What is tensorflow

It is a machine learning framework used to build models, like NLP natural language processing , computer vision , face recognition.

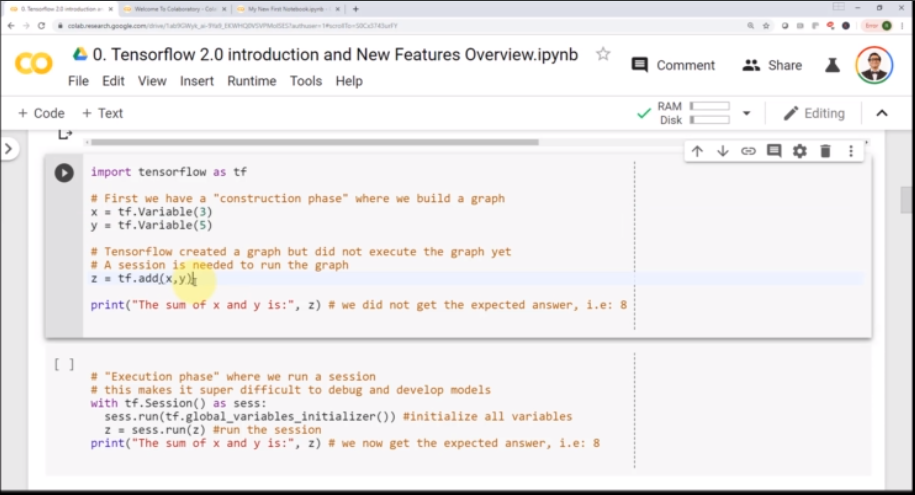
It is developed by google in 2015 by google brain team

GPU and TPU

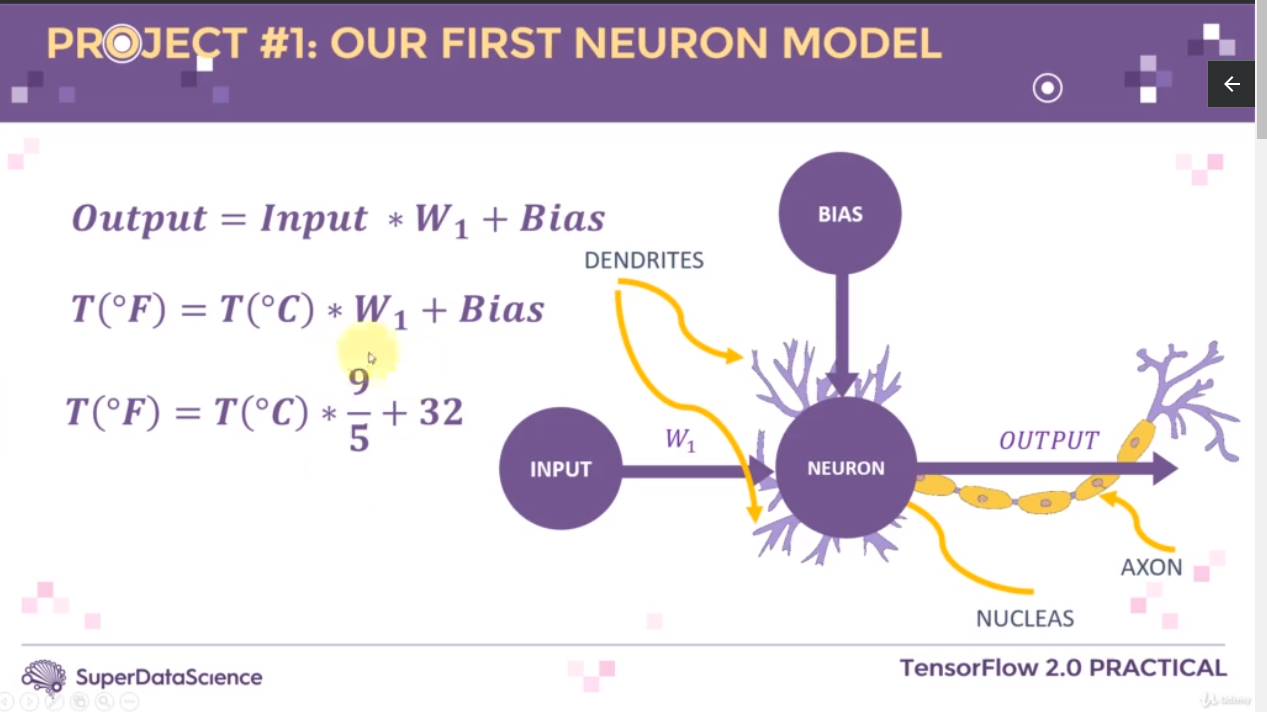
Graphics Processing unit – used in CAD drawing, video editing.

TPU –

Eager Execution – we can execute code line by line , in Tensor flow 1 we had to initiate a session to print or analyse the variables but from tensor flow 2 onwards we do not need it.



Keras – default API to interact with tensorflow



Regression

Regression works by predicting value of one variable Y based on another variable X , where X is called Independent variable and Y is called dependent variable.

Graph 1 when value of x increases the value of y is increases and graph 2 when y decreases if the value of x increases

graph 1 Y

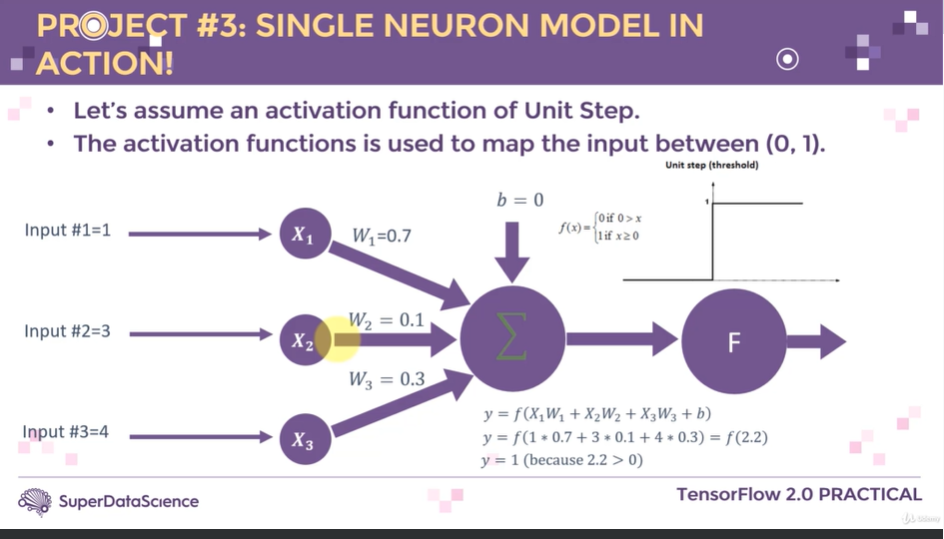
Y

X X Graph 2

Activation function

Activation function takes in inputs \* weigths + bias and makes a decision

For ex : the activation function when finds a positive value it sets the output as 1 and if it is a negative number it sets the value as 0 . this way the model is capable of taking decisions , like the object is a cat or not . i.e if we have range of values and when we want to to default them to some values we need activation function and this will help system to come to a decision



Activation function screen shot above.

Activation function overview

1.Sigmoid, 0,1 – output layer - for classifying data we can use this function

2.Relu – rectified linear units – in hidden layers. – for regression like predicting sales or linear regression problems we can use this function

3. Hyperbolic Tangent Activation - -1 to 1 - in out put layers

Multilayer perceptron network

Network which has hidden layers which connect input to output.

These are called dense network. Or deep learning.

Epochs

Updating the weights is known as epochs because there is an error in the predicted out to the desired output.

When we increase the epochs we minimize the errors and try to match the desired output.

// Create the Model

Model = tf.keras.models.Sequential() // initiate the model

Model.add(tf.keras.layers.Dense(units=100,activation=’relu’,input\_shape=(35, ))) // tell the model number of inputs along with activation function

Model.add(tf.keras.layers.Dense(units=100,activation=’relu’)) // hidden layer 1 or deep network , we don’t have to define input here as it is hidden

Model.add(tf.keras.layers.Dense(units=100,activation=’relu’)) // hidden layer 2 or deep network , we don’t have to define input here as it is hidden

Model.add(tf.keras.layers.Dense(units=1,activation=’linear’) // as we need only one output we have defined the unit as 1 and linear as output which means it will give linear co relation with the data.

// check the summary of the model

Model.summary() // will give you no of parameters to be trained including weights and biases.

/// compile the model.

Model.compile(optimizer=’Adam’ , loss=’mean\_squared\_error’)

// Train the model

Epochs\_hist = model.fit(X\_train,y\_train,epochs=20, batch\_size=50) // model.fit will train the data

**So mainly there are four steps that we need to perform**

Model = tf.keras.model.sequentials() // To create the model object

Model.add() // to create the model with the networks

Model.Summary() // optional but you can check the summary of the model

Model.compile() // compile the model , like compiling the code.

Model.fit() // train the model.

\*\*\* There are tools which can identify the correct and optimized models for us --- Need to do research on this.