

УДК 004.032.26

**СКОРОСТНОЙ МЕТОД ОБУЧЕНИЯ
МНОГОСЛОЙНОГО ПЕРСЕПТРОНА**Шумков Евгений Александрович
к.т.н.Карлов Дмитрий Николаевич
*Кубанский Государственный Технологический
Университет, Краснодар, Россия*

В статье предложен скоростной метод обучения многослойного перцептрона, который отличается высокой скоростью обучения. Приведены результаты сравнения скоростного метода обучения со стандартными методами

Ключевые слова: МНОГОСЛОЙНЫЙ ПЕРСЕПТРОН, СКОРОСТНОЙ АЛГОРИТМ ОБУЧЕНИЯ, МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

UDC 004.032.26

**HIGH-SPEED METHOD OF MULTILAYERED
PERCEPTRON TRAINING**Shumkov Eugeniy Alexandrovich
Cand.Tech.Sci.Karlov Dmitry Nikolaevich
*Kuban State Technological University, Krasnodar,
Russia*

In this article the high-speed method of multilayered perceptron learning which differs in the high speed of training is offered. Results of comparison of a high-speed method of learning with standard methods are resulted

Keywords: MULTILAYERED PERCEPTRON, HIGH – SPEED METHOD OF LEARNING, BACK PROPAGATION LEARNING ALGORITHM

Основным способом обучения многослойного перцептрона является метод обратного распространения ошибки (далее BackProp). Данный метод обучения хорошо себя зарекомендовал при решении многих задач [1,2,4,9]. Но у данного метода есть недостатки, к которым можно отнести: длительное время обучения; частое нахождение локальных, а не глобальных минимумов; проблема выбора адекватной¹ архитектуры сети и другие. Часть данных проблем можно обойти, используя различные способы, например расписание обучения, динамическое добавление нейронов в скрытые слои при старте с заведомо недостаточным количеством нейронов, добавление «шума» во входной сигнал и т.д. [5]. Но основной проблемой обучения нейронной сети является длительное время обучения, то есть через сколько итераций ошибка сети будет меньше заданной. Существуют методы на базе BackProp, которые ускоряют обучение, например метод RProp [7], но выигрывая в скорости, он проигрывает в точности, по сути, в адекватности решения поставленной задачи.

¹ Данную проблему еще можно обозначить, как выбор минимально достаточной конфигурации сети для решаемой задачи.

В данной работе предлагается метод обучения нейронной сети, который также как и алгоритм обратного распространения ошибки использует ошибку на выходах сети и последовательно пропускает ее через слои нейронной сети от выходного к входному. При этом не используются градиентные методы обучения. В основе предлагаемого метода лежит вычисление обратной функции от гиперболического тангенса, подсчет разницы между желательным и реальным значением на входе в нейрон и корректировка синаптических связей на необходимое значение. Предлагаемый метод относится к классу обучения с учителем. Укажем методы, которые отчасти сходны с предлагаемым – это обучение нейронной сети с помощью генетических алгоритмов, метод отжига, метод случайного обучения [3,7].

Для общности введем следующие обозначения, как показано на Рисунке 1, при этом нейрон функционирует стандартным способом:

$$z_j^{(k)} = \sum_{i=0}^n w_{ij}^{(k-1)} \cdot x_i^{(k-1)}, \quad (1)$$

$$y_j^{(k)} = f(z_j^{(k)}) \quad (2),$$

где $x_i^{(k)}$ - вектор, входящий в сумматор i -го нейрона k -го слоя; $f(z)$ - функция активации нейрона (в данной работе рассматривается только гиперболический тангенс); $y_j^{(k)}$ - сигнал на выходе j -го нейрона k -го слоя, w_{ij} - веса синаптических связей между слоями.

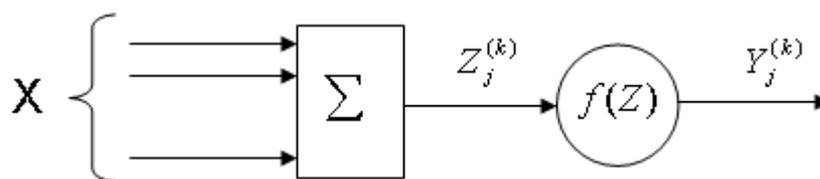


Рисунок 1. Формальный нейрон

В скоростном методе обучения ошибку можно считать:

- по валидационной выборке после каждого поданного во время обучения примера;
- суммарно по всем обучающим примерам.

Алгоритм скоростного обучения многослойного персептрона состоит из следующих шагов:

1. Подача на вход выбранного случайным образом примера и расчет значений y_i на выходе сети. Или, в зависимости от выбранного способа пакетная подача примеров с суммированием ошибки на выходе.
2. Расчет обратной функции от гиперболического тангенса, то есть расчет того, что должно было входить в нейроны выходного слоя для правильного ответа сети. Обратная функция от гиперболического тангенса:

$$\hat{z}_j^{(k)} = \frac{1}{2 \cdot \alpha} \cdot \ln\left(\frac{1 + y_j^{(k)}}{1 - y_j^{(k)}}\right), \quad (3)$$

где α - параметр гиперболического тангенса.

3. Расчет разницы между тем, что должно было входить в нейрон и тем, что реально было на него подано:

$$\delta_j = z_j^{(k)} - \hat{z}_j^{(k)}, \quad (4)$$

где z - число которое должно было быть подано на вход нейрона выходного слоя для правильного ответа, \hat{z} - то что реально вошло в нейрон. Таким образом, мы вычисляем разницу между желаемым и действительным входом в нейрон.

4. Генератором случайных чисел разбиваем δ по всем синаптическим связям, входящим в данный нейрон в процентном соотношении².
При этом:

² Здесь крайне важно использовать генератор действительно случайных чисел, а не псевдослучайных.

$$\sum_{i=0}^n p_i = 1, \quad (5)$$

5. Вычисляем - на сколько нужно изменить сигнал, исходящий из нейрона предыдущего слоя, при этом учитываем, что данный сигнал идет на все нейроны последующего слоя умноженный на соответствующие веса связей. При этом мы хотим распределить ошибку и на веса связей w и на выход предыдущего слоя $z^{(k-1)}$, чтобы изменения затронули и предыдущие слои нейронной сети. Исходная система уравнений для первого нейрона (k-1) слоя:

$$\begin{aligned} \hat{w}_{i1}^{(k)} \cdot \hat{z}_i^{(k-1)} &= w_{i1}^{(k)} \cdot z_i^{(k-1)} - \delta_1 \cdot p_i^1 \\ &\dots\dots\dots \\ \hat{w}_{in}^{(k)} \cdot \hat{z}_i^{(k-1)} &= w_{in}^{(k)} \cdot z_i^{(k-1)} - \delta_1 \cdot p_n^1 \end{aligned} \quad (6)$$

В данном случае получается (n+1) неизвестное – это $\hat{w}_{ij}^{(k)}$ и $\hat{z}_{in}^{(k-1)}$.

Наиболее простым решением является следующее – пусть:

$$\hat{w}_{i1}^{(k)} = w_{i1}^{(k)} - \delta_1 \cdot p_i (1 - \mu), \quad (7)$$

где μ - случайное число ($\mu \in [0;1]$). Далее подставляя установленное значение $\hat{w}_{i1}^{(k)}$ в первое уравнение системы (6) находим $\hat{z}_i^{(k-1)}$, после подставляем найденное значение выхода первого нейрона в остальные уравнения системы (6) находим новые значения весов связей. Система (6) составляется для каждого нейрона слоя (k-1).

6. Далее, аналогично вычисляются изменения для предыдущих слоев, в частности следующий шаг - вычислить через обратную функцию гиперболического тангенса, сколько должно получиться на входах нейронов (k - 1) слоя.

Экспериментальная часть проведена для решения задачи прогноза курса котировок на РТС, в частности использовались временные ряды для различных таймфреймов компании «Роснефть». Разработанный метод

скоростного обучения будем сравнивать со стандартным методом обратного распространения ошибки. Программное обеспечение обоих методов для адекватности оценки написано на одном языке – VBA, при этом практически все функции и процедуры используются и для одного метода и для другого способа обучения, различие лишь в коде функции выполняющей непосредственно обучение³.

При решении задач с помощью нейронных сетей необходимо вначале определиться с входами, выходами, количеством обучающих примеров, размером валидационной выборки и первоначальной структурой сети. Рассмотрим перечисленные этапы по порядку.

Определение входов и выходов. В задачах прогнозирования общепринято использование в качестве входного вектора так называемого «скользящего окна» [8].

Таблица №1

Структура «скользящего окна»

Вход №1	Вход №2	Вход №3	Вход №4
$P(t) - P(t - 1)$	$P(t - 1) - P(t - 2)$	$P(t - 2) - P(t - 3)$	$P(t - 3) - P(t - 4)$
$P(t - 1) - P(t - 2)$	$P(t - 2) - P(t - 3)$	$P(t - 3) - P(t - 4)$	$P(t - 4) - P(t - 5)$
$P(t - 2) - P(t - 3)$	$P(t - 3) - P(t - 4)$	$P(t - 4) - P(t - 5)$	$P(t - 5) - P(t - 6)$
...

Выход в задачах прогнозирования нейронными сетями обычно используется один – прогнозное значение на следующей итерации. В рассматриваемой задаче на выходе должен формироваться сигнал значения изменения котировки на следующий момент времени $P(t + 1) - P(t)$. В качестве основного, прогнозируемого параметра целесообразно выбрать цену закрытия бара *Close*, либо подавать высоту

³ Исходный код нейронной сети можно найти в сети Интернет по адресу: <http://apsheronk.bozo.ru/Forex/forex.html>

японских свечей $Close(t) - Open(t)$ (или $High(t) - Low(t)$)⁴. В качестве дополнительных входов в экспериментах использовались - относительное изменение цены на нефть марки Brent, так как «Роснефть» нефтяная компания, а также индекс РТС. Напомним также о необходимости качественного масштабирования входных данных [8].

На Рисунке 2 и 3 показаны типичные графики ошибок при обучении разными методами. Отметим, что одна итерация в скоростном методе по времени примерно равна 1.1 итерации метода BackProp.

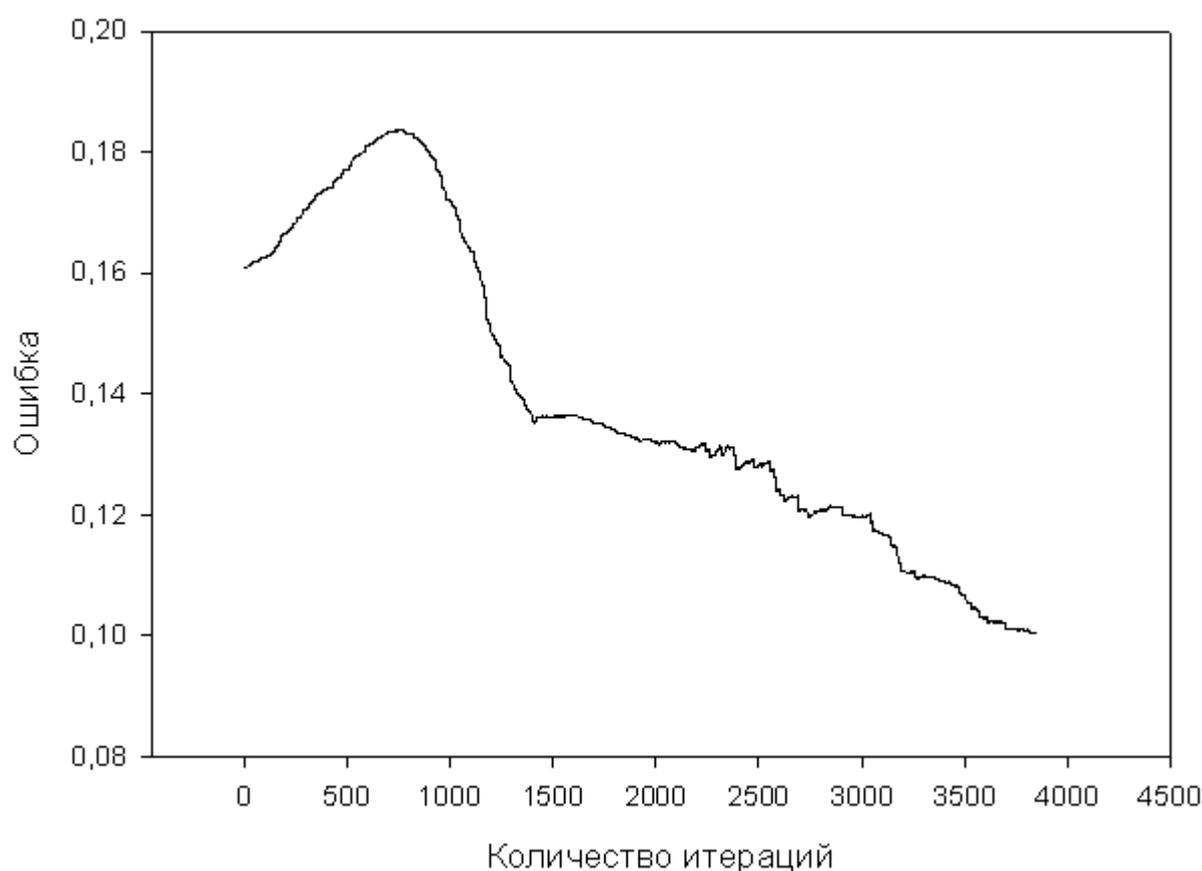


Рисунок 2. График ошибки при методе обучения BackProp.

⁴ Open – цена открытия бара, Low – низшая цена бара, High – наивысшая цена, Close – цена закрытия бара.

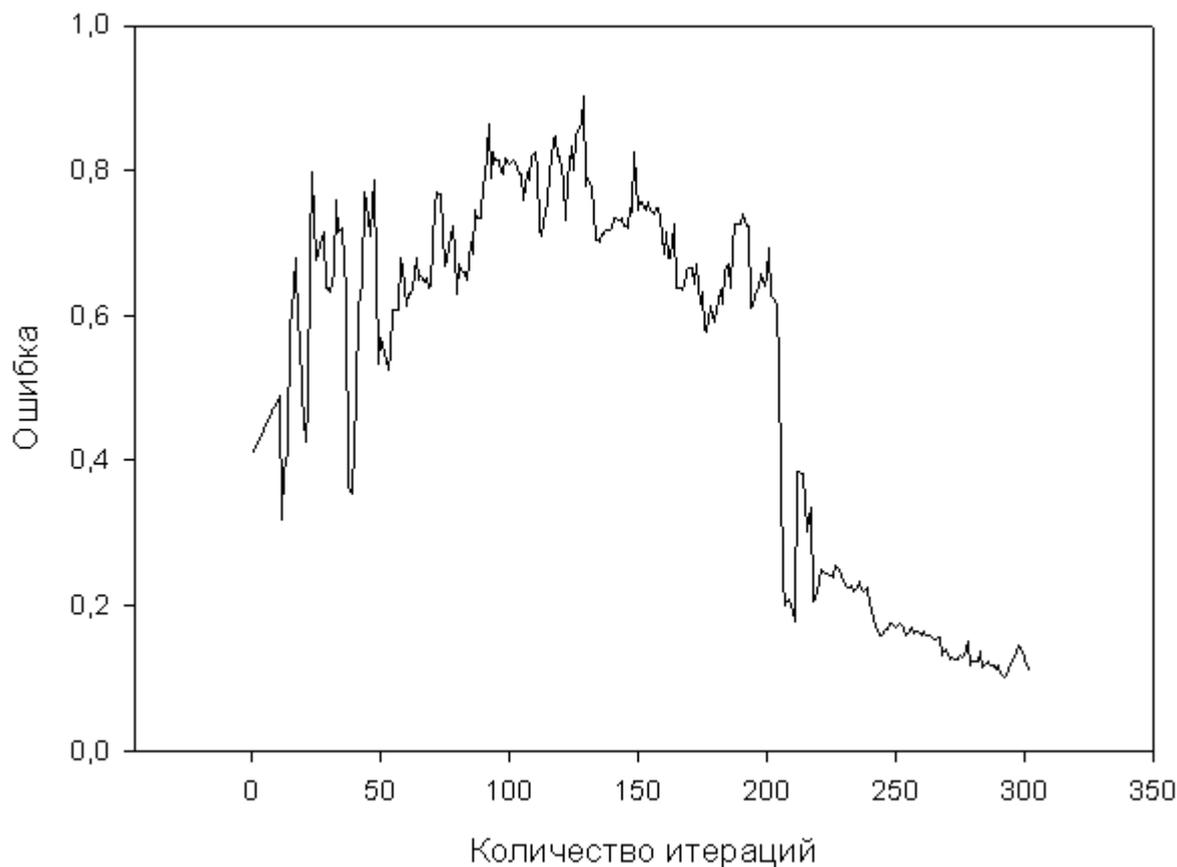


Рисунок 3. График ошибки при скоростном методе обучения

На Рисунке 3 видно высокое перерегулирование в процессе обучения по предлагаемому скоростному методу. В Таблице 2 представлены данные по времени обучения и точности воспроизводимости данных для скоростного и BackProp методов. Во временных показателях не учитывалось время предобработки данных, так как оно одинаково для обеих моделей.

Таблица 2 - Результаты экспериментов

Кол-во обучающих примеров	Ср. кол-во нейронов ⁵		Ср. кол-во итераций обучения		Точность при обучении, %		Точность на новых примерах, %	
	B ⁶	F	B	F	B	F	B	F
50	15	14	2980	347	91	90	81	79
100	26	23	3403	571	87	84	77	71
200	34	36	3720	899	85	80	72	68
300	41	39	4481	1002	81	76	69	62

⁵ Количество нейронов в скрытом слое. Во всех экспериментах использовался один скрытый слой.

⁶ Обозначения в колонках B – BackProp, F – скоростной метод обучения.

Таким образом, из экспериментов видно, что скоростной алгоритм обратного распространения ошибки существенно превосходит по времени обучения стандартный алгоритм обратного распространения ошибки при той же точности на валидационной выборке. В то же время на тех примерах, которые нейронная сеть не видела ни при обучении, ни при тестировании, стандартный алгоритм по точности превосходит скоростной. Авторы объясняют это тем, что стандартный алгоритм обратного распространения ошибки медленно и тщательно перерабатывает все примеры из обучающей выборки, а скоростной алгоритм действует по принципу «ближайшего шага к цели». Скоростной алгоритм обучения многослойного персептрона рекомендуется использовать, как оценочный, перед использованием стандартного алгоритма обратного распространения или его модификаций. То есть – если оценочный метод достигает за заданное количество итераций определенной ошибки, то весьма вероятно, что и стандартный алгоритм может обучиться на заданном множестве примеров. Также целесообразно использовать скоростной алгоритм обучения для задач в которых нет высоких требований по точности, но есть жесткие ограничения по времени обучения, например для задач решаемых в режиме реального времени.

Литература:

1. Widrow B., Lehr M.A. 30 years of adaptive neural networks: perceptron, madaline and backpropagation // Proceeding of the IEEE. – 1990. – Vol. 78. - №9. – P. 1415 – 1442.
2. Вороновский Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Х.: ОСНОВА, 1997. 112 с.
3. Галушкин. А.И. Нейрокомпьютерные системы. - М.: Издательское предприятие журнала "Радиотехника", 2000.- 205с.
4. Ежов А., Чечеткин В.. Нейронные сети в медицине. «Открытые системы.» -1997.- №4. - С. 34-37.
5. Заенцев И. В. Нейронные сети: основные модели. Учебное пособие . Воронеж: ВГУ. 1998.- 76с.
6. Пупков К.А., Егупов Н.Д. Методы классической и современной теории автоматического управления. Т.5. – М.: Издательство МГТУ им. Н.Э.Баумана, 2004. – 784 с.

7. Рутковская Д., Пилиньский М., Рутковский Л. «Нейронные сети, генетические алгоритмы и нечеткие системы». Пер. с польск., И.Д. Рудинского. М.: Горячая линия – Телеком, 2006. 452 с.
8. Шумков Е.А. Система поддержки принятия решений предприятия на основе нейросетевых технологий: Дисс.. канд. техн. наук. Краснодар, КубГТУ, 2004. - 158с.
9. Эрлих А. Технический анализ товарных и фондовых рынков. - М.: Юнити, 1996. - 215 с.