ПРИМЕНЕНИЕ CUDA-ТЕХНОЛОГИИ ДЛЯ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Поступила в редакцию 15 марта 2011

В работе описывается использование CUDA-технологии для обучения искусственных нейронных сетей с учителем на основе обратного распространения ошибки. Рассматриваются вопросы перемещения данных между видеопамятью и оперативной памятью, выделения памяти на устройстве, влияния версии драйверов на производительность.
Ключевые слова: параллельные вычисления, обучение искусственных нейронных сетей, CUDA-технология от Nvidia, вычисления на GPU.

Введение

В настоящее время CUDA-технология применяется в искусственных нейронных сетях (ИНС) для решения широкого спектра задач из разных предметных областей [1].

Среди основных недостатков существующих реализаций можно выделить следующие:

– не многие разработки имеют открытые исходные коды или свободно выложены для безвозмездного пользования, а коммерческие разработки требуют покупки лицензии, что весьма затратно для отечественных научных проектов;

– структура нейронных сетей не является гибкой, т.е. данные ИНС не могут быть адаптированы для решения иных задач, так как нельзя изменить количество нейронов и слоев без изменения всей логики проекта;

– модули создаются под конкретное устройство, при этом не учитываются более широкие аппаратные возможности других видеокарт;

– зачастую выигрыш в производительности теряется из-за медленного обмена данными между оперативной памятью и видеокартой.

Однако применение данной технологии является эффективным в определенных классах искусственных нейронных сетей, так как позволяет существенно ускорить их обучение. Как было показано в [1] уже на этапе прямого прохода выигрыш в производительности достигается в сетях с количеством нейронов в слоях более 32.

Целью данного проекта являлось создание гибкого программного модуля, позволяющего обучать на основе алгоритма обратного распространения ошибки (back propagation) [2] и использовать в режиме прогнозирования ИНС с произвольным количеством слоев и нейронов в них, применяя различные функции активации.

Среди основных задач проекта можно выделить следующие:

– свести к минимуму обмен данными между оперативной памятью и видеокартой;

– выявить закономерности выделения памяти на видеокарте под необходимые параметры;

– выявить пики производительности (т.е. топологии ИНС), при которых достигается максимальный выигрыш во времени обучения на устройствах – типичных представителях линеек продуктов;

– подобрать оптимальные конфигурации ядер для наиболее быстрого обучения ИНС на указанных выше устройствах.

Применение CUDA-технологии для обучения искусственных нейронных сетей

Механизм распараллеливания обучения ИНС на основе алгоритма обратного распространения ошибки основан на пошаговом расчете параметров всех слоев сети, так как, не зная выходных параметров предыдущего слоя (в режиме прогнозирования) или последующего (при обучении), невозможно вычислить параметры текущего слоя. Поэтому существуют следующие варианты распараллеливания данного алгоритма: 1) одновременное обучение нескольких ИНС с последующим выбором наиболее оптимальной из них по критерию

минимальной ошибки; 2) одновременный расчет параметров нескольких нейронов одного слоя. При реализации указанных выше задач был выбран второй способ, так как он требует выделения на устройстве (видеокарте) меньшего количества памяти, чем первый, что позволяет применять его для ИНС с большим количеством нейронов по слоям.

Обучение ИНС может быть представлено в виде трех этапов [3]: 1) прямой проход (вычисление выходов каждого слоя, начиная с первого); 2) вычисление ошибки слоев, начиная с последнего; 3) вычисление корректирующих значений для матриц коэффициентов, векторов порогов и изменение текущих матриц весов и векторов порогов с учетом рассчитанных значений.

Из-за аппаратных особенностей видеокарт количество нейронов в слое должно быть кратно 2^n, где $n = \{1, \ldots, 5\}$, соответственно для сетей с количеством нейронов по слоям больше 8 должно быть кратно 16 [3], для чего матрицы весов, вектора входов, выходов и порогов выравниваются (заполняются нулями) до выбранных размерностей. Схема взаимодействия внешнего и разработанного (в виде библиотеки dll) модулей представлена на рис. 1.

Исходные данные по состоянию сети (W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i для каждого слоя) формируются внешним модулем. Если сеть обучается впервые, то W_i и T_i заполняются случайными числами в диапазоне (0; 1) или используются иные алгоритмы (например, генетические). ΔW_i, ΔT_i, E_i, Y_i заполняются нулями. Если сеть дообучается, то используются ранее сохраненные значения W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i.

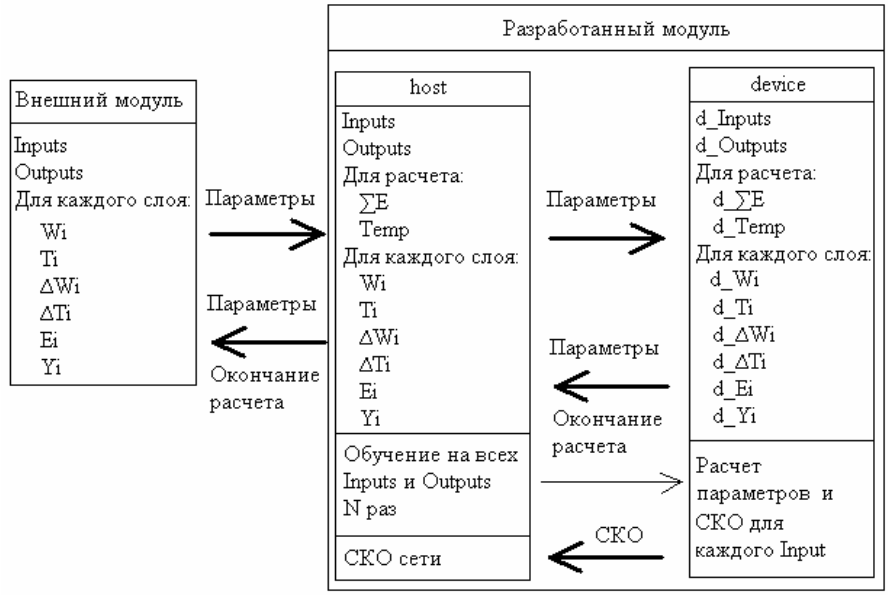

Рис. 1. Схема взаимодействия внешнего и разработанного модулей

Указанные параметры передаются по ссылке на управляющий модуль (host). Под них выделяется память на устройстве (device). Также выделяется память для массивов, необходимых при промежуточных расчетах. Наборы обучающих данных (Inputs, Outputs), входящих одну эпоху, передаются в начале обучения с внешнего модуля на управляющий, а затем на устройство.

Расчет включает в себя последовательное выполнение прямого и обратного проходов для каждого набора входных и выходных данных (Inputs, Outputs), составляющих одну эпоху. При этом на управляющий модуль с устройства копируется СКО (среднеквадратическое отклонение) выходного слоя, используемое при вычислении суммарного СКО каждой эпохи, записываемого в файл. По СКО эпохи можно судить о том, насколько хорошо в данный момент обучена сеть. По окончании обучения (достижения определенного уровня СКО или выполнения заданного количества итераций) устройство возвращает данные по состоянию сети W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i для каждого слоя обратно управляющему модулю, а тот в свою очередь – внешнему модулю.

Таким образом, нет необходимости постоянного перемещения параметров, необходимых для обучения, между устройством и управляющем модулем. Однажды получив их, устройство само обучает сеть, периодически возвращая только СКО для обеспечения возможности отслеживания текущего состояния ИНС. Также при необходимости W_i и T_i для каждого слоя

могут быть переданы для последующего использования во внешних графических модулях, визуализирующих текущее состояние ИНС (см. рис. 2). Однако это увеличивает время обучения. Графическое отображение целесообразно использовать на этапе определения топологии сети, когда визуализация параметров позволяет исключить или дополнить сеть нейронами или слоями.

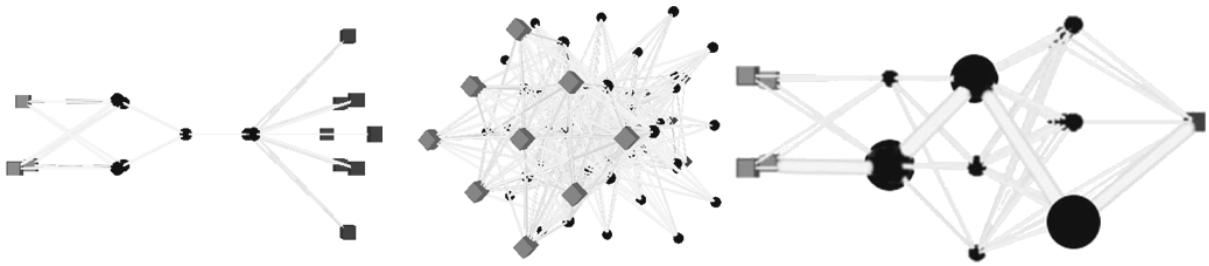

Рис. 2. Отображение текущего состояния ИНС

Особенности выделения памяти

Память под элементы на видеокарте выделяется не вся. При полном объеме памяти в 256 Мб выделяется только 230 Мб (242077696 байт), что составляет 90,2%, остальная память резервируется. При объеме памяти в 512 Мб выделяется 473 Мб (496812032 байт – 92,5%), при объеме 1024 Мб – 951 Мб (997650432 байт– 92,9%). Причем при расчете небольших массивов (например, 512 элементов) память выделяется из резерва.

Детально исследовалась функция, выделяющая память под 3 вектора (размерностью n) и 3 матрицы (n x n). На вектора глобальная память начинает выделяться при $n \geq 1024$, причем на первый вектор выделяется необходимое количество памяти, на второй - кратно 64 Кб (65536 байт).

С матрицами до (1024 x 1024) ситуация аналогичная. Однако при больших размерах массива (1024 x 1024 и выше) на первую матрицу выделяется памяти больше, чем требуется (уравнение, полученное при помощи MS Excel: y = 4 194 304,00x2 – 4 096,00x + 65 536,00, $R^2 = 1,00$, где x – № пункта из таблицы 1, из чего была получена формула (1) для выражения через размерность матрицы):

$$Memo = (2\sqrt{n} - 1)^2 + 2^{16} - 1 , \quad (1)$$

где Memo – количество выделяемой под матрицу памяти (байт), n – количество элементов в матрице. Зависимость была выведена по экспериментальным данным для матриц, приведенных таблице 1.

На остальные матрицы (кроме первой) памяти выделяется ровно столько, сколько нужно.

Таблица 1. **Выделение памяти для размещения в ней матрицы весов нейронов**

№ п.	Размерность матрицы	Количество элементов	Необходимая память, байт (столб.2 *4)	Выделяемая память CUDA, байт	Память по формуле (1), байт	Отношение столб.3/столб4
1	1024 x 1024	1048576	4255744	4194304	4194304	1,01468
2	2048 x 2048	4194304	16834560	16777216	16777216	1,00341
3	3072 x 3072	9437184	37801984	37748736	37748736	1,00141
4	4096 x 4096	16777216	67158016	67108864	67108864	1,00073
5	5120 x 5120	26214400	104902656	104857600	104857600	1,00042

Ограничения на размещение данных в памяти устройства

Также было определено, сколько наборов тренировочных данных может быть размещено в памяти устройства для обучения небольших сетей и выведены формулы, позволяющие оценить максимальные топологи ИНС, удовлетворяющие указанным ниже условиям, которые могут быть обучены на данном устройстве в зависимости от объема его оперативной памяти при использовании описанного подхода:

а) количество скрытых слоев, включая выходной, равно двум;

б) количество нейронов в первом скрытом слое равно $2N_0+1$, где N_0 – количество входных нейронов (что следует из теоремы Колмогорова [4]). Но так как перед началом работы входные данные должны быть нормированы и выровнены [1], [3], то входных нейронов должно быть $N_0' = 16n$ (n – натуральное число), а количество нейронов в первом скрытом слое равно $N_1' = 2N_0'$ (так как $N_0 < N_0'$ для всех N_0, некратных 16, и $2N_0+1 < 2N_0'$). Для второгог случая, когда N_0 кратно 16, $N_1' = 2N_0'+16$;

в) количество выходных нейронов произвольно $N_2' = kN_0$ ($k > 0$) и должно быть выровнено.

Стоит отметить, что для размещения указанных данных на устройстве выделяется чуть больше памяти, чем требуется. Причем для тестируемых топологий ИНС на видеокарте 9500 GT (с размером глобальной памяти 256 Мб) это значение всегда являлось кратным 1 Мб, что хорошо видно на рис. 3. Для карты 9800 GT выделяемая память была кратна ¼ Мб.

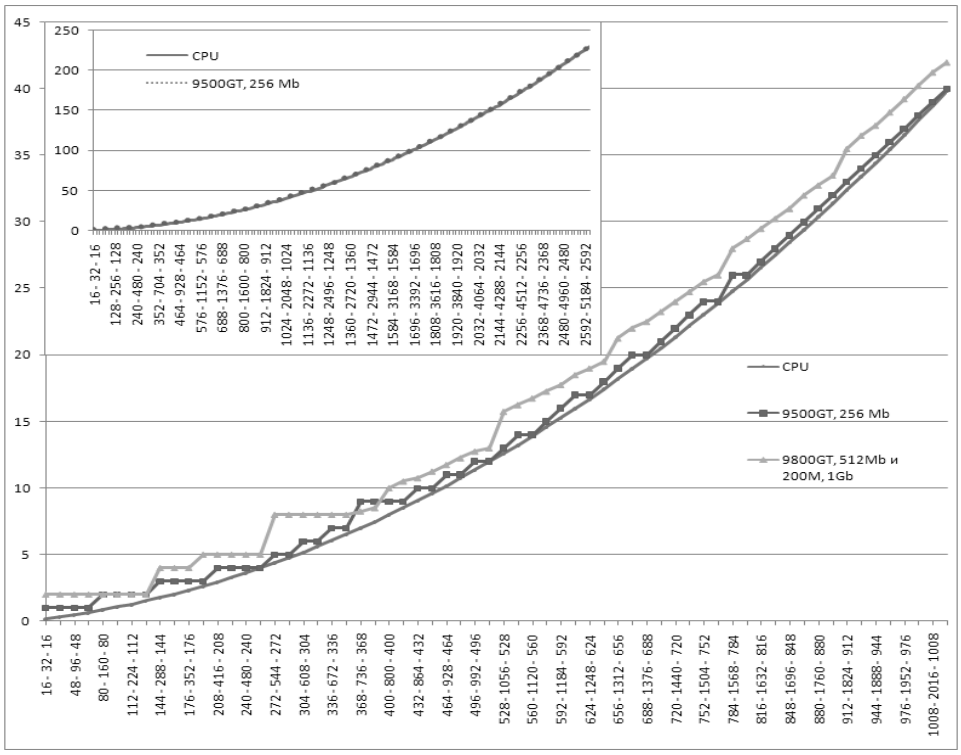

Рис. 3. Выделение памяти под параметры ИНС различных топологий

На рис. 3 представлен график зависимости выделяемой памяти в Мб на устройстве (9800 GT и 220 M) и необходимой памяти, рассчитанной теоретически (CPU) по формуле (2), для размещения всех элементов данных выровненной двухслойной ИНС с разным количеством нейронов в слоях. На нем хорошо видно «ступенчатое» выделение памяти на видеокарте.

Были получены четыре уравнения (формулы (2) – (5)), позволяющие оценить максимальное количество нейронов (N) в выровненных двухслойных ИНС, которые возможно обучать на различных видеокартах в зависимости от объема их оперативной памяти.

Пусть $N_0' = N$, Epoh – количество наборов входных данных, составляющих одну эпоху обучения, Memo – необходимое количество памяти в байтах для размещения всех элементов данных, тогда для N_0, некратных 16 – (2) и (3) для $k \le 2$ и $k \ge 2$ соответственно, для N_0, кратных 16, – (4) для $k \le 2$ и (5) для $k \ge 2$:

$$16(k+1)N^2 + (40+16{,}25k + 4Epoch(k+1))N + 256 - Memo = 0, \quad (2)$$

$$16(k+1)N^2 + (32+20{,}25k + 4Epoch(k+1))N + 256 - Memo = 0, \quad (3)$$

$$16(k+1)N^2 + (160+148{,}25k + 4Epoch(k+1))N + 256 - Memo = 0, \quad (4)$$

$$16(k+1)N^2 + (168+144{,}25k + 4Epoch(k+1))N + 256 - Memo = 0. \quad (5)$$

К примеру, из формулы (2) при решении квадратного уравнения относительно N следует, что при размере эпохи в 1000 входных наборов и количестве выходов, равном количеству входов, на устройстве с доступной памятью 230 Мб (242077696 байт) можно обучить сеть с топологией (2624 – 5248 – 2624), так как N = 2627,5. Но из-за особенностей выделения памяти на устройстве на практике при данных условиях можно разместить в памяти сеть (2608 – 5216 – 2608).

Анализ полученных результатов

Отличительной особенностью разработанного модуля является возможность его использования из любых проектов, поддерживаемых Visual Studio 2008. Причем время, затрачиваемое на обмен данными между основным модулем с данными и управляющим модулем (host), не зависит от языка программирования, на котором написан основной код.

Тестирование приводилось на ЭВМ, оснащенной 4-ядерным процессором Intel Core 2 Duo (по 2,86 ГГц) и видеокартой GeForce 9500 GT, а также для видеокарт GeForce 9800 GT и GeForce 220 M GT, параметры которых приведены в табл. 2.

Таблица 2. **Параметры тестируемых устройств**

Модель **GeForce**	Дата выхода на рынок	Объём видеопамяти (Мб)	Конфигурация ядра *	частоты			Пиковая скорость заполнения		Память			Теоретическая производительность (Гигафлопс)
				Ядро,(MHz)	шейдерный блок, (MHz)	память (MHz)	гигатекселей/с	гигатекселей/с	Пропускная способность (Гб/с)	Тип	Шина (бит)	
220M GT	16.08 2009	1024	32: 16:08	500	1250	1000 1600	4	8	16	DDR2	128	120
9500 GT	29.07 2008	256	32: 16:08	550	1400	1000 1600	4,4	8,8	16 25,6	DDR2 GDDR3	128	134
9800 GT	08 2008	512	112: 56:16	600	1500	1800	9,6	33,6	57,6	GDDR3	256	504

* Унифицированных шейдерных процессоров : Текстурных блоков : Блоков растеризации

Важную роль в быстродействии работы CUDA-совместимого устройства играет версия драйверов, обеспечивающих программный доступ к нему (см. табл. 3).

В табл.3 приведены средние замеры времени в миллисекундах, затраченного на обучение одного скрытого слоя за одну эпоху на одном и том же устройстве, но с разной версией драйверов (2.3 и 3.0). Количество входных сигналов для каждого нейрона равно количеству нейронов в рассматриваемом слое (xxx–N_{i-1}–N_i–xxx), обучающая выборка включала 1000 наборов входных данных. Прирост производительности версии 3.0 по сравнению с версией 2.2 составляет около 9%.

На рис. 4 и рис. 5 показаны графики затраченного времени в секундах на обучение выровненной ИНС с двумя скрытыми слоями (N-2N-N) на 10 и 1000 наборах данных соответственно за 10 итераций.

Таблица 3. **Время обучения одного слоя ИНС на видеокартах с разными версиями драйверов**

N_i=N_{i-1}	Версия драйверов		Прирост, %	N_i=N_{i-1}	Версия драйверов		Прирост, %
	2.3, мс	3.0, мс			2.3, мс	3.0, мс	
32	140	110	27	288	2859	2656	7,6
64	172	157	9,6	320	3813	3531	8
96	266	250	6,4	352	4954	4563	8,6

128	437	390	12	384	6281	5781	8,6
160	688	625	10	416	7875	7250	8,6
192	1000	937	6,7	448	9688	8922	8,6
224	1515	1391	8,9	480	11844	10859	9,1
256	2094	1938	8	512	14094	12969	8,7

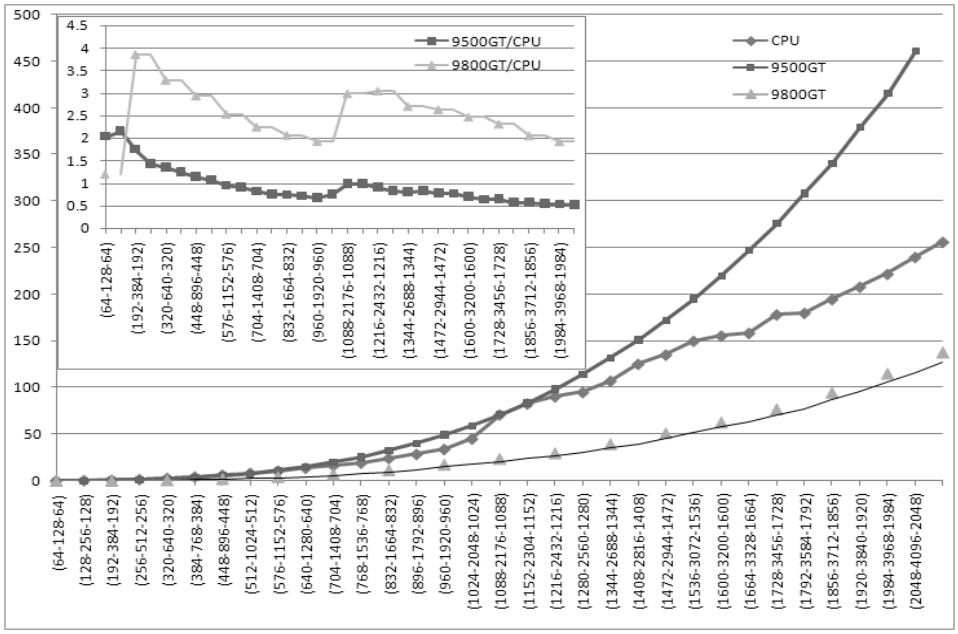

Рис. 4. Затраченное время на обучение ИНС с двумя скрытыми слоями (эпоха из 10 наборов)

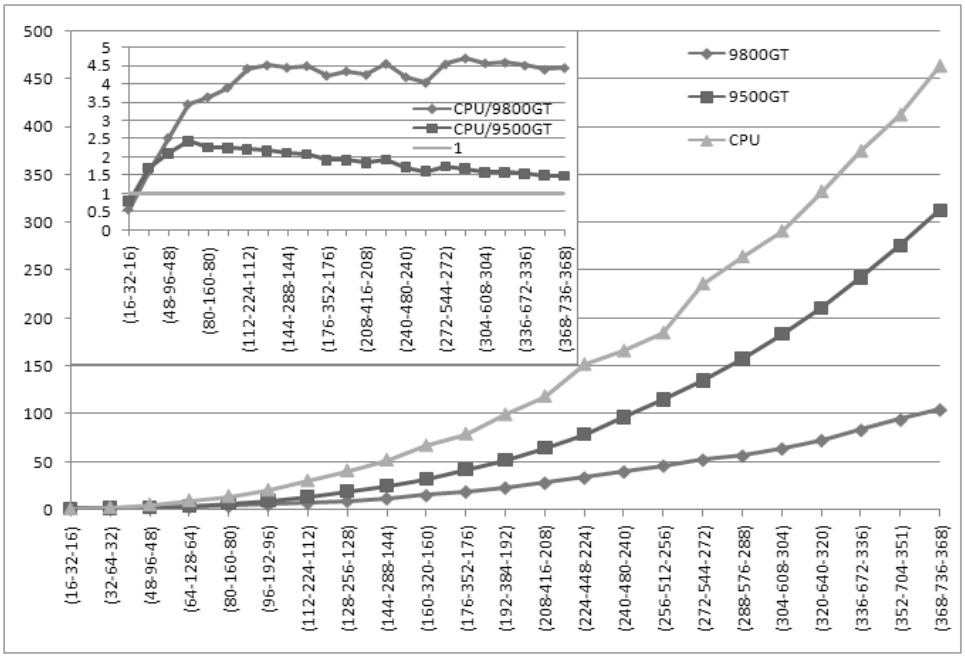

Рис. 5. Затраченное время на обучение ИНС с двумя скрытыми слоями (эпоха из 1000 наборов)

На дополнительных графиках (см. рис. 4 и рис. 5) показано, во сколько раз быстрее обучается ИНС указанной на оси абсцисс топологии на видеокартах GeForce 9800 GT и 9500 GT по сравнению с CPU.

В результате, используя CUDA-технологию, удалось ускорить обучение ИНС по сравнению с CPU до 5 раз на видеокарте 9800 GT и до 2,5 раз на 9500 GT. Время обучения включает в себя время работы с файлами и перемещение данных между host и device. Выигрыш наблюдается уже в сетях (32-64-32). При обучении на 1000 наборах данных наибольший выигрыш

на 9500GT достигается при обучении ИНС (64-128-64), а на 9800GT – при обучении двухслойной ИНС (288-576-288).

Заключение

В результате проведенных исследований установлено, что использование CUDA-технологии ускоряет обучение искусственных нейронных сетей с большим количеством нейронов в слоях в до 5 раз. Производительность резко возрастает при небольшом количестве нейронов по слоям, а затем, достигнув пика, постепенно снижается. Максимальный выигрыш в производительность зависит от конфигурации и частоты ядра, объема и пропускной способности видеопамяти, а также версии установленных драйверов для устройства.

CUDA-TECNOLOGY APPLICATION FOR ANN TRAINING

Abstract

CUDA-technology application for artificial neural networks (ANN) training based on back propagation is described. The problems of data exchange between host and device is discussed. It is shown how drivers' version influences the performance. Memory allocation on device is described.

Using of CUDA-technology accelerates training of ANN with a large number of neurons in layers up to 5 times (on Nvidia GeForce 9800 GT).

Литература

1. *Хилько О.С., Коваленко В.И.* Применение CUDA-технологии для параллельных вычислений с использованием искусственных нейронных сетей // Доклады БГУИР. 2010. №7 (53). С. 83-88.
2. *Хайкин Саймон*. Нейронные сети. Полный курс. СПб : Вильямс, 2006.
3. *Хилько О.С., Коваленко В.И.* Применение CUDA-технологии в искусственных нейронных сетях для решения экологических задач // Сахаровские чтения 2010 года: экологические проблемы XXI века: Мат-лы 10-й междунар. научной конференции, Мн: 2010. С. 87.
4. *Колмогоров А.Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного // Докл. АН СССР. 1957. Т. 114. № 5. С. 953-956.