УДК 004.8

Асп. ***Кудухов А. Н.***
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, РСО-Алания, Россия

АНАЛИЗ СТРУКТУРЫ ОРГАНИЗАЦИИ НЕЙРОННЫХ СЕТЕЙ

В статье рассмотрен алгоритм обратного распространения ошибки для обучения нейронных сетей прямого распространения.

Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около 10^{11} нейронов участвуют в примерно 10^{15} передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими качествами, общими с другими элементами тела, но его уникальной способностью является прием, обработка и передача электрохимических сигналов по нервным путям, которые образуют коммуникационную систему мозга [1].

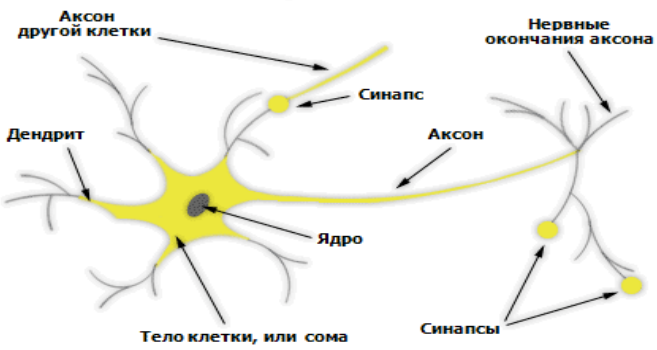

Рис. 1. Биологический нейрон.

Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы подводятся к

телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам.

Нейрон – это составная часть нейронной сети. На рис. 2 показана его структура.

В состав нейрона входят умножители, сумматоры и нелинейный преобразователь. Синапсы осуществляют связь между нейронами и умножают входной сигнал на число, характеризующее силу связи – веса синапсов.

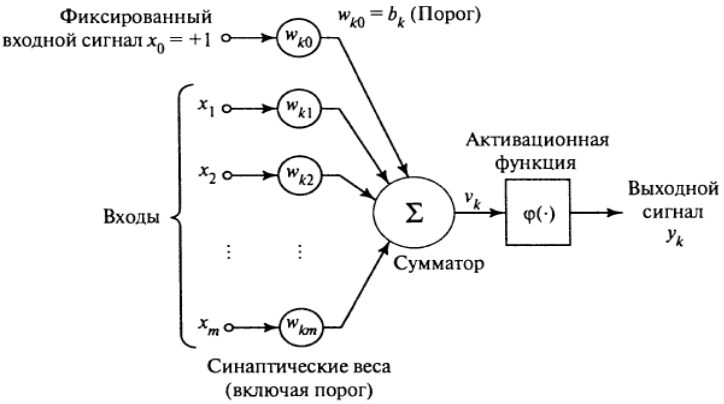

Рис. 2. Структура искусственного нейрона.

Сумматор выполняет сложение сигналов, поступающих по синоптическим связям от других нейронов, и внешних входных сигналов. Нелинейный преобразователь реализует нелинейную функцию одного аргумента – выхода сумматора. Это функция называется "*функцией активации*" или "*передаточной функцией*" нейрона. Нейрон в целом реализует скалярную функцию векторного аргумента. Математическая модель нейрона описывается соотношениями

$$S = \sum_{i=1}^{N} W_i \cdot x_i + b\,,$$

где w_i – вес синапса $(i = 1,\ldots.n)$; b – значение смешения; s – результат суммирования; x_i – компонент входного вектора $(i = 1,\ldots.n)$; y – выходной сигнал нейрона; n – число входов нейрона; f – функция активации.

Алгоритм обратного распространения ошибки

Определение сети

Стандартная нейронная сеть с прямой связью, показанная на рис. 2, также известна как многослойный персептрон (MLP). Обратите внимание, что узлы входного уровня показаны как простые круги. Это должно означать, что никакая обработка не происходит в этих узлах, они служат только в качестве входов сети [5].

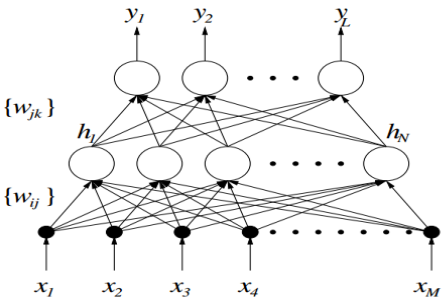

Рис. 2. Многослойный персептрон:
стандартная сеть с прямым распространением.

Каждый узел вычисляет взвешенную сумму его входов, и использует его как входные данные функции преобразования $f(x)$. В классическом MLP функция преобразования – сигмод.

Сигмоида – это гладкая монотонная нелинейная S-образная функция, которая часто применяется для "*сглаживания*" значений некоторой величины:

$$F(x) = \frac{1}{1 + e^{-x}}. \qquad (1)$$

Рассмотрим узел k в скрытом слое. Его выход y_k представляет $y_k = f(net_k)$, где net_k – функция активации (сигмоида), является взвешенной суммой выходов узлов входного слоя A:

$$net_k = \sum_{i=1}^{N} w_{jk} \cdot h_j , \qquad (2)$$

Аналогично – выход каждого узла в каждом слое.

Суммарная квадратичная ошибка и градиентный спуск

Для удобства мы можем рассмотреть входы сети как входной вектор X, где $X = [x_1 \ldots x_n]^T$. Аналогично, выход для сети можно рассматривать как вектор выхода Y, где $Y = [y_1 \ldots y_n]^T$. Учебный набор для сети можно представить рядом пар K входных векторов x_l и желаемых векторов выхода d_l : K={(x_1, d_1), (x_2, y_2) … (x_n, y_n)}.

Каждый раз, когда входной вектор от учебного набора x_l применен к сети, сеть производит фактический выход y_l. Мы можем таким образом определить квадратичную ошибку для этого входного вектора, суммируя квадратичные ошибки в каждом узле выхода:

$$E = \frac{1}{2}\sum_{i=1}^{l}(y_k - d_k)^2. \qquad (3)$$

.

Главная задача в обучении нейронной сети, это минимизировать квадратичную ошибку. Мы можем также определить полную квадратичную ошибку E, суммируя все пары входа – выхода в учебном наборе:

$$E = \frac{1}{2}\sum_{l=1}^{K}\sum_{k=1}^{L}(y_{kl} - d_{kl})^2. \qquad (4)$$

Для минимизации квадратичной ошибки будем использовать алгоритм градиентного спуска. Определим какое направление является "скоростным спуском" на поверхности ошибок и изме-

ним каждый вес w так, чтобы мы двигались в этом направлении. Математически это означает, что каждый вес w будет изменен на небольшое значение dw в направлении уменьшения E:

$$w(t+1) = w(t) + \Delta w(t), \qquad \text{где } \Delta w(t) = -\theta \frac{dE}{dw}|t. \qquad (5)$$

Здесь $w(t)$ – вес во время t и $w(t+1)$ – обновленный вес. Уравнение 5 называется обобщенным дельта-правилом. Чтобы выполнить градиентный спуск, нужно найти частную производную $\frac{dE}{dw}$ каждого веса.

Для корректировки весов между скрытым и выходным слоем нужно найти частную производную для каждого узла выходного слоя:

$$\frac{dE}{dw_{jk}} = (y_k - d_k) y_k (1 - y_k) h_j \,. \qquad (6)$$

Таким образом, мы нашли частную производную ошибки E по весам w_{jk} и можем использовать этот результат в уравнении 5, чтобы выполнить градиентный спуск для всех весов между скрытым и выходным слоями.

Теперь рассмотрим веса, между входным слоем и скрытым слоем:

$$\frac{dE}{dw_{ij}} = \sum_{k=1}^{L} \frac{dE}{dy_k} \frac{dy_k}{dnet_k} \frac{dnet_k}{dh_j} \frac{dh_j}{dnet_j} \frac{dnet_j}{dw_{ij}}, \qquad (7)$$

$$\frac{dE}{dw_{ij}} = (y_k - d_k) y_k (1 - y_k) w_{ij} \; h_j \big(1 - h_j\big) x_i. \qquad (8)$$

Таким образом, мы нашли частную производную ошибки E по весу на основе известных величин (многие из которых мы уже вычислили при получении $\frac{dE}{dw_{jk}}$). Вместе с уравнением (6) это

дает нам весь $\frac{dE}{dw}$ необходимый для того, чтобы уравнение (5) могло использоваться для выполнения градиентного спуска для всех весов в сети.

Появление алгоритма обратного распространения ошибки стало знаковым событием в области нейронных сетей, так как он реализует вычислительно эффективный метод обучения многослойного персептрона [4].

ЛИТЕРАТУРА

1. *Хайкин С.* Нейронные сети. Полный курс. М.: Изд-во «Вильямс», 2006. 1104 с.

2. *Рассел С., Норвиг П.* Искусственный интеллект. Современный подход. М.: Изд-во «Вильямс», 2006. 1408 с.

3. *Уосермен Ф.* Нейрокомпьютерная техника. М.: Мир. 1992. 240 с.

4. Информационный ресурс: http://www.aiportal.ru/articles/neural-networks/back-propagation. – Алгоритм обратного распространения ошибки.

5. Информационный ресурс: http://www.csse.monash.edu.au – Алгоритм обратного распространения ошибки.

———— ✧ ————

УДК 519.687.1

Асп. ***Чипиров З. А.***
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, РСО-Алания, Россия

АНАЛИЗ ЗАВИСИМОСТИ ЭНЕРГОПОТРЕБЛЕНИЯ СОВРЕМЕННЫХ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ В РАЗЛИЧНЫХ РЕЖИМАХ РАБОТЫ

Обзорная статья, анализирующая энергопотребление современных графических процессоров с целью создания модели, позволяющей эффективно использовать ресурсы, снижая уровень энергопотребления.

Современные видеокарты по всем параметрам намного превосходят платы, вышедшие еще несколько лет назад – по скорости, по функциональности, по сложности и, увы, по энергопотреблению и тепловыделению. На сегодняшний день дорогая и