TENSORFLOW

"I think the brain is essentially a computer and consciousness is like a computer program. It will cease to run when the computer is turned off. Theoretically, it could be re-created on a neural network, but that would be very difficult, as it would require all one's memories."

– Stephen Hawking

An understanding of Google's very own open-source machine learning framework for everyone

Submitted by,
Harish Bharatham
Anannya Das
Shilpa Verma
Preena Brar

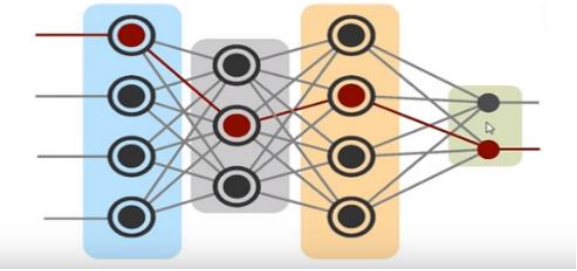
AN INTRODUCTION TO TENSORFLOW –EXPLANATION AND IMPLEMENTATION OF DEEP LEARNING AND NEURAL NETWORKS.

Abstract:
In this paper we intend to provide an insight no how neural networks work and how they can be used to implement deep learning algorithms that can provide better solutions than machine learning algorithms. This paper will walk you through the concepts of deep learning and how neural networks are integrated into it. It will also explain, form the roots, the concepts of tensorflow, difference between machine learning and deep learning, tensorflow execution model. It will take into a detailed process of how we implemented the deep learning in our example and how we displayed it using tensorboard. Finally, to provide a better understanding of neural networks, we explain an interactive way of learning neural networks, Tensorflow playground.

Index – TensorFlow, Deep Learning, Neural Network, TensorBoard

Introduction to Deep learning

-Deep learning is inspired from how a human brain works. It is inspired from our brain cells basically which are called neurons. Multiple artificial neurons are combined together to form different layers as you can see in the diagram.



The learning occurs basically by strengthening the connection between two neurons when both are active at the same time during training. How are the neurons modeled? Each has a propagation function that transforms the outputs of the connected neurons, often with a weighted sum. The output of the propagation function passes to an activation function, which fires when its input exceeds a threshold value.

Difference between machine learning and deep learning

-Most important factor is the performance as the scale of data in quantity changes. Deep learning algorithms won't perform very well because it needs large amount of data to understand it perfectly.
-Deep Learning algorithms heavily depend on high end machines contrary to traditional machine learning algorithms. Because, the requirements of Deep Learning algorithms include GPU's, which are an integral part of its working. Deep Learning algorithms inherently do large amount of matrix multiplications and that is the reason they require high end machines.
-In machine learning, most of the features are to be identified by an expert and then it has to be hand coded and performance of those algorithms depends upon how accurately features are identified and extracted. But deep learning algorithms try to learn high level features from data itself. This is a very distinctive feature of deep learning.

What are Tensors?

-In tensorflow, data is represented in the forms of tensors. We can think of tensors as an n-dimensional array or list. Tensors, defined mathematically, are simply arrays of numbers, or functions, that transform according to certain rules under a change of coordinates.

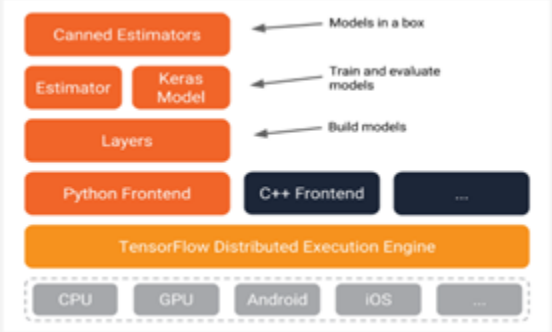
What is TensorFlow?

It is an open source software library released in 2015 by Google. It works mainly by first defining and describing a modeling abstract and then making it reality in sessions. First, we define the computations (what you are going to do) and the run it in sessions. Tensors flow through computational graphs and functions.

Its flexible architecture allows easy deployment of computations across a variety of platforms (CPU's, GPU's, TPU's),

And from desktops to clusters of servers to mobile devices. It comes with strong support for machine learning and deep learning, and flexible numerical computation core is used across many other scientific domains.

The TensorFlow execution model



TensorFlow is cross-platform. It runs on nearly everything: GPUs and CPUs—including mobile and embedded platforms—and even tensor processing units (TPUs), which are specialized hardware to do tensor math on which aren't widely available yet, but have recently been launched as an alpha program.

The TensorFlow distributed execution engine abstracts away the many supported devices and provides a high performance-core implemented in C++ for the TensorFlow platform. On top of that sit the Python and C++ frontends of TensorFlow. The TensorFlow layers module provides a high-level API that makes it easy to construct a neural network. The CNN consists of mainly three layers: Convolutional layers - composed of a stack of convolutional modules that perform feature extraction using ReLu(rectified linear unit) activation function - the activation function of a node defines the output of that node given an input or set of inputs, Pooling layers - extracts sub-regions of the feature map and keeps only required values and discarding the rest, Dense (fully connected) layers : perform classification on the features extracted by the convolutional layers and down-sampled by the pooling layers.

Then we have the estimators - a high-level TensorFlow API that greatly simplifies machine learning programming by encapsulating the actions of training, evaluation, prediction and export for serving. In layman terms, they comprise of a number of commonly used models that are ready to be used out of the box. Estimators provide a safe distributed training loop that controls how and when to:

- build the graph
- initialize variables
- start queues
- handle exceptions
- create checkpoint files and recover from failures
- save summaries for TensorBoard

Methods of Execution of Code

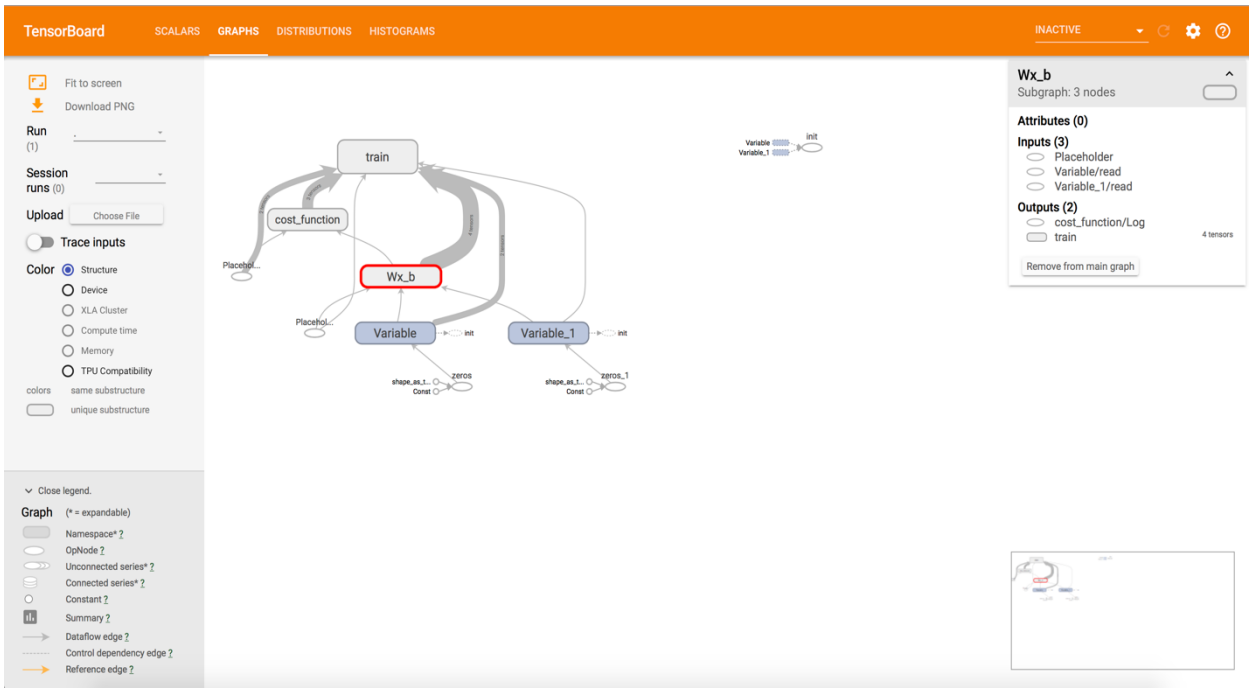
1) Graphs:

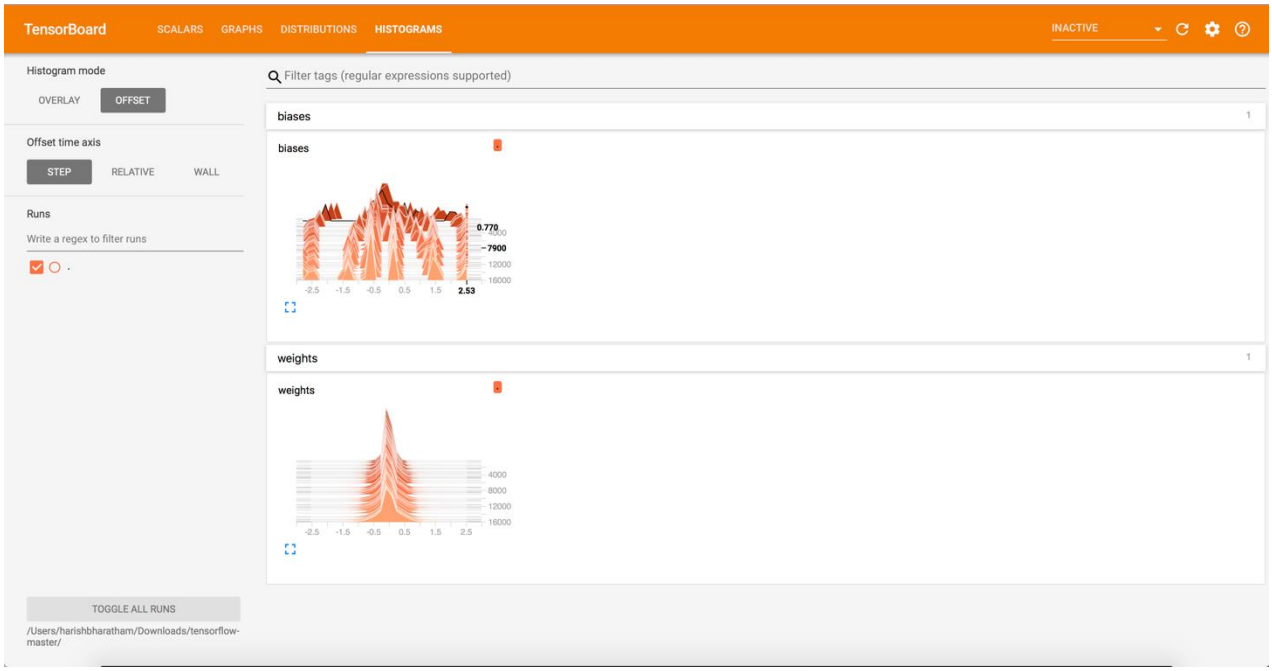
We write code to build a computation graph, then execute it.

2) Eager Execution:
We will be executing TensorFlow kernels immediately, rather than constructing graphs that will be executed later. Once you are satisfied with your TensorFlow code running eagerly, you can convert it to a graph automatically. This will make it easier to save, port, and distribute your graphs.

How is code executed?
Session is a class for running TensorFlow operations. A Session object encapsulates the environment in which Operation objects are executed, and Tensor objects are evaluated. In layman terms, session places the graph operations on the CPUs and GPUs and provides methods to execute them.
We used and input_data file to be imported into python through a board file that we ran through command shell. The code takes 29 iterations to reach to an accuracy level of 91.65% with cost value of 18.095. This process leaves output log directory in the a temporary folder that we use later to direct to tensorboard. The command tensorboard --logdir "location_of_file_on_computer" will provide a link to the localhost running tensorboard.

Tensorboard Execution:
At the end of our prediction in the python code, we were able to run tensorboard on our localhost 6006. Tensorboard is a visualization of how data flows in several nodes and functions and shows the graph of weights and biases. It also shows the cost function performance with the iterations.
There are four representations of data in tensorboard, Scalars, Graphs, Distributions and Histograms. Scalars display the cost-function graph.





Graphs display the flow of tensors (data) between the variables and the functions. Distributions and histograms shows the graphs of biases and weights. The distributions can be more classified by selecting the horizontal axis as step, relative and wall. This can be done with the histogram as well and the graph can be toggled from offset to overlay which changes the visualization from three to two dimensions.

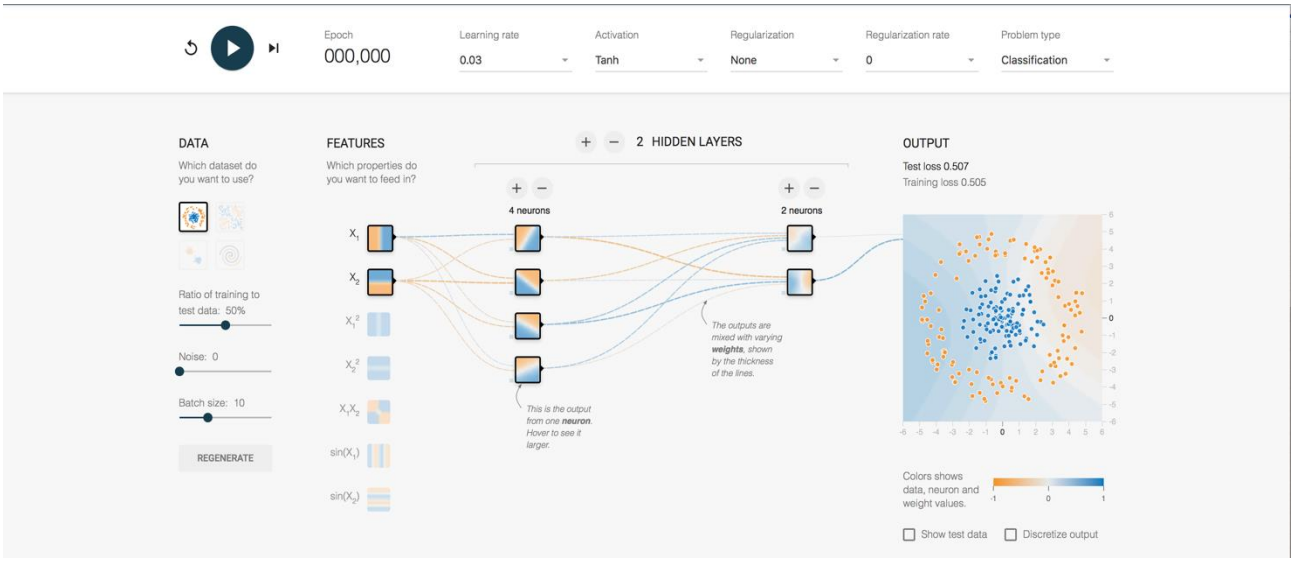
Tensorflow playground:

The tensorflow playground is an online application created by google to show the mechanism or working of the neural networks in deep learning. This application has four different datasets that are on different levels of complexity. Each dataset has a blue and orange dots where blue represents most positive values and orange represents most negative values. The application will use a function to classify these dots and separate them using a function.

The datasets have several features in which we get to choose what features to be included in, to classify the dataset. These features are passed through a function to be solved or classified. The connections between these functions represent the flow of data and the stronger the connection more weight the line is carrying. These lines are from features to functions and carry tensors (data).

The features are passed through hidden layers that contains several neurons. We can select the number of hidden layers and the neurons in them there by making the function as complex as we want. The more complex the function gets the accurate the model becomes.

For the complex datasets we need more neurons, features and hidden layers so as to classify the data properly. We can customize the data with the noise, batch size, train and test data so as to get different results every time.



The activation function can also we be changed which changes the partition type of the dataset. We have Tanh as default but we can also select Sigmoid, reLU and Linear functions. We can also change the problem time entirely from classification to regression.

Once we set all the parameters are desired, we can start the function and it takes a couple of seconds depending on the complexity of the function, to reach to the maximum efficiency of the model. The tensors keep iterating from the features to the model again and again until the test loss is close to 0. As the dataset is more complex the more iterations it takes for the tensors to reach to minimum test loss. Once the iterations reach to an extent where the test loss is almost 0, the program still runs but there will not be any improvement in the output.

The dataset will be classified to the maximum extent and you can see the result of the activation function that you have selected. This is a very good example to explain deep learning and neural networks that we decided to present in class so as to make everyone understand easily.

Conclusion:

Tensorflow uses the concept of neural networks to introduce us to a concept called deep learning. As every industry is moving towards machine learning and data mining to improve business, deep learning deals with more amounts of data and uses more complex algorithms to create the best model itself by iterating the data through the model again and again. We decided presenting our ideas on tensorflow would help all of us in understanding this upcoming concept on the market that promises the application of machine learning concepts to several other huge systems.

REFERENCES

[1]: http://neuralnetworksanddeeplearning.com/index.html - Neural Networks and Deep Learning - By Michael Nielsen

[2]: https://towardsdatascience.com/introducing-deep-learning-and-neural-networks-deep-learning-for-rookies-1-bd68f9cf5883 - Introducing Deep Learning and Neural Networks—Deep Learning for Rookies by Nahua Kang

[3]: https://opensource.com/article/17/11/intro-tensorflow - What is the TensorFlow machine intelligence platform? by Amy Unruh

[4]: https://web.stanford.edu/class/cs20si/2017/lectures/slides_01.pdf - Welcome to TensorFlow! By Chip Huyen