Parallel Convolutional Neural Networks power by GPU acceleration

Yu Gao
Dept. of Computer Science
USC Viterbi School of Engineering
Los Angeles, USA
gaoyu@usc.edu

Abstract—Convolutional neural networks (CNNs) now becomes the most popular statistical learning model in deep learning. In this paper, I will implement parallel CNN on multiple GPUs within a single machine and compare its performance with the CNN implemented on CPU. Framework such as Caffe will be used. CUDA and OpenCL will be used to accelerate operations such as convolution and matrix multiplications.

Keywords—Parallel CNN; GPU; CUDA; OpenCL; Deep Learning;

I. INTRODUCTION

Convolutional neural networks (CNNs) have been widely used in many domains. It is one of the most important and popular statistical models in deep learning. CNNs have made the breakthrough in natural language processing, audio speech recognition and image processing[1]. Convolutional neural networks are biologically-inspired by Multilayer perceptrons (MLPs). CNNs take advantage of spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. According to its structure, the inputs of layer \( m \) are from a subset of units in layer \( m - 1 \). However, the training of CNNs is a computing-heavy and time-consuming task. With the acceleration of GPU, we can short the time of operations like convolution, matrix multiplication. Thousands of research and papers have proven that Simple GPU-based CNN runs much faster than the simple CPU-based CNN. The research which will be presented here wants to examine the effect between parallel CNN accelerated by GPU and parallel CNN based on CPU.

II. TECHNIQUES AND METHODS TO BE USED

There are several types of parallelism in CNNs.

A. intra-layer parallelism

In this type of parallelism, the computations in different layers can be done simultaneously[2]. So the operation like matrix can be done in parallel in GPUs.

B. model parallelism[3]

In this type of parallelism, layers are partitioned in different machines or GPUs. Google developed a software framework called DistBelief to accelerate the training of very large deep networks and introduced the ‘model parallelism’ concept to the public[3].

C. data parallelism[3]

In this type of parallelism, large set of data are partitioned in different machines or GPUs. The famous example is MapReduce developed by Google.

In this paper, I decide to use the mixture of intra-layer parallelism and model parallelism to accelerate my CNN. CUDA will be used to accelerate the operations. As there is huge communication cost in above two parallelism models, the speed of training cannot be in proportion to the amount of GPUs. The figures offered by NVIDIA shows that the speed of training GoogleNet on 4 GeForce GTX TITAN X GPUs is 2.3x faster than on 1 same type of GPU instead of 4x. So I want to reach a balance between the performance and the scale of model.

In this course, I hope I can learn MPI and OpenMP that I can use in CPU control group. I will also try pthreads library in Linux to do the parallelism in CPU.

III. EXPECTED RESULTS

A. The Speed of training parallel CNN on GPU

B. The Speed of training parallel CNN on CPU

I expect that the parallel CNN running on GPU will get much faster than the parallel CNN on CPU.

REFERENCES

