УДК 681.324

Головко В.А., Дунец А.П., Савицкий Ю.В.

ЭМУЛЯЦИЯ НЕЙРОННЫХ СЕТЕЙ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ

ВВЕДЕНИЕ

Нейронные сети характеризуются параллельной архитектурой и соответственно параллельной обработкой информации. Эмуляция нейронных сетей на стандартных микропроцессорах приводит к неэффективности вычислений, что связано с последовательным функционированием микропроцессора и противоречит параллельной природе нейронных сетей. Поэтому для моделирования нейронных сетей на системном уровне используют различные многопроцессорные конфигурации [1]. Для реализации нейронных сетей на многопроцессорных компьютерах необходимо разработать параллельные алгоритмы функционирования и обучения нейронных сетей.

В данной работе рассматривается параллелизация многослойного персептрона, который может применяться в качестве нелинейной авторегрессионой модели при прогнозировании погрешностей от сенсорных устройств. Параллельные алгоритмы разрабатывались и тестировались на суперкомпьютере Origin 2000.

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ ПОСЛЕДОВАТЕЛЬНОГО ОБУЧЕНИЯ

При последовательном обучении модификация весовых коэффициентов происходит после подачи каждого образа на вход нейронной сети [2]. Рассмотрим прогнозирующий персептрон, который имеет один скрытый слой и один выходной нейронный элемент (рис. 1).

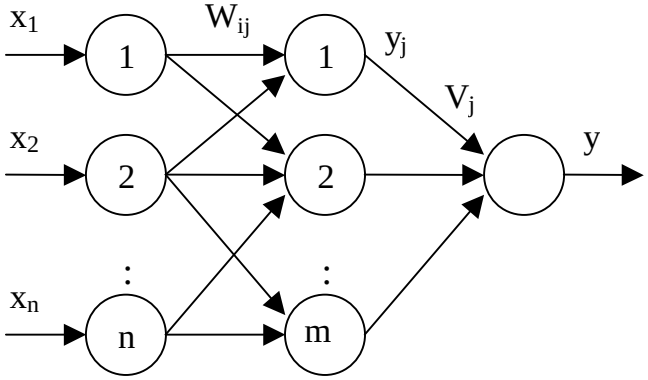

Рисунок 1 – Прогнозирующий персептрон.

Рассмотрим процесс обучения такой сети. Для каждого входного образа он состоит из следующих фаз: прямое распространение (FP), обратное распространение (BPE) и изменение весовых коэффициентов (UPD). При прямом распространении вычисляются выходные значения различных слоев нейронной сети:

$$y_j = F\left(\sum_{i=1}^{n} \omega_{ij} x_i - T_j\right), \quad (1)$$

$$y = F\left(\sum_{j=1}^{m} v_j y_j - T\right) \quad (2)$$

Элементарными операциями при вычислении выходных значений являются операции умножения, сложения (вычитания) и нелинейного преобразования. Общее количество элементарных операций при вычислении выходных значений скрытого и выходного слоя для одного образа соответственно равняются

$$V(1) = m(2n+1), \quad (3)$$

$$V(2) = 2m+1 \quad (4)$$

При обратном распространении вычисляются соответственно ошибки скрытого и выходного слоев:

$$\gamma = (y - t), \quad (5)$$

$$\gamma_j = \gamma v_j, \quad (6)$$

где $j = \overline{1,m}$, t – эталонное значение.

Общее количество операций при этом равняется

$$V(\gamma) = 1, \quad (7)$$

$$V(\gamma_i) = m \quad (8)$$

Фаза модификации состоит в изменении весовых коэффициентов и пороговых значений скрытого и выходного слоев.

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha\gamma_j F'(S_j) x_i, \quad (9)$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j F'(S_j), \quad (10)$$

$$v_j(t+1) = v_j(t) - \alpha\gamma F'(S) y_j, \quad (11)$$

$$T(t+1) = T(t) + \alpha\gamma F'(S) \quad (12)$$

Соответственно общее количество операций для скрытого и выходного слоя определяется при этом следующим образом:

$$V(u_1) = m(4n+3), \quad (13)$$

$$V(u_2) = (4m+3) \quad (14)$$

Анализируя фазу обучения многослойного персептрона, можно заметить, что одновременное выполнение операций возможно только в рамках одного слоя нейронной сети. Так как при последовательном обучении нейронные элементы разных слоев нейронной сети не могут работать параллельно в конвейерном режиме, то наиболее эффективным способом параллелизации здесь является назначение нейронным элементам разных слоев своего процессора.

Это схематично показано на рис. 2 для двух процессоров. В таблице 1 представлены временные последовательности выполнения процесса обучения при использовании данной схемы параллелизации. В таблице ***FP(i)*** обозначает, что производится соответствующая операция над ***i***-м слоем. Здесь предполагается, что процессор ***Pi*** в каждый момент времени может находиться в режиме вычислений или в режиме межмодульного обмена. Общее количество процессоров равняется ***b***. Недостатком предложенной схемы является неравномерная загрузка процессоров.

Дунец Андрей Петрович. *Ст. преподаватель каф. ЭВМ и С Брестского государственного технического университета. Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.*

Таблица 1 – Временные последовательности выполнения процесса обучения.

P1	FP(1)		FP(2)	BPE(2)		BPE(1)	UPD(2)	UPD(1)
P2	FP(1)					BPE(1)	UPD(2)	UPD(1)
…..								
Pb	FP(1)					BPE(1)	UPD(2)	UPD(1)
Фазы вычислений	1		2	3		4	5	6
Фазы межмодульного обмена		1			2			

Таблица 2 – Временные характеристики обучения нейронной сети.

Входы	Скрытый слой	Выходы	Связи	$t_p\ L$	$t_2\ L$	$t_s\ L$
100	15	1	1531	13.73	8.46	10.54
100	25	1	2551	18.36	10.45	15.82
200	25	1	5051	20.75	9.4	22.7
200	35	1	7071	25.21	9.18	32.06

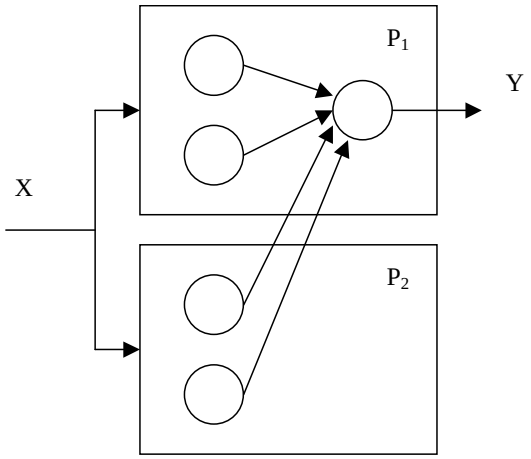

Рисунок 2 – Структура схемы параллелизации.

Рассмотрим временные соотношения, позволяющие провести сравнительный анализ производительности вычислений в параллельном и последовательном режимах. Пусть t_s – время обучения нейронной сети в последовательном режиме для одного образа, а t_2 – время передачи информации от одного процессора к другому в фазе межмодульного обмена. Общее количество передачи данных приближенно равняется $(b-1)$.

Тогда время обучения нейронной сети для одного эталона в параллельном режиме определяется следующим образом:

$$t_p = \frac{t_s}{b} + (b-1)t_2 \qquad (15)$$

Для эффективной параллелизации необходимо, чтобы

$$\frac{t_s}{b} + (b-1)t_2 < t_s \qquad (16)$$

Из последнего выражения можно получить верхнюю границу оценки количества процессоров:

$$b < \frac{t_s}{t_2} \qquad (17)$$

Найдем оптимальное количество процессоров, при котором обеспечивается максимальная производительность. Для этого необходимо минимизировать следующее выражение:

$$\frac{t_s}{b} + (b-1)t_2 - t_s \to min \qquad (18)$$

Беря производную и приравнивая ее к нулю, можно получить, что оптимальное количество процессоров

$$b_0 \approx \sqrt{\frac{t_s}{t_2}} \qquad (19)$$

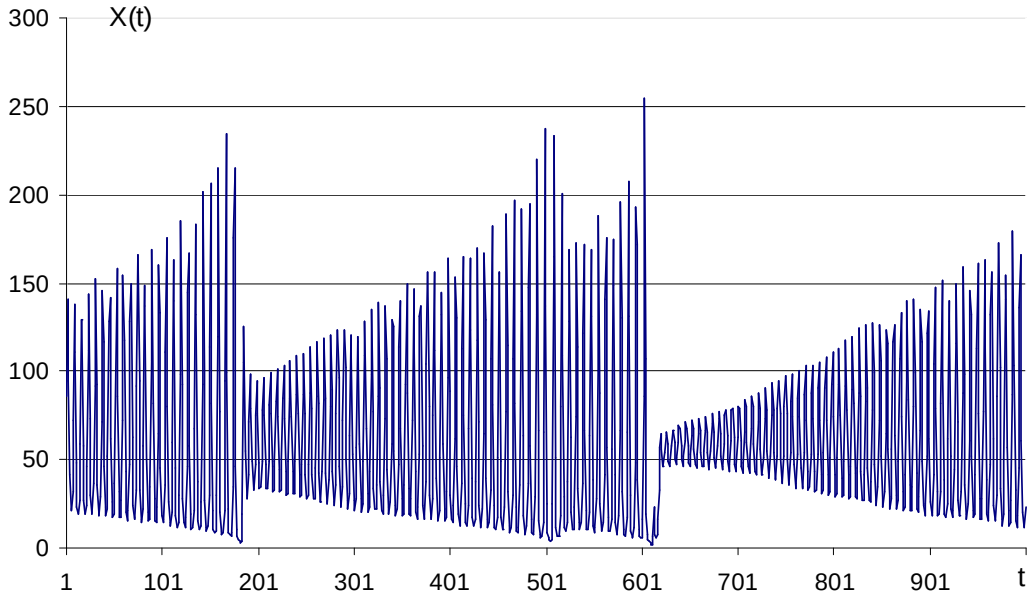

Рисунок 3 – Временной ряд хаотических пульсаций интенсивности NH_3 лазера.

Для тестирования рассмотренной выше схемы параллелизации нейронной сети использовался суперкомпьютер Origin 2000. Реализация параллельных алгоритмов в данной вычислительной среде базируется на библиотеке интерфейса передачи сообщений MPI, который позволяет разрабатывать параллельные приложения в системе программирования C++. В качестве обучающей выборки использовался ряд хаотических пульсаций интенсивности NH_3 лазера размером 650 элементов (рис. 3).

Данный ряд применялся для сравнительного анализа различных моделей прогнозирования в институте Санта Фе [3]. В таблице 2 приводятся временные характеристики обучения нейронной сети с различным количеством входных и скрытых нейронных элементов. В данных экспериментах использовались два процессора.

Как следует из таблицы, применение рассмотренной выше схемы параллелизации становится эффективным, когда $t_2 < t_s/2$. Это хорошо согласуется с предложенными теоретическими соотношениями. Следует также отметить, что в процессе проведения экспериментов, на Origin 2000 выполнялось параллельно 66 других задач. Этим можно объяснить различные значения времени t_2 в таблице 2.

Рассмотрим теперь схему паралеллизации на уровне функционирования нейронной сети. В этом случае можно предложить конвейерно-параллельную схему функционирования нейронной сети (рис. 4).

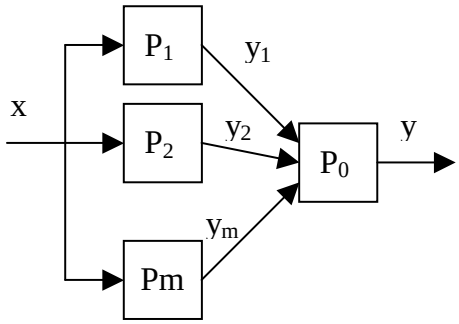

Рисунок 4 – Схема параллелизации функционирования нейронной сети.

В данной схеме на первом этапе процессоры Pi, $i=1,m$ вычисляют выходные значения нейронных элементов скрытого слоя для первого образа и передают их процессору P_0. На втором этапе процессор P_0 вычисляет выходное значение нейронной сети, а процессоры P_i определяют выходные значения нейронов скрытого слоя для второго образа. Пусть t_1 – время выполнения одной элементарной операции (сложения или умножения), а t_2 – время обмена информацией между процессорами P_i и P_0. Тогда время выполнения операций соответственно на первой и второй ступени конвейера

$$T_1 = (2n+1)t_1 + t_2 \quad (20)$$

$$T_2 = (2m+1)t_1 \quad (21)$$

Так, как время такта конвейера определяется временем прохождения самой медленной ступени, то

$$\alpha = \frac{1}{T_1} \quad (22)$$

Определим соотношение между временем обмена информацией t_2 и временем выполнения элементарной операции t_1 для эффективности работы конвейера. Тогда

$$(2n+1)t_1 + t_2 < m(2n+1)t_1 + (2m+1)t_1 \quad (23)$$

Отсюда можно получить

$$t_2 < t_1(2mn+3m-2n) \quad (24)$$

При выполнении последнего соотношения эффективно использовать конвейерно-параллельную параллелизацию для моделирования функционирования нейронной сети.

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ ГРУППОВОГО ОБУЧЕНИЯ

При групповом обучении модификация синаптических связей производится после подачи на вход нейронной сети L образов. В этом случае на стадии обучения и функционирования сети можно использовать конвейерный принцип обработки информации. Он заключается в разбиении задачи на связанные по фазе подзадачи и назначении каждой подзадаче своего процессора. Алгоритм обратного распространения ошибки можно разбить на следующие связанные между собой части: вычисление выходной активности скрытого слоя; вычисление выходной активности последнего слоя; вычисление ошибки γ и модификация синаптических связей последнего слоя; вычисление ошибки γ_i и модификация синаптических связей скрытого слоя. Назначая каждому этапу свой процессор, можно получить конвейерную схему обработки информации, представленную на рис. 5.

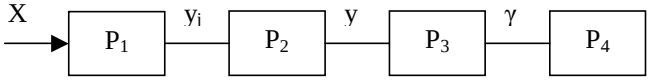

Рисунок 5 – Конвейерная схема обучения.

В таблице 3 представлено функционирование конвейерной схемы обучения для четырех тактов. Обозначение $FP(i,j)$ характеризует соответствующую операцию над i-м слоем и j-м образом сети.

Определим производительность такой схемы. Пусть α - такт конвейера. Общее время вычислений для L образов можно представить следующим образом:

$$t_0 = (4+L)\alpha \quad (23)$$

Тогда количество обрабатываемых образов в единицу времени

$$R = \frac{L}{(4+L)\alpha} \quad (24)$$

Найдем приближенную зависимость производительности конвейерного процессора от размерности обучающей выборки L. Пусть t_2 – общее время обмена информацией между процессорами, а t_s – время обработки информации в последовательном режиме. Тогда время такта конвейера можно при-

Таблица 3 – Функционирование конвейерной схемы обучения.

№ такта	P_1	P_2	P_3	P_4
1	FP(1,1)			
2	FP(1,2)	FP(2,1)		
3	FP(1,3)	FP(2,2)	BPE(2,1) UPD(21,)	
4	FP(1,4)	FP(2,3)	BPE(2,2) UPD(2,2)	BPE(1,1) UPD(1,1)

ближенно определить как

$$\alpha = \frac{t_s + t_2}{4} \quad (25)$$

Тогда общее время вычислений

$$t_0 = \frac{(4+L)(t_s + t_2)}{4} \quad (26)$$

Для эффективности вычислений в конвейерном режиме необходимо выполнение следующего неравенства:

$$\frac{(4+L)(t_s + t_2)}{4} < t_s L \quad (27)$$

Выражая из последнего выражения L, получим

$$L > \frac{4(t_s + t_2)}{3t_s - t_2} \quad (28)$$

Проведем анализ трудоемкости вычислений каждой стадии конвейера. Используя результаты предыдущего раздела, можно получить количество операций, вычисляемых каждым процессором

$$V(P_1) = m(2n+1) \quad (29)$$

$$V(P_2) = (2m+1) \quad (30)$$

$$V(P_3) = (4m+4) \quad (31)$$

$$V(P_4) = m(4n+4) \quad (32)$$

Отсюда следует, что различные ступени конвейера характеризуются различной сложностью вычислений. При этом время такта конвейера будет определяться временем прохождения самой медленной ступени P_4. Для нейтрализации этого недостатка можно предложить конвейерно-параллельную схему. Она заключается в том, что самая медленная ступень разбивается, например, на m параллельно работающих процессоров, как это показано на рис. 6 для четвертой ступени.

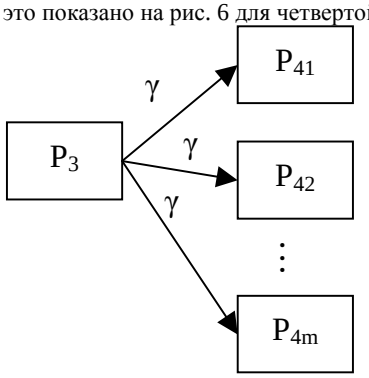

Рисунок 6 – Схема конвейерно-параллельной параллелизации.

В результате этого количество операций четвертой ступени конвейера можно сократить в среднем в m раз. Аналогичные действия можно применить для других ступеней конвейера, что позволяет повысить производительность вычислений.

Следующий вариант распараллеливания алгоритма обучения – групповая параллелизация. Такая схема предполагает разбиение обучающего множества на K групп и использования для каждой группы своей нейронной сети (рис. 7).

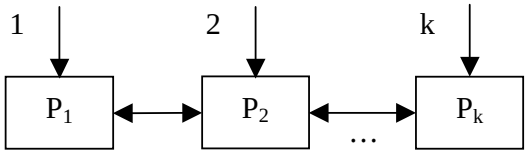

Рисунок 7 – Групповая параллелизация.

Количество образов в группе равняется L/K. Фаза межмодульного обмена в такой схеме происходит после подачи L/K образов в каждый процессор и заключается в изменении соответствующих синаптических связей. В качестве процессора P_i в такой схеме можно использовать последовательный процессор, параллельный процессор, рассмотренный в предыдущем разделе, и одну из конвейерных схем. Если применяется последовательный процессор, то производительность групповой параллелизации в K раз больше по сравнению с одним процессором. Аналогичное соотношение наблюдается для других схем параллелизации.

ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе рассмотрены различные схемы реализации нейронных сетей на многопроцессорных системах. Приведены аналитические соотношения, позволяющие оценивать эффективность различных схем параллелизации. Окончательный выбор варианта пареллелизации зависит от архитектуры многопроцессорной системы и допустимых аппаратных издержек, затрачиваемых на реализацию соответствующего алгоритма параллелизации.

СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Frank M. Thiesing, Ulrich Middelberg, Oliver Vornberger. Parallel Back-Propagation for Sales Prediction on Transputer System. Harrogate. UK. Proc. Of World Transputer Congress. 1995.
2. Головко В.А. Нейроинтеллект: теория и применение. Книга 1: Организация и обучение нейронных сетей с прямыми и обратными связями. Брест. Изд. БПИ, 1999 – 264 с.
3. Weigend A., Gershenfeld N. Time series prediction: forecasting the future and understanding past // Proceedings of the Santa Fe Institute. New Mexico: Addison–Wesley. – 1992. – 336 p.

УДК 681.324: 519.711.7

Махнист Л.П.

ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДА СОПРЯЖЕННЫХ ГРАДИЕНТОВ

ВВЕДЕНИЕ

Рассмотрим нейронную сеть, состоящую из *n* нейронных элементов распределительного слоя и m - выходного слоя (рис. 1).

Для данной сети каждый нейрон распределительного слоя имеет синаптические связи w_{ij}, $\left(i = \overline{1,n},\ j = \overline{1,m}\right)$ со всеми нейронами обрабатывающего слоя. В качестве нейронов выходного слоя используются элементы с некоторой функцией активации F [1, 2]. На вход сети подаются входные

Махнист Леонид Петрович. *К.т.н., доцент каф. высшей математики Брестского государственного технического университета.*
Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.