REPORT

What is TensorFlow?

TensorFlow is an open-source machine learning library for research and production. It is used to create models, train them as well as test the same using different datasets. It uses the key concepts of neural networks and deep learning to solve different problems.

GIVEN

*Try out the tutorial for Deep Learning using **TensorFlow** at https://www. tensorflow.org/tutorials. It has the following lines of code.*

```
1. import tensorflow as tf
2. mnist = tf.keras.datasets.mnist
3.
4. (x_train, y_train),(x_test, y_test) = mnist.load_data()
5. x_train, x_test = x_train / 255.0, x_test / 255.0
6.
7. model = tf.keras.models.Sequential([
8.   tf.keras.layers.Flatten(),
9.   tf.keras.layers.Dense(512, activation=tf.nn.relu),
10.  tf.keras.layers.Dropout(0.2),
11.  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
12.])
13.
14.model.compile(
15.  optimizer='adam',
16.  loss='sparse_categorical_crossentropy',
17.  metrics=['accuracy']
18.)
19.
20.model.fit(x_train, y_train, epochs=5)
21.model.evaluate(x_test, y_test)
```

PROBLEM 1

In the report, write comments for each line of code given above and explain what this framework is doing.

Line#	Comments
1	*Importing* tensorflow library to use in the program and aliasing the same to *tf*
2	Importing *mnist* data which is hand written digits dataset of 28 x 28 pixels
4	*Loading* the mnist training data and testing data of size 6000 and 1000 respectively. Note: x_train and x_test represents the matrix of 28*28, while y_train and y_test is the output.
5	*Normalizing* the dataset with values between 0 to 1

7	***tf.keras.models.Sequential*** creates a deep learning model with linear stack of layers.
8	***tf.keras.layers.Flatten()*** Flattens the input. Here the input is a matrix of 28 x 28. Say x = [[A0_0, A0_1, … A0_27] [A1_0, A1_1, … A1_27] … [A27_0, A27_1, … A27_27]], Then flatten would convert the same to x = [A0_0, A0_1, …., A0_27, A1_0, … A27_27]
9	***tf.keras.layers.Dense(512, activation = tf.nn.relu)*** This creates a layer in the network which has 512 nodes and all with the activation function of 'Rectified Linear Unit'.
10	***tf.keras.layers.Dropout(0.2)*** This helps to evaluate the network better by droping out 20% of nodes for each iterations. This helps to deviate the model from over fitting.
11	***tf.keras.layers.Dense(10, activation = tf.nn.softmax)*** This creates a layer in the network which has 10 nodes and all with the activation function of 'Softmax'. Since this is the last layer, we can say the same as output layer.
14	***model.compile()*** helps us to add configurations to the model. They include learning rates, error rate methods, different optimizer types and many more.
15	Model to use ***Adam*** Optimizer Algorithm
16	Model to use Sparse Categorial Cross Entropy to calculate the ***loss***
17	***Metrics*** to be evaluated by the model during training and testing. Here, accuracy
20	***model.fit(x_train, y_train, epochs=5)*** helps to train the model with input, required output. Here epochs is number of batch iterations.
21	***Model.evaluate(x_test, y_test)*** helps to test the model with input and required output.

Finally it generates a following neural network

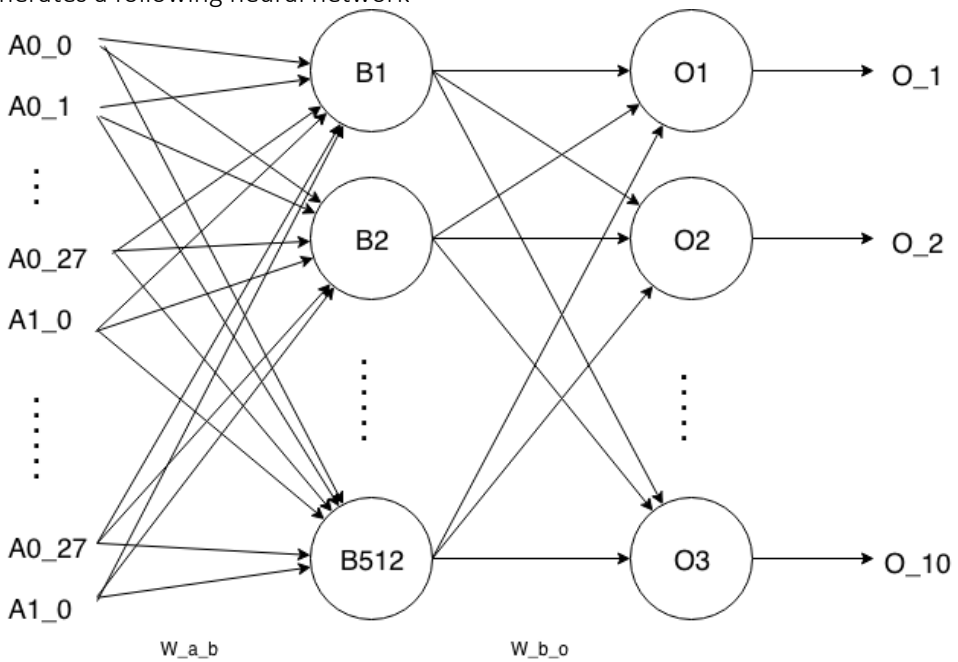

Figure 1 Deep Network for Problem 1

PROBLEM 2

Change the number of hidden nodes to 10 and train this neural network. The trained model contains the weights that it has learned from training. Plot the features in the hidden layer that it has learned from training and include them in the report. That is, reshape the learned weights (vectors) in the first layer (between the input and the 1st hidden layer) to the image dimension (in 2D) and show them.

Changing Line#9 to: `tf.keras.layers.Dense(10, activation=tf.nn.relu),`

Also adding the following to read the weights:

```
22. a = model.get_weights()[0]
23. for i in range(0,10):
24.   b = tf.reshape(a[:,i], [28,28])
25.   print
26.   print "Image " + str(i+1) + ":"
27.   print tf.Session().run(b)
```

The output is stored in a filed named as ***problem2_output.txt*** file along with the attachment.

Tabulated Values:

Term	Representation	Size
a[0]	Weights from layer 'A' to 'B'	784 x 10
a[1]	Weights from bias to 'B'	10 x 1
a[2]	Weights from layer 'B' to 'O'	10 x 10
a[3]	Weights from bias to 'O'	10 x 1

PROBLEM 3

Change the number of hidden nodes to 1, 10, 50 and 100 and report how the testing accuracy changes for the testing dataset. Report the result and your observation in the report.

Testing DataSet size = 10000

Number of Hidden Nodes	*Accuracy*	*Loss*
1	0.3208	1.6599035221099854
10	0.9354	0.22685219403356313
50	0.9713	0.09961129692737013
100	0.9751	0.07967793111021165
512	0.9805	0.06960659905002103

From the above table:

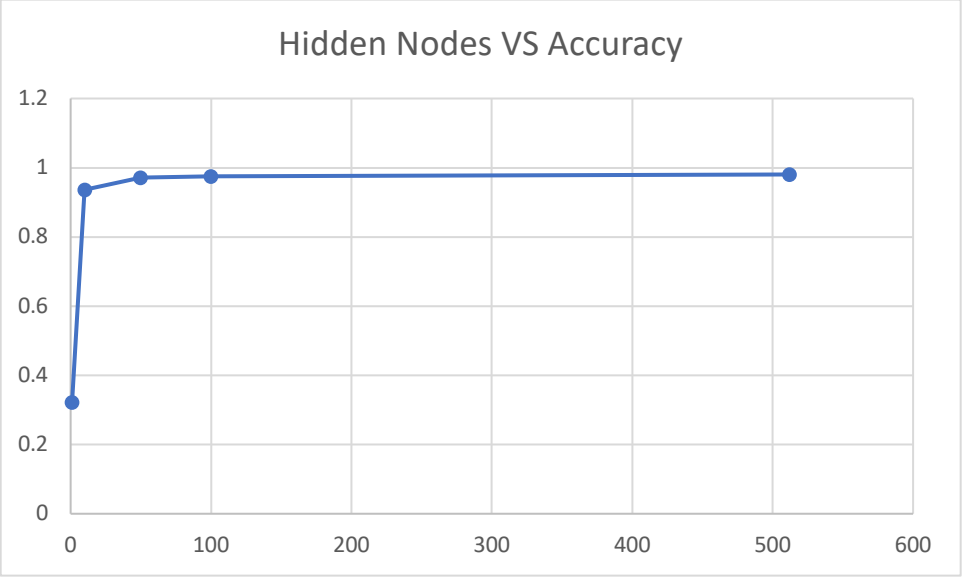

Figure 2 Hidden Nodes VS Accuracy

Observations:

- As we increase the number of hidden nodes, accuracy increases.
- The time taken for training is directly proportional to number of hidden nodes.

The output is stored in a filed named as ***problem3_output.txt*** file along with the attachment.

Note:
The program used is stored in tf.py. This program was used and developed from https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/_index.ipynb

References:

- https://www.tensorflow.org/
- https://stackoverflow.com