CS-671 Deep Learning and Applications

Assignment 4

Autoencoders

Submitted by:
Group 13
Prachi Sharma (V21078)
Ayush Tiwari (T22053)

Instructor:
Prof. A D Dileep

April 16, 2023

Table of Contents

Autoencoders

Autoencoders are a type of feedforward neural networks that find applications in various types of unsupervised learning tasks, like input reconstruction and dimensionality reduction. The basic idea behind an autoencoder is to have an output layer with the same dimension as inputs and try to reconstruct each input by passing it through the network. It has two main parts:

- an *encoder* that encodes the input *x* into a hidden representation *h* and,
- a *decoder* that decodes the input again from the hidden representation.

The purpose of an autoencoder is to reconstruct the input with minimum distortion and learn useful characteristics of input.

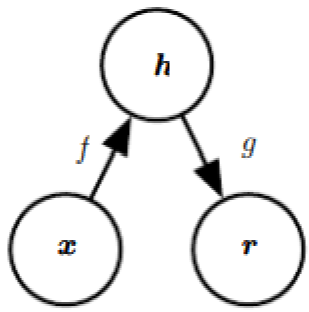


Figure 2.5: Structure of an autoencoder, mapping an input *x* through hidden representation *h* to an output reconstruction *r*. *f* is the encoder function and *g* is the decoder function. Image source: (Bengio et al., 2016)

1. Types of Autoencoders

An autoencoder is of two types depending on the number of units in the hidden representation, also called as encoded representation. An *overcomplete autoencoder* has its hidden representation dimension more than the input dimension. A trivial overcomplete autoencoder is when the autoencoder can learn to exactly copy the input *x* to the hidden layer and then copy the hidden representation h into reconstruction *r*. This type of encoding may not give any useful information about the hidden properties of the input. An *undercomplete autoencoder* has its hidden representation dimension less than the input dimension. It tries to learn the most important features of the input vectors x_n as:

- Encoding: $h_n = f(W^T x_n + w_0^{(e)})$
- Decoding: $r_n = g(W^{*T} h_n + w_0^{(d)})$
- Loss function: $L(\theta)$, θ is model parameters

Choice of activation functions for encoder and decoder functions depends on the type of input variables. In general, sigmoidal function is typically used for encoder functions. For decoder functions, in case of real values, we use linear activation function and logistic for binary input variables. The loss function is the mean squared error between true input and reconstructed representation:

$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=0}^{d}(r_{ij} - x_{ij})^2$$

2. Denoising Autoencoders

To avoid memorization and enable more generalization in the model, we regularize it in different ways. A *denoising autoencoder* is one way to add regularization. In denoising autoencoders, noise is added to the input data using some probabilistic measure before feeding it to the network. The loss function is then calculated between the original input data and the final reconstruction. This way, the network has to depend on the hidden information to learn input data using the noisy data, making it more robust to unseen data.

Experiments, Results and Analysis

1. Dimensionality reduction using PCA

In this section, we have reduced the original input dimension of 784 to a dimension of 32, 64, 128 and 256 using Principal Component Analysis technique. The inputs are transformed into a new coordinate system defined by principal components and these transformations are then used as input to FCNNs to perform classification. The classification accuracy of each model on the validation set is compared to decide the best architecture for a particular compressed representation. Finally, test accuracy for all best performing models is compared to observe the best reduced representation.

To check classification performance of reduced representation, we have done experiments on 3 different FCNN architectures.

Architecture	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Hidden layer 4	Hidden layer 5	Output
A	32/64/128/256	512	256	128	-	-	5
B		512	256	128	64	-	5
C		512	256	128	64	32	5

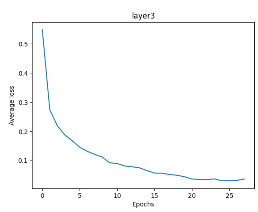
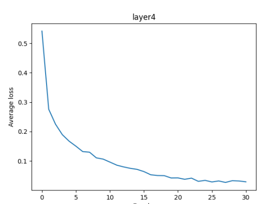
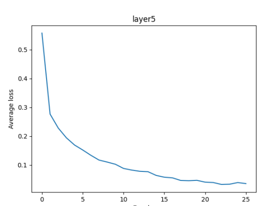
Parameter values for all models:

Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Softmax
Loss function	Cross Entropy
Learning rate (η)	0.001
Batch size	32
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999

Epsilon (ϵ)	10^{-8}
Convergence criterion	difference between successive epochs is less than or equal to 10^{-4}.

A. 32 Principal Components

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
32-512-256-128-5		97.05%	94.80%
32-512-256-128-64-5		97.95%	95.17%
32-512-256-128-64-32-5		**98.84%**	**95.25%**

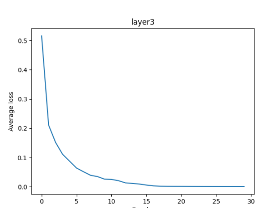
Results of best selected architecture on test data:

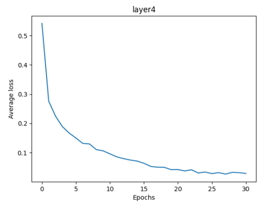
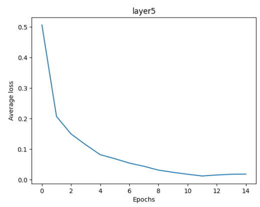
On comparing validation accuracy of different FCNN, we can observe that the network with 5 hidden layers (32-512-256-128-64-32-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

Confusion matrix	**ACTUAL**					
	PREDICTED	749	4	3	3	0
		4	713	21	6	15
		4	25	709	8	13
		1	14	4	738	2
		4	6	15	5	729
Classification accuracy	**Train**		**Validation**		**Test**	
	98.84%		95.25%		95.41%	

B. 64 Principle Components

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
64-512-256-128-5		97.63%	**96.54%**

64-512-256-128-64-5		98.42%	94.83%
64-512-256-128-64-32-5		**99.04%**	95.57%

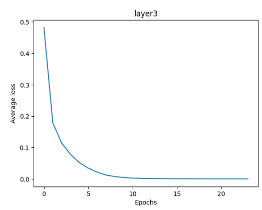
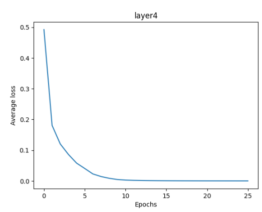
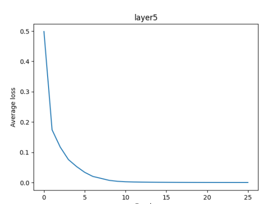
Results of best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 3 hidden layers (64-512-256-128-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

<table>
<tr><td rowspan="6">Confusion matrix</td><td colspan="6">A C T U A L</td></tr>
<tr><td rowspan="5">P R E D I C T E D</td><td>746</td><td>4</td><td>4</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>695</td><td>34</td><td>17</td><td>9</td></tr>
<tr><td>3</td><td>9</td><td>736</td><td>3</td><td>8</td></tr>
<tr><td>2</td><td>8</td><td>5</td><td>741</td><td>3</td></tr>
<tr><td>6</td><td>3</td><td>24</td><td>5</td><td>731</td></tr>
<tr><td rowspan="2">Classification accuracy</td><td colspan="2">Train</td><td colspan="2">Validation</td><td colspan="2">Test</td></tr>
<tr><td colspan="2">97.63%</td><td colspan="2">96.54%</td><td colspan="2">96.17%</td></tr>
</table>

C. 128 Principle Components

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
128-512-256-128-5		98.15%	96.02%
128-512-256-128-64-5		98.84%	**96.15%**
128-512-256-128-64-32-5		**99.17%**	96.04%

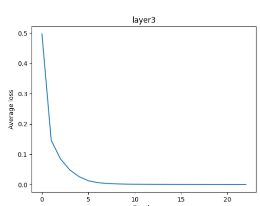
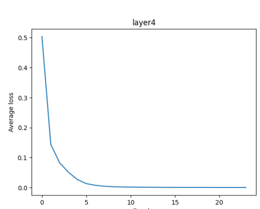
Results of best selected architecture on test data:

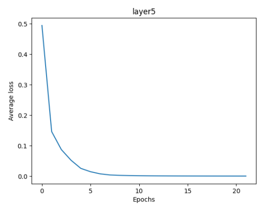
On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (128-512-256-128-64-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

Confusion matrix	ACTUAL					
	PREDICTED	745	7	5	2	0
		0	727	14	8	10
		1	21	719	3	15
		5	12	1	737	4
		5	8	12	4	730
Classification accuracy	**Train**		**Validation**		**Test**	
	98.84%		96.15%		96.07	

D. 256 Principle Components

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
256-512-256-128-5		98.52%	96.49%
256-512-256-128-64-5		**99.24%**	96.41%

256-512-256-128-64-32-5		99.06%	**96.52%**

Results of best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 5 hidden layers (256-512-256-128-64-32-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

Confusion matrix		**ACTUAL**				
	PREDICTED	753	3	1	1	1
		5	719	13	13	9
		3	20	723	5	8
		1	12	3	739	4
		6	5	16	2	730
Classification accuracy	**Train**		**Validation**		**Test**	
	99.06%		96.52%		96.41%	

E. Best Reduced Dimension Representation

I. Performance comparison with different models

Architecture	Classification Accuracy on Test data
784-512-256-128-64-10 (Best of Assignment 3)	**97.44%**
64-512-256-128-64-5 (Best of PCA)	96.17%

2. Reconstruction and Dimensionality reduction using Autoencoders

2.1. Single hidden layer autoencoder

For single hidden layer autoencoder, we have considered 4 architectures

Architecture	Input layer	Hidden layer (Bottleneck layer)	Output layer
1	784	32	784
2	784	64	784
3	784	128	784
4	784	256	784

The output layer has 784 neurons since the flattened input has 784 neurons and our task is to reconstruct the input using different autoencoders. The data is normalized using MinMax normalization to scale all input features in the range of [0,1].

Parameters for all models:

Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Logistic
Loss function	Mean Squared Error
Batch size	32
Learning rate (η)	0.001
Optimiser	Adam

Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}
Convergence criterion	Difference between successive epochs is less than or equal to 10^{-4}.

Further, to check classification performance of reduced representation, encoded representation (output of bottleneck layer) is then used to train different fully connected neural networks. We have done experiments on 3 different FCNN architectures.

Architecture	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Hidden layer 4	Hidden layer 5	Output
A	32/64/128/256	512	256	128	-	-	5
B		512	256	128	64	-	5
C		512	256	128	64	32	5

Parameter values for all models:

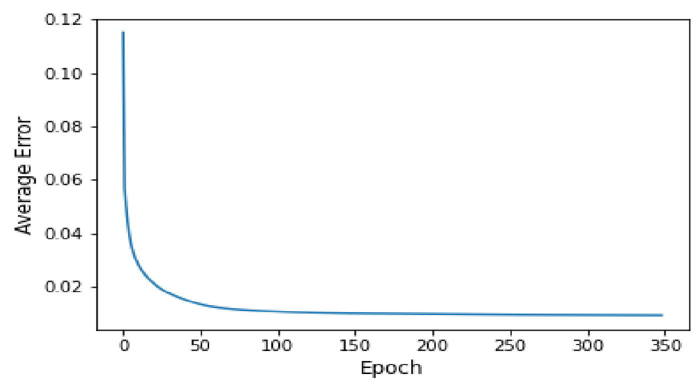
Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Softmax
Loss function	Cross Entropy
Learning rate (η)	0.001
Batch size	32
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}

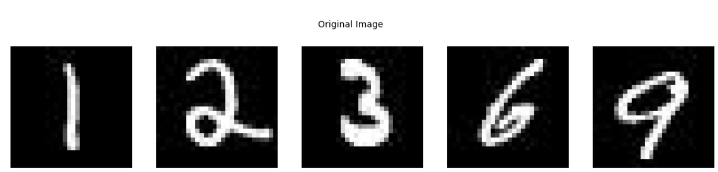
Convergence criterion	difference between successive epochs is less than or equal to 10^{-4}.

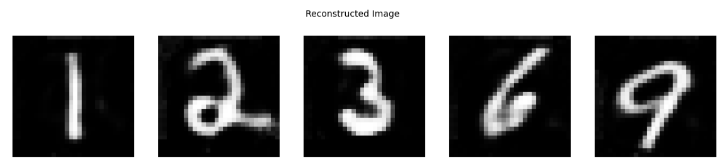
The classification accuracy of each model on the validation set is compared to decide the best architecture for a particular compressed representation. Finally, test accuracy for all best performing models is compared to observe the best reduced representation.

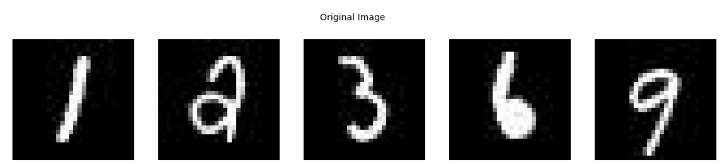
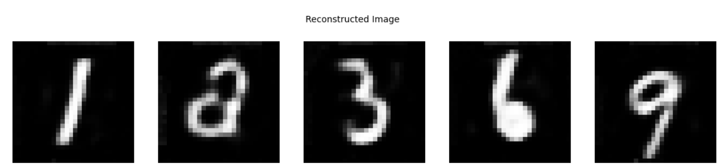
A. Architecture 1 – 784I-32H-784O

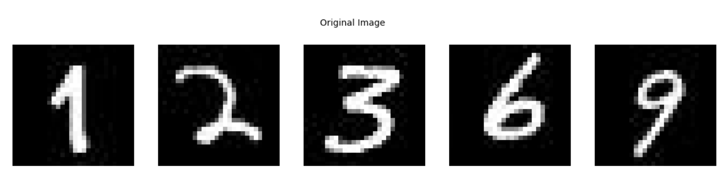
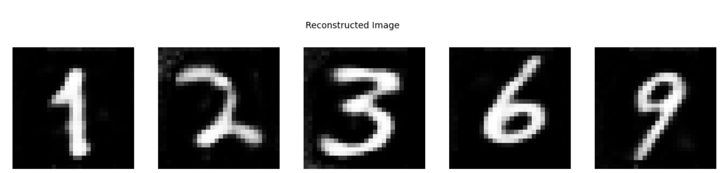
Results and Observations of Autoencoder:

Error v/s Epoch			
Mean Squared Error	**Train**	**Validation**	**Test**
	0.0090	0.106	0.107

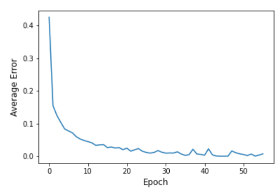
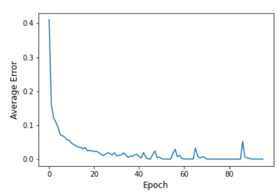
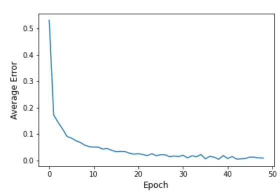
Original and reconstructed image of train data	
Original Image	

Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	
Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
32-512-256-128-5		99.75%	98.81%
32-512-256-128-64-5		**100%**	**98.89%**
32-512-256-128-64-32-5		99.72%	98.55%

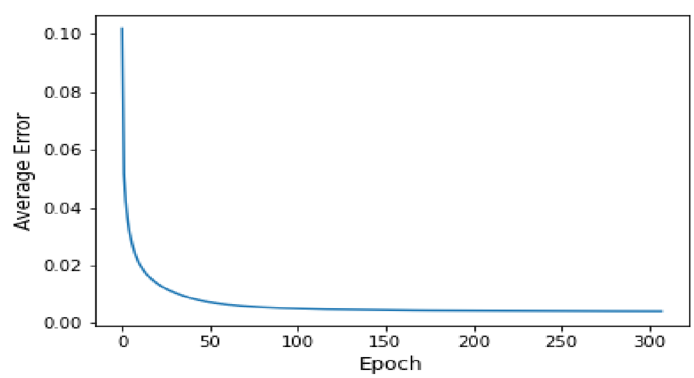
Results of best selected architecture on test data:

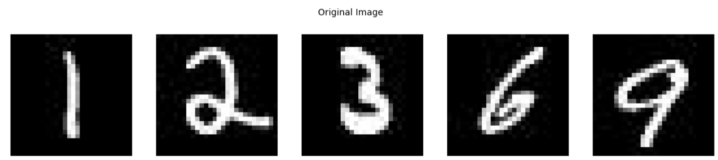
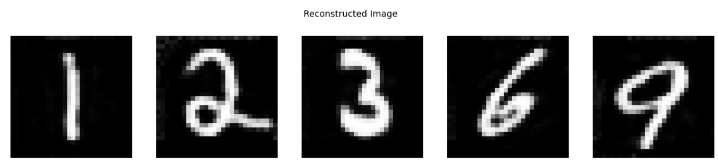
On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (32-512-256-128-64-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

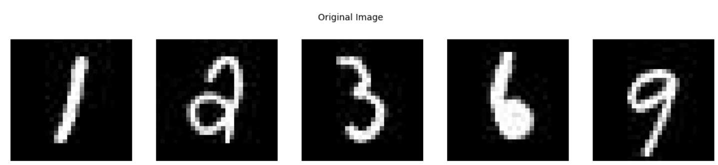
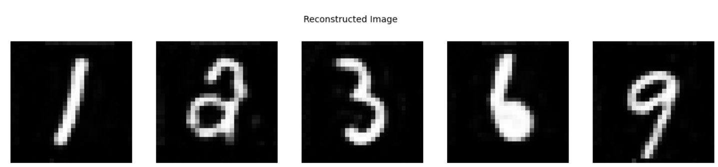
<table>
<tr><td rowspan="6">Confusion matrix</td><td colspan="6">A C T U A L</td></tr>
<tr><td rowspan="5">P
R
E
D
I
C
T
E
D</td><td>749</td><td>7</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>4</td><td>747</td><td>4</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>5</td><td>747</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>3</td><td>3</td><td>750</td><td>2</td></tr>
<tr><td>2</td><td>2</td><td>3</td><td>0</td><td>752</td></tr>
<tr><td rowspan="2">Classification accuracy</td><td colspan="2">Train</td><td colspan="2">Validation</td><td colspan="2">Test</td></tr>
<tr><td colspan="2">100%</td><td colspan="2">98.89%</td><td colspan="2">98.68%</td></tr>
</table>

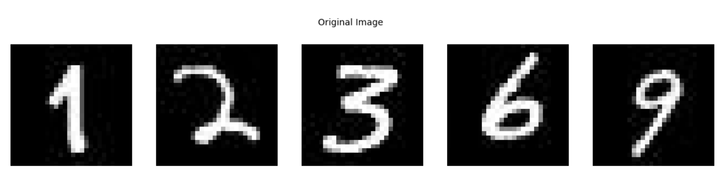
B. Architecture 2 – 784I-64H-784O

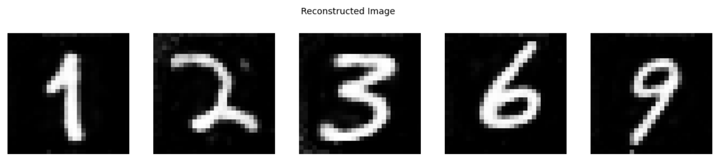
Results and Observations of Autoencoder:

<table>
<tr><td>Error v/s Epoch</td><td colspan="3">
</td></tr>
<tr><td rowspan="2">Mean Squared Error</td><td>Train</td><td>Validation</td><td>Test</td></tr>
<tr><td>0.0040</td><td>0.0055</td><td>0.0055</td></tr>
</table>

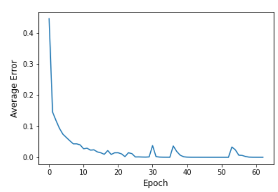
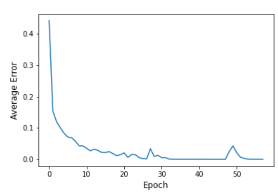
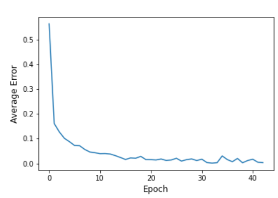
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
64-512-256-128-5		**100%**	98.63%
64-512-256-128-64-5		**100%**	**98.66%**
64-512-256-128-64-32-5		99.91%	98.55%

Results of best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (128-512-256-128-64-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

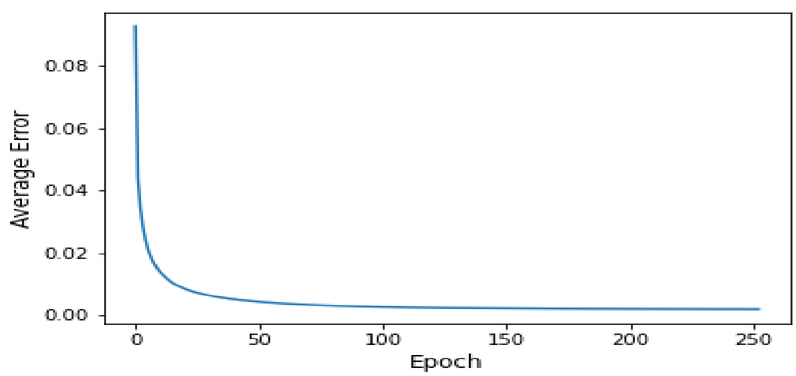
Confusion matrix	A C T U A L

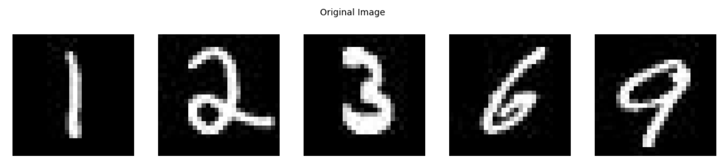
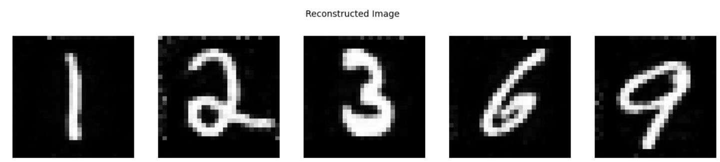
	PREDICTED	754	3	0	1	1
		2	743	6	2	6
		2	3	748	2	4
		2	3	2	751	1
		4	0	3	1	751

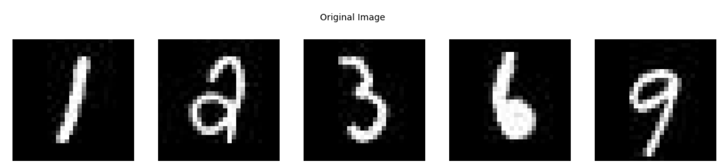
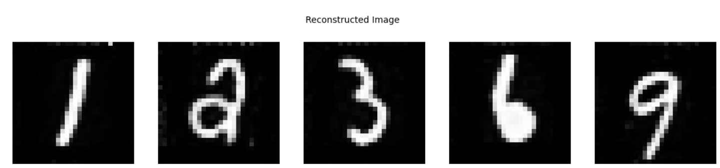
Classification accuracy	Train	Validation	Test
	100%	98.66%	98.73%

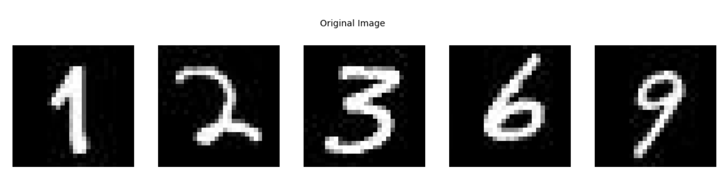
C. Architecture 3 – 784I-128H-784O

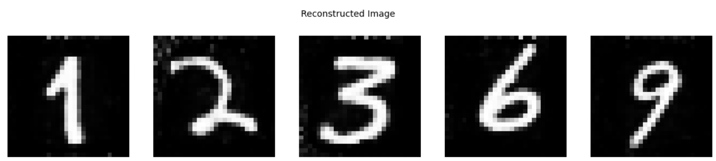
Results and Observations of Autoencoder:

Error v/s Epoch			
Mean Squared Error	Train	Validation	Test
	0.0018	0.0032	0.0032

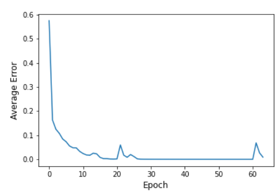
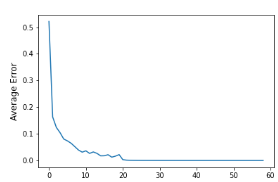
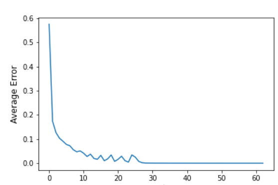
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
128-512-256-128-5		99.66%	98.02%
128-512-256-128-64-5		**100%**	98.15%
128-512-256-128-64-32-5		**100%**	**98.26%**

Results of best selected architecture on test data:

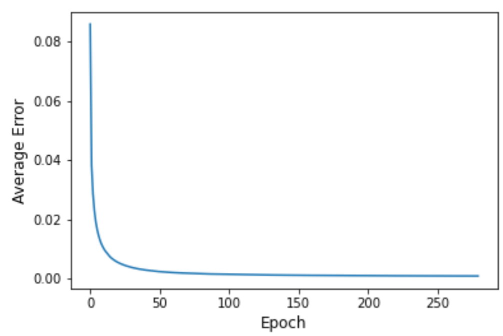
On comparing validation accuracy of different FCNN, we can observe that the network with 5 hidden layers (128-512-256-128-64-32-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

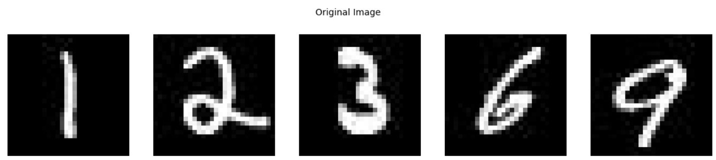
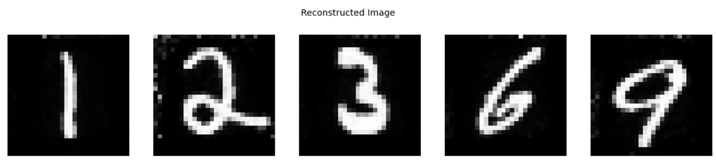
Confusion matrix	**A C T U A L**

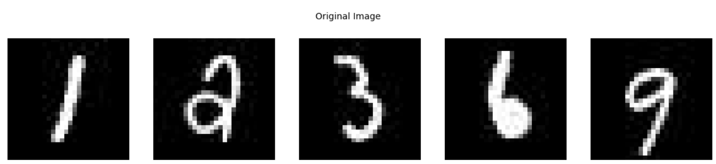
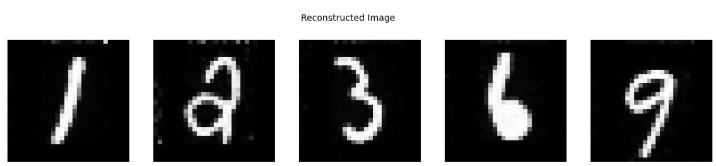
	PREDICTED	752	4	1	2	0
		4	732	14	5	4
		1	10	741	1	6
		1	5	3	748	2
		3	2	7	1	746
Classification accuracy	**Train**		**Validation**		**Test**	
	100%		98.26%		97.99%	

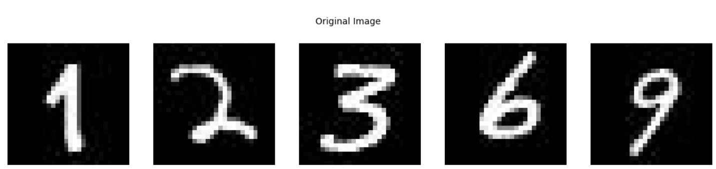
D. Architecture 4 – 784I-256H-784O

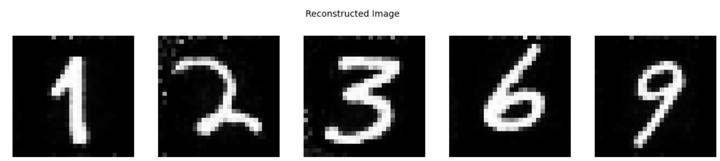
Results and Observations of Autoencoder:

Error v/s Epoch			
Mean Squared Error	**Train**	**Validation**	**Test**
	0.000927	0.0022	0.0023

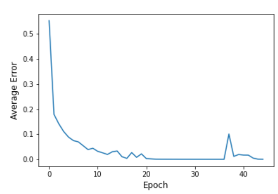
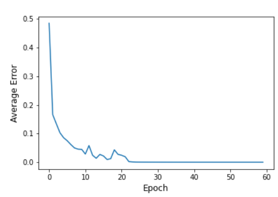
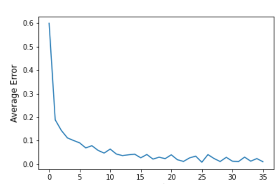
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
256-512-256-128-5		**100%**	**97.92%**
256-512-256-128-64-5		**100%**	97.79%
256-512-256-128-64-32-5		99.68%	97.36%

Results of best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 5 hidden layers (256-512-256-128-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

Confusion matrix	**A C T U A L**

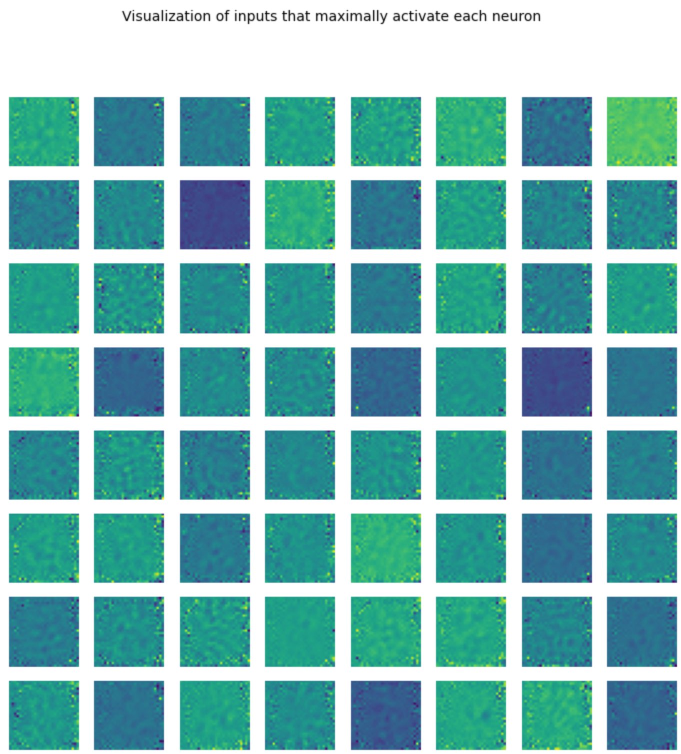
	PREDICTED	751	3	3	1	1
		3	734	11	6	5
		2	10	739	3	5
		2	7	2	747	1
		5	3	9	0	742
Classification accuracy	**Train**		**Validation**		**Test**	
	100%		97.91%		97.83%	

E. Best Reduced Dimension Representation

II. Performance comparison with different models

Architecture	Classification Accuracy on Test data
784-512-256-128-64-10 (Best of Assignment 3)	97.44%
64-512-256-128-64-5 (Best of PCA)	96.17%
64-512-256-128-64-5 (Best of single layer AE)	**98.73%**

III. Weight visualization



2.2. Three hidden layers autoencoder

For three hidden layer autoencoder, we have considered 4 architectures

Architecture	Input layer	Hidden layer 1	Hidden layer 2 (Bottleneck layer)	Hidden layer 3	Output layer
1	784	400	32	400	784
2	784	400	64	400	784
3	784	400	128	400	784
4	784	400	256	400	784

The output layer has 784 neurons since the flattened input has 784 neurons and our task is to reconstruct the input using different autoencoders. The data is normalized using MinMax normalization to scale all input features in the range of [0,1].

Parameters for all models:

Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Logistic
Loss function	Mean Squared Error
Batch size	32
Learning rate (η)	0.001
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}

Convergence criterion	Difference between successive epochs is less than or equal to 10^{-4}.

Further, to check classification performance of reduced representation, encoded representation (output of bottleneck layer) is then used to train different fully connected neural networks. We have done experiments on 3 different FCNN architectures.

Architecture	**Input layer**	**Hidden layer 1**	**Hidden layer 2**	**Hidden layer 3**	**Hidden layer 4**	**Hidden layer 5**	**Output**
A	32/64/128/256	512	256	128	-	-	5
B		512	256	128	64	-	5
C		512	256	128	64	32	5

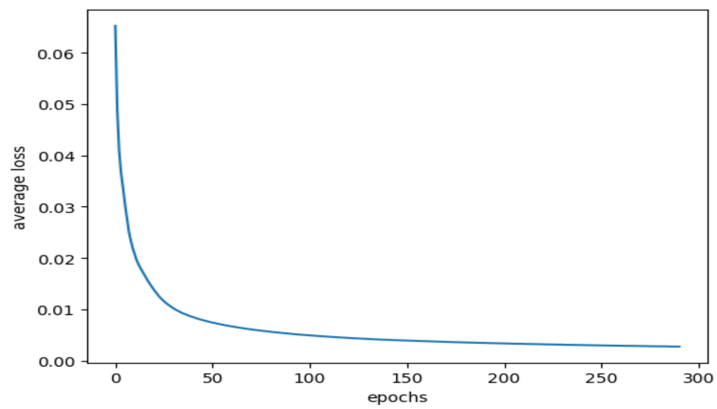
Parameter values for all models:

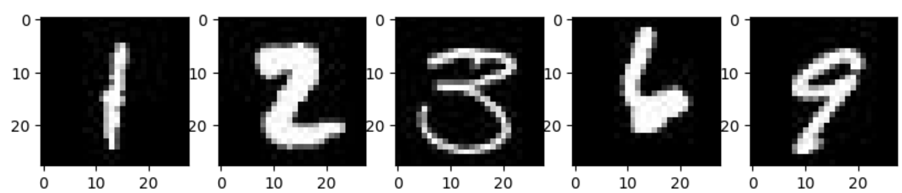
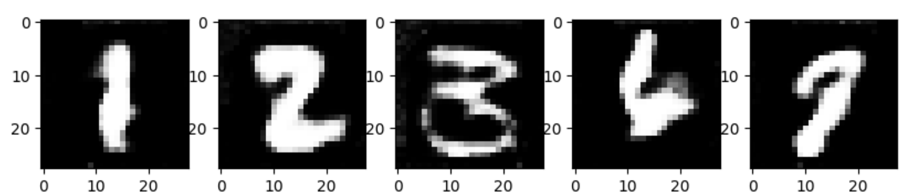
Parameters	**Values**
Hidden layer activation function	Logistic
Output layer activation function	Softmax
Loss function	Cross Entropy
Learning rate (η)	0.001
Batch size	32
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}
Convergence criterion	difference between successive epochs is less than or equal to 10^{-4}.

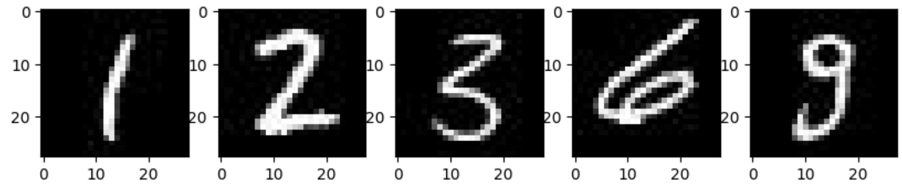
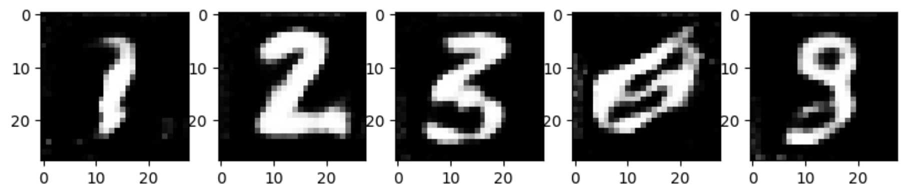
The classification accuracy of each model on the validation set is compared to decide the best architecture for a particular compressed representation. Finally, test accuracy for all best performing models is compared to observe the best reduced representation.

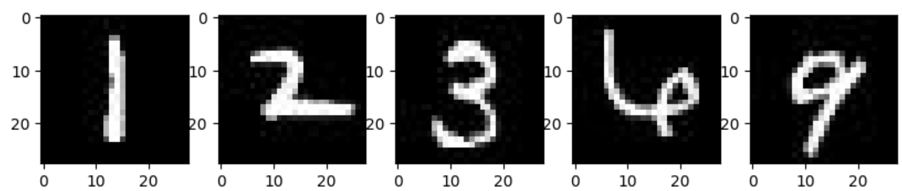
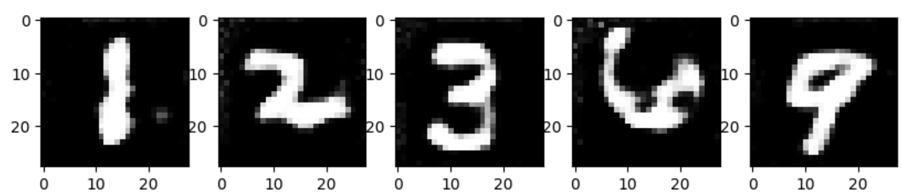
A. Architecture 1 – 784I-400H-32H-400H-784O

Results and Observations of Autoencoder:

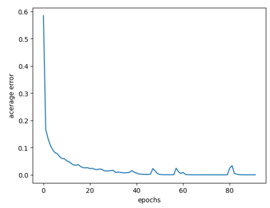
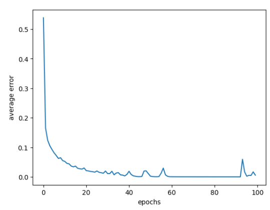
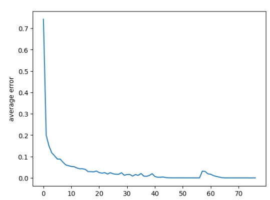
<table>
<tr><td>Error v/s Epoch</td><td colspan="3">
</td></tr>
<tr><td rowspan="2">Mean Squared Error</td><td>Train</td><td>Validation</td><td>Test</td></tr>
<tr><td>0.0077</td><td>0.0108</td><td>0.0093</td></tr>
</table>

<table>
<tr><td colspan="2">Original and reconstructed image of train data</td></tr>
<tr><td>Original Image</td><td></td></tr>
<tr><td>Reconstructed Image</td><td></td></tr>
</table>

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	
Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
32-512-256-128-5		99.25%	98.24%
32-512-256-128-64-5		**99.29%**	**98.36%**
32-512-256-128-64-32-5		99.03%	98.21%

Results on best selected architecture on test data:

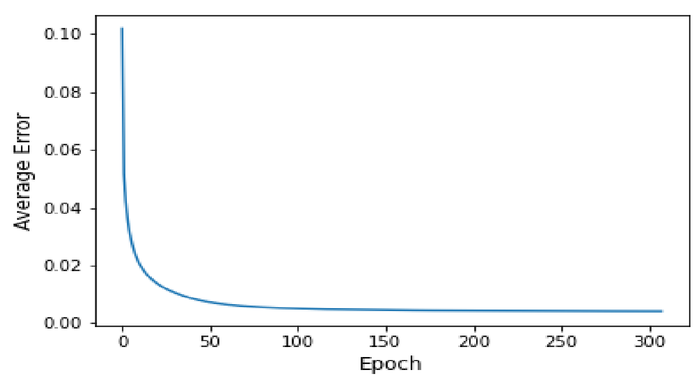
On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (32-512-256-128-64-5) performs best. Following is the confusion matrix and classification accuracy on test data for the best selected model.

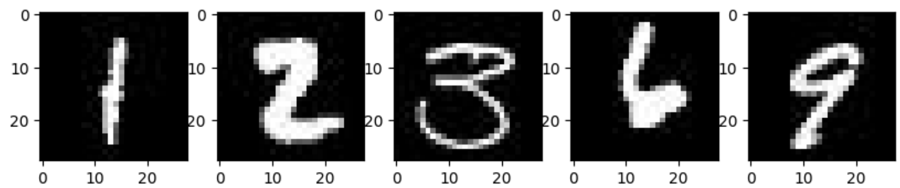
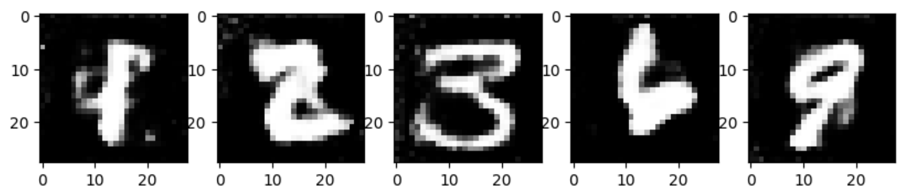
Confusion matrix	ACTUAL					
	PREDICTED	753	3	3	0	0
		4	744	6	0	5
		1	4	748	1	5
		1	5	0	752	1
		2	3	4	0	750

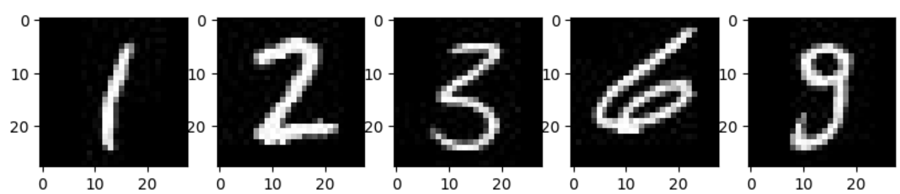
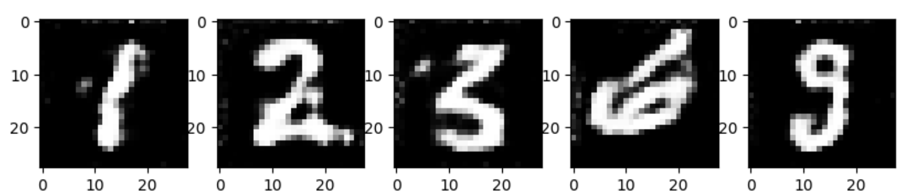
Classification accuracy	Train	Validation	Test
	99.29%	98.36%	98.73%

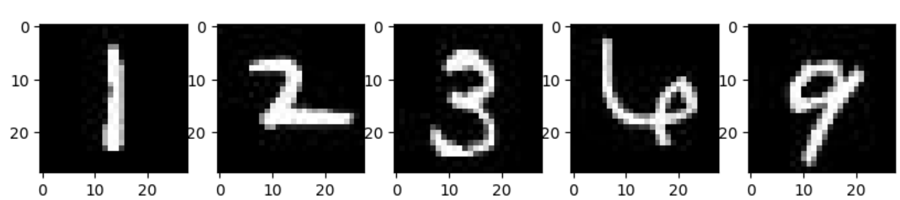
B. Architecture 2 – 784I-400H-64H-400H-784O

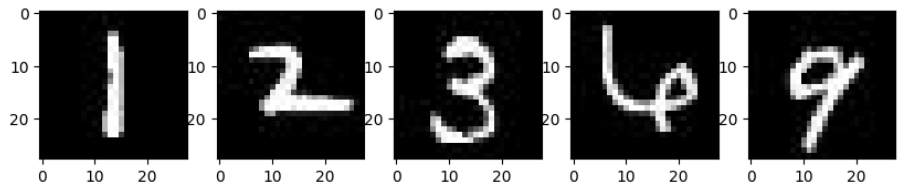
Results and Observations of Autoencoder:

Error v/s Epoch			
Mean Squared Error	**Train**	**Validation**	**Test**
	0.0059	0.0088	0.0070

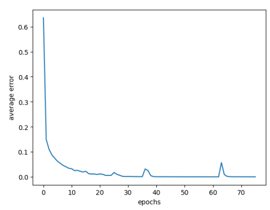
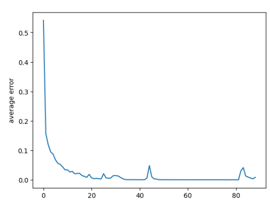
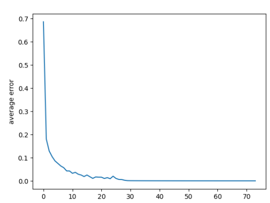
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
64-512-256-128-5		99.33%	98.34%
64-512-256-128-64-5		**99.42%**	**98.36%**
64-512-256-128-64-32-5		99.30%	98.28%

Results on best selected architecture on test data:

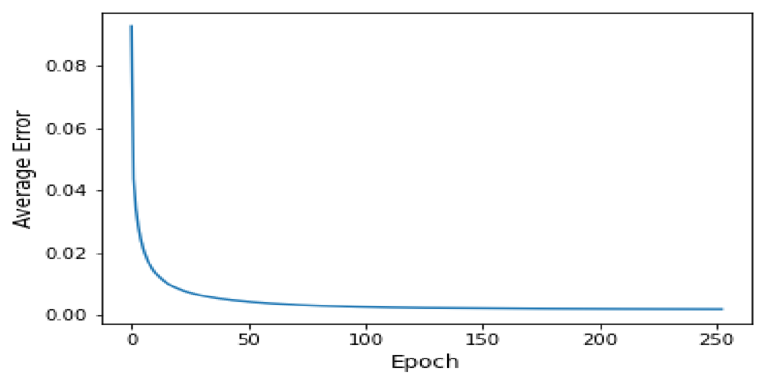
On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (64-512-256-128-64-5) performs best. Following is the confusion matrix on test data and classification accuracy for the best selected model.

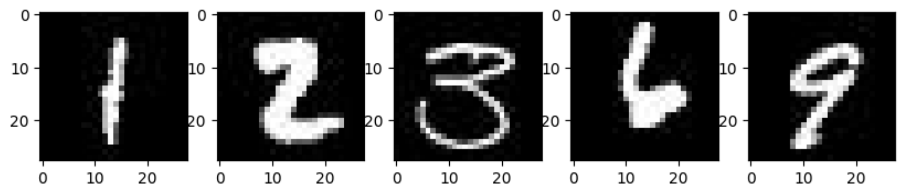
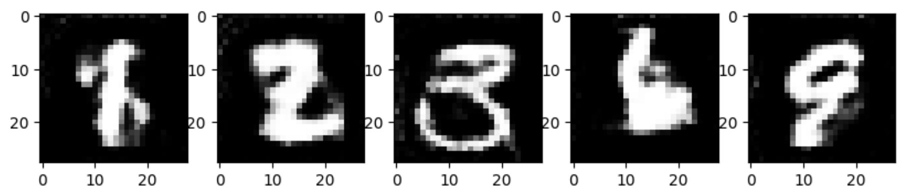
Confusion matrix	A C T U A L

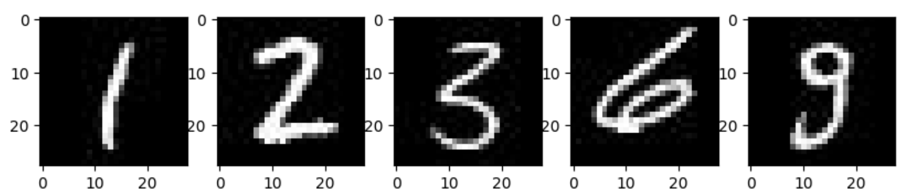
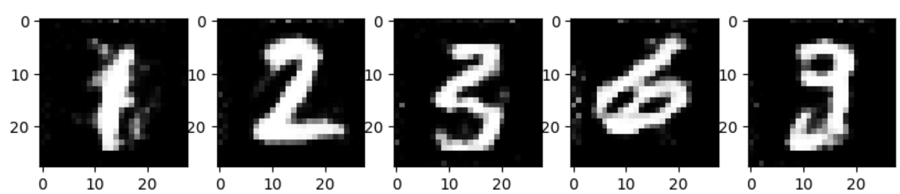
<table>
<tr><td rowspan="5"></td><td rowspan="5">**P R E D I C T E D**</td><td>746</td><td>9</td><td>3</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>739</td><td>11</td><td>1</td><td>7</td></tr>
<tr><td>1</td><td>3</td><td>750</td><td>2</td><td>3</td></tr>
<tr><td>0</td><td>4</td><td>4</td><td>750</td><td>1</td></tr>
<tr><td>4</td><td>2</td><td>19</td><td>1</td><td>733</td></tr>
<tr><td rowspan="2">**Classification accuracy**</td><td colspan="2">**Train**</td><td colspan="2">**Validation**</td><td colspan="2">**Test**</td></tr>
<tr><td colspan="2">99.42%</td><td colspan="2">98.36%</td><td colspan="2">97.97%</td></tr>
</table>

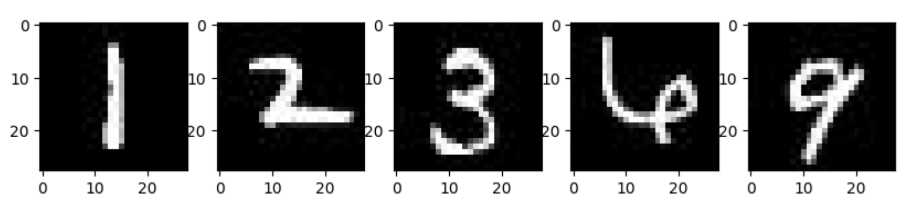
C. Architecture 3 – 784I-400H-128H-400H-784O

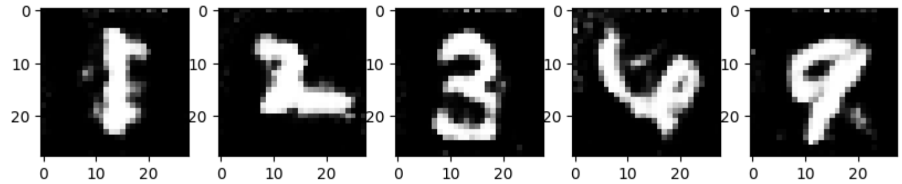
Results and Observations:

<table>
<tr><td>**Error v/s Epoch**</td><td colspan="3">
</td></tr>
<tr><td rowspan="2">**Mean Squared Error**</td><td>**Train**</td><td>**Validation**</td><td>**Test**</td></tr>
<tr><td>0.0051</td><td>0.0078</td><td>0.0057</td></tr>
</table>

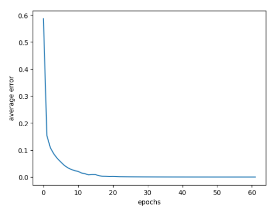
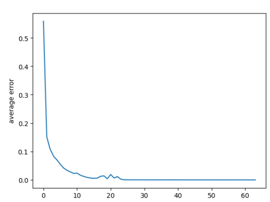
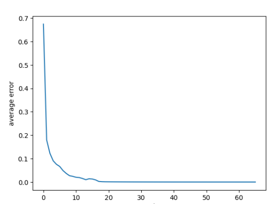
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
128-512-256-128-5		**99.41%**	**98.21%**
128-512-256-128-64-5		99.40%	98.13%
128-512-256-128-64-32-5		**99.41%**	98.11%

Results on best selected architecture on test data:

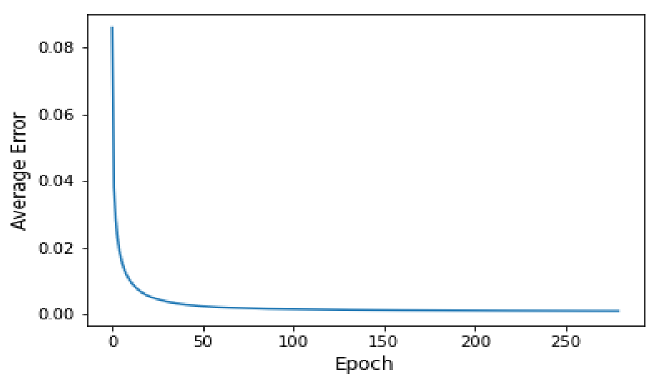
On comparing validation accuracy of different FCNN, we can observe that the network with 3 hidden layers (128-512-256-128-5) performs best. Following is the confusion matrix on test data and classification accuracy for the best selected model.

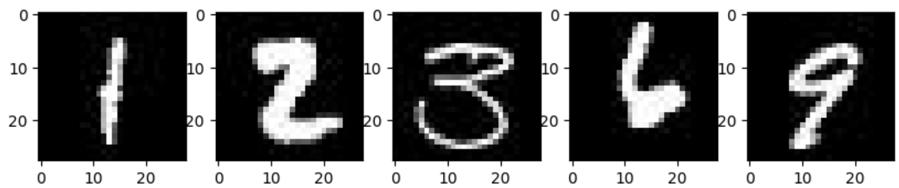
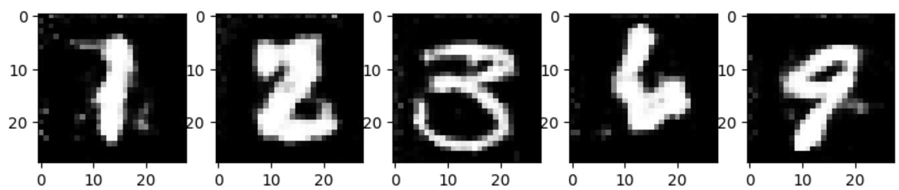
Confusion matrix	**A C T U A L**

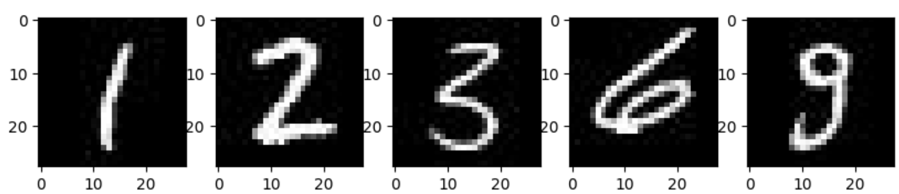
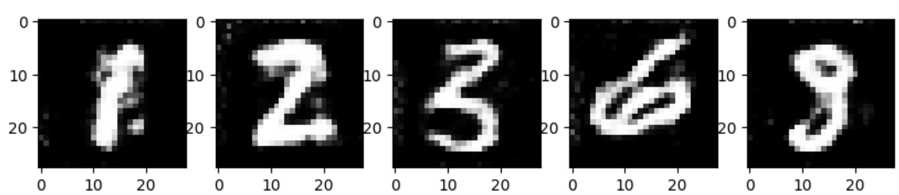
	PREDICTED	750	6	1	2	0
		1	744	8	3	3
		2	5	745	3	4
		0	2	0	756	1
		2	3	8	0	746
Classification accuracy	**Train**		**Validation**		**Test**	
	99.41%		98.21%		98.57%	

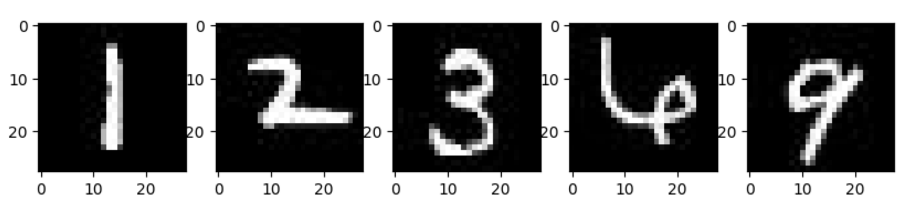
D. Architecture 4 – 784I-400H-256H-400H-784O

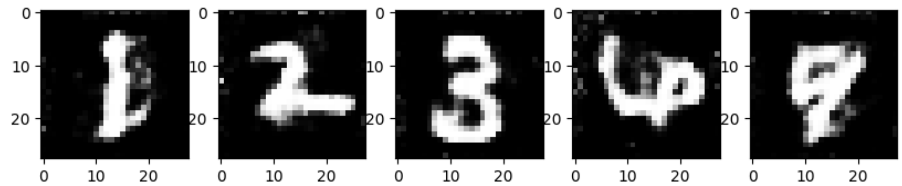
Results and Observations:

Error v/s Epoch	 b		
Mean Squared Error	**Train**	**Validation**	**Test**
	0.0044	0.0070	0.0050

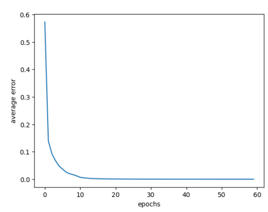
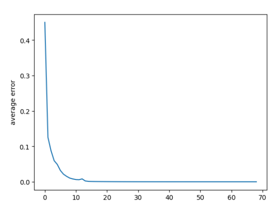
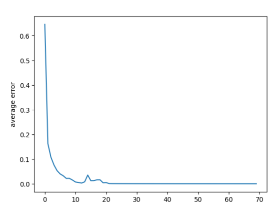
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
256-512-256-128-5		99.50%	98.20%
256-512-256-128-64-5		**99.63%**	98.18%
256-512-256-128-64-32-5		99.52%	**98.25%**

Results on best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 5 hidden layers (256-512-256-128-64-32-5) performs best. Following is the confusion matrix on test data and classification accuracy for the best selected model.

Confusion matrix	**A C T U A L**

	PREDICTED	750	5	3	0	1
		1	740	11	1	6
		3	5	745	2	4
		1	8	1	748	1
		4	5	6	0	744
Classification accuracy	**Train**		**Validation**		**Test**	
	99.52%		98.25%		98.20%	

E. Best Reduced Dimension Representation

IV. Performance comparison with different models

Architecture	Classification Accuracy on Test data
784-512-256-128-64-10 (Best of Assignment 3)	97.44%
64-512-256-128-64-5 (Best of PCA)	96.17%
64-512-256-128-64-5 (Best of single hidden layer AE)	**98.73%**
128-512-256-128-5 (Best of three hidden layers AE)	98.57%

3. Reconstruction and Dimensionality reduction using Denoising Autoencoders

For the single hidden layer denoising autoencoder, we have considered the architecture with best reconstruction from our experiments with different autoencoders. The single hidden layer autoencoder with best classification performance on test data is considered to be giving best input representation among all.

Architecture	Input layer	Hidden layer (Bottleneck layer)	Output layer
1	784	64	784

The output layer has 784 neurons since the flattened input has 784 neurons and our task is to reconstruct the input using the denoising autoencoders with 20% and 40% noise. The data is normalized using MinMax normalization to scale all input features in the range of [0,1].

Parameters for all models:

Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Logistic
Loss function	Mean Squared Error
Batch size	32
Learning rate (η)	0.001
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}

Convergence criterion	Difference between successive epochs is less than or equal to 10^{-4}.

Further, to check classification performance of reduced representation, encoded representation (output of bottleneck layer) is then used to train different fully connected neural networks. We have done experiments on 3 different FCNN architectures.

Architecture	Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3	Hidden layer 4	Hidden layer 5	Output
A	32/64/128/256	512	256	128	-	-	5
B		512	256	128	64	-	5
C		512	256	128	64	32	5

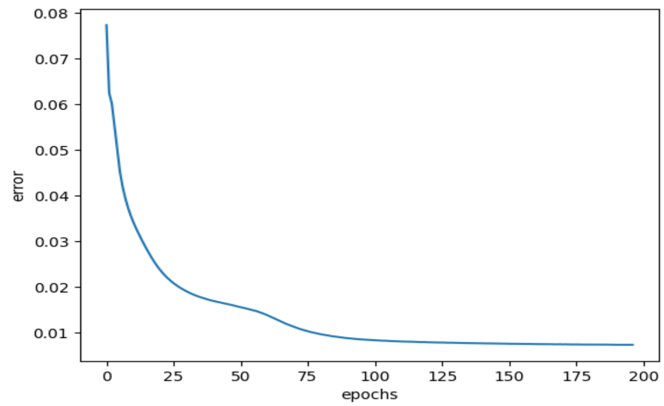
Parameter values for all models:

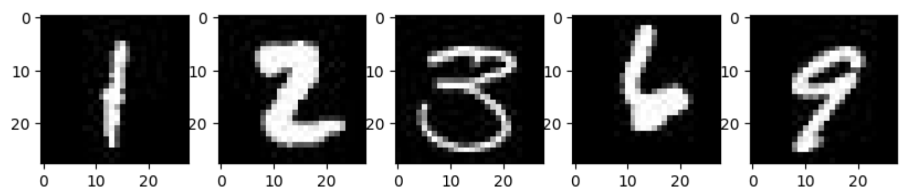
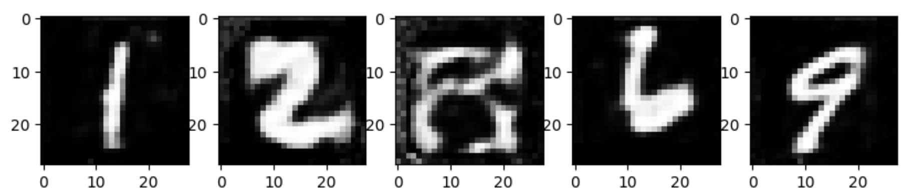
Parameters	Values
Hidden layer activation function	Logistic
Output layer activation function	Softmax
Loss function	Cross Entropy
Learning rate (η)	0.001
Batch size	32
Optimiser	Adam
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.999
Epsilon (ϵ)	10^{-8}
Convergence criterion	difference between successive epochs is less than or equal to 10^{-4}.

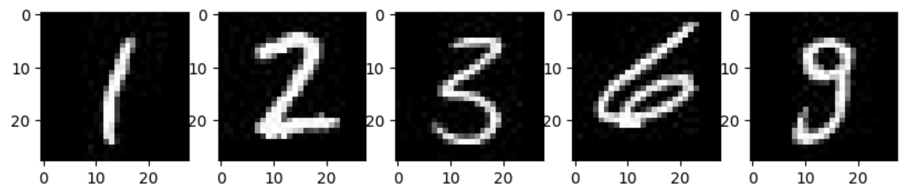
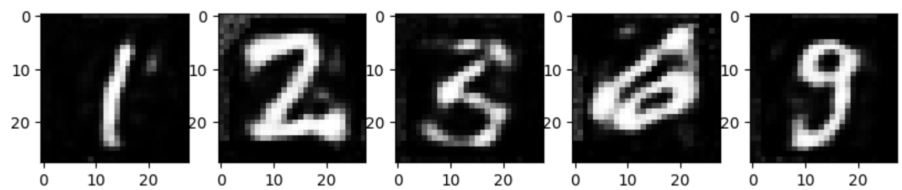
The classification accuracy of each model on the validation set is compared to decide the best architecture for a particular compressed representation. Finally, test accuracy for all best performing models is compared to observe the best reduced representation.

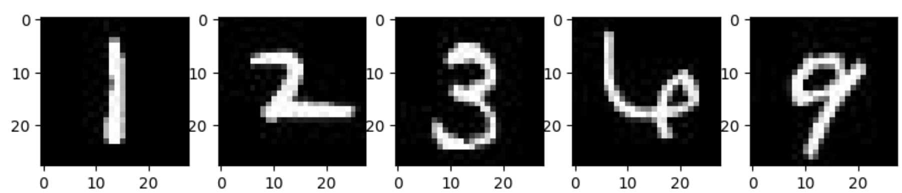
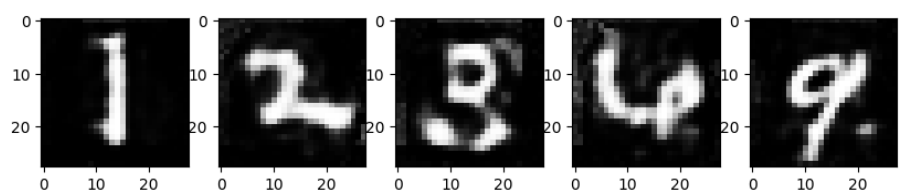
3.1. 20% noise

Results and Observations of Denoising Autoencoder:

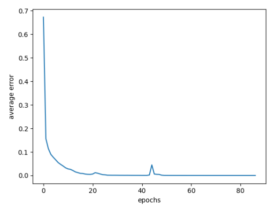
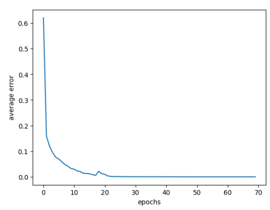
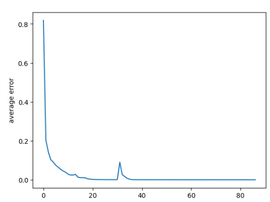
Error v/s Epoch			
Mean Squared Error	**Train**	**Validation**	**Test**
	0.0050	0.00592	0.00596

Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	
Reconstructed Image	

Results and Observations of FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
64-512-256-128-5		**99.48%**	98.38%
64-512-256-128-64-5		99.36%	**98.39%**
64-512-256-128-64-32-5		99.39%	98.34%

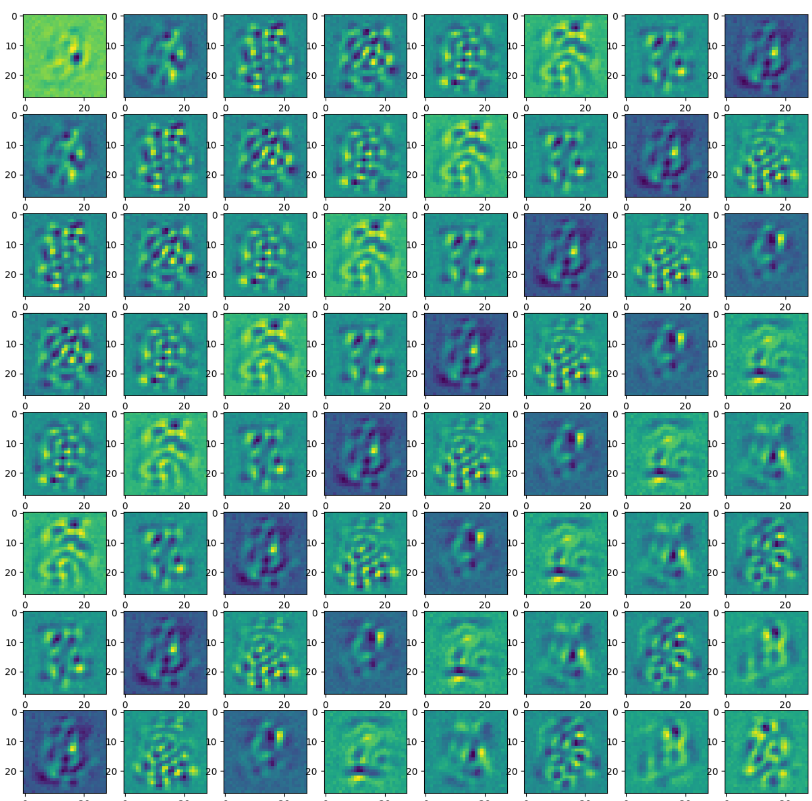
Results of best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 4 hidden layers (64-512-256-128-64-5) performs best. Following is the confusion matrix on test data and classification accuracy for the best selected model.

Confusion matrix	ACTUAL					
	PREDICTED	753	4	1	1	0
		4	741	6	1	7
		0	4	750	1	4
		2	3	0	754	0
		5	1	7	1	745

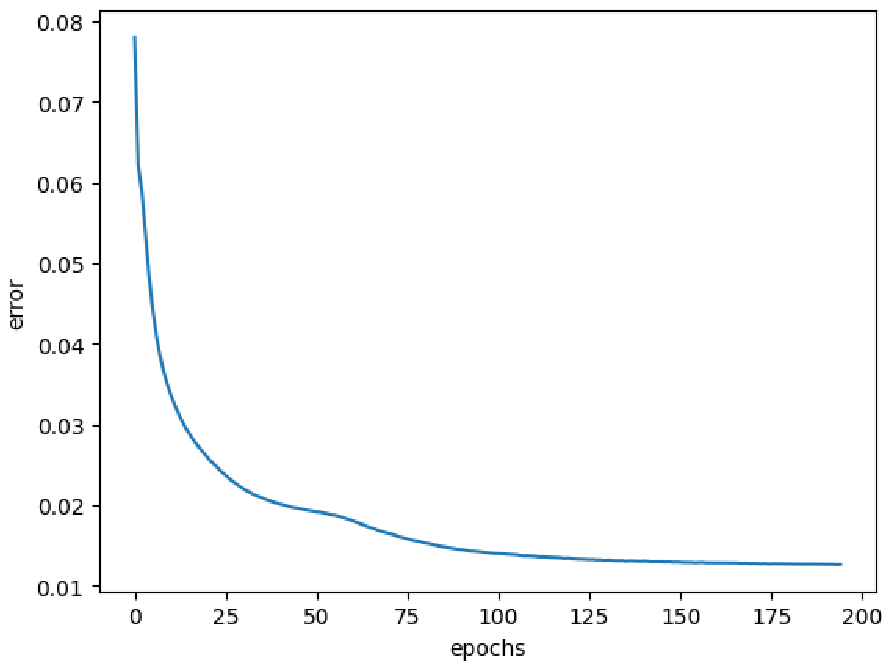
Classification accuracy	Train	Validation	Test
	99.36%	99.39%	98.69%

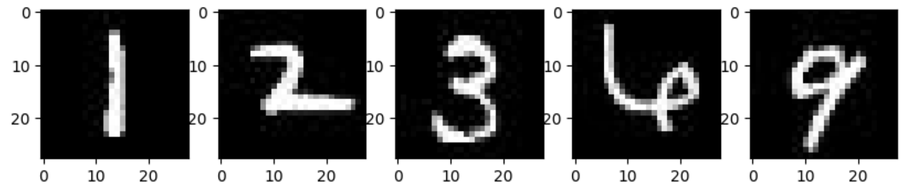
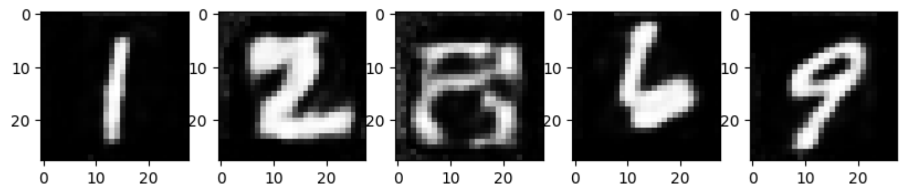
Weight visualization:

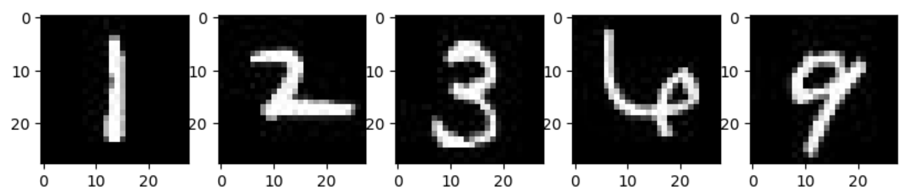
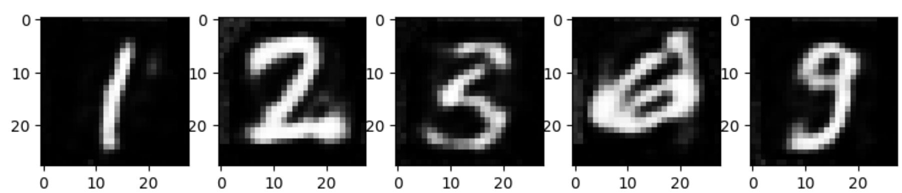


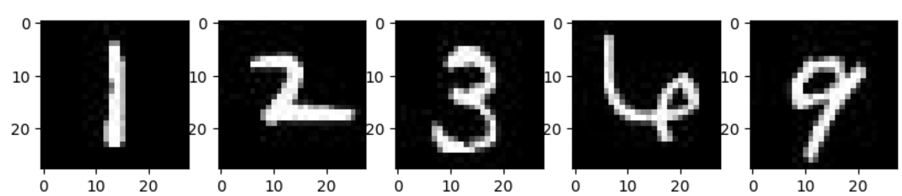
3.2. 40% noise

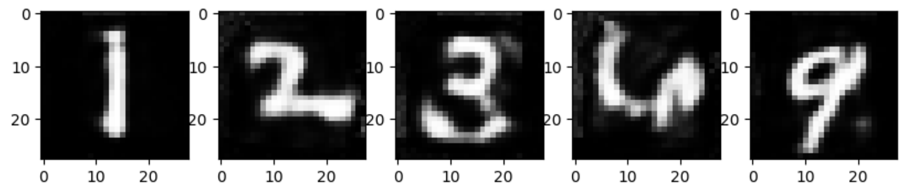
Results and Observations of Denoising Autoencoder:

Error v/s Epoch			
Mean Squared Error	**Train**	**Validation**	**Test**
	0.0066	0.0074	0.0074

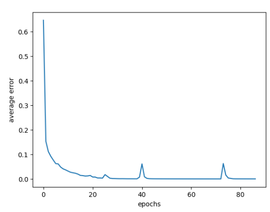
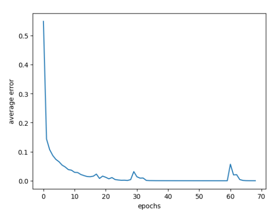
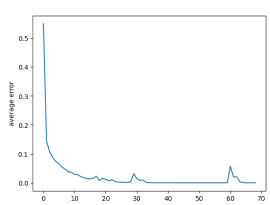
Original and reconstructed image of train data	
Original Image	
Reconstructed Image	

Original and reconstructed image of validation data	
Original Image	
Reconstructed Image	

Original and reconstructed image of test data	
Original Image	

Reconstructed Image	

Results and Observations o FCNNs:

Comparison of various model performance			
Architecture	**Error vs epoch**	**Training accuracy**	**Validation Accuracy**
64-512-256-128-5		**99.40%**	**98.43%**
64-512-256-128-64-5		99.31%	98.32%
64-512-256-128-64-32-5		99.31%	98.32%

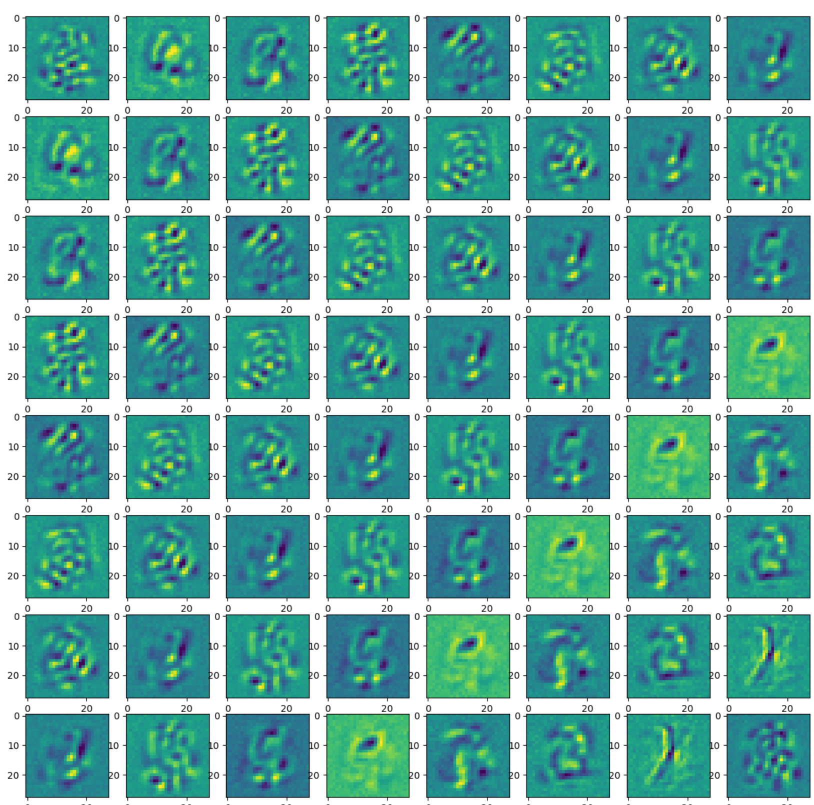
Results on best selected architecture on test data:

On comparing validation accuracy of different FCNN, we can observe that the network with 3 hidden layers (64-512-256-128-5) performs best. Following is the confusion matrix on test data and classification accuracy for the best selected model.

Confusion matrix	ACTUAL					
	PREDICTED	751	5	2	0	1
		3	741	8	1	6
		1	6	747	1	4
		1	4	1	750	3
		3	2	5	0	749

Classification accuracy	Train	Validation	Test
	99.40%	98.43%	98.49%

Weight visualization:



3. Inferences and Conclusion

In this assignment, we performed different experiments to understand the role of autoencoders in reconstruction, dimensionality reduction and classification of input patterns. We did dimensionality reduction using PCA as well as single and three hidden layer autoencoders. This compressed input was then sent to a classifier neural network to perform classification. We also did similar tasks with denoising autoencoders. The inferences and conclusions that we can draw from our experiments are as follows:

- The measure we have used to compare reconstruction in case of autoencoders is the Mean Squared Error (MSE). If we compare the MSE for the same dimension in single and three hidden layer autoencoders, in most cases, single hidden layer autoencoders have lesser MSE values. Also, in some cases, the difference in MSEs of both autoencoders is ≈ 0.005.
- The denoising autoencoders have comparatively larger MSEs than that of a standard autoencoder with the same architecture because it is trained to reconstruct clean inputs from noisy inputs and has an additional task to remove noise from inputs while learning.
- On observing the classification accuracy on train, validation and test data, it can be noted that, in all cases, single hidden layer autoencoder has 100% test accuracy while three hidden layer autoencoder has accuracy $> 99\%$. However, on validation and test data, the performance of both autoencoders is highly competitive.
- Principal component analysis technique also works well at classification with accuracy $> 95\%$ on validation as well as test data. It is possible that with PCA, compressed representation is not able to retrieve all the relevant information required to predict output labels.
- It is evident that denoising autoencoders, although have higher MSEs, their compressed representation performs really well at classification with accuracy $> 99\%$ on validation data. The reason might be since denoising acts as a regularization, denoising autoencoder's compressed representation is able to keep more relevant information about the inputs that is required to predict the output labels.
- The weight visualization also depicts that the hidden representation in denoising autoencoder has learnt more strokes than in case of single hidden layer autoencoders.

We saw that autoencoders are a very good way to reconstruct input patterns as well as reduce its dimension. From our current as well as previous experiments, one can conclude that using an autoencoder to reduce dimension and retain useful information in inputs to further perform the task at hand can improve our results as well as decrease our training time. Also, adding regularization, like adding noise, can make our model more robust to overfitting.

References

1. Bengio, Y., Courville, A., & Goodfellow, I. (2016). *Deep Learning*. MIT Press.