Теоретический материал к лабораторной работе №4

РАСПОЗНАВАНИЕ ПЕЧАТНЫХ И РУКОПИСНЫХ БУКВ

На рис. 1 представлена схема персептрона, предназначенного для распознавания букв русского алфавита. Персептрон имеет 33 выходных нейрона: каждой букве алфавита соответствует свой выходной нейрон. Полагается, что сигнал первого выходного нейрона y_1 должен быть равен единице, если персептрону предъявлена буква «А», и нулю для всех остальных букв. Выход второго нейрона y_2 должен быть равен единице, если персептрону предъявлена буква «Б», и нулю во всех остальных случаях. И так далее до буквы «Я».

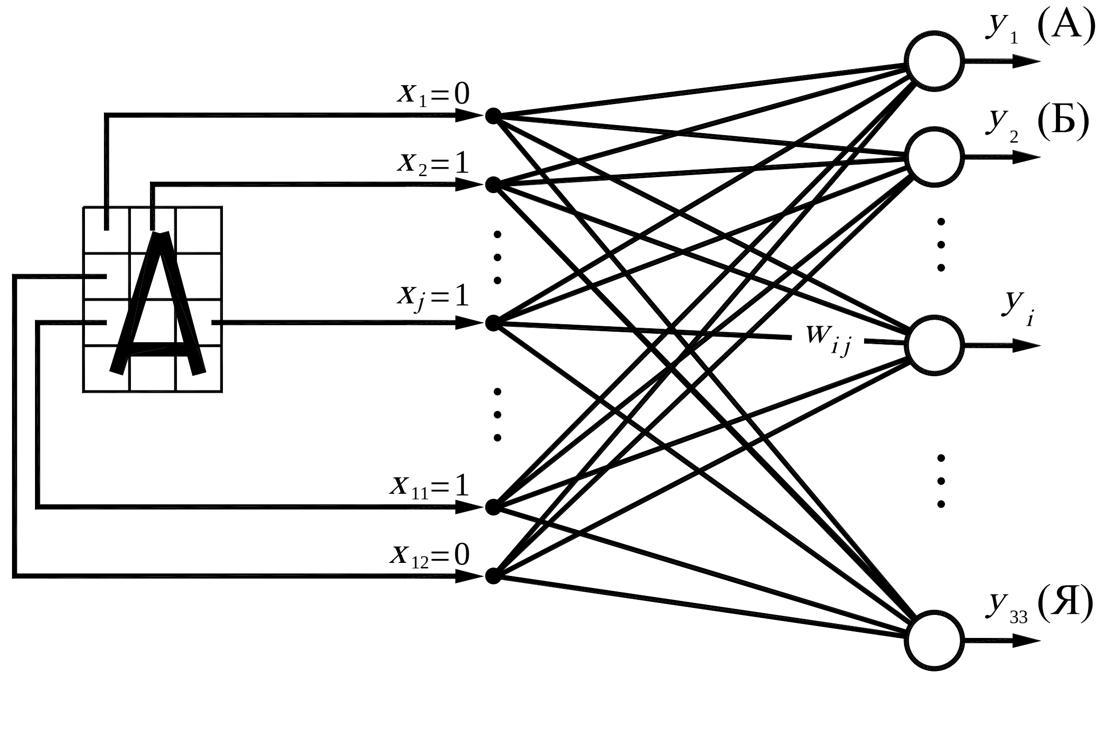

Рис. 1. Персептрон, предназначенный для распознавания букв русского алфавита

При выполнении предыдущей лабораторной работы мы убедились, что персептрон научился распознавать не только буквы, на которых его обучали, но и буквы, которых в обучающем множестве не было, если они не слишком отличались от букв обучающего множества. Свойство распознавать новые образы, которые персептрон никогда «не видел», мы назвали свойством *обобщения*.

Дальнейшее развитие идеи персептрона было связано с попытками расширить круг его применения и усовершенствовать алгоритм обучения. Существенное развитие персептрона было сделано американскими учеными

Б.Уидроу и *М.Е.Хоффом*, которые вместо изученной на первой лабораторной работе ступенчатой активационной функции ввели непрерывную нелинейную функцию активации

$$y = \frac{1}{1+e^{-S}}, \qquad (1)$$

график которой изображен на рис. 2.

Эту функцию назвали *сигмоидой*, из-за того, что ее графическое изображение напоминает латинскую букву «S». Другое название сигмоиды – *логист ическая функция.*

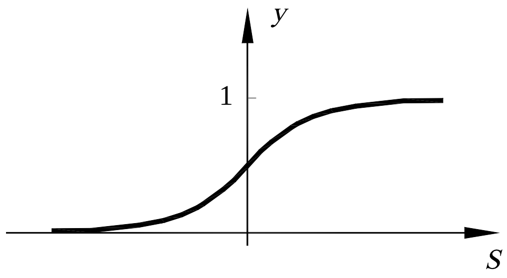

Рис. 2. Сигмоидная активационная функция $y = f_\sigma(S)$

Подобно обычной пороговой функции активации, сигмоида отображает точки области определения $(-\infty, +\infty)$ в значения из интервала $(0, +1)$. Практически сигмоида обеспечивает непрерывную аппроксимацию классической пороговой функции.

Для сигмоиды приняли обозначение $y = f_\sigma(S)$. Персептроны с сигмоидными активационными функциями с одним выходом назвали *адалайн*, с несколькими выходами – *мадалайн* (от английских слов *ADAptive LInear NEuron* и *Many ADALINE*).

Появление персептронов с непрерывными активационными функциями обусловило появление новых подходов к их обучению. Б.Уидроу и М.Е.Хофф предложили минимизировать квадратичную ошибку, определяемую формулой:

$$\varepsilon = \frac{1}{2}\sum_{i=1}^{I}(d_i - y_i)^2, \qquad (2)$$

в которой d_i – требуемый (желаемый) выход i-го нейрона, а y_i – тот, который получился в результате вычислений персептрона.

Рассмотрим алгоритм коррекции весовых коэффициентов персептрона, имеющего J входов и I выходов (рис. 3).

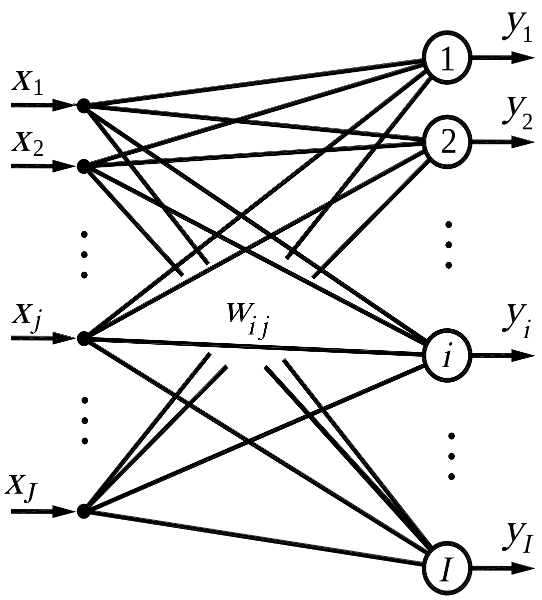

Рис. 3. Персептрон с J входами и I выходами

Квадратичная ошибка обучения персептрона ε зависит от того, какими являются весовые коэффициенты w_{ij}. Другими словами ε является функцией от весовых коэффициентов w_{ij}: $\varepsilon = \varepsilon(w_{ij})$. В школьных курсах обычно изучаются функции только от одного аргумента: $y = y(x)$, которые на координатной плоскости x, y изображаются, как известно, в виде кривых линий. Если функция z зависит от двух аргументов: $z = z(x, y)$, то она изображается в трехмерной системе координат x, y, z в виде поверхности. Функция-ошибка персептрона $\varepsilon = \varepsilon(w_{ij})$ зависит от большого количества аргументов w_{ij}, поэтому для ее графического представления требуется многомерная система координат, которую мы в нашем трехмерном мире представить себе не можем. В этой многомерной системе координат функция $\varepsilon = \varepsilon(w_{ij})$ изображается в виде многомерной поверхности, называемой *гиперповерхностью*.

Чтобы хоть как-то представить себе гиперповерхность, предположим, что все аргументы w_{ij} имеют постоянные значения за исключением двух, например w_{11} и w_{12}, которые являются переменными. Тогда в трехмерной системе координат w_{11}, w_{12}, ε гиперповерхность будет иметь вид фигуры, напоминающей параболоид, которую назовем *псевдопарабалоидом* (рис. 4). Процесс обучения персептрона теперь можно представить как отыскание такого сочетания весовых коэффициентов w_{ij}, которому соответствует самая нижняя точка *гиперпсевдопараболоида*. Задачи подобного рода называются *оп-*

тимизационными. Говорят, что оптимизационная задача состоит в *минимизации функции* $\varepsilon = \varepsilon(w_{ij})$ в многомерном пространстве параметров w_{ij}.

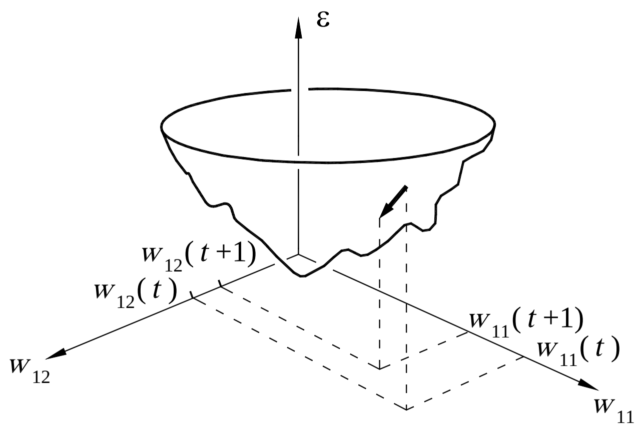

Рис. 4. Графическое изображение функции-ошибки персептрона $\varepsilon = \varepsilon(w_{ij})$ в трехмерной системе координат w_{11}, w_{12}, ε

Таким образом, если раньше говорили, что персептрон обучают методом «поощрения – наказания», то теперь стали говорить, что задача обучения персептрона – это задача оптимизации (минимизации) персептронной ошибки (погрешности).

Существует множество методов решения оптимизационных задач. Наиболее простым методом является перебор весовых коэффициентов w_{ij} с последующими вычислениями и сравнениями между собой соответствующих этим коэффициентам значений функции ε. Более эффективен метод *градиент ного спуска*, согласно которому изменение (коррекция) каждого весового коэффициента Δw_{ij} производится в сторону, противоположную *градиенту* функции ε. Градиент функции является очень важным математическим понятием, которое обычно проходят на первых курсах вузов. Здесь мы не будем на нем останавливаться, а только укажем, что градиент функции $\varepsilon = \varepsilon(w_{ij})$ представляет собой вектор, проекциями которого на оси координат являются производные от функции ε по этим координатам (их обозначают $\partial\varepsilon/\partial w_{ij}$), и что градиент функции всегда направлен в сторону ее наибольшего возрастания. Поскольку наша задача состоит в отыскании минимума функции $\varepsilon = \varepsilon(w_{ij})$, то нам надо опускаться по поверхности ошибок, что обеспечивается движением в сторону, противоположную градиенту этой функции. Отсюда и упомянутое выше название – *метод градиентного спуска*.

Движение в сторону, противоположную градиенту (т.е. противоположную направлению возрастания функции), будет осуществляться, если на каждой итерации к координатам текущей точки w_{ij} мы будем добавлять величину, прямо пропорциональную производной по координате w_{ij}, взятую с противоположным знаком:

$$\Delta w_{ij} = -\eta \frac{\partial \varepsilon}{\partial w_{ij}}, \tag{3}$$

где η – некоторый коэффициент, обычно задаваемый в пределах от 0,05 до 1, и называемый, как и раньше, *коэффициент ом скорост и обучения.*

Обратите внимание, что согласно формуле (3) мы движемся не только в сторону убывания функции, но и со скоростью, прямо пропорциональной скорости убывания (крутизне) функции, т.к. делаем шаг Δw_{ij}, пропорциональный производной, взятой со знаком минус.

Квадратичная ошибка ε является сложной функцией, зависящей от выходных сигналов персептрона y_i, которые, в свою очередь, зависят от w_{ij}, т.е. $\varepsilon = \varepsilon(y_i(w_{ij}))$. По правилу дифференцирования сложной функции

$$\frac{\partial \varepsilon}{\partial w_{ij}} = \frac{\partial \varepsilon}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}}. \tag{4}$$

Выходные сигналы нейронов y_i вычисляются с помощью сигмоидных активационных функций $y_i = f_\sigma(S_i)$, аргументом которых являются суммы $S_i = \sum_{j=1}^{J} w_{ij} x_j$. Следовательно,

$$\frac{\partial y_i}{\partial w_{ij}} = \frac{\partial f_\sigma(S_i)}{\partial S_i} \frac{\partial S_i}{\partial w_{ij}} = f'_\sigma(S_i) x_j. \tag{5}$$

Кроме того, если продифференцировать (2) по y_n, где $n \in [1, I]$, то получится $\frac{\partial \varepsilon}{\partial y_n} = -(d_n - y_n)$, значит

$$\frac{\partial \varepsilon}{\partial y_i} = -(d_i - y_i) . \tag{6}$$

Подставив (5) и (6) в (4) и затем полученное выражение в (3), окончательно будем иметь

$$\Delta w_{ij} = -\eta\left(-(d_i - y_i)\, f'_\sigma(S_i) x_j\right) = \eta(d_i - y_i)\, f'_\sigma(S_i) x_j. \tag{7}$$

Это выражение получено для нейронов с активационными функциями любого вида. Если $f_\sigma(S_i)$ – сигмоида, заданная формулой (1), то

$$f'_\sigma(S_i) = \left((1 + e^{-S_i})^{-1}\right)' = f_\sigma(S_i)\left(1 - f(S_i)\right). \tag{8}$$

Подставив это выражение в (7), получим:

$$\Delta w_{ij} = \eta(d_i - y_i)\, f_\sigma(S_i)\left(1 - f(S_i)\right) x_j = \eta(d_i - y_i) y_i (1 - y_i) x_j. \tag{9}$$

Итак, мы получили *итерационную формулу* для обучения персептрона

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}, \tag{10}$$

где

$$\Delta w_{ij} = \eta \delta_i x_j, \tag{11}$$

$$\delta_i = y_i (1 - y_i)(d_i - y_i). \tag{12}$$

Введенную здесь с помощью формулы (12) величину δ_i в дальнейшем будем называть нейронной ошибкой. Алгоритм (10)-(12) называют *обобщенным дельта-правилом*. Его преимущество по сравнению с обычным дельта-правилом состоит в более быстрой сходимости и в возможности более точной обработки входных и выходных непрерывных сигналов, т.е. в расширении круга решаемых персептронами задач.

Итак, ведение сигмоидной функции активации вместо функции-ступеньки и появление нового алгоритма обучения – обобщенного дельта-правила, расширило область применения персептрона. Теперь он может оперировать не только с бинарными (типа «ноль» и «единица»), но и с непрерывными (аналоговыми) выходными сигналами.