УДК 004.04

МОДЕЛИРОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С ПОМОЩЬЮ ГРАФИЧЕСКОГО АДАПТЕРА ОБЩЕГО НАЗНАЧЕНИЯ

Королев Александр Аркадьевич,
магистрант каф. автоматизированных систем обработки информации и управления ФГБОУ ВПО «Ижевский государственный технический университет им. М.Т. Калашникова»,
Россия, 426069, г. Ижевск, ул. Студенческая, 7. E-mail: ak-hpc@yandex.ru

Кучуганов Александр Валерьевич,
канд. техн. наук, доцент каф. автоматизированных систем обработки информации и управления ФГБОУ ВПО «Ижевский государственный технический университет им. М.Т. Калашникова»,
Россия, 426069, г. Ижевск, ул. Студенческая, 7. E-mail: Aleks_KAV@udm.ru

Актуальность *работы обусловлена тем, что искусственные нейронные сети, будучи наиболее успешным подходом к решению некоторых задач искусственного интеллекта, предъявляют высокие требования к вычислительным ресурсам. При этом в большинстве случаев именно высокая вычислительная нагрузка оказывается ограничивающим фактором, снижающим на практике функциональность и применимость аппарата искусственных нейронных сетей.*

Цель работы: *повышение эффективности при решении задач искусственного интеллекта с применением искусственных нейронных сетей путём увеличения производительности моделирования за счёт применения высокопараллельных вычислений на графическом адаптере общего назначения.*

Методы исследования. *Теоретические исследования выполнены с использованием теории параллельных вычислений, теории графов, векторной алгебры и методов системного анализа. В ходе экспериментальных исследовании осуществлена апробация программного комплекса системы анализа изображений с использованием предложенных подходов.*

Результаты. *Предложен подход для моделирования нейронных сетей различных архитектур с высокой степенью параллелизма обработки, позволяющий перенести вычисления на графический адаптер общего назначения, заключающийся в группировке связей между нейронами по признаку их параллелизма по времени обработки. Такая группировка позволила заранее определить, какие вычислительные задачи могут выполняться параллельно и какие последовательно, что заметно упростило перенос вычислений на графический адаптер, а также позволило реализовать пакетную обработку, ускоряющую вычисления и на центральном процессоре. Достигнутый коэффициент ускорения обработки за счёт использования параллельных вычислений на графическом адаптере достигает коэффициента отношения его пиковой теоретической производительности к таковой характеристике центрального процессора, что говорит о высокой эффективности предложенного подхода.*

Ключевые слова:
Искусственные нейронные сети, массовый параллелизм, графические адаптеры общего назначения, CUDA, распознавание образов.

Введение

Графические адаптеры общего назначения (GPGPU – General Purpose Graphics Processing Unit) получили широкое распространение среди персональных компьютеров [1]. Любой современный настольный ПК или ноутбук оснащён видеоадаптером, поддерживающим технологии GPGPU, при этом примерно половина рынка видеоадаптеров поддерживает технологию CUDA (Compute Unified Device Architecture) [2]. Возможность распараллеливания алгоритма позволяет без дополнительных затрат увеличить эффективность в 10–150 раз, применяя технологию CUDA [3]. При этом ключевым моментом, определяющим эффективность реализации, является степень параллелизма алгоритма, поскольку архитектура GPGPU как вычислительного устройства подразумевает значительно большее относительно CPU количество вычислительных единиц – процессорных ядер (64–4096 против 1–12) [4].

Обработка нейросетевых процессов – крайне ресурсоёмкая задача [5]. Доступные на данный момент библиотеки и алгоритмы для моделирования нейронных сетей, как правило, не поддерживают параллельную обработку [6] либо являются узкоспециализированными решениями [7–9]. Основной сложностью переноса вычислений на GPU является необходимость разработки алгоритма высокопараллельной обработки [3], который при этом может сильно отличаться для нейронных сетей различных архитектур. В рамках данной статьи будет описан подход к распараллеливанию нейронной сети произвольной архитектуры с возможностью переноса вычислений на графический адаптер общего назначения.

Группировка связей между нейронами при моделировании искусственной нейронной сети

Нейронная сеть по определению является структурой, состоящей из большого количества простых элементов, взаимодействующих между собой. Именно взаимодействие является ограничивающим фактором при организации параллельной обработки [3, 10]. Архитектура нейронной сети од-

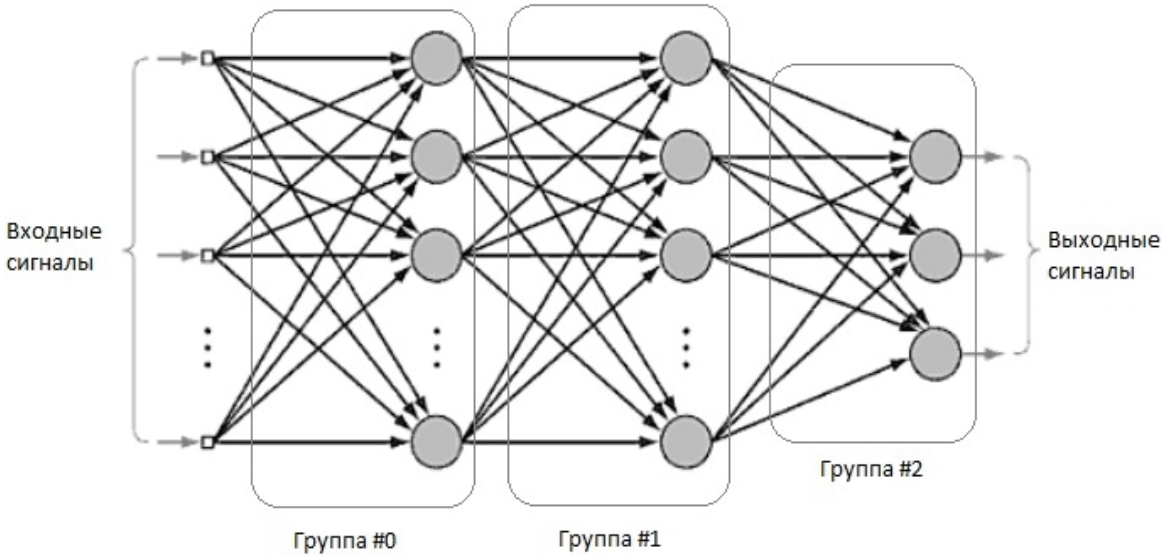

Рис. 1. *Многослойный перцептрон (MLP) с указанием групп параллельных связей*

Fig. 1. *MultiLayer Perceptron (MLP) indicating groups of parallel links*

нозначно определяет направление взаимодействий между нейронами [11]. Предлагается выделить группы связей между нейронами, которые физически могут обрабатываться параллельно, другими словами – порядок обработки которых не влияет на конечный результат. Как правило, архитектура нейронной сети предполагает разбиение нейронов на группы, или слои, между которыми и происходит взаимодействие [11]. Для эффективной организации параллельной обработки предлагается разбивать на группы не сами нейроны, а связи между ними. Последующая обработка будет опираться на списки связей нейронной сети, а не на сами нейроны, как это принято в большинстве современных реализаций [5–9, 11]. Формирование взвешенной суммы сигналов, входящих в нейроны текущего слоя, будет осуществляться с максимальной степенью параллелизма на этапе обработки списка связей. Затем обрабатываются сами нейроны, формируя исходящие сигналы в текущем слое. Такая группировка позволяет без дополнительных затрат определить, какие вычислительные задачи могут выполняться параллельно, а какие последовательно (в том числе с указанием порядка последовательности, аналогично слоям нейронов). Другими словами – в группированных списках связей будет храниться архитектура нейронной сети, и, опираясь на эти списки, можно будет осуществлять обработку без нарушения логики самой модели нейронной сети. Кроме того, при таком подходе значительно возрастёт степень параллелизма вычислений, поскольку количество связей между нейронами в нейронной сети, как правило, значительно больше, чем количество самих нейронов [11].

В качестве примера предлагается рассмотреть многослойный перцептрон (MLP – MultiLayer Perceptron), изображённый на рис. 1 и являющий собой классическую архитектуру нейронной сети [12].

Обработку MLP легко представить как последовательную обработку этапов, каждый из которых можно обрабатывать параллельно. Нейроны в каждом конкретном слое не взаимодействуют друг с другом, однако их обработка требует, чтобы все нейроны предыдущего слоя были обработаны [12]. Следовательно, каждая конкретная группа связей будет объединять только те связи, которые входят в нейроны соответствующего слоя. Таким образом определяются группы связей и, собственно, последовательная и параллельная части всего алгоритма обработки MLP.

Группировка на примерах различных топологий искусственных нейронных сетей

Важно отметить, что такое разбиение подходит для нейронной сети любой топологии. Примерами частных случаев являются: рассмотренный MLP, широко распространённые в распознавании образов свёрточные нейронные сети (CNN – Convolutional Neural Network) [13], рекуррентные нейронные сети с обратными связями (RNN – Recurrent Neural Network) [14]. Кроме того, группы автоматически формируют относительный временной порядок трассировки сигнала по сети, что позволяет полноценно моделировать произвольные сети с обратными связями и задержками сигнала. На рис. 2 изображена ограниченная машина Больцмана (RBM – Restricted Boltzmann Machine) со схематичным указанием групп параллельных связей.

Топология RBM тривиальна, тем не менее данная нейронная сеть резко отличается от классических моделей тем, что один и тот же слой нейронов (внешний) используется и как вход сети, и как её выход [15]. Предложенный подход к обработке нейронных сетей позволяет моделировать данную нейронную сеть, организовав две группы, состоящие из одних и тех же связей, но с разным напра-

влением (что соответствует логике модели RBM). При этом сохраняется максимальная степень параллелизма вычислений.

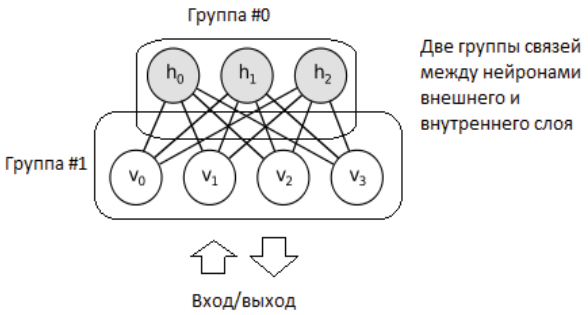

Рис. 2. *Ограниченная машина Больцмана (RBM)*

Fig. 2. *Restricted Boltzmann Machine (RBM)*

Аналогичным образом строится модель нейронной сети с обратными связями. Схема нейронной сети Элмана с указанием групп связей изображена на рис. 3.

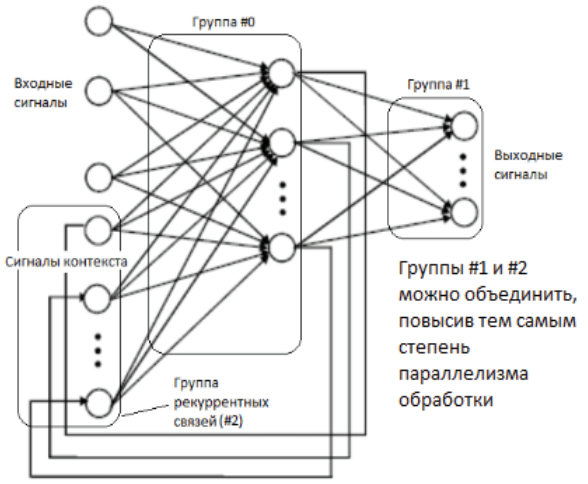

Рис. 3. *Нейронная сеть Элмана, частный случай нейронной сети с рекуррентными связями (RNN)*

Fig. 3. *Ellman neural network, a special case of Recurrent Neural Network (RNN)*

На схеме, изображённой на рис. 3, группу #1 и группу #2 можно объединить, не нарушая логику обработки нейронной сети. Данный пример показывает, как два различных направления распространения сигнала (между тремя различными слоями нейронной сети) могут быть объединены на этапе построения сети, повышая тем самым эффективность обработки при использовании параллельных вычислений.

Аналогичным образом можно построить модель нейронной сети Хопфилда и других сетей с временными задержками, а также и другие разновидности нейронных сетей. Исключением являются сети, в которых сигналы и/или взаимодействие между нейронами происходит не с помощью явного указания связей, а основываясь на некоторых показателях состояния нейронов и самой нейронной сети. Примером таких сетей являются самоорганизующаяся карта Кохонена и осцилляторные нейронные сети [16, 17]. Очевидно, для обработки таких нейронных сетей, предложенный выше алгоритм необходимо отдельно адаптировать.

Предложенный алгоритм разбиения обработки на последовательные и параллельные части требует соблюдения следующих условий. Необходимо, чтобы множество связей между нейронами было логически отделено от множества самих нейронов. На практике это означает наличие самостоятельного массива связей между нейронами в памяти компьютера. Также необходимо хранение в памяти списков связей и нейронов, принадлежащих каждой группе. В случае, если гарантируется отсутствие повторного использования одной связи, достаточно указать количество связей и нейронов в группе. Таким образом, предложенный подход требует минимальных дополнительных затрат по памяти и обеспечивает максимальную степень параллелизма.

Реализация пакетной обработки

В рамках данного подхода можно также реализовать пакетную обработку – одновременное моделирование некоторого множества экземпляров искусственных нейронных сетей. Нейронные сети часто применяются для обработки большого количества данных [18], например, при попиксельной или сегментированной обработке изображений и звуковых дорожек [19, 20]. Очевидно, что такую обработку можно дополнительно распараллелить, создав несколько экземпляров нейронных сетей и объединив связи из всех экземпляров в соответствующие группы. Схематичный пример организации пакетной обработки изображён на рис. 4.

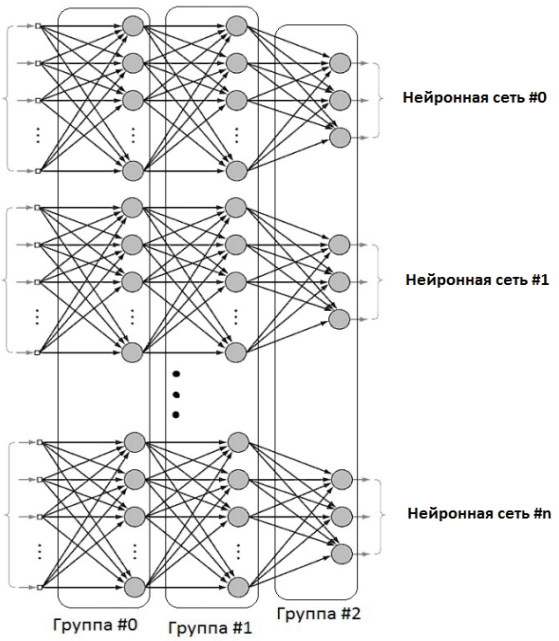

Рис. 4. *Пакетная обработка – объединение обработки нескольких экземпляров нейронных сетей*

Fig. 4. *Batch processing – consolidation of processing several instances of neural networks*

Таблица. *Время трассирования сигнала по нейронной сети (Single – без пакетной обработки, Batch – с пакетной обработкой в пересчёте на 1 экземпляр)*

Table. *Time of signal tracing along neural network (Single – without batch processing, Batch – with batch processing in terms of an instance)*

Количество экземпляров в пакете Amount of instance in a batch	1	1	16	256	4096	65536
Тип сети (MLP)/Network type (MLP)	192×8	1024×8	512×8	256×2	64×8	16×4
Общее кол-во нейронов/связей в сети Total amount of neurons/links in a network	1538 258432	8194 7342080	65568 29376512	131584 16908288	2105344 117964800	4325376 52428800
CPU Single, ms.	1,2610	20,2733	110,2064	115,3024	963,3792	1520,4352
CPU Batch, ms.	1,2610	20,2733	82,2590	51,4965	433,9707	371,8467
GPU Batch, ms.	0,4540	1,3140	3,2691	2,2678	13,5141	12,5394
Ускорение GPU, коэфф./GPU acceleration, ratio	2,78	15,43	25,16	22,71	32,11	29,65
Ускорение CPU/GPU относительно CPU Single, коэфф. CPU/GPU acceleration relative to CPU Single, ratio	1,00 2,78	1,00 15,43	1,34 33,71	2,24 50,84	2,22 71,29	4,10 121,25

При использовании описанного подхода данные нейронных сетей (сигналы и весовые коэффициенты) можно хранить в памяти локально (физически в неразрывной области памяти), что на практике приводит к более эффективному использованию механизмов кеширования и, как следствие, к ускорению обработки (до 6 раз) [21].

Тестирование и анализ производительности

Предложенные подходы были использованы при реализации библиотеки нейросетевой обработки на языке C++ с использованием технологии CUDA для программирования GPU. Тестирование производилось с точки зрения производительности, при абстрактной обработке сигналов в нейронной сети, а также при обучении нейронной сети распознаванию рукописных символов. Результаты замеров времени трассировки сигнала по нейронной сети различной конфигурации в различных режимах указаны в таблице (CPU Core i5–4670K, GPU GeForce GTX 770).

Из таблицы видно, что за счёт использования GPU удалось ускорить обработку примерно в 32 раза, что практически соответствует отношению пиковых GFLOPS использованных GPU и CPU – 3213 и 95 GFLOPS соответственно. Это может говорить о том, что достигнута практически максимальная параллельная утилизация GPU (задействованы все 1536 параллельных ядер GTX 770). При этом пакетная обработка ускоряет вычисления даже в режиме на CPU (до 4-х раз, за счёт применения локальности данных), а режим GPU относительно непакетного режима CPU позволил ускорить обработку примерно в 120 раз.

Применение разработанной библиотеки при исследовании возможности распознавания образов с помощью свёрточных нейронных сетей на основе автоэнкодеров (кластеризаторов в виде многослойных перцептронов с «узким горлом») позволило сократить время обучения с ~5,75 часов до 31-й минуты с помощью режима GPU (обучающая выборка содержала 60 тыс. образцов). При этом в режиме на CPU без использования пакетной обработки теоретическое время обучения составило бы ~17 часов.

Выводы

Искусственные нейронные сети генерируют высокую вычислительную нагрузку, но при этом предлагают наиболее качественное решение некоторых задач искусственного интеллекта. Исследование искусственных нейронных сетей, так же как и применение их на практике, может быть ограничено из-за высоких требований к вычислительной аппаратуре.

Предложенный подход позволяет существенно повысить производительность вычислений при моделировании искусственных нейронных сетей. Подход позволил реализовать библиотеку для моделирования разнообразных искусственных нейронных сетей с высокой эффективностью обработки за счёт достижения максимальной степени параллелизма. Достигнутые коэффициенты ускорения за счёт применения GPU могут говорить о максимальной параллельной утилизации (задействовании всех параллельных ядер GPU) и, соответственно, о повышении эффективности моделирования искусственных нейронных сетей.

Работа выполнена при поддержке Госзадания МОиН РФ.

СПИСОК ЛИТЕРАТУРЫ/REFERENCES

1. Kim H., Vuduc R., Baghsorkhi S. *Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU)*. San Francisco, CA, USA: Morgan & Claypool Publishers, 2012. 96 p.
2. Hwu W.W. *GPU Computing Gems Jade Edition (Applications of GPU Computing Series)*. San Francisco, CA, USA, Morgan Kaufmann, 2011. 560 p.
3. Kirk D.B., Hwu W.W. *Programming Massively Parallel Processors: a Hands-on Approach*. San Francisco, CA, USA, Morgan Kaufmann, 2010. – 280 p.
4. Hennessy J.L. *Computer Architecture: a Quantitative Approach*. San Francisco, CA, USA, Morgan Kaufmann, 2011. 856 p.
5. Gurney K. *An Introduction to Neural Networks*. London, Routledge, 1997. 423 p.
6. O'Neil M. Neural Network for Recognition of Handwritten Digits. *Code Project*. 2006. Available at: http://www.codeproj-

ect.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi (accessed 06 April 2013).
7. Nissen S. Fast Artificial Neural Network Library. *Code Project*. 2008. Available at: http://leenissen.dk/fann/wp/ (accessed 06 April 2013).
8. Verber D. Implementation of Massive Artificial Neural Networks with CUDA Cutting Edge Research in New Technologies. *Intech*. 2012. Available at: http://www.intechopen.com/books/cutting-edge-research-in-new-technologies/ implementation-of-massive-artificial-neural-networks-with-cuda (accessed 06 April 2013).
9. Conan B., Guy K. A Neural Network on GPU. *Code Project*. 2008. Available at: http://www.codeproject.com/Articles/24361/A-Neural-Network-on-GPU (accessed 06 April 2013).
10. Pacheco P. *An Introduction to Parallel Programming*. San Francisco, CA, USA, Morgan Kaufmann, 2011. 392 p.
11. Haykin S. *Neural Networks: a Comprehensive Foundation*. 2nd ed. NJ, Prentice Hall NJ, 1998. 842 p.
12. Wasserman Ph. *Advanced Methods in Neural Computing*. New York, Van Nostrand Reinhold, 1993. 653 p.
13. Behnke S. *Hierarchical Neural Networks for Image Interpretation*. New York, Springer-Verlag, 2003. 43 p.
14. Elman J. *Rethinking Innateness*. New York, Van Nostrand Reinhold, 1996. 145 p.
15. Ciresan D., Meier U., Schmidhuber J. *Multi-column Deep Neural Networks for Image Classification*. San Francisco, CA, USA, Morgan Kaufmann, 2012. 430 p.
16. Smith M. *Neural Networks for Statistical Modeling*. New York, Van Nostrand Reinhold, 1993. 181 p.
17. Nauck D., Klawonn F., Kruse R. *Foundations on Neuro-Fuzzy Systems*. Chichester, Wiley, 1997. 132 p.
18. Duda R.O., Hart P.E., Stork D.G. *Pattern classification*. 2nd ed. Chichester, Wiley, 2001. 438 p.
19. Ripley B.D. *Pattern Recognition and Neural Networks*. Cambridge, Cambridge University Press, 1996. 375 p.
20. Theodoridis S., Koutroumbas K. *Pattern Recognition*. 4th ed. Burlington, Ma, Academic Press, 2009. 293 p.
21. Cragon H. G., *Memory Systems and Pipelined Processors*. Sudbury, Massachusetts, Jones and Barlett Publishers, 1996. 209 p.

Поступила 09.06.2014 г.
Received: 09 June 2014.

UDC 004.04

SIMULATION OF ARTIFICIAL NEURAL NETWORKS USING GENERAL PURPOSE GRAPHICS PROCESSING UNIT

Alexander A. Korolev,
M.T. Kalashnikov Izhevsk State Technical University, 7, Studencheskaya street, Izhevsk, 426069, Russia. E-mail: ak-hpc@yandex.ru

Alexander V. Kuchuganov,
Cand. Sc., M.T. Kalashnikov Izhevsk State Technical University, 7, Studencheskaya street, Izhevsk, 426069, Russia. E-mail: Aleks_KAV@udm.ru

The relevance *of the discussed issue is caused by the high computational load generating by an artificial neural network simulation, while the latter is the most successful solution for several AI tasks. In most cases, the high computational load of artificial neural network simulation causes a decline of its functionality and restricts its applicability.*
The main aim *of the study is to improve the efficiency of resolving the AI tasks using artificial neural networks by improving simulation performance applying parallel computations on general purpose graphics processing unit.*
The methods. *The theoretical researches were carried out using concurrency theory, graph theory, vector algebra and methods of systems analysis. During the experimental study the authors tested an image analysis system software complex that uses the proposed approaches.*
The results. *The authors proposed an approach to simulate the variety of artificial neural networks with high degree of parallelism, which is based on specific precomputation of the groups of compute-time parallel connections between neurons. This group defines explicitly what parts of overall computational task can be performed in parallel. The approach allows transferring as well a computational load to graphics processing unit and performing a batch processing on central processing unit. The achieved performance speed-up ratio reaches the ratio of GPU peak theoretical performance to that of CPU indicating the high efficiency of the proposed approach.*

Key words:
Artificial neural networks, massive parallelism, general purpose graphics processing unit, CUDA, image recognition.

The research was supported by the State task of the Ministry of Education and Science of the Russian Federation.