} 1, & \text{если } S \geq 0; \\ -1, & \text{если } S < 0, \end{cases}$$

где $S = \sum_{i=0}^I x_i w_i$, причем $x_0 = 1$ – величина сигнала дополнительного входа, а w_0 – его вес, имитирующий нейронное смещение b (которое равно порогу чувствительности нейрона, взятому с противоположным знаком: $b = -\theta$).

Ступенчатые активационные функции обычно используются в задачах распознавания образов. Персептроны со ступенчатыми активационными функциями могут обучаться с помощью правил Хебба и дельта-правила. Обобщенное дельта правило и алгоритм обратного распространения ошибки для обучения таких персептронов не годятся, так как в эти алгоритмы включают нахождение производных от активационных функций, что невозможно для функций, имеющих разрыв.

Линейные активационные функции. На рис. 7,*а* представлен график линейной активационной функции

$$y = S.$$

Область изменения этой функции неограниченна.

Такие активационные функции обычно применяются в нейронах входного слоя. Они также неплохо работают при решении простых линейно разделимых задач, причем с обучением таких персептронов могут справляться алгоритмы дельта-правила, обобщенного дельта-правила и алгоритм обратного распространения ошибки.

Иногда применяют линейные активационные функции с ограниченной областью изменения (рис. 7, *б*):

$$y = \begin{cases} -1, & \text{если } S < -1 \\ S, & \text{если } -1 \leq S \leq 1 \\ 1, & \text{если } S > 1 \end{cases}$$

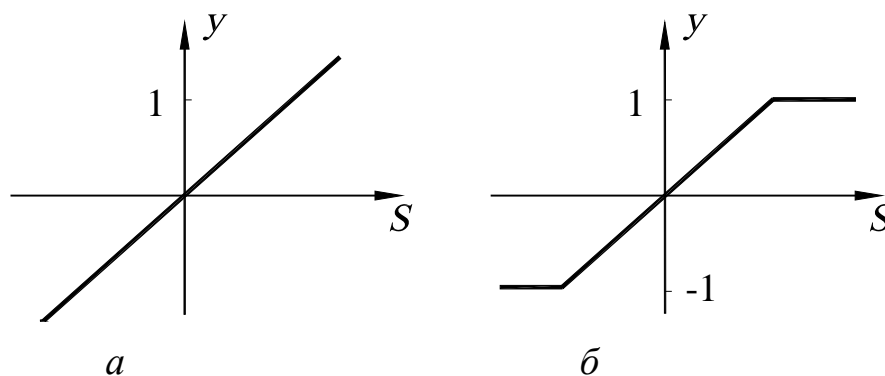


Рис. 7. Линейные активационные функции с неограниченной (а) и ограниченной (б) областями изменения

Сигмоидные активационные функции. На рис. 8,а изображен график сигмоидной функции, заданной уравнением

$$y = \frac{1}{1 + e^{-\alpha S}},$$

а на рис. 8,б – график функции, заданной уравнением

$$y = \frac{1 - e^{-\alpha S}}{1 + e^{-\alpha S}}.$$

В этих уравнениях коэффициент α влияет на угол наклона линий к оси S .

Аналогичный представленному на последнем графике вид имеют функции арктангенса $y = \frac{2}{\pi} \arctg \alpha S$ и гиперболического тангенса $y = \tanh \alpha S$,

а также функция $y = \frac{\alpha S}{1 + |\alpha S|}$, которые тоже называют сигмоидными.

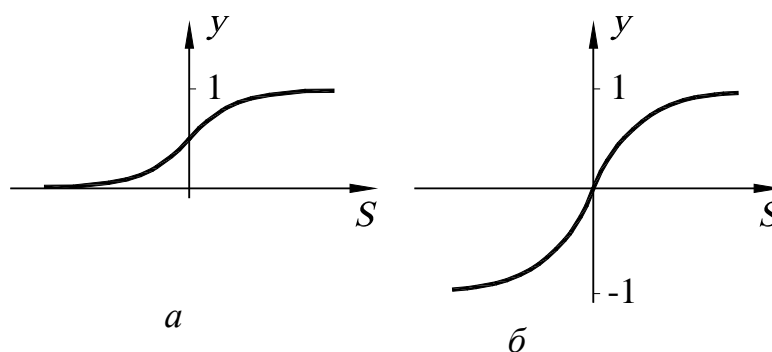


Рис. 8. Сигмоидные активационные функции с несимметричной (а)

и симметричной (δ) областями изменения

Персептроны с сигмоидными активационными функциями хорошо обучаются с помощью алгоритма обратного распространения ошибки, а также с помощью дельта правила и обобщенного дельта-правила (если персептрон однослойный).

Логарифмические активационные функции. На рис. 9 представлен график активационной функции, заданной уравнением

$$y = \ln(S + \sqrt{S^2 + 1}) . \quad (18)$$

В отличие от сигмоидной эта функция имеет неограниченную область изменения. Иногда это удобно, т.к. не требуется масштабирования выходных сигналов персептрона (подробнее об этом см. теоретический материал к лабораторной работе №8). Кроме того, логарифмические активационные функции позволяют избегать нежелательного эффекта, называемого *параличем сети* – потерей чувствительности сети к вариациям весовых коэффициентов и, как следствие, замирание процесса обучения при попадании взвешенных сумм входных сигналов нейрона в область насыщения сигмоиды (см. также лабораторную работу №8).

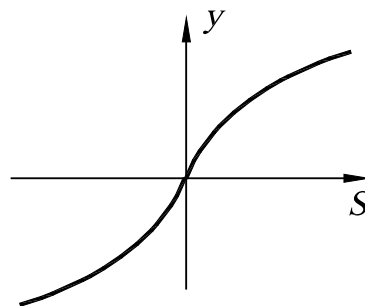


Рис. 9. Логарифмическая активационная функция

Радиально-базисные активационные функции. В последнее время получают распространение нейросети, нейроны которых имеют активационные функции в форме функции Гаусса (см. рис. 10):

$$y = e^{-\frac{S^2}{2\sigma^2}},$$

где $S = \|X - C\|$ – евклидово расстояние между входным вектором X и центром активационной функции C ; σ – параметр гауссовой кривой, называемый шириной окна. Такие активационные функции называют *радиально-базисными (RBF)*, а соответствующие нейронные сети – *RBF-сетями*. Методы обучения RBF-сетей, их преимущества и недостатки в нашем учебном курсе не рассматриваются.

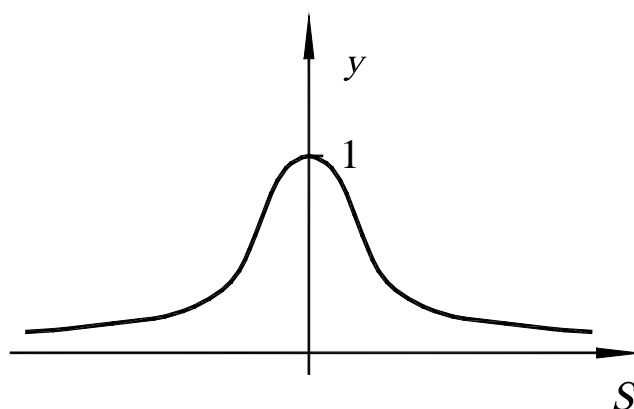


Рис. 10. Радиально-базисная активационная функция

Отметим, что все приведенные выше активационные функции, за исключением пороговой и кусочно-линейной, являются непрерывно дифференцируемыми. Линейная функция и логарифмическая функция выполняют преобразование бесконечного входного множества значений переменных S в бесконечное множество переменных y . Пороговые активационные функции преобразуют множество значений S в бинарные множества $y = 0$ и $y = 1$ или $y = -1$ и $y = 1$. Остальные активационные функции преобразуют бесконечное входное множество значений S в ограниченные выходные множества: $y \in (0,1)$, $y \in (-1,1)$ и $y \in (0,1]$. От вида используемых активационных функций зависят функциональные возможности нейронных сетей, а также выбор способов их обучения.

----- Коротко о главном -----

Активационные функции осуществляют преобразование взвешенной суммы входных сигналов нейрона в его выходной сигнал. От вида активационных функций зависят функциональные возможности нейронных сетей, а также выбор способов их обучения.