

УДК 681.513

Н.А. Тесленко

АЛГОРИТМ ОБУЧЕНИЯ АВТОАССОЦИАТИВНОЙ ИСКУССТВЕННОЙ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ

Введение

В настоящее время искусственные нейронные сети получили широкое распространение в задачах интеллектуального анализа данных, благодаря своим универсальным аппроксимирующим свойствам [1, 2]. Наряду с задачами, так или иначе связанными с проблемой аппроксимации (эмуляция, управление, распознавание образов, прогнозирование и т. п.), достаточно часто приходится сталкиваться с проблемами сжатия информации (сокращение размерности входного пространства признаков), постоянно возникающими при обработке больших массивов информации.

Постановка задачи

Для исследования и сжатия массивов многомерных данных, характеризующихся линейной зависимостью признаков исходного многомерного пространства, наиболее часто применяется техника факторного анализа (главных компонент), позволяющая путем выявления доминирующих латентных признаков понизить размерность исходной задачи. Для решения той же проблемы в существенно нелинейных ситуациях в [3] было предложено использовать архитектуры типа многослойного персептрона специального вида с сигмоидальными активационными функциями, обучаемые с помощью процедур обратного распространения ошибок. Основным недостатком этого подхода является низкая скорость обучения, присущая всем многослойным нейронным сетям с нелинейными активационными функциями типа логистической кривой или гиперболического тангенса.

Для решения задачи понижения размерности исходного пространства признаков (сжатия информации) в данной работе предлагается использовать многослойную автоассоциативную искусственную нейронную сеть прямого распространения (bottleneck), построенную на формальных нейронах с полиномиальной функцией активации вида [4]:

$$\psi_j(u_j) = 1,5 \left(u_j(k) - \frac{u_j^3(k)}{3} \right), \quad (1)$$

где u_j — внутренний сигнал активации j -го формального нейрона; $k = 1, 2, 3, \dots$ — текущее дискретное время; $-1 < \psi_j(u_j) < 1$; $-1 < u_j < 1$. Вид функции (1) приведен на рис. 1.

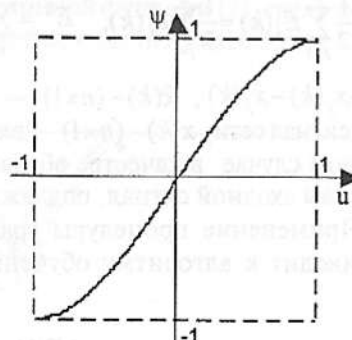


Рис. 1. Активационная функция формального нейрона

Данная активационная функция обладает стандартными свойствами сигмоидальных функций, то есть является монотонной, возрастающей, ограниченной и имеет отличную от нуля производную на всей области определения.

Как было показано в [4–6], алгоритм обучения формального нейрона, построенный на основе активационной функции вида (1), позволяет существенно повысить скорость настройки синаптических весов формального нейрона, схема которого приведена на рис. 2.

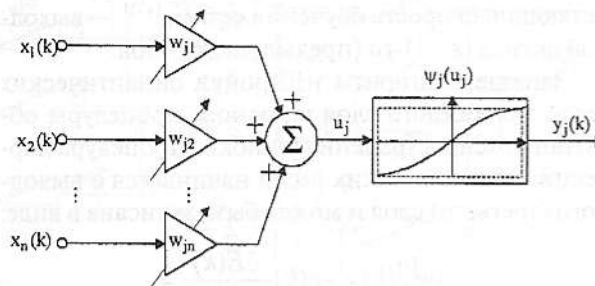


Рис. 2. Схема формального нейрона с полиномиальной функцией активации

Предлагаемая нейронная сеть состоит из трех слоев, образованных нейронами с полиномиальными функциями активации. Число нейронов первого скрытого и выходного слоев совпадает с размерностью входного пространства и принимается равным n . Число нейронов второго скрытого слоя принимается $m < n$ и определяется степенью сжатия информации. Выходной сигнал каждого слоя является входным сигналом для последующего слоя.

Требуется разработать алгоритм обучения данной архитектуры.

Архитектура сети и алгоритм обучения

Процесс обучения нейронной сети сводится к настройке матриц синаптических весов $W^{[s]}$, $s = 1, 2, 3$, путем минимизации принятой целевой функции, в качестве которой используется выражение:

$$E(k) = \frac{1}{2} \sum_{j=1}^n E_j(k) = \frac{1}{2} \sum_{j=1}^n e_j^2(k), \quad E^k = \sum_k E(k),$$

где $e_j(k) = x_j(k) - \hat{x}_j(k)$, $\hat{x}(k) - (n \times 1)$ — выходной векторный сигнал сети, $x(k) - (n \times 1)$ — входной вектор. В данном случае в качестве обучающего используется сам входной сигнал, подлежащий компрессии. Применение процедуры градиентного спуска приводит к алгоритму обучения общего вида:

$$\begin{aligned} w_{ji}^{[s]}(k+1) &= w_{ji}^{[s]}(k) - \eta(k) \frac{\partial E(k)}{\partial w_{ji}^{[s]}(k)} = \\ &= w_{ji}^{[s]}(k) - \eta(k) \frac{\partial E(k)}{\partial u_j^{[s]}(k)} \frac{\partial u_j^{[s]}(k)}{\partial w_{ji}^{[s]}(k)} = \\ &= w_{ji}^{[s]}(k) + \eta(k) \delta_j^{[s]}(k) o_i^{[s-1]}(k), \\ \Delta w_{ji}^{[s]}(k) &= -\eta(k) \frac{\partial E(k)}{\partial w_{ji}^{[s]}(k)}, \end{aligned}$$

где $\delta_j^{[s]}(k) = -\frac{\partial E(k)}{\partial u_j^{[s]}(k)}$ — локальная ошибка j -го ней-

рона s -го слоя; $\eta(k)$ — скалярный параметр, определяющий скорость обучения сети; $o_i^{[s-1]}$ — выходной сигнал $(s-1)$ -го (предыдущего) слоя.

Запишем алгоритм настройки синаптических весов для каждого слоя на основе процедуры обратного распространения ошибки. Процедура коррекции синаптических весов начинается с выходного (третьего) слоя и может быть записана в виде:

$$\begin{aligned} \Delta w_{ji}^{[3]}(k) &= -\eta(k) \frac{\partial E(k)}{\partial w_{ji}^{[3]}(k)} = \\ &= -\eta(k) \frac{\partial E(k)}{\partial u_j^{[3]}(k)} \frac{\partial u_j^{[3]}(k)}{\partial w_{ji}^{[3]}(k)} = \eta(k) \delta_j^{[3]}(k) o_i^{[2]}(k), \end{aligned}$$

при этом локальная ошибка третьего выходного слоя с учетом (1) определяется следующим образом:

$$\begin{aligned} \delta_j^{[3]}(k) &= \frac{\partial E(k)}{\partial u_j^{[3]}(k)} = \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial u_j^{[3]}(k)} = \\ &= \psi_j^{[3]'}(u_j^{[3]}(k)) e(k) = 1,5(1 - u_j^{[3]2}(k)) e(k), \end{aligned}$$

а, следовательно, процедура настройки весов выходного слоя окончательно может быть записана в виде:

$$\Delta w_{ji}^{[3]}(k) = 1,5\eta(k)(1 - u_j^{[3]2}(k)) e_j(k) o_i^{[2]}(k),$$

где $e_j(k) = (e_1(k), e_2(k), \dots, e_n(k))^T - (n \times 1)$ — вектор ошибок обучения в момент времени k .

Алгоритм настройки синаптических весов второго скрытого слоя имеет вид:

$$\Delta w_{ji}^{[2]}(k) = \eta(k) \delta_j^{[2]}(k) o_i^{[1]}(k),$$

а его локальная ошибка определяется соотношениями:

$$\begin{aligned} \delta_j^{[2]}(k) &= \frac{\partial E(k)}{\partial u_j^{[2]}(k)} = \psi_j^{[2]'}(u_j^{[2]}(k)) \frac{\partial E(k)}{\partial o_j^{[2]}(k)} = \\ &= 1,5(1 - u_j^{[2]2}(k)) \frac{\partial E(k)}{\partial o_j^{[2]}(k)}, \end{aligned} \quad (2)$$

$$\frac{\partial E(k)}{\partial o_j^{[2]}(k)} = \sum_{i=1}^n \frac{\partial E(k)}{\partial u_i^{[3]}(k)} \frac{\partial u_i^{[3]}(k)}{\partial o_j^{[2]}(k)} = \sum_{i=1}^n \delta_i^{[3]}(k) w_{ji}^{[3]}(k), \quad (3)$$

$$\delta_j^{[2]}(k) = 1,5(1 - u_j^{[2]2}(k)) \sum_{i=1}^n \delta_i^{[3]}(k) w_{ji}^{[3]}(k). \quad (4)$$

Таким образом, алгоритм настройки синаптических весов второго скрытого слоя можно записать в виде:

$$\Delta w_{ji}^{[2]}(k) = 1,5\eta(k)(1 - u_j^{[2]2}(k)) o_i^{[1]} \sum_{l=1}^n \delta_l^{[3]}(k) w_{li}^{[3]}(k).$$

Синаптические веса первого скрытого слоя настраиваются с помощью процедуры:

$$\Delta w_{ji}^{[1]}(k) = \eta(k) \delta_j^{[1]}(k) o_i^{[0]}(k) = \eta(k) \delta_j^{[1]}(k) x_i(k),$$

а локальная ошибка первого скрытого слоя вычисляется аналогично соотношениям (2-4) и имеет следующий вид:

$$\delta_j^{[1]}(k) = 1,5(1 - u_j^{[1]2}(k)) \sum_{i=1}^m \delta_i^{[2]}(k) w_{ji}^{[2]}(k). \quad (5)$$

Окончательно процедура настройки синаптических весов первого скрытого слоя с учетом (5) описывается выражением:

$$\Delta w_{ji}^{[1]}(k) = 1,5\eta(k)(1 - u_j^{[1]2}(k)) x_i(k) \sum_{l=1}^m \delta_l^{[2]}(k) w_{li}^{[2]}(k).$$

Таким образом, алгоритм обучения данной сети представляет собой процедуру настройки трех матриц синаптических весов, начиная с выходного (третьего скрытого) слоя и имеет вид:

$$\begin{cases} \Delta w_{ji}^{[3]}(k) = 1,5\eta(k)(1 - u_j^{[3]2}(k)) e_j(k) o_i^{[2]}(k), \\ \Delta w_{ji}^{[2]}(k) = 1,5\eta(k)(1 - u_j^{[2]2}(k)) o_i^{[1]} \sum_{l=1}^n \delta_l^{[3]}(k) w_{li}^{[3]}(k), \\ \Delta w_{ji}^{[1]}(k) = 1,5\eta(k)(1 - u_j^{[1]2}(k)) x_i(k) \sum_{l=1}^m \delta_l^{[2]}(k) w_{li}^{[2]}(k). \end{cases} \quad (6)$$

На рис. 3 приведена схема обучения трехслойной автоассоциативной сети для сжатия информации на основе (6).

Численный эксперимент

Рассмотрим задачу компрессии информации на примере данных ирисов Фишера [7] с помощью описанной выше нейронной сети. Рассматриваются три вида цветков ириса: *Setosa*, *Versicolor* и *Virginica*. Всего имеется по 50 экземпляров каждого вида, и для каждого из них измерены четыре величины: длина и ширина чашелистика, длина и ширина лепестка. Размерность сжатых данных равна 2. Для данной нейронной сети в качестве активационных используются функции:

$$\psi_j(u_j) = 1,5u_j(k) - 0,5u_j^3(k), \quad (7)$$

$$\psi_j(u_j) = \sin\left(\frac{\pi u_j(k)}{2}\right), \quad (8)$$

$$\psi_j(u_j) = \tanh(u_j(k)). \quad (9)$$

Алгоритм компрессии информации с активационной функцией (7) позволяет достичь ошибки порядка 4,5 за 8-9 эпох или за среднее время 7с. при использовании комплекса программно-технических средств на базе ПК (Mb: КТ266А, CPU: Diron 800Mhz, RAM: 1Gb). Алгоритм компрессии

информации с активационной функцией (8) достигает порядка 4,5 за 55 эпох или за среднее время 40с. Алгоритм компрессии информации с активационной функцией (9) при равных начальных условиях достигает ошибки только порядка 13.

На рис. 4 представлены данные ирисов Фишера (3 класса), сжатые алгоритмом с использованием активационной функции (7), среди которых при сжатии линейно отделим класс *Setosa*.

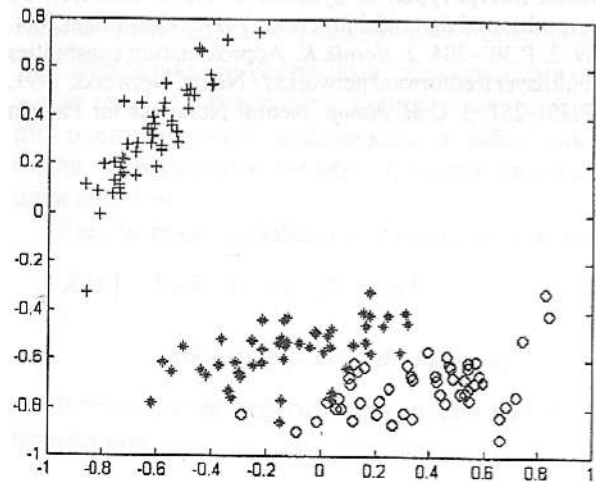


Рис. 4. Графическое изображение сжатых данных ирисов Фишера

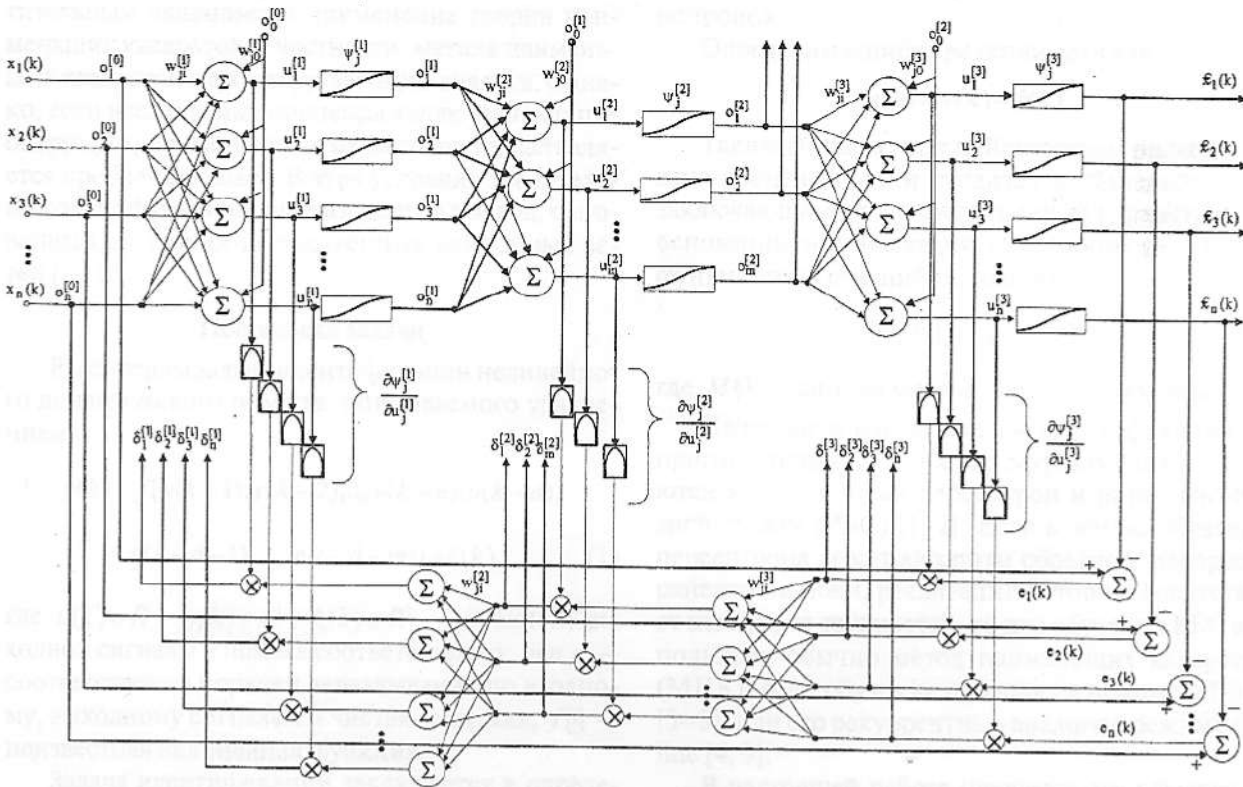


Рис. 3. Схема обучения автоассоциативной искусственной нейронной сети прямого распространения

Заклучение

Для задачи сжатия информации предложен алгоритм обучения нейронной сети типа трехслойного персептрона архитектуры bottle-neck, который позволяет упростить численную реализацию процедуры обратного распространения ошибки, что, в свою очередь, позволяет сократить время обучения. Данная нейронная сеть с предложенным алгоритмом обучения может использоваться для сжатия и восстановления данных в реальном времени.

Список литературы: 1. *Cybenko G.* Approximation by superposition of sigmoidal function // *Math. Contr. Sign. Syst.* 1989. 2. P.303-304. 2. *Hornik K.* Approximation capabilities of multilayer feedforward networks // *Neural Networks*. 1991. 4. P.251-257. 3. *C.M. Bishop.* *Neural Networks for Pattern*

Recognition. Clarendon Press — Oxford. 1995. 4. *Бодянский Е.В., Полушкина (Тесленко) Н.А.* Алгоритм обучения формального нейрона с полиномиальной активационной функцией // *Радиоэлектроника и информатика*. 2003. №4. С.45-48. 5. *Полушкина (Тесленко) Н.А., Штанько В.И.* Алгоритм обучения нейронной сети с кубическими функциями активации // *Междунар. науч. конф. «Теория и техника передачи, приема и обработки информации»: Тез. докл. Харьков: ХНУРЭ. 2003. С.397-398.* 6. *Бодянский Е.В., Слипченко А.Н., Тесленко Н.А.* Прогнозирующая нейронная метасеть с нетрадиционными функциями активации // *Автомобильный транспорт: Сб. науч. трудов. Харьков: ХНАДУ. 2003. Вып. 13. С.273-275.* 7. *Нейронные сети. STATISTICA Neural Networks.* Пер. с англ. М.: Горячая линия — Телеком. 2001. 182 с.

Поступила в редколлегию 05.11.2004