Programming Assignment - 1

Karan Goel
8116-0185
karangoel16@ufl.edu

Links:

1. GitHub (https://github.com/karangoel16/MNIST)
2. GoogleDrive(https://docs.google.com/document/d/1pfG0hDpkjOUw4ZNgvBdQpBUJwrYw7nBCqhH_x_NbX8w/edit?usp=sharing)

We implemented two different model to get , the result.

1. KNN MODEL

In first model implemented , we used KNN model to classify the numbers of the MNIST test and was able to get accuracy of 96% on it.

Code of this is in the GITHUB folder with the filename **Project_assignment-1.py**.

2. CNN MODEL

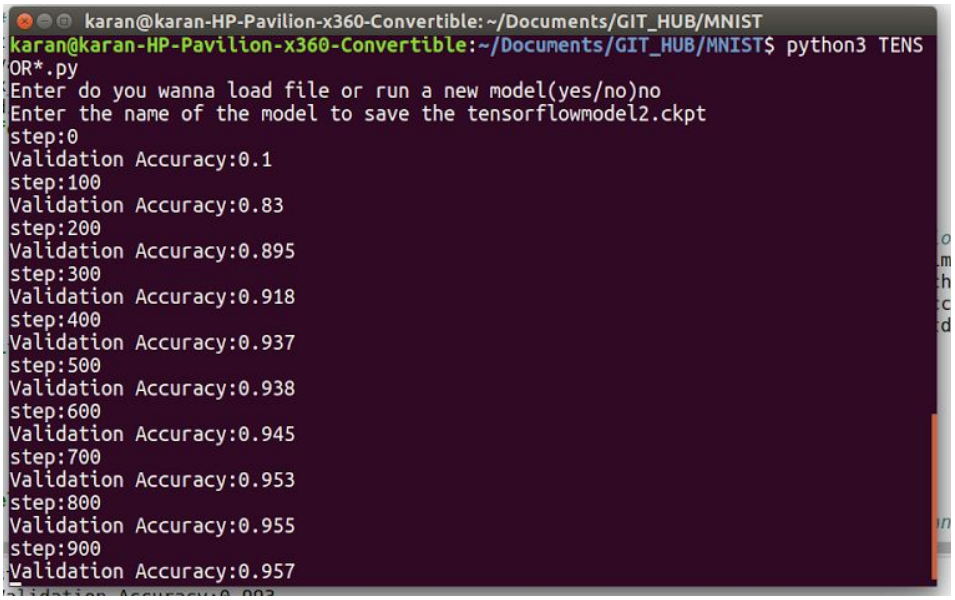
We can load the trained model and get the result , or if want to train our own model and save it for later usage , we can run the program , Tensorflow_mnist.py , and we can change iteration value , but we have tried to use the same model multiple times.

Also to take advantage of the scholastic batch training method , we have created a method and then added another layer of gaussian error at the very input that would not increase our cases and help us develop more number of training data sets, by augmenting the data.

Code to train is in the zip folder with the filename **TENSORFLOW_MNIST.py**.

Code to test on the test set on the model saved by training on the data is in the zip folder with the filename **LOAD_MODEL.py**.

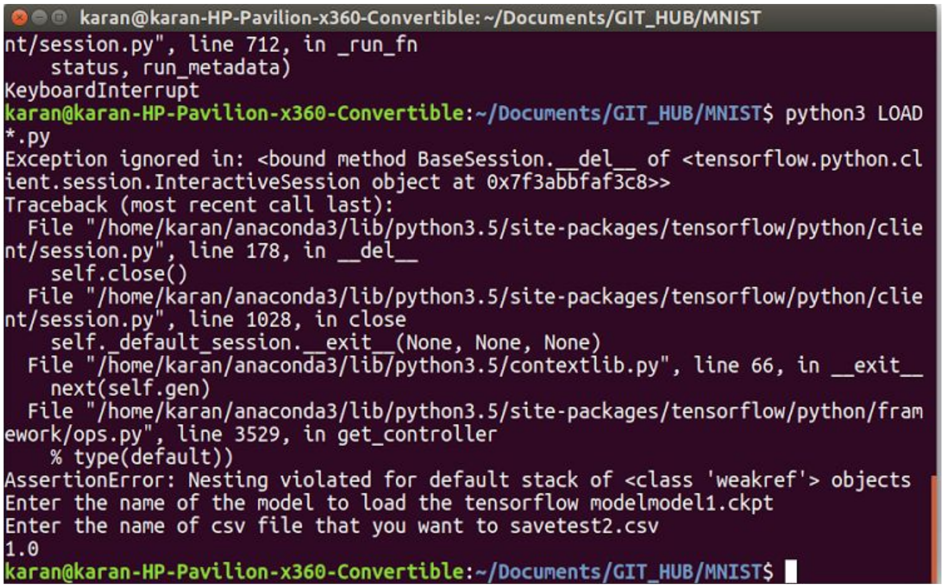
ScreenShot:



karan@karan-HP-Pavilion-x360-Convertible: ~/Documents/GIT_HUB/MNIST

```
karan@karan-HP-Pavilion-x360-Convertible:~/Documents/GIT_HUB/MNIST$ python3 TENS
OR*.py
Enter do you wanna load file or run a new model(yes/no)no
Enter the name of the model to save the tensorflowmodel2.ckpt
step:0
Validation Accuracy:0.1
step:100
Validation Accuracy:0.83
step:200
Validation Accuracy:0.895
step:300
Validation Accuracy:0.918
step:400
Validation Accuracy:0.937
step:500
Validation Accuracy:0.938
step:600
Validation Accuracy:0.945
step:700
Validation Accuracy:0.953
step:800
Validation Accuracy:0.955
step:900
Validation Accuracy:0.957
```

While Generating/Training the model

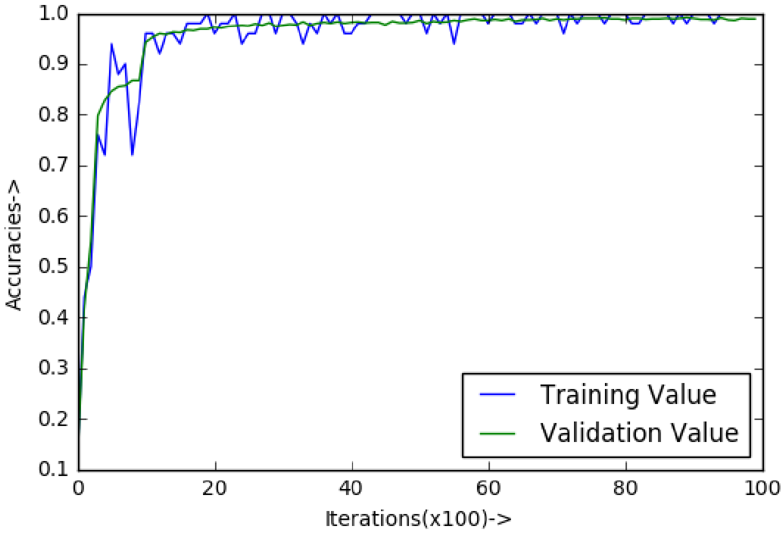


karan@karan-HP-Pavilion-x360-Convertible: ~/Documents/GIT_HUB/MNIST

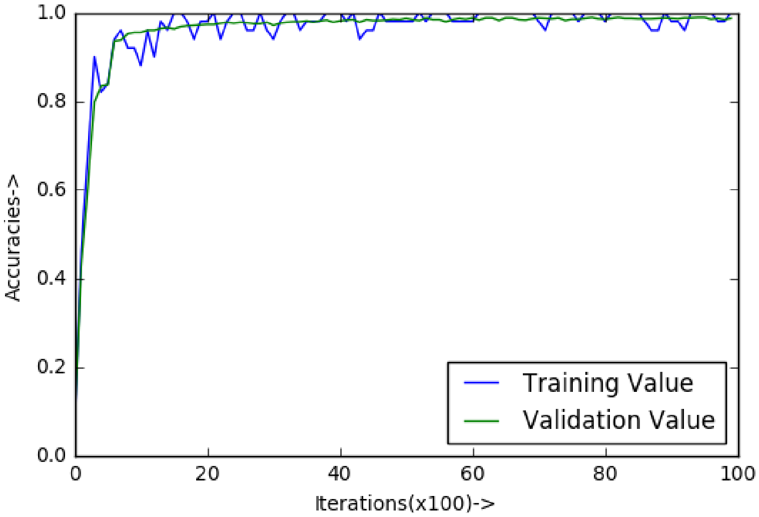
```
nt/session.py", line 712, in _run_fn
    status, run_metadata)
KeyboardInterrupt
karan@karan-HP-Pavilion-x360-Convertible:~/Documents/GIT_HUB/MNIST$ python3 LOAD
*.py
Exception ignored in: <bound method BaseSession.__del__ of <tensorflow.python.cl
ient.session.InteractiveSession object at 0x7f3abbfaf3c8>>
Traceback (most recent call last):
  File "/home/karan/anaconda3/lib/python3.5/site-packages/tensorflow/python/clie
nt/session.py", line 178, in __del__
    self.close()
  File "/home/karan/anaconda3/lib/python3.5/site-packages/tensorflow/python/clie
nt/session.py", line 1028, in close
    self._default_session.__exit__(None, None, None)
  File "/home/karan/anaconda3/lib/python3.5/contextlib.py", line 66, in __exit__
    next(self.gen)
  File "/home/karan/anaconda3/lib/python3.5/site-packages/tensorflow/python/fram
ework/ops.py", line 3529, in get_controller
    % type(default))
AssertionError: Nesting violated for default stack of <class 'weakref'> objects
Enter the name of the model to load the tensorflow modelmodel1.ckpt
Enter the name of csv file that you want to savetest2.csv
1.0
karan@karan-HP-Pavilion-x360-Convertible:~/Documents/GIT_HUB/MNIST$ 
```

While Retrieving the output

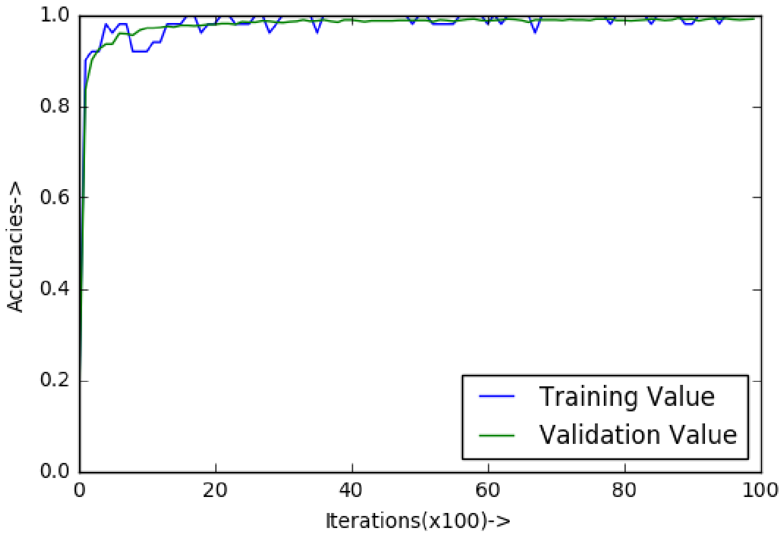
Figures:



(a)Graph for Accuracy vs number of iterations (2 layer- CNN model [5x5 , 5x5], 2 fully connected)



(b) Graph for Accuracy vs number of Iterations (3- layers -CNN , 2 fully connected layer).
Additional 1x1 filter CNN layer, more than the layer above.



After doing data augmentation , we got better result from our CNN model.

Training accuracies and Validation accuracies increasing with the number of iterations and then later we have saved the model , so the we can train more on this model and use it to predict the test cases as well. “Model1.ckpt” is the name of the file that has been used to save the trained model and then the trained model was used to predict the value of the test cases.

The final accuracy of 99.06%, came from the last described model and this model has been used to finally get judged in the kaggle competition.

Dropout: It is a regularization technique used to solve the problem of overfitting , this keeps a certain parameter with probability (t) and removed it from the probability of (1-t) , it can be applied to both hidden and visible layer in the network.

The function in tensorflow for this is : tf.nn.Dropout();
Reference for the following is https://www.tensorflow.org/api_docs/python/tf/nn/dropout.

Libraries and other important things needed:

Adding another layer of gaussian error and saving the model to check the model training and validation is few steps , we tried to make the model better and give better result.Though we still believe much few more type of data augmentation could have made the result bit better.

References:

1. Tensor Flow(https://www.tensorflow.org/)
2. Kaggle(https://www.kaggle.com/innerproduct/state-farm-distracted-driver-detection/tensorflow)