В.А. КРИСИЛОВ, К.В. ЧУМИЧКИН
Одесский национальный политехнический университет. Институт компьютерных систем.
VictorK@OL405.paco.net, k_chum@ukr.net

УСКОРЕНИЕ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ЗА СЧЕТ АДАПТИВНОГО УПРОЩЕНИЯ ОБУЧАЮЩЕЙ ВЫБОРКИ

Аннотация

В работе предпринята попытка улучшить параметры сходимости алгоритмов обучения НС, основанных на методе градиентного спуска, за счет упрощения ОВ на ранних этапах обучения с последовательным усложнением (детализацией) на последующих этапах.

V.A. KRISSILOV, K.V. CHUMICHKIN
Odessa national polytechnic university. Computer systems institute.
VictorK@OL405.paco.net, k_chum@ukr.net

THE ACCELERATION OF NEURAL NETWOTKS TRAINING WITH ADAPTIVE TRAINING PATTERN SIMPLIFICATION

Abstract

The approach to improve convergence parameters of gradient-based neural networks training methods is discussed in this work. The method based on adaptive training pattern simplification is proposed.

Введение

С увеличением размерности задач, возлагаемых на интеллектуальные системы, повышение скорости и качества обучения НС приобретает все большее значение. Подтверждением этому является значительное количество исследований, направленных на ускорение обучения НС. Во многих предложенных методах увеличение скорости достигается за счет модификации метода градиентного спуска, основанной на предположении о том или ином характере исходных данных [1]. Также часто используется динамическое изменение шага обучения [2, 3], дополнительные методы вывода НС из локального минимума [4]. Однако перечисленные методы не учитывают то обстоятельство, что дополнительное повышение скорости и качества обучения может быть достигнуто и за счет обработки данных, на которых это обучение происходит.

Обратившись к естественным обучающимся системам, можно заметить, что чаще всего обучение происходит не сразу на всем обучающем множестве (которым для естественных систем являются объекты реального мира), а на его упрощенной модели, отражающей лишь некоторые примеры и закономерности. По мере усвоения более простого материала модель становится все более подробной и адекватной. Т.е. обучение происходит как бы «от простого к сложному». Исследование применимости такого подхода для повышения качества и скорости обучения ИНС и является целью данной работы.

Сложность ОВ, способы ее снижения

Под сложностью ОВ подразумевается сложность ее аппроксимации нейронной сетью, которую для пары наборов $(X;Y)_i$, $(X;Y)_j$ можно охарактеризовать следующим образом [5]:

$$L_{ij} = \frac{\left\| Y_i - Y_j \right\|}{\left\| X_i - X_j \right\|} \tag{1}$$

где X и Y – соответственно входные и выходные вектора.

Сложность воспроизведения всей ОВ может быть получена расчетом среднего или максимального и минимального значений L_{ij} для всех пар наборов. Применение соотношения (1), в теории непрерывных функций называемого константой Липшица, с целью оценки обучающей возможности ОВ неоднократно обсуждалось в литературе и показало свою практическую применимость [5, 6].

Одним из способов снижения сложности ОВ является искусственное сближение выходных векторов для наборов, входные вектора которых находятся близко друг к другу. При этом выходной вектор набора k упрощенной выборки ОВ’ рассчитывается как среднее выходных векторов наборов исходной выборки ОВ, взвешенное по функции от расстояния до входного вектора k-го набора (2).

$$Y_k' = \frac{\sum_i Y_i \cdot c_{ik}}{\sum_i c_{ik}} \tag{2}$$

Роль взвешивающей функции может выполнять функция от расстояния между входными векторами, удовлетворяющая следующим условиям:

1. Существовать и быть неотрицательной на всем множестве возможных значений расстояния.
2. Убывать с увеличением расстояния.

3. В зависимости от некоторого параметра α изменять скорость убывания.

Таким образом, параметр α определяет «крутизну» взвешивающей функции и задает степень упрощения исходной выборки.

Одной из наиболее известных и широко применяемых функций, удовлетворяющих перечисленным условиям, является функция Гаусса (3), которую и предлагается использовать в качестве взвешивающей в данной работе.

$$c_{ik} = e^{-\left(\frac{\|X_k - X_i\|}{\alpha}\right)^2} \tag{3}$$

Ниже приведен пример упрощения функции одной переменной при различных значениях параметра α.

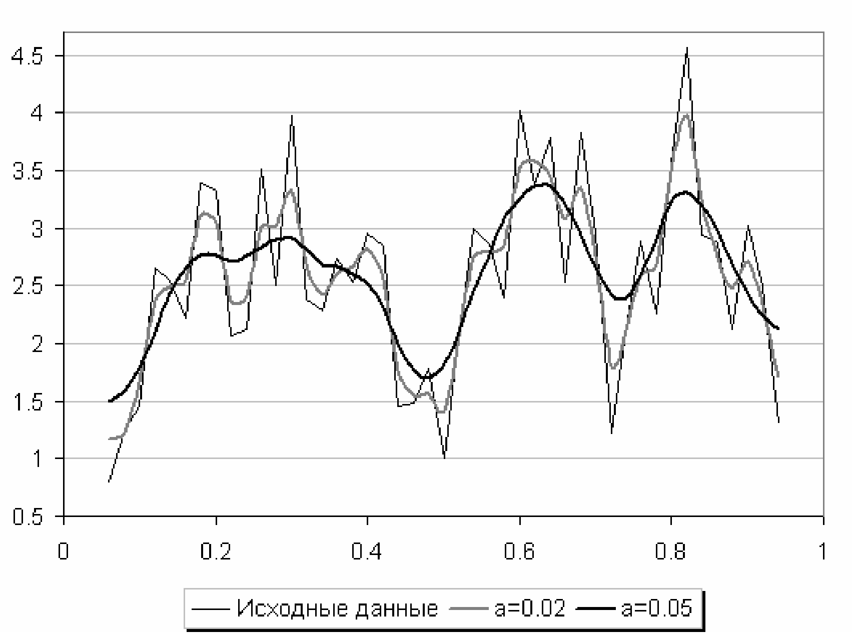

Рис 1. Пример упрощения исходной функции.

Усовершенствованный алгоритм обучения НС

Для количественной оценки упрощения ОВ в процессе обучения НС рассмотрим следующие величины:

δ(OB'; OB) – отклонение упрощенной выборки от исходной;

δ(НС; ОВ') – ошибка НС на упрощенной выборке ОВ';
δ(НС; ОВ) – ошибка НС, обученной на упрощенной выборке, рассчитанная для исходной выборки ОВ.
Пусть эти величины определены как среднее расстояние между выходными векторами в выбранной метрике. Тогда имеет место неравенство:

$$\delta(OB';OB)+\delta(HC;OB')\geq\delta(HC;OB) \tag{4}$$

Это позволит воспользоваться левой частью неравенства в качестве критерия остановки обучения, а не тратить время на дополнительный расчет *δ(НС; ОВ).*

Вместе с тем, нет необходимости обучаться на ОВ' с точностью, большей точности самой ОВ'. Следовательно, должно выполняться соотношение:

$$\delta(OB';OB)\leq\delta(HC;OB') \tag{5}$$

Учитывая (4) и (5), можно предложить следующий алгоритм обучения НС:

1. Задается начальное значение параметра упрощения.
2. Формируется упрощенная выборка ОВ' и рассчитывается *δ(ОВ'; ОВ).*
3. Производится обучение НС выборке ОВ' до тех пор, пока не выполнится одно из условий:
   a) $\delta(OB';OB)+\delta(HC;OB')\leq\delta_{доп}$, где $\delta_{доп}$ – допустимая ошибка, определяемая требуемой точностью решения задачи. Выполнение условия означает окончание обучения.
   b) $\delta(OB';OB)\geq\delta(HC;OB')$. При выполнении – уменьшение параметра α и переход на шаг 2.

Данный алгоритм позволяет изменить процесс обучения так, что в начале НС будет обучаться основным тенденциям и закономерностям, несколько теряя в точности, но зато не повторяя возможно присутствующий в исходной выборке шум. По мере усложнения выборка ОВ' будет приближаться к исходной и, в конечном итоге, либо повторит ее, либо обеспечит достаточную точность решения задачи, что для НС будет означать финальный этап обучения.

Особенности реализации

Следует отметить, что реализация обучения НС по предложенному алгоритму несколько усложняет процесс обучения и может потребовать дополнительных затрат. Однако, используя некоторые особенности задачи, быстродействие можно повысить, сохранив преимущество в скорости обучения. Отметим возможные пути ускорения алгоритма:

1. В начале процесса обучения, когда параметр α велик, нет необходимости учитывать все наборы для расчета усредненного значения, достаточно выполнить расчет для некоторого количества наборов, отобранных по методу Монте-Карло.
2. При малом значении α с увеличением расстояния взвешивающая функция быстро стремится к нулю. Поэтому при расчете значений выходного вектора набора упрощенной выборки можно пренебречь вкладом наборов, удаленных от него.
3. Для выборок большого размера п.1 и п.2 могут одновременно иметь место.

Кроме того, благодаря независимости процессов обучения НС и упрощения ОВ, кардинальным решением проблемы повышения быстродействия может быть их распараллеливание.

Практические результаты

В ходе предварительных экспериментов производилось сравнение ошибок обучения МНС с применением предложенного подхода и без него. Исходными данными служила функция 2-х переменных с шумовой составляющей, взятая из набора функций, сформированных для сравнения обучающих алгоритмов [7]. Обучающая выборка содержала 225 наборов, а контрольная - 10000. Сеть обучалась в течение 5000 эпох, затем рассчитывалась ошибка на контрольной выборке (КВ).

За время обучения было произведено 4 итерации упрощения ОВ. К моменту окончания обучения среднеквадратичное отклонение упрощенной выборки от исходной составляло 0,13, среднеквадратичная ошибка НС на упрощенной выборке – 0,14, на исходной – 0,17. Ошибка на КВ составила 0,48.

В случае с не изменяющейся ОВ, к окончанию обучения ошибка на ОВ составила 0,18; на КВ – 0,56.

В результате для случая с упрощением ОВ было отмечено снижение ошибок как на ОВ (6%), так и на КВ (15%).

Выводы

Таким образом, использование адаптивного упрощения ОВ позволяет снизить время и, что более важно, повысить качество обучения НС. Это достигается в основном за счет снижения избыточной подробности обучающего множества на ранних этапах обучения, что вполне характерно для естественных обучающихся систем.

Используемые в подходе преобразования относятся только к исходным данным и не затрагивают алгоритма настройки весовых коэффициентов НС. Это делает подход совместимым со многими известными методами ускоренного обучения НС, тем самым давая дополнительный выигрыш во времени и качестве обучения.

В отношении дальнейшего развития подхода можно отметить исследование неравномерного упрощения ОВ, когда коэффициент упрощения различен для каждого набора и определяется с учетом ошибки НС на данном наборе, а не на всей выборке в среднем.

Список литературы

1. Parker D. B. 1987. Second order back propagation: Implementing an optimal *0(n)* approximation to Newton's method as an artificial neural network. Manuscript submitted for publication.
2. Крисилов В.А., Олешко Д.Н., Лобода А.В. Методы ускорения нейронных сетей.// Вестник СевГТУ. Информатика, электроника, связь, Вып. 32, 2001, с. 19.
3. Комарцова Л.Г., Модифицированный алгоритм обучения многослойного персептрона. // Научная сессия МИФИ – 2003. V всероссийская научно-техническая конференция «Нейронформатика-2003»: Сборник научных трудов. М.: МИФИ, 2003. Ч.1. С. 85-92.
4. Wasserman P. D. 1988a. Combined backpropagation/Cauchy machine. Proceedings of the International Neural Network Society. New York: Pergamon Press.
5. Миркес Е.М., Нейрокомпьютер. Проект стандарта. – Новосибирск: Наука, Сибирская издательская фирма РАН, 1998. С. 56-60.
6. Тарасенко Р.А., Крисилов В.А. Предварительная оценка качества обучающей выборки для нейронных сетей в задачах прогнозирования временных рядов. // Труды Одесского политехнического университета. – Одесса, 2001. – Вып.1. – С. 90-93.
7. Hwang J.-N., Lay S.-R., Maechler M., Martin R. D. and Schimert J. (1994) "Regression modeling in back-propagation and projection pursuit learning", IEEE Transactions on Neural Networks, vol. 5, no. 3, pp. 342-353, [http://www.cs.toronto.edu/~delve/data/hwang/hwangDetail.html].