ACCELERATING THE TRAINING OF ARTIFICIAL NEURAL NETWORKS USING DATA PARALLELIZATION

Jorge López

Distributed Systems and Multimedia Processing Laboratory (DSMP)
Department of Computer Science
Toronto Metropolitan University
245 Church Street, Toronto ON, Canada

ABSTRACT

In training deep neural networks, we address the research question of how to accelerate their training for edge devices. We explore *Data Parallelization* (DP) and *"CPU-affinity"* (CAF) as techniques that can help to achieve this goal. Even though GPUs are the de facto standard for training a network, they are expensive and may not be used for edge computing, we want to demonstrate that the usage of CPUs still provides a significant acceleration when parallelizing the training of a network.

Keywords: Data Parallelization, Distribution, CPU-affinity

1 INTRODUCTION

Today, the training time of NNs has grown to the point that it deters research and development. This is because massively large datasets have become available to train NNs. In this work, we analyze *DP* and *CAF* as techniques for making it possible to reduce the time it takes to train them.

2 EXPERIMENTS WITH CONVOLUTIONAL NEURAL NETWORK (CNN) FOR IMAGE DATA

Our experiments with CNN were performed using two programs to train it in serial and parallel; we used the MNIST dataset. Firstly, we trained the network serially in the cloud (Google Cloud Platform GCP) with an Intel-based CPU with 16GB of memory, obtaining a processing time of 749 seconds. Then, we did the same but using the parallel program, obtaining a time of 222 seconds. Representing a speed-up gain of 337.76%, we also experimented with one GPU (Nvidia GTX A100) and CPU (Intel Xeon CPU @ 2.30GHz) in parallel (using 2 cores) and serial. Figure 1 show the results of these experiments that show both training time and accuracy. As expected, the run with GPU is the fastest, with 144.90 seconds. All runs achieved excellent accuracy; we can also observe that even though the fastest processing time is using a GPU, the highest accuracy (99.11) occurred when we used the CPU in serial mode, suggesting that there may be a trade-off between speed and accuracy. We performed a further experiment to demonstrate that using CPUs instead of GPUs are still a viable option for parallelizing the training of a neural network, where we trained the CNN with the MNIST dataset (with 60000 images). The hardware that we used was a virtual machine with 16 cores, Intel CPU @ 2.90GHz, with one GPU GTX A100. We used the PyTorch framework to implement our program. Figure 1 shows the accuracy and processing time of training achieved using two processes. We notice that accuracy and processing times are very similar, with the accuracy slightly greater.

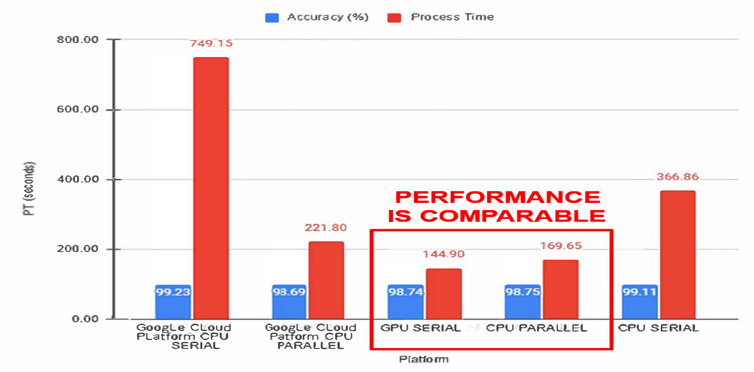

Figure 1: Training of CNN in serial and parallel in CPU and GPU

3 EXPERIMENTS WITH A TEXT NETWORK

The following set of experiments were implemented using the *DSMP* simulator varying the CPU-affinity. This simulator was developed in the *DSMP* laboratory, and simulates a distributed processing environment such as cloud computing for parallel running of a recommendation algorithm that can be used to test both scalability and accuracy of the employed algorithm. These experiments trained a Tweets network of $94,000$ posts varying the CPU-affinity that simulates a multiprocessors system. Table 1 shows the results of the experiment (3 repeats) while we keep the number of nodes fixed to 1 and we vary the CAF from 1..8. Here we can observe that the highest acceleration (199.91%) vs. the slowest node takes place when the CAF is set to 4 processors. We have used an Intel PC @3.3Ghz 60Gb RAM with 12 processors.

Table 1: Runs with simulator for N=1 and setting CPU-affinity from 1 to 8 processors

CPU-affinity (Number of processors used	PT in MS	PT% gain vs. slowest
$1(0)$	507,069	198.04
$2(0,1)$	1,004,192	SLOWEST
$4(0,1,2,4)$	502,318	199.91
$8(0,1,2,4,5,6,8,9)$	582,539	172.38

4 CONCLUSIONS

- For image data, CPU parallel training performance is comparable to do the same using GPU in serial.
- In the case of text data, training a DNN in parallel with $N = 1$ and varying the CAF from 1..8, we obtain the highest acceleration when we set the CAF to 4 processors, see Table 1 .Using CPU-affinity is similar to parallelization in multi-process systems that shows improvements with more cores.

REFERENCES

Abhari, Abdolreza and Li, Jason 2019. "DSMP Lab a Multi-agent System Simulator for Distributed Recommender System http://scs.ryerson.ca/~aabhari/MAS_Demo_Narration.mp4".

Keuper, J., and F.-J. Preundt. 2016. "Distributed training of DNNs". In *2016 2nd Workshop on ML in HPC Environments*, pp. 19–26. IEEE.