<u>CSE – 575 STATISTICAL MACHINE LEARNING</u>

<u>PROJECT 3</u> : <u>CLASSIFICATION USING NEURAL NETWORKS AND DEEP LEARNING</u>

<u>Analysis:</u>

In this project, we implement the classification of digits using simple convolutional neural network. The code provided in coursera lab, described the definition of various layers and functions including Convolutional Layer, Fully-Connected Layer, Pooling Layer, Activation Layer, Loss function. The dataset given is MNIST dataset and the training and testing samples are restricted to 2000 samples. Our task is to complete the evaluate function for the given set of code, (initial network) to calculate the loss and accuracy of the predicted samples. Another part from the project is to run the baseline code for which we have the test accuracy 0.983. But when we change the kernel size, the test loss is reduced slightly. When we edit the baseline code for number of feature maps, the test accuracy reduces to 0.02 percent.

<u>Results:</u>

<u>Final Training Accuracy:</u> 0.850

<u>Final Training Loss:</u> 0.420

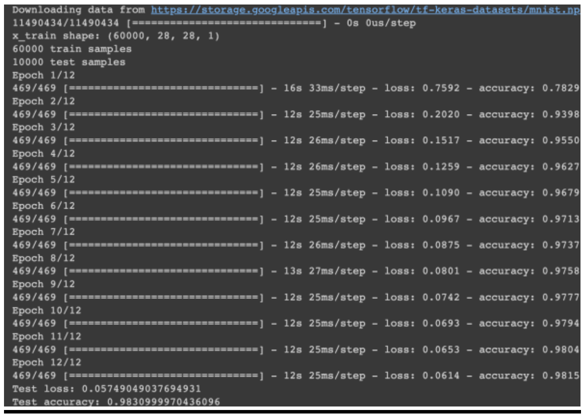
<u>Final Testing Accuracy:</u> 0.807

<u>Final Testing loss:</u> 0.530

<u>Evaluate Function:</u>

```
def evaluate(net, images, labels):
    acc = 0
    loss = 0
    batch_size = 1
    pass
    for batch_index in range(0,
images.shape[0], batch_size):
# Extract the data and label, and as we have
batch size as 1, extraction is not looped.
        data=images[batch_index]
        label=labels[batch_index]
       # Forward pass
        for l in range(net.lay_num):
            output =
net.layers[l].forward(data)
            data = output
       # Calculate the loss based on the given
cross entropy function.
        loss += cross_entropy(output, label)
      # Calculate accuracy
        if np.argmax(output) ==
np.argmax(label):
            acc += 1
    # Calculate the average loss and accuracy
    loss /= (float(images.shape[0]) /
float(batch_size))
    acc /= (float(images.shape[0]) /
float(batch_size))
    return acc, loss
```

<u>Task 2 (Kernel size 3*3):</u>

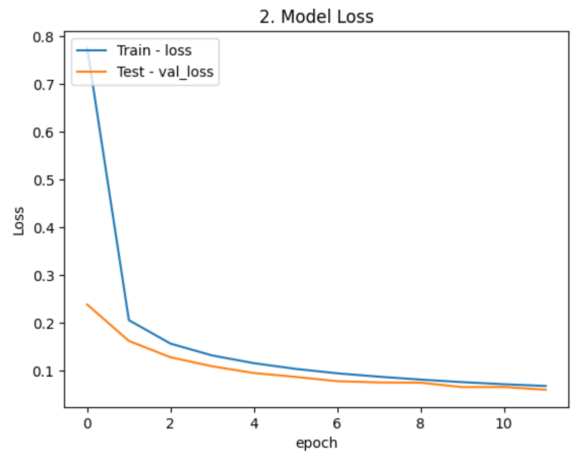


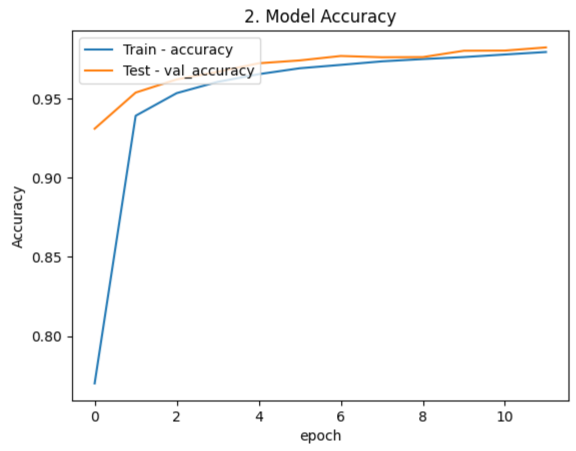
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.np
11490434/11490434 [==============================] - 0s 0us/step
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Epoch 1/12
469/469 [==============================] - 16s 33ms/step - loss: 0.7592 - accuracy: 0.7829
Epoch 2/12
469/469 [==============================] - 12s 25ms/step - loss: 0.2020 - accuracy: 0.9398
Epoch 3/12
469/469 [==============================] - 12s 26ms/step - loss: 0.1517 - accuracy: 0.9550
Epoch 4/12
469/469 [==============================] - 12s 26ms/step - loss: 0.1259 - accuracy: 0.9627
Epoch 5/12
469/469 [==============================] - 12s 25ms/step - loss: 0.1090 - accuracy: 0.9679
Epoch 6/12
469/469 [==============================] - 12s 25ms/step - loss: 0.0967 - accuracy: 0.9713
Epoch 7/12
469/469 [==============================] - 12s 26ms/step - loss: 0.0875 - accuracy: 0.9737
Epoch 8/12
469/469 [==============================] - 13s 27ms/step - loss: 0.0801 - accuracy: 0.9758
Epoch 9/12
469/469 [==============================] - 12s 25ms/step - loss: 0.0742 - accuracy: 0.9777
Epoch 10/12
469/469 [==============================] - 12s 25ms/step - loss: 0.0693 - accuracy: 0.9794
Epoch 11/12
469/469 [==============================] - 12s 25ms/step - loss: 0.0653 - accuracy: 0.9804
Epoch 12/12
469/469 [==============================] - 12s 25ms/step - loss: 0.0614 - accuracy: 0.9815
Test loss: 0.05749049037694931
Test accuracy: 0.9830999970436096
```

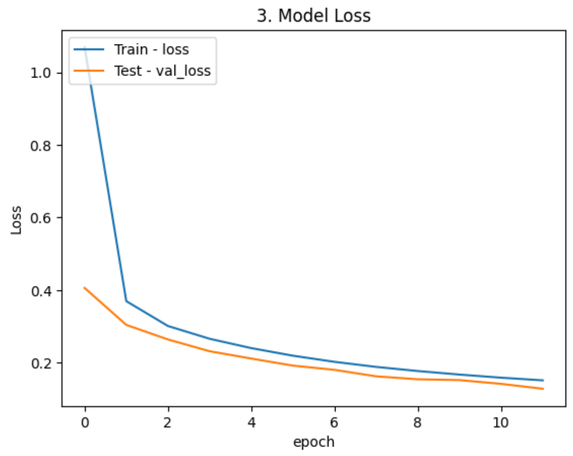
Task 3 (Changing kernel size):

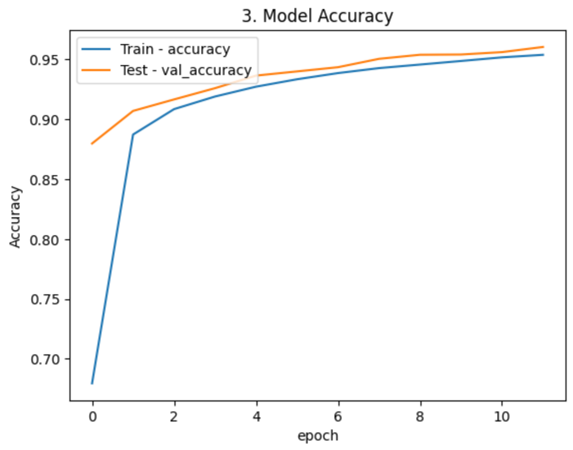
```
Test loss: 0.05997486412525177
```

```
Test accuracy: 0.982200026512146
```









Task 4 (Changing number of feature map):

```
Test loss: 0.12820255756378174
```

```
Test accuracy: 0.9603999853134155
```