УДК 681.327.12.001.362

ДОРОГОЙ Я.Ю.,
ЯШИН В.Е.

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ СИМУЛЯЦИИ МНОГОПОТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Данная статья посвящена проблеме создания эффективного программного инструментария для симуляции нейронных сетей. Дан краткий обзор существующих программных решений, выявлен перечень недостатков рассмотренных средств. Описана архитектура разработанного многопоточного симулятора нейросетей, продемонстрирована его работоспособность.

This article deals with the problem of creating an effective software tool for simulating neural networks. A brief overview of existing software solutions was given, revealed a list of the deficiencies of discussed means. An Description of proposed multithread neural networks simulator architecture was given and results of its work.

Общая постановка проблемы

На сегодняшний день нейронные сети находят все большее применение в различных научных областях и промышленных отраслях. Для реализации готовых нейросетевых моделей уже выработаны определенные техники, такие как реализация с помощью элементов ПЛИС, DSP-процессоров, а также спеиализированных высокоскоростных нейропроцессоров. В большинстве этих аппаратных решений обученная выполнять определенную задачу нейросетевая система работает в параллельном режиме, т.е. затраты времени на выполнение задачи очень малы.

В это же время основной проблемой в моделировании и реализации нейросетей остается дизайн сети, выбор оптимальной архитектуры, а также обучение. Очень часто для решения определенной задачи приходится определять оптимальную конфигурацию и параметры нейросетевой системы методом проб и ошибок, проводя большое количество тестов и циклов обучения. Именно обучение нейронной сети на сегодняшний день является самым ресурсоемким и продолжительным по времени процессом.

Для проведения дальнейших исследований в области построения нейронных сетей нам был необходим хороший инструмент — набор программных средств, которые позволили бы с минимальными временными затратами создавать модели нейронных сетей, обучать нейронные сети и проводить их сравнительный анализ.

Обзор существующих решений

В ходе работы были рассмотрены некоторые из существующих ныне популярных программных решений поставленной проблемы. Среди них: Neural Network Toolbox от Matlab, пакет SNNS и библиотека Joone. Рассмотрим каждое решение подробней.

Пакет Matlab [1]. В состав программного комплекса Matlab входит популярный пакет neural network toolbox. Этот пакет позволяет достаточно быстро смоделировать нейронную сеть, провести ее обучение несколькими алгоритмами. Кроме того, в состав пакета Matlab входят и другие пакеты, которые позволяют проводить анализ, пре- и постобработку данных, подаваемых на вход нейронной сети и получаемых на ее выходе (например анализ главных компонентов(PCA), усреднение входных данных и др.). Но к сожалению, Neural Network Toolbox не дает требуемой гибкости. Он оперирует нейронной сетью, как набором слоев, что не позволяет смоделировать сложные типы сетей, не предусмотренные авторами пакета, такие как сверточные сети и неокогнитрон. Модификация пакета представляет слишком сложную задачу, т.к. требует значительного изменения большинства входящих в пакет модулей. Поэтому от использования этого пакета как основного средства моделирования и анализа нам пришлось отказаться.

Пакет SNNS. Пакет SNNS представляет собой мощную библиотеку, изначально разработанную в Штуттгардском университете. Ее основное назначение — анализ и моделирование нейросетей практически любой топологии. К сожалению, исходный код библиотеки SNNS является довольно сложным для моди-

фикации и не позволяет в приемлимые сроки выполнить его оптимизацию для современных аппаратных платформ, предоставляющих возможность параллельного выполнения нескольких процессов.

Библиотека Joone [2]. Библиотека Joone может использоваться для создания наглядных, простых моделей нейронных сетей, но библиотека спроектирована с допущением, что все нейронные сети состоят из слоев. Также предусмотрен только 1 алгоритм обучения.

Ограничения рассмотренных пакетов, а также их низкая производительность сыграли решающую роль в решении создать собственную реализацию библиотеки для симуляции нейронных сетей произвольной топологии.

Постановка задачи

Цель данной работы – разработка программного обеспечения для симуляции нейронных сетей с устраненными ограничениями, найденными в рассмотренных выше пакетах, а также повышение общей производительности обучения и тестирования нейронных сетей за счет использования многопоточной архитектуры.

Архитектура библиотеки

Библиотека нейросетевой симуляции PANN предполагает моделирование сети произвольной топологии. Пользователь сам задает количество входных нейронов, сам соединяет нейроны между собой. В получившейся структуре библиотека автоматически производит сегментацию на слои с использованием модифицированного волнового алгоритма, для дальнейшего разделения нейронов сети между рабочими потоками. Следует отметить, что архитектура сети не обязательно должна быть слоистой.

После подачи в сеть входного сигнала происходит его прямое распространение до выходных нейронов. Библиотека подразумевает, что в зависимости от алгоритма обучения, над каждым отдельно взятым нейроном проводятся различные действия. Объект, который выполняет над каждым нейроном необходимые действия передается сети как параметр. Таким образом, каждый алгоритм обучения или алгоритм прямого распространиния может сохранять в нейроне свой набор параметров, посвоему обрабатывать входы нейрона, вычислять значение функции активации и формировать выход. За счет этого достигается огромная гибкость комплекса, выигрыш в скорости.

Также в состав библиотеки входят модули для создания различных топологий нейронных сетей, алгоритмы обучения, модули формирования входных данных (могут подавать данные на вход сети как результат вычисления некоторой функции, либо после чтения набора данных или графического изображения из файлов на внешнем носителе).

Существует возможность сохранения существующей обученной нейронной сети на диск для последующей загрузки и продолжения обучения. Данные сохраняются в формате XML, в бинарном виде или как текстовый файл, что позволяет с легкостью переносить существующие модели нейронных сетей в другие программы.

Отдельно стоит рассмотреть вспомогательную модуль библиотеки – визуализатор топологии нейронной сети – **pann_viewer.** Довольно часто возникает необходимость визуализации созданной нейронной сети, особенно если ее части была сгенерирована автоматически. Для этого был разработан вспомогательный программный модуль, задачей которой является визуализация модели нейронной сети. Модуль автоматически визуализирует нейронную сеть из ранее сохраненного файла, позволяет вращать модель сети, масштабировать изображение и пр.

Для проверки работоспособности разработанной библиотеки было проведено множество различных экспериментов. Опишем два из них.

Компьютерный эксперимент 1. Для демонстрации возможностей созданного программного комплекса, был проведен ряд симуляций. Использованная в данном опыте нейронная сеть – многослойный персептрон с конфигурацией 1 – 9 – 4 – 1. Для скрытых нейронов использовалась модифицированная функция гиперболичес-кого тангенса: y = 1.7179*tanh(2/3 * x). Были сгенерированы 2 набора данных — для функций y = x^2, на интервале [-1;+1] и y = sin(x), на интервале [-3;+3]. Каждый набор данных включал 20 точек. Алгоритм обучения – метод обратного распространения ошибки. На рисунке 1 показана 3D модель нейронной сети эксперимента 1 (построена при помощи модуль визуализации pann_viewer):

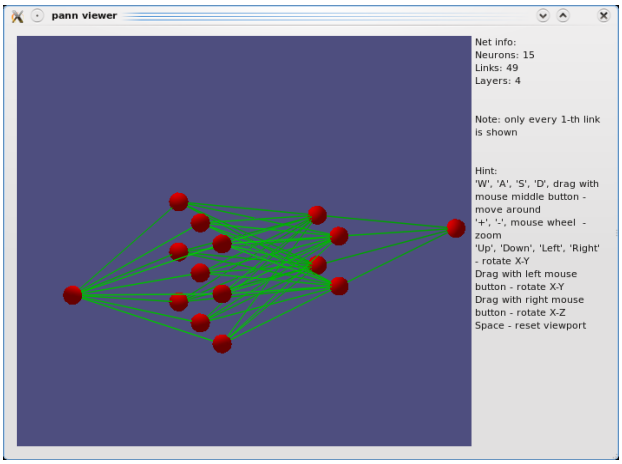

Рис. 1. 3D модель нейронной сети эксперимента 1

На рисунках 2 и 3 показаны графики распределения ошибки для двух функций (по оси абсцис указан номер эпохи обучения, по оси ординат – абсолютное значение усредненной среднеквадратической ошибки нейронной сети за данную эпоху):

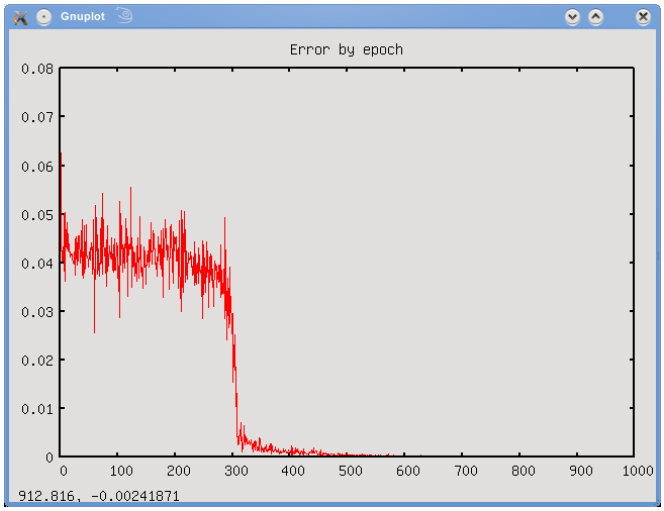

Рис. 2. Распределение ошибки обучения сети для функции y=x2 на интервале [-1;+1]

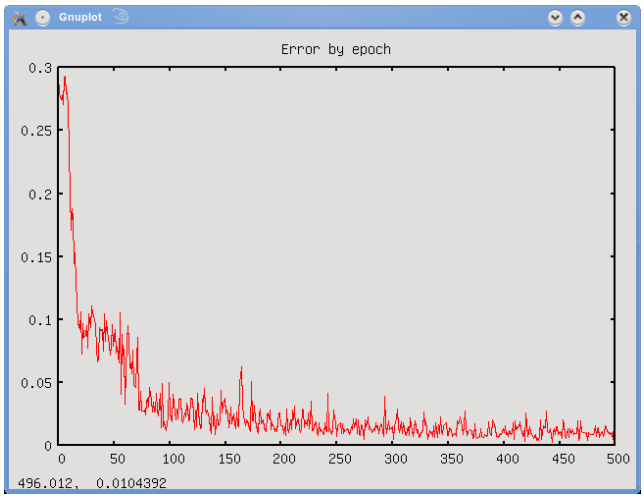

Рис. 3. Распределение ошибки обучения сети для функции y=sin(x) на интервале [-3;+3]

На рисунках 4 и 5 представлены результаты тестирования обученной нейронной сети:

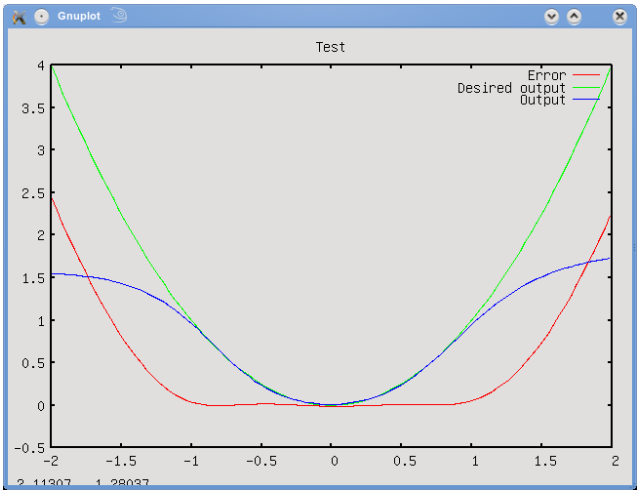

Рис. 4. Тестирование обученной нейронной сети для функции y=x² на интервале [-1;+1]

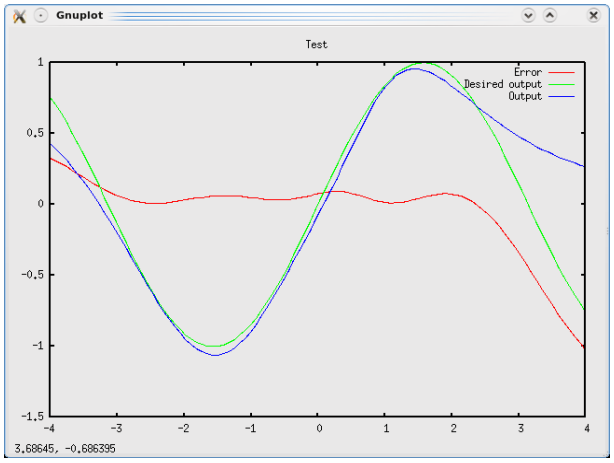

Рис. 5. Тестирование обученной нейронной сети для функции y=sin(x) на интервале [-3;+3]

Таким образом, обученная на примерах нейронная сеть демонстрирует правильную работу используемых алгоритмов обучения.

Компьютерный эксперимент 2. Далее приведены результаты нескольких симуляций, выполненных на многопроцессорной системе с различным количеством используемых рабочих потоков. Эти тесты позволяют количественно оценить преимущество, полученное в результате оптимизации библиотеки для работы в режиме нескольких потоков.

Используемая для симуляции нейроная сеть – многослойный персептрон с конфигурацией 1 – 1000 – 1000 – 1000 – 1. Проводились замеры времени обучения такой нейронной сети для задачи предсказания значения функции на заданном интервале с использованием от 1 до 8 параллельно выполняемых потоков команд.

Для тестов использовалась машины с 8 процессорами с тактовой частотой 2.0ГГц (машина 1) и 4 процессорами с тактовой частотой 3.2 ГГц (машина 2). В ходе эксперимента было проведено 10 независимых тестов. Усредненные данные проведенных тестов представлены в таблице 1 и на рисунке 6.

Табл. 1. Результаты эксперимента (время в с)

Кол-во потоков	1	2	3	4	5	6	7	8
Машина 1	89.8	67.9	55.7	50.8	46.7	45.9	47.4	49.2
Машина 2	68.6	40.9	36.5	33.8	35.2	36.0	34.2	34.4

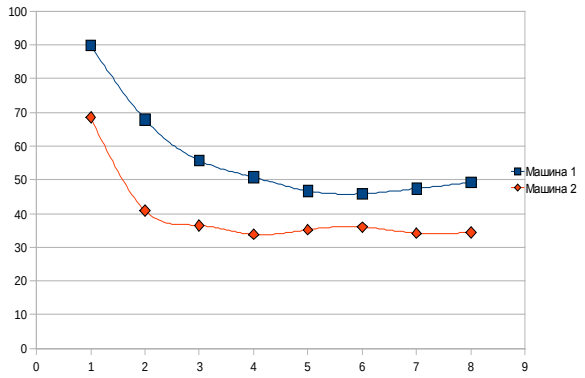

Рис. 6. Время обучения нейронной сети в зависимости от количества потоков

По полученным данным можно сделать следующие выводы: увеличение количества рабочих потоков не дает огромного прироста производительности. Реальное значение прироста сильно зависит от сложности алгоритма обучения. На получаемое преимущество в скорости влияют задержки, связанные с синхронизацией потоков друг с другом, что является неизбежным явлением при многопоточном программировании. Также заметно, что при количестве потоков превышающем реальное количество процессоров, задержки вносимые операционной системой при переключении между несколькими потоками одного процессора становятся достаточно велики, в то время как из-за затрат на синхронизацию потоков отсутствует какой-либо выигрыш.

Резюмируя все вышесказанное, можно сказать, что использование нескольких параллельных потоков действительно дает преимущество по производительности для больших нейронных сетей и довольно сложных алгоритмов обучения. Полученные результаты позволяют рекомендовать разработанное ПО для симуляции нейронных сетей для использования в задачах связанных с распознаванием образов и компьютерным зрением.

Выводы

В процессе работы над библиотекой нейросетевой симуляции был создан программный комплекс для моделирования нейронных сетей произвольной топологии. Спроектированный комплекс обладает следующими отличительными особенностями:

– высокий уровень абстракции (за счет использования объектно-ориентированного языка программирования), что позволяет добавлять или менять функционал библиотеки за короткое время, с минимальной переработкой существующего кода;

– высокий уровень модульности – все алгоритмы обучения, функции активации, модули нейросетевого прототипирования – являются модулями. Таким образом можно легко добавлять новые нейросетевые модели, алгоритмы обучения и т.д., используя существующий простой, но эффективный интерфейс взаимодействия с другими модулями;

– ориентация на современные процессорные архитектуры(SMP). Вся обработка данных выполняется параллельно в несколько потоков, в зависимости от количества присутствующих в системе процессоров. Взаимодействие между отдельными потоками сведено к минимуму для устрания эффекта «гонок» за общие ресурсы, и как следствие – повышение производительности. Данное решение позволяет получить значительный прирост в

скорости при тестировании и обучении нейронных сетей;

– независимость от топологии – возможность моделирвания сетей, связи в которых распределены любым образом, т.е. архитектура сети не обязательно предполагает наличие слоев. Также поддерживаются рекуррентные связи.

Разработанное ПО позволяет выполнить быстрое моделирование обучение и анализ полученных результатов, с задействованием современных многопроцессорных архитектур. Этот инструментарий в дальнейшем будет пополняться новыми нейросетевыми моделями, алгоритмами обучения и др., а его гибкость и расширяемость позволит провести ряд важных исследований архитектур нейронных сетей.

Дальнейшие перспективы развития данного ПО

В дальнейшем предполагается реализация моделей сверточной нейронной сети и компактной ячеистой нейронной сети [3,4] для исследований в области распознавания человека.

Список литературы

1. Hyun-Yeop Lee, Kyung-Il Moon. Fuzzy Logic and Neural Networks Using MATLAB. 2008.
2. Jeff Heaton. Using Joone for Artificial Intelligence Programming. Журнал Developer.com. 2008. – Vol. 2. – P.40-46.
3. Дорогой Я.Ю. Компактные ячеистые сверточные нейронные сети для локализации лица человека.//Збірник тезисів міжнародної науково-практичної конференції «Обробка сигналів і негауссівських процесів», Черкаси, 2007.
4. Дорогой Я.Ю. Применение компактных ячеистых сверточных нейронных сетей для биометрической идентификации человека по лицу.//Вісник НТУУ «КПІ», «Інформатика, управління та обчислювальна техніка», №46. – 2007. – С. 135-149.