В табл. 3 приводятся не масштабированные суммы устойчивых значений, полученные на тестовой выборки данных.

Выводы

1. В нейронной сети прямого распространения возможна реализация точечных отображений входных значений на активационных функциях её нейронов.
2. Модель сети, основанной на точечных отображениях, позволяет рассчитывать коэффициенты наклона α и смещения β для нейронов сети и число нейронов в различных слоях сети. Веса синапсов являются значениями весовой функции $W(\zeta)$, рассчитанными для определённых значений её аргумента. Вид $W(\zeta)$ определяется желаемыми характеристиками низкочастотной фильтрации входного сигнала и задает форму окна фильтрации в частотной области.
3. Наличие точечных отображений позволяет использовать сеть в качестве классификатора, определяющего принадлежность входного сигнала к одному из заранее известных классов.
4. Нейронная сеть, рассчитанная для определения принадлежности объектов к одному заранее известных классов, позволила в экспериментальной проверке достичь точности классификации свыше 96 %.

СПИСОК ЛИТЕРАТУРЫ

1. Хайкин С. Нейронные сети: полный курс. Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.
2. Лоскутов А.Ю., Михайлов А.С. Введение в синергетику. – М.: Наука, 1990. – 270 с.
3. Миронов С. Ирисы Фишера [Электронный ресурс]. – режим доступа: http://www.delphikingdom.com/asp/viewitem.asp?catalogid=400. – 21.10.2008.

Поступила 27.10.2008 г.

УДК 681.3.06:681.323

ОТОБРАЖЕНИЕ НЕЙРОСЕТЕВЫХ АЛГОРИТМОВ АНАЛИЗА ИЗОБРАЖЕНИЙ НА РЕГУЛЯРНЫЕ СТРУКТУРЫ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

М.С. Тарков

Институт физики полупроводников СО РАН, г. Новосибирск
E-mail: tarkov@isp.nsc.ru

Предложены алгоритмы отображения матрицы весов первого (скрытого) слоя нейронной сети на распределенные вычислительные системы с тороидальной структурой при параллельном решении задач анализа изображений. Показано, что выбор способа отображения зависит от отношения числа нейронов в слое к числу весовых коэффициентов нейрона (пикселей изображения). В частности, для сети Хопфилда распределение по процессорам строк матрицы весов дает более высокую эффективность распараллеливания вычислений, чем распределение столбцов.

Ключевые слова:
Анализ изображений, нейронные сети, распределенные вычислительные системы, отображение алгоритмов.

Введение

В настоящее время наблюдается устойчивая тенденция роста объема перерабатываемой визуальной информации в современных информационных системах. Вместе с этим возрастают и требования к производительности таких систем. Одним из наиболее перспективных направлений в решении этой проблемы является использование нейрокомпьютерных технологий обработки изображений [1, 2]. В качестве преимущества такого рода технологий можно выделить возможность в рамках единого методико-алгоритмического базиса решать самые различные задачи обработки изображений. Кроме того, следует отметить, что большинство задач обработки изображений допускают естественный параллелизм вычислений в реализации соответствующих вычислительных процедур в силу специфики представления самого цифрового изображения (как двумерного или многомерного массива чисел).

Элемент нейронной сети (нейрон) реализует преобразование вида $y=f((w,x))$, где (w,x) – скалярное произведение вектора входных сигналов x на вектор весовых коэффициентов нейрона, f – нелинейная функция. Каждый весовой коэффициент соответствует одному входу (синапсу) нейрона. Множество нейронов, обрабатывающих один и тот же вектор входных сигналов x, образует слой нейронов. Функционирование слоя описывается формулой

$$Y = f(Wx), \quad (1)$$

где W – матрица, строки которой являются весовыми векторами нейронов слоя, Y – вектор выходных сигналов слоя. Основную часть вычислений в нейронной сети [1] образуют операции умножения матрицы весовых коэффициентов слоя нейронов на вектор, элементы которого однозначно соответствуют пикселям обрабатываемого изображения, что приводит к большому объему вычислений и необходимости использования высокопараллельных вычислительных систем [3, 4]. Максимально достижимая степень параллелизма вычислений равна числу пикселей изображения.

Распределенная вычислительная система (ВС) представляет собой совокупность элементарных машин (ЭМ), соединенных между собой сетью линий связи, управляемой из этих машин. Структура распределенной ВС описывается графом, вершины которого соответствуют ЭМ, а ребра – межмашинным соединениям. В современных суперкомпьютерах с распределенной памятью в качестве графа межмашинных соединений наиболее часто используется трехмерный тор [3, 4] (E_3 – граф). Класс E_k – графов вычислительных систем составляют k-мерные евклидовы решетки с замкнутыми границами (тороидальные решетки). Группа автоморфизмов E_k такой структуры есть прямое произведение циклических подгрупп C_{p_k}: $E_k=\otimes_{i=1}^{k} C_{p_i}$, где p_i – порядок подгруппы C_{p_i}, $\otimes$ – символ прямого произведения. Размер решетки определяется набором образующих p_i, i=1,...,k, по каждому из k измерений. В структурах этого класса каждый узел соединен с $2k$ другими узлами при значениях образующих p_i>2.

Предварительным этапом обработки изображения является фильтрация [5], которая зачастую описывается сверткой изображения с множеством весовых коэффициентов фильтра. Фильтрация обычно предшествует другим преобразованиям, например, преобразованию вида (1). В этом случае результат фильтрации (изображение) представлен в преобразовании (1) вектором x. Для вычисления свертки обычно используется квадратное окно размером $(2M+1)\times(2M+1)$, $M<<\min(N_1,N_2)$, где N_1 и N_2 – размеры изображения. Вычисление значения в точке изображения связано с обработкой малой окрестности этой точки, т.е. алгоритмы фильтрации являются локальными. Из локальности свертки следует, что: 1) соседство процессоров в системе должно соответствовать соседству пикселей изображения; 2) отображение фрагментов данных, обрабатываемых алгоритмами свертки, на процессоры должно сохранять соседство фрагментов; 3) в качестве графа системы параллельных процессов, реализующих фильтрацию изображения, целесообразно использовать евклидову решетку, которая естественным образом вкладывается в тороидальную структуру ВС.

Далее всюду полагаем, что компоненты изображения равномерно распределены по машинам системы так, что соседние пиксели всегда располагаются либо в одной ЭМ, либо в соседних машинах решетки.

Способ организации межмашинных обменов при параллельном выполнении операции (1) будет определяться распределением по машинам коэффициентов матрицы весов W. В настоящее время разработано много методов отображения нейронных сетей на параллельные вычислительные системы [6, 7], но эти методы не учитывают специфики вышеописанного геометрического параллелизма алгоритмов предварительной обработки изображений. Решение этой проблемы является целью данной работы, в которой рассматривается два способа вложения слоя нейронов в структуру распределенной ВС: 1) размещение строк матрицы весовых коэффициентов по машинам системы (параллелизм нейронов); 2) размещение столбцов матрицы весов по машинам системы (параллелизм синапсов).

1. Вложение слоя нейронной сети путем размещении строк матрицы весов по машинам

Рассмотрим организацию межмашинных обменов при распределении строк матрицы весов W по машинам. Так как каждая строка матрицы весов соответствует одному нейрону сети, то распределение строк матрицы весов описывает размещение нейронов сети по машинам. Чтобы выполнить вычисления для всех нейронов по формуле (1), необходимо собрать в каждой машине компоненты изображения x, т. е. выполнить трансляционно-циклический обмен («все со всеми») компонентами вектора x. В результате умножение строк матрицы весов на этот вектор можно выполнить во всех машинах параллельно (количество одновременно выполняемых умножений пар векторов равно числу машин).

Трансляционно-циклический обмен в k?мерном торе сводится к выполнению трансляционно-циклических обменов в кольцах тора, т. е. структурах, описываемых циклическими подгруппами. В каждом кольце обмены выполняются следующим образом. Каждая машина M_j, j=1,...,p_i передает свой массив пикселей машине $M_{(j-1)\bmod p_i}$, j=1,...,p_i, а затем получает массив от машины $M_{(j+1)\bmod p_i}$, j=1,...,p_i. Предполагается, что линии связи кольца могут работать одновременно. Описанные действия продолжаются до тех пор, пока каждая машина кольца не получит всех пикселей, распределенных по его машинам. Обмены выполняются параллельно для всех колец i-измерения и последовательно по всем i=1,2,...,k измерениям. Для двумерного тора (рис. 1) обмены, например, могут быть выполнены параллельно в горизонтальных кольцах, а затем параллельно во всех вертикальных кольцах.

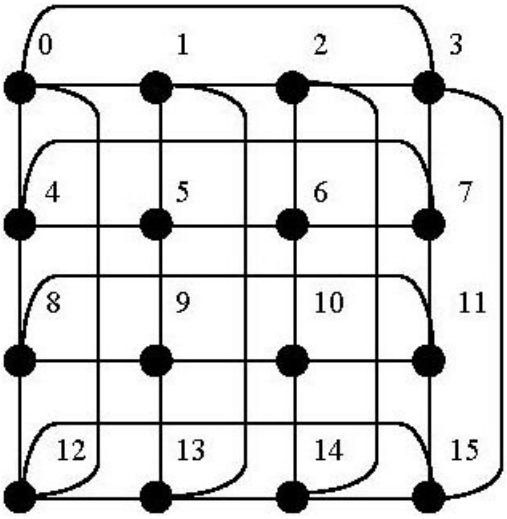

Рис. 1. *Пример 2-мерного тора*

Пусть n – число пикселей изображения, m – число нейронов в слое, p – число машин в системе, t_o – время выполнения одной арифметической операции, t_w – время передачи одного элемента данных, p_i – порядок i-й образующей тора. Тогда по завершении l-го шага обмена, $l \in 1,2,...k$, каждая машина содержит $\frac{n}{p}\prod_{i=1}^{l} p_i$ элементов данных, и соответственно после k шагов – n элементов, т. к. $\prod_{i=1}^{k} p_i = p$.

При этом время выполнения l шагов обмена равно

$$T_e(l) = \frac{n}{p}\left[(p_1 - 1) + \sum_{i=2}^{l}(p_i - 1)\prod_{j=1}^{i-1} p_j\right]t_w. \qquad (2)$$

При $l=k$, преобразуя формулу (2), получаем

Утверждение 1. Время $T_{ex}=T_e(k)$ трансляционно-циклического обмена данными не зависит от размерности тора k и равно

$$T_{ex} = n\left(1 - \frac{1}{p}\right)t_w. \qquad (3)$$

С учетом того, что число пикселей изображения $n>>1$, получаем время последовательного выполнения операции умножения Wx в слое

$$T_{seq} \approx 2mnt_o. \qquad (4)$$

Пусть число нейронов $m=kp$ и число пикселей $n=k_n p$, где $k \geq 1$ и $k_n \geq 1$ – целые числа. При равномерном распределении нейронов по машинам для тора из p машин, получаем время параллельной реализации вычислений в слое (в силу $n>>1$ временем вычисления нелинейной функции f пренебрегаем)

$$T_r = \frac{T_{seq}}{p} + T_{ex} = \frac{2mn}{p}t_o + n\left(1 - \frac{1}{p}\right)t_w. \qquad (5)$$

Из (4) и (5) следует

Утверждение 2. При распределении строк матрицы W по процессорам (параллелизм нейронов) коэффициент ускорения не зависит от числа пикселей изображения и равен

$$S_r = \frac{T_{seq}}{T_r} = p\frac{1}{1 + \frac{(p-1)t_w}{2m\ t_o}}. \qquad (6)$$

2. Вложение слоя нейронной сети при размещении столбцов матрицы весов по машинам

При размещении столбцов матрицы весов W по машинам системы параллельное вычисление произведения Wx можно организовать следующим образом:

1. Параллельно выполнить поэлементное умножение коэффициентов матрицы W на соответствующие компоненты вектора x и для каждого нейрона выполнить суммирование получившихся произведений. При $n=2^d=kp$ в p машинах системы параллельно вычисляются частичные суммы.
2. Для вычисления полных сумм для каждого нейрона необходимо произвести обмены, используя двоичное дерево межмашинных соединений, вложенное в граф вычислительной системы. Количество вычисляемых таким образом сумм равно числу нейронов m, которое может быть произвольным и, в частности, кратным числу машин.

Чтобы максимально загрузить машины системы, необходимо обеспечить одновременность выполнения максимального числа операций суммирования. Этому требованию удовлетворяет схема суммирования, называемая «бабочкой», которая позволяет на каждом этапе одновременно вычислять несколько сумм и, если слагаемых достаточно, то число одновременно выполняемых операций сложения равно числу машин системы. На рис. 2 представлен пример бабочки при $p=8$. Здесь x_i, $i=0,1,...,7$ – массивы частичных сумм, вычисленных на шаге 1.

Так как не все процессы бабочки работают одновременно, то следует объединить операции тех процессов, которые не могут выполняться параллельно. На рис. 2 объединяемые процессы (операции) лежат на одной вертикальной линии. В результате слияния процессов бабочки образуется гиперкуб (рис. 3). Здесь числа в скобках показывают номер шага взаимодействий между вершинами гиперкуба.

Двунаправленные стрелки на рис. 3 показывают, что линии связи между вершинами гиперкуба являются дуплексными, т. е. позволяют выполнять передачу сообщений в обоих направлениях. Благодаря этому свойству, можно на каждом этапе «бабочки» (на рис. 2 – этапы (1)–(3)) выполнять независимое суммирование компонент для двух разных массивов.

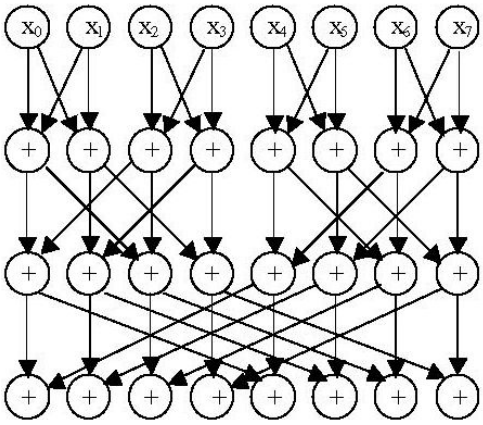

Рис. 2. *Бабочка при d=3*

Полученный слиянием процессов бабочки гиперкуб может быть вложен в тор [8, 9]. При этом неизбежны растяжения ребер гиперкуба. Показано [8], что d-мерный гиперкуб может быть вложен в тор $E_k(2^{d1},...,2^{dk})$, где $\sum_{i=1}^{k} d_i=d$, со средним растяжением

$$D=\frac{\left(\sum_{i=1}^{k}3\cdot 2^{d_i-2}\right)-k}{d}\geq 1.$$

Время перемножения соответствующих компонент вектора весов и изображения равно

$$T_{mult}=\frac{mn}{p}t_o.$$

Время суммирования получившихся произведений

$$T_{add}=m\left(\frac{n}{p}-1\right)t_o.$$

Тогда время вычисления всех частных сумм в работающих параллельно машинах равно

$$T_{ps}=T_{mult}+T_{add}=\frac{mn}{p}t_o+m\left(\frac{n}{p}-1\right)t_o=$$
$$=m\left(\frac{2n}{p}-1\right)t_o. \qquad (7)$$

Далее для каждого из m нейронов вычисляются полные суммы на вложенном в тор гиперкубе из p процессоров за $\log_2 p$ этапов. На каждом из этих этапов могут выполняться операции суммирования не более чем для двух нейронов. Поскольку при переходе от этапа к этапу количество слагаемых уменьшается вдвое и минимально возможное число сумм, вычисляемых на каждом этапе, равно 1, то время вычисления всех полных сумм для m нейронов равно

$$T_{cs}=(Dt_w+t_o)\sum_{i=1}^{\log_2 p}\max\left(1,\frac{m}{2^i}\right). \qquad (8)$$

Полное время параллельной реализации

$$T_c=T_{ps}+T_{cs}=$$
$$=m\left(\frac{2n}{p}-1\right)t_o+(Dt_w+t_o)\sum_{i=1}^{\log_2 p}\max\left(1,\frac{m}{2^i}\right). \qquad (9)$$

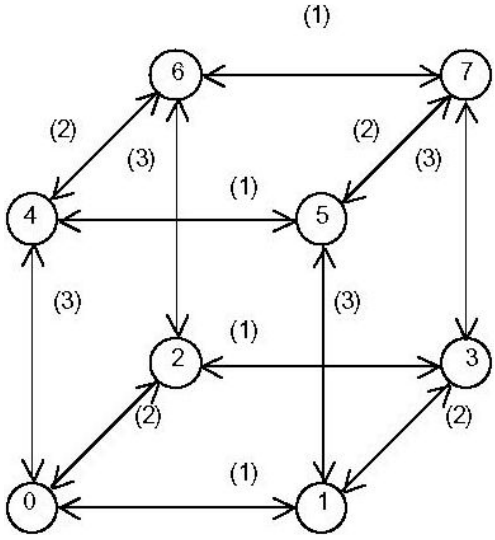

Рис. 3. *Гиперкуб, полученный слиянием процессов бабочки*

При $m=kp$, где $k\geq 1$ – целое число, и $n>>1$ с учетом (4) из (7)–(9) получаем

Утверждение 3. При распределении столбцов матрицы W по процессорам (параллелизм синапсов) коэффициент ускорения S_c не зависит от числа m нейронов в слое и равен

$$S_c=\frac{T_{seq}}{T_c}=p\frac{1}{1+\frac{(p-1)Dt_w}{2nt_o}}. \qquad (10)$$

Из (6) и (10) следует

Утверждение 4. Если $m>n/D$, то $S_r>S_c$, иначе $S_r\leq S_c$, то есть, если число нейронов больше, чем отношение числа синапсов нейрона (пикселей изображения) к среднему растяжению D ребер гиперкуба на торе, то распределение по машинам строк матрицы весов W (параллелизм нейронов) является более эффективным, чем распределение ее столбцов (параллелизм синапсов), и наоборот.

Для получения численных значений ускорения используем параметры суперкомпьютера Cray T3E [2]: производительность процессора (1200 мегафлопс) и пропускную способность канала связи (480 Мб/с). Предположим, что размер элемента данных равен 4 байтам. Тогда

$$t_o=\frac{1}{1,2\cdot 10^9}\approx 0,83\cdot 10^{-9}\text{ с и }t_w=\frac{4}{480\cdot 10^6}\approx 8,3\cdot 10^{-9}\text{ с.}$$

Полагая p=1024, получаем по вышеприведенным формулам коэффициенты ускорения S_r и S_c, приведенные в таблице.

Таблица. *Пример зависимости коэффициента ускорения от числа нейронов*

m	1024	2048	4096	8192	16384	32768	65536
S_r	171	293	455	630	780	885	949
S_c	753	753	753	753	753	753	753

Из таблицы следует, что при большом числе нейронов в слое (m=16384,32768,65536) выгоднее производить распараллеливание по нейронам, а при значениях $m \le 8192$ целесообразно производить распараллеливание по синапсам.

3. Вложение сети Хопфилда

Нейронная сеть Хопфилда представляет собой однослойную сеть с обратной связью. Ее функционирование описывается рекуррентной формулой

$$x^{k+1} = f(Wx^k).$$

Соответствующая матрица весовых коэффициентов W в данном случае является квадратной, т. е. число нейронов m равно числу синапсов (числу пикселей изображения) n. С учетом этого из (6) получаем

$$S_r = p\frac{1}{1+\frac{(p-1)t_w}{2nt_o}}. \qquad (11)$$

Из сравнения (10) и (11) следует

Утверждение 5. При $D>1$ и любых значениях параметров изображения и вычислительной системы для отображения сети Хопфилда распараллеливание по нейронам является более эффективным, чем распараллеливание по синапсам, т. е. $S_r > S_c$.

Заключение

Рассмотрены методы отображения слоя нейронов на распределенные вычислительные системы с тороидальной структурой при обработке изображений:

- распределение строк матрицы весовых коэффициентов по машинам (параллелизм нейронов);
- распределение столбцов матрицы весов (параллелизм синапсов).

Показано, что выбор способа отображения зависит от отношения числа нейронов в слое и числа весовых коэффициентов нейрона (числа пикселей изображения): если число нейронов относительно мало, то более эффективным является метод распределения по столбцам, иначе более эффективным является распределение по строкам. В частности, для сети Хопфилда, для которой характерно равенство числа нейронов и числа весовых коэффициентов нейрона, лучший результат дает распределение по строкам при любом числе машин в системе.

Предложенные методы отображения дают равномерное распределение по машинам тороидальной ВС результатов преобразования (1). Следовательно, отображение матрицы весов второго слоя нейронов может быть реализовано аналогично отображению первого слоя. При этом способ отображения (по строкам или по столбцам) также определяется отношением числа нейронов второго слоя к числу его входных сигналов.

СПИСОК ЛИТЕРАТУРЫ

1. Нейрокомпьютеры в прикладных задачах обработки изображений. Кн. 8 / Под ред. А.Н. Балухто, А.И. Галушкина. – М.: Радиотехника, 2003. – 224 с.
2. Ghennam S., Benmahammed K. Image Restoration Using Neural Networks // Lecture Notes in Computer Sciences. – Springer-Verlag, 2001. – V. 2085. – P. 227–234.
3. Cray T3E [Электронный ресурс]. – режим доступа: http://www.cray.com/products/systems/crayt3e/1200e.html.
4. Yu H., Chung I-Hsin, Moreira J. Topology Mapping for Blue Gene/L Supercomputer // Proc. of the ACM/IEEE SC2006 Conf. on High Performance Networking and Computing. – November 11–17, 2006, Tampa, FL, USA. – ACM Press, 2006. – P. 52–64.
5. Прэтт У. Цифровая обработка изображений. – М.: Мир, 1982. – Кн. 1. – 312 с.
6. Sundararajan N., Saratchandran P. Parallel Architectures for Artificial Neural Networks. Paradigms and Implementations. – IEEE Computer Society, 1988. – 380 p.
7. Ayoubi R.A., Bayoumi M.A. Efficient Mapping Algorithm of Multilayer Neural Network on Torus Architecture // IEEE Trans. on Parallel and Distributed Systems. – 2003. – V. 14. – № 9. – P. 932–943.
8. Gonzalez A., Valero-Garcia M., Diaz de Cerio L. Executing Algorithms with Hypercube Topology on Torus Multicomputers // IEEE Trans. on Parallel and Distributed Systems. – 1995. – V. 6. – № 8. – P. 803–814.
9. Tarkov M.S., Mun Y., Choi J., Choi H.-I. Mapping Adaptive Fuzzy Kohonen Clustering Network onto Distributed Image Processing System // Parallel Computing. – 2002. – V. 28. – № 9. – P. 1239–1256.

Поступила 14.07.2008 г.