Performance of CPUs/GPUs for Deep Learning workloads

Preprint · May 2018

DOI: 10.13140/RG.2.2.22603.54563

3 authors, including:

Amr Kayid
German University in Cairo
10 PUBLICATIONS **85** CITATIONS
SEE PROFILE

Yasmeen Khaled
German University in Cairo
4 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Media Engineering and Technology Faculty
German University in Cairo

Performance of CPUs/GPUs for Deep Learning workloads

Research Report

Author: Amr Kayid & Yasmeen Khaled
Supervisors: Dr. Mohamed Elmahdy

Submission Date: 07 May, 2018

Acknowledgments

This research report is an assignment from the *Computer System Architecture course*, to help us get familiar with the research field and get familiar with the current technologies in the field of Computer Architecture.

Abstract

Deep learning is a very computational intensive task that is known to demand significant computing horsepower. Traditionally GPUs have been used to speed-up computations by several orders of magnitude.

When it comes to training the models used in deep learning,the capability of developing new algorithms and improving the existing ones is determined by the speed at which these models can be trained and tested. and to gains significant performance, this can be done through hardware acceleration.

In this research paper, we will compare performance of CPUs/GPUs for different Deep Learning workloads and their effects to accelerate training the models, we will also examines current research on deep learning and it's performance and outlines some of the challenges and directions for future work.

Contents

Chapter 1

Introduction

Deep Learning [1] has revolutionized the machine learning recently with some of the great works being done in this field. It has dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection, natural language processing, recommendation systems, and many other domains [3].

Deep Learning is an active field of research too, nothing is settle or closed, scientists are still searching for the best models, topology of the networks, best ways to optimize their hyper-parameters and more.

There has been a lot of research going on in the field of Deep Learning and it's performance comparing both CPUs and GPUs to get the best outcomes for the long term algorithms and it's training time .[6].

The success of deep learning techniques for machine learning and artificial intelligence is directly related to three trends:

- New algorithms.
- Availability of big data
- Increasing computational power.

Improving one of these areas usually demands improvements in the others. and the huge computational demand from existing deep learning methods is driving a variety of new hardware solutions that emerge as deep learning application platforms.

Until the late 2000s, we were still missing a reliable way to train very deep neural networks. Nowadays, with the development of several simple but important theoretical and algorithmic improvements, *the advances in hardware mostly GPUs, now TPUs*, and the exponential generation and accumulation of data, Deep Learning came naturally to fit this missing spot to transform the way we do machine learning.

Hardware design has started to be shaped according to the needs of deep learning models with performance improvements that range from 10 to 100 times over conventional

computing systems. As a result, previously intractable research problems turned into overnight jobs, opening up new types of learning algorithms and research opportunities.

Deep learning involves huge amount of matrix multiplications and other operations which can be massively parallelized and thus sped up on GPUs. A single GPU might have thousands of cores while a CPU usually has less number of cores. Although GPU cores are slower than CPU cores, they more than make up for that with their large number and faster memory if the operations can be parallelized. Sequential code is still faster on CPUs [5].

Figure 1.1. Show the Performance of Deep Learning over the past 3 years comparing CPUs and GPUs performance.

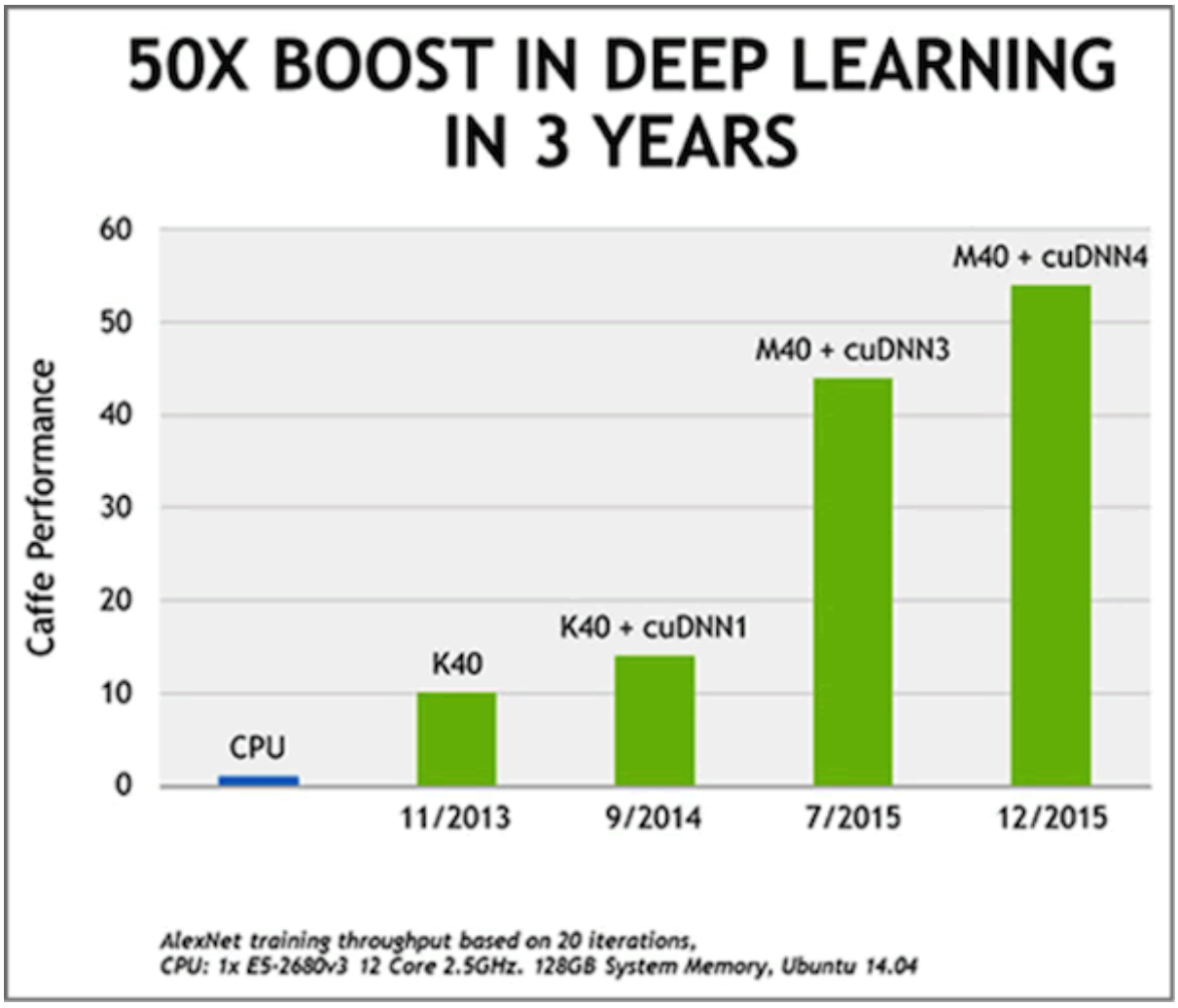

Figure 1.1: Deep Learning performance in 3 years [12]

Chapter 2

Background

2.1 Deep Learning

To understand what deep learning is, we first need to understand the relationship deep learning has with machine learning, neural networks, and artificial intelligence.

Artificial intelligence can be considered the all-encompassing umbrella. It refers to computer programs being able to think, behave, and do things as a human being might do them. Its usually classified as either general or applied/narrow (specific to a single area or action).

Machine learning goes beyond that. It involves providing machines with the data they need to learn how to do something without being explicitly programmed to do it. An algorithm such as decision tree learning, inductive logic programming, clustering, reinforcement learning, or Bayesian networks helps them make sense of the inputted data. Machine learning was a giant step forward for AI.

The development of neural networks *a computer system set up to classify and organize data much like the human brain* has advanced things even further.

The easiest way to think of their relationship is to visualize them as concentric circles with AI *the idea that came first, the largest*, then machine learning *which blossomed later*, and finally deep learning *which is driving todays AI explosion* fitting inside both.

Figure 2.1. Show the Relation Between A.I., ML and DL

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the brain structure called artificial neural networks brain model. that is capable of solving complex computational -signal processing- intensive problems related to A.I. Fields.

The origin of deep learning can be traced back to Rosenblatts perceptron, first published in 1957. The idea of deep learning is to have multiple association layers to describe more complex objects and this idea was deemed too complex and computationally expensive to realize.

The expression *deep learning* was first used when talking about Artificial Neural Networks (ANNs) by Igor Aizenberg and colleagues in or around 2000.

The key idea is for a system to be able to recognize, or perceive, a large number of objects without storing explicit information about these objects.

these neural networks are combination of many layers that needs to be trained using large dataset, and the training is mainly based on matrix multiplications that needs a lot of computational power.

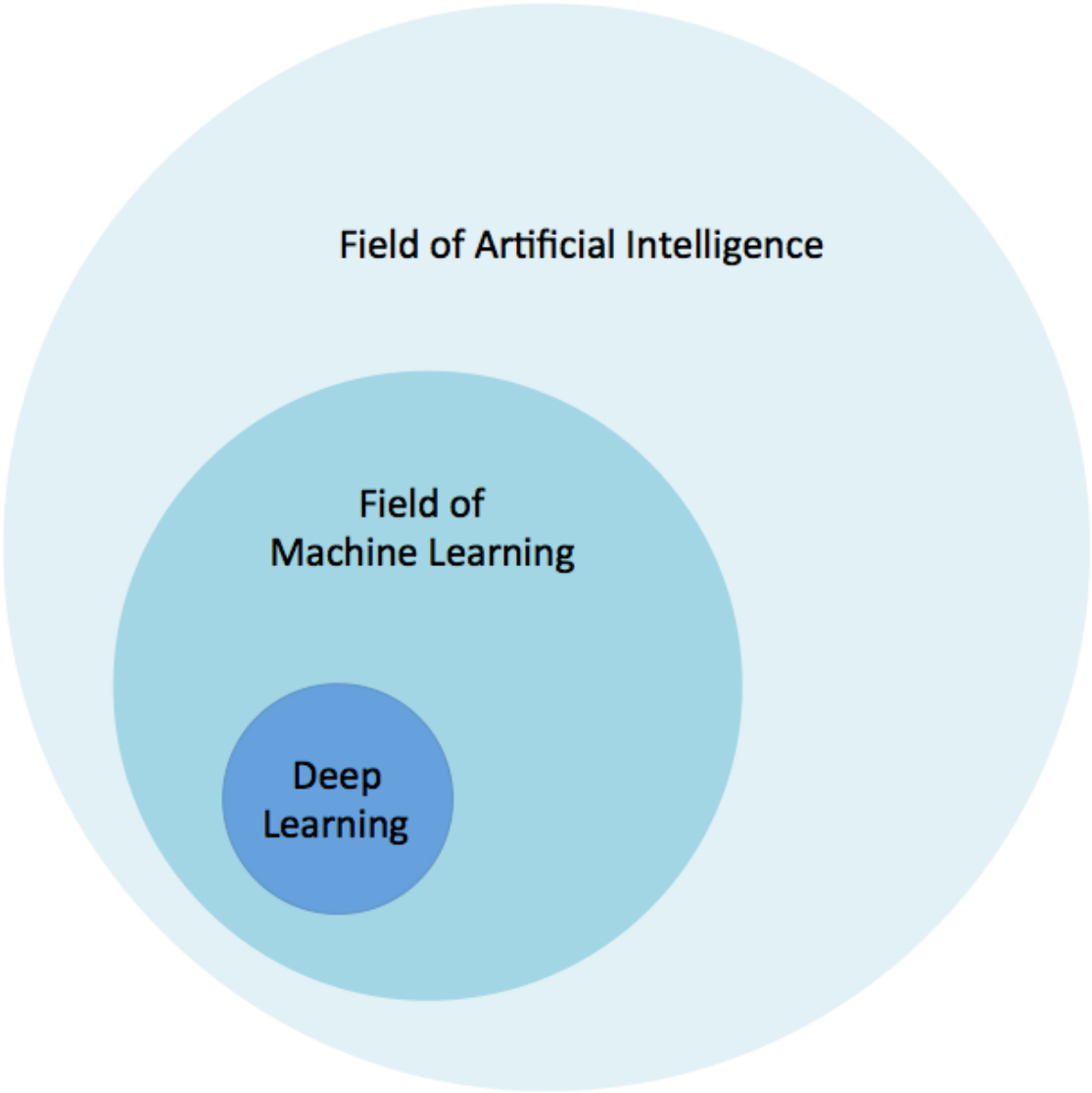

Figure 2.1: Relation Between A.I., ML and DL [13]

2.2 Training and Performance

Training a deep neural network needs gathering a very large labeled data set and design a network architecture that will learn the features and model. this might take from hours to weeks depending on the dataset, the computational power and the algorithms being used for the training.

Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

The *well-trained neural network* uses what it has learned to recognize images, spoken words, or suggest new stuff someone is likely to buy next. This speedier and more efficient version of a neural network infers things about new data it is presented based on its training.

Figure 2.2. Show the training of existing data and prediction of new data of deep learning

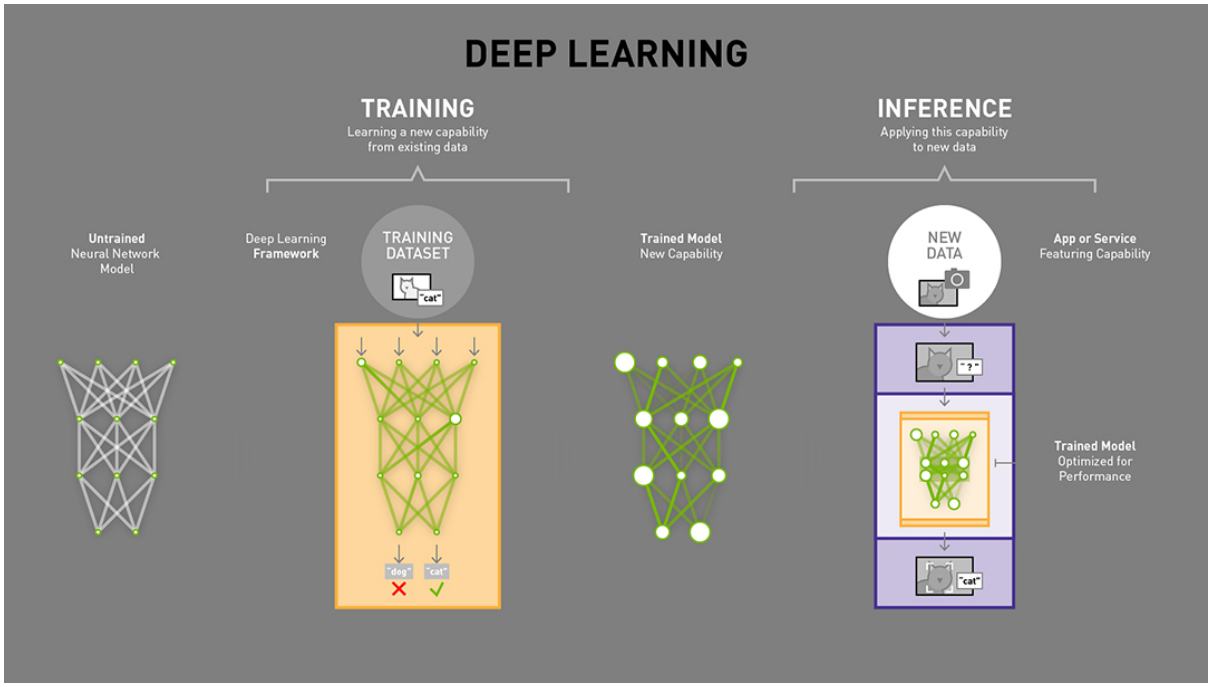

Figure 2.2: Deep Learning training & inference [10]

Inference can not happen without training. Like how we gain and use our own knowledge for the most part. And just as we do, inference does not require all the infrastructure of its training to do its job well. When training a neural network, training data is put into the first layer of the network, and individual neurons assign a weighting to the input based on the task being performed.

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks. The term *deep* usually

refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as **150**.

In an image recognition network, the first layer might look for *edges*. The next might look for *how these edges form shapes*. The third might look for particular features and so on. Each layer passes the image to the next, until the final layer and the final output determined by the total of all those weightings is produced.

Most deep learning applications use the **transfer learning approach**, a process that involves a pretrained model. where it starts with an existing network, such as ***AlexNet or GoogLeNet***, and feed in new data containing previously unknown classes. After making some tweaks to the network, you can now perform a new task, such as categorizing only dogs or cats instead of 1000 different objects. This also has the advantage of needing much less data processing thousands of images, rather than millions, so computation time drops to minutes or hours.

How Inferencing/Predictions Works

The first approach looks at parts of the neural network that do not get activated after it is trained. These sections just are not needed and can be pruned away. The second approach looks for ways to fuse multiple layers of the neural network into a single computational step.

What that means is that we all use inference all the time. for example, Your smartphones voice-activated assistant uses inference, as does Googles speech recognition, image search and spam filtering applications. Baidu also uses inference for speech recognition, malware detection and spam filtering. Facebooks image recognition and Amazons and Netflixs recommendation engines all rely on inference.

GPUs, with the advantages of their parallel computing capabilities *the ability to do many things at once* are good at both training and inference.

Systems trained with GPUs allow computers to identify patterns and objects as well as or in some cases, better than humans.

After training is completed, the networks are deployed into the field for classifying data to infer a result. Here too, GPUs offer benefits, where they run billions of computations based on the trained network to identify known patterns or objects.

We will discuss more about CPUs and GPUs architectures and how it affect the deep learning workloads in *chapter 3*

Chapter 3

Architecture and Performance

3.1 CPU Vs GPU Architecture

There is a major difference between the CPU and the GPU Architectures. A CPU *Central Processing Unit* consists of multiple ALUS *Arithmetic Logic Units* ,one Control unit to control those ALUS, a Cache memory and a DRAM *Dynamic Random Access Memory*.A standard CPU has between one and four processing cores clocked anywhere from 1 to 4 GHz.

However, GPU *Graphic Processing Unit* consists of hundreds of ALUS, many Control Units along with many Cache memories and one DRAM.It runs at a lower clock speed then a CPU and has as seen many more cores the CPU.

Figure 3.1. Show the Architecture of both CPU and GPU

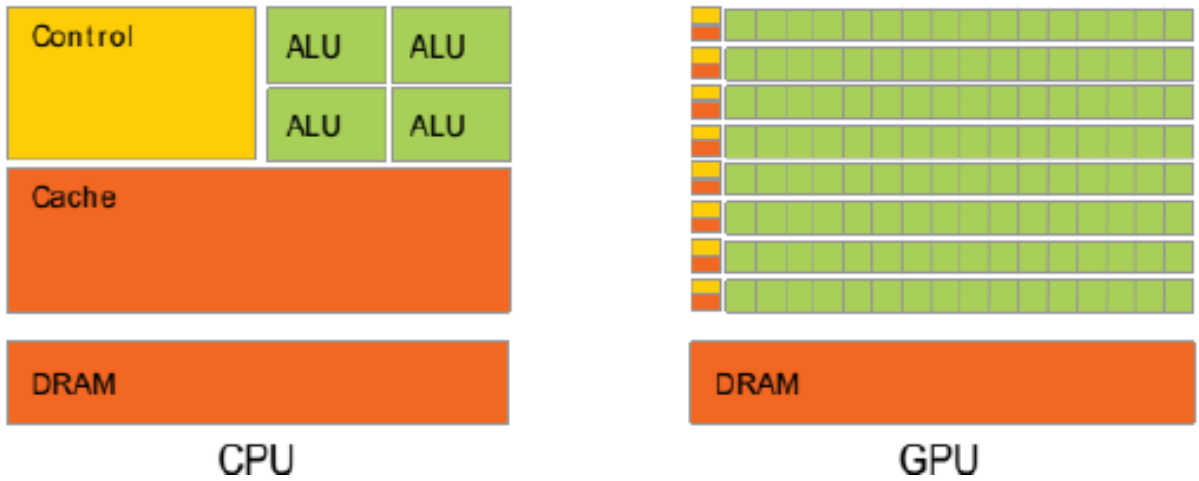

Figure 3.1: CPU & GPU Architecture

3.2 CPU Vs GPU general performance

It is obvious that the structural difference of the CPU and GPU aims to provide different performance, Sometime CPU gives better performance than GPU and sometimes GPU gives better performance than CPU ,however better here is relative to the application CPU and GPU will be used and they both function very differently.

Figure 3.2. Overview on how CPU and GPU works

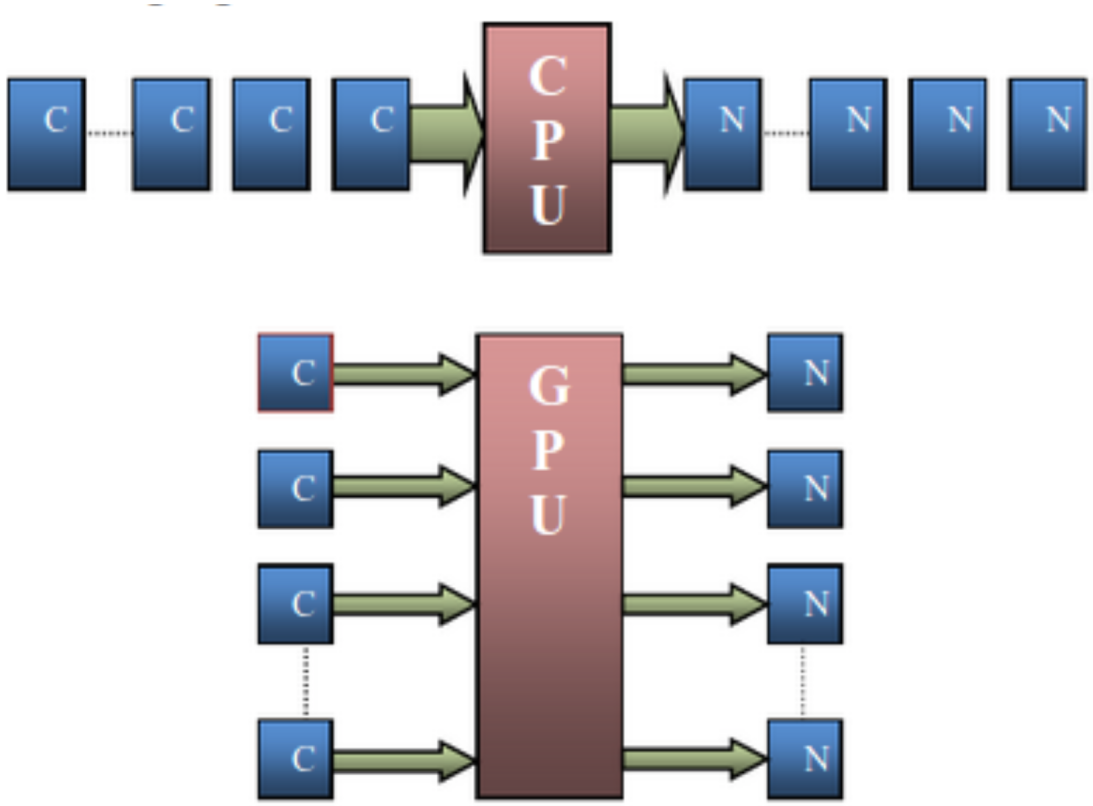

Figure 3.2: Overview on how CPU and GPU works

A CPU is very powerful and it is the reason why computers manages to do any task, they tend to be more flexible than GPUS and they include a larger instruction set , they run at higher clock speeds, they are responsible of IO operations of all computer components and they are responsible for the integration with virtual memory in the operating system which is something a GPU cant do .CPUs perform their tasks sequentially ie. they can do one thing at a time as shown in figure 2.This means they can perform serial algorithms very fast.

On the other hand, the GPU is optimized to display graphics and do very specific computational tasks. Even though the GPU doesnt have a large instruction set as the

CPU which make it able to only do a fraction of the operations the CPU can make, it performs these operations with a very high speed due its lower clock speeds, which makes it very useful at image processing. The GPU uses its hundreds of cores to perform time-sensitive calculations for thousands of pixels at a time, thus be able to display nearly any type of images including complex 3D graphics as well. The reason it can does these calculations in such a fast time is due to its architecture which includes hundreds of cores that allows it to perform multiple parallel operations at the same time as show in figure 2.Thus, GPUs tend to be very handy when parallelism will improve the performance of a certain operation.

Graphic processing is a repetitive and highly-parallel computing task , that needs to be done at high speeds . Also sometimes it needs multiple mathematical operations that needs to be done during processing of images.

3.3 CPU Vs GPU in *deep learning* performance

As mentioned above,CPUs have larger instruction set than GPUs making GPUs less flexible , however GPUs are said to be dedicated for parallel computing even for same instructions.Deep Neural Networks (DNN) are structured in a very uniform manner such that at each layer of the network thousands of identical artificial neurons perform the same computation.Therefore,its way of computation is quite similar to how GPU computes instructions.

As mentioned also,image processing is an expensive task that requires many calculations and since GPUs have more computational units and have a higher bandwidth to retrieve from memory ,GPUs perform quite well in that task with high speed as well,deep learning includes massive image processing operations with large datasets making parallelism a needed feature in deep learning computing .

The main weakness of GPUs as compared to CPUs is memory capacity on GPUs are lower than CPUs. The highest known GPU contains 24GB of RAM, in contrast, CPUs can reach 1TB of RAM. A secondary weakness is that a CPU is required to transfer data into the GPU card.CPU helps to feed GPU with enough data and read/write files from/to RAM/HDD during training. If the CPU is weak, it can only feed as few data as possible thus cant keep up with your powerful GPU. Ideally Deep Learning training systems should have CPU with maximum number of processing cores to handle more work to catch up with a GPU.In addition, GPU clock speeds are 1/3rd that of high end CPUs, so on sequential operations won't be as fast as if they were processed using a CPU.

However, on saying the real reason why GPUs are so efficient and fast in matrix multiplication and convolution,is not only parallelism but memory bandwidth as well.

We need to make it clear that CPUs are latency optimized while GPUs are bandwidth optimized.lets think of a CPU as a high speed car and a GPU as a large truck , so if their task is to pick some packages from a loacation x and transport them to loaction y.The

CPU can fetch those memory packages in the RAM faster than the GPU which has much higher latency.However, in the sense of the car and the truck , the car must go many times back and forth to do its job ,on the other hand, the turck can fetch much more memory packages at once.Strictly speaking, CPU can fetch small amounts of memory much faster than GPU which can fetch large amout of memory at slower rate.This means the larger the computational operations , the larger the advantages of GPUs over CPUs.But as said before, GPUs are not latency optimized , this means that latency might hurt the GPUs performance,this comes in the sense of the long waiting time until the different sets of packages arrive.However this problem was solved by thread paralellism.This means many trucks will be working at the same time , so you don't have to wait for the truck to unload , all trucks woring will just queue in the unloading area and you will have access to the packages at that area.By this we can say that GPUs provide best memory bandwidth while having no drawback due to latency when thread parallelism is used.This is one of the main reasons why GPUs are faster than CPUs for deep learning.

one more advantage of GPUs is that they consist of a small pack of registers for every processing unit ,therefore a lot of register memory which is small and fast, this provides GPUs with registers size more than 30 times bigger compared to CPUs but yet very fast .This difference in size is much more important than difference in speed and it doesn make a difference.And a good compilier tools that can exactly indicate when we are uusing too much or too few registers ,maximal performace is sure guaranteed.

This eventually leads to the conclusion of that we can store a lot of data on register files on GPUs ,in order to be able to reuse convolutional and matrix multiplication tiles. you have a 100MB matrix, you can split it up in smaller matrices that fit into your cache and registers, and then do matrix multiplication with three matrix tiles at speeds.That is again why GPUs are much faster than CPUs in deep learning whick requires alot of matrix multiplications.

To sum up, High band width ,hiding memory access latency under thread parallism and having large and fast register files that can be easily programmable , makes GPUs more fit when it comes to deep learning.

Chapter 4

Conclusion

In conclusion,CPUs and GPUs are differently designed for different purposes,CPUs excels at performaing sequential tasks that requires very high speeds, while GPUs manages to performs different parallel tasks at the same time with lower speeds.However,deep learning tasks are very expensive and requires alot of computions , like said image processing which requires alot of convolution and matrix multiplications,high band width ,hiding memory access latency under thread parallism and having large and fast register files that can be easily programmable , gives GPUs the upper hand when it comes to deep learning.CPUs do not quite live up to the needy requirments of deep learning like the GPUs however CPUs provide some help to GPUs,CPUs helps to feed GPU with enough data and read/write files from/to RAM/HDD during training.

Chapter 5

Future Work

Scaling ML workloads across multiple on-node GPUs is becoming increasingly important. There is a lot of research and active work happening to think of ways to accelerate computing. the performance of deep learning algorithms is determined by the structure of the model, using of the perfect hardware and the large available datasets for the training. So, Google is working to come out with Tensorflow Processing Units (TPUs), which promises an acceleration over and above current GPUs. Similarly Intel is working on creating faster FPGAs, which may provide higher flexibility in coming days. In addition, the offerings from Cloud service providers (e.g. AWS) is also increasing. We will see each of them emerge in coming months.

Appendix

Appendix A

Lists

List of Figures

Bibliography

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning.* MIT Press.

[2] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). *Deep Learning* Nature

[3] eng, L.; Yu, D. (2014). *Deep Learning: Methods and Applications* Foundations and Trends in Signal Processing. 7(34)

[4] Robert Adolf, Saketh Rama, Brandon Reagen, Gu-Yeon Wei, and David Brooks (2016) *Fathom: Reference Workloads for Modern Deep Learning Methods*

[5] John Lawrence, Jonas Malmsten, Andrey Rybka, Daniel A. Sabol, and Ken Triplin (2017) *Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud*

[6] Tung D. Le, Taro Sekiyama, Yasushi Negishi (2018) *Involving CPUs into Multi-GPU Deep Learning*

[7] E. H. Norman *Japan's emergence as a modern state* 1940: International Secretariat, Institute of Pacific Relations.

[8] Bob Tadashi Wakabayashi *Anti-Foreignism and Western Learning in Early-Modern Japan* 1986: Harvard University Press.

[9] Nathan R. Tallent, Nitin A. Gawande, Charles Siegel *Evaluating On-Node GPU Interconnects for Deep Learning Workloads*

[10] Nvidia *Whats the Difference Between Deep Learning Training and Inference* 2016.

[11] Mauricio Guignard, Marcelo Schild (2018) *Performance Characterization of State-Of-The-Art Deep Learning Workloads on an IBM Minsky Platform*

[12] Nvidia *Accelerating AI with GPUs: A New Computing Model* 2016

[13] Dataversity *A Brief History of Deep Learning*