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

Machine Learning seems to be everywhere around us. With the recent success of Deep, Convolutional and other Neural Networks, these kind of Machine Learning tools experience a kind of renaissance and usage in applications from industrial to service oriented. But also in modern automation and research machine learning becomes a greater field managing to find more efficient ways to achieve better results in many expertise. We want to have a look at the most performance influencing factors of theses networks while designing and implementing them for specific applications for both small and large scale. At first we attend the choice of neural network and the impact of this decision. Afterwards we chose to have a look, on one hand, at small scale neural networks running on a single machine without large data sets and a low query frequency after the initial training, and at the other hand, large scale neural networks operating with more layers and nodes, large data sets and distributed system to handle the computational power required by the neural network training algorithms to be completed in reasonable time.

- 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 Processing 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.

FW 1 Thread 4 Threads

Fig. 1. Simulation results for the network.

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 [1].

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 CPU-based approachs performance for deep learning algorithms over optimized floting-point baseline [1]. 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 to clarify a bit on strategies to create an efficient CPU-GPUs interchange with as little overhead as possible.

TODO: Spanish Paper on SuperComputer Learning

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 [1]

TODO: Takeaway, different ideas, innnovative

Acknowledgment

The author would like to thank Gabriel Campero Durand for advise and help to write the paper

References

[1] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking state-of-the-art deep learning software tools," *arXiv preprint arXiv:1608.07249*, 2016.