УДК 004.38: 004.8.032.26

ПРИМЕНЕНИЕ CUDA-ТЕХНОЛОГИИ ДЛЯ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

О.С. ХИЛЬКО

Международный государственный экологический университет им. А.Д. Сахарова
Долгобродская, 23, Минск, 220009, Беларусь

Поступила в редакцию 11 апреля 2011

Описывается использование CUDA-технологии для обучения искусственных нейронных сетей с учителем на основе алгортима обратного распространения ошибки. Освещены вопросы перемещения данных между видеопамятью и оперативной памятью, выделения памяти на видеокарте, 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 задач из разных предметных областей: физики, экономики, экологии и др. [1].

Описанные в [1] программные реализации имеют ряд недостатков:

- не многие разработки имеют открытые исходные коды или свободно выложены для безвозмездного пользования, а коммерческие разработки требуют покупки лицензии, что в ряде случаев затратно для отечественных научных проектов;
- структура нейронных сетей не является гибкой, т.е. описываемые ИНС не могут быть адаптированы для решения иных задач, так как нельзя изменить количество нейронов и слоев без преобразования бизнес-логики проекта;
- модули создаются под конкретную видеокарту, при этом не учитываются более широкие аппаратные возможности других CUDA-видеокарт (далее – устройств);
- зачастую выигрыш в производительности вычислений теряется из-за медленного обмена данными между оперативной памятью и видеокартой.

Однако применение данной технологии на видеокартах Nvidia GeForce (начиная с 8 серии) является эффективным в определенных классах ИНС, так как позволяет существенно ускорить их обучение. Как было показано в [1], уже на этапе прямого прохода выигрыш в скорости обучения достигается в сетях с количеством нейронов в слоях более 32.

В этой связи практический интерес представляет 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.

Достижение указанной цели подразумевает решение нескольких задач:

- выявление закономерности выделения на видеокарте памяти под необходимые параметры ИНС;
- подбор оптимальной конфигурации ядер для наиболее быстрого обучения ИНС на указанных выше устройствах;
- едение к минимуму обмена данными между оперативной памятью и видеокартой;

- явление пиков (т.е. топологий ИНС), при которых достигается максимальный выигрыш во времени обучения на устройствах разного типа.

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 (в режиме прогнозирования) или последующего (при обучении) слоя, невозможно вычислить параметры текущего слоя [2]. Существует два варианта 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. Для обучения ИНС был выбран второй способ. Он требует выделения на устройстве меньшего количества памяти, чем первый, что позволяет применять его для ИНС с большим количеством нейронов в слоях.

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 число нейронов должно быть кратно 16 [3], для чего матрицы весов (W_i), вектора входов (Y_{i-1}), выходов (Y_i) и порогов (T_i) выравниваются (заполняются нулями) до выбранных размерностей.

Схема взаимодействия внешнего и разработанного (в виде библиотеки) программных модулей (далее – 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). Корректирующие значения весов и порогов (ΔW_i и ΔT_i), текущие значения вектора ошибки (E_i) и вектора выходов слоя (Y_i) заполняются нулями. Если сеть дообучается, то используются ранее сохраненные значения W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i.

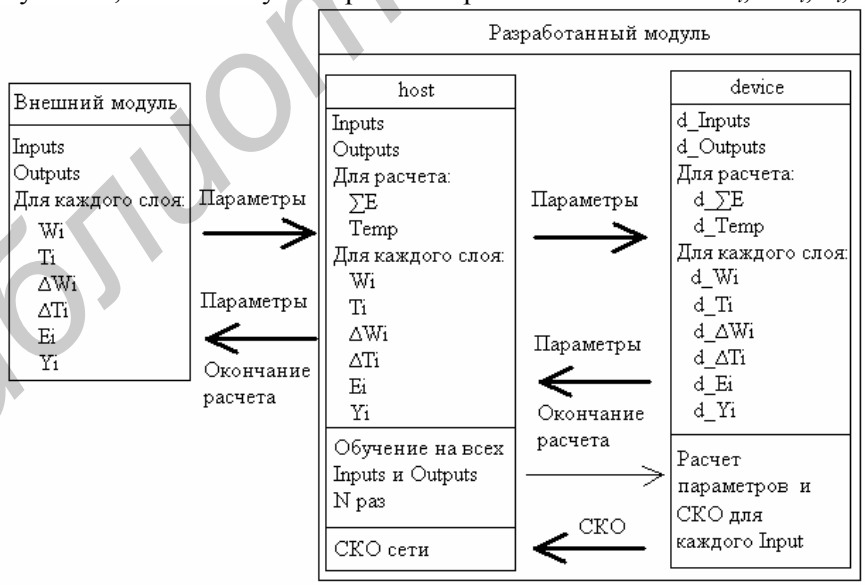

Рис. 1. Схема взаимодействия внешнего и разработанного модулей

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 в 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 значение среднеквадратического отклонения (СКО) выходного слоя, используемое при вычислении суммарного СКО каждой эпохи, записываемого в файл. По СКО эпохи можно судить о том, насколько 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, исключается необходимость постоянного перемещения требуемых для обучения параметров между устройством и управляющем модулем. Однажды получив их, устройство само обучает сеть, периодически возвращая только значение СКО для обеспечения возможности отслеживания текущего состояния ИНС. 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 состояния сети целесообразно использовать на этапе определения ее топологии, когда визуализация параметров позволяет изменять структуру для минимизации СКО.

Рис. 2. Отображение текущего состояния ИНС

Особенности выделения памяти

Необходимо учитывать, что память под параметры ИНС на видеокарте выделяется не вся. Экспериментально установлено, что 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 (92,5%), а при объеме 1024 Мб – 951 Мб (92,9%). Причем при расчете небольших массивов (например, 256 элементов) память выделяется из резерва.

На видеокарте GeForce 9500 GT детально исследовалась функция, выделяющая память под 3 вектора (размерностью m) и 3 матрицы ($m \times m$). На векторы глобальная память начинает выделяться при $m \geq 1024$, причем на первый вектор выделяется необходимое количество памяти, на второй и последующие – кратное 64 Кб (65536 байт).

С матрицами до 1024 × 1024 ситуация аналогичная. Однако при больших размерах массива (1024 x 1024 и выше) на первую матрицу выделяется памяти больше, чем требуется (уравнение, экспериментально полученное в MS Excel: $y = 4\,194\,304x^2 - 4\,096x + 65\,536$, $R^2 = 1{,}00$, где x – номер пункта в табл. 1, из чего получена формула (1) для выражения через размерность):

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

где *Memo* – объем выделяемой под матрицу памяти (байт), n – количество элементов в матрице.

Зависимость (1) была выведена по экспериментальным данным для матриц из табл. 1. На остальные матрицы (кроме первой) памяти выделяется ровно столько, сколько нужно.

Таблица 1. **Выделяемая под матрицы весов нейронов память**

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

Ограничения на размещение данных в памяти устройства

Расчетным путем было определено, сколько наборов тренировочных данных может быть размещено в памяти устройства для обучения сетей с количеством входных нейронов до 2608. Также выведены формулы, позволяющие оценить максимальные топологии нейронных сетей, удовлетворяющих 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 второго случая, когда N_0 кратно 16, $N_1 = 2N_0+16$;

в) количество выходных нейронов задается произвольно: $N_2 = kN_0$ ($k > 0$), но выровнено.

Следует 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, см. табл. 2) это значение всегда являлось кратным 1 Мб, что хорошо видно на рис. 3. Для карты 9800 GT выделяемая память была кратна ¼ Мб.

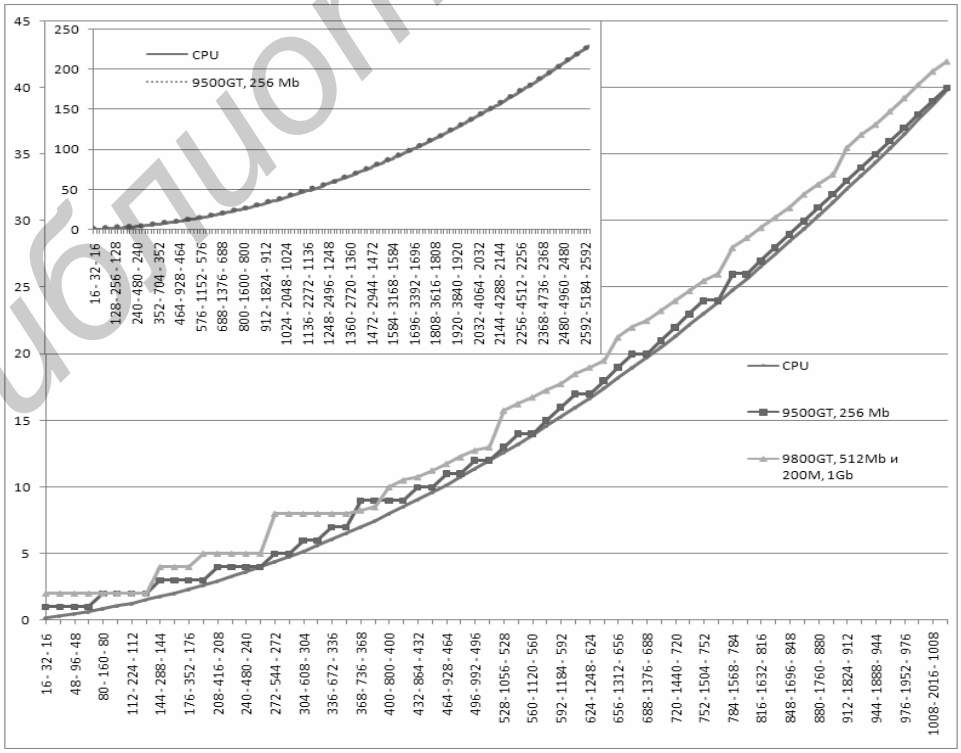

Рис. 3. Выделяемая память под параметры ИНС различных топологий

На рис. 3 представлен график зависимости выделяемой памяти в Мб на устройствах (9800 GT и 220 M) и необходимой памяти, рассчитанной теоретически (на рис. – CPU) по формуле (2), для размещения всех параметров двухслойной ИНС с разным количеством нейронов в слоях. На нем хорошо виден «ступенчатый» характер выделения памяти на видеокарте.

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

Пусть $N_0 = N$, *Epoch* – 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 $k \leq 2$ и $k \geq 2$ соответственно, для N_0, кратных 16, – (4) (4) для $k \leq 2$ и (5) для $k \geq 2$:

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

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

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

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

К примеру, из формулы (2(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 программного модуля является возможность его использования в любых проектах, поддерживаемых Visual Studio 2008 и выше. Причем время, затрачиваемое на обмен данными между основным и управляющим (host) модулями, не зависит от языка программирования, на котором написан основной код.

Тестирование скорости работы разработанного модуля проводилось на ЭВМ, оснащенной 4-ядерным процессором Intel Core 2 Quard (по 2,86 ГГц) и видеокартой GeForce 9500 GT, а также на видеокартах 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	Объем 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	август 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 (…–N_{i-1}–N_i–…), 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		Выигрыш во времени, %	N_i=N_{i-1}	Версия драйверов		Выигрыш во времени, %
	2.3, мс	3.0, мс			2.3, мс	3.0, мс	
32	140	110	21,4	288	2859	2656	7,1
64	172	157	8,7	320	3813	3531	7,4
96	266	250	6,0	352	4954	4563	7,9
128	437	390	10,7	384	6281	5781	8,0
160	688	625	9,1	416	7875	7250	7,9
192	1000	937	6,3	448	9688	8922	7,9
224	1515	1391	8,2	480	11844	10859	8,3
256	2094	1938	7,4	512	14094	12969	8,0

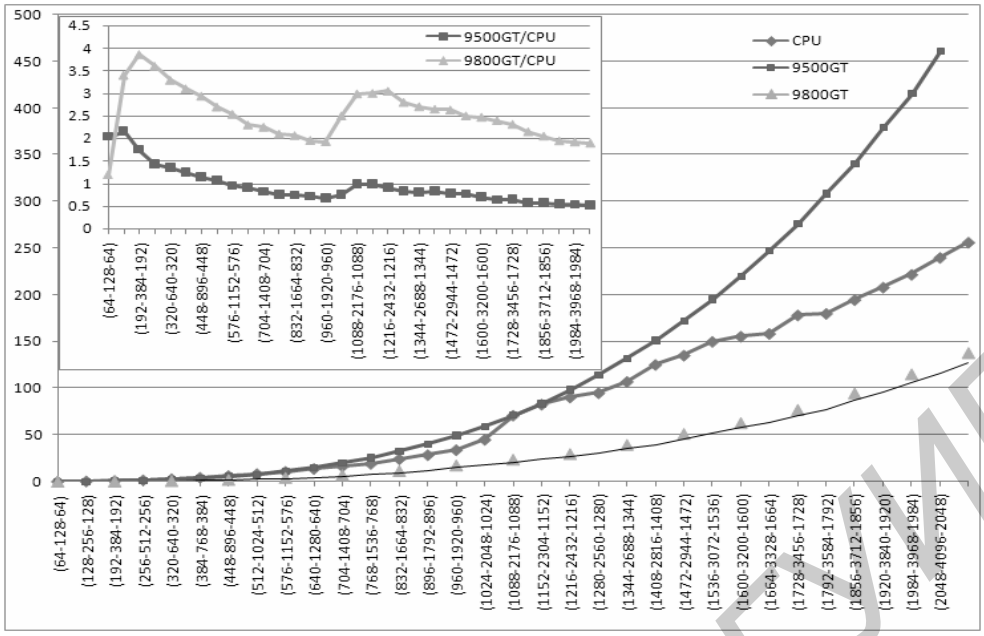

Рис. 4. Затраченное время на обучение ИНС с двумя скрытыми слоями (эпоха из 10 наборов)

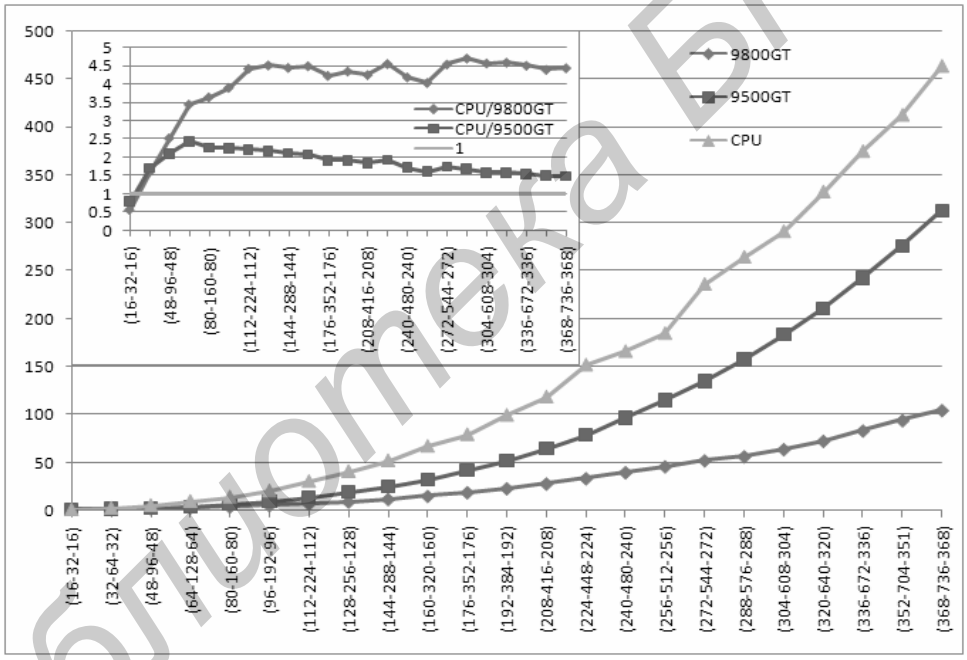

Рис. 5. Затраченное время на обучение ИНС с двумя скрытыми слоями (эпоха из 1000 наборов)

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

Используя CUDA-технологию, удалось ускорить обучение ИНС по сравнению с CPU до 5 раз на 9800 GT и до 2,5 раз на 9500 GT. Время обучения включает в себя время работы с файлами и минимальное перемещение данных между host и device. Выигрыш наблюдается уже в сетях (32 – 64 – 32). При обучении на 1000 наборах данных наибольший выигрыш на 9500 GT достигается при обучении ИНС (64 – 128 – 64), на 9800 GT – при обучении ИНС (288 – 576 – 288).

Заключение

В результате проведенных исследований установлено, что использование CUDA-технологии на основе разработанного подхода от 1,1 до 5 раз ускоряет обучение ИНС с коли-

чеством нейронов в слоях от 32. Однако такой эффект ускорения обучения по сравнению с CPU может быть получен не на любой сети. Не всякая сеть также может быть размещена в памяти вместе с необходимыми параметрами. При увеличении тренировочной выборки для размещения в памяти всех необходимых для обучения параметров размерность слоев должна быть уменьшена.

Производительность резко возрастает при количестве нейронов в слоях до 128, а затем, достигнув пика, постепенно снижается. Выигрыш во времени обучения зависит от конфигурации и частоты ядра, объема и пропускной способности видеопамяти, версии установленных драйверов.

На видеокарте Nvidia GeForce 9500 GT при увеличении количества нейронов в слоях от 128 выигрыш во времени обучения начинает снижаться, а слои с количеством нейронов более 512 обучаются медленнее, чем на CPU. Таким образом в двухслойных ИНС с равным количеством нейронов (N) во входном и выходном слоях ($N - 2N+1 - N$) применение CUDA-технологии на данной видеокарте является более эффективным при обучении небольших ИНС (с количеством входных нейронов от 128 до 512).

На Nvidia GeForce 9800 GT ИНС топологии ($N - 2N+1 - N$), где $64<N\leq2048$, обучается быстрее в 2 – 4,7 раз, чем на CPU, что указывает на целесообразность использования данного устройства при обучении таких ИНС.

CUDA-TECNOLOGY APPLICATION FOR ARTIFICIAL NEURAL NETWORK TRAINING

O.S. HILKO

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.

Литература

1. *Хилько О.С., Коваленко В.И., Кундас С.П.* // Докл. БГУИР. 2010. №7 (53). С. 83–88.
2. *Хайкин Саймон.* Нейронные сети. СПб., 2006.
3. *Хилько О.С., Коваленко В.И.* // Сахаровские чтения 2010 года: экологические проблемы XXI века: Матлы 10-й междунар. научной конференции. Минск, 2010. С. 87.
4. *Колмогоров А.Н.* // Докл. АН СССР. 1957. Т. 114, №5. С. 953–956.