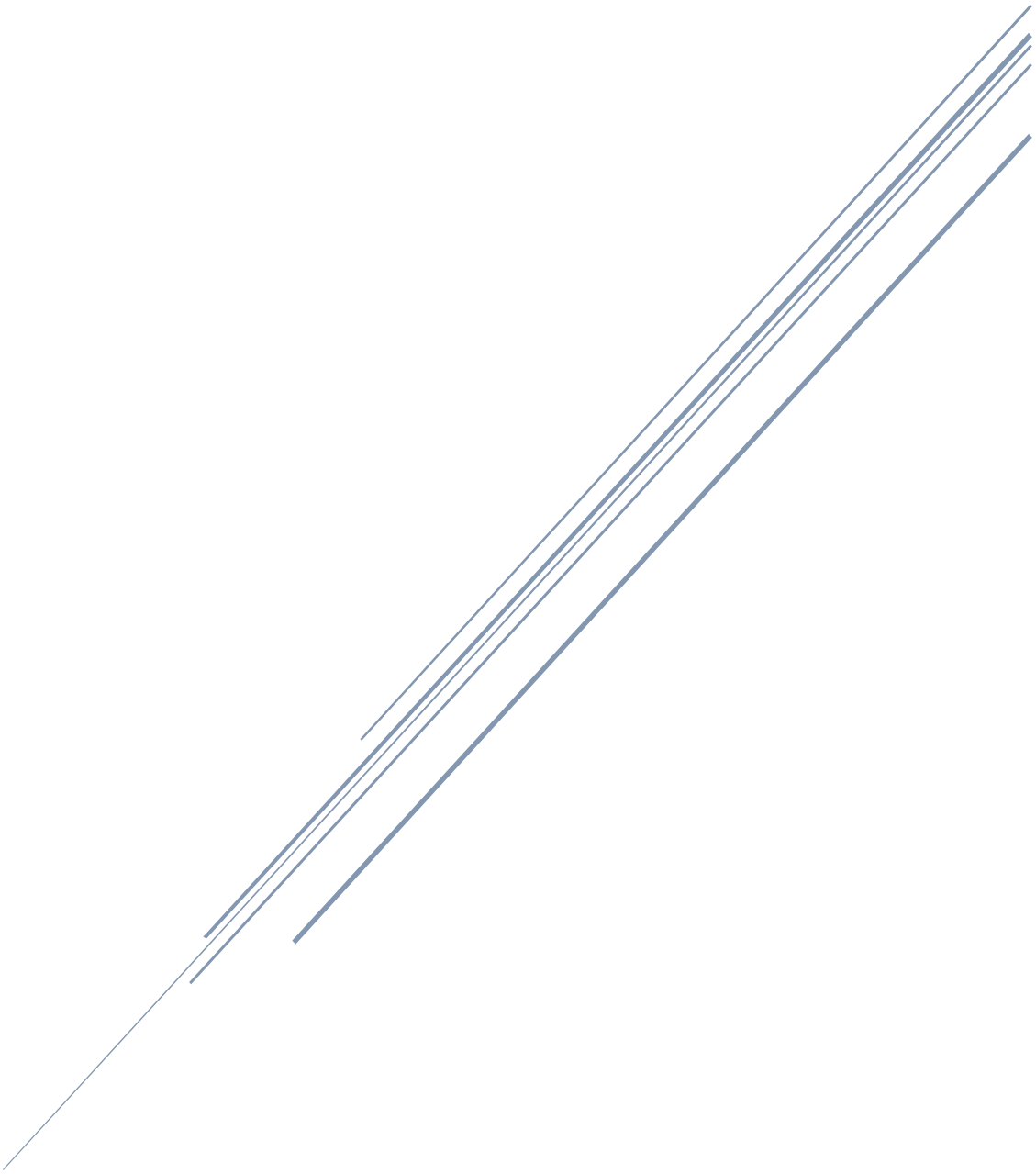
DEEP LEARNING IN THE CLOUD

Module 3



v-cardona

Using GPUs to Scale and Speed-up Deep Learning

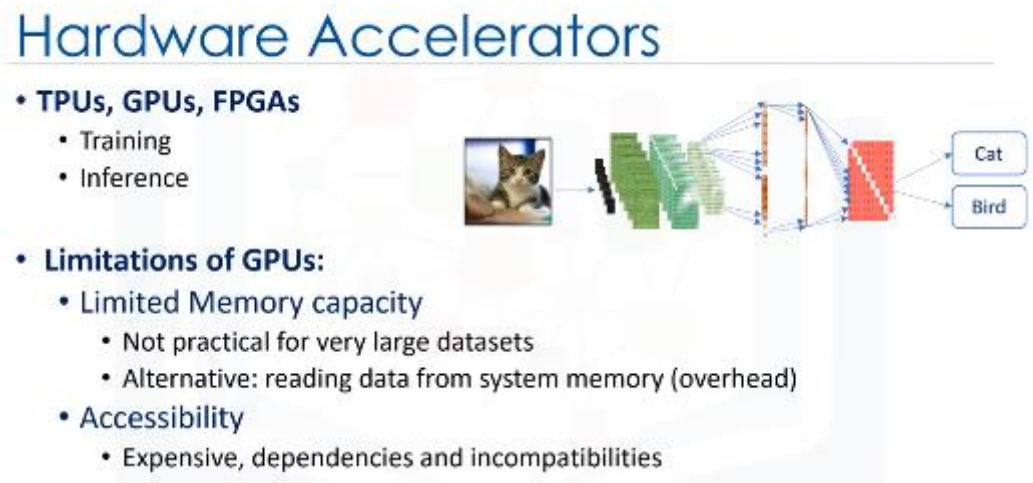
Contenido

Hardware accelerators

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).

GPUs have two key limitations:

1. 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. You need a platform that can handle fast memory access in system memory, and also fast data exchange between GPUs.
2. The second limitation of GPUs is that you cannot easily buy these accelerators and embedthem 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.



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

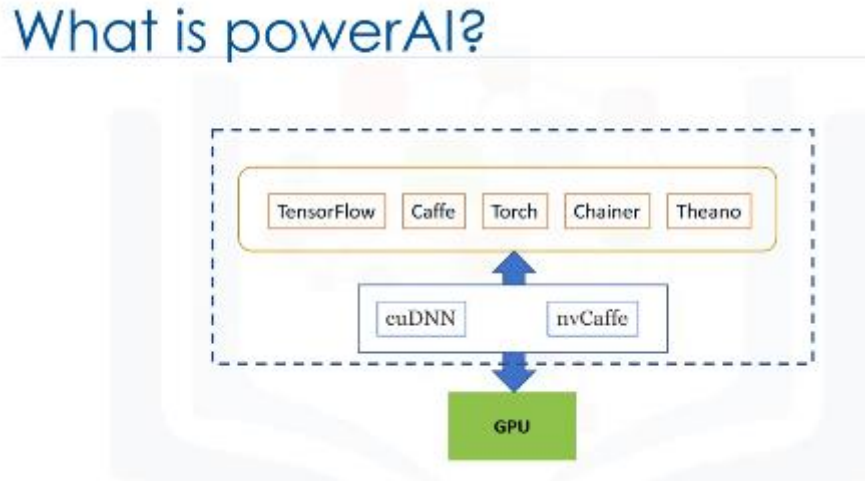
2. **Using a GPU on a cloud service**
   - IBM cloud, AWS or Google cloud
   - You can customize it
   - Needs to move your data in the cloud
   - Options for Single-GPU and Multi-GPU

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

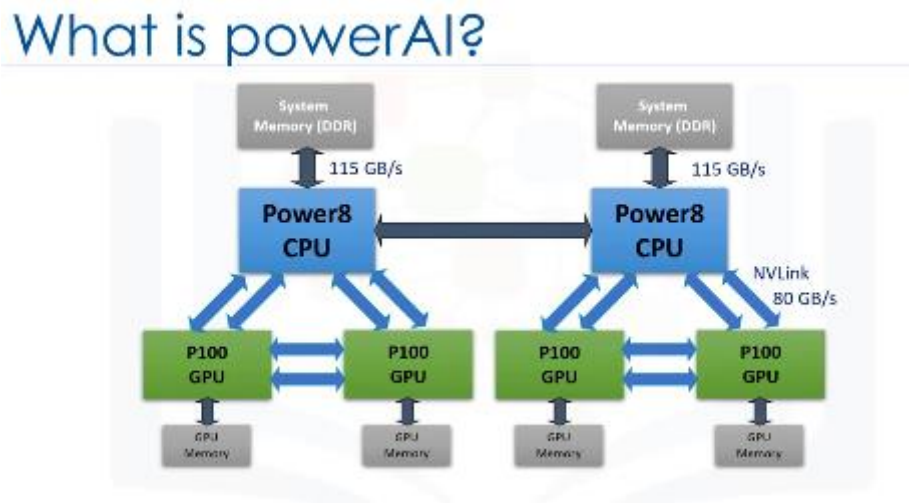
4. **Using a GPU cluster on-premises**
   - Keep your data locally
   - Cost- effective
   - Perfect for sensitive data
   - IBM PowerAI

Deep learning in the cloud - PowerAI

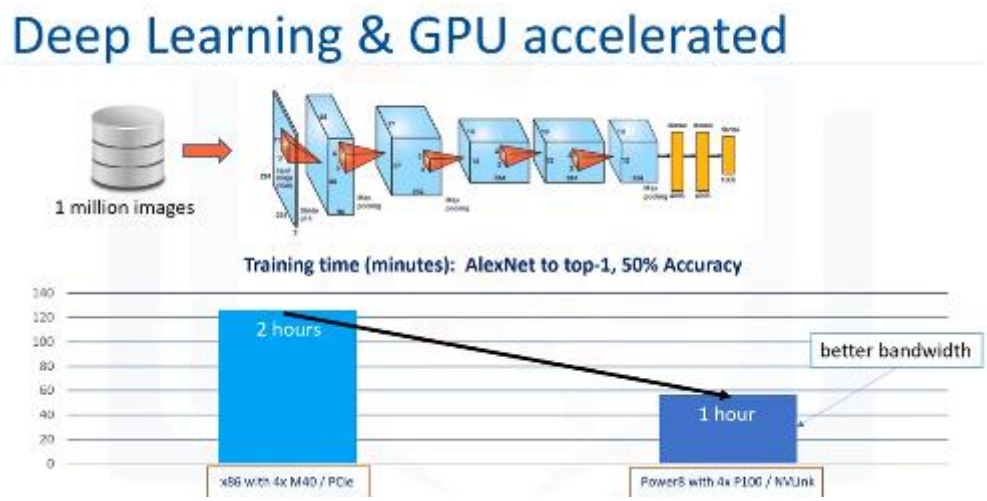
There are various software frameworks for building and training a Deep Learning model, such as TensorFlow, Caffe, Torch, Chainer, and Theano. These frameworks can take advantage of graphical processing units (or GPUs) to accelerate the training or inference process, but need different types of extensions 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.



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. Therefore, we can say that bandwidth is the problem here, which highlights a truth. The data flow will determine the final performance for your workload. PowerAI takes advantage of NVLink to increase system bandwidth.



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 cannot 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.



Last picture is a graphic of how much time does cost to train, both with GPUs but with PowerAI, which take into account and profit of bandwidth. It can reduces the time one hour.