Программно-аппаратные эмуляторы нейронных сетей на процессоре PowerXCell™ 8i

И.И. Зиновьев

Владимирский государственный университет
izinoviev@vlsu.ru

Сейчас искусственные нейронные сети (ИНС) широко используются как средство интеллектуального анализа данных. Учитывая высокие темпы роста объемов накопленной в современных хранилищах данных информации, роль такого анализа трудно переоценить. Нейросетевые технологии предоставляют сегодня широкие возможности для решения задач прогнозирования, обработки сигналов и распознавания образов. По сравнению с традиционными методами математической статистики, классификации и аппроксимации, эти технологии обеспечивают достаточно высокое качество решений при меньших затратах.

Поэтому актуальной является задача эффективной реализации ИНС. В настоящее время реализация нейросетевых алгоритмов осуществляется программной, программно-аппаратной эмуляцией и чисто аппаратными средствами. Программная реализация представляет собой последовательную программу для компьютеров. Она полезна в процессе знакомства и обучения работе с ИНС, где не требуется высокое быстродействие. Чисто аппаратными средствами реализации ИНС являются многопроцессорные нейроподобные СБИС и нейрочипы. Аппаратная реализация ИНС алгоритмов является самым быстродействующим, но и дорогостоящим вариантом. На сегодня достаточно универсальным и эффективным способом построения масштабируемых конфигураций нейрокомпьютеров является программно-аппаратные эмуляции, выполняемые на базе компьютеров с широкомасштабным параллелизмом.

Искусственные нейронные сети представляют собой устройства параллельных вычислений, состоящие из множества взаимодействующих простых процессоров. Но, несмотря на то, что функционирование ИНС предполагает распараллеливание процесса вычислений, существуют общеизвестные проблемы при их программной реализации на высокопроизводительных вычислительных системах.

Во-первых, как было сказано выше ИНС – сеть простых процессоров и все ее свойства проявляются в результате их взаимодействия. Следовательно, при реализации нейросетевых алгоритмов не избежать массовой пересылки информации между узлами нейронной сети. Это приводит к тому, что узким местом в программе становятся именно каналы передачи данных.

Во-вторых, традиционные алгоритмы обучения нейронных сетей (например, алгоритм обратного распространения ошибки) либо принципиально не распараллеливаются, либо не достаточно эффективны. Поэтому в [1] при создании параллельных программ-нейроимитаторов предлагается в первую очередь распараллеливать не обучение единичной нейросети, а сервисные процедуры: методы оптимизации предобработки данных, схемы отбора примеров в обучающую выборку, и т.д..

Для преодоления первой проблемы в настоящее время возможно использование новых архитектур, которые позволяют обойти ограничения пропускной способности памяти. Наибольший интерес представляют современные графические процессоры и многоядерные микропроцессоры типа CELL [2].

Что касается второй проблемы, то широкое применение находят новые алгоритмы обучения ИНС, такие как генетические или другие рандомизированные алгоритмы оптимизации. Данный класс алгоритмов является ресурсоемким, но в тоже время легко поддающимся распараллеливанию.

Рассмотрим возможность и эффективность программной реализации ИНС на базе процессора PowerXCell 8i. Для начала необходимо определить основные этапы функционирования ИНС:

1. Выбор архитектуры ИНС;
2. Подбор и подготовка обучающих данных;
3. Обучение ИНС;
4. Работа обученной ИНС.

Реализация каждого этапа влияет на эффективность и скорость функционирования ИНС. Обратим особое внимание на последние два этапа, как традиционно не распараллеливаемые и наиболее ресурсоемкие. Эти два этапа являются взаимосвязанными, т.к. любой алгоритм обучения основан на сравнении полученных значений в результате работы ИНС с эталонными. Использование генетических алгоритмов обучения ИНС предполагает создание исходной популяции, оценку приспособленности для каждой особи текущей популяции, применение к выбранным особям генетических операторов и создание новой популяции. Все эти процессы могут быть реализованы одновременно на нескольких процессорах, что пропорционально повысит скорость работы алгоритма.

Но такой подход не применим при непосредственной работе обученной ИНС, т.к. мы имеем всего один экземпляр нейронной сети. Поэтому необходимо искать другие способы повышения быстродействия. Рассмотрим ряд архитектурных особенностей процессора PowerXCell 8i, применимых при программной реализации ИНС.

Во-первых, процессор PowerXCell 8i состоит из одного процессора элемента PPE и 8 отдельных специализированных процессорных элементов SPE, оптимизированных для арифметических операций над вещественными числами с одинарной точностью. Каждый из них имеет полный доступ к общей памяти. Следовательно, существует аппаратная возможность распараллеливания нейронной сети.

Во-вторых, процесс функционирования ИНС является векторным, поэтому, благодаря поддержке процессором набора векторных инструкций, самый очевидный способ его оптимизации - это векторизация кода, также известная как SIMD (single instruction, multiple data – один поток команд - много потоков данных), или информационный параллелизм. В процессорах SPU большинство инструкций могут оперировать регистрами так, как если бы они содержали несколько независимых значений (таким образом, одна инструкция действует на несколько элементов данных). Каждый 128-разрядный регистр может рассматриваться как 16 независимых байтов, 8 полуслов, 4 слова, 2 двойных слова, или как один элемент. Так что, если векторизовать код, рассматривая значения как 32-разрядные слова, то это будет означать, что каждая инструкция будет оперировать одновременно четырьмя значениями.

В-третьих, процессор имеет уникальную трехуровневую архитектуру памяти с асинхронным доступом и поддержкой до 16 одновременных обращений. Это делает возможным применение двойной или мульти буферизации для загрузки данных из общей памяти в локальную одновременно с выполнением вычислений. Следовательно, минимизируются простои программы при пересылке данных, что особенно актуально для программ-имитаторов нейронных сетей.

И, наконец, процессор SPU не только может обрабатывать через свой конвейер несколько значений одновременно, но также поддерживает инструкции dual-issue режима, которые обрабатываются через различные конвейеры. SPU имеет два конвейера – четный (иногда его называют конвейер 0 или конвейер выполнения), и нечетный (конвейер 1 или конвейер загрузки). Фактически SPU загружает одновременно две инструкции из границ, выровненных до двойного слова. Это называется групповой операцией. Если первая инструкция групповой операции выполняется на четном конвейере, а вторая инструкция – на нечетном, то обе этих инструкции могут быть выполнены

одновременно. При правильном подборе инструкций это дает дополнительное увеличение скорости выполнения любой программы, в том числе и программы-нейроимитатора.

Правильное применение вышеизложенных особенностей архитектуры процессора PowerXCell 8i делает его привлекательной аппаратной платформой для программной реализации ИНС.

Литература

1. Царегородцев В.Г. Перспективы распараллеливания программ нейросетевого анализа и обработки данных. // Материалы III Всеросс. конф. "Математика, информатика, управление - 2004". Иркутск, 2004.
2. Аляутдинов М.А., Галушкин А.И., Троепольская Г.В. Программно-аппаратные эмуляторы нейронных сетей на современных графических процессорах. // «Искусственный интеллект» 3'2006
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. – М.: Горячая линия – Телеком, 2006
4. Федорова Н.Н., Терехов С.А. Параллельная реализация алгоритмов обучения нейронных сетей прямого распространения с использованием стандарта MPI. // Российский Федеральный Ядерный Центр – Всероссийский НИИ Технической Физики
5. IBM SDK for Multicore Acceleration v 3.1, Programming Tutorial.