

**АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ СНИЖЕНИЯ
ВЫБОРОЧНОЙ КОНСТАНТЫ ЛИПШИЦА СЕТИ
КАК ИНСТРУМЕНТОВ ПОВЫШЕНИЯ СКОРОСТИ ОБУЧЕНИЯ
НЕЙРОСЕТЕВЫХ АЛГОРИТМОВ****Т. В. Азарнова, И. Н. Терновых***Воронежский государственный университет***Поступила в редакцию 28.03.2011 г.**

Аннотация. В статье рассматривается проблема обучения нейронных сетей при больших объемах обучающих выборок и проводится экспериментальный анализ по оценке степени повышения скорости обучения сети при использовании методов предобработки, базирующихся на понижении выборочной константы Липшица сети.

Ключевые слова: нейронная сеть, обучение сети, характеристики скорости обучения, методы предобработки, константа Липшица сети.

Annotation. The paper addresses the problem of training neural networks with large volumes of training samples and carried out experimental analysis to evaluate the degree of improvement of training speed network using preprocessing techniques, based on lowering the selective Lipschitz constant network.

Keywords: neural network, learning network, the characteristics of speed training, methods of pretreatment, the Lipschitz constant of the network.

ВВЕДЕНИЕ

Практические задачи, которые используют нейросетевые механизмы, как правило, оперируют достаточно большим набором многомерных входных данных. Размер обучающей выборки может достигать тысяч и даже десятков тысяч обучающих элементов. В ходе обучения нейронных сетей градиентными методами производятся достаточно сложные вычисления, и время обучения при большой размерности обучающей выборки может достигать десятков часов или даже нескольких суток. Сеть при этом не всегда гарантированно обучится. Это объясняет актуальность задачи разработки и исследования инструментов сокращения скорости обучения нейронных сетей на базе больших обучающих выборок. В данной работе проводится анализ эффективности методов снижения выборочной константы Липшица сети как инструментов повышения скорости обучения нейросетевых алгоритмов обратного распространения ошибки. Для оценки эффективности методов предобработки, базирующихся на понижении выборочной константы Липшица сети, проводится

комплексный эксперимент, включающий: построение архитектуры сети, реализующей метод обратного распространения ошибки; генерацию разбитых на кластеры многомерных обучающих выборок, отвечающих условиям эксперимента; измерение скорости обучения сети; статистический анализ данных эксперимента.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Рассмотрим нейронную сеть, построенную по принципу многослойного персептрона с сигмоидальными активационными функциями нейронов, осуществляющую нелинейное преобразование многомерных входных сигналов в многомерные выходные сигналы. Обучающая выборка для обучения сети представляет собой набор пар вход-выход, где каждый элемент пары – это числовой вектор:

$$\begin{cases} LS = \{(x^i, y^i)\}, i = 1..N, \\ x^i = \{x_j^i\}, j = 1..p_0, \\ y^i = \{y_k^i\}, k = 1..p_1, \end{cases}$$

где N – общее число элементов обучающей выборки, p_0 – размерность входов нейронной сети, p_1 – размерность выходов.

Для того чтобы нейронная сеть могла обучиться и хорошо приблизить заданную таблично функцию f (вход – выход) необходимо, чтобы реализуемая сетью функция F при изменении входных сигналов с x^i на x^j могла изменить значение с $f^i = y^i$ на $f^j = y^j$. Сложность аппроксимации таблично заданной функции задается выборочной оценкой константы Липшица, вычисляемой по следующей формуле:

$$\Lambda_l = \max_{i \neq j} \frac{\|f^i - f^j\|}{\|x^i - x^j\|}.$$

Для оценки способности сети заданной конфигурации решить задачу, необходимо оценить константу Липшица сети и сравнить ее с выборочной оценкой. Константа Липшица сети вычисляется по следующей формуле:

$$\Lambda_n = \sup_{x,y} \frac{\|F(x) - F(y)\|}{\|x - y\|}.$$

В случае $\Lambda_n < \Lambda_l$ сеть принципиально не способна решить задачу аппроксимации функции f .

Найдем среди всех входов такую пару (или несколько пар), в которой достигается максимум $\Lambda_l = \max_{i \neq j} \frac{\|f^i - f^j\|}{\|x^i - x^j\|}$. Данные входы наиболее сложны при обучении нейронной сети, и нужно увеличить между ними расстояние. Определим среди координат входов x^i и x^j координату l , на которой достигает минимума величина $\|x_l^i - x_l^j\|$, исключив из рассмотрения совпадающие координаты. Увеличить расстояние между входами x^i и x^j можно путем специальной обработки координаты l . Один из способов заключается в разбиении координаты на две. Введем следующие обозначения:

$$\begin{cases} a = x_l^i, \\ b = x_l^j, \\ c = \frac{a-b}{2}. \end{cases}$$

Рассмотрим в координатной форме все входы обучающей выборки

$$\begin{cases} x^1 = (x_1^1, x_2^1, \dots, x_{l-1}^1, x_l^1, x_{l+1}^1, \dots, x_{p_0}^1), \\ x^2 = (x_1^2, x_2^2, \dots, x_{l-1}^2, x_l^2, x_{l+1}^2, \dots, x_{p_0}^2), \\ \dots \\ x^N = (x_1^N, x_2^N, \dots, x_{l-1}^N, x_l^N, x_{l+1}^N, \dots, x_{p_0}^N). \end{cases}$$

и произведем разбиение l -го признака на два по следующей формуле:

$$\begin{cases} x_l^i \rightarrow (x_{l_1}^i, x_{l_2}^i), i = 1..N \\ \begin{cases} x_{l_1}^i = \frac{(x_l^i - a)(b - a)}{c - a} + a, \text{ если } x_l^i < c \\ x_{l_2}^i = a \end{cases} \\ \begin{cases} x_{l_1}^i = b \\ x_{l_2}^i = \frac{(x_l^i - a)(b - a)}{c - a} + a, \text{ если } x_l^i > c \end{cases} \end{cases}$$

Для преобразованных входных векторов обучающей выборки

$$\begin{cases} x^1 = (x_1^1, x_2^1, \dots, x_{l-1}^1, x_{l_1}^1, x_{l_2}^1, x_{l+1}^1, \dots, x_{p_0}^1), \\ x^2 = (x_1^2, x_2^2, \dots, x_{l-1}^2, x_{l_1}^2, x_{l_2}^2, x_{l+1}^2, \dots, x_{p_0}^2), \\ \dots \\ x^N = (x_1^N, x_2^N, \dots, x_{l-1}^N, x_{l_1}^N, x_{l_2}^N, x_{l+1}^N, \dots, x_{p_0}^N). \end{cases}$$

производится линейный сдвиг и масштабирование:

$$\begin{cases} x_j^i = \frac{(x_j^i - x_j^{\min})(b - a)}{(x_j^{\max} - x_j^{\min})} + a \\ x_j^{\min} = \min_i x_j^i \\ x_j^{\max} = \max_i x_j^i \\ i = 1..N, j = 1..p_0, \end{cases}$$

где для сигмоидальной функции полагают $a = 0$, $b = 1$.

В практической части работы будет описан эксперимент и проведено исследование по оценке влияния рассмотренного метода снижения выборочной константы Липшица сети на скорость обучения нейронной сети.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Для генерации обучающих выборок различной степени сложности (с позиции обучения) был создан специальный генератор кластеров из многомерных числовых векторов. Генератор позволяет гибко настроить число генерируемых кластеров, минимальное расстояние между кластерами, число элементов в кластере, степень рассеивания элементов относительно центра кластера и закон рассеивания

На рисунке 1 представлены несколько примеров сгенерированных данных, спроецированных на двумерную плоскость и отмасштабированных по обеим осям в интервал $[0;1000]$. Общее число входов колеблется от 1000 до 2500.

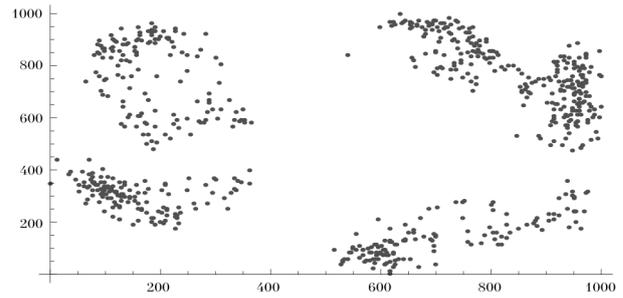
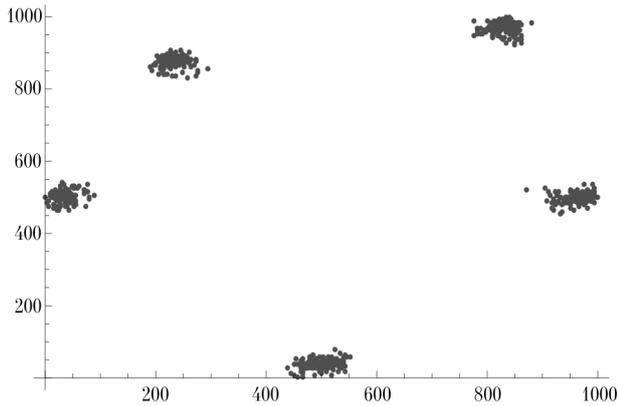


Рис. 2. Примеры сгенерированных данных, спроецированных на двумерную плоскость и отмасштабированных (среднее число итераций)

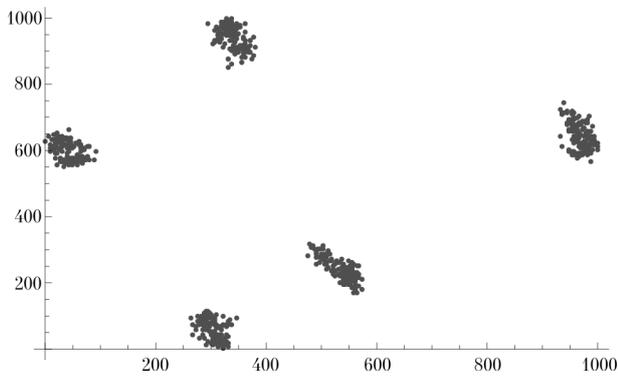
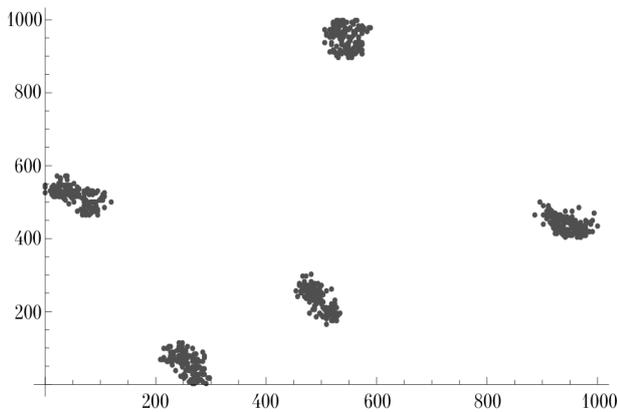


Рис. 1. Примеры сгенерированных данных, спроецированных на двумерную плоскость и отмасштабированных (большое число итераций)

Для проецирования использовались самоорганизующиеся карты признаков Кохонена применительно к входным векторам обучающих множеств. Кластеры на картинках не пересекаются – это особенность проекции с большим числом итераций. При уменьшении числа итераций при проецировании ситуация изменится, это отражено на рисунке 2.

Кластеры не имеют четко выраженных центров, присутствует множество близких входов (под степенью близости понимается евклидово расстояние), попадающих, тем не менее, в разные кластеры. Это отвечает целям эксперимента.

Для проведения исследования было сгенерировано по 500 обучающих множеств из 5–7 пересекающихся кластеров для размерности входов в 10, 25, 50, 100, 250, 500.

Таблица 1

№	Размерность входов	Число кластеров	Число элементов в кластере	Общее число элементов в выборке
1	10	5	500	2500
2	25	6	417	2500
3	50	5	500	2500
4	100	7	357	2500
5	250	5	500	2500
6	500	6	417	2500

Общее число экспериментов достигло 6000 (с учетом повторного обучения на предобработанных данных). Для проведения эксперимента использовались 8 однотипных компьютерных систем (ведомые) и 1 управляющая (ведущий). Конфигурация ведомых в таблице 2.

В ходе эксперимента генерировались обучающие выборки в соответствии с таблицей 1 и на каждой из машин (исключая ведущую) запускался процесс обучения многослойного персептрона с одним скрытым слоем и сигмоидальной функцией активации. В качестве алгоритма использовался стандартный алгоритм обратного распространения. Процесс обучения для каждой выборки повторялся дважды – без предобработки и с предобработкой. В случае успешного обучения результаты отсылались на веду-

Таблица 2

Характеристика	Значение ведомого	Значение ведущего
Марка процессора	Core i5	AMD TurionTM 64x2
Модель процессора	760	TL-56
Частота процессора	2.8 ГГц	1,8 ГГц
Сокет	S1156	S1
Чипсет системной платы	iN55 Express	AMD M690T
Интегрированная сетевая плата	10/100/1000Мбит	10/100Мбит
Тип памяти	DDR3 1333	DDR2 800
Объем оперативной памяти (ОЗУ)	4096МБ	2048МБ
Интегрированная видеоплата	нет	ATI x1150 Express
Интерфейс накопителя	SATAII	SATA
Скорость привода жесткого диска	7200rpm	5400rpm

щий компьютер и сохранялись для дальнейшего анализа. В случае неуспешного обучения заново инициализировались веса сети и обучение повторялось. Под критерием неуспешного обучения понималось хаотическое изменение ошибки сети на контрольном множестве на протяжении 500 итераций (паралич) или отсутствие значимого изменения ошибки сети на протяжении 1000 итераций (попадание в локальный минимум). Число элементов в скрытом слое составляло 60–75% от числа нейронов во входном. Коэффициент скорости обучения алгоритма имел постоянное значение 0.1. Не применялись методы контрастирования (редукции) сети. Схема алгоритма расчетов приведена на рисунке 3.

Общее время эксперимента достигло 112 дней вычислений (в среднем по 14 дней на каждую из 8 ведомых машин). Получены следующие итоговые усредненные данные:

Таблица 3

Число входов	Время на обучение в секундах	Итераций за обучение	Время на обучение (предобработанные входы) в секундах	Итераций за обучение (предобработанные входы)	% успешных обучений с первого раза
10	96	14234	74	10972	32,0%
25	154	18667	104	12606	27,0%
50	591	22873	433	16758	22,0%
100	1101	24356	875	19356	12,5%
250	2899	27819	2207	21179	7,0%
500	6317	33256	4900	25796	4,0%

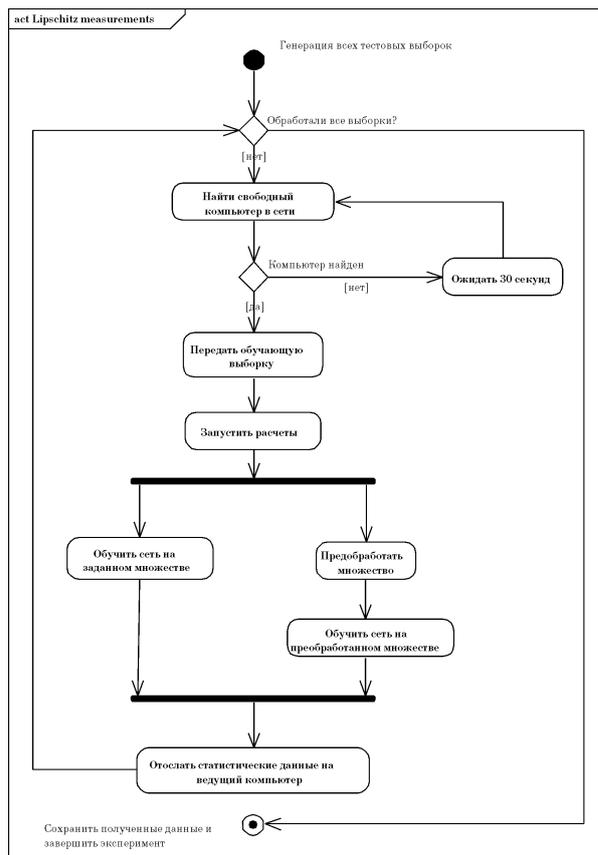


Рис. 3. Схема нейросетевого алгоритма

На следующем рис. 4 приведено сравнительное время обучения сети в случае эталонных и предобработанных входов обучающей выборки.

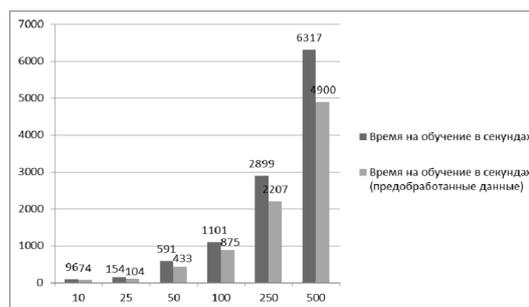


Рис. 4. Диаграмма сравнения времени обучения для исходных и предобработанных данных

Далее приведено сравнительное число итераций для обучения сети в случае эталонных и предобработанных входов обучающей выборки.

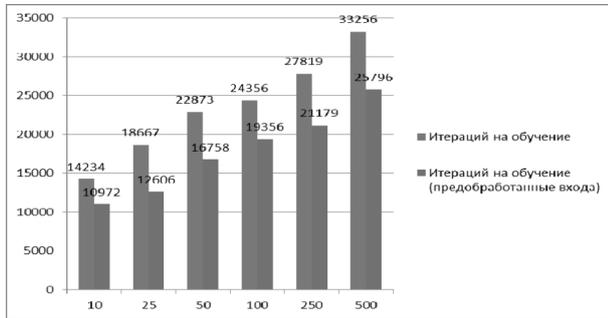


Рис. 5. Диаграмма сравнения количества итераций обучения для исходных и предобработанных данных

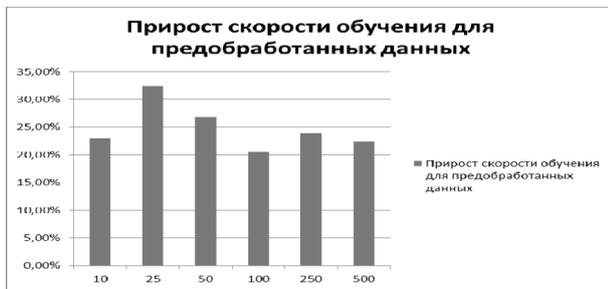


Рис. 6. Прирост скорости обучения для предобработанных данных

Азарнова Татьяна Васильевна – доктор технических наук, доцент кафедры ММИО факультета ПММ, Воронежский государственный университет. E-mail: ivdas92@mail.ru.

Терновых Илья Николаевич – аспирант кафедры ММИО факультета ПММ Воронежский государственный университет.

На основании эксперимента можно сделать вывод, что предобработка входов позволяет увеличить скорость обучения в среднем на 24,82%.

СПИСОК ЛИТЕРАТУРЫ

1. *Kohonen T.* Self-Organizing Maps. Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001. Third Extended Edition, 501 pp.

2. *Царегородцев В. Г.* Целенаправленная минимизация константы Липшица обучающей выборки и эффекты в нейронных сетях / В. Г. Царегородцев // Проблемы нейрокибернетики : Материалы Юбилейной междунар. конф. по нейрокибернетике. – Ростов-на-Дону. – 2002. – Т.2. – 6 с.

Azarnova T.V. – Doctor of Technic Sciences, Associate Professor, the dept. of the Mathematical Methods of Operation Research , Voronezh State University. E-mail: ivdas92@mail.ru.

Ternovich I.N. – Post-Graduate Student , , the dept. of the Mathematical Methods of Operation Research , Voronezh State University.