Why Deep Learning Uses GPUs?

And why you should too...

German Sharabok Jul 26, 2020 · 5 min read ★

Photo by Joseph Greve on Unsplash

There is a lot of information out there about GPUs for Deep Learning. You have probably heard that this field requires some huge computers and incredible power. Maybe you have seen people train their models for days or even weeks just figure out there was a bug in their code. Additionally, we hear about GPUs mostly when talking about gaming and sometimes graphical design. This article will help you understand what is actually going on here and why Nvidia is a huge innovator in Deep Learning.

Graphics Processing Unit (GPU)

A GPU is a processor that is great at handling *specialized* computations. We can contrast this to the Central Processing Unit(CPU), which is great at handling *general* computations. CPUs power most of the computations performed on the devices we use daily.

GPU can be faster at completing tasks than CPU. However, it is not true for every case. The performance hugely depends on the type of computation being performed. GPUs are great at tasks that can be run in parallel.

Parallel Computing

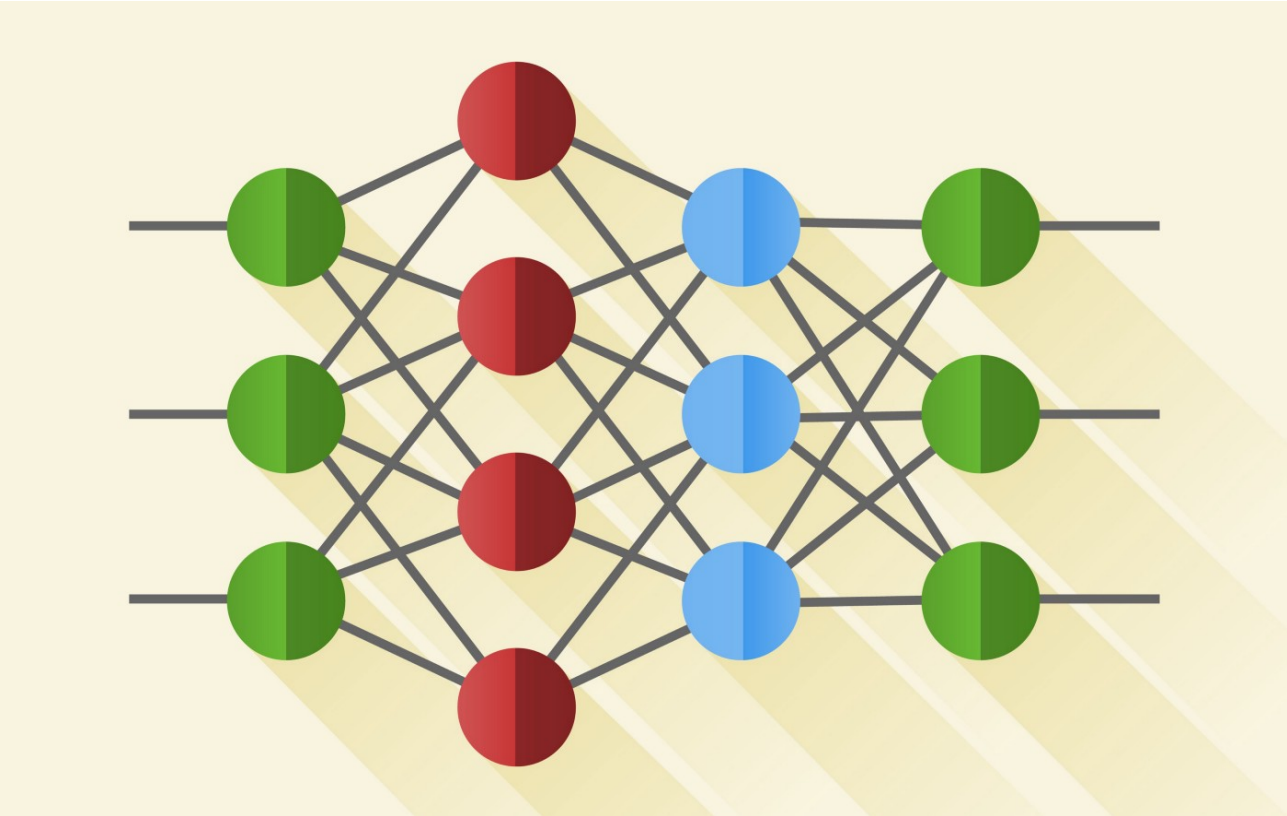
> ***Parallel computing*** *is a type of computing architecture in which*

> *several processors simultaneously execute multiple, smaller calculations broken down from an overall larger, complex problem.*

If we have multiple cores in our processing unit we can split our tasks into multiple smaller tasks and run them at the same time. This will make use of the processing power we have available and complete our tasks much faster.

CPUs generally have four, eight, or sixteen, while GPUs could have **thousands**! From here we can conclude that GPU is best suitable for tasks that can be completed simultaneously. Since parallel computing deals with such tasks, we can see why a GPU would be used in that case.

Neural Networks Are Parallel

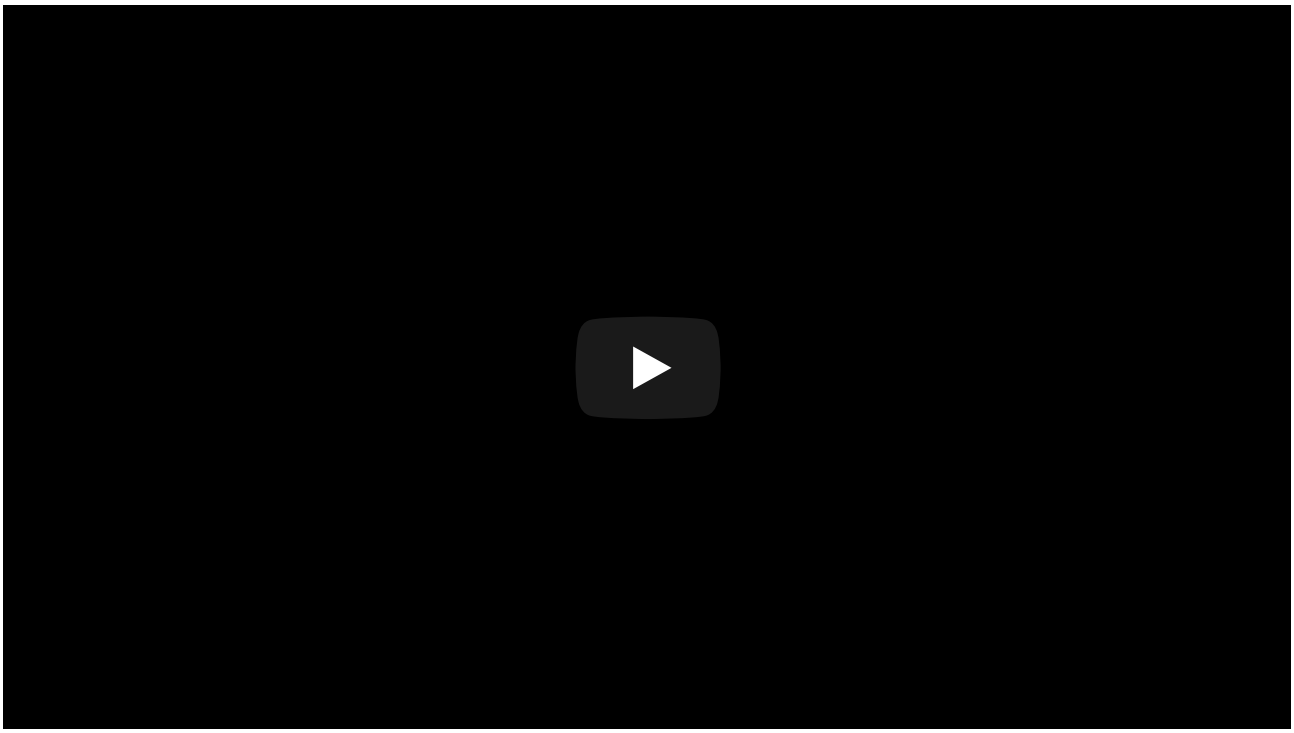


(Image by Brighterion)

We have concluded that GPUs are best used when a huge task can be broken down into many smaller ones, which is the reason GPUs are used in parallel computing. If we take a look into neural networks, we can notice that they are *embarrassingly parallel*. It means that we do not even need to slit the tasks and decide which part goes to which core. The neural networks are specifically made for running in parallel. Since they are the base for deep learning, we can conclude that GPUs are perfect for this task.

Additionally, neural networks are parallel in such a way that they do not have to depend on each other's results. Everything could run simultaneously without having to wait for other cores. An example of such computation that is hugely independent is convolution.

Convolution



(Animation by Israel Vicars)

Here is an example of how convolution without numbers could look like. We have an input channel on the left and the output on the right. In the animation, the process of computation is happening sequentially, which is not the case in real life. Actually, all of the operations could be happening at the same time and neither one of them depends on the results of any other computation.

As a result of this, the computations can happen in parallel on a GPU and the result can be produced. From the example, we can see that parallel computing and GPUs can seriously accelerate the convolution operation. In comparison, running the same convolution on a CPU will lead to sequential execution, similar to the one in the animation. This process will take a lot more time.

Nvidia Hardware and Software

(Image from Nvidia.com)

This is where we can learn about CUDA. Nvidia is a company that produces GPUs and they have created CUDA, which is a software that nicely connects to the hardware they produce. This software allows developers to easily utilize the power of parallel computing with the Nvidia GPUs.

To put it simply, **GPU is the hardware and CUDA is the software**.

As you might have guessed an Nvidia GPU is required to use CUDA, and CUDA can be downloaded from Nvidia's website entirely free.

GPU vs CPU

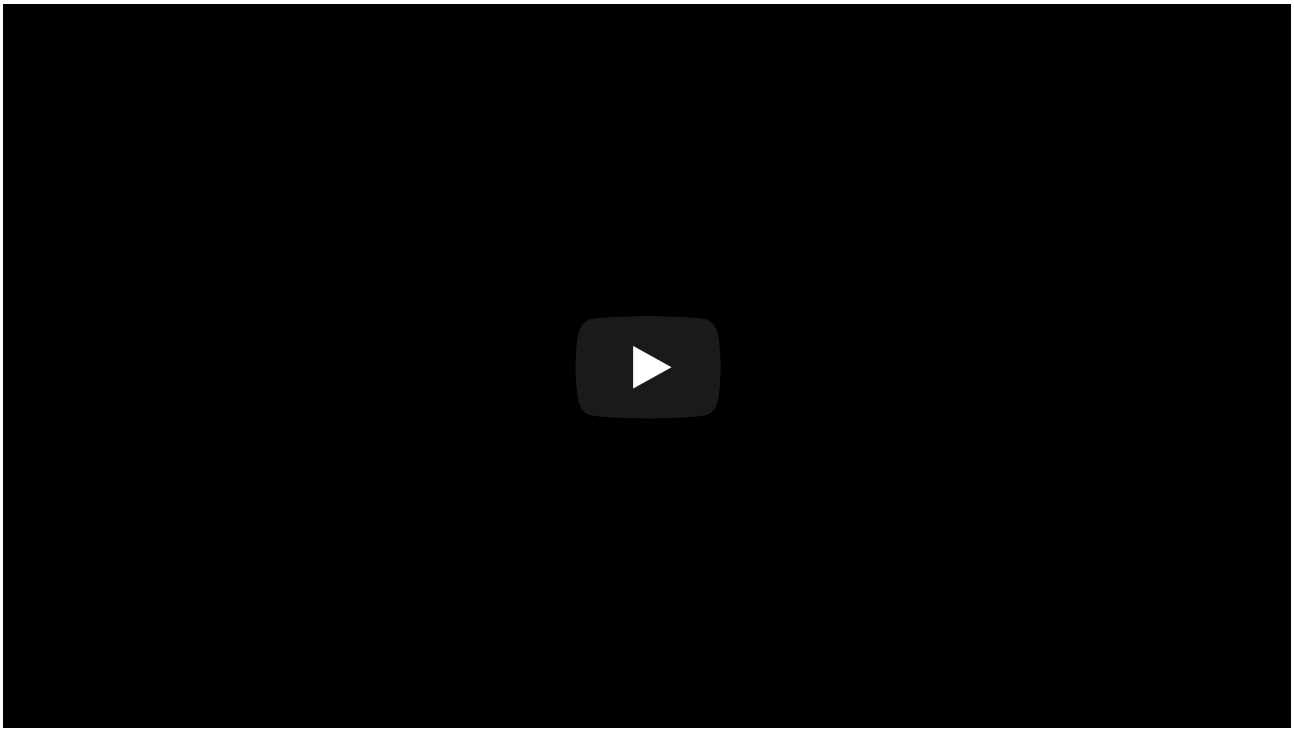
(Image from Nvidia.com)

CUDA allows us to selectively run computations on either the GPU or the CPU. Why not use GPU for everything if it is so much better?

The answer is that GPUs are better only for specific types of tasks. For example, if our data is on the CPU, moving it to the GPU can be costly. So in the case when the task is rather simple, it will be more costly to use the GPU. We can conclude that the GPU will perform better only when tasks are sufficiently large and can be broken down into smaller tasks. For small processes, it will only make things slower.

For that reason, it is often acceptable to use the CPU when just getting started, since the initial tasks will be short and simple.

GPGPU



(GTC China by Nvidia)

When the GPU was first created, it was mostly aimed at dealing with computer graphics, which is where the name comes from. Now, more and more tasks are moved to GPUs and Nvidia is a pioneer in that space. CUDA was created nearly 10 years ago and only now developers are starting to make use of it.

Deep learning and other types of parallel computing have led to the development of a new field, called *GPGPU* or *general-purpose GPU computing*.

The GTC talk by Nvidia is a must-see for everyone in the field of deep learning. When we hear about the GPU computing stack, we should think of GPU as the hardware on the bottom, CUDA software architecture in the middle, and libraries like cuDNN on top.

Conclusion

GPUs play a huge role in the current development of deep learning and parallel computing. With all of that development, Nvidia as a company is certainly a pioneer and leader in the field. It provides both the hardware and software for creators.

It is certainly alright to get started creating neural networks with just a CPU. However, modern GPUs can speed up the task and make the learning process much more enjoyable.

Resources:

[1] Israel Vicars. (2018, May 23). *Convolutional Neural Network Visualization by Otavio Good* [Video]. YouTube. https://youtu.be/f0t-OCG79-U

[2] NVIDIA. (2016, Sep 26). *GTC China: AI, Deep Learning with Jen-Hsun Huang & Baidu's Andrew Ng* [Video]. YouTube. https://youtu.be/zeSIXD6y3WQ

[3] Parallel Computing Definition. (n.d.). Retrieved from https://www.omnisci.com/technical-glossary/parallel-computing

Gpu | Deep Learning | Machine Learning | Nvidia