**Deep Learning with HPC-ABDS**

Judy Qiu

Indiana University

Deep Learning has advanced dramatically over the last ten years [1-7] and shown remarkable achievements in areas including the recognition and classification of images, speech and handwriting (see Table 1). Some emerging critical challenges include the development of a general easy to use library on which an environment can be built to explore and develop new learning networks. The networks can address new problem areas such as analysis of exascale visualizations and new algorithms for different (all) parts of the system. At the highest level, one needs to evaluate Stochastic Gradient Descent and other Steepest Descent approaches to see if they can be used with larger sample sizes so that efficient parallelism is possible over input data sets such as images. Currently the pioneering work on 64 GPU’s [12] is harder to run on larger systems as it exploits parallelism over pixels and communication overheads grow as number of pixels assigned to each node declines, a well-known parallel computing issue. At a lower level we need to enable rapid experimentation with different network structures – number of layers and linkage between layers. The first step is development of an attractive “problem solving environment” for this issue that makes it as convenient as possible to modify algorithms and develop new applications while executing each experiment quickly. We have started to address this in collaboration with Stanford/Baidu and the University of Tennessee at Knoxville. This initial work is focused on enhancing the popular Caffe system from UC Berkeley. It will provide optimized kernels that run in parallel on systems with multiple CPU’s and GPU’s. We will package these kernels in a Python front-end that can support the specification of neural and manage its execution on an HPC system as well as its visualization/analysis. This problem is an excellent model to test and compare the different approaches to exascale runtime. One extension is to also support kernels using the ParalleX runtime developed in the CREST center at Indiana University. Further different execution frameworks besides Python could be considered; for example it is not possible to use a Python scripting approach without also parallelizing all the data analysis steps necessary to evaluate each execution. At the most ambitious level, this could involve a parallel Python invoking the optimized runtime described above. In any case, one needs modules for the final analysis stage that run in parallel and handle large networks, such as a recent run that was performed with 11 billion parameters. Our current work aims at producing a single benchmark based on ImageNet, however, this is not sufficient to even provide a proper set of requirements. We need to survey deep learning applications of interest to DoE and define both system requirements and benchmarks.

My group has pioneered the development of HPC-ABDS which is an integration of HPC and Apache (and other) open source big data technologies ABDS; our current catalog has identified 200 software subsystems divided into 17 layers summarized in Figure 1. The key idea is that new HPC ideas should be developed so they integrate well with ABDS rather than competing with this rapidly developing software stack which has a clear vitality and innovation with a sustainable software model. In particular I have shown that previous standalone enhanced versions of MapReduce can be replaced by a Hadoop plug-in that offers useful scientific data abstractions, high performance iteration and communication using best available (MPI) approaches. We suggest that deep learning be set up in this architecture exploiting resource managers like Yarn, as well as storage models such as HDFS and MongoDB/HBase. This would enable us to look at approaches such as Apache Pig (data parallel language) and Crunch (workflow) to drive the deep learning explorations.



Table 1. Summary of recent research progress in Large-Scale Deep Learning

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Computing Power** | **Number of Examples and Free Parameters** | **Average Running Time** |
| DBN [8] | NVIDIA GTX 280 GPU with a GB memory | One million images and 100 million parameters○ | ~ 1 day |
| CNN [9] | Two GTX 580 GPUs, each with 3GB memory | 1.2 million high resolution (256 x 256) images and 60 million parameters○ | ~ 5-6 days |
| DisBelief [10] | 1,000 CPUs with Downpour SGD with Adagrad | 1.1 billion audio examples and 42 million model parameters○ | ~ 16 hours |
| Sparse autoencoder [11] | 1,000 CPUs with 16,000 cores | 10 million 200 x 20 pixel images and one billion parameters | ~ 3 days |
| COTS HPC [12] | 64 NVIDIA GTX 680 GPUs, each with 4GB memory | 10 million 200 x 200 images and 11 billion parameters | ~ 3 days |

Figure 1. Kaleidoscope of HPC-ABDS

**References**

[1] G. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527-1554.

[2] Y. Bengio, P. Lamblin, D. Popovici, et al, Greedy layer-wise training of deep networks. Proc. Of the 12th Annual Conference on Neural Information Processing System. 2006:153-160.

[3] Y. Lecun, L. Bottou, Y. Bengio, et al, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[4] Markoff, J. (2012). How many computers to identify a cat? 16,000. *New York Times*.

[5] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, *29*(6), 82-97.

[6] Chen, X. W. & Lin, X. Big Data Deep Learning: Challenges and Perspectives.

[7] Bengio, Y. (2013). Deep learning of representations: Looking forward in *Statistical Language and Speech Processing* (pp. 1-37). Springer Berlin Heidelberg.

[8] R. Raina, A. Madhavan, and A. Ng, ‘‘Large-scale deep unsupervised learning using graphics processors,’’ in Proc. 26th Int. Conf. Mach. Learn., Montreal, QC, Canada, 2009, pp. 873–880.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in Proc. Adv. NIPS, 2012, pp. 1106–1114.

[10] J. Dean et al., ‘‘Large scale distributed deep networks,’’ in Proc. Adv. NIPS, 2012, pp. 1232–1240.

[11] Q. Le et al., ‘‘Building high-level features using large scale unsupervised learning,’’ in Proc. Int. Conf. Mach. Learn., 2012.

[12] A. Coats, B. Huval, T. Wng, D. Wu, and A. Wu, ‘‘Deep Learning with COTS HPS systems,’’ J. Mach. Learn. Res., vol. 28, no. 3, pp. 1337–1345, 2013.