

В.И. Дубровин, С.А. Субботин – «Нейроинформатика-2002»

В.И. ДУБРОВИН, С.А. СУББОТИН

Запорожский национальный технический университет
e-mail subbotin@zstu.zaporizhzhе.ua

АЛГОРИТМ УСКОРЕННОГО ОБУЧЕНИЯ ПЕРСЕПТРОНОВ

Аннотация

Разработан алгоритм, позволяющий существенно ускорить сходимость обучения многослойных нейросетей путем слежения за процессом обучения сети на основе градиентного алгоритма. Исследованы свойства следающего алгоритма при решении практических задач диагностики. Предложен способ разбиения исходной выборки на обучающую и контрольную выборки.

V.I. DUBROVIN, S.A. SUBBOTIN

Zaporozhye National Technical University
e-mail subbotin@zstu.zaporizhzhе.ua

ALGORITHM OF QUICKEN TRAINING OF PERCEPTRONS

Abstract

The algorithm permitting essential to speed up convergence of multilayer neural networks training is developed. It based on the watching of learning process of a network that trained by gradient algorithm. The properties of watching algorithm are investigated at the solution of diagnostic problems. The method of source sample splitting on learning and test samples is offered.

Введение

Одной из важнейших задач для отечественной промышленности является обеспечение высокого качества выпускаемой продукции и продление ресурса эксплуатируемого оборудования. Для решения этой задачи необходимо иметь адекватные модели объектов и процессов, позволяющие своевременно осуществлять диагностические мероприятия как на этапе производства, так и в процессе эксплуатации.

Достаточно эффективным средством для построения таких моделей являются искусственные нейронные сети (НС), обладающие высокими адаптивными и аппроксимационными способностями [1].

<http://neuroschool.narod.ru>

В качестве наиболее часто используемой на практике модели НС выступает многослойная НС (МНС) - многослойный перцептрон, которая способна обучаться аппроксимации многомерных функций по точечным данным.

Однако, обучение МНС аппроксимации сложных существенно нелинейных зависимостей представляет собой достаточно длительный процесс, который при большом объеме многомерных экспериментальных данных может занимать продолжительное время даже на современных ЭВМ. Поэтому представляется достаточно актуальной разработка методов, позволяющих увеличить скорость обучения МНС, что и является целью данной работы.

1. Многослойный перцептрон

МНС характеризуется следующими параметрами и свойствами: M - число слоев сети, N_μ - число нейронов μ -го слоя, связи между нейронами в слое отсутствуют, нейроны соседних слоев соединяются между собой: выходы нейронов μ -го слоя, $\mu = 1, 2, \dots, M-1$ поступают на входы нейронов только следующего $\mu+1$ -го слоя. Внешний векторный сигнал x поступает на входы нейронов только первого слоя, выходы нейронов последнего M -го слоя образуют вектор выходов сети $y^{(M)}$.

Каждый j -й нейрон μ -го слоя (μj -й нейрон) преобразует входной вектор $x^{(\mu, j)}$ в выходную скалярную величину $y^{(\mu, j)}$.

$$y^{(\mu, j)} = \psi \left(w_0^{(\mu, j)} + \sum_{i=1}^N w_i^{(\mu, j)} x_i^{(\mu, j)} \right),$$

где $w_0^{(\mu, j)}$ – порог, $w_i^{(\mu, j)}$ – вес i -го входа j -го нейрона μ -го слоя, ψ - функция активации, которая является монотонной и ограниченной. В частности, при неотрицательных выходах нейрона такой функцией может быть сигмоидная функция $\psi(x) = 1/(1+e^{-x})$ [2].

Целевой функцией обучения МНС, как правило, является минимум среднеквадратической ошибки сети. Обучение МНС традиционно осуществляют на основе градиентных алгоритмов [3,4], для которых корректирующее правило весов МНС в обобщенной матричной форме имеет вид:

$$w_{k+1} = w_k + s_k(w_k),$$

где w_k, w_{k+1} – множество весов и порогов сети на k и $k+1$ циклах обучения, соответственно, s_k – корректирующее приращение - некоторый оператор, учитывающий вклад весов в ошибку обучения.

2. Следящий алгоритм обучения

В том случае, когда целевая функция обучения МНС является достаточно гладкой, не содержит сложных изгибов и большого количества локальных минимумов, традиционно применяемые градиентные алгоритмы обучения НС позволяют достигнуть цели обучения, обеспечивая достаточно быструю сходимость.

Когда же целевая функция является существенно нелинейной и содержит большое количество локальных минимумов, а объем обучающих данных велик, градиентные алгоритмы обеспечивают быструю сходимость только в начале обучения, после чего во многих случаях сходимость обучения существенно замедляется, что можно объяснить попаданием сети в локальные минимумы и недостаточно большим корректирующим приращением, чтобы быстро выбраться из локального минимума.

Поэтому необходимо разработать алгоритм, позволяющий ускорить сходимость градиентных алгоритмов обучения МНС в случае попадания сети в локальные минимумы.

Для ускорения работы градиентных алгоритмов при попадании НС в локальный минимум необходимо решить две задачи:

- 1) обнаружить, что сеть попала в локальный минимум и скорость сходимости является низкой;
- 2) вывести сеть из локального минимума.

Для решения первой задачи предлагается следить за процессом обучения МНС, запоминая значения целевой функции на каждом цикле обучения сети, и в случае, если в течение заданного количества циклов целевая функция уменьшилась менее чем на заданную величину, выполнять процедуру вывода сети из локального минимума.

Вывод сети из локального минимума предлагается осуществлять, изменяя значения весов и порогов сети на некоторые относительно небольшие величины таким образом, чтобы сеть сохранила весь предыдущий положительный опыт и в то же время могла изменить свое положение в многомерном пространстве весов и порогов.

Заметим, что коррекция весов сети даже на достаточно малую величину может приводить к существенным изменениям, причем, не только положительным, но и к отрицательным, когда следящий алгоритм ухудшает сходимость, достигнутую градиентным алгоритмом. Поэтому результаты изменений весов МНС на основе следящего алгоритма должны приниматься только в том случае, когда они улучшают сходимость, в противном

случае должны восстанавливаться значения весов, полученные после применения корректирующего правила градиентного алгоритма.

Обобщая вышесказанное, запишем следящий алгоритм.

Шаг 1. Инициализация параметров градиентного алгоритма обучения МНС [4]. Инициализация параметров следящего алгоритма: шага α , размера окна слежения Δt (в циклах), критерия целесообразности применения алгоритма слежения на данном этапе ξ и указателя ячейки окна слежения p_t : $p_t = 1$. Резервирование памяти для окна слежения $\text{Err}(\Delta t)$, здесь $\text{Err}(\Delta t)$ – массив Δt элементов.

Шаг 2. Установить счетчик циклов обучения $\text{epoch}=0$.

Шаг 3. Если $\text{epoch} > \text{Epochs}$, где Epochs – заданное максимальное допустимое количество циклов обучения НС, тогда перейти на шаг 11, в противном случае – перейти на шаг 4.

Шаг 4. Вычислить значение perf целевой функции обучения МНС.

Шаг 5. Если номер ячейки памяти $p_t \leq \Delta t$, тогда принять $\text{Err}(p_t) = \text{perf}$, $p_t = p_t + 1$; в противном случае, принять $p_t = 1$, $\text{Err}(p_t) = \text{perf}$.

Шаг 6. Проверить критерий останова для градиентного алгоритма. Если градиентный алгоритм должен прекратить работу, то перейти на шаг 11, в противном случае выполнить коррекцию весов для данного цикла обучения на основе градиентного алгоритма.

Шаг 7. Если $\text{epoch} > \Delta t$ и $|\text{perf} - \text{Err}(\text{mod}(\text{epoch}, \Delta t) + 1)| / \Delta t < \xi$, где $\text{mod}(a, b)$ – остаток от целочисленного деления a на b , тогда перейти на шаг 8, в противном случае – перейти на шаг 10.

Шаг 8. Принять: $w^* = R(w)$, где w и w^* – наборы значений весов и порогов МНС, соответственно, до и после применения корректирующего правила весов НС следящего алгоритма $R(w)$, и вычислить значение целевой функции perf^* после применения правила $R(w)$.

Шаг 9. Если $\text{perf}^* < \text{perf}$, где perf – значение целевой функции до применения корректирующего правила весов следящего алгоритма, тогда установить: $w = w^*$.

Шаг 10. Установить: $\text{epoch} = \text{epoch} + 1$. Перейти на шаг 3.

Шаг 11. Останов.

В качестве правила $R(w)$ предлагается использовать следующие выражения:

$$R(w) = \alpha w, \quad (1)$$

$$R(w) = w + \alpha w; \quad (2)$$

$$R(w) = w + \alpha \text{rand}, \quad (3)$$

где rand – случайное число в диапазоне $[0, 1]$.

4. Эксперименты и результаты

Для исследования свойств слеящего алгоритма и алгоритма планирования обучающего эксперимента были разработаны процедуры на языке MATLAB, позволяющие использовать данные методы для ускорения сходимости градиентных алгоритмов, реализуемых процедурами модуля Neural Network Toolbox пакета MATLAB 5.2 фирмы Mathworks Inc.

Во всех экспериментах, рассмотренных далее, использовались МНС с одинаковой топологией, одинаковыми функциями активации, начальными весами и порогами соответствующих нейронов.

Для сравнения разработанного слеящего алгоритма и наиболее быстрого градиентного алгоритма Левенберга-Марквардта на основе МНС решались задачи классификации объектов и прогнозирования значений параметров сложных изделий по реальным данным [5-7]. Результаты обучения представлены на рис. 1.

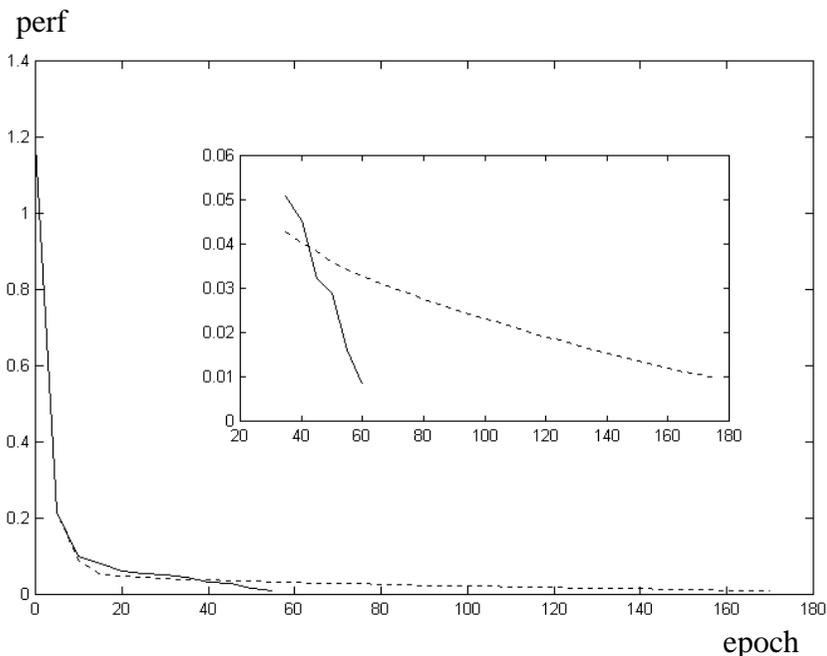


Рис. 1 - Графики убывания ошибки perf при обучении НС классификации (пунктирной линией выделен график ошибки алгоритма Левенберга-Марквардта, сплошной линией – слеящего алгоритма)

Как видно из рис. 1, предложенный алгоритм позволяет существенно сократить время обучения МНС и увеличить скорость сходимости градиентного алгоритма. В разных экспериментах время обучения МНС с использованием следящего алгоритма сокращалось на 20 – 60 %, что является очень хорошим результатом.

Анализируя графики сходимости градиентного и следящего алгоритмов, можно сделать следующие выводы:

1) в начале оба алгоритма обеспечивают одинаковую сходимость, что объясняется двумя причинами: следящий алгоритм включается только после выполнения первых Δt циклов обучения и сходимость, обеспечиваемая градиентным алгоритмом, в начале обучения довольно высока и не требует включения следящего алгоритма;

2) после прохождения начальной фазы обучения градиентный алгоритм в течение некоторого времени обеспечивает более быструю сходимость, чем следящий. Это, по-видимому, объясняется тем, что даже изменения, уменьшающие значения целевой функции, вносимые следящим алгоритмом, могут иногда негативно влиять на общий процесс сходимости, т.е. мешают работе градиентного алгоритма на данном этапе.

3) на завершающей стадии обучения, когда градиентный алгоритм обучения НС начинает блуждать по локальным минимумам целевой функции и обеспечивает очень медленную сходимость, следящий алгоритм обеспечивает существенно более быструю сходимость и превосходит по скорости градиентный алгоритм.

Следует отметить, что применение следящего алгоритма для решения задач классификации легко разделяемых образов может не давать выигрыша в скорости обучения, поскольку из-за высокой скорости сходимости градиентного алгоритма следящий алгоритм в работу не будет вступать.

Эксперименты по обучению МНС решению практических задач на основе следящего алгоритма позволяют рекомендовать правила (1) и (2) для обучения НС классификации, а правило (3) – для обучения аппроксимации. Применение правил (1) и (2) для обучения аппроксимации нецелесообразно. Это объясняется тем, что данные правила способны вносить довольно сильные возмущения в набор весов сети.

Из описания следящего алгоритма следует, что, кроме выбора соответствующего правила $R(w)$, необходимо задавать значения определенных параметров. Для того, чтобы выяснить, какие значения наиболее приемлемы для соответствующих параметров следящего алгоритма проводились эксперименты, результаты которых показывают, что приемлемыми для большинства случаев могут быть следующие значения параметров

следящего алгоритма: $\alpha = 1.03$, $\Delta t = 6-20$. Значение параметра ξ следует принять равным 0.1.

В целом при решении различных задач на основе НС разработанный алгоритм показал себя достаточно хорошо и его можно рекомендовать для внедрения на практике при обучении НС классификации сложно разделимых образов и аппроксимации существенно нелинейных зависимостей.

Список литературы

1. Дубровин В.И., Субботин С.А. Диагностика на основе эвристических алгоритмов в условиях ограниченного объема обучающей выборки // Proceedings of International conference "Soft computing and measurement" SCM-2000, 27-30 June 2000.-Saint-Petersburg: Saint-Petersburg State Electrotechnical University (LETI), 2000.-CD-ROM
2. Dubrovin V., Subbotin S. Choice of neuron transfer functions and research of their influence for learning quality of neural networks // Proceedings of International Conference on Modern Problem of Telecommunications, Computer Science and Engineers Training TCSET'2000, February 14-19, 2000, Lviv-Slavsko, P. 114-115.
3. Дубровин В.И., Субботин С.А. Нейросетевое моделирование и оценка параметров нелинейных регрессий // Нейрокомпьютеры и их применение / Сборник докладов 6-ой Всероссийской конференции, Москва 16-18 февраля 2000.- М.:Издательское предприятие журнала "Радиотехника", 2000.- С. 118-120.
4. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: в 2-х кн. Кн. 1.-М.:Мир, 1986.-349 с.
5. Dubrovin V., Morshchavka S., Piza D., Subbotin S. Plant recognition by genetic algorithm based back-propagation // Proceedings, Remote Sensing 2000: from spectroscopy to remotely sensed spectra. Soil Science Society of America, Bouyocos Conference, Corpus Christi, Texas, October 22-25, 2000.-P. 47- 54
6. Dubrovin V.I., Subbotin S.A., Morshchavka S.V., Piza D.M. The plant recognition on remote sensing results by the feed-forward neural networks // Smart Engeneering Systems Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, ANNIE 2000:the 10-th Anniversary edition / ed. C. H. Dagli et al.-Missouri-Rolla: ASME Press, 2000, vol.10, P. 697-701
7. Dubrovin V., Subbotin S. Model of Magnetic Heads Audio Characteristics // Proceedings of the Sixth International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2001.-Lviv-Slavsko, 2001.-P. 232-233