Task -1 Based on different optimizers

i. Number of epochs considered for convergence for each of the architectures

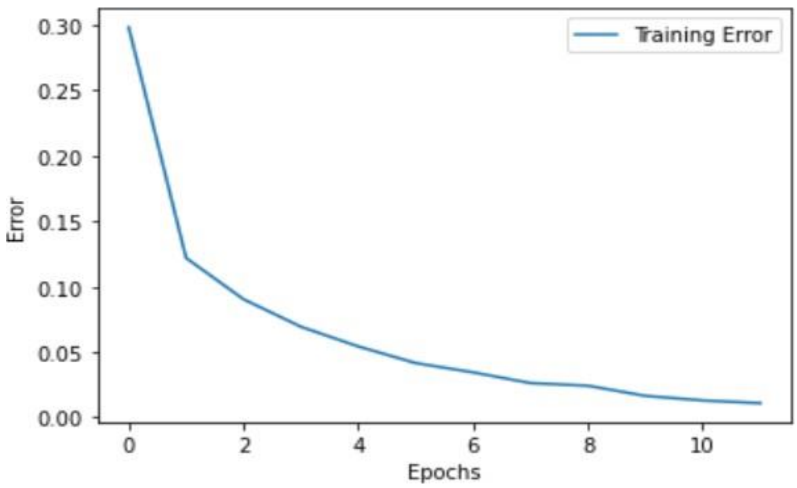
Optimizer	Number of Epochs	No. of Nodes
Stochastic gradient descent (SGD) algorithm	12	64,128,256
Batch gradient descent algorithm (vanilla gradient descent)	12	64,128,256
SGD with momentum (NAG)	12	64,64,128
RMSProp algorithm	12	128,128,128
Adam optimizer	12	128,256,128

Based on the accuracy of different numbers of epochs, 12 is considered as the optimal number of epochs for convergence for each of the architectures.

---

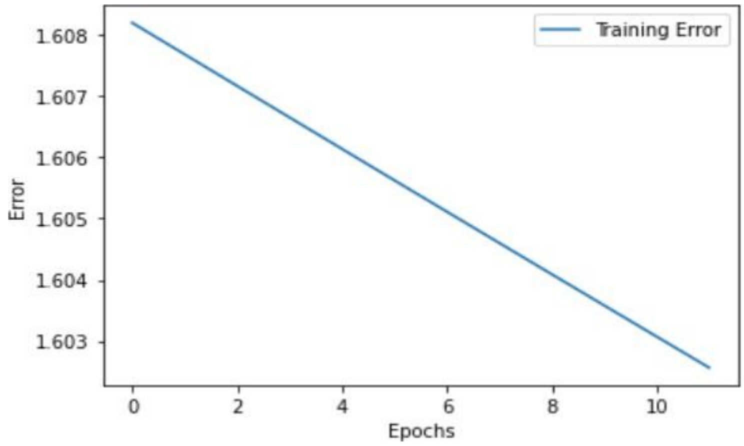
ii. Average training error (y-axis) vs. epochs (x-axis) for each of the architectures

(a) Stochastic gradient descent (SGD) algorithm



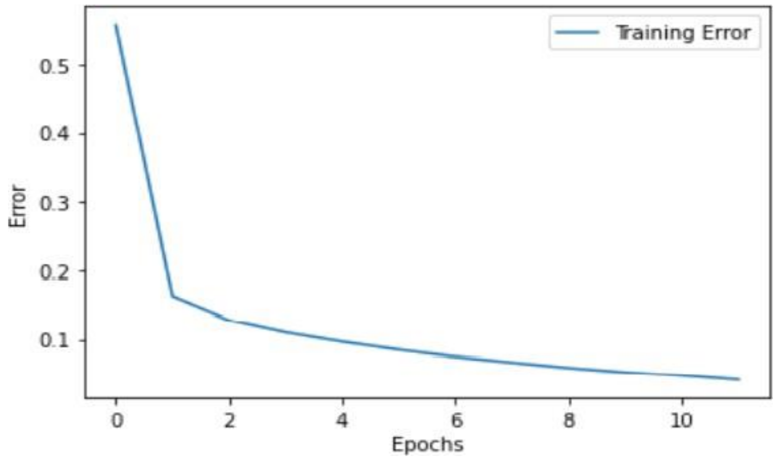
As the number of epochs increases, the average training error decreases. The error first decreases to around 0.12 as the first epoch is completed. Afterwards, the error decreases gradually with an increase in the number of epochs.

(b) Batch gradient descent algorithm



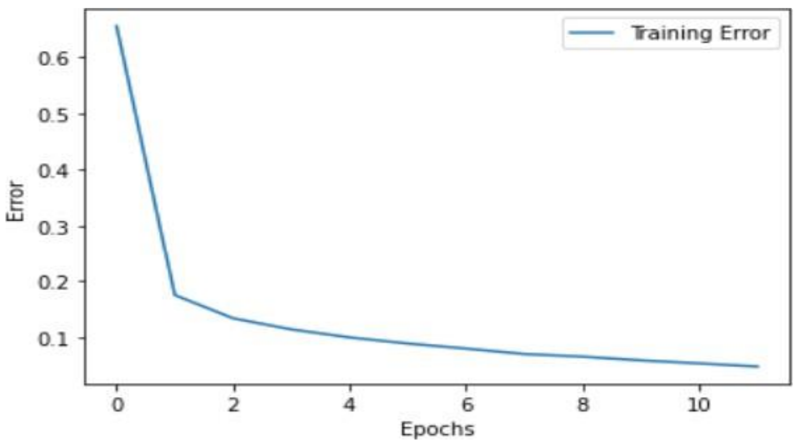
The average training error decreases linearly with an increase in the number of epochs.

(c) SGD with momentum (NAG)



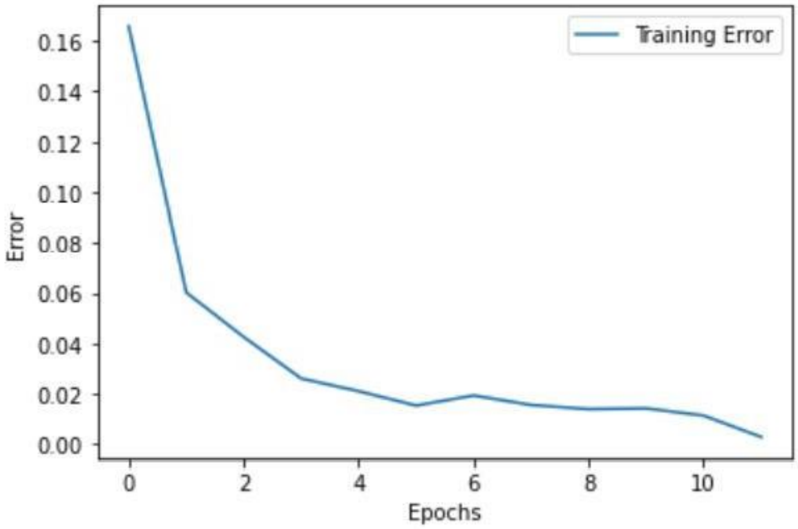
As the number of epochs increases, the average training error decreases.

(d) RMSProp algorithm



As the number of epochs increases, the average training error decreases.

(e) Adam optimizer



As the number of epochs increases, the average training error decreases. The error is less as compared to the other optimizers.

---

iii. The training accuracy and validation accuracy for each of the optimizers

Optimizer	Training Accuracy	Validation Accuracy
Stochastic gradient descent (SGD) algorithm	99.75	98.08
Batch gradient descent algorithm (vanilla gradient descent)	21.98	22.8
SGD with momentum (NAG)	98.61	97.58
RMSProp algorithm	98.59	97.84
Adam optimizer	99.94	98.76

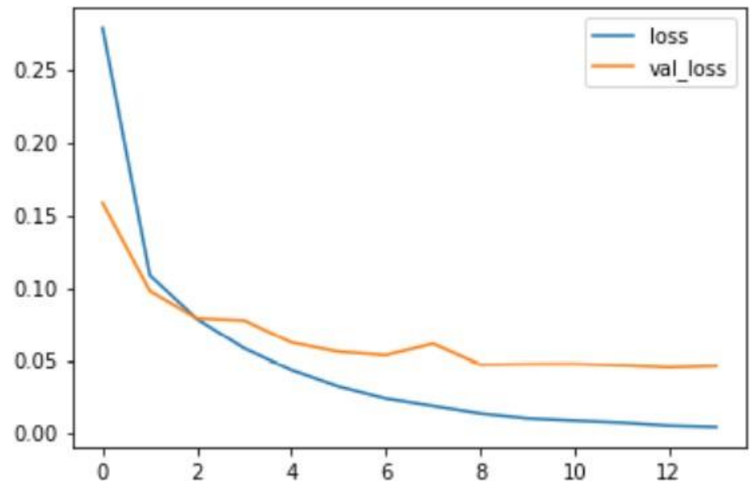
The training accuracy is the best in the case of Adam optimizer followed by SGD, NAG, RMSProp and Vanilla Gradient Descent.

The validation accuracy is the best in the case of Adam optimizer followed by SGD, RMSProp, NAG and Vanilla Gradient Descent.

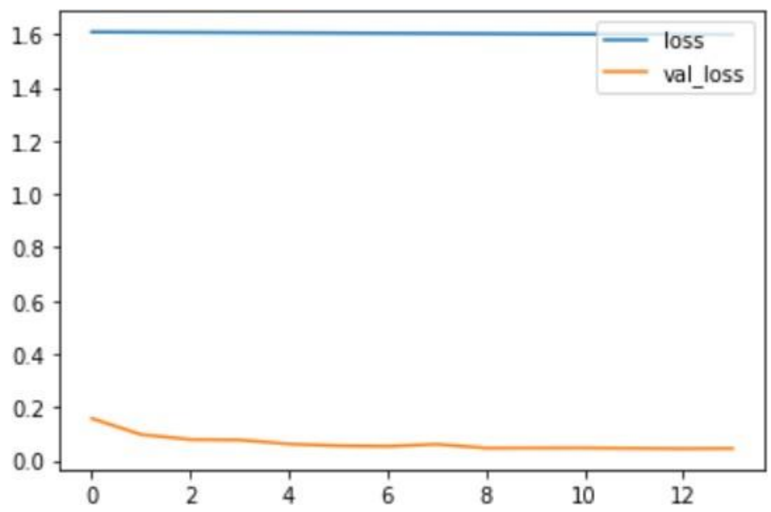
Adam optimizer updates the parameter by combining decaying learning rate and momentum, thus, overcoming most of the problems faced by other optimizers and giving the best accuracy.

Plots of training error and validation error for each of the optimizers

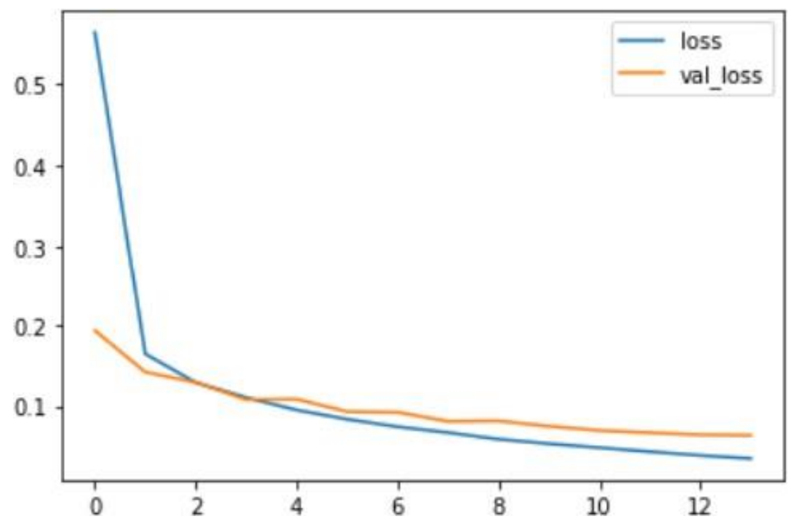
(a) Stochastic gradient descent (SGD) algorithm



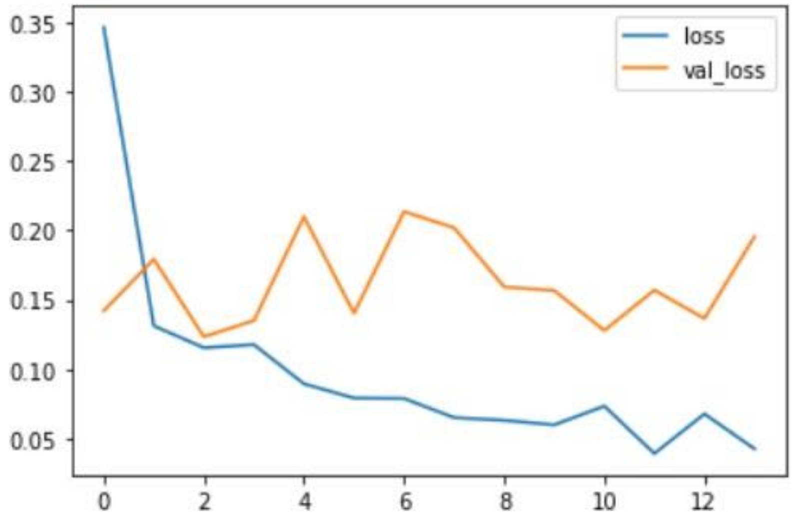
(b) Batch gradient descent algorithm



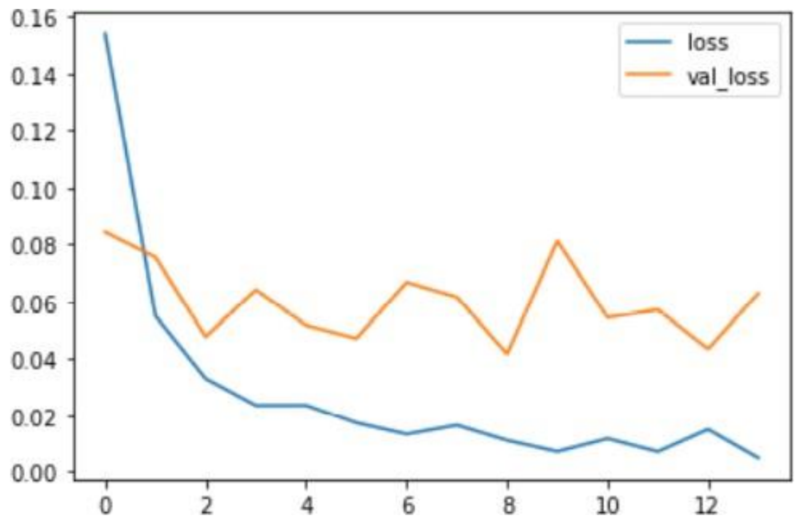
(c) SGD with momentum (NAG)



(d) RMSProp algorithm



(e) Adam optimizer



---

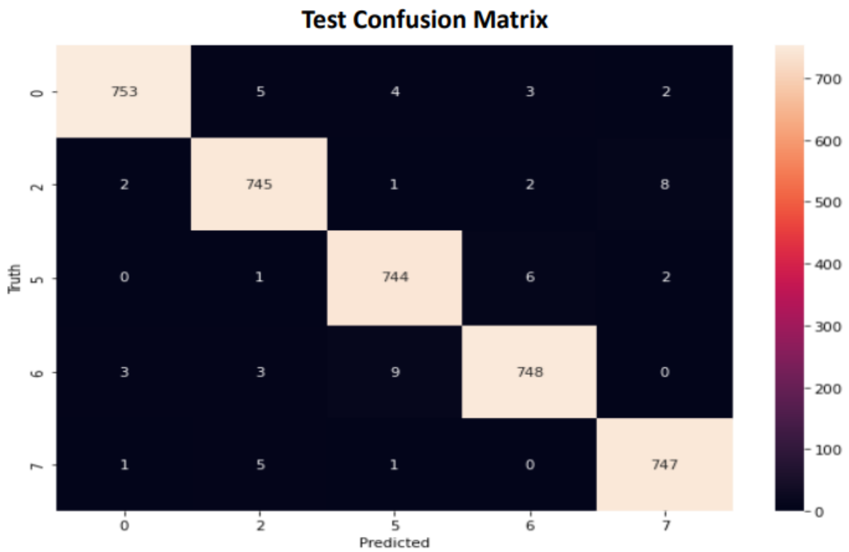
iv. The best architecture based on validation accuracy

Adam optimizer

- Training Accuracy: 99.54

- Validation Accuracy: 98.76

- Testing Accuracy: 98.47



The training accuracy for Adam optimizer is greatest followed by validation accuracy and test accuracy. Validation accuracy is less than training accuracy because the model is already familiar with the training data and validation data is a collection of data points which is new to the model.

The diagonal elements in the test confusion matrix represent the true positives and negatives. These give the accuracy i.e., 3737/3795=98.47

Task -2 Based on autoencoders

i. Autoencoders with one hidden layer

No. of Neurons	Training Error	Validation Error
32	0.0692	0.0692
64	0.0692	0.0694
128	0.0550	0.0514
256	0.0259	0.0230
512	0.0111	0.0094
1024	0.0019	0.0023

Autoencoder with 1024 neurons shows exact generated images as the training data which means the autoencoder is overtrained. Autoencoder with 512 have the least training as well as validation error as we can see in the above table (except 1024 neurons).

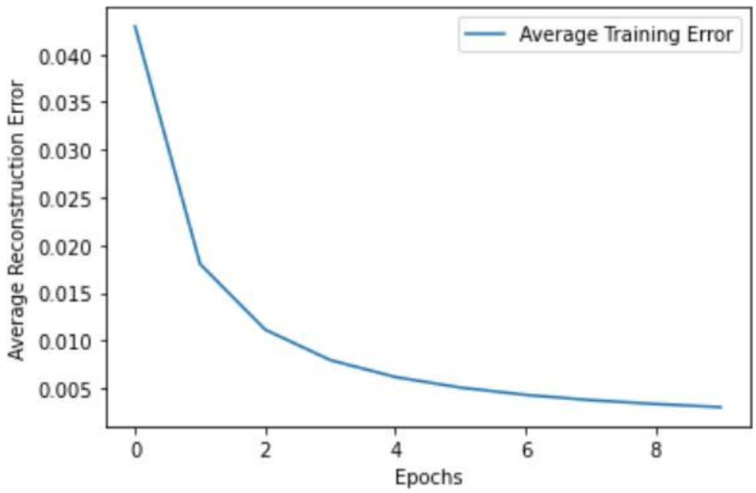
Also, if we take neurons more than 512(say 1024) error is decreasing but error decrement is less as compared to the previous successive models with less neurons. Thus, considering time complexity and error decrement rate, we selected an autoencoder with 512 neurons.

(a) Test reconstruction error for the chosen best architectures

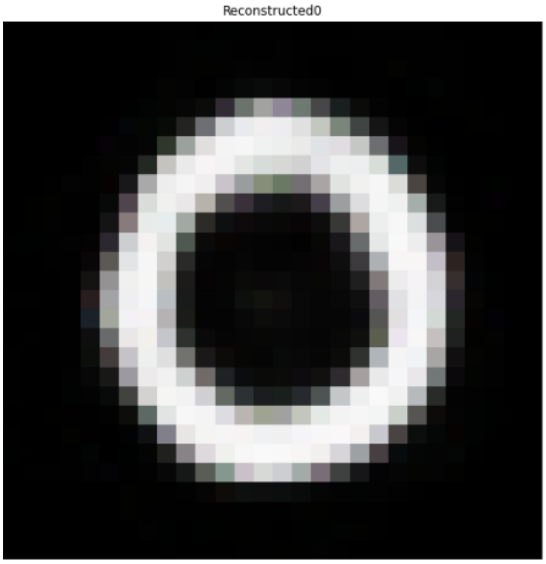
Test reconstruction error for an autoencoder with 512 neurons: 0.0034

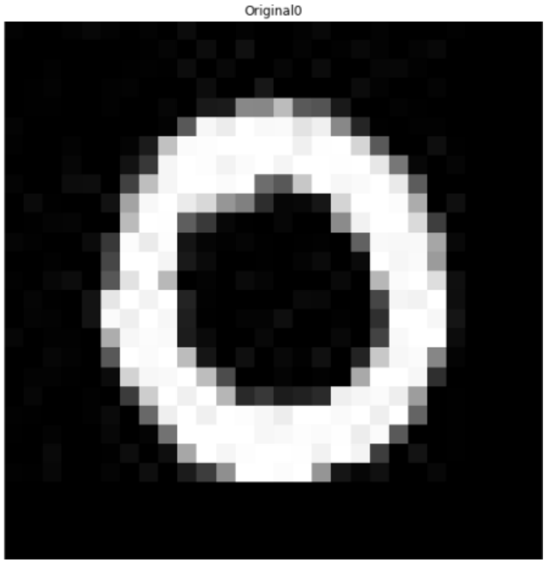
Cosine Similarity: 0.9842

(b) The plots of average training reconstruction error (y-axis) vs. epochs (x-axis) for the best architecture



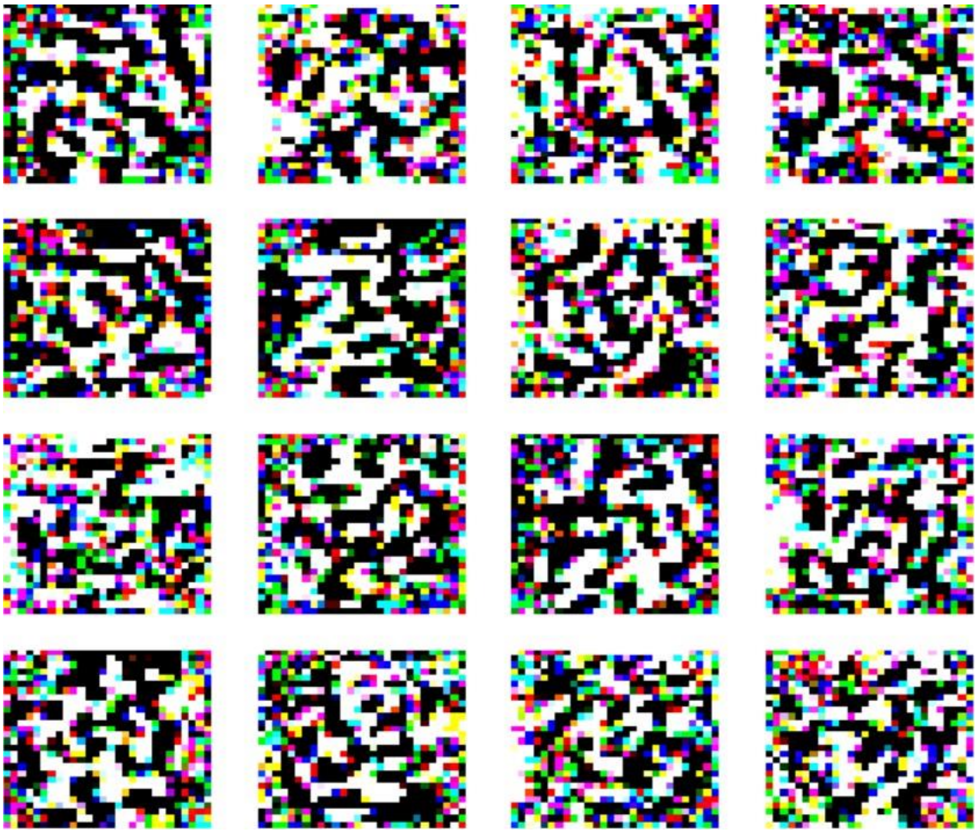
(c) Image from the Training set





(d) Image from the Validation set

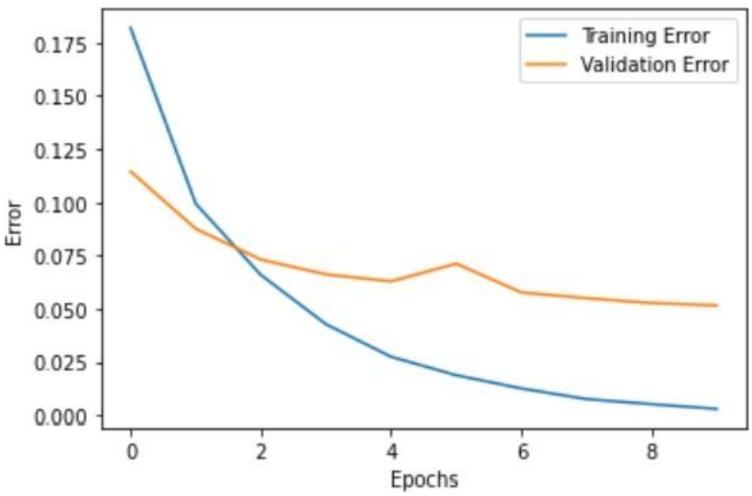
(e) Weight Visualization



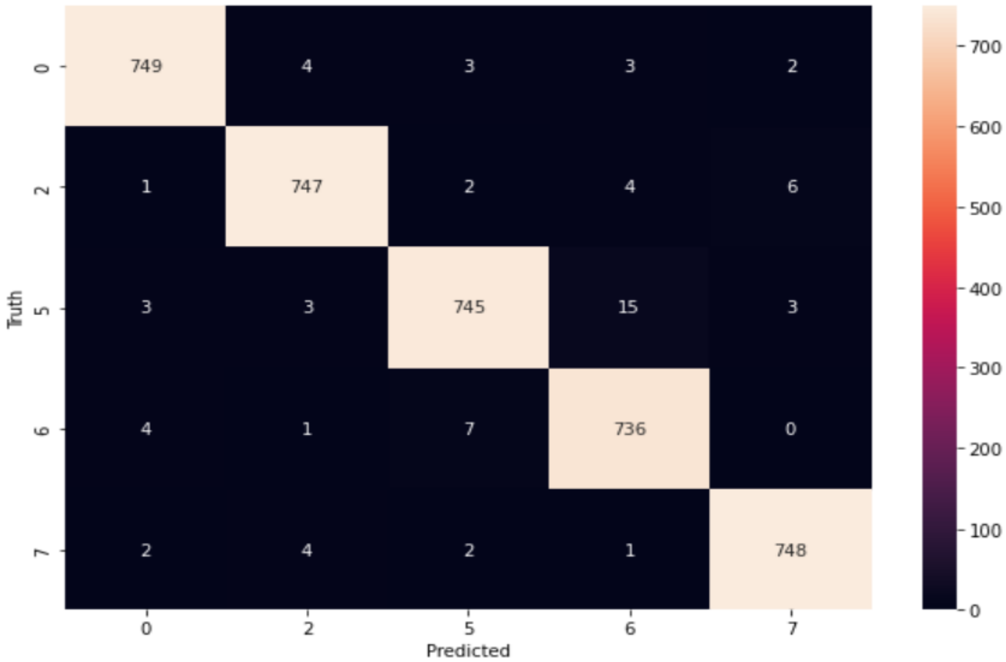
(f) Classification using the compressed representation from the encoder with one hidden layer

Training Accuracy: 99.99

Validation Accuracy: 98.5



Confusion matrix:



Test Accuracy: 98.2

---

ii. Autoencoders with 3 hidden layers

No. of Neurons	Training Error	Validation Error
32,64,128	0.0229	0.0233
64,64,64	0.0240	0.0241
64,128,128	0.0179	0.0184
64,128,256	0.0141	0.0148
128,256,512	0.0081	0.0093

Autoencoder with 128, 256, 512 neurons have the least training as well as validation error as we can see in the above table. Also, if we take neurons more than that error is decreasing but error decrement is less as compared to the previous successive models with less neurons.

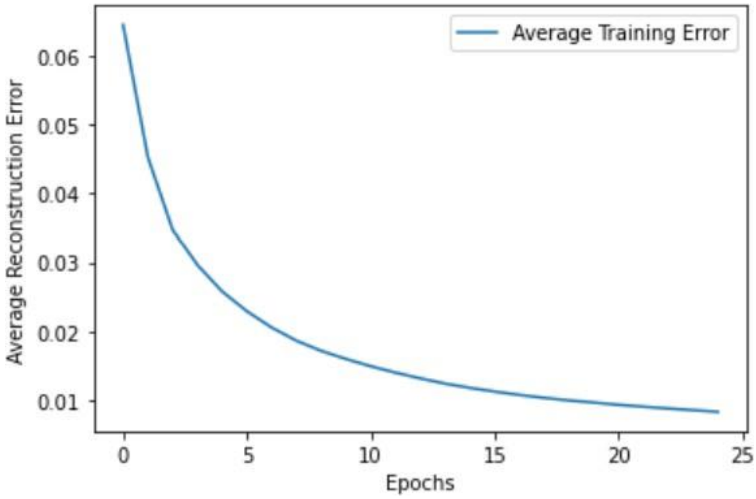
Thus, considering time complexity and error decrement rate, we selected an autoencoder with 128, 256, 512 neurons.

(a) Test reconstruction error for the chosen best architectures

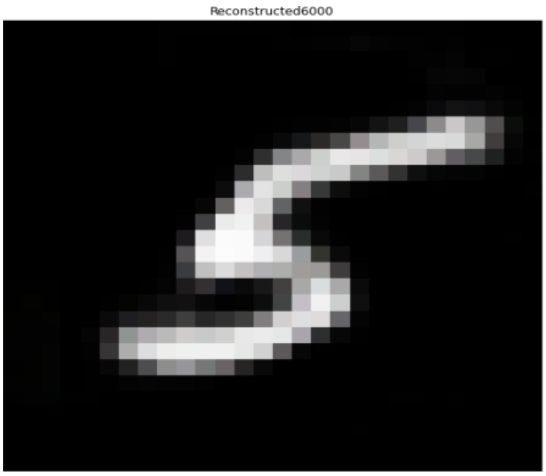
Test reconstruction error for an autoencoder with 128, 256, 512 neurons: 0.0096

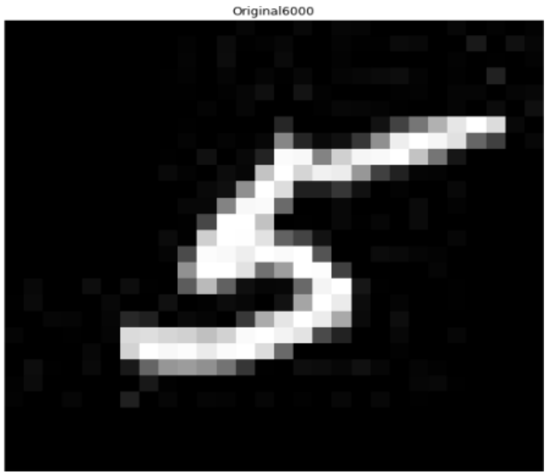
Cosine Similarity: 0.9551

(b) The plots of average training reconstruction error (y-axis) vs. epochs (x-axis) for the best architecture



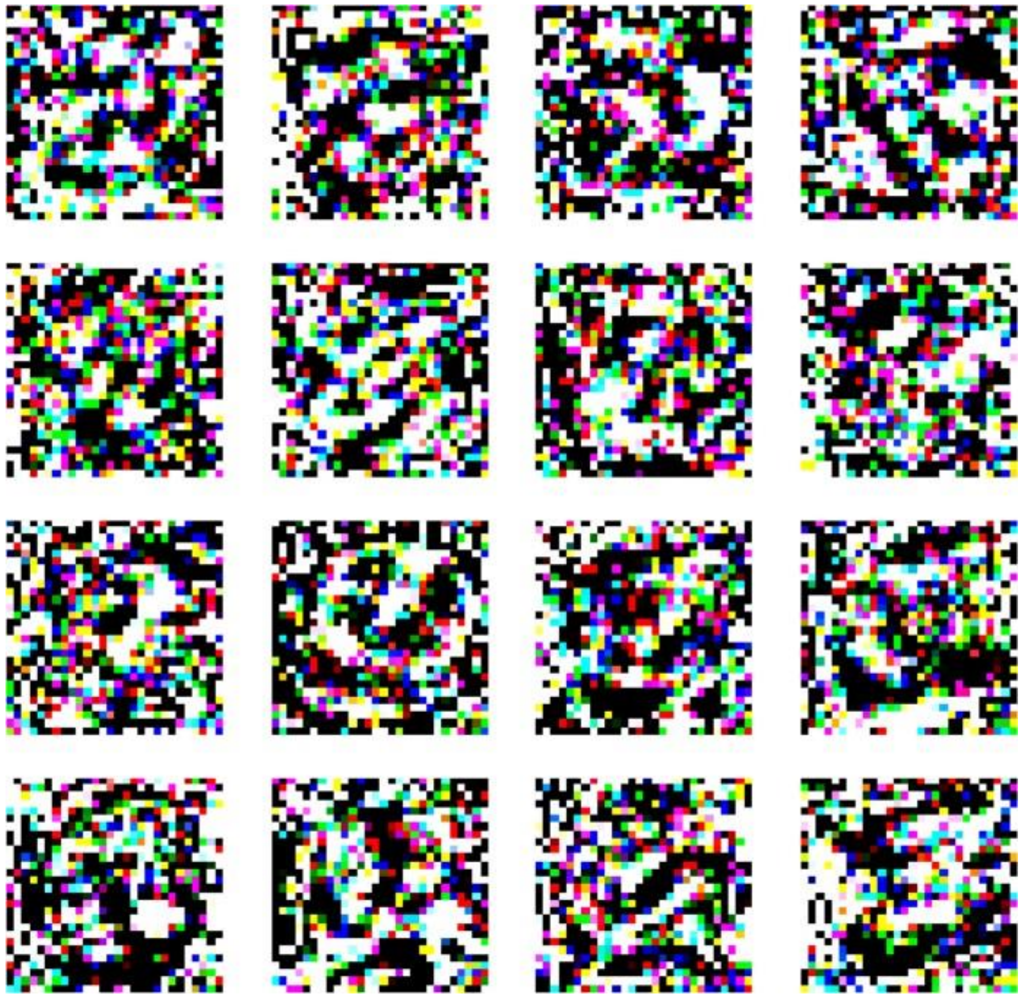
(c) Image from the training set





(d) Image from the Validation set

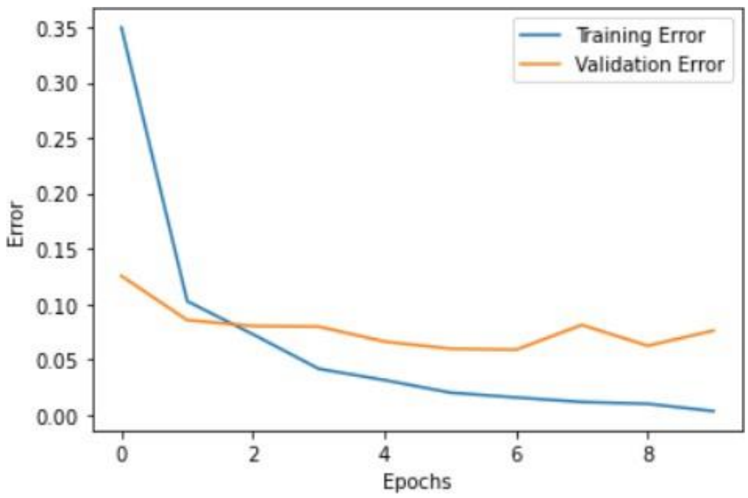
(e) Weight Visualization



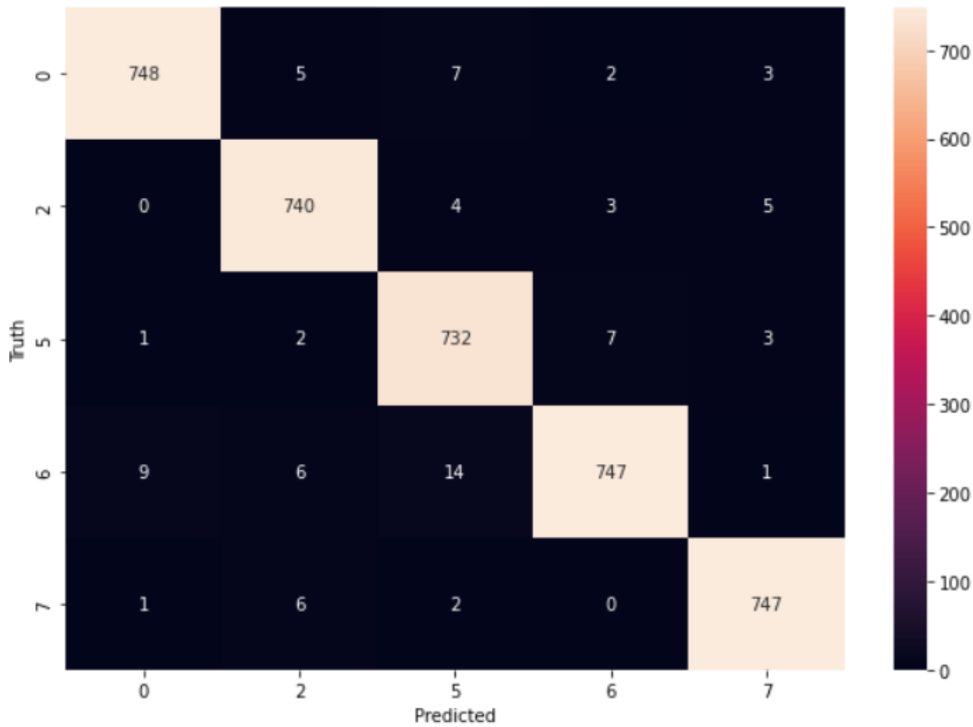
(f) Classification using the compressed representation from the encoder with one hidden layer

Training Accuracy: 99.93

Validation Accuracy: 98.18



Confusion matrix:



Test Accuracy: 97.9