DELL EMC

PowerEdge Product Group

Direct from Development

Building the Optimal Machine Learning Platform

Tech Note by:
Austin Shelnutt
Paul Steeves

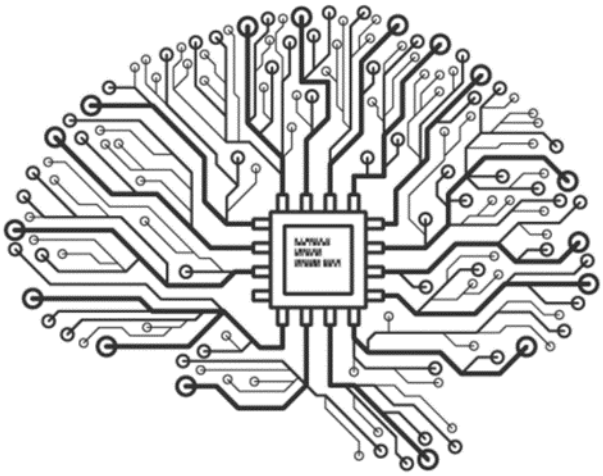
SUMMARY

Machine Learning customers have more choices than ever for neural network models and frameworks.

Those choices impact the type, number, and form factor of the preferred accelerator, the dataflow topology between accelerators and CPUs, the amount and speed of direct attached storage, and the necessary bandwidth of I/O devices.

This tech note provides a brief overview of some of the basic principles of Machine Learning and describes the challenges and trade-offs involved in constructing the optimal Machine Learning platform for different use cases.

While various forms of machine learning have existed for several decades, the past few years of development have yielded some extraordinary progress in democratizing the capabilities and use cases for artificial intelligence in a wide multitude of industries. Image classification, voice recognition, fraud detection, medical diagnostics, and process automation are just a handful of the burgeoning use cases for machine learning that are reinventing the very world we live in. This blog provides a brief overview of some of the basic principles of Machine Learning and describes the challenges and trade-offs involved in constructing the optimal Machine Learning platform for different use cases.



Neural Networks are key to machine learning

At the center of the growth in machine learning is a modeling technique referred to as neural networks (also known as deep neural networks, or deep learning), which is based on our understanding of how the human brain learns and processes information. Neural networks are not a new concept, and have been proposed as a model for computational learning since the 1940's. What makes neural networks so attractive for machine learning is that they provide a mathematical ecosystem that allows the decision making accuracy of a computer to scale beyond explicit programming rules and, in a sense, learn from experience.

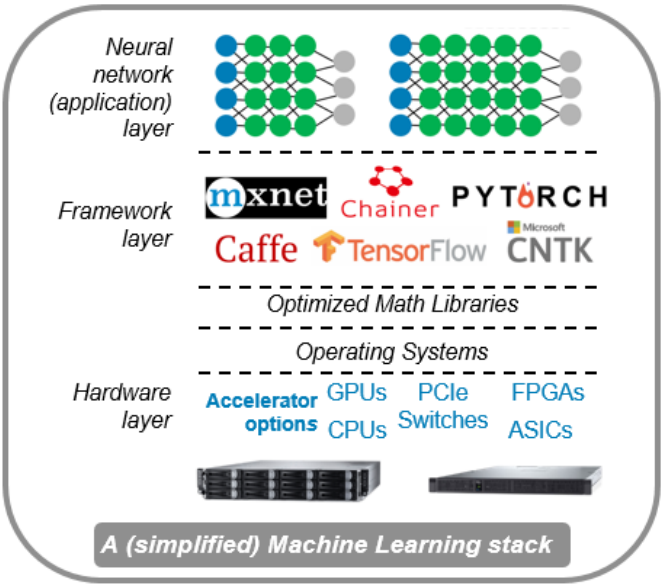
Previously, the limiting factor of neural network models has been that they are extremely computation intensive and require a tremendous amount of labeled data input to be able to "learn". This double hurdle of processing power and available data had prevented them from becoming relevant…. until now.

DELL EMC

The Machine Learning platform stack

As shown in in the figure to the right, the machine learning stack consists of:

- The **neural network (**application) layer – this is the data analysis model
- The **framework** layer - provides the specialized software neural networks run on
- The **math libraries** layer - houses the math routines the frameworks call
- The **operating system** layer – choice of OS
- The **hardware platform** layer -offers a number of different accelerator options



The platform choices made at each of these layers can impact the performance and capabilities of the targeted learning function. The following sections expand on important points that should be considered for these layers.

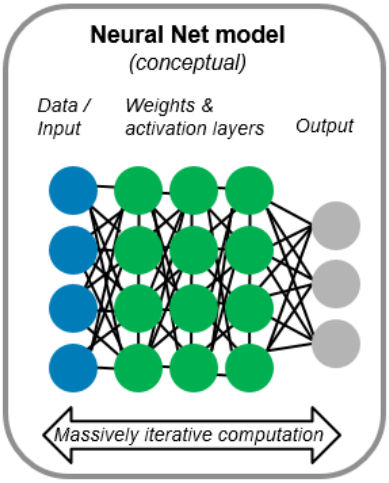
Neural Network layer

Neural networks are symbolic representations of the mathematical models created for a specific learned function (for example, speech recognition). Neural networks come in many different shapes, sizes and functions, depending upon both the type of data being ingested and the intended goal (output) of the learned function. The complexity of neural network construction can vary by:

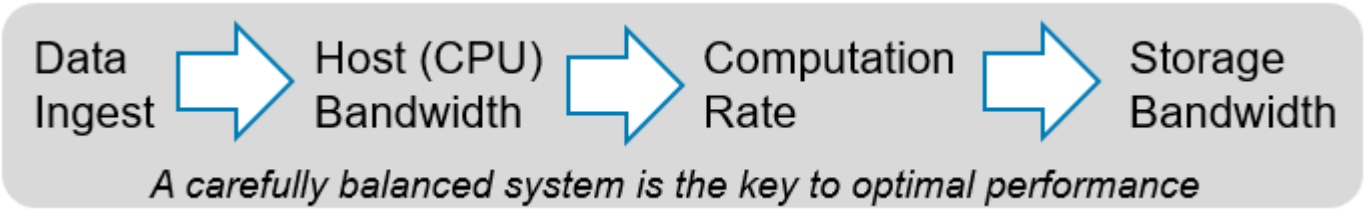
- Specific activation functions
- Number of activation layers
- Data set manipulation types: forward/backward propagation, convolutions, recurrence, LSTM, etc.

At the highest level, all neural networks break down input features (such as the pixels in a photo) into multi-dimensional arrays of data (tensors) and then pass them through one or more layers of parameters (or weights) into activation functions which can be represented as neurons in the neural network.

Input tensors are multiplied by parameter tensors and activation functions to yield a hypothesis that can be used for a decision – for example, classification that an object shown in a given picture is a cat or a dog. The size and dimensions of the input features and the number of activation layers is what determines how to handle the necessary math operations in the hardware layer (i.e., you may require multiple GPUs).



When designing the optimal platform to use for a neural network, how that particular neural network is constructed is crucial in determining what options are best for it at other layers of the stack. ***In general, the platform designer's goal is to understand how data is moved in, out, and around inside of the system to tune features in a manner that most efficiently eliminates data choke points or bottlenecks***.



For example, small neural networks that can be computed relatively quickly might create a tremendous demand on data set ingest bandwidth either from local storage or remote data pools and consequently would be potentially bottlenecked by slow storage devices or narrow I/O bandwidth. Pairing this type of neural model with a high performance accelerator platform that lacks significant I/O bandwidth would also result in under-utilized compute hardware.

As another example, very large neural networks with a large number of input features and/or activation layers, may not fit comfortably inside of a single accelerator's onboard memory or need to swap weight calculations in and out of the page file during each iteration. This type of model might operate most efficiently when the stored weights can be exchanged and multiplied across multiple accelerators. So, a hardware platform that offers multiple accelerators would be the right choice in this case. But note that the distribution of operations to multiple accelerators is handled differently by different hardware offerings and frameworks, so the efficiency of distribution varies accordingly. Also note that not every neural network benefits equally from multiple accelerators – or at least not at the same scaling efficiency. (See the following sections.)

Framework layer

Neural network models run on deep learning software frameworks. The proliferation of frameworks, while primarily open source in nature, has largely stemmed from academia and a number of hyperscale service providers – each attempting to advance their own particular code. You can run virtually any neural network on any deep learning framework, but they are certainly not all created equal. The manner in which frameworks utilize the underpinning hardware varies from framework to framework. While end users often choose a framework based on coding familiarity, there are a number of factors to consider that impact neural network performance:

- How a framework makes math library calls (and which libraries it uses), how it pulls apart the tensor multiplication operations, and how it maps these operations into the physical hardware are all unique to that framework.
- Some frameworks are better at scaling outside of a single server to use multiple servers working together - and some are not capable of scaling out at all.
- Some frameworks are well suited to orchestrating neural network mathematics across a large number of parallel compute devices (i.e. GPUs) within a single server, while others scale very poorly on multiple accelerators.

Each of these points needs to be considered in light of the characteristics of the specific neural network. They may ultimately influence the choice of framework and the accelerator options.

Hardware Platform Layer

Choosing the right hardware technology to support a given machine learning application is another challenge for platform design. While CPUs can be used for deep learning, they are scalar multiplication engines by nature, and poorly suited to the higher-order tensor operations common to deep learning (vectors, matrices, and beyond). So machine learning platforms typically incorporate some form of accelerator technology - GPU, FPGA, or ASIC. But even at that level there are trade-offs to consider – particularly concerning the distribution of operations across multiple accelerators and how that impacts scaling. These considerations are described below.

GPUs

GPUs have been the cornerstone of the deep learning growth in recent years because of their powerful parallel compute capabilities derived from their relatively large number of independent logic cores. The different models for how data is exchanged between GPUs is a differentiating feature when considering platform design.

FPGAs & ASICs

Though GPUs currently occupy a fortress on the deep learning market today, technology vendors from across the globe are lining up to take aim at specific soft spots in the GPU's dominance. The latest FPGA and ASIC technology delivers new levels of component-level performance-per-dollar, performance-per-watt and small-batch efficiency that will result in competitive offerings in 2018 and beyond that will provide alternatives to current deep learning hardware.

PCIe-based Accelerators

Using PCI-Express accelerators for machine learning has become popular for a number of previously discussed reasons, however, one primary benefit is the ability to 'scale-up' to use multiple accelerators in the same server. The challenge for effectively using more than one accelerator is data exchange between the cards. The latency and bandwidth limitations for data going back through the host CPU's PCIE root complex, for example, can be a large performance penalty that negates the multi-accelerator benefit, as shown in Figure 4 below.

Modern non-blocking PCIE switches, as in Figure 5 below, can be a great solution to this challenge by allowing the PCIE accelerators to exchange data directly without passing through the host root complex, if the framework comprehends this type of communication path.

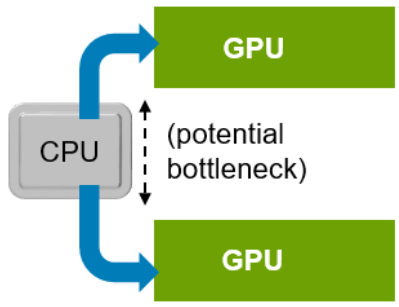


Figure 4: Acceleration using multiple GPUs tied to a single CPU can result in performance bottlenecks.

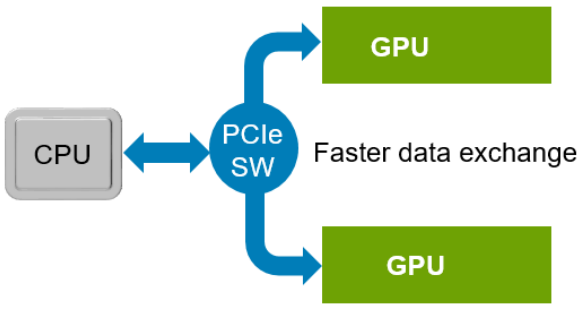


Figure 5: Acceleration using non-blocking PCIe switch

Again, here, balance is the key. As you add accelerators to the switch, eventually the host bandwidth between the switch and the (single host) CPU becomes the new bottleneck. Unfortunately, due to the variations in neural networks, data sets, and frameworks, this point is a moving target, and very difficult to predict.

Specialized accelerator-to-accelerator communication

Many technology companies are now implementing specialized accelerator-to-accelerator connection links, as conceptualized in Figure 6 below. Nvidia's NVlink is an example of a specialized communication path that dramatically improves bandwidth between accelerators for applications that benefit from peer-to-peer data exchange.

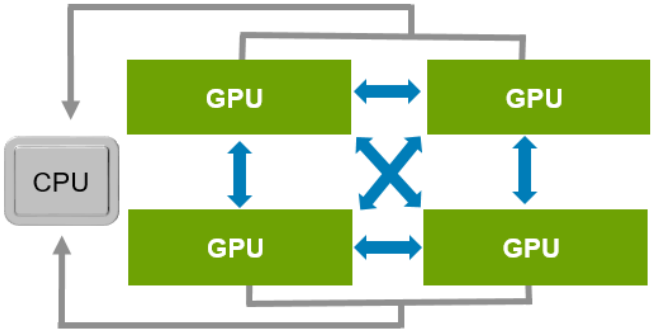


Figure 6: Accelerator-to-accelerator connection links

To be clear, while these auxiliary connection types are extremely valuable for some end customer use cases, there are other deep learning applications that yield very little benefit from this type of interconnect. Furthermore, these specialty interconnects can be costly, both in terms of materials and design changes required to accommodate them.

In fact, the current proprietary interconnect trend is driving unique server designs - just to support the interconnect; resulting in wide variations in hardware from vendor to vendor. Accelerator technology vendors are, seemingly, abandoning all forms of conventional design guidelines in their own pursuit of maximum peer-to-peer bandwidth. This may be the single biggest pain point for designing a truly optimized deep learning platform.

What's next for Machine Learning platforms?

Physical manifestation of the peer-to-peer interconnect is not the only place where deep learning technology providers are departing from conventional techniques. In the pursuit of ever-improved performance, some vendors are moving beyond PCIE form factor, pushing beyond the accepted power/heat limits, and writing new math libraries. Platform designers need to be aware that the technology underpinning the explosive growth in machine learning is still very fluid and divergent.

Conclusion

Machine learning customers have more choices than ever for neural network models and frameworks. Those choices impact the type, number, and form factor of the preferred accelerator, the dataflow topology between accelerators and CPUs, the amount and speed of direct attached storage, and the necessary bandwidth of I/O devices. The resulting platform must:

- Serve 'their' specific learning model– not an unrelated deep learning model.
- Stay within their data center requirements for server form factor, rack depth, power and cooling
- Be management agnostic

Solving a platform optimization challenge with this many degrees of freedom may seem daunting, but Dell EMC is committed to helping our customers meet this challenge. Today, we are already working with a wide range of customers, across a number of industries to solve some of the most complex and interesting machine learning problems. And going forward, we are committing resources to ensure we remain a technology leader in this arena.

For more information on what Dell EMC Extreme Scale Infrastructure is doing with Machine Learning, contact ESI@dell.com .