In-Lab

Task 1

Importing Libraries:

```
from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D, Flatten, Dense
from keras.datasets import mnist
from keras.utils import to_categorical
```

Task 2

```
(train_images, train_labels), (test_images, test_labels) =
mnist.load_data()

train_images =
train_images.reshape((60000,28,28,1)).astype('float32')/255
test_images =
test_images.reshape((10000,28,28,1)).astype('float32')/255
```

Task 3

```
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

Task 4

```
model = Sequential()
model.add(Conv2D(32,(3,3),activation = 'relu',input_shape=(28,28,1)))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
```

```
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

Task 5

```
model.compile(optimizer =
'adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

Task 6

```
model.fit(train_images,train_labels,epochs=5,batch_size=64,validation_data=(test_i
```

```
Epoch 1/5
938/938 [==============================] - 70s 73ms/step - loss: 0.1797 - accuracy: 0.9460 - val_loss: 0.0672 - val_accuracy: 0.9789
Epoch 2/5
938/938 [==============================] - 49s 52ms/step - loss: 0.0573 - accuracy: 0.9828 - val_loss: 0.0447 - val_accuracy: 0.9863
Epoch 3/5
938/938 [==============================] - 51s 55ms/step - loss: 0.0404 - accuracy: 0.9873 - val_loss: 0.0375 - val_accuracy: 0.9869
Epoch 4/5
938/938 [==============================] - 50s 54ms/step - loss: 0.0302 - accuracy: 0.9908 - val_loss: 0.0302 - val_accuracy: 0.9906
Epoch 5/5
938/938 [==============================] - 49s 52ms/step - loss: 0.0239 - accuracy: 0.9924 - val_loss: 0.0368 - val_accuracy: 0.9885
<keras.src.callbacks.History at 0x7a2280ea83a0>
```

Figure 1

Task 7

```
test_loss,test_acc = model.evaluate(test_images,test_labels)
print(f'Test Accuracy : {test_acc}')
```

```
313/313 [==============================] - 2s 8ms/step - loss: 0.0368 - accuracy: 0.9885
Test Accuracy : 0.9884999990463257
```

Figure 2

What is an neural network:

A neural network is a computer system that learns patterns from data. It mimics the brain by connecting nodes (neurons) together. These nodes work together in layers, processing information, and adjusting connections (weights) to make predictions or decisions based on

input data. It's great for tasks like recognizing images, understanding language, and making predictions from complex data.

Layers Used:

1. Input Layer
2. Convolution Layer
    a. First Convolution layer
    b. Max Pooling Layer (after the first CON2D)
    c. Second Convolution Layer
    d. Max Pooling Layer (after the second CON2D)
3. Flatten Layer
4. Dense Layers
    a. First Dense Layer
    b. Output Layer

Working of layers used:

- CON2D:

Working: Conducts convolution (feature extraction) on the input image using small grids (kernels/filters) to detect patterns.
Activation: Applies an activation function (ReLU in this case) to introduce non-linearity

- MaxPooling2D layer:

Working: Reduces the spatial dimensions of the previous layer's output by taking the maximum value within each region of the grid.
Purpose: Helps in reducing computational complexity and retains important features.

- Flatten Layer:

Working: Converts the 2D output of the convolutional layers into a 1D array.
Purpose: Prepares the flattened output to be fed into the dense (fully connected) layers.

- Dense Layer:

Working: Every neuron in one layer is connected to every neuron in the next layer.
Activation: Applies an activation function (ReLU for hidden layers, softmax for output layer) to the weighted inputs.
Purpose: Learns complex patterns in the data through these interconnected layers, ultimately providing output predictions.

Weights:

In the convolution layers the network learns the weight that are convolved with the input image to produce feature maps.

In the dense layer each neuron has its weight connecting to the previous layer.