УДК 681.324

Савицкий Ю.В.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СХЕМ ПАРАЛЛЕЛИЗАЦИИ ФУНКЦИОНИРОВАНИЯ НЕЙРОННОЙ СЕТИ

Введение. В последние несколько лет исследования в области нейронных сетей ориентированы в основном на создание специализированных систем для решения конкретных задач. Разработано большое количество нейросистем, которые успешно применяются в самых различных областях – бизнесе, медицине, технике, геологии, физике. Нейронные сети вошли в практику везде, где нужно решать задачи прогнозирования, классификации или управления. Такой впечатляющий успех определяется несколькими причинами [1]:

- нейронные сети представляют исключительно мощный метод моделирования, позволяющий воспроизводить сложные зависимости;
- возможность извлечения закономерностей на основе только множества обучающих эталонов. Пользователь нейронной сети подбирает представительную выборку, после чего производит обучение, в процессе которого нейронная сеть автоматически воспринимает структуру данных. При этом от пользователя требуется некоторый набор знаний и опыт в том, как следует отбирать и подготавливать данные, выбирать нужную архитектуру сети и интерпретировать результаты.

Нейронные сети – это сети, состоящие из связанных между собой простых элементов – формальных нейронов и связей между ними. Каждая связь является неделимой частью нейронной сети, служащей для передачи сигнала и его линейного усиления или ослабления. Таким образом, нейронные сети представляют собой однородную систему, значения элементов которой вычисляются по одному и тому же правилу. Они характеризуются параллельной архитектурой и соответственно возможностью параллелизации нейросетевых алгоритмов обработки информации. Эмуляция нейронных сетей на стандартных однопроцессорных вычислительных системах приводит к неэффективности вычислений, что связано с последовательным функционированием подобных систем и противоречит параллельной природе нейронных сетей. Поэтому для моделирования нейронных сетей лучше использовать многопроцессорные (многомашинные) конфигурации, обеспечивающие эффективную параллелизацию процессов обработки.

Однако при этом остается открытым вопрос, насколько эффективны в плане производительности те или иные схемы параллелизации, реализованные в виде конкретных программных систем.

1. Схемы параллелизации нейросетевых алгоритмов. В качестве базовой архитектуры нейронной сети примем многослойный персептрон, для которого разработано наибольшее количество успешно функционирующих нейросетевых приложений.

Основными фазами функционирования нейронной сети, подвергаемыми параллелизации, являются вычисление активности выходных нейронов (прямое распространение сигналов) и обучение (модификация межнейронных связей нейронов, осуществляемая, как правило, на основе алгоритма обратного распространения ошибки и его более быстрых модификаций).

Рассмотрим схемы параллелизации функционирования многослойного персептрона.

Схема параллелизации "один нейрон для одного вычислительного модуля (ВМ)". В архитектуре многослойного персептрона есть нейронные элементы, которые могут работать и обучаться в параллельном режиме (независимо друг от друга). К ним относятся нейроны одного слоя, так как вычисление их выходного значения зависит только от нейронных элементов предыдущего слоя, а от нейронов этого же слоя – не зависит. Поэтому можно использовать это свойство нейронной сети для разработки параллельного алгоритма ее обучения и функционирования, позволяющего ускорить этот процесс.

Смысл этого алгоритма заключается в следующем. При обучении на вход сети последовательно подаются эталоны из обучающей выборки. Для каждого эталона последовательно выполняются фазы обучения: прямое распространение сигнала, обратное распространение сигнала, изменение весовых коэффициентов и порогов нейронных элементов. При выполнении каждой из этих фаз каждый нейронный элемент текущего слоя "обрабатывается" на отдельном (ВМ). Затем результаты вычислений всех нейронов текущего слоя передаются всем ВМ, которые затем начинают выполнять вычисления над нейронными элементами следующего слоя. ВМ должно быть не меньше, чем максимальное количество нейронных элементов в одном из слоев.

Схема параллелизации "группа нейронов для одного ВМ". Главное отличие между схемой параллелизации "группа нейронов для одного ВМ" и схемой параллелизации "один нейронный элемент для одного ВМ" состоит в том, что каждый ВМ при выполнении фаз прямого распространения сигнала, обратного распространения сигнала, изменения весовых коэффициентов и порогов нейронных элементов производит вычисления не над одним нейроном слоя, а над группой нейронов этого слоя. Нейронные элементы каждого слоя разбиваются на М групп (М – количество ВМ). Размерность групп нейронов может не совпадать, но должна отличаться не более, чем на один нейрон. Каждый ВМ выполняет вычислительные действия при обучении и функционировании многослойного персептрона над определенной группой нейронов. Поэтому структура этой схемы параллелизации будет аналогичной предыдущей.

Достоинство этого метода заключается в том, что при большой размерности слоев нейронной сети не нужно использовать большое количествоВМ. При его использовании нейронную сеть с любым количеством нейронов в слоях можно обучить даже на двух ВМ.

Схема параллелизации "слой за слоем". В основе метода лежит принцип конвейерной обработки информации. При работе конвейера, процесс обработки организуется таким образом, чтобы каждый блок был всегда загружен выполнением соответствующей обработки цепочки задач. При этом результаты обработки последовательно передаются следующему операционному блоку, который работает в аналогичном режиме. Режим функционирования конвейера характеризуется периодом загрузки – это начальный период функционирования операционного конвейера, в течение которого осуществляется загрузка всех операционных устройств. Очевидно, что спустя время загрузки конвейера на его выходе формируются первые результаты вычислений. Период загрузки определяется числом операционных устройств и максимальным временем выполнения какого-либо этапа обработки.

В данном случае в качестве операционных блоков используются ВМ. Их количество равняется количеству слоев нейронной сети, над нейронными элементами которых нужно выполнять определенные действия. При обучении многослойного персептрона фазы прямого распространения сигнала, обратного распространения сигнала, изменения весовых коэффициентов и порогов нейронных элементов можно разбить на ряд шагов. На каждом шаге обрабатываются нейроны определенного слоя (число шагов равно числу обрабатываемых слоев) на соответствующем вычислительном модуле. Затем результаты обработки передаются следующему ВМ, принимаются результаты обработки следующего эталона от предыдущего ВМ.

***Савицкий Юрий Викторович**, к.т.н., доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*
Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

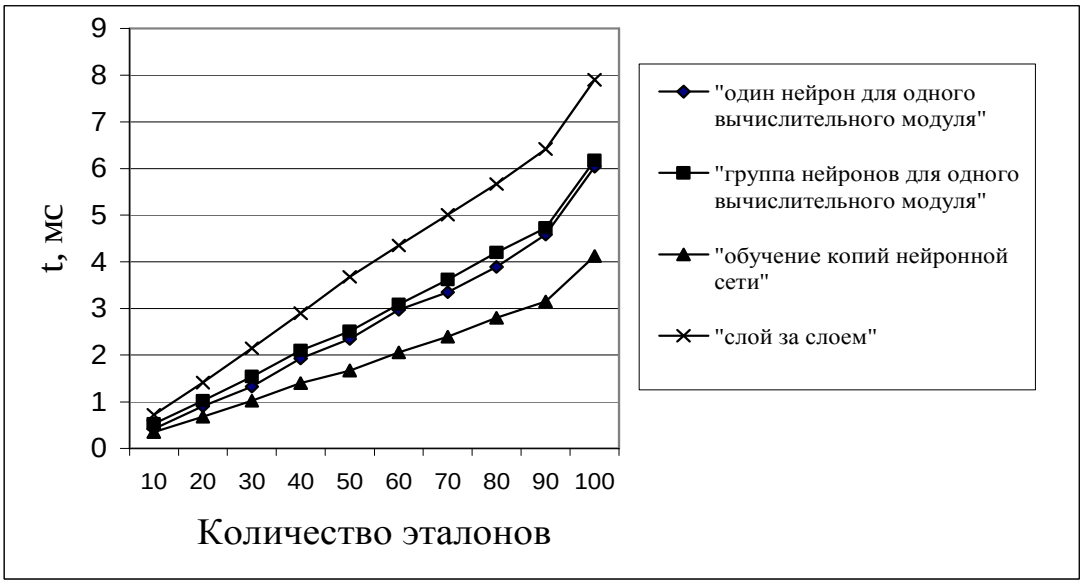

Рис. 1. *График временной зависимости функционирования нейронной сети для параллельных алгоритмов от количества эталонов без учета межмодульных пересылок*

Этот процесс продолжается до тех пор, пока не будут обработаны все эталоны из обучающей выборки.

Данная схема параллелизации реализует групповой метод обучения, когда изменение весовых коэффициентов и порогов нейронных элементов происходит только после того, как будут вычислены значения нейронных элементов и их ошибок для всех эталонов. Изменение происходит с учетом каждого эталона.

Схема параллелизации "использование копий нейронной сети". Главная идея этого алгоритма заключается в разделении обучающего набора эталонов R на подмножества *R(1), R(2), …, R(k)* и использовании *k* копий многослойного персептрона для обработки каждого из этих подмножеств. Каждая копия нейронной сети обучается на своем обучающем подмножестве. Результаты обучения копий тем или иным способом интегрируются между собой. В итоге получается нейронная сеть, обученная при помощи всего обучающего набора элементов. При обучении копий многослойного персептрона может быть использована любая из предыдущих схем параллелизации. Например, возможны два варианта структуры схем параллелизации: в первом случае для обучения каждой копии нейронной сети используется метод, в котором каждый ВМ обрабатывает группу нейронных элементов, а во втором случае применяется метод, использующий конвейерный принцип обработки информации.

2. Программные модели схем параллелизации. Вышеописанные схемы параллелизации реализованы в виде трех программных модулей, реализованных в системе Visual C++ на базе протокола передачи сообщений MPI [2]. MPI (Message passing interface) представляет собой библиотеку функций для передачи сообщений, позволяющую со сравнительно небольшой трудоемкостью трансформировать последовательную программу в программу, которая будет полностью использовать параллельную архитектуру используемой вычислительной системы. По сути это - программный инструментарий для обеспечения связи между ветвями параллельного приложения.

Часто приходится слышать утверждение, что низкоуровневые пересылки через разделяемую память и семафоры предпочтительнее применения таких библиотек как MPI, потому что работают быстрее. Однако можно выделить ряд несомненных преимуществ описываемого подхода: 1) в хорошо распараллеленном приложении на собственно взаимодействие между ветвями (пересылки данных и синхронизацию) тратится небольшая доля времени - несколько процентов от общего времени работы. Таким образом, замедление пересылок, допустим, в два раза не означает общего падения производительности вдвое - она понизится на несколько процентов. Зачастую такое понижение производительности является вполне приемлемым; 2) MPI - это изначально быстрый инструмент. Для повышения скорости в нем используются приемы, о которых прикладные программисты зачастую просто не задумываются. Например, встроенная буферизация позволяет избежать задержек при отправке данных - управление в передающую ветвь возвращается немедленно, даже если ветвь-получатель еще не подготовилась к приему. MPI использует многопоточность (multi-threading), вынося большую часть своей работы в потоки (threads) с низким приоритетом. Буферизация и многопоточность сводят к минимуму негативное влияние неизбежных простоев при пересылках на производительность прикладной программы. На передачу данных типа "один-всем" затрачивается время, пропорциональное не числу участвующих ветвей, а логарифму этого числа [2].

Параллельное приложение состоит из нескольких ветвей, или процессов, или задач, выполняющихся одновременно. Процессы обмениваются друг с другом данными в виде сообщений. Сообщения проходят под идентификаторами, которые позволяют программе и библиотеке связи отличать их друг от друга. Каждый процесс в зависимости от его номера приступает к выполнению соответствующей части расчетов. В MPI ветвь запускается и работает как обычный процесс, связанный через MPI с остальными процессами, входящими в приложение. В остальном процессы следует считать изолированными друг от друга: у них разные области кода, стека и данных.

3. Сравнительный анализ схем параллелизации. Производительность – это характеристика вычислительной мощности системы, определяющая количество вычислительной работы, выполняемой системой за единицу времени. В контексте данных исследований производительность параллельных алгоритмов функционирования нейронной сети будем оценивать при помощи временных соотношений функционирования схем параллелизации.

Сначала проанализируем временные соотношения функционирования нейронной сети без учета времени межмодульных пересылок в зависимости от количества нейронных элементов и в зависимости от количества входных эталонов для всех параллельных алгоритмов.

Из графиков видно (рис.1, рис. 2), что самое большое время функционирования без учета межмодульных пересылок получается при использовании алгоритма "слой за слоем" и последовательного алгоритма, так как в этом случае каждый процесс "обрабатывает" все нейроны слоя. Время функционирования нейронной сети в большей степени зависит от количества входных эталонов, чем от количества нейронных элементов в слоях. Увеличение числа нейронов в слоях практически не влияет на время функционирования нейронной сети при использовании алгоритма "один нейрон для одного ВМ", поскольку все

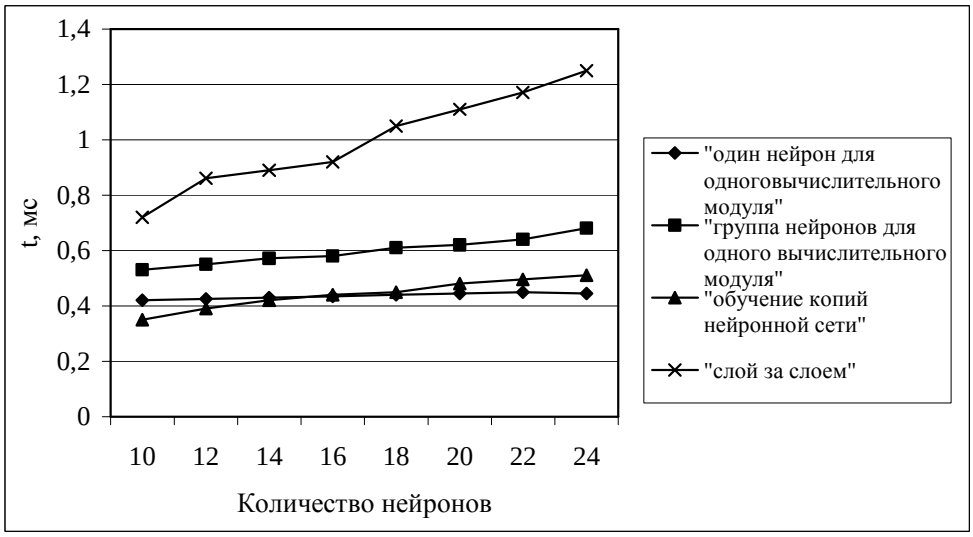

Рис. 2. *График временной зависимости функционирования нейронной сети для параллельных алгоритмов от количества нейронов в слоях без учета межмодульных пересылок*

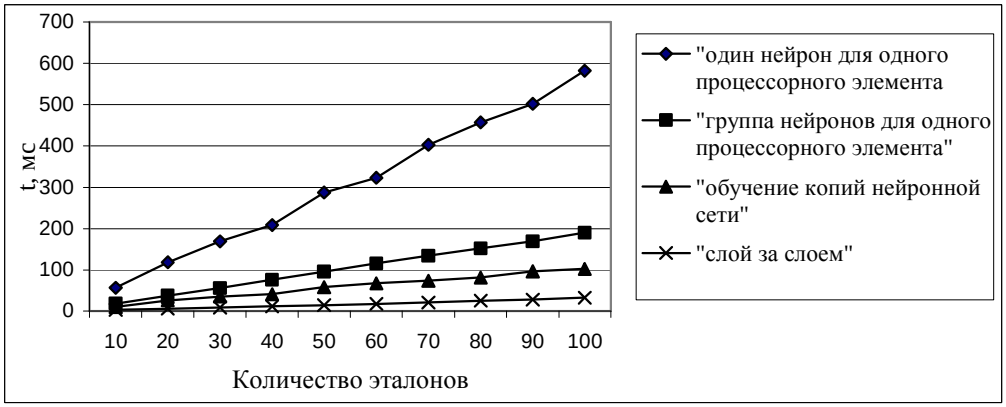

Рис. 3. *График временной зависимости функционирования нейронной сети для параллельных алгоритмов от количества эталонов с учетом межмодульных пересылок*

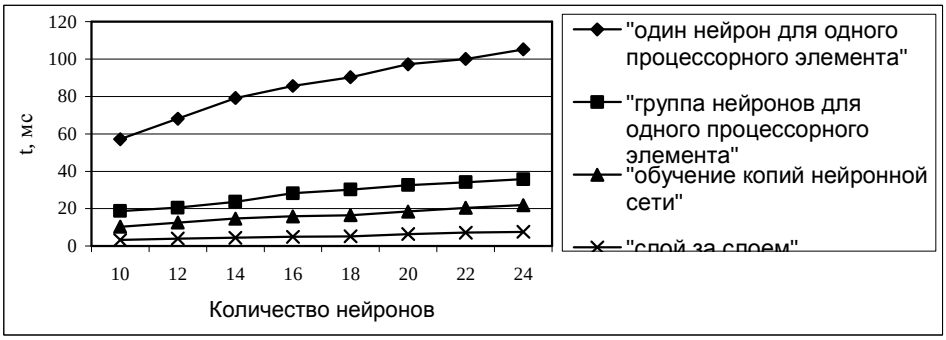

Рис. 4. *График временной зависимости функционирования нейронной сети для параллельных алгоритмов от количества нейронов в слоях с учетом межмодульных пересылок*

процессы выполняют вычислительные операции над одним нейроном, однако увеличивается число создаваемых процессов.

Также проведем ряд тестов для оценки времени функционирования нейронной сети в зависимости от количества входных эталонов и в зависимости от количества нейронных элементов в слоях, но с учетом межмодульных пересылок. Их результаты отображены на рис. 3 и рис. 4.

Из графиков видно, что время функционирования сети существенно возросло. Также из графиков видно, что время функционирования сети получается меньше, если использовать алгоритмы, в которых используется меньшее число обменов информацией между процессами. Таким образом, более производительными получаются те алгоритмы функционирования нейронной сети, в которых используется меньше межпроцессорных обменов. К ним относятся: “слой за слоем”, “обучение копий нейронной сети” и “группа нейронов для одного ВМ”. Самым непроизводительным является алгоритм “один нейрон для одного ВМ”, из-за использования большого числа обменов данными между процессами.

Таким образом, возможными путями оптимизации программных MPI-систем в схемах параллелизации являются: использование многокомпонентных вычислительных систем с меньшими интервалами межмодульных пересылок; уменьшение количества межмодульных пересылок, в том числе за счет их объединения.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Головко, В.А. Нейроинтеллект: теория и применение. Книга 1: Организация и обучение нейронных сетей с прямыми и обратными связями. - Брест: БПИ, 1999.
2. Оленев, Н.Н. Основы параллельного программирования в системе MPI. - М.: ВЦ РАН, 2005.

Материал поступил в редакцию 25.09.2008

SAVITSKY Yu.V. The comparative analysis of power of neural network functioning parallel schemes

In this article the results of power of parallel neural network functioning algorithm are considered. The parallel schemes are made as program system based on the MPI (Message Passing Interface) protocol. The computational experiments of time dependence from number of neurons, number of training patterns are discussed. The analysis of parallel algorithms time complexity with MPI message passing and without MPI message passing is given.

УДК 681.324

Савицкий Ю.В.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СХЕМ ПАРАЛЛЕЛИЗАЦИИ АЛГОРИТМОВ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

Введение. Как показывает практика, в нейросетевых методах обработки информации наиболее трудоемким этапом в вычислительном и временном аспекте является процесс обучения нейронной сети [1]. Процесс обучения, также как и процесс функционирования, характеризуются параллельной архитектурой и соответственно возможностью параллелизации и увеличения производительности.

В статье автора «Сравнительный анализ производительности схем параллелизации обучения нейронной сети» были рассмотрены схемы параллелизации функционирования многослойного персептрона *“один нейрон для одного вычислительного модуля (ВМ)”, “группа нейронов для одного ВМ”, “слой за слоем”, “использование копий нейронной сети”.*

Задачей данной работы является всесторонне исследование производительности вышеуказанных схем параллелизации, примененных в задаче обучения многослойной нейронной сети.

Сравнительный анализ схем параллелизации. При обучении нейронной сети кроме вычисления выходных значений нейронных элементов всех слоев (функционирование нейронной сети) производится вычисление значений ошибок нейронных элементов, изменение весовых коэффициентов и порогов нейронных элементов каждого слоя и для каждого эталона. Следовательно, это более длительный и сложный в вычислительном аспекте процесс, чем функционирование нейронной сети.

Производительность параллельных алгоритмов обучения нейронной сети, будем оценивать при помощи временных соотношений их исполнения на базе разработанной MPI-программы [2]. Чем меньше время выполнения всех действий для некоторого числа эталонов нейронной сети, тем выше производительность алгоритма обучения (количество “обработанных” эталонов за определенное время).

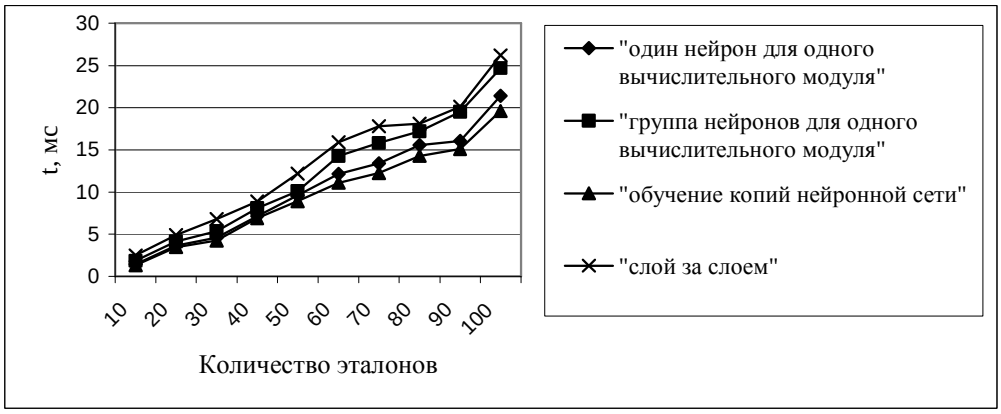

Рис. 1. *График временной зависимости обучения нейронной сети для параллельных алгоритмов от количества эталонов без учета межмодульных пересылок*

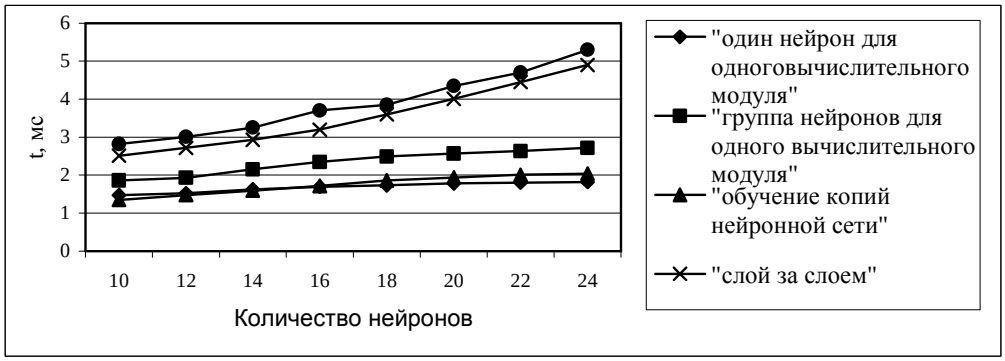

Рис. 2. *График временной зависимости обучения нейронной сети для параллельных алгоритмов от количества нейронов в слоях без учета межмодульных пересылок*