A Beginners Guide to Perfomance Factors of Machine Learning Learning Applications with Focus on Neural Nets

Johannes Wünsche
Otto-von-Guericke University Magdeburg
Email: johannes.wuensche@st.ovgu.de

***Abstract*—**

- **Deliver short introduction for neural networks with different aspect focused examples**
- **What we want to show in this paper(performance factors)**
- **To what end we came**
- **some other stuff that comes up during the paper and is important enough to be named at the beginning**

I. Introduction

BE A BIT PITCHIER!

Machine Learning seems to everywhere around us. With the success of Deep Learning, Convolutional, Deep Belief, and other Neural Net specifications in the recent years this trend continues to be more and more influencing everyday application, may it be from music recommendation to personalized advertisement directly addressed to the consumer. But also in modern automation and research machine learning becomes a greater and greater field. We want to have a look at the most performance influencing factors while designing and implementing these networks.

- Deliver a short introduction coining the expressions neural network(deep learning , recurrent, deep belief ...) and explain to focus in the further paper on deep learning
- To what extend did we explore the performance of machine learning related to which factors and specialization of specific kind of nets
- Show which methods were used to explore neural nets
- Name the result again that was reached at the end

II. Choice of Calculation Unit

The next important choice is the choice of the actual operating calculation unit. We distinguish them into three main models, CPU, GPU and GPU-Clustering.

A. CPU

As the first most naive option we looked on the CPU for training of the neural net with the simple single thread and more complex multi-thread calculation.

1) Single-thread: The most naive implementation is the single-thread CPU calculation, because oft this more brute force approach and sequential execution and calculation this is also the slowest and with a larger number of connected nodes simply unfeasible because of the excessive calculation time required *NUMBER OF EXAMPLE CAN BE ADDED HERE.*

2) Multi-thread: A more complex approach consist of utilizing the multi-core characterisitics of modern CPUs, by splitting of training calculation with only requirements to already calculated values. This is for example preferrable when training with the help of a *Backpropagation* algorithm. Though this approach is limited by the actual core number available and should not overstep the number of physical cores, like with intel *hyper-threading* technology, which can lengthen the required calculation time because of a more inefficient usage[link performance paper here(believe it was benchmarking state of the art neural network frameworks)]

B. GPU

A more advanced implementation is to use the shared memory, multi-core environment of modern GPUs. For example with NVIDIAs CUDA this can be done efficiently and result in a speed-up of calculation of up to 3 times the naive approachs performance. Allmost all recent neural net learning frameworks support this calculation unit because of its excellent performance rating for neural networks.

Problems lie in the tightly restricted memory of graphic cards that may not be able to contain all nodes with their corresponding weights as well as training dataset. This leads to some communication overhead depending on the communication strategy chosen.

III. Communication strategy

Here I want clarify a bit on strategies to create an efficient CPU-GPUs interchange with as little overhead as possible.

IV. Conclusion

yeah ... pretty much the conclusion shortly repeating the basis that is important for it and point the way they were created, should be larger part to emphasize the importance of some factors over others and the reasoning behind this classification

Acknowledgment

The authors would like to thank...

References

[1] pap, S. Shi, Q. Wang, P. Xu and X. Chu, *Benchmarking State-of-the-Art Deep Learning Software*, HK Baptist University, Hong Kong, 17th Feb 2017