ASSIGNMENT 2

MUNTASHIR BIN SOLAIMAN

Student ID - 22021468

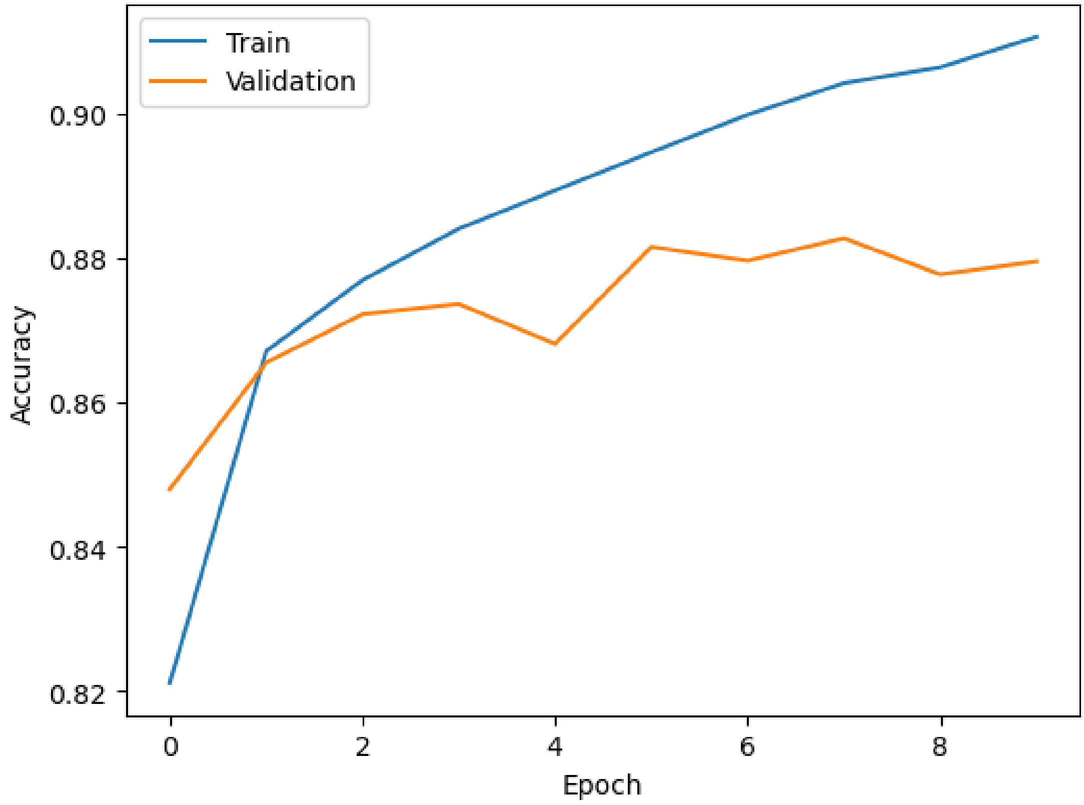
Task 1

The goal was to classify the images. A neural network was used without convolutional layers. The preprocessing steps were splitting the data, reshaping the data, normalising the data and one hot encoding labels. The data was split into testing and training set. This was done to prevent overfitting and get performance metrics later. The data was reshaped to meet the requirements of the network. It also helped the model learn the spatial relationship between each pixel. Normalisation was done so that the model can learn more effectively and efficiently. The classes were one-hot encoded. This was done so that the model does not associate order in the data by mistake. It was necessary for SoftMax function and categorical cross-entropy function. There were two neural networks. The first model had an input layer, two dense layers and an output layer. The input layer flattened the shape into a 1D array. The first dense layer had 128 neurons and Rectified Linear Unit (ReLU) activation function. This layer understood basic patterns of the images. ReLu made the model learn complex patterns through non-linearity. 0 was the output if the weighted sum was negative. The second dense layer had 64 neurons. The lower number prevented overfitting. Higher level patterns were learned. The output layer had 10 neurons. They were for each class. SoftMax function was used. The outputs were a probability. The second neural network had 4 dense layers. The neuron numbers were 256, 128, 64, 32 respectively in each layer. ReLu, SoftMax, Adam and categorical cross entropy was used. This was like the first one. 64 Images were processed at a time. The models were trained on 938 images. There were 10 epochs. The results were: -

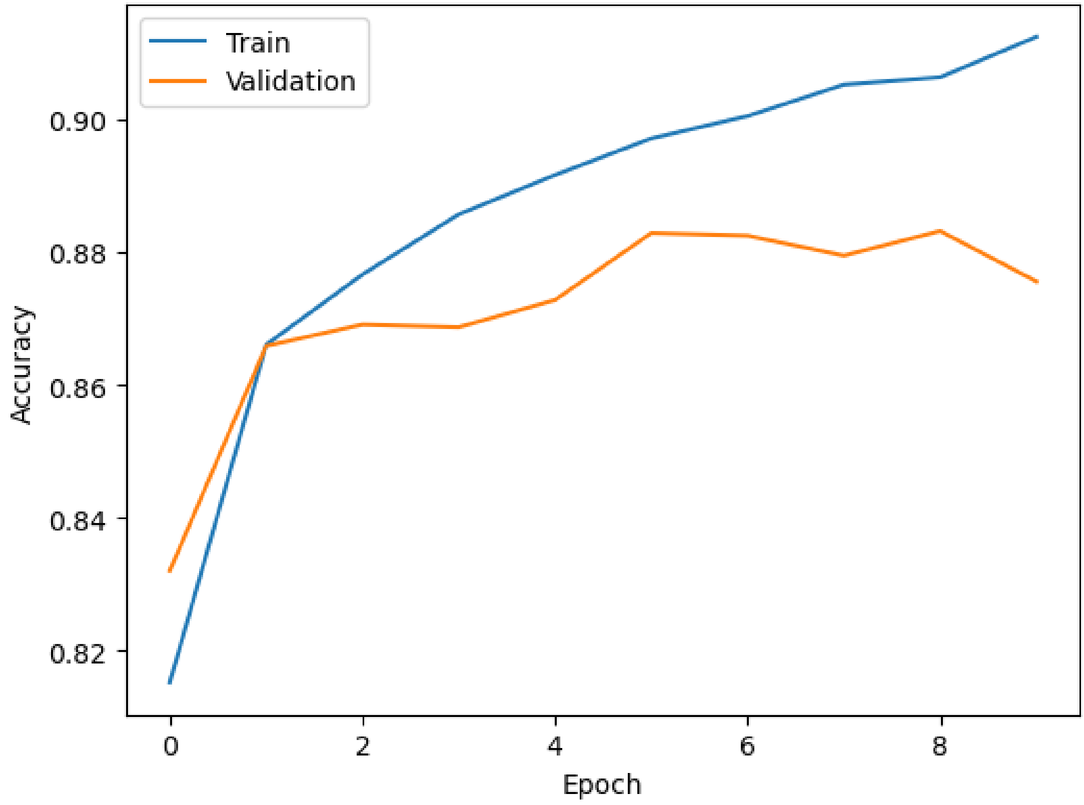
	First neural network	Second neural network
Loss in first epoch	0.5044	0.5179
Loss in last epoch	0.3357	0.3693
Accuracy score in first epoch	0.8212	0.8152
Accuracy score in last epoch	0.9106	0.9125
Test accuracy score	0.8794999718666077	0.8755999803543091

Both models improved prediction. The model worked well against unseen data. First neural network’s accuracy score was slightly higher than the second one’s. The plots for the first one and the second one were: -

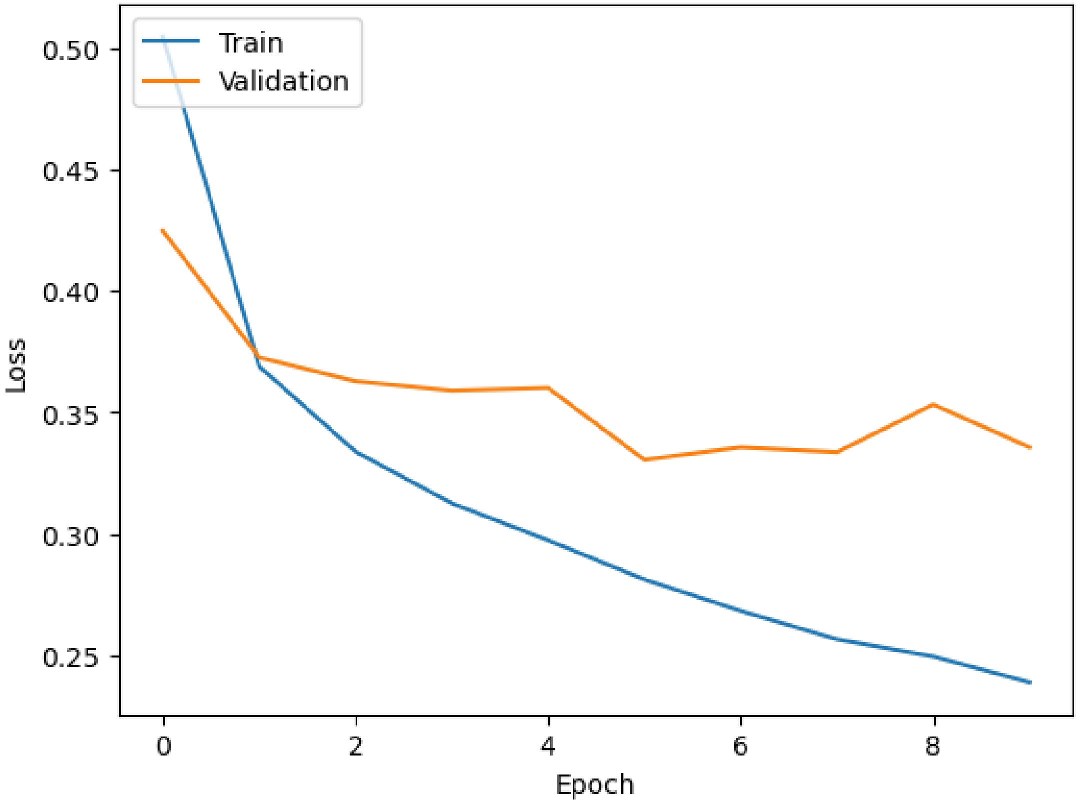
First neural network’s accuracy:-



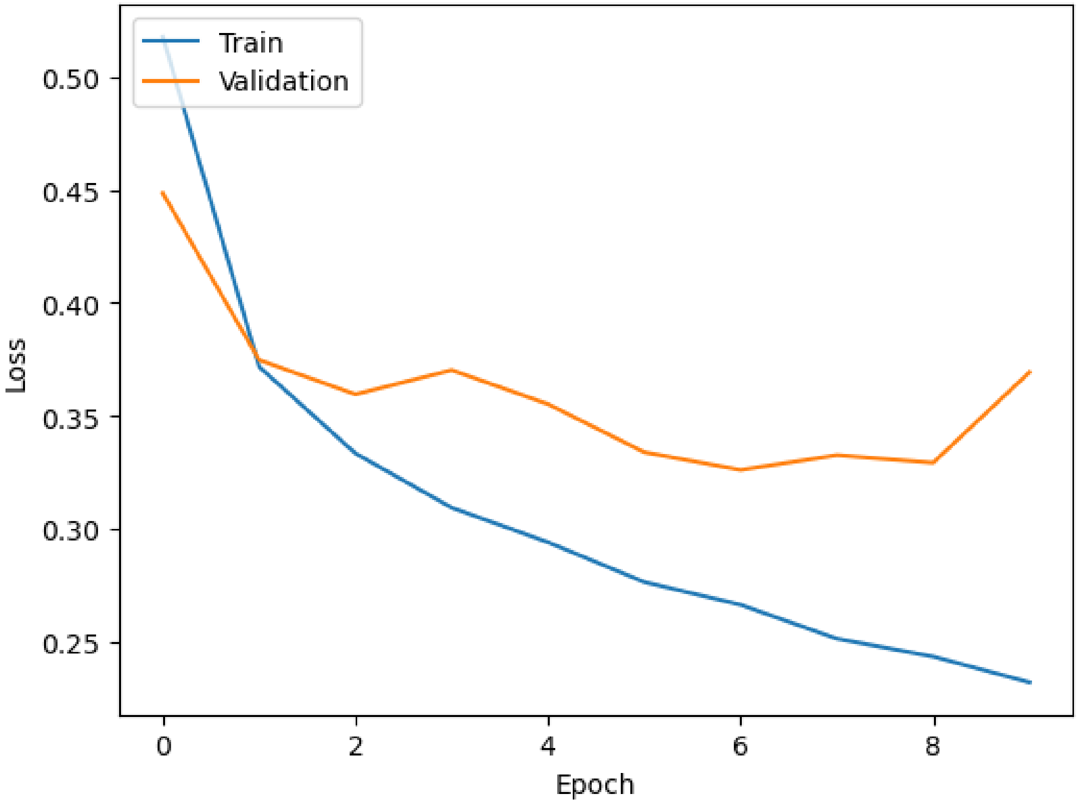
Second Neural networks accuracy: -



First Neural networks loss: -



Second Neural networks loss: -



The first one had a larger difference between training performance and testing performance. The difference was smaller for the second one. Therefore, although accuracy score decreased after adding more dense layers the performance difference in training and testing also reduced.

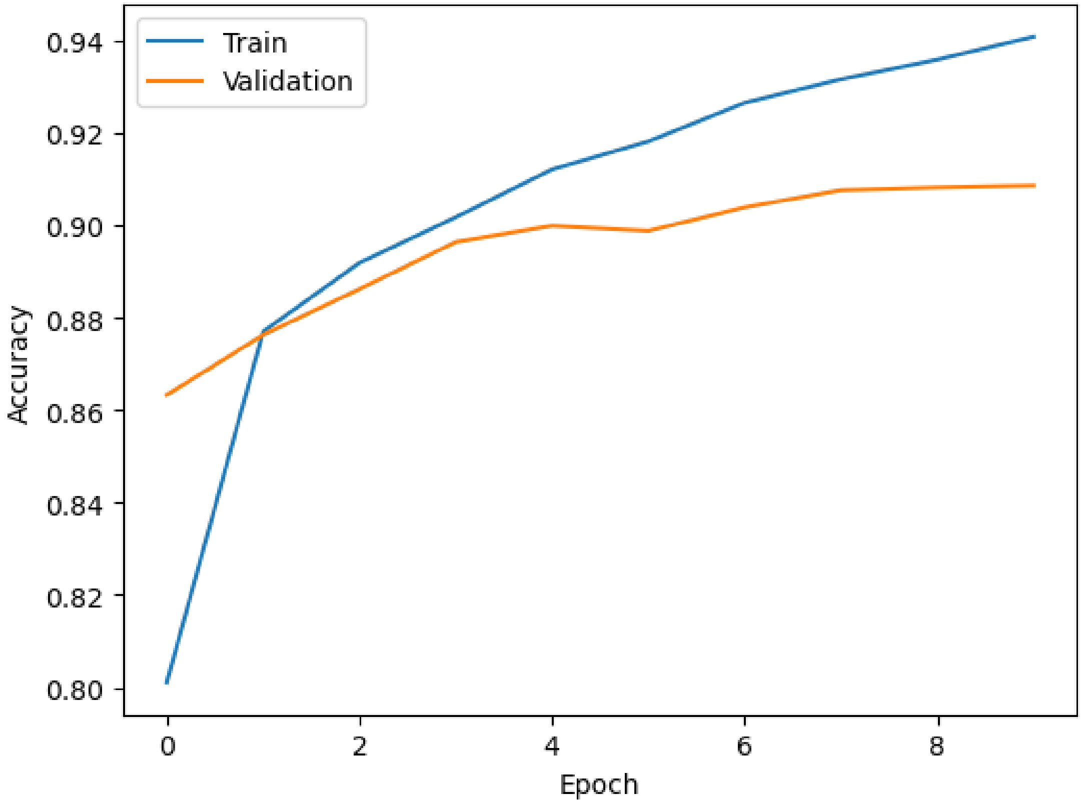
Task 2

There were 3 convolutional layers, 1 Dense layer, 1 output layer and 2 Maxpool layers. The first convolution layer had 32 filters. They were of size 3*3. Large number of filters allowed for richer feature extraction. Fine details were captured by small filter size. The size of the filter in max pooling was 2*2. It allowed the model to focus more on the significant features. It introduced simplicity to the model. The second and third convolutional layer had 64 filters of size and 3*3. This was so that the model can understand more complexity. The dense layers received flattened feature map. This was passed to 64 neurons which had their own weights and biases. If the output was negative, ReLu activation function outputted 0. Else the positive value was outputted. These were passed to the output layer. It had 10 neurons for 10 classes. SoftMax transformed the previous outputs into a probability. Categorical cross entropy calculated the loss. The loss was the difference between the true output and the predicted output. Adam optimiser adjusted weights. It made convergence fast. It used momentum and adaptive learning rates. Momentum prevented drastic weight updates. Adaptive learning rate scaled updates.

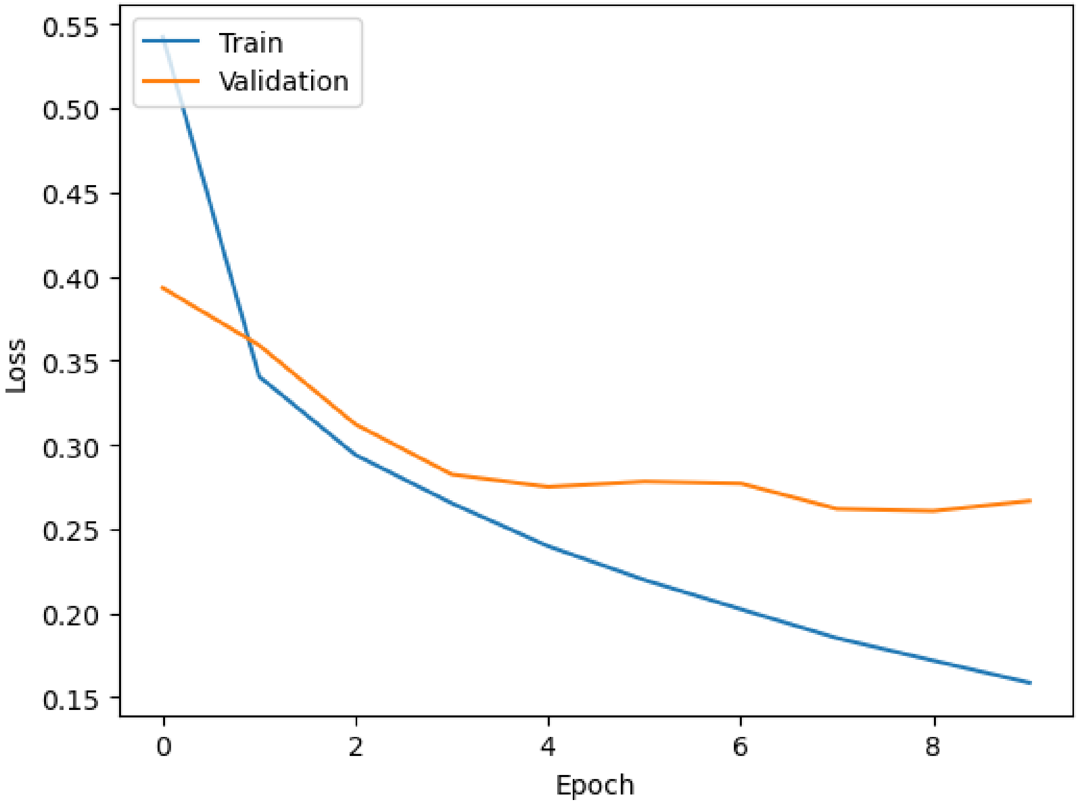
The model's accuracy improved during training. It reached 0.9408 after 10 epochs. The loss decreased from 0.5424 to 0.2667. This showed the model learned well. On the test set, the model achieved 0.9085999727249146 accuracy score. This meant it could generalize to new data.

The plots were: -

For accuracy:-


Train
Validation
Accuracy
0.94
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0
2
4
6
8
Epoch

For loss:-

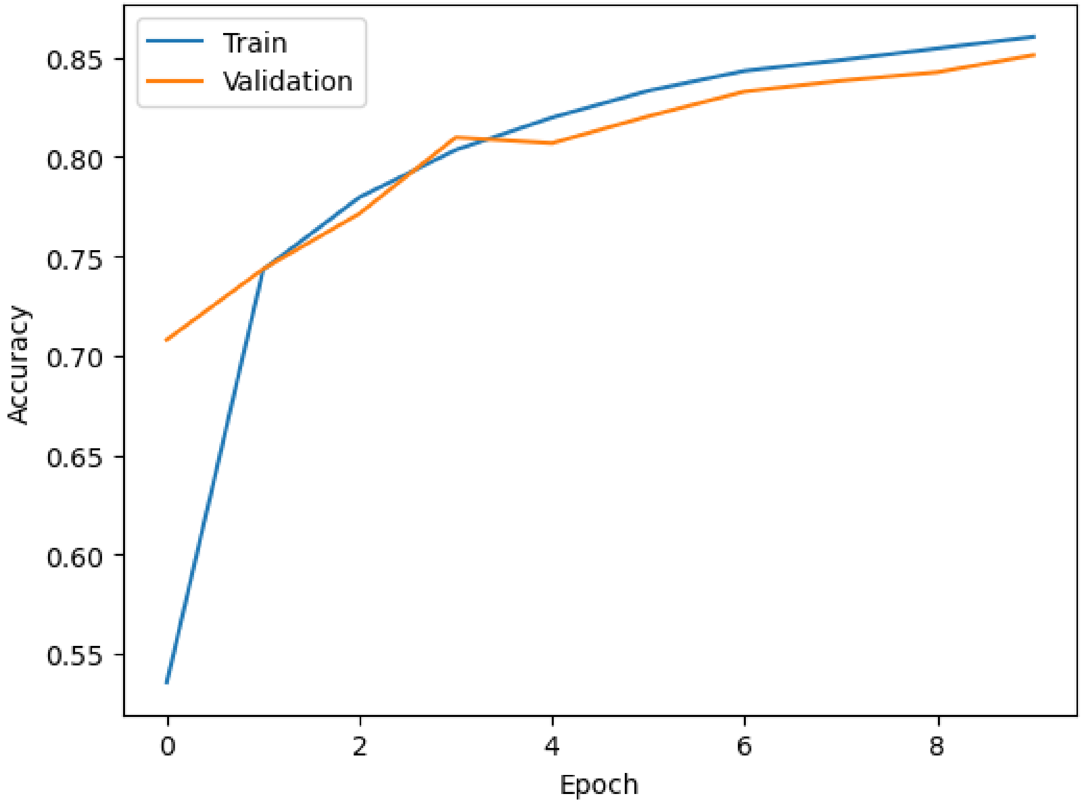


Overall, both training and testing were similar. They had some differences. This was shown by the gaps between the two graphs.

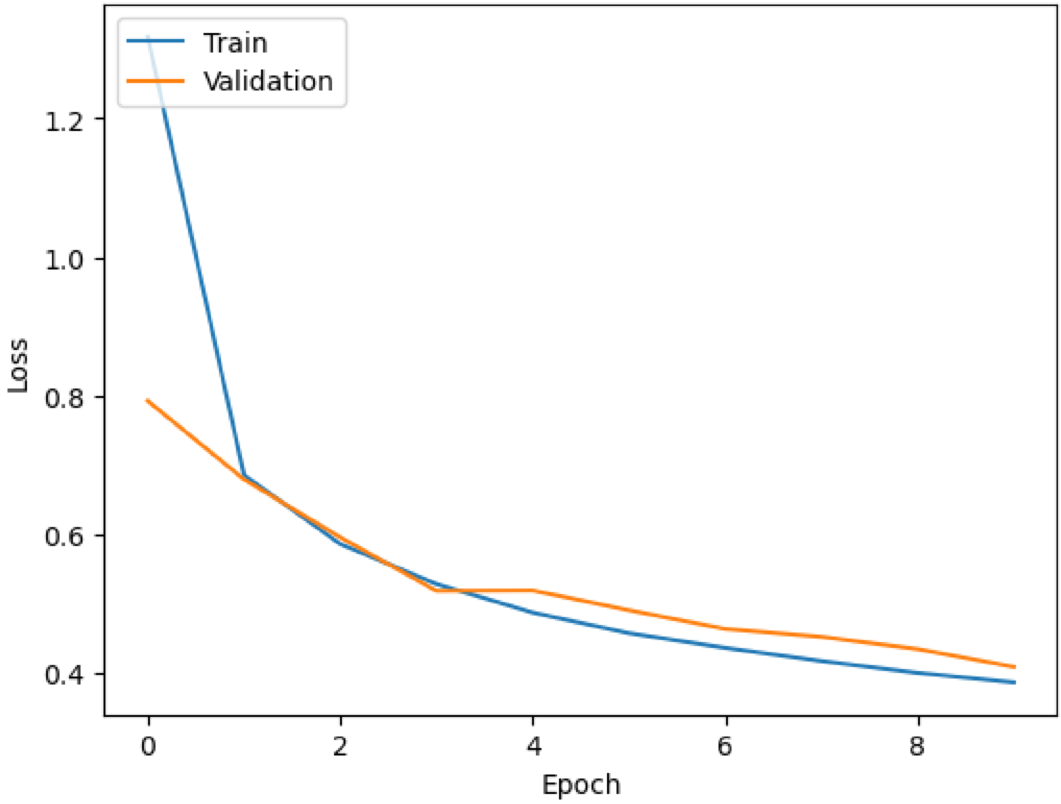
Task 3

Stochastic gradient descent was used. It updated weight based on the error calculated from batch of training data. It differed from Adam optimiser by having a fixed learning rate. It updated weights based on the steepness of the loss function. Adam used momentum and adaptive learning rates. The accuracy score increased from 0.5354 to 0.8605. The test accuracy score was 0.8513000011444092. The plots were:-

For Accuracy:-



For loss:-



There was hardly any difference between training and testing.

With adam optimiser the convolutional neurcal network scored better than stochastic gradient descent (SDG). Both models improved prediction. The models worked well against unseen data.

The accuracy scores for the all models were: -

Neural network with two dense layers	Neural network with two dense layers 4 dense layers	Convolutional neural network with Adam optimiser	Convolutional neural network with Adam optimiser SDG
0.8794999718666077	0.8755999803543091	0.9085999727249146	0.8513000011444092

The rankings of the models were best to worst performance were:-

- Convolutional neural network with Adam optimiser
- Neural network with two dense layers

- Neural network with two dense layers 4 dense layers
- Convolutional neural network with Adam optimiser SDG