НАУЧНОЕ ИЗДАНИЕ МГТУ ИМ. Н. Э. БАУМАНА
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС77 - 48211. Государственная регистрация №0421200025. ISSN 1994-0408
электронный научно-технический журнал

О реализации нейросетевого алгоритма распознавания лиц на графических процессорах

 # 12, декабрь 2013
DOI: 10.7463/1213.0659387
Терехов В. И.
УДК 004.032.26; 004.272.2; 004.8

Россия, МГТУ им. Н.Э. Баумана
terekchow@bmstu.ru

1 Введение

В настоящее время в целом ряде задач активно применяются алгоритмы распознавания лиц. Особенно важное место такие алгоритмы занимают в задачах биометрической идентификации. К сожалению, современные алгоритмы распознавания лиц обладают весьма большой вычислительной сложностью, что обуславливает низкую производительность их реализаций на обычных процессорах. Это сильно сужает сферу применения таких алгоритмов. Необходимость производить распознавание лиц в больших объемах видеоинформации (например, поиск лиц террористов в видеопотоке, получаемом с камер наблюдения) приводит к необходимости радикального увеличения скорости распознавания лиц.

Одним из основных подходов к распознаванию является использование нейросетевых алгоритмов [1-3]. Такие алгоритмы хорошо распараллеливаются, что делает возможной их эффективную реализацию на графических процессорах [4, 6, 7, 10]. Графические процессоры в настоящее время активно используются для разнообразных вычислений в задачах математического моделирования. В ряде работ [5, 8, 9, 11, 12] рассматривалось их применение

для моделирования искусственных нейронных сетей. Однако для распознавания лиц они еще не были использованы.

В настоящей работе описывается реализация нейросетевых алгоритмов распознавания лиц, и приводятся данные о точности распознавания и быстродействии.

2 Методика

Для экспериментов мы использовали фотографии лиц 50 студентов. Для каждого студента были использованы фотографии, сделанные под разными углами. Студентов просили придать лицам различные выражения. Всего для каждого испытуемого было получено около 100 фотографий. Все изображения имели размер 128x128 пикселей и представляли собой растровые фотографии в градациях серого.

Для распознавания образов использовались классические нейронные сети. В качестве алгоритма обучения был выбран алгоритм обратного распространения ошибок [1-3].

Нейронная сеть представляет собой классический многослойный полносвязный перцептрон. В качестве функции активации нейронов выбрана сигмоида:

$$f(x) = \frac{1}{e^{-cx} + 1}. \qquad (1.1)$$

Количество скрытых нейронов было выбрано равным 1024.

Решалась задача обучения сети для каждого испытуемого в отдельности. Для каждого испытуемого, половина изображений была использована для обучения сети, а оставшаяся половина – для тестирования.

Входом нейронной сети являлось изображение. Входом каждого нейрона входного слоя являлся один пиксел. Значение яркости каждого пиксела было представлено в виде действительного числа из диапазона $[0;1]$.

Выходом нейронной сети является переменная, принимающая значения из диапазона [0;1]. В том случае, если выходное значение превышало 0.85, делался вывод о том, что поданное на вход изображение принимается, а в случае, если выходное значение было меньше 0.15 – отвергается.

3 Реализация

В качестве платформы для реализации был выбран стандарт OpenCL. Эксперименты проводились на графическом адаптере AMD Radeon HD7790. При этом для функционирования каждого нейрона использовался отдельный поток.

В качестве реализации сравнения использовалась реализация для CPU, которая была запущена на процессоре Intel Core 7 3770. Эта реализация использовала восемь потоков.

Программы были разработаны на языке C++ и скомпилированы с помощью компилятора, входящего в среду Microsoft Visual Studio 2012.

4 Результаты

Полученный в результате экспериментов график зависимости частоты ошибок распознавания от числа шагов обучения приведен на рис. 1. Задавшись уровнем ошибок в 1%, исходя из этого графика, выберем число шагов обучения равным 30.

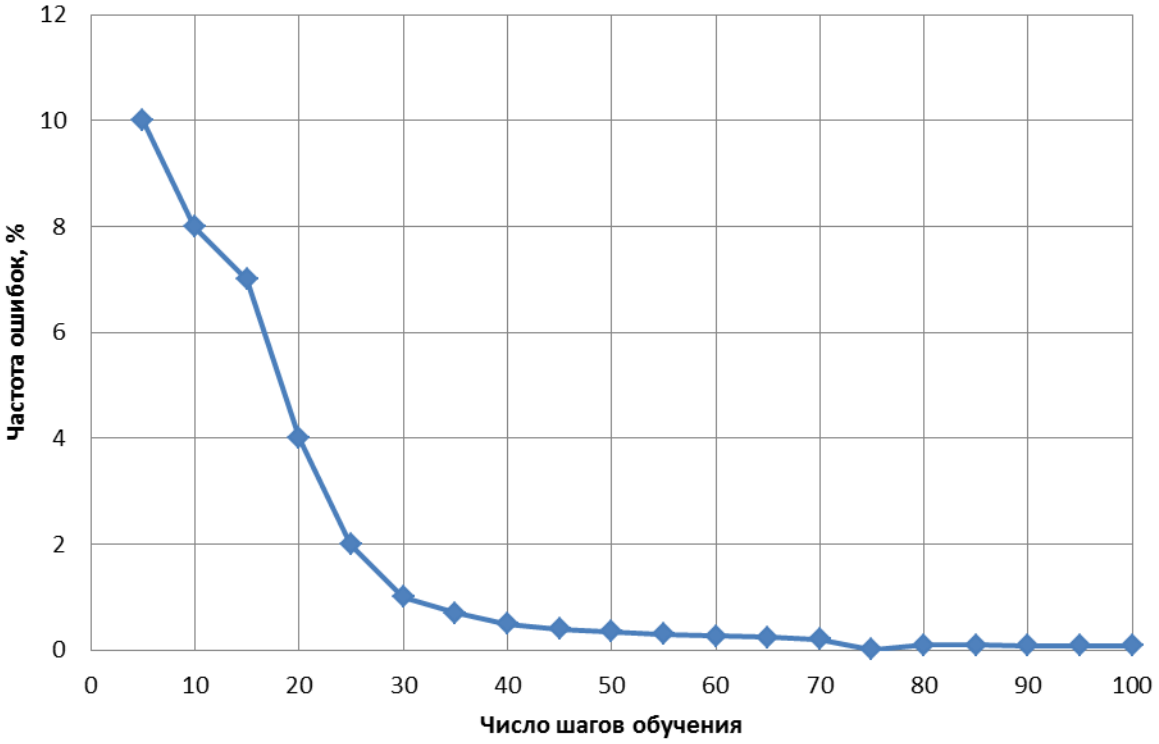

Рис. 1 – зависимость частоты ошибок от числа шагов обучения

В ходе эксперимента было произведено сравнение времени обучения при 30 шагах обучения. Полученные результаты приведены на рис. 2. Из него видно, что реализация на GPU имеет приблизительно в 7 раз большую производительность, по сравнению с реализацией на CPU.

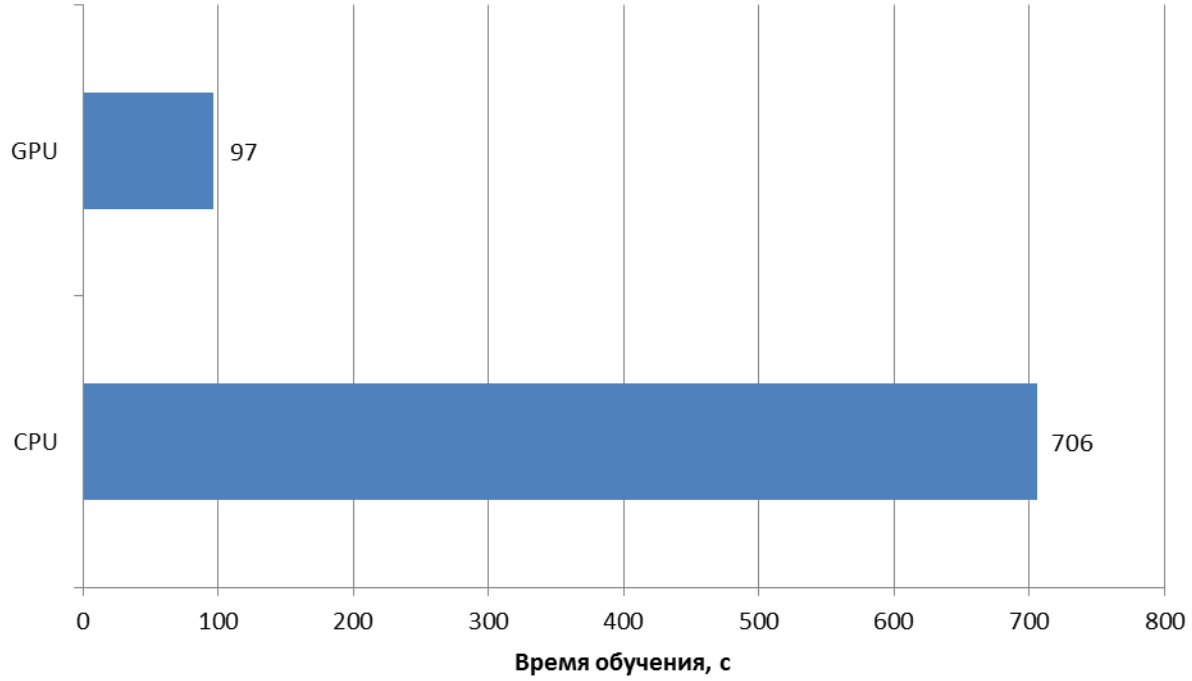

Рис. 2 – Время обучения.

5 Заключение

Таким образом, в работе показано, что с помощью графических процессоров можно произвести значительное (приблизительно в семь раз) ускорение работы нейросетевых алгоритмов распознавания. Этот результат указывает на перспективность реализации нейросетевых алгоритмов на графических процессорах.

Список литературы

1. Дулесов А.С. Нейронные сети и нейрокомпьютеры в интеллектуальных информационных системах : учеб. пособие. Абакан: Изд-во Хакас. гос. ун-та им. Н. Ф. Катанова, 2005. 113 с.

2. Круг П.Г. Нейронные сети и нейрокомпьютеры : учеб. пособие по курсу "Микропроцессоры" для студентов, обучающихся по направлению "Информатика и вычислительная техника". М.: Изд-во МЭИ, 2002. 176 с.

3. Лисс А.А., Степанов М.В. Нейронные сети и нейрокомпьютеры : учеб. пособие. СПб.: СПб. гос. электротехн. ун-т, 1997. 61 с.

4. Gaster B. Heterogeneous computing with OpenCL. Waltham, MA: Morgan Kaufmann, 2013. 291 p.

5. Jang H., Park A., Jung K. Neural network implementation using CUDA and OpenMP // 2008 Digital Image Computing: Techniques and Applications (DICTA). IEEE Conference Publications, 2008. P. 155-161. DOI: 10.1109/DICTA.2008.82

6. Kowalik J.S., Puźniakowski T. Advances in parallel computing. Using OpenCL : programming massively parallel computers. Amsterdam; Washington, DC: IOS Press, 2012. 295 p.

7. Munshi A. OpenCL programming guide. Upper Saddle River, NJ: Addison-Wesley, 2012. 603 p.

8. Nageswaran J.M., Dutt N., Krichmar J.L., Nicolau A., Veidenbaum A.V. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors // Neural Networks. 2009. Vol. 22, no. 5. P. 791-800.

9. Oh K.-S., Jung K. GPU implementation of neural networks // Pattern Recognition. 2004. Vol. 37, no. 6. P. 1311-1314.

10. Scarpino M. OpenCL in action : how to accelerate graphics and computation. Shelter Island, NY: Manning, 2012. 434 p.

11. Seiffert U. Artificial neural networks on massively parallel computer hardware // Neurocomputing. 2004. Vol. 57. P. 135-150.

12. Sierra-Canto X., Madera-Ramirez F., Uc-Cetina V. Parallel training of a back-propagation neural network using CUDA // 2010 Ninth International Conference on Machine Learning and Applications (ICMLA). IEEE Conference Publications, 2010. P. 307-312. DOI: 10.1109/ICMLA.2010.52

SCIENTIFIC PERIODICAL OF THE BAUMAN MSTU

SCIENCE and EDUCATION

EL № FS77 - 48211. №0421200025. ISSN 1994-0408

electronic scientific and technical journal

GPU realization of a neural network face recognition algorithm

 # 12, December 2013

DOI: 10.7463/1213.0659387

Terehov V.I.

Bauman Moscow State Technical University, 105005, Moscow, Russian Federation

terekchow@bmstu.ru

In this paper the author considers implementation of a neural network face recognition algorithm on a graphical processing unit. The algorithm is based on a classic multilayer perceptron. The training method is backward propagation of errors. This implementation has a seven times performance boost compared with traditional CPU implementation. Error rate of recognition approximately equals to 1%.

Publications with keywords: neural network, graphics processors

Publications with words: neural network, graphics processors

References

1. Dulesov A.S. *Neyronnye seti i neyrokomp'yutery v intellektual'nykh informatsionnykh sistemakh* [Neural networks and neurocomputers in intelligent information systems]. Abakan, Katanov KhSU Publ., 2005. 113 p.

2. Krug P.G. *Neyronnye seti i neyrokomp'yutery* [Neural networks and neurocomputers]. Moscow, MEI Publ., 2002. 176 p.

3. Liss A.A., Stepanov M.V. *Neyronnye seti i neyrokomp'yutery* [Neural networks and neurocomputers]. St. Petersburg, Publ. of St. Petersburg Electrotechnical University "LETI", 1997. 61 p.

4. Gaster B. *Heterogeneous Computing with OpenCL*. Waltham, MA, Morgan Kaufmann, 2013. 291 p.

5. Jang H., Park A., Jung K. Neural network implementation using CUDA and OpenMP. *2008 Digital Image Computing: Techniques and Applications (DICTA'08).* IEEE Conference Publications, 2008, pp. 155-161. DOI: 10.1109/DICTA.2008.82

6. Kowalik J.S., Puźniakowski T. *Advances in Parallel Computing. Using OpenCL : Programming Massively Parallel Computers*. Amsterdam; Washington, DC, IOS Press, 2012. 295 p.

7. Munshi A. *OpenCL Programming Guide*. Upper Saddle River, NJ, Addison-Wesley, 2012. 603 p.

8. Nageswaran J.M., Dutt N., Krichmar J.L., Nicolau A., Veidenbaum A.V. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Networks*, 2009, vol. 22, no. 5, pp. 791-800.

9. Oh K.-S., Jung K. GPU implementation of neural networks. *Pattern Recognition*, 2004, vol. 37, no. 6, pp. 1311-1314.

10. Scarpino M. *OpenCL in Action : How to Accelerate Graphics and Computation*. Shelter Island, NY, Manning, 2012. 434 p.

11. Seiffert U. Artificial neural networks on massively parallel computer hardware. *Neurocomputing*, 2004, vol. 57, pp. 135-150.

12. Sierra-Canto X., Madera-Ramirez F., Uc-Cetina V. Parallel training of a back-propagation neural network using CUDA. *2010 Ninth International Conference on Machine Learning and Applications (ICMLA).* IEEE Conference Publications, 2010, pp. 307-312. DOI: 10.1109/ICMLA.2010.52