ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Сосинская С. С., канд. техн. наук, доц.,
Нгуен Данг Минь, аспирант
Иркутский государственный технический университет

ПРИМЕНЕНИЕ MEX-ФАЙЛОВ ПРИ РЕАЛИЗАЦИИ ПРОГРАММ ПАРАЛЛЕЛЬНОГО ОБУЧЕНИЯ ГИБРИДНЫХ СЕТЕЙ В СРЕДЕ MATLAB

soldier.85@mail.ru

Рассматривается способ создания внешних программ, написанных на языке программирования C++. Приводятся результаты вычислительных экспериментов при реализации программы параллельного обучения адаптивной нейро-нечеткой сети с использованием MEX-файла. Проводится сравнение эффективности с программой, написанной на языке MATLAB.

Ключевые слова: *MEX–файл, гибридная сеть, алгоритм обучения, алгоритм параллельного обучения.*

Введение

MATLAB является высоко производительной средой, написание и отладка программ в этой среде обычно занимает значительно меньше времени, чем на обычных языках программирования, таких как C, C + + или Fortran.

Однако, довольно часто эффективность подобных программ оставляет желать лучшего. Как правило, подобная ситуация возникает, когда алгоритм плохо векторизуется, например, при обработке матриц нельзя выразить этот алгоритм, пользуясь векторными операторами языка MATLAB, и приходится писать вложенные циклы, перераспределять память и т.п. В этом случае программа на языке C будет исполняться во много раз быстрее аналогичной программы на языке MATLAB. Это весьма важно, если время счета математической задачи исчисляется часами или сутками.[6]

Существует несколько способов взаимодействия системы MATLAB с внешними программами, один из них состоит в написании модуля, расширяющего набор функций MATLAB на языках программирования C/C++, Fortran.

Основный алгоритм обучения адаптивной системы нейро-нечеткого вывода (ANFIS) в пакете FuzzyLogicToolbox написан на языке C++ для повышения эффективности вычислений. Но как правило, обучение нейронных сетей требует значительных временных затрат для выполнения большого количества итераций и корректировки весов сетей. Решением является использование технологии параллельнных вычислений при обучении сети, что значительно снижает затраты времени.

В статье описывается создание MEX - файлов, написанных на языке программирования C. Приводится экспериментальная оценка эффективности применения MEX-файлов для параллельного обучения ANFIS.

Метод обратного распространения ошибки для обучения гибридной сети

Метод обратного распространения ошибки (backpropagation) - один из самых распространенных методов обучения сети. Это градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения целевого значения выходной переменной от желаемого значения выходной переменной многослойных нейронных сетей [4].

Алгоритм действует итеративно. Во время каждой итерации на вход сети подаются все объекты обучающей выборки, выходные значения сети сравниваются с целевыми значениями и вычисляется ошибка. Значение ошибки используется для корректировки весов, после чего все действия повторяются.

Пусть $y_{вых}^{i}$ является выходным значением сети для *i*-го объекта обучающей выборки, y^{i} является целевым значением *i*-го объекта, частное квадратичное отклонение вычисляется по формуле:

$$E_i = \left(y^i - y_{вых}^i\right)^2.$$

Вычисляемый сигнал ошибки распространяется обратно по сети и используется для вычисления градиента параметров сети.

Вектор частного градиента

$$g^i(b) = \left(\frac{dE_i}{db_1}, \frac{dE_i}{db_2}, \cdots, \frac{dE_i}{db_M}\right)^T,$$

где $b_1, b_2, \cdots, b_M$ - параметры сети.

Среднее отклонение $E_{ср}$ (значение невязки) и вектор градиента $g(b)$ вычисляются по формулам:

$$E_{ср} = \sqrt{\left(\sum_{i=1}^{N} E_i\right) \Big/ N},$$

$$g(b) = \sum_{i=1}^{N} g^i(b),$$

где: N – объем обучающей выборки.

Градиент поверхности ошибок L является модулем вектора $g(b)$.

Пусть S является шагом обучения или коэффициентом скорости обучения (на практике часто выбирают начальное значение S = 0.01), тогда параметры сети пересчитываются по формуле:

$$b_j = b_j - S \times \frac{W_j}{L},$$

где $W_j = \sum_{i=1}^{N} \frac{dE_i}{db_j}$, $j = 1,2,\cdots,M.$

После каждой итерации обучения сети коэффициент обучения S корректируется, если $\left(E_{ср}^{i} < E_{ср}^{i-1} < E_{ср}^{i-2} < E_{ср}^{i-3} < E_{ср}^{i-4}\right)$то S увеличивается (умножается на 1.1), а если $\left(E_{ср}^{i} > E_{ср}^{i-1}\right)$; $\left(E_{ср}^{i-1} < E_{ср}^{i-2}\right); \left(E_{ср}^{i-2} > E_{ср}^{i-3}\right);$ $\left(E_{ср}^{i-3} < E_{ср}^{i-4}\right)$, S уменьшается (умножается на 0.9).

Программную реализацию алгоритма обучения адаптивной нейро-нечеткой сети можно представить в виде набора взаимосвязанных этапов:

- Вычисление частных квадратичных ошибок E_i,
- Вычисление частных градиентов $g^i(b)$,
- Вычисление среднего отклонения $E_{ср}$, вектора градиента $g(b)$и обновление параметров b_j,
- Корректировка коэффициента обучения S.

При обучении сети основное время тратится на вычисление частных квадратичных ошибок E_i и частных градиентов $g^i(b)$, поэтому эти значения можно вычислять с помощью алгоритма параллельного вычисления.

Рассмотрим способ создания MEX - файлов, написанных на языке C/C++ и используемых для вычисления E_i и $g^i(b)$.

Создание MEX- файлов из языка программирования C/C++.

В документации по системе MATLAB для подобных расширений употребляется термин MEX-файл (MatlabEXtension), и сама MATLAB по этому расширению имени файла может определить, что данный модуль является ее расширением. Создание MEX–файлов на входном языке компилятора позволяет максимально использовать производительность компьютера, так как есть возможность написать наиболее оптимальный код.

Для создания MEX–файлов на входном языке компилятора необходим непосредственно ANSI C/C++ компилятор, VisualStudio 2008 или компилятор, встроенный в среду MATLAB. С целью компиляции в среде MATLAB при его установке выбирается дополнительная компонента MATLAB Compiler.

Двоичный MEX-файл создается с помощью сценария сборки *.mex,* который компилирует и соединяет исходные файлы в разделяемую библиотеку, его можно запустить в командной строке MATLAB, так же, как любую функцию MATLAB.

MEX-файл состоит из:

- *интерфейсной функции* − интерфейсы C / C + + и данных MATLAB,
- *вычислительных процедур* – они написаны на C / C ++ и выполняют вычисления, которые необходимо реализовать в двоичным MEX-файле,
- *макросов препроцессора* − для создания независимого от платформы кода[5].

Следующая диаграмма показывает, как задавать входные данные для MEX-файлов, какие функции выполняются, и как выходные данные передаются в MATLAB.

Прототип интерфейсной функции объявляется в заголовочном файле \matlab\extern \include\mex.h следующим образом:

*void**mexFunction**(*
*intnlhs, mxArray *plhs[],*
*intnrhs, constmxArray *prhs[]){/* дальнейшийкодна C/C++ ... */ }*

Название интерфейсной функции должно быть *mexFunction*; она должна содержать параметры *prhs, nrhs, plhs* и *nlhs,* имеющие следующее назначение:

prhs, plhs - массивы указателей на входные, выходные параметры,

nrhs, nlhs - количества входных, выходных параметров.

Параметры *prhs*и *plhs* имеют структуру *mxArray.*

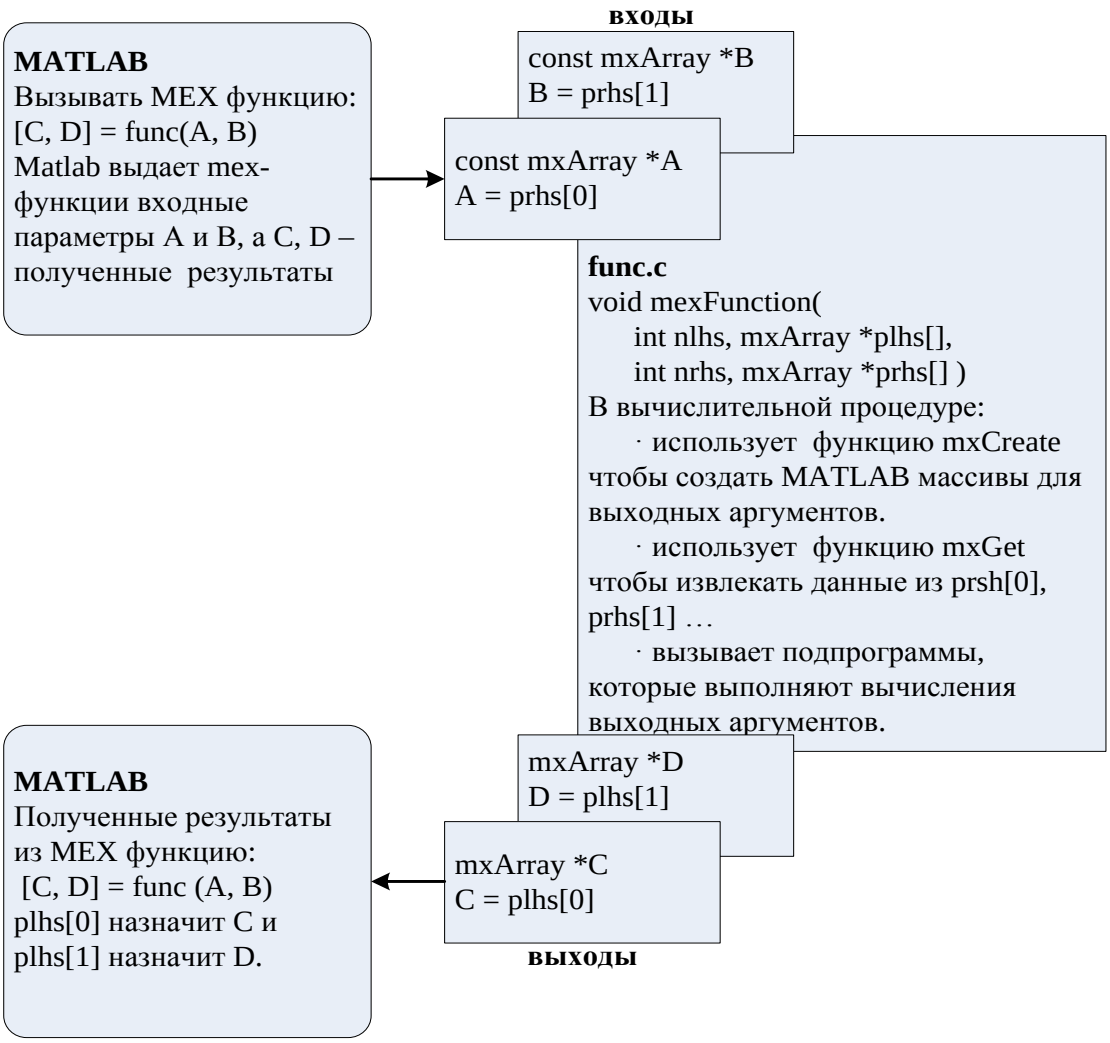

Рис. 1. Цикл MEX диаграммы

Автором написана функция Panfis.c для вычисления вектора градиентов и сумм частных квадратичных отклонений. На входе первый параметр был полем *fismat*, хранящим описание структуры сети, второй параметр является множеством обучающих выборок *trn_data*. Вызываемые функции написаны на языке C++, некоторые функции вызываются из пакета MATLAB FuzzyLogicToolbox.

Результат компиляции MEX-файла можно запускать в среде MATLAB. Для этого необходимо ввести следующую строку *:[E, g] = Panfis(fismat, trn_data);*

MEX-файл *Panfis.mexw64* можно использовать для реализации параллельного обучения сети. Алгоритм параллельного обучения гибридных нейро-нечетких сетей описан в статье [3].

Средства для реализации параллельных вычислений в MATLAB

В связи с необходимостью решения сложных вычислительных задач в параллельной среде, в сентябре 2010 г. компания MathWorks выпустила MATLAB DistributedComputingServer 5.0 (DCS), позволяющий выполнять алгоритмы, реализованные в среде MATLAB, на объединенных в сеть компьютерах. Для этого пользователь MATLAB должен создать набор задач (tasks), объединить их в задание (job) и передать задание для выполнения планировщику - менеджеру заданий (jobmanager), который является частью DCS. Менеджер заданий выполняет задания из очереди, распределяя содержащиеся в них задачи по известным ему процессам – исполнителям [1, 2].

Экспериментальная оценка эффективности

Для проведения экспериментов был использован локальный кластер, созданный на ноутбуках Alienware R2 и R3, имеющих следующие характеристики:

- Процессоры Intel Core i7 - 2720 - 2.2 Гц.
- Оперативная память 8Гб.
- Операционная система Window8 64-bit.
- Локальная сеть Ethernet со скоростью передачи 1Gb/s.

На каждом ноутбуке находится MATLAB версии 2010b с пакетом DistributedComputingServer 5.0 и VisualStudio 2008.

В данной работе исследуется эффективность параллельного обучения ANFIS для классификации втулок, которые имеют максимальные диаметры и высоты меньше 30 миллиметров, класс детали определен в соответствии с положениями, приведенными в [7, 8]: детали, характеризующиеся определенном соотношением высоты и максимального диаметра, попадут в определенный класс по технологии их обработки. Поэтому можно предположить наличие 5 классов для определенного соотношения высоты и диаметра, то есть 5 значений выходной величины в системе нечеткого вывода. Случайным образом были сгенерировано множество обучающих примеров в количестве 3000 объектов.

На рисунке 2 показана зависимость временных затрат от числа процессов при параллельном обучении гибридной сети с использованием MEX-файла. Сеть обучена после 1000 итераций для генерирования системы нечеткого вывода, имеющей две входные лингвистические переменные, каждая из которых содержит по пять термов, тип функций принадлежности - пи-подобные.

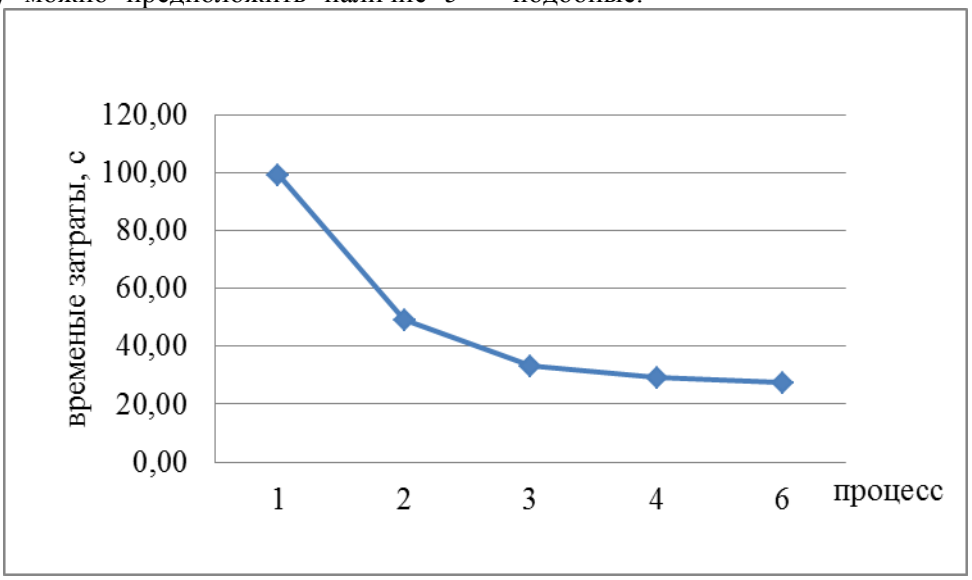

Рис. 2. Зависимость времени обучения от количества процессов

Ранее был описан алгоритм обучения адаптивной системы нейро-нечеткого вывода с использованием технологий параллельных вычислений [3].

На рисунке 3 показан график зависимости времени обучения ANFIS от количества итераций, в этом случае программа написал на языке MATLAB, характеристики сети одной и той же сети, обучающейся с использованием MEX-файла. Число объектов обучающей выборки 3000, число процессов постояно равно 6.

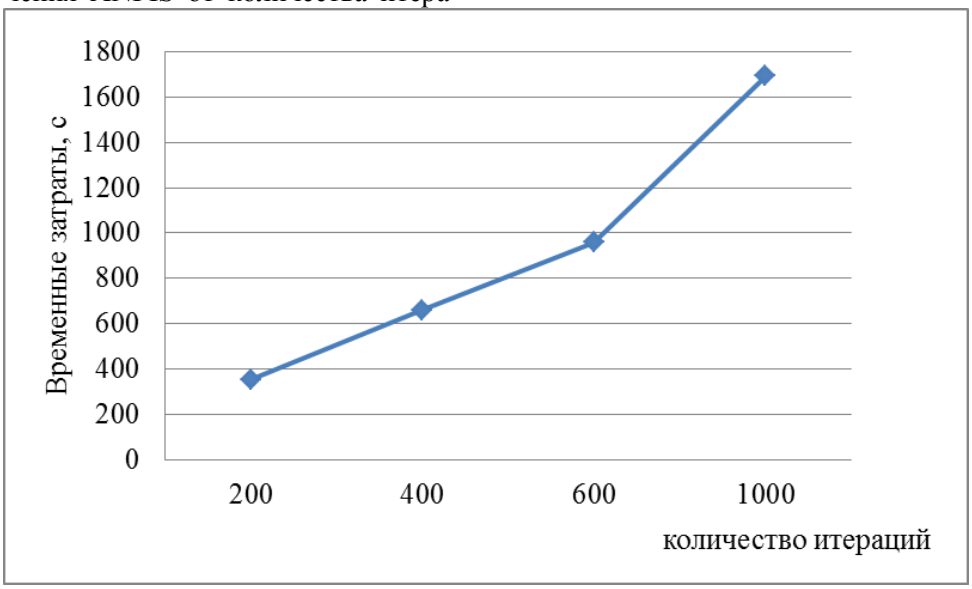

Рис. 3. Зависимость временных затрат от количества итераций

Из графиков видно, что алгоритм обучения ANFIS с использованием MEX-файлов эффективнее. При обучении на однойи той же сети на одном множестве данных в одинаковых условиях, программа, написанная с использованием MEX-файла, работает более в 60 раз быстрее, чем программа, написанная только средствами MATLAB.

Заключение

Рассмотрен способ создания MEX-файлов, написанных на языке программирования C++. Проведены эксперименты по применению созданного файла для реализации программы параллельного обучения гибридных сетей. На основе проделанных экспериментов можно сделать вывод, что использование MEX-файла является более эффективным, так как при обработке матриц программа на языке C исполняется во много раз быстрее аналогичной программы на языке MATLAB.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. MATLAB® Distributed Computing Server™ System Administrator's Guide.

2. Parallel Computing Toolbox™ User's Guide.

3. НгуенДангМинь. Технология распределенных и параллельных вычислений для повышения эффективности обучения адаптивной нейро-нечеткой сети // Вестник ИрГТУ. № 5. 2013. С. 12 – 16.

4. S. Suresh, S. N. Omkar and V. Mani. Parallel implementation of back-propagation algorithm in networks of workstations // IEEE Transactions on Parallel and Distributed Systems, pp. 24-34, 2005.

5. MATLAB User's Guide.

6. Написание расширений Matlab на языке C. URL: http://www.butovo.com/~zss/matlab/5/2.htm

7. Митрофанов С.П. Научная организация серийного производства. - Изд-во «Машиностроение», 1970. 768 стр.

8. Митрофанов С.П. Групповая технология изготовления заготовок серийного производства. Л.: Машиностроение, Лелингр. отд-ние, 1985. 240 с.