

УДК 004.85

DOI: 10.15827/0236-235X.126.258-262

Дата подачи статьи: 26.11.18

2019. Т. 32. № 2. С. 258–262

Модифицированный алгоритм обучения нейронных сетей

В.Н. Зуев^{1,2}, зав. лабораторией, аспирант, zvn_tver@mail.ru

В.К. Кемайкин², к.т.н., доцент, vk-kem@mail.ru

¹ НИИ «Центрпрограммистем», г. Тверь, 170024, Россия

² Тверской государственный технический университет, кафедра «Информационные системы», г. Тверь, 170026, Россия

В статье исследуется эвристическое улучшение алгоритма обратного распространения ошибки с использованием пакетного режима обучения.

Алгоритм обратного распространения ошибки является одним из самых распространенных алгоритмов обучения нейронных сетей. Использование его сопряжено с рядом сложностей, главная из которых – обеспечение приемлемой способности к обобщению нейронной сети. Способность к обобщению полученных знаний является одним из важнейших свойств нейронной сети и заключается в генерации нейронной сетью ожидаемых значений на данных, не участвующих непосредственно в процессе обучения. Однако использование зашумленных и ошибочных данных может привести к переобучению и снижению способности к обобщению обученной нейронной сети.

Рассматриваемые в статье вопросы являются важной частью процесса обучения нейронных сетей. Предложенный метод дает возможность более эффективно рассчитывать значения целевой функции, лежащей в основе алгоритма обратного распространения ошибки, а также игнорировать ошибочные значения в данных для обучения, исключая их на ранних стадиях обучения. Кроме того, метод позволяет использовать для обучения нейронных сетей неоднородные выборки данных, а также учитывать при обучении априорную информацию о ценности отдельных примеров.

В статье приведен алгоритм работы данного метода. Использование метода позволит повысить точность работы нейронной сети для задач классификации и аппроксимации.

Ключевые слова: нейронная сеть, обратное распространение, целевая функция, функция ошибки, алгоритм обучения.

В статье представлен метод отбора примеров в обучающую выборку с учетом их влияния на процесс обучения сети. Показано, что в задаче обучения сети с учителем при использовании пакетного режима [1] скорость обучения и качество работы обученной нейронной сети зависят от величины ошибки на выходе, обусловленной не только уровнем обученности сети, но и характеристиками обучающей выборки: типичностью примеров, их количеством для различных исходов, а также достоверностью. Целью работы является повышение качества обучения сети за счет оценки обучающего множества и формирования обучающей выборки.

Обучение нейронной сети – это процесс минимизации в пространстве обучаемых параметров функции оценки. При использовании метода обратного распространения ошибки корректировка синаптической карты весов нейронной сети выполняется после подачи всех обучающих примеров по усредненному значению градиента целевой функции, формулируемой в виде квадратичной суммы разностей между фактическими и ожидаемыми значениями выходных сигналов [2]:

$$E(w) = \frac{1}{2} \sum_{k=1}^m (y_k - d_k)^2, \quad (1)$$

где y – выходное значение нейронной сети; d – желаемое значение выхода; m – количество нейронов в выходном слое; k – номер нейрона в выходном слое.

Основная идея рассматриваемого в данной работе подхода состоит в том, что используемые для обучения примеры должны получать весовые коэффициенты, улучшающие нейронную сеть, полученную в процессе обучения.

Необходимость использования весов примеров при обучении может быть обусловлена следующими причинами:

- 1) один из примеров плохо обучается;
- 2) число примеров разных классов в обучающем множестве сильно отличается;
- 3) примеры в обучающем множестве имеют различную достоверность [3].

Первая причина актуальна в случае, когда имеется априорная информация о значимости примера и необходимо, чтобы нейронная сеть научилась его воспринимать. Однако в силу своих особенностей пример не учитывается

при обучении нейронной сети обычными методами.

Вторая причина актуальна для случая, когда в обучающей выборке есть классы, число примеров которых мало по сравнению с другими классами. В пакетном режиме обучения ошибка, рассчитанная по этим примерам, может потеряться в суммарной ошибке по всей выборке, в результате чего такие примеры могут быть проигнорированы. Ошибка по этим примерам останется большой, и нейронная сеть не научится их воспринимать. По этой причине существует критика пакетного режима обучения. Так, в работе [4] показывается преимущество *online*-обучения в контексте объема вычислений, осуществляемых в процессе обучения. Однако последовательный режим обучения не решает данную проблему. Еще одним подходом, используемым для решения данной проблемы, является уменьшение размеров нейронной сети. В работе [5] доказана несостоятельность данного метода.

Третья причина актуальна при наличии в данных ошибочных значений – выбросов, которые могут давать большую ошибку. Пытаясь научиться воспринимать такие значения, нейронная сеть может ухудшить свою способность к обобщению [6].

Для решения проблемы, вызванной первой и второй причинами, обучающим примерам необходимо присвоить весовые коэффициенты. Они будут использоваться при расчете ошибки обучения E и усиливать вклад выбранных примеров в суммарную ошибку обучения.

Для решения третьей проблемы в работе [7] предложена редукция данных, основанная на ограничении диапазона значений признака, однако при данном подходе существует вероятность исключения значимых примеров.

Учитывая все перечисленные особенности, становится очевидной необходимость более эффективного представления обучающей выборки.

Так, в работе [8] предлагается задавать пороговое значение ошибки, при превышении которого пример не должен рассматриваться. Более гибкий подход реализован в *Lazy training* (ленивое обучение) [9], однако этот алгоритм предназначен исключительно для задач классификации. Его основная идея заключается в том, что для коррекции весов используются только те наблюдения, которые были классифицированы неправильно. Помимо узкой направленности, этот алгоритм имеет и другие недостатки. Для решения проблемы, вызван-

ной третьей причиной, необходим алгоритм, учитывающий описанные недостатки.

Основная идея рассматриваемого подхода состоит в том, что обучающие примеры, ошибки по которым оказываются слишком большими, не должны участвовать в обучении нейронной сети. Однако задание граничного значения ошибки, разделяющего исходное множество на используемые и неиспользуемые примеры, неэффективно. Если использовать для обучения часть примеров, то ошибка выхода по ним будет постепенно уменьшаться. При этом ошибка на неиспользуемых примерах может расти. Таким образом, использование граничного значения может привести к следующему эффекту. Если ошибка, полученная на рассматриваемом примере, меньше заданного значения, то при следующей итерации этот пример не будет использован. При этом ошибка по этому примеру может вырасти и снова преодолеть установленную границу. При очередной итерации данный пример опять будет использован в обучающей выборке. В результате нейронная сеть перестанет обучаться.

Для избежания подобного эффекта предложено использовать два граничных значения ошибки обучения: нижнее (E_1) и верхнее (E_2), которые определяют три области и являются функцией количества циклов обучения. Для каждого примера в зависимости от значения ошибки E и его положения относительно значений E_1 и E_2 принимается решение о выборе значения весового коэффициента:

- $E < E_1$ – пример не усиливается весовым коэффициентом (получает весовой коэффициент, равный единице);
- $E_1 < E < E_2$ – пример усиливается весовым коэффициентом;
- $E > E_2$ – пример игнорируется (получает нулевой весовой коэффициент).

Весовые коэффициенты рассчитываются индивидуально для каждого примера на каждом шаге обучения.

В начале процесса обучения используются все имеющиеся примеры. Поскольку нейронная сеть перед началом обучения инициализируется случайными значениями, распределение ошибок по используемым примерам равномерно. При этом ошибки по всем примерам не превышают значение E_2 . В процессе обучения нейронной сети ошибка обучения приобретает нормальный вид распределения. У большинства примеров ошибка стремится к нулю, у части примеров она остается большой. Попав в интервал $E_1 < E < E_2$, пример получает весовой

коэффициент и ошибка по нему начинает уменьшаться быстрее.

Как только ошибка примера пересекает границу E_1 , он получает единичный коэффициент. Если при дальнейшем обучении ошибка примера превысит E_1 , он снова получит усиливающий весовой коэффициент. Если ошибка примера превысила значение E_2 , то пример считается выбросом и не участвует в дальнейшем обучении. Направления изменений ошибок показаны пунктирными стрелками на рисунке 1.

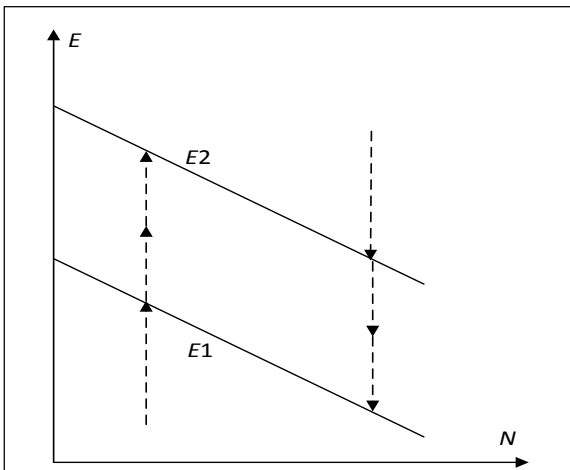


Рис. 1. Направления изменений ошибок выхода в процессе обучения

Fig. 1. Directions for changing output errors during the learning process

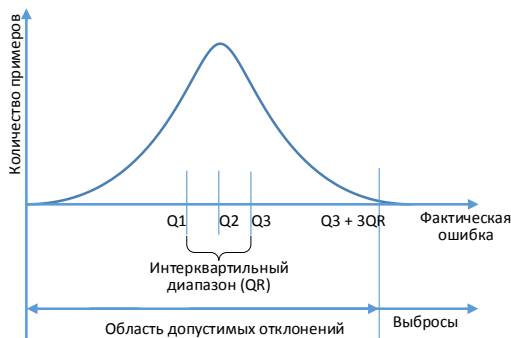


Рис. 2. Диапазоны расположения допустимых значений и экстремальных выбросов

Fig. 2. The ranges of permissible values and extreme emissions

Граничные значения E_1 и E_2 рассчитываются на каждом шаге обучения сети. Для расчета значений E_1 и E_2 используется критерий, основанный на интерквартильном размахе. Метод основан на вычислении трех квартилей, делящих данные на четыре равные группы по

ошибке обучения. Интерквартильный размах [10] считается как разность между первой и третьей квартилями:

$$IQR = Q_3 - Q_1. \tag{2}$$

Значения E_1 и E_2 рассчитываются следующим образом:

$$E_1 = Q_1, \tag{3}$$

$$E_2 = Q_3 + Q_R. \tag{4}$$

На рисунке 2 показана схема, выражающая эти определения.

Предложенная реализация алгоритма обучения позволяет избежать переобучения нейронной сети, связанного с чрезмерным стремлением достичь нулевой ошибки. При этом данный алгоритм в большей степени учитывает свойства малочисленных групп примеров, чем стандартный алгоритм. Блок-схема алгоритма изображена на рисунке 3.



Рис. 3. Блок-схема алгоритма обучения нейронной сети с использованием адаптивной функции ошибки

Fig. 3. The flowchart of the neural network training algorithm using the adaptive error function

Таким образом, алгоритм будет следующим.

Шаг 1. Формируем обучающую выборку, используя все обучающее множество.

Шаг 2. Вычисляем значение среднего градиента для предоставленной выборки.

Шаг 3. Выполняем корректировку весов нейронной сети.

Шаг 4. Устанавливаем веса примерам, используя данные об ошибках выхода. Для этого создается матрица, содержащая флаг для каждого наблюдения. Если ошибка, соответствующая наблюдению, меньше значения ошибки E_1 , этому наблюдению ставится единичный коэффициент. Если ошибка больше значения ошибки E_2 , наблюдению ставится нулевой коэффициент. Если ошибка находится в диапазоне E_1 – E_2 , наблюдению ставится усиливающий коэффициент.

Шаг 5. Используя полученный набор флагов, разделяем выборку на две части: используемую для следующего уточнения весов нейронной сети и игнорируемую.

Шаг 6. Создаем новую обучающую выборку для следующей итерации.

Шаг 7. Возвращаемся к шагу 2.

Проведенные исследования показали, что применение описанного подхода позволяет повысить качество обучения нейронной сети для классификации и регрессии.

Литература

1. Дьяконов В.П., Круглов В.В. MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. М.: Солон-Пресс, 2006. 456 с.

2. Оссовский С. Нейронные сети для обработки информации; [пер. с польск. И.Д. Рудинского]. М.: Финансы и статистика, 2004. 344 с.

3. Миркес Е.М. Нейроинформатика. Красноярск: Изд-во КГТУ, 2002. 347 с.

4. Царегородцев В.Г. Общая неэффективность использования суммарного градиента выборки при обучении нейронной сети // Нейроинформатика и ее приложения: матер. XIII Всерос. семинара. 2004. С. 145–151.

5. Царегородцев В.Г. Редукция размеров нейросети не приводит к повышению обобщающей способности // Нейроинформатика и ее приложения: матер. XII Всерос. семинара. Красноярск, 2004. С. 163–165.

6. Хайкин С. Нейронные сети: полный курс; [пер. с англ. Н.Н. Куссуль, А.Ю. Шелестова]. М.: Вильямс, 2006. 1104 с.

7. Царегородцев В.Г. Оптимизация предобработки данных для обучаемой нейросети: критерии оптимальности предобработки // Междунар. конф. по нейрокибернетике: сб. докл. Ростов н/Д, 2005. Т. 2. С. 64–67.

8. Xiao-Ping Zhang. Thresholding neural network for adaptive noise reduction. Proc. IEEE Transactions on Neural Networks, 2001, vol. 12, no. 3, pp. 567–584.

9. Rimer M.E., Anderson T.L. and Martinez T.R. Improving backpropagation ensembles through lazy training. Proc. IEEE IJCNN'01, 2001, pp. 2007–2112.

10. Певзнер М.З. Систематизация, статистический анализ данных, контроль и управление производственными процессами. Киров: Изд-во ВятГУ, 2012. 165 с.

An improved neural network training algorithm

V.N. Zuev^{1,2}, Head of Laboratory, Postgraduate Student, zvn_tver@mail.ru

V.K. Kemajkin², Ph.D. (Engineering), Associate Professor, vk-kem@mail.ru

¹ R&D Institute Centerprogramsystem, Tver, 170024, Russian Federation

² Tver State Technical University, chair of Information's Systems, Tver, 170026, Russian Federation

Abstract. The paper describes heuristic modification of backpropagation algorithm using for remote batching.

The backpropagation algorithm is a common algorithm for neural network training. It causes some difficulties. The main problem is enabling the generalizing of a neural network. The ability of generalizing is a most important characteristics of a neural network. It assumes that a neural produces anticipated values on data that is not a part of a training process. However, using of noisy data causes retraining and decreasing of a generalizing ability of a neural network.

The problems considered in the paper are an important part of a neural network training process. The paper describes a method to use objective functions effectively. The proposed method allows more effective calculating of the values of a goal function that is a base of the backpropagation algorithm. It also ignores failure values in training data and excludes them at earlier stages. In addition, the method allows using heterogeneous

data samples for training neural networks, as well as taking into account prior information on the significance of some examples when training.

The paper describes the algorithm of the proposed method. The method will improve the accuracy of a neural network for classification and regression tasks.

Keywords: neural network, backpropagation, objective function, error function, learning algorithm.

References

1. Dyakonov V.P., Kruglov V.V. *MatLab 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Artificial Intelligence and Bioinformatics Tools*. Moscow, Solon-Press, 2006, 456 p.
2. Asowsky S. *Neural Networks for Information Processing*. Moscow, Finansy i statistika Publ., 2004, 344 p.
3. Mirkes E.M. *Neuroinformatics*. Krasnoyarsk, 2002, 347 p.
4. Tsaregorodtsev V.G. The overall inefficient use of a sample total gradient when training a neural network. *Proc. 13th All-Russ. Workshop "Neuroinformatics and Its Applications"*. Krasnoyarsk, 2004, pp. 145–151 (in Russ.).
5. Tsaregorodtsev V.G. Reducing of neural network sizes does not lead to an increasing generalization ability. *Proc. 13th All-Russ. Workshop "Neuroinformatics and Its Applications"*. Krasnoyarsk, 2004, pp. 163–165 (in Russ.).
6. Khaykin S. *Neural Networks: a Comprehensive Foundation*. 2nd ed., Moscow, Vilyams Press, 2006. 1104 p.
7. Tsaregorodtsev V.G. Optimization of data preprocessing for a trained neural network: preprocessing optimality criteria. *Proc. 14th Intern. Conf. on Neurocybernetics*. Rostov-on-Don, 2005, vol. 2, pp. 64–67 (in Russ.).
8. Xiao-Ping Zhang. Thresholding neural network for adaptive noise reduction. *Proc. IEEE Trans. on Neural Networks*. 2001, vol. 12, no. 3, pp. 567–584.
9. Rimer M.E., Anderson T.L., Martinez T.R. Improving backpropagation ensembles through lazy training. *Proc. IEEE Intern. Joint Conf. on Neural Networks IJCNN'01*. 2001, pp. 2007–2112.
10. Pevzner M.Z. *Systematization, Statistical Data Analysis, Control and Management of Manufacturing Processes*. Kirov: VyatGU Press, 2012, 165 p.

Для цитирования

Зуев В.Н., Кемайкин В.К. Модифицированный алгоритм обучения нейронных сетей // Программные продукты и системы. 2019. Т. 32. № 2. С. 258–262. DOI: 10.15827/0236-235X.126.258-262.

For citation

Zuev V.N., Kemajkin V.K. An improved neural network training algorithm. *Software & Systems*. 2019, vol. 32, no. 2, pp. 258–262 (in Russ.). DOI: 10.15827/0236-235X.126.258-262.