Optimizing DNN Implementations on GPU and Heterogeneous Platforms

Ziyang Qi, Shuangchen Li, Yuan Xie

University of California, Santa Barbara, CA 93106

http://seal.ece.ucsb.edu/ SEAL@UCSB

Abstract

Leading increased performance in benchmark tasks and to enable discovery of complex high-level features, scaling up deep learning algorithms have earned researchers' interests. Recent works train deep neural networks with deep layers and very high dimension of parameters with the assistance of vast amount of computing power. The Graphics Processing Units (GPUs) have been applied successfully in many areas for parallel computing in recent years. Compared with the traditional CPU cluster, GPU has an obvious advantage of low cost of hardware and electricity consumption.

Deep learning algorithms including Deep Neural Networks (DNN), whose forward and backward propagation contains many inner products or matrix multiplications, which can utilize the parallelism of GPU.

In this work, we are going to perform optimizations on the implementations of DNN on GPU and heterogeneous platforms including AMD APU, Intel CPU + Nvidia GPU, AMD CPU, and Intel CPU. Evaluation and analysis of the implementations are compared on respective platforms.

Related Work

In order to tackle both the speed and energy challenges, there are a plenty of work implement DNN (or other similar machine learning algorithms) on parallel computing platforms, i.e., GPGPU, FPGA or ASIC solutions, which is discussed below.

GPU Solution

In recent years, the use of graphics processing units (GPUs) becomes a significant advance to speed up the training process of large scale neural networks by taking advantage of the massive parallelism capabilities of GPUs. Li et al. [1] explores the potential parallelism of the recurrent neural network and propose a fine-grained two-stage pipeline implementation and their GPU implementation can achieve 2X-11X speed-up compared with the basic CPU implementation. Chen et al. [2] implement a variant of the deep belief network (DBN) on NVIDA's Tesla K20 GPU and their GPU implementation results 7 to 11 times speedup over the CPU platform. Moreover, Lopes et al. [3] implement a multi-core GPU parallel version of the CD-k algorithm, which drastically reduces the pre-training time. With careful design, their approach is vital to obtain speedups of up to 46X.

FPGA/ASIC Solution

Since training large-scale datasets is time consuming, some fast FPGA implementations have been suggested in the literatures. Kim et al. [4] proposes a fully pipelined parallel FPGA architecture that combines many input cases to compute each set of weight updates to accelerate ANN training time, and a 100-fold acceleration versus CPU platform is obtained. Chen et al. [5] proposed an accelerator for Deep Learning (CNNs and DNNs) with small neural networks. Later they introduce a cus- tom multi-chip machine-learning architecture [6] and show that for large neutral networks their implementation is possible to achieve a speedup of 450.65x over a GPU, and reduce the energy by 150.31x on average for a 64-chip system.

Neuromorphic Computing Solution

New computing paradigms such as neuromorphic computing are attractive since it takes advantage of massive parallelism that comes from the distributed computing and localized storage in neural networks. Circuit implementation of neural networks is an active research area with several neural platforms successfully deployed and tested on real world problems [7]. For example, the spiking neural network architecture (SpiNNaker) project aims to deliver a massively parallel million- core computer whose interconnect architecture is inspired by the connectivity characteristics of the mammalian brain. The SpiNNaker platform [8] is a fully digital implementation supporting a wide range of neuron models (MLP, IZH, LIF, etc.), providing high scalability (up to 1000 neurons per core), and offering low- power consumption of 12 to 45 nJ/ms per neuron. It is inspired by the model of human brain [9], which is able to process very complex task instantly within 10W (comparing with 10MW of supercomputers). Neuromorphic computing is also inherently error- tolerant, thus it is especially attractive for applications such as im- age or speech recognition which involve input data sets in a changing and indeterministic environment. In 2011, IBM demonstrated SRAM crossbar based "brain processor" [10, 11]. Their follow-up work leverages digital technology to provide an accelerated simulation platform with up to 512 neurons and 100 000 synapses per core [12].

Goal

In this work, we perform an implementation of DNN in an application of hand digits recognition tasks on both CPU, CPU+GPU, and heterogeneous platforms. Supported by experimental results, we compare the cons and pros of DNN's implementation on different platforms. Further, our future work includes obtaining more DNN's optimization techniques, and make comparisons of their respective speedup on different platforms.

We aim at mapping Neural Networks Algorithms to APU platforms, with solid implementations and optimizations, and make comparisons with other GPU, CPU platforms.

We utilized a fully implemented CNN on both platforms to perform the hand-written digit recognition task, on MNIST Dataset.

Experiments

We implemented MLP with fully connected feed-forward neural network, and also a CNN with 2 convolutional layers on different of platforms, including CPUs, CPU and GPU, and APU. With the help of OpenCL, we do almost the same programming on different platforms. We measured the data of convergence speed and time efficiency on different platforms.

Conclusions

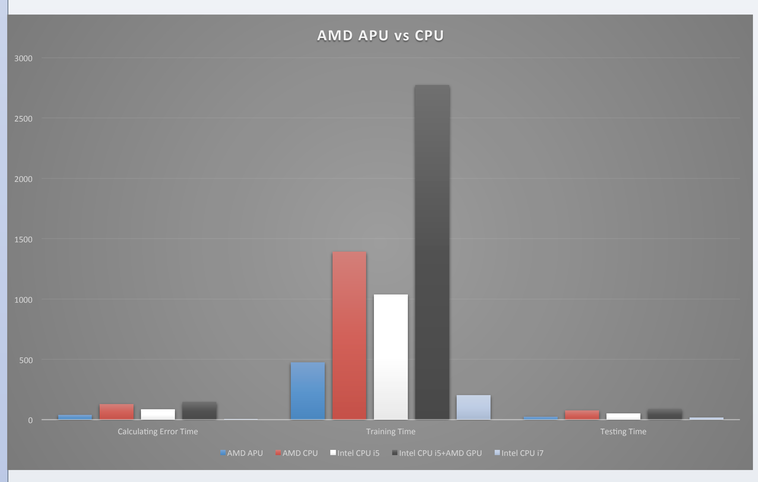
The projects are implemented in C++ with OpenCL on Linux Systems. On average, in each training iteration, training time is 473.52s, and error testing time is 37.56s. And for the each testing iteration, the average testing time is 23.67s. For AMD CPU, we also have respective values.

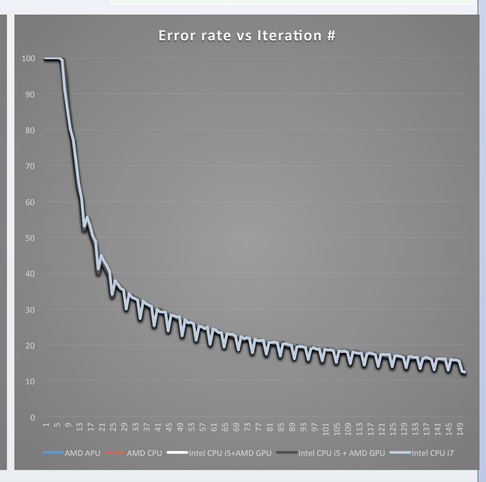
Taking advantage of APU accelerations, the speed-ups in calculating error time, training time and testing time are 3.42x, 2.94x, and 3.15x.

Future Work

More platforms for comparation, and optimizations from more aspects, for comparing difference in acceleration on different platforms.

Platform Specs	
aAMD CPU	AMD CPU SteamrollerB @ 3.7 GHz
AMD APU	AMD APU A10-7850B R7
Intel CPU	Intel Core i7-4770 CPU @ 3.40GHz
Intel CPUs +GPUs	Intel Core i7-4770 CPU @ 3.40GHz + GeForce GTX 645 OEM





Partial References

1. B. Li, E. Zhou, B. Huang, J. Duan, Y. Wang, N. Xu, J. Zhang, and H. Yang, "Large scale recurrent neural network on gpu," in *International Joint Conference on Neural Networks (IJCNN),*, July 2014, pp. 4062–4069.
2. Z. Chen, J. Wang, H. He, and X. Huang, "A fast deep learning system using gpu," in *IEEE International Symposium on Circuits and Systems (ISCAS),*, June 2014, pp. 1552–1555.
3. N. Lopes and B. Ribeiro, "Towards adaptive learning with improved convergence of deep belief networks on graphics processing units," *Pattern Recogn.*, vol. 47, no. 1, pp. 114–127, Jan. 2014.
4. L.-W. Kim, S. Asaad, and R. Linsker, "A fully pipelined fpga architecture of a factored restricted boltzmann machine artificial neural network," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 1, pp. 5:1–5:23, Feb. 2014.
5. T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS, 2014, pp. 269–284.
6. T. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, J. Li, Y. Chen, Z. Xu, N. Sun, and O. Temam, "Dadiannao: A machine-learning supercomputer," in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO, 2014, pp. 1–12.
7. J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomput.*, vol. 74, no. 1-3, pp. 239–255, Dec. 2010.
8. E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits,*, vol. 48, no. 8, pp. 1943–1953, 2013.
9. J. E. Smith, "Efficient digital neurons for large scale cortical architectures," in *Proceeding of the 41st annual international symposium on Computer architecuture*. IEEE Press, 2014, pp. 229–240.
10. J.-s. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha *et al.*, "A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *IEEE Custom Integrated Circuits Conference (CICC),*. IEEE, 2011, pp. 1–4.
11. P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm," in *IEEE Custom Integrated Circuits Conference (CICC),*, 2011, pp. 1–4.
12. J. Arthur, P. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S. Esser, N. Imam, W. Risk, D. Rubin, R. Manohar, and D. Modha, "Building block of a programmable neuromorphic substrate: A digital neurosynaptic core," in *The International Joint Conference on Neural Networks (IJCNN),*, June 2012, pp. 1–8.

Contacts

Potential Cooperations: Please Contact, Ziyang Qi (ziyangqi@ece.ucsb.edu) or Yuan Xie(yuanxie@ece.ucsb.edu)