Role of Neural Networks on Cloud Computing

Master of Science

Information & Communication Technology

Rich Fallat

University of Denver University College

March 12, 2020

Faculty: Galina Pildush, PhD

Director: Michael Batty, PhD

Dean: Michael J. McGuire, MLS

Abstract

This paper researches platforms and frameworks, which provide high level abstractions on top of deep neural networks, distributed computing, Kubernetes, and multi-cloud machine learning pipelines. Mentioned open source products include Keras, Kubeflow, and Pachyderm. Artificial intelligence and distributed computing require demystification. However, the research contends that an inflection point is near, which makes deep neural networks, Kubernetes, and cloud computing accessible to individuals and organizations who wish to leverage the technologies.

Table of Contents

Artificial Intelligence (AI) **1**

Deep Neural Networks (DNNs) **2**

Facial Recognition Metaphor 4
Convolutional Neural Networks (CNNs) 5
Unsupervised Learning 5

Distributed DNN **6**

Cloud Computing **8**

Google Cloud Platform (GCP) Services 9
Kubernetes 11
ML and Kubernetes 12

Kubeflow 12
Pachyderm 13

Conclusion **14**

References **15**

Artificial Intelligence (AI)

AI does not equate to computing necessarily, and this paper does not address the philosophies of AI. However, the writing leverages AI as a catch-all phrase for machine learning (ML) and subclasses of ML. Machine Learning involves computation of data to optimize the processing of data (Weill 2018, 5). ML ranges from shallow learning techniques, such as linear regression, to Deep Learning (DL), which uses neural networks. Brain neural networks inspired Deep neural networks (DNNs). Figure 1 displays a Venn diagram, which illustrates Deep learning as a subclass of both ML and AI.

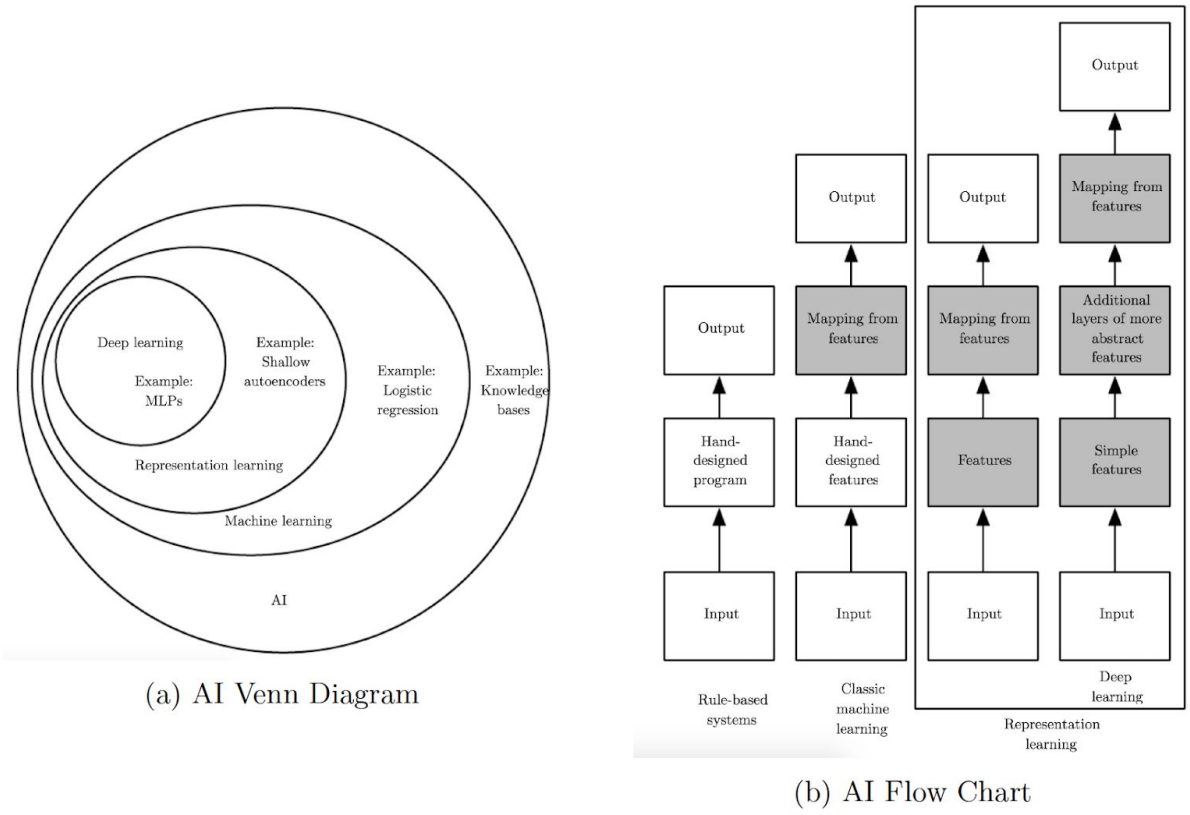

Figure 1. AI Venn Diagram. Source: (Weill 2018, 5).

Deep Neural Networks (DNNs)

Deep learning algorithms create high-level abstractions by crunching large datasets. Deep learning models accomplish perceptive tasks through a hierarchical learning process, which extracts data representations. These representations produce state-of-the-art ML results in computer vision and natural language processing (NLP) to name a few (Najafabadi et al. 2015).

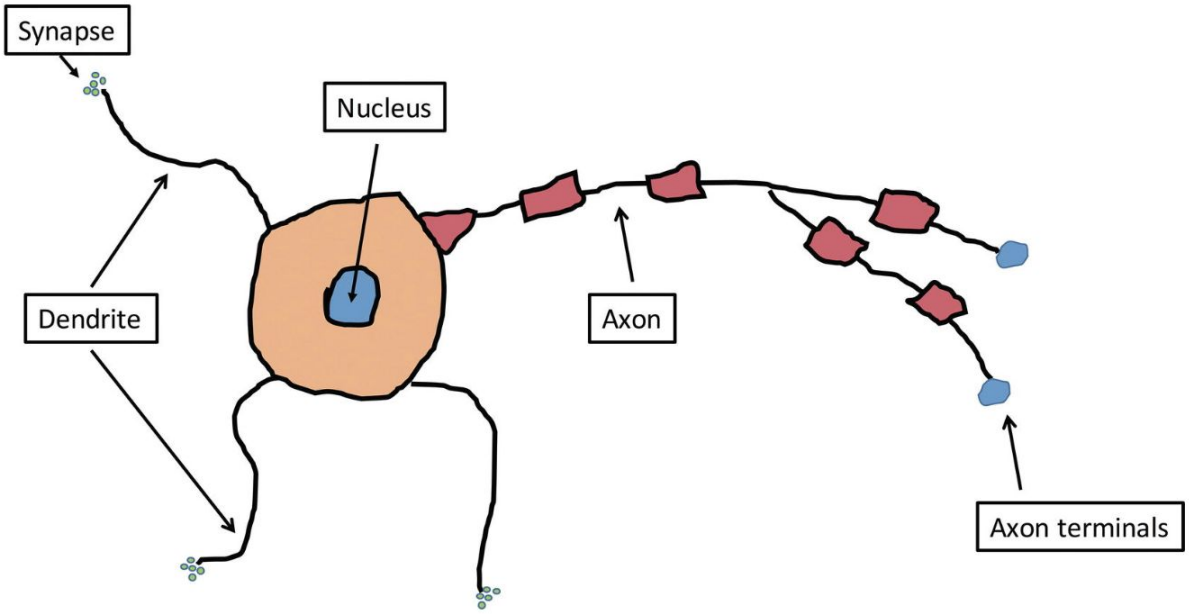

Figure 2. Brain Neuron Illustration. Source: (Bisong 2019, 5.27)

Artificial neural networks (ANNs) are a deep learning type. In nature, neurons act as a network of intelligent agents inside the brain. The neurons pass information via electric signals (Bisong 2019, 5.27). Figure 2 illustrates important parts of the biological neuron, which artificial neural networks attempt to mimic. Table 1 describes the relevant biological functionalities.

Component	*Significance*
Axon	Passes electric signals from the nucleus to other neuron cells through axon terminals
Dendrite	Receives information as electrical impulses from other neuron cells
Synapse	Passes information through dendrite to the nucleus of the neuron

Table 1. Significant Neuron Components.
Source: (Bisong 2019, 5.27)

Deep learning algorithms can identify patterns in huge amounts of data, which do not contain labels (Saiyeda 2018). Neurons connect to other neurons, and know information about adjacent neuron hierarchies. Hierarchies can contain many neurons, and model computations increase exponentially with additional neurons. Each neuron contains a representation of data, which makes the machine learning algorithm deep. Figure 3 illustrates a deep neural network (DNN), at a high-level.

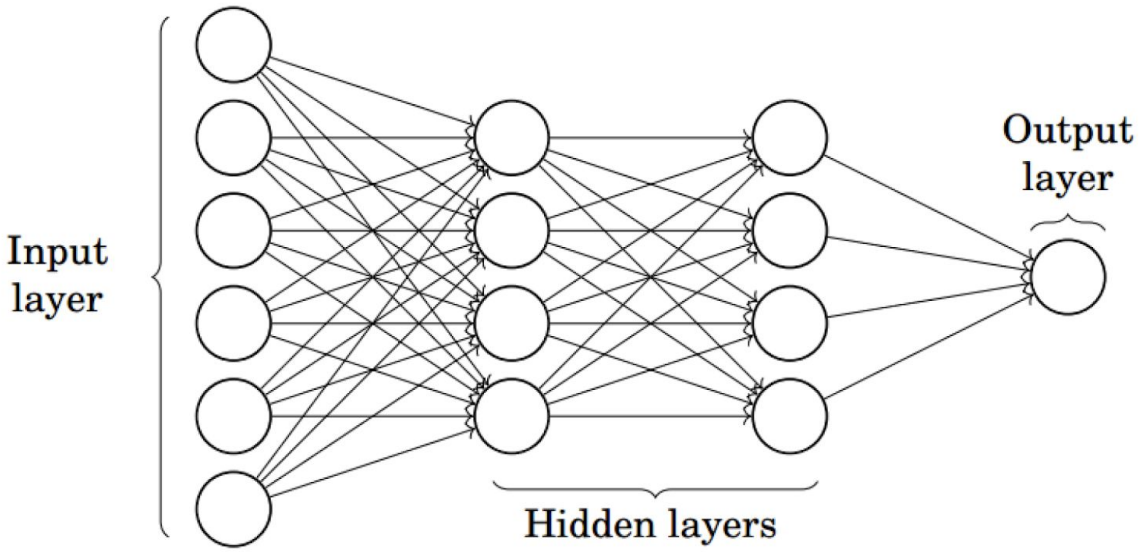

Figure 3. High-level Deep Neural Network Example. Source: (Weill 2018, 10).

Facial Recognition Metaphor

Imagine the *circles* in Figure 3 represent a bunch of lightbulbs. The *Input layer* is an image of a person. The *Output layer* is a particular person's face. At an earlier point in time, the *neural network* learned this person's face. Some light bulbs produce bright light, some bulbs produce dim light, and other bulbs produce no light. The learnt representation of a person's face corresponds to a specific combination of light bulb brightnesses. Thus, a DL model reads the image, adjusts the bulb dimmers, and calculates a probability, whether the person's face is in the image (Sanderson 2017). Facebook popularized such methods, which feed DNNs to learn faces (Saiyeda 2018). Figure 4 displays a similar metaphor with the image of a *number five*.

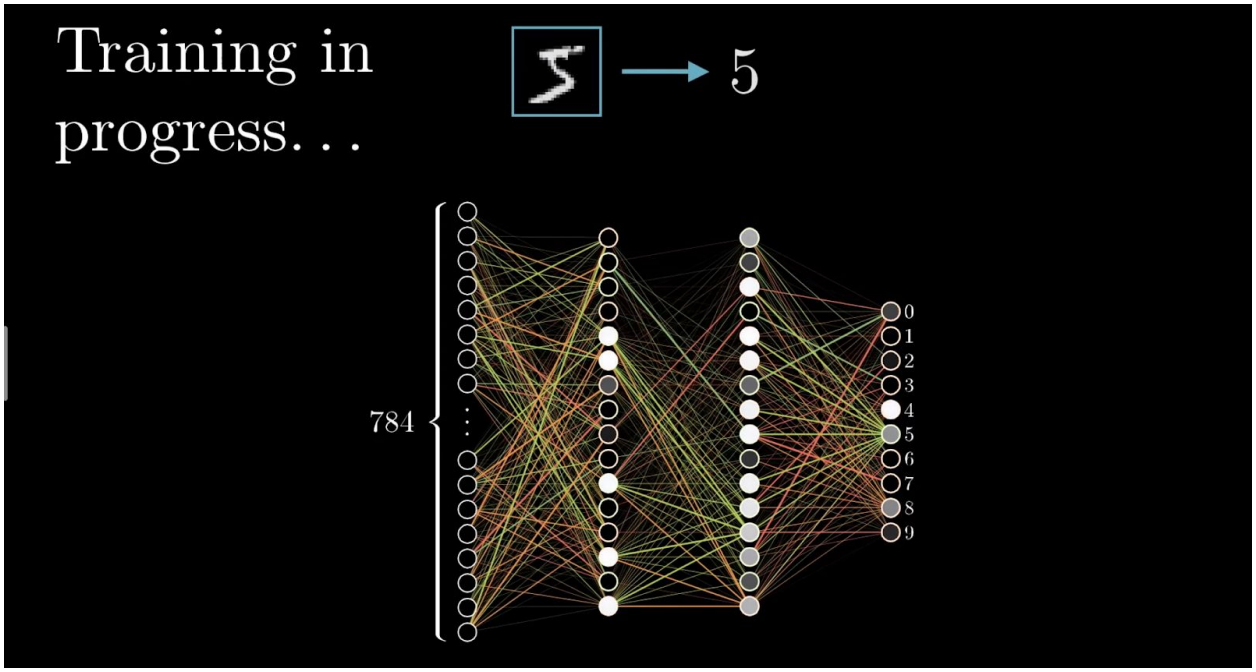

Figure 4. Convolutional Neural Network Illustration Matches a Hand-written Number Five.
Source: (Sanderson 2017).

Convolutional Neural Networks (CNNs)

The Mammalian visual cortex inspired the creation of CNNs. Image classification and object recognition leverage CNNs to automatically extract discriminant and local features from image data (Gilbert et al. 2019). Thus, convolutional neural networks are often the DNN of choice when the requirements involve image data, video data, and the like.

Unsupervised Learning

CNNs boast strong track records for image classification (Street 2018). In the above facial recognition example, what if the training data does not contain the names of people in the imagery? What if there are no labels on the input data? DNNs boast capabilities to recognize patterns in data without labels. For example, a DNN can cluster unlabeled images,

which contain the same person's face. Along the same lines, unsupervised learning can automate image tagging.

In addition, unsupervised models learn from other data types. Search engines bolster semantic indexing by leveraging unsupervised learning, which improves speed and efficiency. Google's word2vec automates semantic representations from a huge text corpus. To accomplish such tasks, Google leverages a distributed framework, which trains neural networks on massive amounts of text data (Najafabadi et al. 2015).

Distributed DNN

DL models improve with training. For example, facial recognition neural networks get better by increasing the number of images that contain human faces. More images coincide with more data. DNNs could train on tens of millions of images. In general, neural network training requires significant CPU and GPU resources, and a single machine may not contain enough horsepower. To put it another way, parallel computing can create efficiency and reduce training clock time. Figure 5 aims to persuade audiences to adopt cloud native infrastructures and distributed computing.

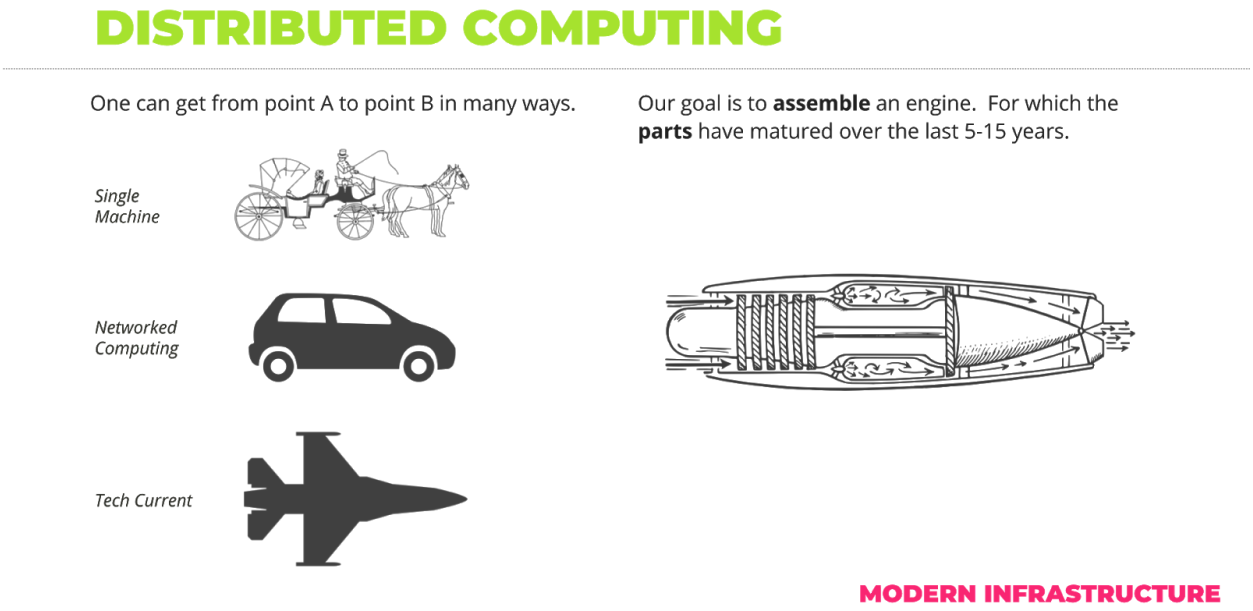

Figure 5. Distributed Computing Persuasion Infographic.

Distributed DNNs provide a means to cut down on training times and increase training iterations. Distributed DL frameworks allow high performance and scalability across multiple machines and GPUs (Shi, Wang, and Chu 2018). Popular distributed frameworks include:

- Caffe-MPI
- CNTK
- MXNet
- TensorFlow
- PyTorch
- Pachyderm

Prior to 2015, Google leveraged an in-house software framework, DistBelief, to train large-scale AI models. DistBelief could utilize computing clusters with thousands of machines, support parallelism across machines, and train DNNs, which contained billions of parameters (Najafabadi et al. 2015). Large data centers possess resources, which unlock the feasibility to train complex and evolving DNNs.

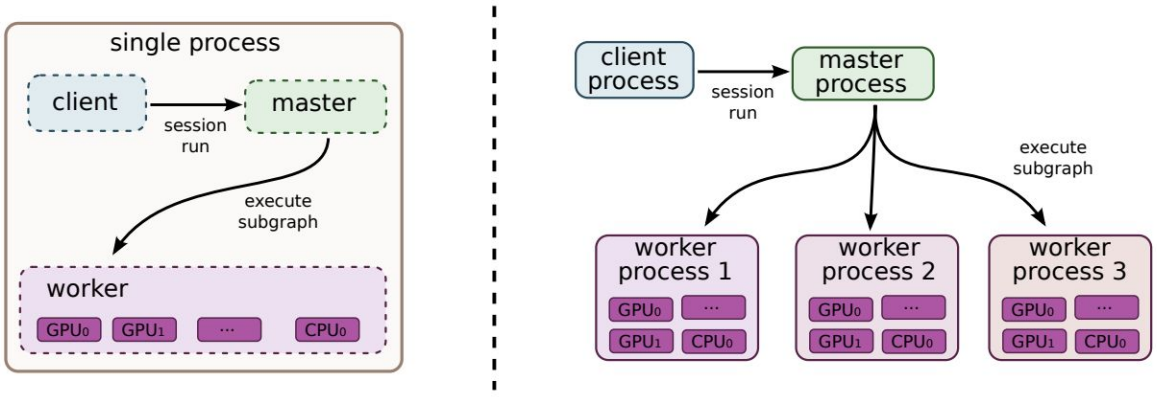

Figure 6. TensorFlow Single Machine and Distributed System Structure. Source: (Abadi et al. 2016)

In 2015, Google open-sourced the TensorFlow application programming interface (API) under the Apache 2.0 license.

> "TensorFlow takes computations described using a dataflow-like model and maps them onto a wide variety of different hardware platforms, ranging from running inference on mobile device platforms such as Android and iOS to modest sized training and inference systems using single machines containing one or many GPU cards to large-scale training systems running on hundreds of specialized machines with thousands of GPUs (Abadi et al. 2016)."

Figure 6 demonstrates TensorFlow structures on a single machine and on distributed systems. Google developed TensorFlow to compute DNNs with hundreds of billions of parameters and example records (Abadi et al. 2016). In 2019, Google released TensorFlow 2.0, which eases model building with Keras. A topic covered in the following section. Distributed frameworks matured over the last several years, and are available under open source licensing.

Cloud Computing

Cloud providers promise to eliminate the constraining nature of purchased hardware. Customers choose the type, amount, location, specifications, and the like, when leasing hardware from a cloud service. Providers shoulder the burden of maintenance, power, reliability, upgrades, and so on. In addition, cloud providers include services, which are fine-tuned to run on their infrastructure. Cloud service providers bundle Big Data platforms,

which can run data science computations, such as DNNs. Figure 7 demonstrates the complexity of a machine learning pipeline, which cloud providers support and simplify.

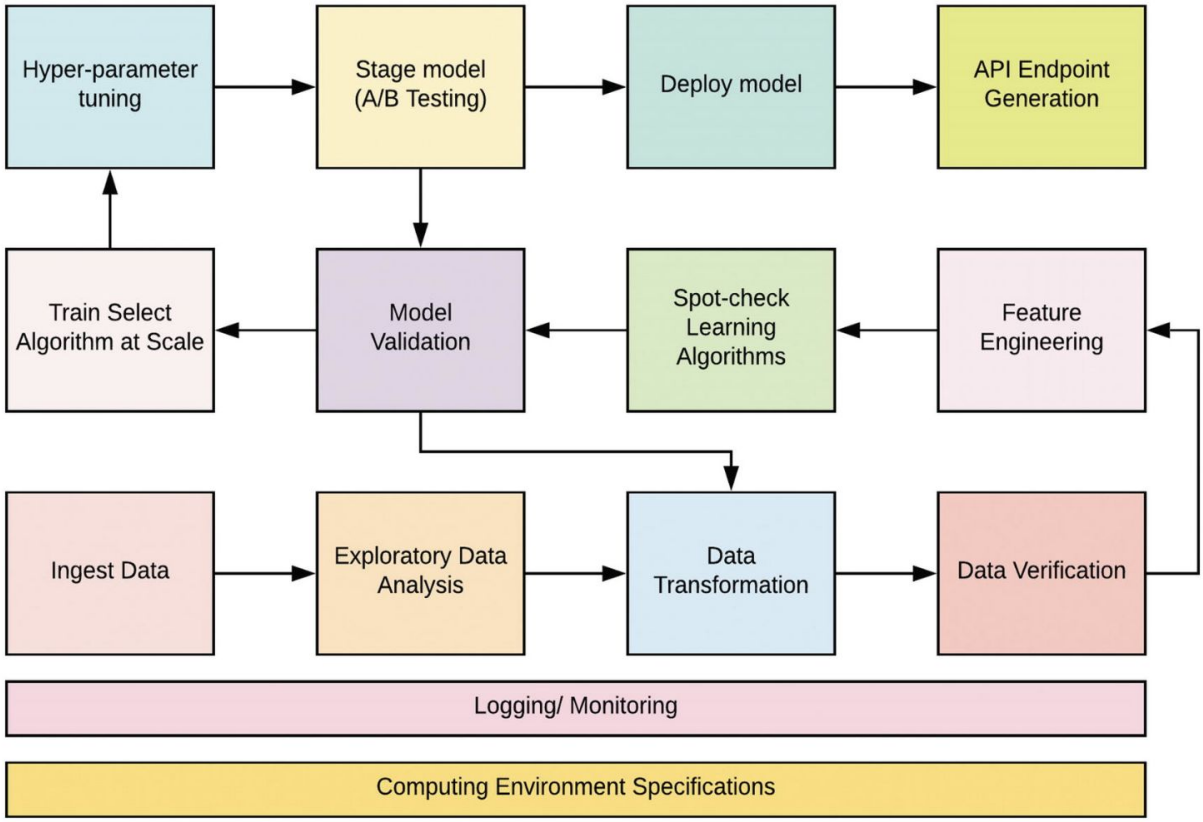

Figure 7. Machine Learning Pipeline. Source: (Bisong 2019, 8.46)

Options for cloud services seem to grow at a high rate. The following section digs in to one of the large cloud service providers, and their distributed offerings, which can train DNNs.

Google Cloud Platform (GCP) Services

As mentioned, Google created both DistBelief and TensorFlow to aid their distributed deep neural network pipelines. With TensorFlow 2.0, Google embraced Keras, which is a high level specification for building DNNs. Keras seems to promise a lower barrier of entry for individuals and organizations who wish to get their hands dirty with DNNs.

Google Cloud Platform (GCP) provides services to customers who leverage Google Cloud. Table 2 displays a brief overview of select services offered on Google Cloud Platform.

Service	*Select Features*
Google BigQuery	● Managed data-warehouse product ● Optimized for data analytics ● One of many serverless products
Google Cloud Dataprep	● For quick data exploration and transformation ● Google Cloud Dataflow distributed processing
Google Cloud Dataflow	● Serverless, parallel, and distributed infrastructure ● Runs batch and stream processing jobs ● Major piece of data/ML pipeline on GCP
Google Cloud Machine Learning Engine (MLE)	● For training and serving large scale ML models ● Modes for serving or consuming trained models ● Uses online prediction to auto-scale infrastructure
Google AutoML: Cloud Vision	● Facilitates vision models for image recognition
Google AutoML: Cloud Natural Language Processing	● Natural Language Processing (NLP)
Google Kubernetes Engine (GKE)	● Managed environment for deploying containers

Table 2. Google Cloud Platform Services.
Source: (Bisong 2019, 7.38)

In addition, Google created Kubernetes, which leads as a top container orchestration system. Google possesses an innovative track record, and individuals can safely bet on GCP to begin DL exploration. At this point, cloud computing provides bountiful options. The aforementioned research is not comprehensive, and does not take a firm stance on cloud providers. Amazon Web Services (AWS) (Kaul et al. 2018), Microsoft Azure (Briggs 2019), and the like offer similar Big Data services to their customers.

Kubernetes

Containers date back to the seventies, but re-emerged over the last decade. History demonstrates that many organizations did not leverage their compute resources in an efficient manner. Before virtualization, a single monolithic application may reside on a single machine and occupy the resources of the entire machine. Next, virtual machines provided a means to run multiple applications and operating systems on a single machine. Finally, containers covered more unused resource gaps by sharing, for example, libraries and a single operating system, with other containers. Figure 8 displays the evolution of virtualization, at a high level.

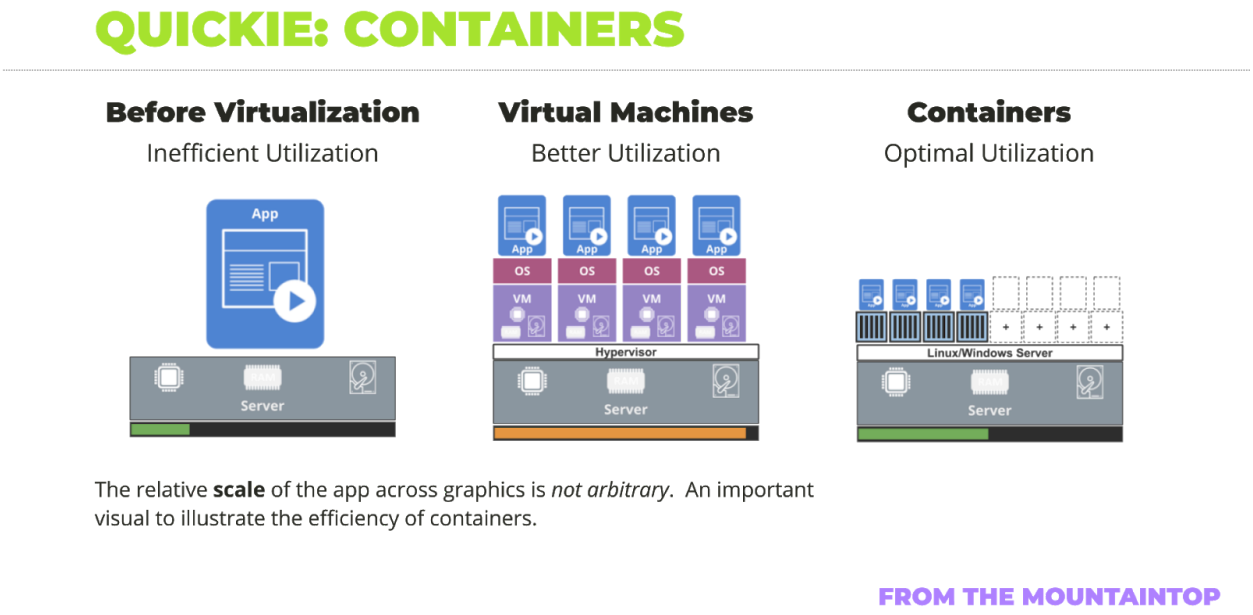

Figure 8. High-level History of Virtualization

Kubernetes provides an automated orchestration system for containers. Kubernetes dynamically scales containers, heals failed compute nodes, load balances compute clusters, and uses domain name system (DNS) to manage containers. In short, Kubernetes offers reliability, scalability, and dynamic infrastructure scaffolding for machine learning pipelines. Google developed both Kubernetes and GCP services which seems to indicate a tight bridge.

ML and Kubernetes

The following subsections explore two ML platforms, which run on Kubernetes, and unlock the capability to stretch ML pipelines across multiple cloud service providers.

Kubeflow

Google released Kubeflow, which demonstrated their eagerness to get folks building and training distributed ML models. Artificial neural networks, cloud computing, and

Kubernetes seem to be an incredibly wide and deep stacks of technologies. At this point in time, open source provides a wealth of pipeline tooling to create ML pipelines. Google designed Kubeflow to extend Tensorflow, and in 2017, open sourced the machine learning toolkit (Kubeflow n.a.).

> "The goal of Kubeflow is to abstract away the technicalities of managing a Kubernetes cluster so that a machine learning practitioner can quickly leverage the power of Kubernetes and the benefits of deploying products within a microservice framework. Kubeflow has its history as an internal Google framework for implementing machine learning pipelines on Kubernetes (Bisong 2019, 8.46)."

Kubeflow provides a means to run popular machine learning components on Kubernetes, such as TensorFlow, JupyterLab, and others. In addition, Kubeflow supports multi-cloud machine learning pipelines. In other words, one can create pieces of an ML pipeline on GCP, AWS, and Microsoft Azure, which Kubernetes and Kubeflow glue together. As Paul Krill states, "Kubernetes tasks itself with making it easier to manage distributed workloads, while Kubeflow centers on making the running of these workloads portable, scalable, and simple (Krill 2018)."

Pachyderm

Pachyderm is a data analytics platform, which runs on containers and Kubernetes. Pachyderm also boasts the ability to run multi-cloud ML pipelines. Nitin Naik believes the platform displays promise and may replace Hadoop (Naik 2017). However, as of this writing, Pachyderm lacks a critical mass of users and research publications. Pachyderm differentiates itself by creating a system of data lineage. In other words, the platform version controls things like data, AI models, and so on, which provides the ability to roll back and audit prior states.

Pachyderm qualifies as a novel example, which abstracts the complexity of Kubernetes, distributed computing, and multi-cloud ML pipelines.

Conclusion

This paper described some of the opportunities associated with creating legitimate deep neural networks at scale. For example, Keras and GCP services lower the barrier of entry to create DNNs. Organizations battle-tested their systems, and now provide said systems as open source platforms, toolkits, and so on. Individuals or companies who race into these technologies from the ground up may face an incredibly high hill to climb. Platforms such as Kubeflow and Pachyderm provide climbing gear for those new to the technology. They make Kubernetes accessible. They provide ability for consumers to shop the market for cloud resources, which suit their needs, and also the glue to connect multiple clouds into a cohesive ML pipeline. They help lay a foundation towards cloud ubiquity and reduced vendor lock-in. Perhaps most important, they provide data scientists a means to concentrate on developing AI models by eliminating the constraints imposed by a single machine.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2016. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." Ithaca: Cornell University Library. arXiv.org.

Ahn, Shinyoimg, Joongheon Kim, and Sungwon Kang. 2018. "A Novel Shared Memory Framework for Distributed Deep Learning in High-performance Computing Architecture." *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings (ICSE '18).* New York: Association for Computing Machinery. 191–192.

Bisong, Ekaba. 2019. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Apress.

Briggs, Bill. 2019. "How Machine Learning is Unlocking the Secrets of Human Movement – And Reshaping Pro Sports." In *news.microsoft.com*. https://news.microsoft.com/transform/machine-learning-unlocking-secrets-human-movement-reshaping-pro-sports/.

Chentanez, Nuttapong, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. 2018. "Physics-based Motion Capture Imitation with Deep Reinforcement Learning." *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games (MIG '18),* no.1 New York: Association for Computing Machinery. 1–10.

Gibert, Daniel, Carles Mateu, Jordi Planes, and Ramon Vicens. 2019. "Using Convolutional Neural Networks for Classification of Malware Represented as Images." *Journal of Computer Virology and Hacking Techniques*, no. 15. 15–28.

Krill, Paul. 2018. "Kubeflow brings Kubernetes to Machine Learning Workloads." In *InforWorld.com.* San Mateo: CA. (August): https://du.idm.oclc.org/login?url=https://search-proquest-com.du.idm.oclc.org/docview/2097845259?accountid=14608

Kaul, Pallavi, Henirk Agarwal, and Gaurav Raj. 2018. "Effective Prediction in Amazon Web Service based Clustered Data using Artificial Neural Networks." *International Conference on Advances in Computing and Communication Engineering (ICACCE).* Paris: IEEE. (June).

Kubeflow. n.a. "An Introduction to Kubeflow." In *kubeflow.org.* Accessed: 2020. (March) https://www.kubeflow.org/docs/about/kubeflow/

Naik, Nitin. 2017. “Docker Container-based Big Data Processing System in Multiple Clouds for Everyone.” *2017 IEEE International Systems Engineering Symposium (ISSE).* Vienna: IEEE. 1-7.

Najafabadi, Maryam M., Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. “Deep Learning Applications and Challenges in Big Data Analytics.” *Journal of Big Data* 2, no. 1. 1–21.

Sanderson, Grant. 2017. “But What is a Neural Network? Deep learning: Chapter 1.” In *3Blue1Brown*. Season 3, no. 1. (October): https://youtu.be/aircAruvnKk.

Saiyeda, Anam, and Mansoor Ahmad Mir. 2017. “Cloud Computing for Deep Learning Analytics:A Survey of Current Trends and Challenges.” *International Journal of Advanced Research in Computer Science* 8, no. 2. (March): www.ijarcs.info.

Shi, Shaohuai, Qiang Wang, and Xiaowen Chu. 2018. “Performance Modeling and Evaluation of Distributed Deep Learning Frameworks on GPUs.” Ithaca: Cornell University Library.

Street, Alexander. 2018. “Practical Convolutional Neural Networks.” In *PACKT Publishing*. Directed by Anonymous. Produced by PACKT. https://video-alexanderstreet-com.du.idm.oclc.org/watch/practical- convolutional-neural-networks.

Weill, Edwin. 2018. “Edge-Computing Deep Learning-Based Computer Vision Systems.” Order no. 13420170. Clemson: Clemson University.