УДК 004.93

И. В. Новосельцев, Н. Г. Аксак

ПРИМЕНЕНИЕ МНОГОПРОЦЕССОРНЫХ СИСТЕМ ДЛЯ РЕАЛИЗАЦИИ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА

В данной работе предложена модель реализации МП на многопроцессорной системе при ограниченном количестве процессоров. В качестве топологии вычислительной системы предложена организация процессоров в виде звезды. Данный подход позволяет значительно снизить время обучения НС, и дает возможность получения максимального ускорения, ограниченного числом имеющихся процессоров.

ВВЕДЕНИЕ

Сегодня по-прежнему остается актуальной задача распознавания изображения. Применение нейронных сетей (НС) является перспективным для решения задач в этой области [1]. Однако сложность подбора оптимальной архитектуры сети для решения конкретной задачи, а также большие вычислительные затраты, требуемые для обучения нейросетей, сдерживает их массовое применение.

Время обучения нейронных сетей, особенно при использовании стандартных персональных компьютеров и больших объемах исходных данных, может быть очень велико. Проблему высокой вычислительной сложности обучения нейросетей чаще всего решают, используя более мощный компьютер или специализированный аппаратный нейроускоритель. Однако такой путь доступен далеко не всем.

Снизить вычислительную сложность нейросетевых алгоритмов можно путем распараллеливания однотипных операций, которые выполняются в процессе функционирования нейронной сети. Подобные подходы позволяют обучать нейронные сети с меньшими затратами времени и дают возможность получения максимального ускорения, ограничиваемого числом имеющихся процессоров.

В данной работе рассмотрен метод распараллеливания при ограниченном количестве процессоров [2]. В качестве топологии вычислительной системы предложена организация процессоров в виде звезды. Управляющий процессор в подобной топологии может использоваться для загрузки вычислительных процессоров исходными данными и для приема результатов выполненных вычислений.

1 АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ

Для решения задачи распознавания образов применяется многослойный персептрон (МП) с архитектурой $N_i - N_j - N_k$, где N_i – количество нейронов во входном слое, N_j – количество нейронов в скрытом слое, N_k – количество нейронов в выходном слое (рис. 1) [3–5].

Обучающий пример представляется парой $\{X = [x_1, x_2, ..., x_i], D = [d_1, d_2, ..., d_k]\}$, где X – входной массив, D – вектор цели.

Функционирование нейрона определяется следующим соотношением:

$$y_j^{(1)}(t) = \varphi\left(\sum_{l=1}^{i} w_{lj}^{(1)}(t)x_l(t) + b_j^{(1)}\right) \quad \text{для скрытого слоя,} \qquad (1)$$

$$y_k^{(2)}(t) = \varphi\left(\sum_{l=1}^{j} w_{lk}^{(2)}(t)y_l^{(1)}(t) + b_k^{(2)}\right) \quad \text{для выходного слоя,} \qquad (2)$$

где $w_{ij}^{(1)}$ – синаптический вес связи i-го входа с j-м нейроном, $b_j^{(1)}$ – порог j-го нейрона скрытого слоя, $w_{jk}^{(2)}$ – синаптический вес связи j-го выхода нейрона скрытого слоя с k-м нейроном выходного слоя, $b_k^{(2)}$ – порог k-го нейрона выходного слоя, $\varphi(\bullet)$ – функция активации нейрона.

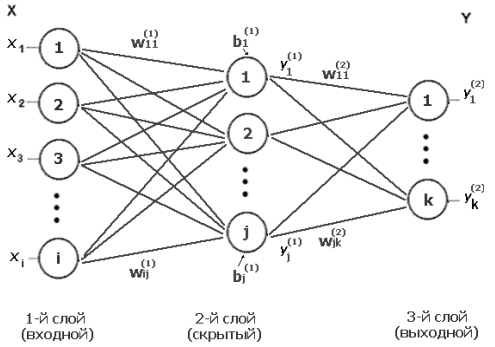

Рисунок 1 – Архитектура многослойного персептрона

В качестве функции активации для всех слоев выбрана сигмоидальная функция

$$\varphi(y^{(L)}(t)) = \frac{1}{1+\exp(-ay^{(L)}(t))}, \qquad (3)$$

где a – параметр наклона сигмоидальной функции ($a > 0$).

Обучение сводится к минимизации среднеквадратической ошибки. Подстройка весов $w_{ij}^{(1)}$ и $w_{jk}^{(2)}$ проводится после подачи каждого примера (последовательный режим обучения). Настройка весов слоя L осуществляется в соответствии

$$w_{ij}^{(L)}(t+1) = \alpha w_{ij}^{(L)}(t) + \eta\delta_j^{(L)}(t)y_j^{(L)}(t), \qquad (4)$$

где η – параметр, отвечающий за скорость обучения, α – постоянная момента ($\alpha = 1$), $\delta_j^{(L)}$ – локальный градиент j-го нейрона слоя L. Для выходного слоя локальный градиент вычисляется следующим образом:

$$\delta_k^{(2)}(t) = e_k(t), \qquad (5)$$

где $e_k(t)$ – ошибка k-го нейрона выходного слоя.

А для скрытого слоя:

$$\delta_j^{(1)}(t) = \sum_{l=1}^{k}\delta_l^{(2)}(t)w_{jl}^{(2)}(t)y_j^{(1)}(t). \qquad (6)$$

На однопроцессорной системе данные вычисления выполняются последовательно, т. е. при формировании выходного вектора скрытого слоя $Y^{(1)}$ последовательно вычисляются $y_1^{(1)}$, $y_2^{(1)}$, …,$y_j^{(1)}$, затем аналогично вычисляются значения выходного слоя.

2 РЕАЛИЗАЦИЯ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА НА МНОГОПРОЦЕССОРНОЙ СИСТЕМЕ

Реализация МП (рис. 1) на многопроцессорной системе может быть представлена схемой, приведенной на рис. 2.

Соотношение (1) можно представить в матричном виде как умножение матрицы весовых коэффициентов $W^{(1)}$ скрытого слоя на вектор входа X:

$$Y^{(1)} = \varphi(W^{(1)T}X + B^{(1)}) = \phi\left(\begin{bmatrix} W_1^{(1)} \\ W_2^{(1)} \\ \dots \\ W_j^{(1)} \end{bmatrix} \times X + B^{(1)}\right) =$$

$$= \varphi\left(\begin{bmatrix} w_{11}^{(1)} w_{21}^{(1)} \dots w_{i1}^{(1)} \\ w_{12}^{(1)} w_{22}^{(1)} \dots w_{i2}^{(1)} \\ \dots \\ w_{1j}^{(1)} w_{2j}^{(1)} \dots w_{ij}^{(1)} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_i \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \dots \\ b_j^{(1)} \end{bmatrix}\right). \qquad (7)$$

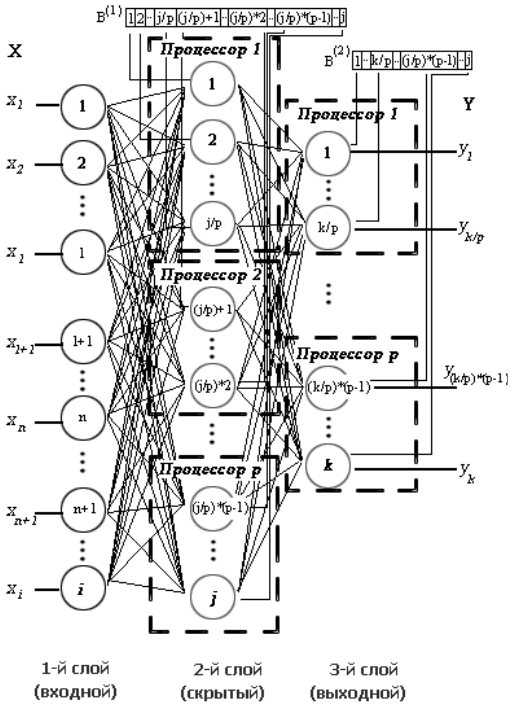

Рисунок 2 – Схема реализации многослойного персептрона

Соотношение (2) можно представить в виде умножения матрицы весовых коэффициентов $W^{(2)}$ выходного слоя на вектор $Y^{(1)}$ (выход скрытого слоя), полученный в результате операции (1):

$$Y^{(2)} = \varphi(W^{(2)T}Y^{(1)} + B^{(2)}) = \phi\left(\begin{bmatrix} W_1^{(2)} \\ W_2^{(2)} \\ \dots \\ W_j^{(2)} \end{bmatrix} \times Y^{(1)} + B^{(2)}\right) =$$

$$= \varphi\left(\begin{bmatrix} w_{11}^{(2)} w_{22}^{(2)} \dots w_{j1}^{(2)} \\ w_{12}^{(2)} w_{22}^{(2)} \dots w_{j2}^{(2)} \\ \dots \\ w_{1k}^{(2)} w_{2k}^{(2)} \dots w_{jk}^{(2)} \end{bmatrix} \times \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ \dots \\ y_k^{(1)} \end{bmatrix} + \begin{bmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \dots \\ b_k^{(2)} \end{bmatrix}\right). \quad (8)$$

Соотношение (6) – умножение матрицы весовых коэффициентов $W^{(2)}$ скрытого слоя на вектор локальных градиентов выходного слоя:

$$\delta_j^{(1)} = \begin{bmatrix} \delta_1^{(2)} w_{11}^{(2)} \delta_2^{(2)} w_{12}^{(2)} \dots \delta_k^{(2)} w_{1k}^{(2)} \\ \delta_1^{(2)} w_{21}^{(2)} \delta_2^{(2)} w_{22}^{(2)} \dots \delta_k^{(2)} w_{2k}^{(2)} \\ \dots \\ \delta_1^{(2)} w_{j1}^{(2)} \delta_2^{(2)} w_{j2}^{(2)} \dots \delta_k^{(2)} w_{jk}^{(2)} \end{bmatrix} \times \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ \dots \\ y_j^{(1)} \end{bmatrix}. \quad (9)$$

Тем самым, получение результирующего вектора $Y^{(2)}$ на каждом шаге обучения предполагает повторение $n = (2j + k)$ однотипных операций по умножению строк матриц на векторы. Выполнение каждой такой операции включает поэлементное умножение элементов строки матрицы на векторы и последующее суммирование полученных произведений. Общее количество необходимых скалярных операций необходимых для обучения НС оценивается величиной:

$$T_1 = 2n^2 M, \quad (10)$$

где M – количество эпох, n – количество однотипных операций.

Если, например, размерность изображения 100*100, обучающая выборка состоит из 1000 примеров и количество эпох обучения равно 500, тогда $T_1 = 2(2*100*100+4)^2*500*1000 = 400160016000000$.

При использовании количества процессоров $p \le n$ параллельная вычислительная схема умножения матрицы на вектор может быть конкретизирована следующим образом:

– на каждый из имеющихся управляющих процессоров пересылается вектор X и $s = n/p$ строк матрицы;

– выполнение операции умножения строк матрица на вектор выполняется при помощи последовательного алгоритма.

3 ВЫБОР ТОПОЛОГИИ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

В соответствии с характером выполняемых межпроцессорных взаимодействий, для предложенной вычислительной схемы в качестве топологии выбрана организация процессоров в виде звезды. На рис. 3 представлен фрагмент распараллеливания соотношения (7).

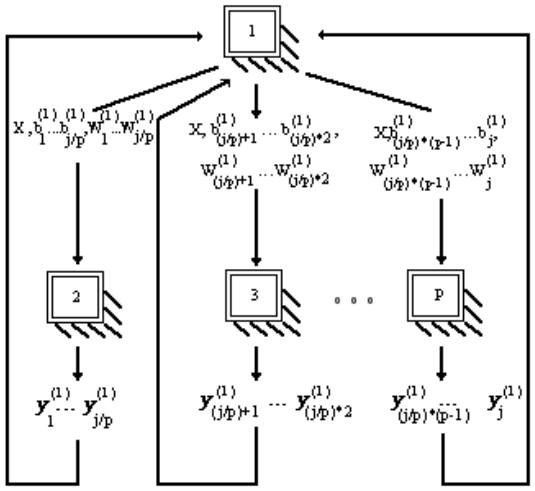

Рисунок 3 – Топология вычислительной системы

Управляющий процессор формирует данные для обработки вычислительными процессорами (рис. 3):

– посылает входной вектор X всем процессорам;

– распределяет матрицу весовых коэффициентов $W^{(1)}$ и вектор смещений $B^{(1)}$ между процессорами, а именно: строки $W_1^{(1)}, W_2^{(1)}, \dots, W_{j/p}^{(1)}$, $b_1^{(1)}, b_2^{(1)}, \dots, b_{j/p}^{(1)}$ – первому процессору, строки $W_{(j/p)+1}^{(1)}, \dots, W_{(j/p)*2}^{(1)}$, $b_{(j/p)+1}^{(1)}, \dots, b_{(j/p)*2}^{(1)}$ – второму процессору, …, строки $W_{(j/p)*(p-1)}^{(1)}, \dots, W_j^{(1)}$, $b_{(j/p)*(p-1)}^{(1)}, \dots, b_j^{(1)}$ – p-му процессору.

После выполнения операций, управляющий процессор осуществляет прием результатов выполнения вычислений. Таким образом формируется выход первого слоя НС – $Y^{(1)}$. Аналогично производится вычисление соотношений (8) и (9).

ЗАКЛЮЧЕНИЕ

С целью снижения времени обучения НС в работе решена актуальная задача распределения НС по процессорам.

Научная новизна работы заключается в том, что впервые предложена новая модель реализации МП на многопроцессорной системе, которая дает возможность получения максимального ускорения, ограниченного числом имеющихся процессоров, за счет распределения однотипных операций умножения, осуществляющихся в процессе функционирования НС.

Практическая ценность работы состоит в том, что проведенные эксперименты, реализующие предложенную модель, показали высокую эффективность. Данная модель реализована с помощью технологии параллельного программирования MPI.

ПЕРЕЧЕНЬ ССЫЛОК

1. *Richard O. Duda, Peter E. Hart, David G. Stork.* Pattern classification: Wiley-Interscience; 2nd edition, 2000. – 680 p.
2. *Гергель В. П., Стронгин Р. Г.* Основы параллельных вычислений для многопроцессорных вычислительных систем: Учебное пособие Нижегородский госуниверситета. – Нижний Новгород, 2003. – 184 с.
3. *Саймон Хайкин.* Нейронные сети: полный курс, 2-е издание. М.: Издательский дом «Вильямс», 2006. – 1104 с.
4. *Миркес Е. М.* Учебное пособие по курсу НЕЙРОИНФОРМАТИКА. – Красноярск, 2002. – 347 с.
5. *Ben Krose, Patrick van der Smagt.* An Introduction to Neural Networks: University of Amsterdam, 1996. – 154 p.

Надійшла 2.03.2007

In this work the MP model realization on multiprocessing system is offered. The processors organization as a star topology is offered. The given approach allows considerably to lower time of training neural network and enables receptions of the maximal acceleration limited to number of available processors.

В даної роботі запропонована модель реалізації багатошарового персептрону на багатопроцесорної системі при обмеженій кількості процесорів. В якості топології обчислювальної системи запропонована організація процесорів у вигляді зірки. Даний підхід дозволяє значно знизити час навчання нейронної мережі та дає можливість одержання максимального прискорення, що обмежений числом наявних процесорів.

УДК 519.7:004.93

С. О. Субботін, А. О. Олійник

СТРУКТУРНИЙ СИНТЕЗ НЕЙРОМОДЕЛЕЙ НА ОСНОВІ ПОЛІМОДАЛЬНОГО ЕВОЛЮЦІЙНОГО ПОШУКУ

Досліджено проблему структурного синтезу нейромережних моделей. Розроблено метод полімодального еволюційного пошуку з кластеризацією хромосом, що дозволяє одержати різні структури нейромереж, підвищуючи ймовірність знаходження моделі, адекватної розв'язуваній задачі. Проведено експерименти по синтезі моделей залежності стану здоров'я населення від забруднення навколишнього середовища.

ВСТУП

Етап вибору оптимальної архітектури нейромоделі є одним з найважливіших завдань при синтезі нейромереж [1], оскільки на цьому етапі формується топологія зв'язків та обираються функції активації нейронів, які надалі визначають принцип функціонування мережі й її ефективність для вирішення досліджуваної задачі. Так, нейромережі, що мають невелику кількість нейронів і лінійні функції активації, як правило, через свої обмежені апроксимаційні здатності не дозволяють вирішувати реальні практичні задачі. У той же час вибір надлишкової кількості нейронів у мережі призводить до проблеми перенавчання й втрати апроксимаційних властивостей нейромоделі [2, 3].

У наш час структурний синтез нейромережних моделей відбувається за участю експерта в предметній області розв'язуваної задачі, що призводить до значного впливу рівня досвіду й знань експерта на ефективність побудованої нейромоделі [4].

Існуючі методи автоматичного пошуку оптимальної структури нейромережних моделей використовують жадібну стратегію пошуку [5]. Так конструктивні (constructive) методи починають пошук з мінімально можливої архітектури мережі (нейромережа із мінімальною кількістю шарів, нейронів і міжнейронних зв'язків) і по-слідовно на кожній ітерації додають нові шари, нейрони й міжнейронні зв'язки. При використанні деструктивних методів (destructive) на початковій ітерації оцінюється ефективність нейромоделі, що містить максимально припустиму кількість шарів, ней-