Module 3 - Deep Learning in the Cloud

- 3.1 Hardware Accelerators
- 3.2 How does one use a GPU?
- 3.3 Deep Learning in the Cloud
- 3.4 Lab - How to run deep learning model in the Cloud
- 3.5 Graded Review Questions (3 Questions)

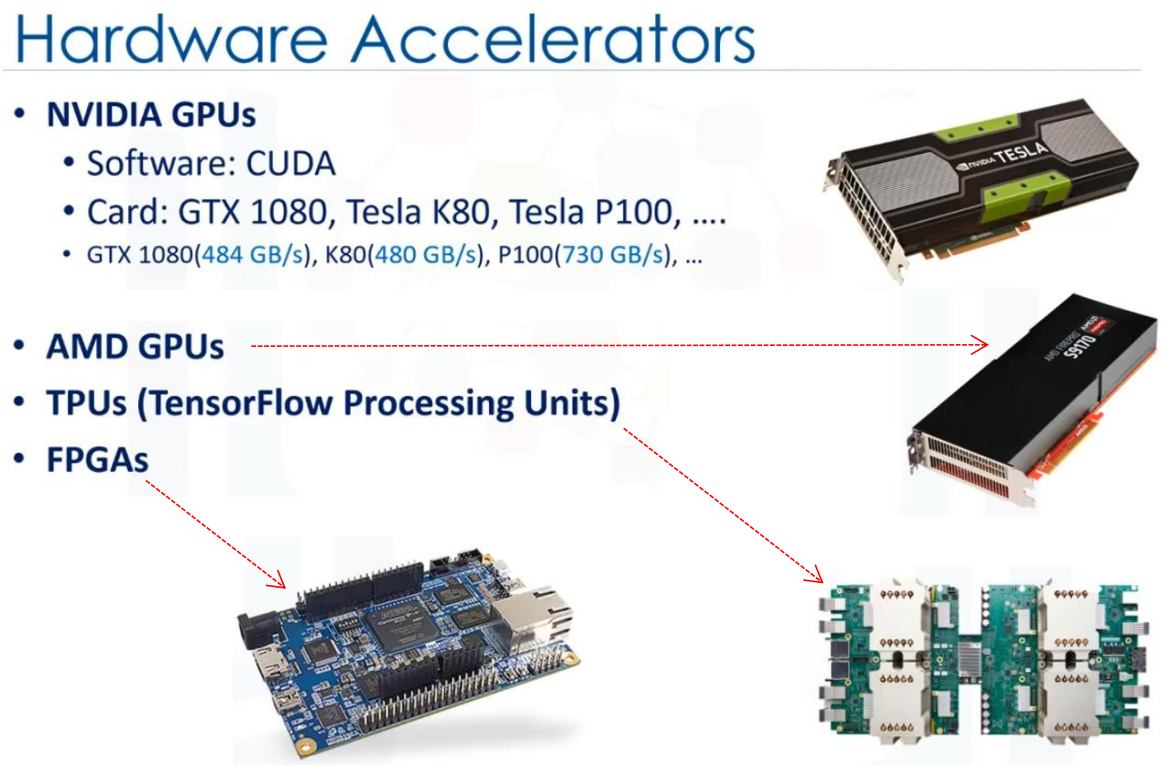
Learning Objectives

In this lesson you will learn about:

- Different hardware accelerators
- Running deep learning in the cloud

总结:

3.1 Hardware Accelerators



In this video, we will examine and compare different hardware accelerators.

Let's quickly take a look at some well-recognized accelerating hardware (and their associated software) that have succeeded in reducing the training time, several times over.

NVIDIA is one of the main vendors of GPU and with **CUDA** software on top of that, we see it on most platforms. Let me explain CUDA. You don't have to understand low level graphics processing to implement your algorithm code on a GPU. Nvidia has a high-level language, known as CUDA, that helps you write programs from graphic processors. Looking at different NVIDIA GPUs, you will find various features, architectures and cards, such as GTX cards and Tesla, in which their performance and speed is highly dependent on their memory bandwidth. Again, just recall that the most important feature of GPUs, which made them a good match for your deep learning, is memory bandwidth to fetch high dimensional data. So, we can say it's an important metric for a GPU.

AMD cards are also an option, but AMD's **OpenCL**, the software on top of AMD cards, is not very popular among developers, who are working on deep learning libraries right now.

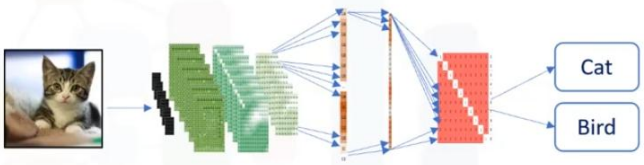
TensorFlow Processing Units (or **TPUs)** is the Google hardware accelerator solution developed specifically for TensorFlow, which is Google's open-source machine learning framework. TPU promises an acceleration over-and-above the current GPUs.

FPGAs (or, **Field Programmable Gate Arrays**) are programmable or customizable hardware. That is, in contrast to a processor that has a fixed number of resources, you build a custom circuit with resources and hence exploit all the parallelism in a program. Intel is working on creating faster FPGAs, which may provide high flexibility.

总结:

Hardware Accelerators

- **TPUs, GPUs, FPGAs**
  - Training
  - Inference



- **Limitations of GPUs:**
  - Limited Memory capacity
    - Not practical for very large datasets
    - Alternative: reading data from system memory (overhead)
  - Accessibility
    - Expensive, dependencies and incompatibilities

Now the question is that among all of these existing options, what kind of accelerator should you get?
You can use Google's TPUs, Nvidia GPU or even FPGA to accelerate your deep learning network computation time. These chips are particularly designed to support the training of neural networks, as well as the use of trained networks (that is, inference).

But there are some general limitations in these accelerators as well. Let me start by focusing on GPUs, as they're the most-popular accelerator.

GPUs have two key limitations:

- The first involves limited memory capacity. Yes, GPUs are very fast for data parallelism and, as such, we can take full advantage of their massive computing power. That said, we still need to store the data inside the GPU memory in order to access it and process it. Unfortunately, GPUs currently have up to 16GB of memory, so this is not practical for very large datasets. In this case you have to read data from system memory, and it's a huge overhead. So, you need a platform that can handle fast memory access in system memory, and also fast data exchange between GPUs.
- The second limitation of GPUs is that you cannot easily buy these accelerators and embed them into your local machine. They're usually expensive and there are some dependencies and incompatibilities, which is the same as most hardware. Also, sometimes, you need a number of GPUs to handle your big datasets. So, these accelerators are not readily accessible, at least not for now.

At this point, the question is, where can I get a GPU to train my deep learning network? Thanks for watching this video.

总结:

3.2 How does one use a GPU?

Required Hardware

1. **A laptop with an embedded GPU**
   - Not enough to solve real deep learning problems
   - Needs to scale down the dataset

2. **Using a GPU on a cloud service**
   - IBM cloud, AWS or Google cloud
   - You can customize it
   - Needs to move your data in the cloud
   - Options for Single-GPU and Multi-GPU

3. **Using a GPU cluster in the cloud**
   - IBM cloud

4. **Using a GPU cluster on-premises**
   - Keep your data locally
   - Cost- effective
   - Perfect for sensitive data
   - IBM PowerAI

Now, the question is: ”Where shall I get an accelerator, such as a GPU, for running my deep learning pipeline?” In this video, we'll look at different options for getting an accelerator.
First, you always have to keep in mind that deep learning requires a lot of computational power to run on, but it’s totally dependent on the task at hand. You don’t really need to buy or find a large Datacenter with many GPUs to run your model. There are a number of options that are available to you and you really should look around.

① **The first one to consider is that some personal computers have an embedded GPU**. For example, a laptop with a recent NVIDIA GPU should support CUDA, so you can at least play around with deep learning networks, and you can use it to train your deep learning, to some extent, but usually not enough to solve particularly deep learning problems. In this case, you need to scale down the dataset, or the model, to something that fits on a laptop, which often delivers significantly worse results.

② **Many cloud providers are offering GPU services as virtual machines with GPUs that you can use, like IBM cloud, Amazon AWS, or Google cloud**. The difference between these providers is usually the hardware that they offer, and of course the price that you pay for it. In general, these are good options to start building and training your model, as you can configure an instance with the ratio of processors, memory and GPUs you might need. However, you should consider that you need to upload all your data on the cloud and train the network on the cloud, which sometimes is a hassle. Usually, you can find services that offer you single or multi-GPU access. My suggestion is to use a fast-enough single GPU to do experiments with sample data to verify many things before going full scale. After the experiments, you’ll need an 8-or 16-GPU instance to finish the job.

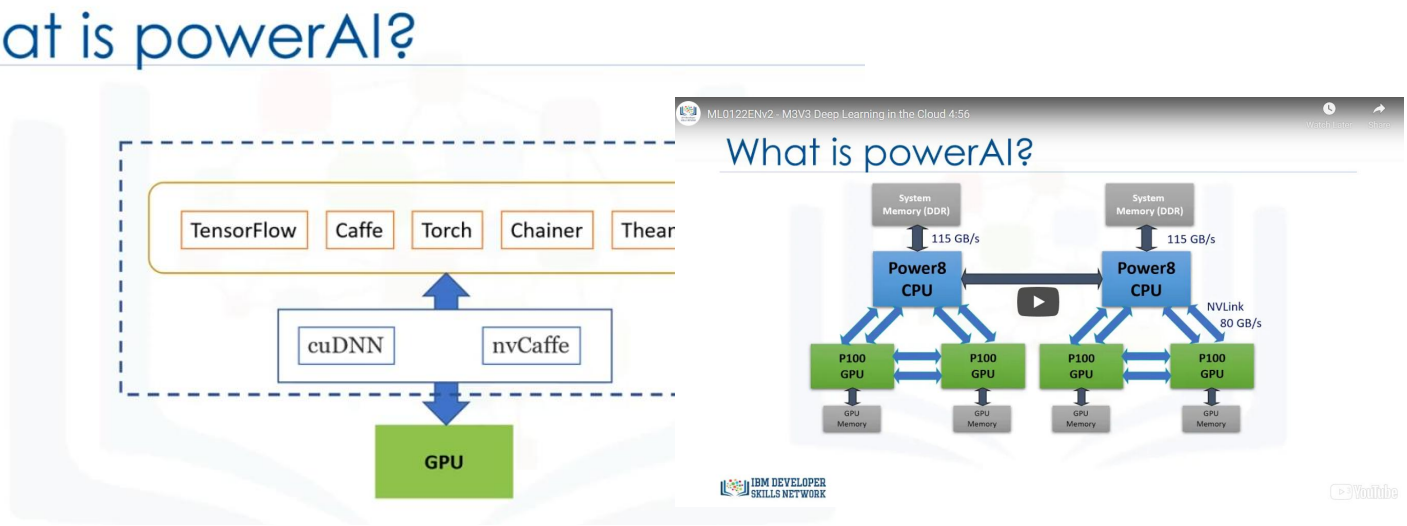
③ If your data is very big in terms of volume and computational requirements, you’ll need a very large computational system to handle your data. In this case, you’ll likely need a cluster of GPUs to distribute the whole computational workload. Using a GPU cluster and running computations in parallel by **a cluster of GPUs on one of the cloud services such as IBM**, could be a good solution for such a scenario.

④ Although using a cloud service is a good choice for getting started and solving some problems, you should keep in mind that building ‘cloud-based deep learning’ could be very costly -- especially if you need to train models for more than 1000 hours. In this case, **using a GPU cluster or doing multi-GPU computing On-premise is the best option,**, as it’ll allow you to keep your data locally and do computations on your servers, which becomes cost-effective. Additionally, your data might be sensitive, with need to analyze it yourself On-premises. If so, you may not feel comfortable to upload it into public clouds. In this case, you’ll need to use an in-house system with GPU support, such as IBM PowerAI. So, the required hardware includes: 1. A laptop with an embedded GPU, 2. Using a GPU on a cloud service, 3. Using a GPU cluster on the cloud, and 4. Using a GPU cluster on-premises. Thanks for watching this video.

总结：

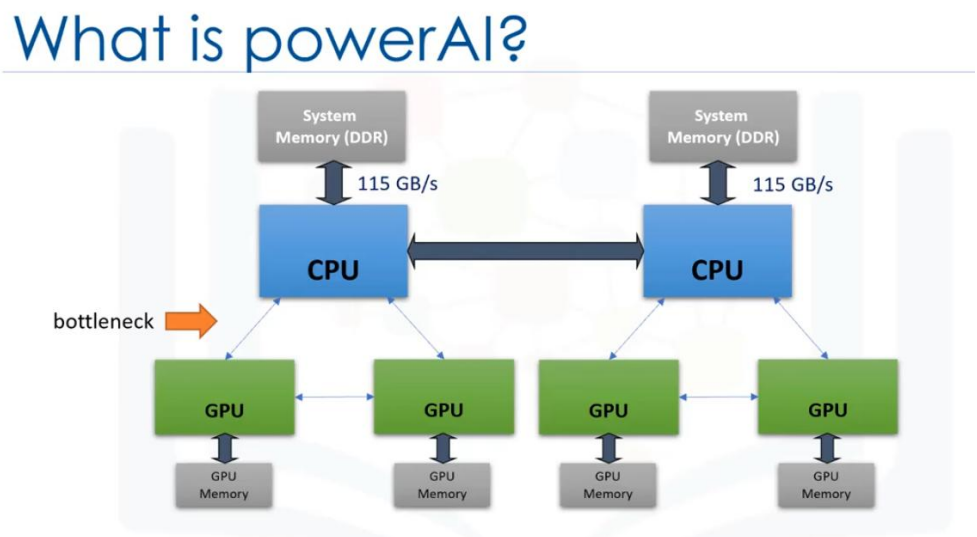
3.3 Deep Leaning in the Cloud

详见下载下来的网页：
D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)**M3_3 Deep Learning in the Cloud.html**



In this video, we look at the IBM offering for accelerated deep learning, and see how it's solved the limitations of using GPUs. IBM offers a platform called PowerAI for deep learning. Let me explain what PowerAI is.

As you know, there are various software frameworks for building and training a Deep Learning model, such as TensorFlow, Caffe, Torch, Chainer, and Theano. These frameworks can take advantage of graphical processing units (or GPUs) to accelerate the training or inference process, but need different types of extensions to work on GPUs, for example, CUDA Deep Neural Network, and nvCaffe libraries. **IBM PowerAI is a package of software distributions for these types of software**.



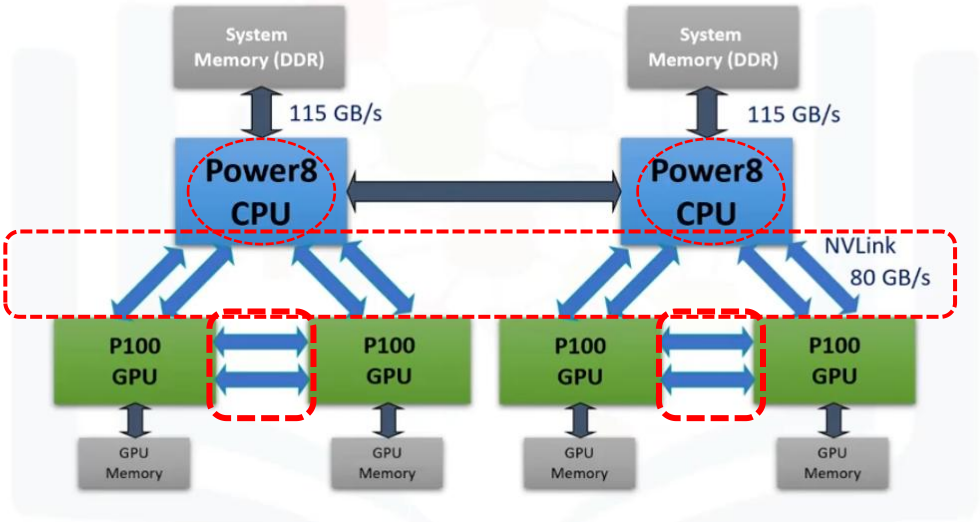
On this platform (**IBM PowerAI**), the problem of memory access is addressed.

To start the story, consider that Data, or rather Big Data, is located in system memory of your computer. When an application starts to analyze this data using a GPU, you have to move chunks of it over to the GPU via the CPU. As you can imagine, moving data from the CPU to the GPU creates a **bottleneck** because most of that data is going through a thin pipe, called **PCIe**. So, we can say that **bandwidth** is the problem here, which highlights a simple truth. It's the data flow that will determine the final performance for your workload.

PowerAI takes advantage of **NVLink** to increase system bandwidth.

总结：

What is powerAI?



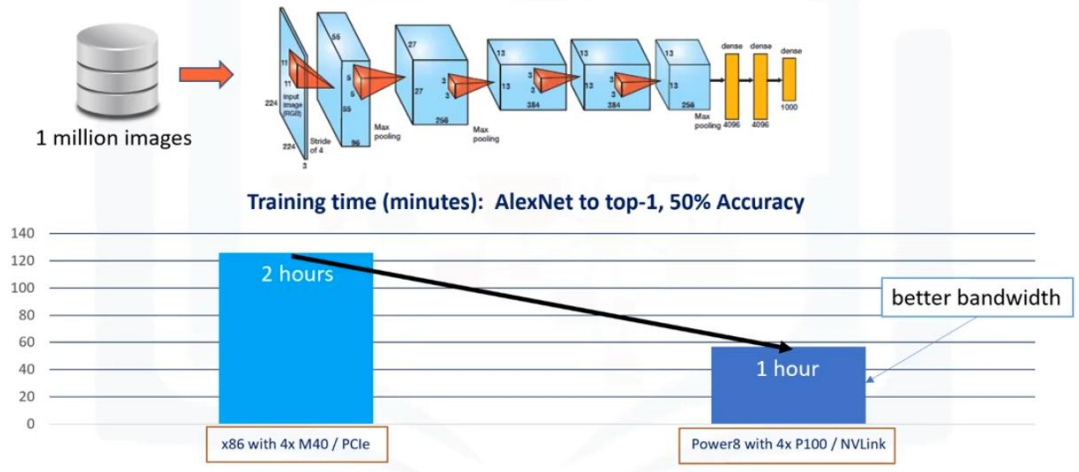
What is NVlink? Let me explain. **NVLink** is **a high bandwidth protocol** that enables **faster GPU-to-GPU communication**, by providing multiple **point-to-point connections**. For Deep Learning workloads, this decreases memory-cache copy time, reducing GPU wait, and allowing the GPUs to execute more training cycles in a shorter amount of time.

You should also note that on the PowerAI platform, full NVLink connectivity between CPU and GPU, provides **the same wide data path between the GPU and system memory**. This allows workloads using large training datasets a faster way to "reload". This is particularly important because large datasets can't be stored only in a GPU's limited memory. The outcome is faster training and the ability to train with larger datasets for improved accuracy.

So, in brief:
① two NVLink connections between GPUs reduces GPU wait,
② and high-speed connection between GPU and CPU enables the machine to perform fast memory access to large datasets in system memory.

IBM uses Tesla P100 and Power8 for this specific customization. Built on IBM's Power Systems, PowerAI is a scalable software platform that accelerates deep learning and AI (or artificial intelligence) with blazing performance for individual users or enterprises.

Deep Learning & GPU accelerated



I am sure you already appreciate that deep learning software frameworks scale well with GPU accelerators.

Nevertheless, it was important to show you just how much faster the GPU acceleration could be using PowerAI. Based on a benchmark, this new architecture leads to much faster training time.

Let's compare the training time of running a deep learning sample on hardware accelerators by PCIes and Nvlink to understand the difference. Please notice that it's not comparing training time with and without GPUs. It is about how we can take full advantage of GPU acceleration for deep learning, using better bandwidth. For this benchmark we use Alexnet. **Alexnet** is a large, deep convolutional neural network to classify more than 1 million high-resolution images of a famous training set into the 1000 different classes. Today, this network is a common benchmark for deep-learning training. Training this network using 4 GPUs, allows you the capacity to reach 50 percent accuracy in almost 2 hours. But, running the same network on PowerAI shows an approximately 2 times faster learning, which is a promising result.
If you need more information on how PowerAI works, you can go to the link shown below the video. Thanks for watching!

总结:

3.4 Lab - Deep Learning in the cloud

详见下载下来的网页：
D:\KeepStudy\01_Edx\1 Deep Learning with Python and PyTorch (IBM-DL0110EN)\Labs**M3_4 Lab - How to run deep learning model in the Cloud.html**
配套代码：
D:\KeepStudy\01_Edx\1 Deep Learning with Python and PyTorch (IBM-DL0110EN)\Labs\

3.5 Graded Review Questions (3 Questions)

Which are the most popular hardware accelerators in use today?

- ○ FPGAs (programmable or customizable hardware)
- ○ AMD cards with OpenCL software
- ○ Tensorflow Processing Units (TPUs)
- ○ NVIDIA GPUs with CUDA software
- ◉ All of the above

✔

"In some situations, your data might be very huge in terms of volume and computation in such a way that you need a really large computational system to handle it. In this case, you need a cluster of GPUs to distribute the whole computational workload." Is this statement TRUE or FALSE?

- ◉ TRUE
- ○ FALSE

✔

Which of the following statement is TRUE about deep learnong in the cloud?

- ○ Building cloud-based deep learning could be really costly if you need to train models for more than 1000 hours.
- ○ You need to analyze your data on-premise, when your data is sensitive and you may not feel comfortable to upload it into public clouds.
- ○ If your data is big, use a fast enough single GPU to do experiments with sample data to verify many things before going full scale.
- ◉ All of the above.

✔

总结：