A REVIEW ON TENSORFLOW: AS DEEP LEARNING FRAMEWORK FOR COMPUTER VISION

By: **Lateef Olalekan ADERINOYE (1)**
Osarumen Solomon OSAZUWA (2)
Divine BAKALA (3)

E-mail: LESTUDENT01@GMAIL.COM
SOLOLAMBERT20@GMAIL.COM
DIVINEBAKALA@YAHOO.COM

Abstract: - Deep learning is a branch of artificial intelligence employing deep neural network architectures that has significantly advanced the state-of-the-art in computer vision, image recognition and other domains. In November 2015, Google released TensorFlow, an open source deep learning software library for defining, training and deploying machine learning models. In this paper, we breifly reviewed TensorFlow and put it in context of modern deep learning concepts and software. We discussed its basic computational paradigms and distributed execution model, its programming interface as well as accompanying visualization toolkits. Then, we compare the TensorFlow with few other frameworks, such as Torch or Caffe and finally comment on observed use-cases of TensorFlow industry.

Keywords: Artificial Intelligence, Machine Learning, Neural Networks, Distributed Computing, Open source software, Software packages

I. Introduction:

Deep Learning (DL) is a subset of NNs that makes the computational multi-layer NN feasible. Typical DL architectures are deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GAN), and many more [9].

Modern artificial intelligence systems and machine learning algorithms have revolutionized approaches to scientific and technological challenges in a variety of fields. We could observed remarkable improvements in the quality of state-of-the-art in computer vision, natural language processing, speech recognition and other techniques. Moreover, the benefits of recent breakthroughs have trickled down to the individual, improving everyday life in numerous ways. Personalized digital assistants, recommendations on e-commerce platforms, financial fraud detection, customized web search results and social network feeds as well as novel discoveries in genomics have all been improved, if not enabled, by current machine learning methods. [8] [9].

It is necessary to note that before now while deep learning algorithms and individual architectural components such as representation transformations, activation functions or regularization methods may initially be expressed in mathematical notation, they must be transcribed into a computer program for real world usage. For this reason, there exist a number of open source as well as commercial machine learning software libraries and frameworks. Among these are Theano [1], Torch [8], and many more. In November 2015, this list was extended by TensorFlow, a novel machine learning software library released by Google [8]. As per the initial publication, TensorFlow aims to be “an interface for expressing machine learning algorithms” in “large-scale [...] on heterogeneous distributed systems” [7]. **Section 2** of this paper briefly discussed the TensorFlow framework.

II. A Brief History of Tensorflow as a Deep Learning Framework

This section systematically discussed very briefly the historical background of Tensorflow as a Deep Learning Framework for and not limited to Computer Vision.

A. Deep Learning

Deep learning frameworks have played a crucial role in disseminating ideas. The first generation of frameworks allowing for easy modeling encompassed are, but not limited to, Caffe [3], Torch [4], and Theano [5]. Many seminal papers were written using these tools. By now, they have been superseded by TensorFlow [6] (often used via its high level API Keras [7]), CNTK [9], and Apache MXNet [1].

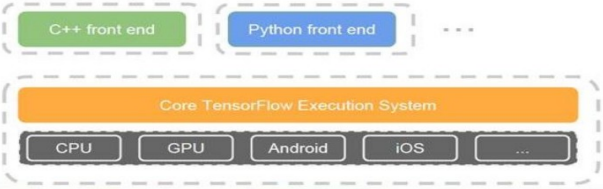
The Tensorflow Architecture

TensorFlow, edges represent data flowing from one operation to another and are referred to as tensors. A tensor is a multi-dimensional collection of homogeneous values with a fixed, static type. The number of dimensions of a tensor is termed its rank. A tensor's shape is the tuple describing its size, i.e. the number of components, in each dimension [8].

TensorFlow is a software library or framework, designed by the Google team in November 2015 to implement machine learning and deep learning concepts in the easiest manner [8]. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions easily with the help of multi-dimensional arrays called tensors. It is a programme in support of deep neural networks and machine learning techniques, a high scalable feature of computation with various datasets and uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used [12**]**.

In TensorFlow, machine learning algorithms are represented as computational graphs. The architecture is represented by the followings:

- Low-level core (C++/CUDA)
- Simple Python API to define the computational graph
- High-level API (TF-Learn, TF-Slim, soon Keras…)

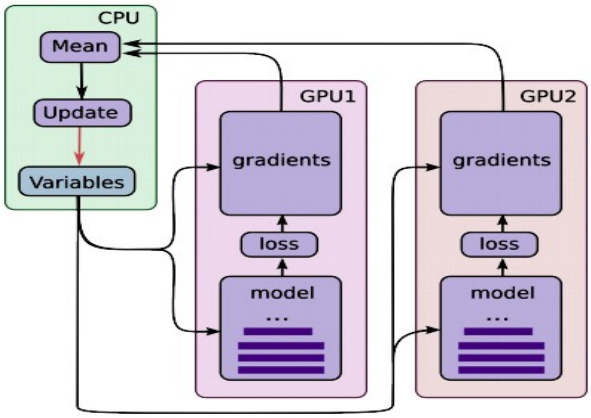


TensorFlow Architechture (Figure 1). Source: Deep Learning Group, 2017

III. TensorFlow computational graph

The TensorFlow computational graphical comprised the following according to Deep Learning Group, 2017:

- Auto-differentiation!
- easy multi-GPU/multi-node - native C++ multithreading
- device-efficient implementation for most ops
- whole pipeline in the graph: data loading, preprocessing, prefetching...



TensorFlow computational graph (Figure 2). Source: Deep Learning Group, 2017

A. TensorFlow Development

+ bleeding edge (GitHub yay!)

+ division in core and contrib => very quick merging of new hotness

+ a lot of new related API: CRF, BayesFlow, SparseTensor, audio IO, CTC, seq2seq

+ so it can easily handle images, videos, audio, text...

+ if you really need a new native op, you can load a dynamic lib

- sometimes contrib stuff disappears or moves

- recently introduced bells and whistles are barely documented.

B. The TensorFlow Programming Interface

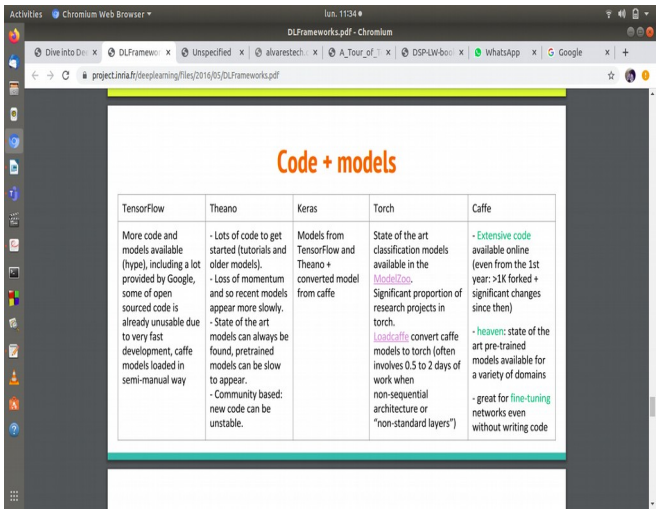
Having conveyed the abstract concepts of TensorFlow's computational model in Section III, we will now concretize those ideas and speak to TensorFlow's programming interface.

We begin with a brief discussion of the available language interfaces. Then, we provide a more hands-on look at TensorFlow's Python API by walking through a simple practical example. Lastly, we give insight into what higher-level abstractions exist for TensorFlow's API, which are especially beneficial for rapid prototyping of machine learning models.

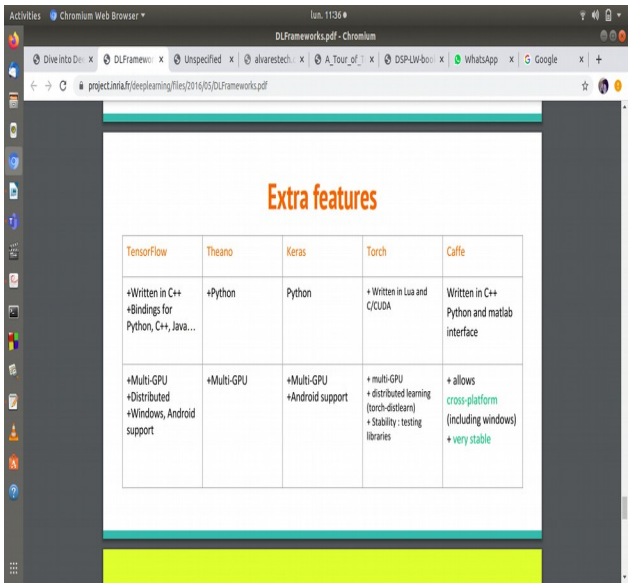
Interfaces

There currently exist two programming interfaces, in C++ and Python, that permit interaction with the TensorFlow back-end. The Python API boasts a very rich feature set for creation and execution of computational graphs. As of this writing, the C++ interface (which is really just the core backend implementation) provides a comparatively much more limited API, allowing only to execute graphs built with Python and serialized to Google's Protocol Buffer 20 format. While there is experimental support for also building computational graphs in C++, this functionality is currently not as extensive as in Python [1].

C. Comparing TensorFlow with few other frameworks



Comparison of TensorFlow and other framewors (Figure 3)



Extra Features (Figure 4)

IV Use Cases of Tensorflow Today

We begin with a review of selected mentions of TensorFlow in literature. Then, we discuss where and how TensorFlow is used in industry.

A. In Literature: The first noteworthy mention of TensorFlow is [1], published by Szegedy, Ioffe and Vanhoucke of the Google Brain Team in February 2016. In this work, the authors use TensorFlow to improve on the Inception model

[4], which achieved best performance at the 2014 ImageNet classification challenge.

B. In Industry: Adoption of TensorFlow in industry is currently limited only to Google, at least to the extent that is publicly known. I have found no evidence of any other small or large corporation stating its use of TensorFlow. As mentioned, I link this claim to [1].

Recently, Google has begun augmenting its core search service and accompanying PageRank algorithm [1] with a system called RankBrain [1], which makes use of TensorFlow. RankBrain uses large-scale distributed deep neural networks for search result ranking. According to [1], more than 15 percent of all search queries received on www.google.com are new to Google's system. RankBrain can suggest words or phrases with similar meaning for unknown parts of such queries. There are other areas where google uses TensorFlow but stopped due to scope of this paper.

IV. Conclusion

We discussed TensorFlow, a novel open source deep learning framework based on computational graphs. Its ability to perform fast automatic gradient computation, its inherent support for distributed computation and specialized hardware as well as its powerful visualization tools make it a very welcome addition to the field of machine learning. Its low-level programming interface gives fine-grained control for neural net construction, while abstraction libraries such as TFLearn allow for rapid prototyping with TensorFlow. In the context of other deep learning toolkits such as Theano or Torch, TensorFlow adds new features and improves on others.

References:

[1] Collobert, R., Bengio, S. and Marithoz, J., "Torch: A modular machine learning software library," 2002

[2] Cordell, R., "Machine Learning + Libraries: A Report on the State of the Field", pp. 6, Version Published: July 14, 2020

[3]Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding.arXiv preprint arXiv:1810.04805.

[4]Doucet, A., De Freitas, N., & Gordon, N. (2001). An introduction to sequential monte carlo methods. Sequential Monte Carlo methods in practice (pp. 3–14). Springer.

[5]Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121–2159.

[6]Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

[7]Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2414– 2423).

[8]Goldsborough, P., "A Tour of TensorFlow", Proseminar Data Mining, arXiv:1610.01178v1 [cs.LG] 1 Oct 2016. [Online].

[9]LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[10]Nair, V., and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML-10), J. FÃijrnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814.

[11]Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López A., Ignacio Heredia, G., Malík, P., and Hluchý, L., "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey", published online by the Author, 2019

[12]Tutorials Point (I) Pvt. Ltd., 2018 [Online]. Available:https://www.tutorialspoint.com/tensorflow/tensorflow_tutorial.pdf

[13]Zhang A., Zachary C. Lipton, Mu Li, and Alexander J. S., "Dive into Deep Learning", Release 0.15.1, November 6, 2020.