HIVENet: Neural Network Implementation Tech Review

Alex Garcia, CS 461 Fall 2018

Abstract

Neural Networks remain a complex issue within the field of computer science. They require sophisticated APIs to develop and manage, as well as expensive hardware to properly train. Many within academia and within the IT industry are racing to find better solutions to these demands, and the landscape of machine learning is changing frequently. This tech review is designed to weigh several of the options available for implementing and training an artificial neural network, keeping in mind the objectives an constraints of the project at hand.

1 Neural Network Frameworks

While the term Artificial Neural Network has become ubiquitous within the field of computer science, it is not a monolithic term. There are a variety of NN implementations using different infrastructures and learning paradigms. Each variant has its own optimizations and drawbacks, specializing it to handle certain kinds of data and to solve certain kinds of problems. While we already know that we want to implement a neural network for this project, which architecture will best suit our application still needs to be determined. Furthermore, the architecture of the network may influence how we want to compute our solutions, discussed later in this document.

1.1 Convolutional architecture

Convolutional neural networks (CNNs) are the common NN implementation to handle image recognition applications. These types of networks use several hidden layers, including one or more convolutional layers in between fully connected network layers. The overall design takes inspiration from ocular neurons that enable biological vision, where neurons are connected to receptive fields that capture data received from light. The convolutional layers of the network filter the aggregated input, reducing the dimensionality and passing their data forward to the next layer. This reduction of the dimensionality vastly reduces the number of neurons necessary to process high resolution images, allowing computers to process image data within more feasible time-frames. CNNs are often trained using supervised learning.

1.2 Supervised Learning

Supervised learning makes use of example pairs (x,y) where f(x) should lead to y; the aim of neural networks then becomes to find the function f. Supervised learning is commonly used for problems of pattern recognition where computers must be able to sift through complex datasets and be able to accurately identify the patterns that arise within them. This learning paradigm can enable image classification as well as speech or gesture recognition. What makes this learning supervised is the fact that training set (x) must be labeled by a developer to illustrate what features (y) are within the set. The neural network creates inferences from the dataset, and utilizes learning algorithms in order to get closer to accurately predicting (y).

1.3 Reinforcement Learning

Reinforcement learning is a learning paradigm that emphasizes an action-reward structure, where neural networks are trained to make certain decisions within a designated environment in order to maximize a certain reward. Reinforcement learning differs from supervised learning, as it does not require labeled input/output pairs in order to determine its level of performance. Rather, the neural network chooses to explore the problem space, finding potential actions to take and evaluating the performance of each. Genetic algorithms are sometimes implemented to naturally select the highest performer of a number of reinforcement-learning neural networks. Once the model is selected iterative deviations are created from that model, each of them subsequently evaluated in a similar fashion over the course of generations.

2 Cloud Computing Solutions

Acquiring the hardware to start machine learning projects at home can be costly, and requires a level of hardware expertise that prospective developers may want to circumvent. Large computing corporations are developing new structures of cloud computing specifically targeting the needs of developers looking for on-demand computational power. Because of the resources available to these large companies they are able to maintain stocks of servers equipped with high end processors, including CPUs, GPUs, and even specialized hardware developed to handle the intensive demands of machine learning computation. Most pricing structures either use an hour/server time model or charge for individual calls to the server processors.

2.1 Amazon Web Services EC2

Amazon offers their EC2 P2 instances, designed for GPGPU computations using CUDA and OpenCL. The service is scalable and offers hourly rates for access to GPU accelerated parallel computation. Whereas individual projects would be restricted to mid-level hardware (discussed later in the review), Amazon is able to parcel out access to some of the most powerful processors on the market. The pricing structure is roughly $1/GPU-hour with options for instances equipped with 1, 8, or 16 GPUs. Secondarily, starting an AWS account gives 750 hours of EC2 compute access for free for the first 12 months.

2.2 Google Compute Engine

Similarly to Amazon Google offers access to various CPUs and GPUs through cloud services, generally charging hourly rates for the service. Like Amazon Google has access to hardware that would cost thousands of dollars for any individual developer to purchase, allowing customers to access hardware that outpaces a home desktop by orders of magnitude. Additionally, Google offers discounts of up to 70% to those who allow for their resources to be preemptible. That is to say that high priority computations may preempt the resources being accessed mid-calculation. While it may certainly cause work-flow issues during the training process, it allows access to Googles resources for very little cost.

2.3 Google Cloud TPU

Google has invested heavily in exploring the horizons of machine learning. Not satisfied to rely on GPU acceleration, Google is developing an application specific integrated circuit called the Tensor Processing Unit (TPU) designed specifically for machine learning. While the cutting edge technology is not available for purchase, Google does offer a cloud computing service where customers may benefit from their stocks of TPU servers to manage machine learning datasets at speeds that would be impossible to reach otherwise. Google offers cloud TPU access through hourly pricing models, generally much more costly than other cloud computing services on the market but with an accelerated timeline.

3 General Purpose GPUs

GPUs have recently become explosively popular for use in computer deep learning and neural networks. Their ability to manage many parallel operations can create huge time benefits over CPU calculations. When dealing with complex inputs, image recognition chief among them, it is almost always necessary to utilize the power of a GPU to process the rich data in a timely fashion. Additionally, several GPUs may be installed in parallel, allowing for greater savings in time (at the expense of cost). Many GPUs are developed with these complex an intensive operations in mind, optimizing them for machine learning applications.

3.1 NVIDIA (GTX) 10 Series

Thanks to NVIDIAs CUDA API, NVIDIA series graphics cards saturate the field of neural network training by allowing developers to use their GPU to accelerate the process of neural network training without developing their own complex instructions. The drawback of the CUDA framework is the fact that the model and dataset must fit across the DRAM capacity of the GPU(s) in order to take advantage of GPU acceleration. This creates a demand for plentiful GPU memory in addition to raw processing power in order to ensure that substantial training sets can fit within the memory constraints. The most moderate NVIDIA GPUs on the market are the NVIDIA 10 Series graphics cards, offering a robust GPGPU that can handle training on its own or work in parallel to give higher levels of performance. Common MSRP for these cards is around $300.

3.2 NVIDIA (RTX) 20 Series

While the last generation GTX cards offer affordable performance per card, NVIDIA has recently released the new generation of RTX graphics cards with an updated architecture and faster memory speed to match. While improvement over the previous 10 series is limited, the 20 series iterates on the technology to sufficiently warrant consideration. The novelty of the technology is reflected in the price point, around $500 for the RTX 2070, but it remains one of the more cost efficient options for individual/small-team scale machine learning. Due to scale issues within this project, a 20 series graphics card would be better used in a central workstation that manages much of the workload of the greater network, as outfitting individual nodes with these GPUs is cost prohibitive.

3.3 AMD ROCm enabled GPUs

AMD is attempting to catch up to NVIDIA with its own GPU acceleration framework called Radeon Open Compute or ROCm. While the technology is lagging behind NVIDIAs current market dominance, it is beginning to add support for machine learning APIs such as TensorFlow, potentially allowing it to offer true competition in the GPGPU market. Development of the ROCm framework is ongoing, and may contain several blind spots for aspiring developers. Unless specific outside circumstance would provide advantages to the AMD series graphics cards, the ubiquity of the CUDA framework still posits NVIDIA as the GPGPU leader at this time.