УДК 004.932

В. А. КОЛБАСИН, канд. техн. наук, каф. САиУ, НТУ «ХПИ»

ИСПОЛЬЗОВАНИЕ РАЗДЕЛЯЕМОЙ ПАМЯТИ ПЛАТФОРМЫ CUDA В ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ

У статті розглядається вплив способу використання пам'яті, що розділюється, на продуктивність реалізації штучної нейронної мережі на платформі CUDA. Розглянуто варіанти розміщення кількох вікон вихідних даних та вагових коефіцієнтів в пам'яті, що розділюється. Показано, що через нераціональне використання часу очікування завантаження даних з глобальної пам'яті продуктивність цих варіантів не перевершує продуктивності базової схеми розпаралелювання.

В статье рассматриваются влияние способа использования разделяемой памяти на производительность реализации искусственной нейронной сети на платформе CUDA. Рассматриваются варианты размещения нескольких окон исходных данных и весовых коэффициентов в разделяемой памяти. Показано, что из-за нерационального использования времени ожидания загрузки данных из глобальной памяти производительность этих вариантов не превосходит производительности базовой схемы распараллеливания.

The performance of several schemes of shared memory usage in artificial neural network implementation on a CUDA platform is considered. The placement of several windows of input data and neuron inputs weights in shared memory is investigated. It is shown, that due to waiting while data is loaded from global memory, performance of these schemes doesn't exceed the performance of basic scheme of parallelization.

Введение. Аппарат искусственных нейронных сетей (ИНС) широко применяется для решения многих практических задач, связанных с распознаванием образов, классификацией сигналов различной природы [1,2]. Одним из существенных ограничений при использовании ИНС является их высокая ресурсоемкость, что усложняет обработку данных в режиме реального времени.

В ИНС каждый нейрон является независимым вычислителем, и все нейроны одного слоя могут работать параллельно. Поэтому для уменьшения времени обработки данных искусственными нейронными сетями могут быть использованы технологии параллельных вычислений.

Для параллельной реализации ИНС используются технологии команд потоковых SIMD-расширений процессора (SSE), средства многопроцессорных и распределенных вычислений (OpenMP и MPI, соответственно), технологии неграфических вычислений на процессорах видеокарт (AMD StreamComputing, NVidia CUDA).

Технология NVidia CUDA (Computer Unified Device Architecture) [3,4] позволяет использовать процессоры видеокарт для выполнения неграфических расчетов, обеспечивая высокую вычислительную производительность при сравнительно низкой стоимости решения. В работах [5-7] было показано, что реализация ИНС на базе платформы CUDA позволяет достичь существенного прироста производительности в задачах обработки данных ИНС, и были предложены схемы параллельной реализации ряда ИНС. В работе [7] автором была предложена схема распараллеливания вычислений для обработки нейронной сетью прямого распространения потока данных. Данная схема предполагает обработку одного блока данных на одном мультипроцессоре, при этом значения выходов обрабатываемых слоев нейронов располагаются в разделяемой памяти CUDA. В случаях, когда размеры ИНС малы, объем разделяемой памяти позволяет разместить в ней переменные для обработки нескольких блоков данных и даже весовые коэффициенты нейронов. Однако, из-за определенных ограничений на доступ к разделяемой памяти эти меры могут и не привести к увеличению скорости обработки. Проверке данных возможностей ускорения обработки посвящена данная работа.

Схема обработки данных с использованием ИНС. Как и в работе [7], будем исходить из предположения, что входной поток данных разбивается на окна анализа фиксированной длины, а затем данные каждого окна анализа передаются на вход ИНС. Также в данной работе предполагается, что если в прикладной задаче окна анализа перекрываются, то и ИНС они обрабатываются независимо друг от друга. То есть, задача оптимизации обработки зоны перекрытия окон анализа в данной работе не рассматривается.

Также ограничимся рассмотрением многослойной ИНС прямого распространения с сигмоидной активационной функцией и количеством нейронов выходного слоя, равным количеству нейронов входного слоя, поскольку такая ситуация является наиболее сложной с точки зрения производительности.

Значение на выходе каждого нейрона вычисляется по формуле [1]:

$$y_i = f\left(\sum_{j=0}^{N-1} x_j \cdot w_{i,j}\right), \qquad (1)$$

где x_j - значение на j-м входе i-го нейрона;

$w_{i,j}$ - весовой коэффициент;

y_i - значение на выходе i-го нейрона;

$f(z) = 1/\left(1+e^{-x}\right)$ - функция активации.

Слои нейронов обрабатываются последовательно: сначала вычисляются значения на выходах всех нейронов первого слоя, затем, используя полученные значения как входные значения следующего слоя, вычисляются значения на выходах второго слоя и т.д. В качестве результатов обработки данных ИНС рассматриваются значения выходов нейронов последнего слоя.

Оптимизация вычисления ИНС для CUDA. Обработку данных ИНС предлагается проводить по следующему принципу. Каждое окно исходных

данных будет обрабатываться в одном блоке потоков, а каждый поток будет вычислять значение на выходе одного или нескольких нейронов. При этом количество блоков потоков определяется исходя из требований к скорости реакции системы. Использование большего числа блоков, чем количество установленных на аппаратуре потоковых мультипроцессорах (SMP), позволяет уменьшить затраты на запуск потоков и, если достаточно ресурсов, запустить несколько блоков выполняться на одном SMP.

Таким образом, схема обработки данных имеет следующий вид:

1. Накапливается набор данных из $N \cdot G$ отсчетов, где N – длина окна анализа, а G – количество обрабатываемых за один раз окон анализа. В свою очередь, значение G имеет смысл делать кратным числу имеющихся SMP.

2. Накопленные данные копируются в память устройства CUDA и запускается их обработка с помощью ИНС. При этом исходные данные копируются в разделяемую память, и послойно выполняется расчет значений выходов каждого слоя нейронов. Значения на выходах последнего слоя нейронов записываются в память устройства CUDA.

3. Результаты обработки копируются в память компьютера.

Для случая малого размера ИНС предлагается проанализировать влияние на производительность следующих модификаций данного подхода.

Обработка нескольких окон исходных данных в одном блоке. Как было показано в работе [7], наибольшая производительность достигается, если число потоков в блоке превосходит 100. Если максимальное число нейронов в слое в несколько раз меньше числа потоков, можно попытаться обработать несколько окон данных в одном блоке. При этом во избежание конфликтов доступа к разделяемой памяти и сопутствующего этому снижения производительности данные каждого блока должны быть выровнены по границе банков памяти, то есть на 16 элементов.

Хранение весов нейронов в разделяемой памяти. Пусть для максимального числа нейронов в слое K выполняется соотношение

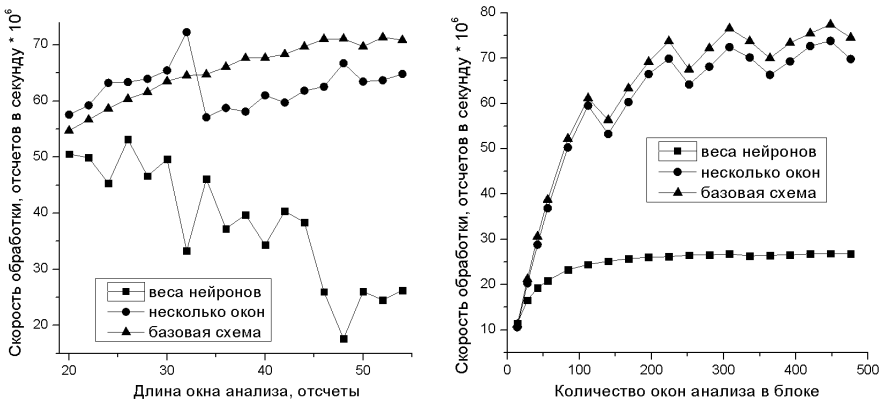
$$2 \cdot A(K) + A(K) \cdot K \le SM , \qquad (2)$$

где SM – объем разделяемой памяти в переменных с плавающей точкой;

$A(K)$ – оператор округления в большую сторону до кратности 16.

Тогда весовые коэффициенты входов нейронов можно разместить в разделяемой памяти. Однако начала строк матрицы весов должны быть выровнены по границе 16, чтобы не возникало конфликтов доступа к разделяемой памяти.

Результаты. В рамках данной работы были измерены скорости обработки данных для базового алгоритма и для его модификаций. Результаты для числа потоков в блоке не более 128 приведены на графике.



а) б)

Зависимость скорости обработки от длины окна анализа при 196 окнах на блок потоков (а) и от количества окон анализа в блоке при длине окна анализа 50 (б).

Как видно из приведенных графиков, скорость обработки данных с использованием базовой схемы в большинстве случаев выше, чем у предложенных вариантов. В результате детального анализа с использованием средств профилирования было определено, что причиной данного явления стала нехватка ресурсов для запуска достаточного числа потоков выполнения, которые бы сгладили задержки доступа к глобальной памяти устройства. Незначительный прирост производительности наблюдался при обработке нескольких окон анализа в одном блоке, когда в блоке помещалось не менее четырех окон анализа. Данные зависимости могут быть использованы при проектировании технических систем, использующих обработку данных при помощи ИНС в режиме реального времени.

Список литературы: 1. *Бодянский Е. В.* Искусственные нейронные сети / *О. Г. Руденко, Е. В. Бодянский*. - Х.: Компания СМИТ, 2005. - 408 с. **2.** *Осовский С.* Нейронные сети для обработки информации / *С. Осовский.* – М. Финансы и статистика, 2004. – 344 с. **3.** NVidia CUDA Programming Guide [Электронный ресурс] / NVidia Corp, 2008. – Режим доступа: http:// developer.download.nvidia.com/compute/cuda/3_2/toolkit/docs/CUDA_C_Programing_Guide.pdf. - 10.05.2011 г. - Загл. с экрана. **4.** *Боресков А. В.* Основы работы с технологией CUDA [Текст] / *А. В. Боресков, А. А. Харламов*. – М.:ДМК Пресс, 2010. – 232 с. **5.** *Jang H. H.* Neural Network Implementation Using CUDA and OpenMP / *H. H. Jang, A. J. Park, K. C. Jung* // Proceeding of Computing: Techniques and Applications, 2008. – p. 155–161. **6.** *Uetz R.* Large-scale Object Recognition with CUDA-accelerated Hierarchical Neural Networks Intelligent Computing and Intelligent Systems / *R. Uetz, S. Behnke* // Proceeding of Intelligent computing and Intelligent Systems, 2009. – p. 536 – 541. **7.** *Колбасин В. А.* Параллельная обработка данных искусственными нейронными сетями на платформе CUDA / *В.А. Колбасин* // Восточно-Европейский журнал передовых технологий. – Харьков, 2011. - № 3/3 (51). – С. 54-57.

Надійшла до редколегії 22.08.2011