Image Classification in Python

Image Classification has become one of the trending topics in today's world. It is most required features in today's smartphone and many other applications. Image Classification is found in most popular areas such as Google Assistant, Microsoft Bing, Google Lens and so on.

Image Classification can be applied in many ways using Deep learning such as Pytorch, Theana, Keras, TensorFlow and so on.

TensorFlow is one of the Google's Open Source Deep Learning Library which uses two ways towards programming i.e., the Tensors and the approach of Dataflow programming.

Tensors are Multi-Dimensional arrays which are used in computation. These arrays generally are multi-dimensional matrices with rows and columns that makes the computation of images very efficient. Tensors are broader concepts of Vectors. Vectors are generally represented as column matrices with 2 or 3 dimensions. Vectors are quantities which show scalar quantities along with directions. Vectors are generally 2D or 3D. It is represented along the directions of X, Y,Z,XY,YZ,ZX,XYZ directions. But if we consider a 3D surface with Area Vector in X or Y or Z directions, the Area is generally 2D.the force can act in any of the 3 directions such as X,Y,Z and so on , on other directions. when such forces need to be represented, we represent is as Axx, Ayy and so on. Then when all these forces are represented, we will have 3X3 matrix of rank2.So it will be a 3X3 tensor.

Similarly, an image of 784 pixels are represented by 28X28 tensor and can be used as input.

The TensorFlow uses Dataflow method of programming. The Dataflow method has the formation of graphs where the nodes that are formed are operations and the edges represent the tensors on which the operation is performed, or variables required mostly in computation, respectively. The Operations that is needed to be carried out in the programming sequence are generally represented as computational graphs.

The ML model that is to be built is generally converted to TensorFlow computational graph and then fed to tensor flow runtime for execution.

Image Classification works on Deep Learning.

Deep learning is a subset of Machine Learning. Deep learning is a mimic of human Neural network called the perceptron. Deep learning 's simple example can be Multi-Layer Perceptron network where there is formation 3 major layers which are the Input Layer, Output Layer and the hidden layer. The Input layer is the layer to which the input is given during the training or using mode, and then the output layer is the layer the gives out the outputs in terms of probabilities and finally the output neuron which is predicted to have highest probability is said to be the actual output. The hidden layer is the layer where all the preprocessing happens.it starts receiving the input values from the input layer. At the input layer, each input is assigned a weight. Weight represents the amount of weightage that is generally given to a particular input,

And the bias represents how much your weight differs from the threshold weighted sum so that you can decide if activation of a neuron through activation function is possible or not.

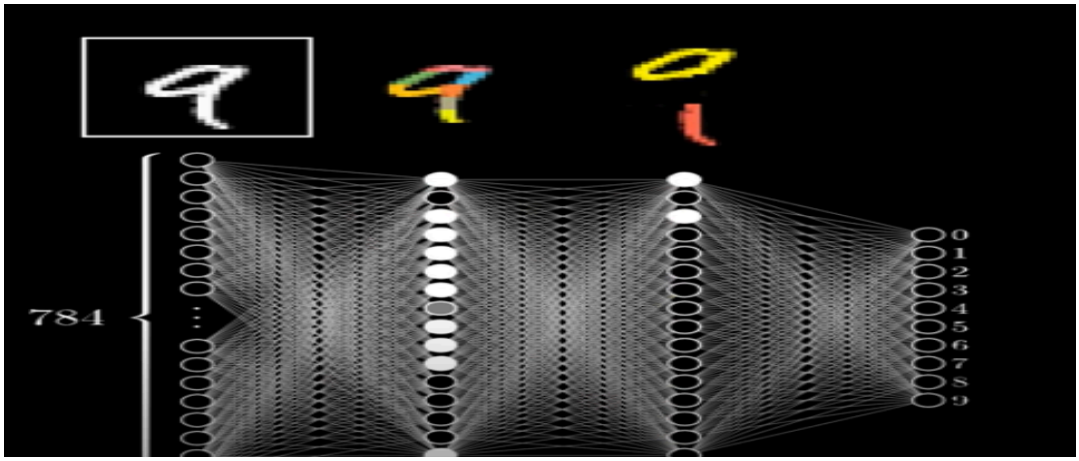
We can Consider each input neuron to be connected to all the other neurons in the second layer through edges or string like structure. These are assigned random weights and biases on which further output is calculated. Depending upon how much error your network makes these weights and biases are recalculated and updated until the output meets the actual output properly, that is the cost has been reduced which can be found via gradient descent algorithm.

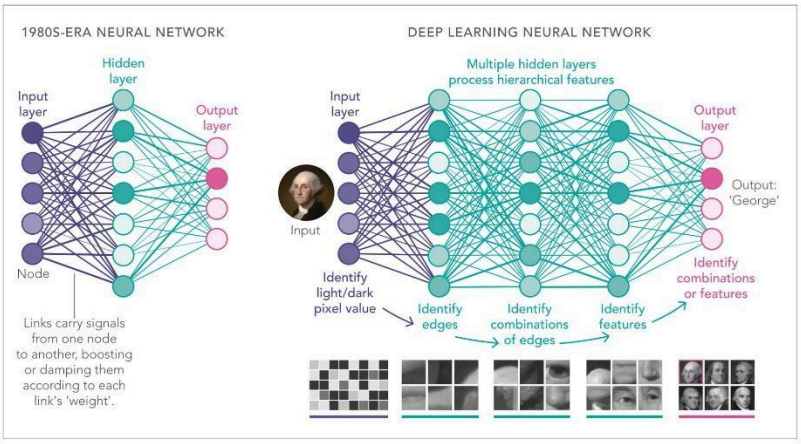
The Neural network is formed just like the neurons getting connected in the brain. When brain receive senses from sensory neurons they are take to the brain and based on the training given to it and analysis the action is signaled by the brain through motor neurons.

Here the 28X28 grid of tensors actually will hold the values of color or grayscale amounts in the image supplied to it. Each pixel will be given a value according to the color or the intensity of grayscale from a count of 0 to 1.

Next these inputs according to their weights are multiplied along with adding to the bias gives a value which will be supplied to an activation function according to which the particular neurons get fired, identifying some patterns in the network.

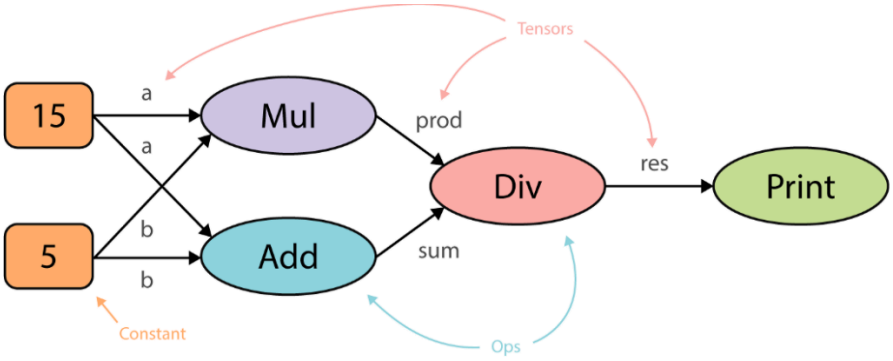
Next these possible patterns found are passed to the next layer where the possible big patterns are found and they fire up the neuron. Afterwards search for an even more big and complex pattern is made and finally the output is predicted. Typically, all input nodes are connected to all nodes in the hidden layer.





"Neural network" models of AI process signals by sending them through a network of nodes analogous to neurons. Signals pass from node to node along links, analogs of the synaptic junctions between neurons. "Learning" improves the outcome by adjusting the weights that amplify or damp the signals each link carries. Nodes are typically arranged in a series of layers that are roughly analogous to different processing centers in the cortex. Today's computers can handle "deep-learning" networks with dozens of layers. Image credit: Lucy Reading-Ikkanda (artist).

Tensor flow computes a graphs **that** helps to distribute the **computational** load if one must deal with complex models containing a large number of nodes and layers. Finally, a **neural network** can be compared to a composite function where each **network** layer can be represented as a function.



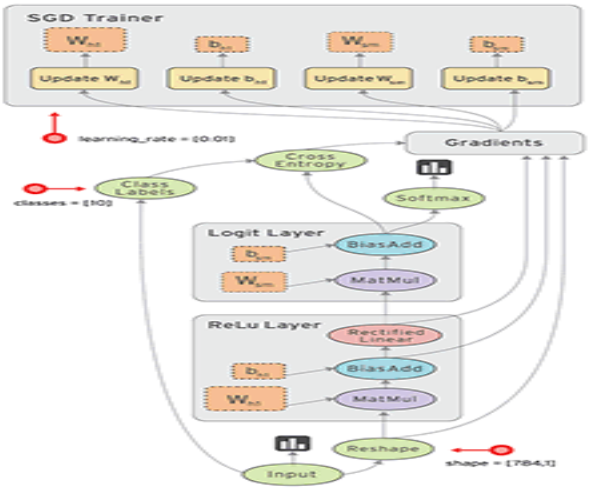
Computational graph representing our simple program and its data flow

The above shows a simple computational graph in tensor flow on a simple operation

A=15

B=5

Print((a*b)/(a+b));



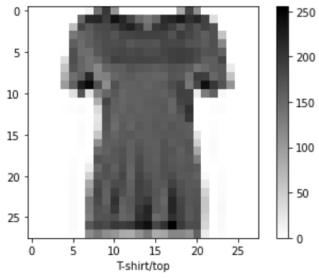
The above is the example of the TensorFlow graph on the Convolutional neural network, shows the computational graph of a three layer neural network.

Here it can be seen how computational graphs can be used to represent the calculations in neural networks, and this, of course, is what TensorFlow excels at this in a very efficient way.

Hence TensorFlow Supports multiple operations that can happen in parallel on the CPU or GPU respectively.

The Fashion MNIST dataset:

The Fashion MNIST dataset is a image classification dataset that is inbuilt and can be imported through Keras or TensorFlow. It is a dataset which has 60,000 training and 10,000 testing images of grayscale type which has 28X28 pixeled images each of different types of dresses such as ankleboots,T-shirt/Top,Shoes,Sandals, Sneaker and so on. These images are loaded to notebook.



This is how the images are in the dataset when plotted using matplotlib.

Keras is an API library which can learn individually or on libraries such as tensor flow as well.

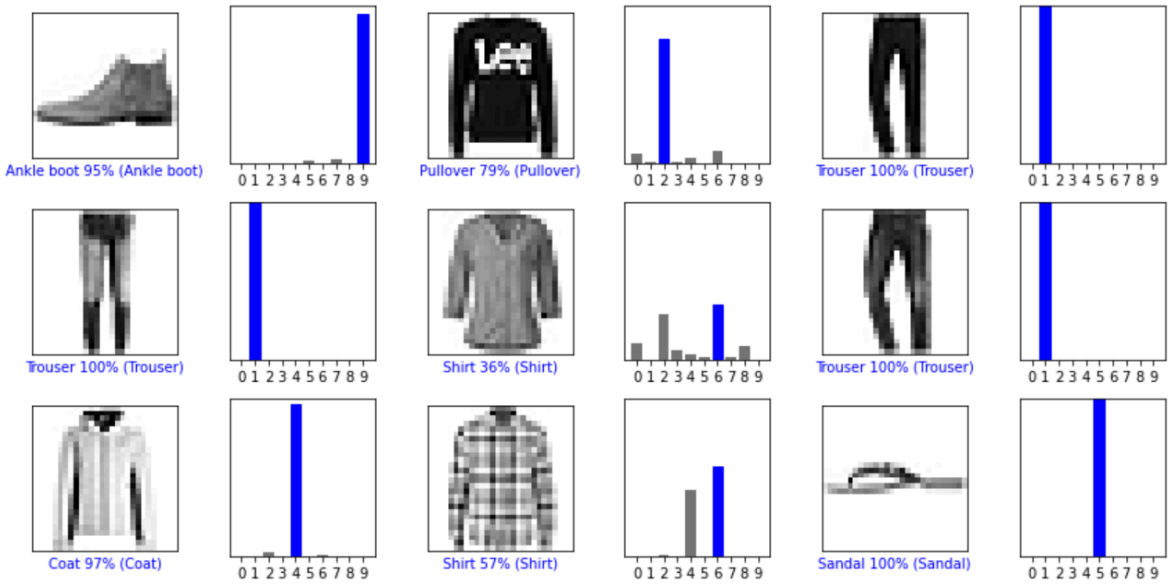
The API is used here to build the model and the respective neural network layers ,

Where there are 28,28 layered input that is initially flattened which is the input layer, later there are 128 dense neuron layer that is formed. Then the there is a output layer built to get the output of 10 perceptron respectively.

The model is fit with the training data and trained for 15 epochs.

Then the model is tested with the test data. And the following outputs were observed.

You can see in the below picture how the probabilities are being predicted for each test image:



An Accuracy of 82% is obtained for this model.

Code :
https://colab.research.google.com/drive/14iuNxuD7LvUnxO0C0vBtR61YQ50hWvS9?usp=sharing