УДК 004.38: 004.8.032.26

ПРИМЕНЕНИЕ CUDA-ТЕХНОЛОГИИ ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

О. С. Хилько, В. И. Коваленко, С. П. Кундас

Рассматривается применение CUDA-технологии для организации параллельных вычислений, базирующихся на искусственных нейронных сетях (ИНС) для прогнозирования миграции загрязняющих веществ в почвах. Показано, что CUDA-технология эффективна для ИНС с большим количеством нейронов в слоях. Эффективность использования зависит от конфигурации исполняемого ядра и возрастает с увеличением размерности блоков разделяемой памяти.

Введение

В настоящее время все более актуальными становятся технологии параллельных вычислений, которые с учетом современных архитектур настольных компьютеров позволяют значительно ускорять решение вычислительных задач, использующих массивы данных больших размерностей. Одной из таких технологий выступает CUDA фирмы Nvidia, позволяющая выполнять программы, написанные на подмножестве языка C [1], на графическом процессоре (GPU). Среди основных достоинств этой технологии можно выделить следующие: доступность конечному пользователю (ее широкое применение связано с развитой игровой индустрией, где GPU, поддерживающие CUDA-технологию, используются на протяжении последних лет); открытость архитектуры (наличие доступных для изучения программных реализаций алгоритмов под CUDA-платформу).

Применение CUDA-технологии в нейронных сетях

В настоящее время CUDA-технологии широко применяются в рамках нейросетевых решений для прогнозирования изменения курсов валют [2], распознавания графических образов [3], идентификации личности по 3D портрету [4] и др. К примеру, разработанная в [4] CUDA-библиотека, по мнению производителей, ускоряет обучение ИНС в сотни раз. Авторы [2] подчеркивают, что их CUDA-решение в 10 раз быстрее реализации на CPU.

В проекте SPS (Simulation Processes in Soil) [6] CUDA-технология используется в реализации алгоритмов обучения и вычисления выходов ИНС для прогнозирования миграции загрязняющих веществ почве и на ее поверхности. Обучение и расчет выходов ИНС ведется послойно, так как при прямом проходе выходы каждого предыдущего слоя являются входами для последующего. При обратном проходе без получения величины погрешности веса нейрона (δ_i) невозможно найти величину Δw, на которую нужно скорректировать веса и пороги нейронов [7].

При вычислениях на GPU используется быстрая разделяемая память (shared memory) для минимизации обращений к глобальной памяти, так как часто затраты на обмен данными между устройством (device) и управляющим модулем (host) могут быть сопоставимы с самим расчетом.

Вычисление параметров ИНС при прямом проходе

В рамках настоящей работы проводилось исследование зависимости производительности от конфигурации ядра CUDA-функции (количества потоков в блоке ядра, использование одно- или двумерных блоков) и размеров блоков разделяемой памяти (табл. 1). Замеры производительности проводились для вычисления выходов последующего слоя ИНС (Y_j) по параметрам двух смежных слоев:

$$Y_j = Fact\left(\sum W_{ij}^T Y - T\right), \quad (1)$$

где *Fact* – функция активации, W_{ij} – матрица весов, с которыми *i*-й слой влияет на *j*-й, *Y* – вектор-столбец *i*-го слоя, *T* – вектор столбец *j*-го слоя.

Таблица 1 – Сравнение времени расчета параметров смежных слоев для GPU и CPU

итерацийКоличество	ИНСТопология слоев	GPU1			GPU2			выполнения, мсCPU, время	Отношения затраченного времени		
		Размерность блоков	Время выпол-нения, мс	GPU_k / GPU_m	Размерность блоков	мсВремя выпол-нения,	GPU_k / GPU_m		GPU1/GPU2	GPU1/CPU	GPU2/CPU
1	8x8	8x8	0,046		8x8	0,019			2,448		
10	8x8	8x8	0,123	2,63	8x8	0,092	4,817		1,341		
100	8x8	8x8	0,894	7,22	8x8	0,817	8,862		1,094		
1000	8x8	8x8	25,84	28,91	8x8	7,980	9,767		3,238		
5000	8x8	8x8	164,45	6,36	8x8	39,80	4,987		4,131		
10000	8x8	8x8	346,77	2,1	8x8	79,54	1,998	31,2	4,359	11,11	2,549
50000	8x8	8x8	1764,14	5,08	8x8	404,4	5,084	109,2	4,362	16,15	3,703
1	16x16	16x16	0,055		16x16	0,019			2,872		
10	16x16	16x16	0,194	3,49	16x16	0,102	5,268		1,907		
100	16x16	16x16	1,591	8,16	16x16	0,872	8,535		1,824		
1000	16x16	16x16	29,80	18,73	16x16	8,526	9,776		3,496		
5000	16x16	16x16	170,74	5,72	16x16	42,43	4,976	31,2	4,023	5,47	1,36
10000	16x16	16x16	345,61	2,02	16x16	84,78	1,998	78	4,076	4,43	1,086
5000	16x1	16x1	1772,2	5,12	16x1	429,	5,071	296,	4,12	5,97	1,45

0	6	6	1		6	9		4	1		
1	32x32	32x32	0,085		16x16	0,022			3,864		
10	32x32	32x32	0,382	4,46	16x16	0,123	5,565		3,101		
100	32x32	32x32	3,472	9,08	16x16	1,151	9,344		3,015		
300	32x32	32x32	11,719	3,37	16x16	3,379	2,934	15,6	3,467	0,751	0,216
500	32x32	32x32	19,161	1,63	16x16	5,601	1,657	15,6	3,42	1,228	0,359
1000	32x32	32x32	36,735	1,92	16x16	11,13	1,988	31,2	3,298	1,177	0,357
5000	32x32	32x32	177,587	4,83	16x16	55,79	5,009	109,2	3,182	1,626	0,51
10000	32x32	32x32	356,691	2	16x16	113,0	2,025	234	3,156	1,524	0,482
20000	32x32	32x32	711,708	1,99	16x16	229,6	2,031	436,8	3,1	1,629	0,525
50000	32x32	32x32	1776,9	2,49	16x16	564,8	2,46	873,6	3,146	2,033	0,646

Так, были измерены значения затраченного времени на расчет выходов последующего слоя по параметрам смежных слоев ИНС (при количестве нейронов по слоям 8, 16, 32) на GPU с различным числом потоков в блоках. Необходимо так же учесть, что первый запуск расчетов на GPU выполняется в несколько раз медленнее, чем последующие [1], в связи с этим в табл. 1 приведены усредненные результаты 10 расчетов.

Тестирование приводилось на ЭВМ, оснащенной 4-ядерным процессором Intel Core 2 Duo (по 2,86 ГГц) и видеокартой GeForce 9500.

В первом варианте реализации (GPU1) использовались **одномерные** блоки ядра с количеством потоков до 32. Это было обусловлено аппаратным ограничением на размер разделяемой памяти (в видеокартах NVIDIA GeForce серий 8x/9x максимальный размер разделяемой памяти для блока – 16384 байта). В связи с этим, при работе с матрицей и вектором из 32-разрядных элементов с плавающей точкой используются двумерные квадратные блоки разделяемой памяти 32 x 32 (т.е. в памяти находится 32 x 32 x 4 + 32 x 4 = = 4224 байта). Однако при увеличении размера блоков до 64 x 64 в памяти недостаточно места для хранения всех параметров ИНС (хранятся лишь элементы матрицы, т.е. 64 x 64 x 4 = 16384).

Во втором варианте реализации (GPU2) применялись **двумерные** квадратные блоки ядра с количеством потоков до 16. Это обусловлено ограничением на количество потоков в блоке (в указанных выше видеокартах максимальное количество потоков в блоке – 512). При двухмерной реализации с количеством потоков в блоке равном 16 в блок включается 256 потоков (16 x 16 = 256). При использовании двухмерных блоков по 32 потока (32x32 = 1024) число потоков в блоке будет равным 1024.

В табл. 1 также приведено отношение затраченного времени для выполнения предыдущих итераций расчета к затраченному времени для выполнения расчета текущих итераций (GPU_k/ GPU_m). Эта величина характеризирует наличие прироста в скорости вычислений расчетов с увеличением числа итераций расчета. Например, для выполнения 10 итераций топологии 32x32 было затрачено 0,382 мс, а для 100 итераций той же топологии было затрачено 3,472 мс. Увеличив в 10 раз число итераций (от 10 к 100), затраченное на расчет время (GPU_k/ GPU_m) увеличилось в 9,08 раз.

В табл. 2 показаны отношения затраченного времени на расчет параметров слоев ИНС с различной размерностью блоков ядра (одномерная реализация с количеством потоков 8, 16 и 32), где при одинаковом количестве итераций сравнивается затраченное время на расчет параметров ИНС с размерностью блоков, отличающейся в 2 раза, и топологией, отличающейся в 4 раза.

Из результатов исследований, приведенных в табл. 3 (сравнение затраченного времени на расчет параметров смежных слоев ИНС по 64 и 128 нейронов для GPU (двумерная реализация с размером блоков ядра 16x16 и таким же объемом блоков разделяемой памяти) и CPU), видно, что именно в слоях с большим количеством нейронов (от 32) применение CUDA-технологии является эффективным.

Таблица 2 – Отношение затраченного времени на расчет параметров слоев на GPU

Количество итераций	Времена выполнения, мс			Отношение времен выполнения	
	8 (8-8)	16(16-16)	32(32-32)	16/8	32/16
1	0,046	0,055	0,085	1,189	1,535
10	0,123	0,194	0,382	1,576	1,960
100	0,893	1,591	3,472	1,780	2,181
1000	25,84	29,809	36,73	1,1533	1,232
5000	164,451	170,741	177,587	1,038	1,040
10000	346,775	345,615	356,691	0,996	1,032
50000	1764,143	1772,212	1776,898	1,004	1,002

Таблица 3 – Сравнение времени на расчет параметров смежных слоев ИНС на GPU

Количество итераций	GPU2		CPU	Отношение времен выполнения	
	Топология слоев	Время выполнения, мс	Время выполнения, мс	GPU_k/GPU_m	CPU/GPU2
1	64x64	0,048			
100	64x64	3,706		75,950	
300	64x64	11,093	31,250	2,993	2,816
500	64x64	18,518	46,875	1,669	2,531
1000	64x64	36,915	62,500	1,993	1,693
5000	64x64	184,732	343,750	5,004	1,860
10000	64x64	369,475	781,250	2,000	2,114
300	128x128	66,133	93,750	1,666	1,417
500	128x128	110,218	140,625	1,994	1,275
1000	128x128	219,786	328,125	1,994	1,492

Анализ полученных результатов

Проведенные исследования показывает, что CUDA-технология эффективна для ИНС с большим количеством нейронов в слоях. Эффективность возрастает с увеличением размерности блоков разделяемой памяти, но на ее максимальный размер существуют аппаратные ограничения. При количестве итераций от 100 до 50 000 пропорционально возрастают затраты времени на расчет параметров ИНС.

При количестве выполненных итераций меньше 100 затраты времени на расчет возрастает пропорционально увеличению размерности блоков. Однако так как при этом матрица весов увеличивается в 4 раза, то наблюдается прирост производительности в 2 раза по сравнению с CPU (табл. 1). При выполнении более 1000 итераций с увеличением в 2 раза размерности блоков время расчета увеличивается на 15-20%, а при количестве итераций более 5000 – на 1-5%, что дает еще больший прирост производительности.

Двумерная реализация GPU2 более быстрая, чем одномерная GPU1 (табл. 1), поэтому целесообразно использовать *двумерные* квадратные блоки ядра 16x16 потоков при такой же размерности блоков разделяемой памяти.

При выборе оптимальной размерности блока для решения конкретной задачи число нейронов из двух слоев выбирается по тому слою, где их количество выше и округляется в большую сторону до 2n, где n = {1, …, 5}. Матрица весов, векторы входов, выходов и порогов выравниваются (заполняются нулями) до выбранных размерностей.

Целесообразным является выбор конфигурации блоков для каждой пары слоев в отдельности. Например, в сети с топологией (26-16-1) для первой пары оптимальным является размер блока 32x32, для второй – 16x16. Возможно и комбинированное применение CUDA-вычислений и расчетов на CPU при проведении анализа количества нейронов в каждой из пар слоев и использовании CUDA для слоев с большими размерностями.

Литература:

1. Фролов, В. Введение в технологию CUDA / В. Фролов // Компьютерная графика и мультимедиа. – №6(1). – 2008. [Электр. рес.]: – Режим доступа: www.cgm.computergraphics.ru/issues/issue16/cuda – Дата дост.: 14.03.2010.
2. Калитин, Д.В. Использование технологии CUDA фирмы NVIDIA для САПР нейронных сетей / Д.В. Калитин // Устойчивое инновационное развитие: проектирование и управление. – №4. – 2009. – С. 16-19.
3. Яровой, А.А. Прикладная реализация масштабных нейронных и нейроподобных параллельно-иерархических сетей на основе технологий GPGPU // А.А. Яровой, Ю.С. Богомолов, К.Ю. Вознесенский / Наукові праці ВНТУ. – №2. – 2009.
4. A Neural Network on GPU. The code project – development resource [Electronic resource] / University of California, USA, 2007. – Mode of access: www.codeproject.com/KB/graphics/GPUNN.aspx– Date of access: 15.11.2009.

5. Нейросетевое увеличение и фильтрация цифровых изображений [Электронные ресурс] / Павлин Техно, 2006-2010 – Режим доступа: http://www.pawlin.ru/content/view/106/3. – Дата доступа: 15.11.2009.
6. Коваленко, В.И. Применение нейросетевых моделей программного комплекса SPS для прогнозирования миграции радионуклидов в почве / В.И. Коваленко, О.С. Хилько, С.П. Кундас // Сахаровские чтения 2009 года: экологические проблемы XXI века : Мат-лы 9-й Междунар. научн. конф., Минск, 21-22 мая 2009 г. / МГЭУ им. Сахарова: редкол.: С.П. Кундас, С.Б. Мельнов, С.С. Позняк [и др.]. – Минск, 2009. – С. 248-249.
7. Кундас, С.П. Реализация алгоритма обратного распространения ошибки с использованием дополнительного сигнала // С.П. Кундас, В.И. Коваленко, О.С. Хилько / Вестник ПГУ. Серия С, Фундамент. науки. – №9.– 2009.

Ольга Сергеевна Хилько, аспирант кафедры экологических информационных систем Международного государственного экологического университета (МГЭУ) им. А.Д. Сахарова, olga_hilko@tut.by

Валерий Иванович Коваленко, младший научный сотрудник НИЛ «Информационные системы и технологии в экологии» (ИСиТвЭ) МГЭУ им. А.Д. Сахарова, kovalenko@iseu.by

Семен Петрович Кундас, д.т.н., проф., главный научный сотрудник НИЛ ИСиТвЭ, ректор МГЭУ им. А.Д. Сахарова. kundas@iseu.by