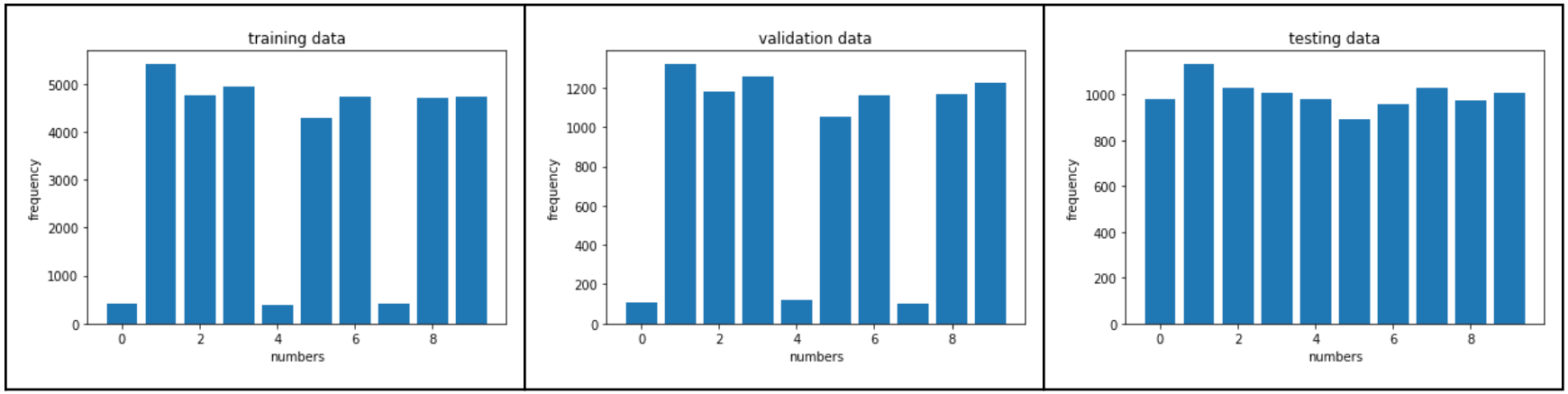
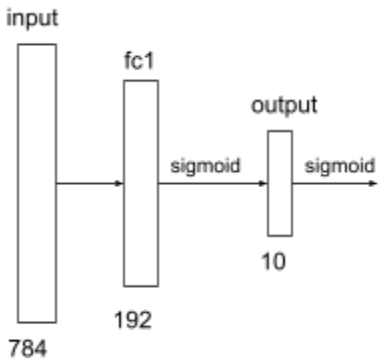
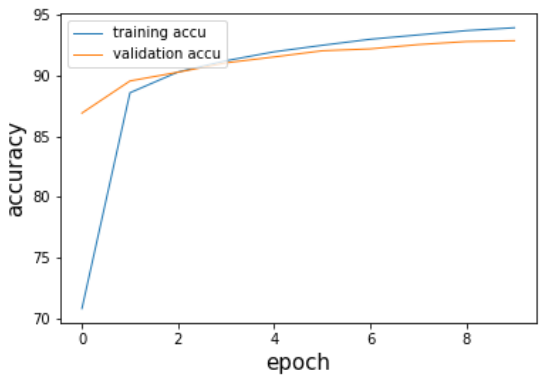
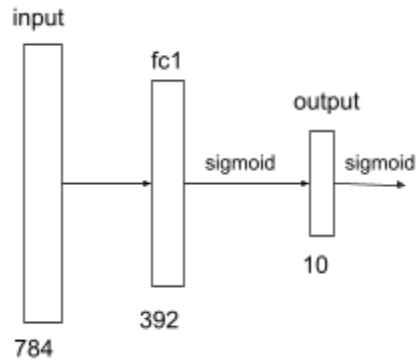
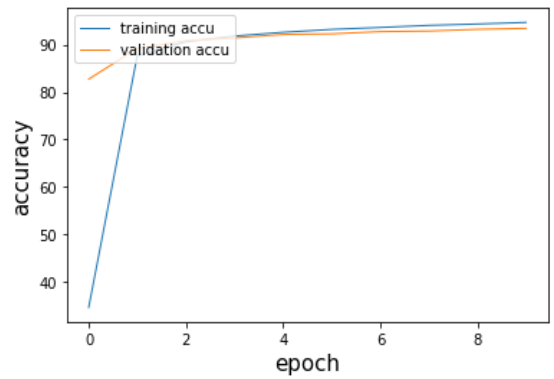
Task 1

Data Distribution:



Hyperparameter settings:

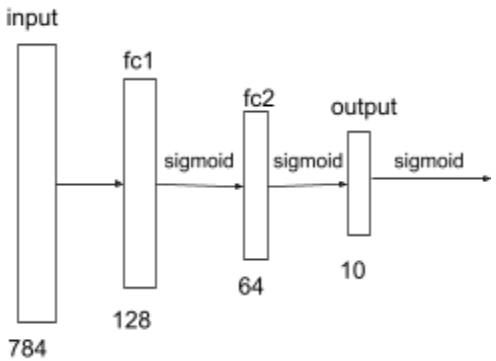
1. **Architecture:**

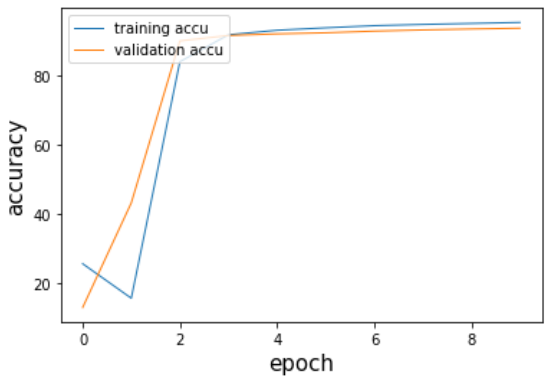
Hyperparameters						
Lr	**Optimizer**	**Loss**	**Epochs**	**Val % of train data**	**Training BS**	**Dropout**
0.01	Adam	Cross Entropy	10	0.2	1000	0
Arch.	Architecture					
1	  **Max training accuracy: 93.9** **Max validation accuracy: 92.8** **Max testing accuracy: 67.8**	  **Max training accuracy: 94.6** **Max validation accuracy: 93.4** **Max testing accuracy: 68.1**				

<table>
<tr><td></td><td colspan="2">

- By adding more neurons the accuracy improved by 1% and the model started becoming overfit at 3 epoch with 90% acc.

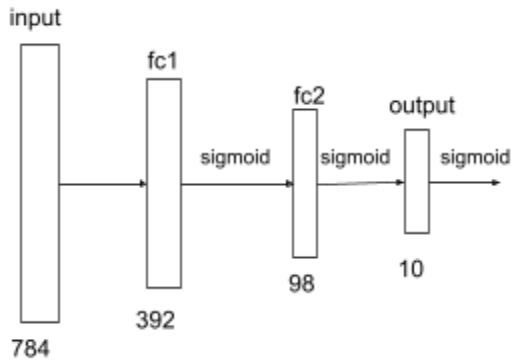
</td></tr>
<tr><td>2</td><td>

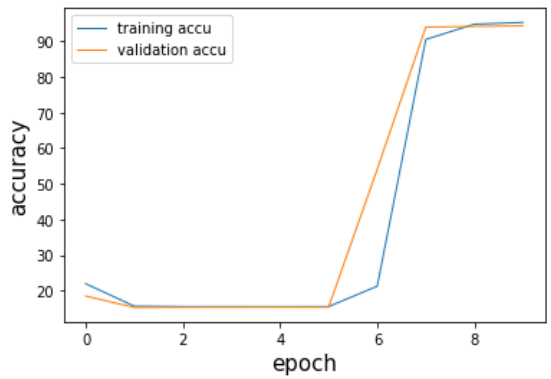




Max training accuracy: 95.3
Max validation accuracy: 93.6
Max testing accuracy: 68.4

</td><td>



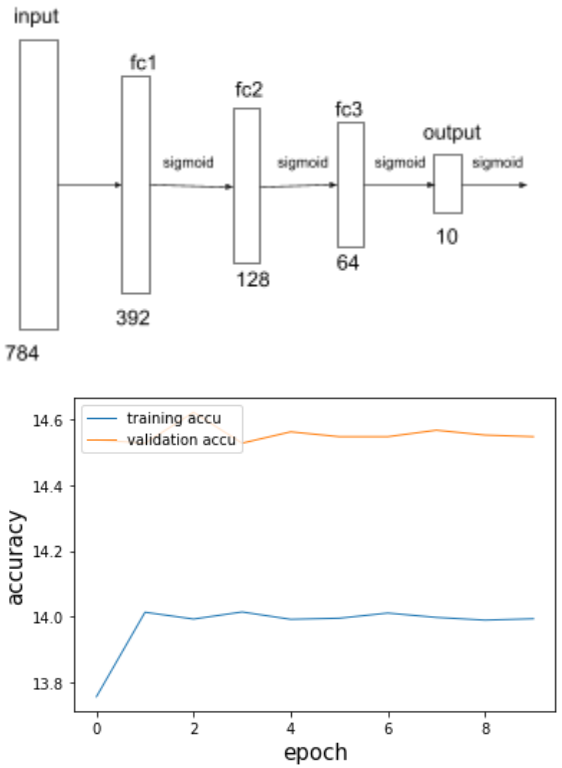
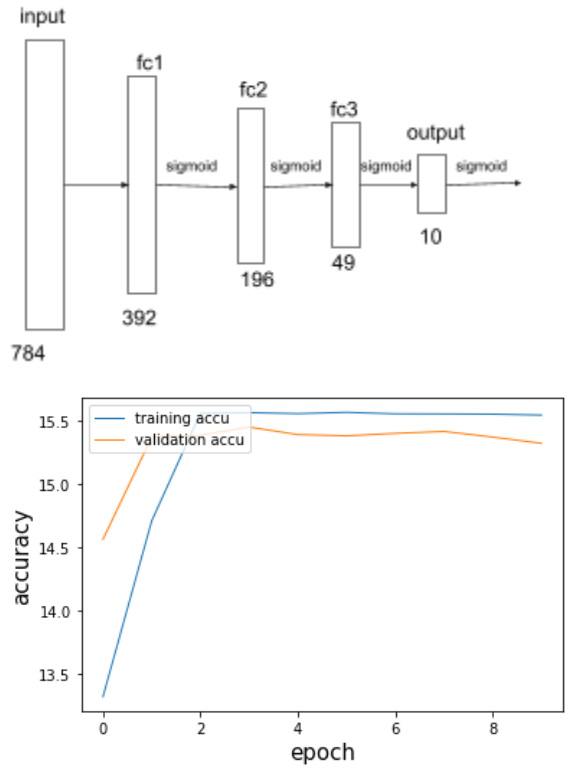


Max training accuracy: 95.2
Max validation accuracy: 94.3
Max testing accuracy: 60.5

</td></tr>
<tr><td></td><td colspan="2">

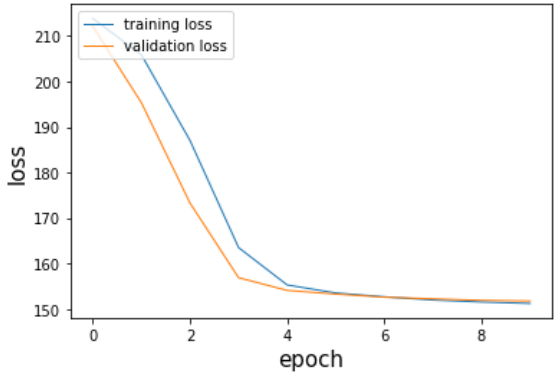
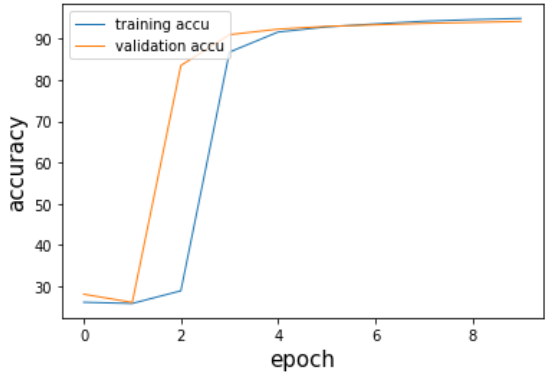
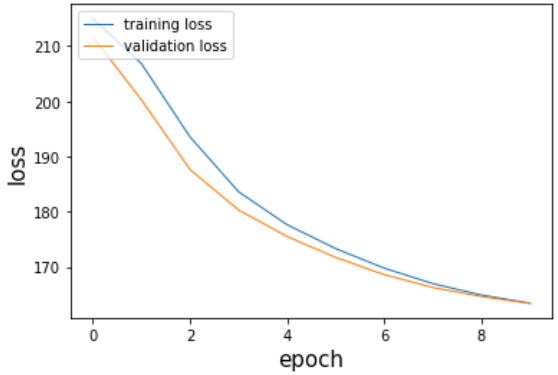
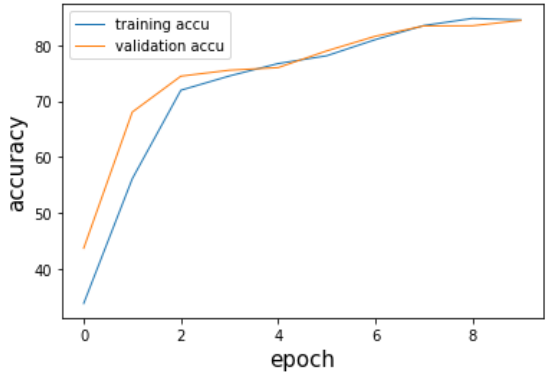
- There is an inadequate increase in the accuracy by increasing the number of neurons in the hidden layers but the training and computation time in high so selecting the left model.

</td></tr>
</table>

3		
	● The loss did not decrease in both cases and accuracy remain at the same point because I am using sigmoid 3 times consecutive due to which the gradient becomes very small and the model is unable to learn. Because the images contain 0 value at maximum pixels and the sigmoid of 0 is 0. ● I can try using Relu in some layers, especially with the first layer which might help. ● I am selecting the model with 3 hidden layers = [784, 128, 64, 10] because it overfits at 92% accuracy, whereas models with 2 hidden layers overfit at 90%. And with more than 3 layers the model is unable to learn.	

2. Learning Rate

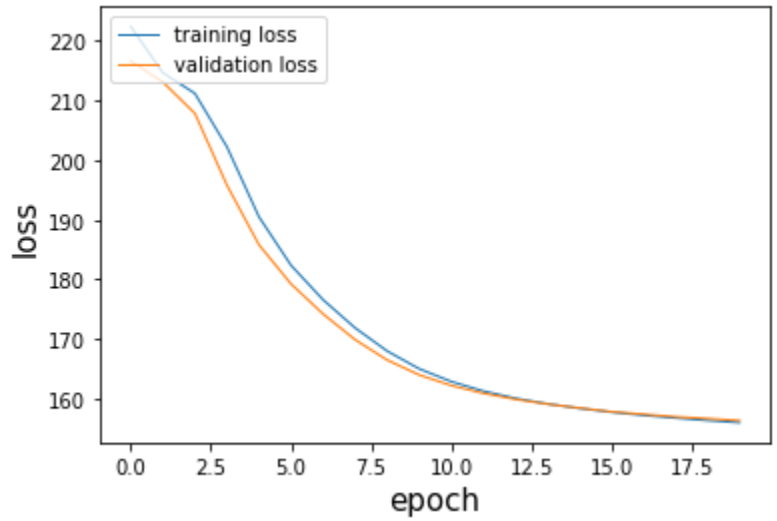
Architecture = [784,128, 64, 10], all other parameters are same.

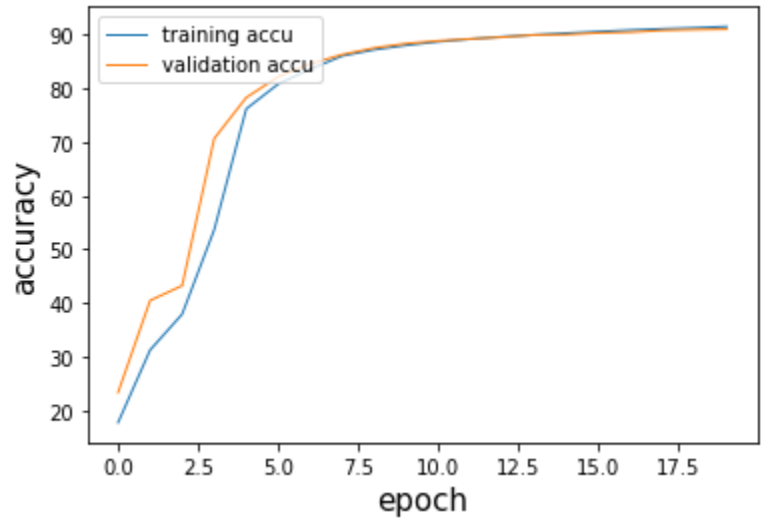
Lr = 0.01	Lr = 0.001
 	 
Max training accuracy: 94.9 **Max validation accuracy: 94.2** **Max testing accuracy: 68.5**	**Max training accuracy: 84.5** **Max validation accuracy: 84.4** **Max testing accuracy: 62.1**

I am using the LR = 0.001 because that would help to achieve better accuracy before overfitting.

3. Epochs

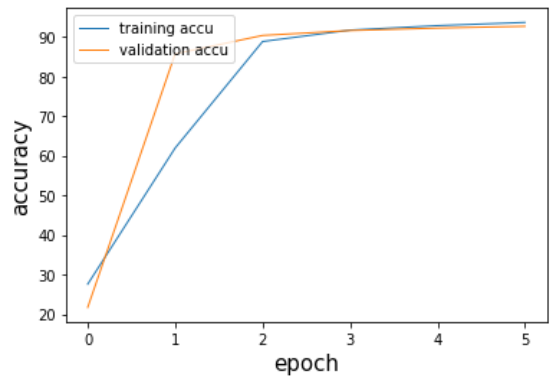
Lr = 0.001, Epochs = 20, Architecture = [784,128, 64, 10], all other parameters are same.

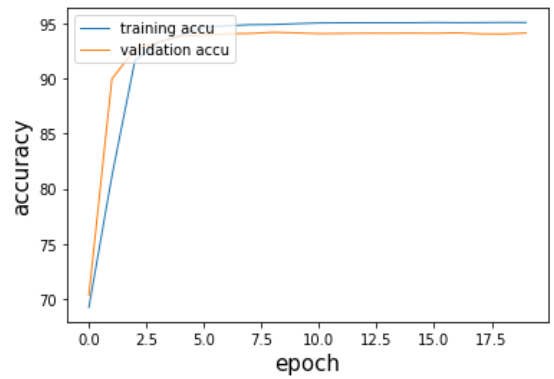


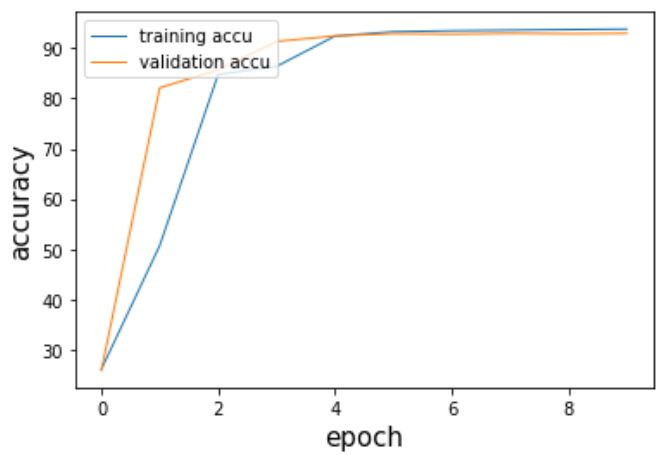


The model overfits at 90% accuracy so, then I would use LR = 0.01 and try using other techniques like learning rate decay rather than using the same learning rate for the whole training or the dropout. And use the epochs = 20

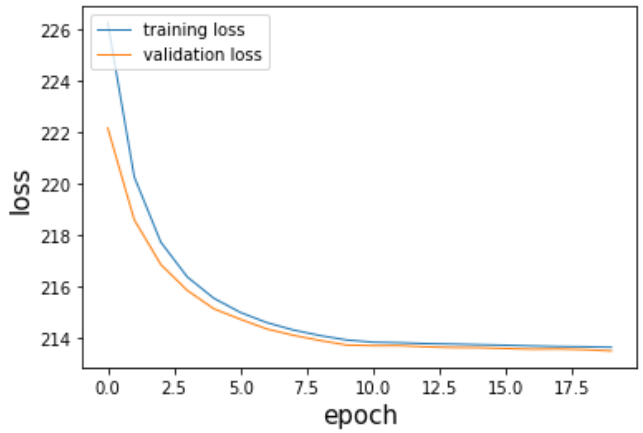
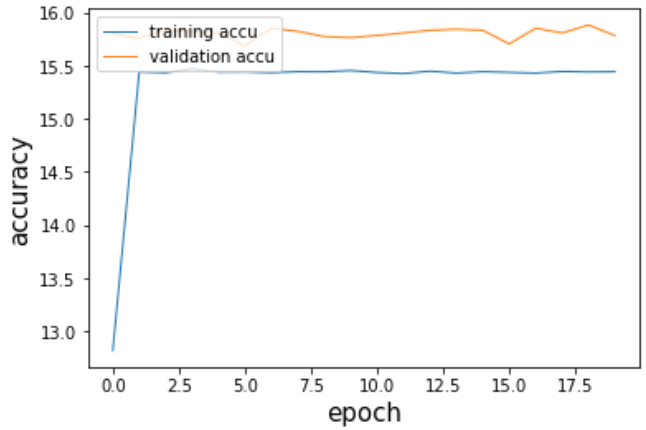
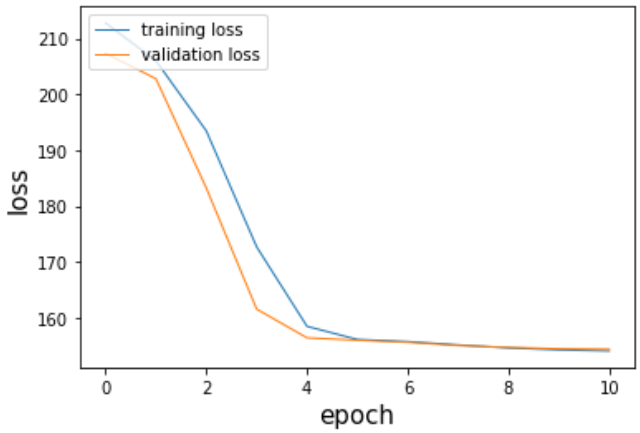
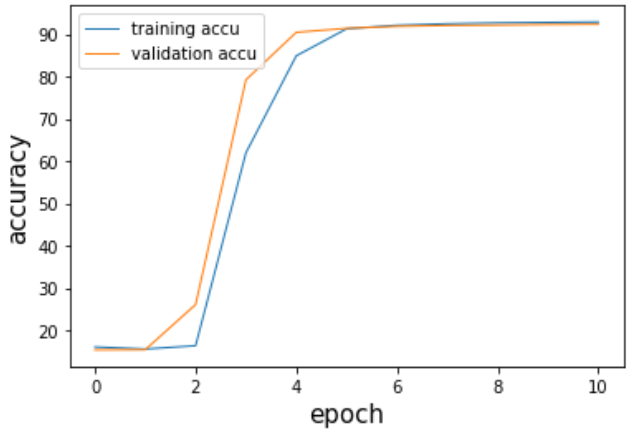
4. Early stopping and Learning Rate Decay

<table>
<tr><td colspan="2">Lr = 0.01, Epochs = 20, Architecture = [784,128, 64, 10], all other parameters are same.</td></tr>
<tr><td>Early Stopping, N = 3</td><td>StepLR with step_size = 5 and gamma = 0.1
the lr divides by 10 after each 5 iterations</td></tr>
<tr><td> 

The training stops after 6 epochs with 93.6 training accuracy and 92.6 validation accuracy.</td><td> 

With schedular(), the model achieved the accuracy of 95%.</td></tr>
<tr><td> 
</td><td colspan="1">Starting Lr = 0.01, Epochs = 20 With N = 5 for early stopping, step_size = 5, gamma = 0.1 for StepLR.
The training stopped after 10 epochs with accuracy of 93.8%.</td></tr>
</table>

5. Optimizers

SGD, Starting Lr = 0.1, Epochs = 20 With N = 5 for early stopping, step_size = 10, gamma = 0.2 for StepLR.	Adam, Starting Lr = 0.01, Epochs = 20 With N = 5 for early stopping, step_size = 5, gamma = 0.1 for StepLR.
 	 

- With Adam the loss reached to 160 and accuracy to 90% but with SGD the loss reached 214 and accuracy remain 15%.
- SGD training was very slow even with the higher starting learning rate and step_size for StepLR and lower gamma value.

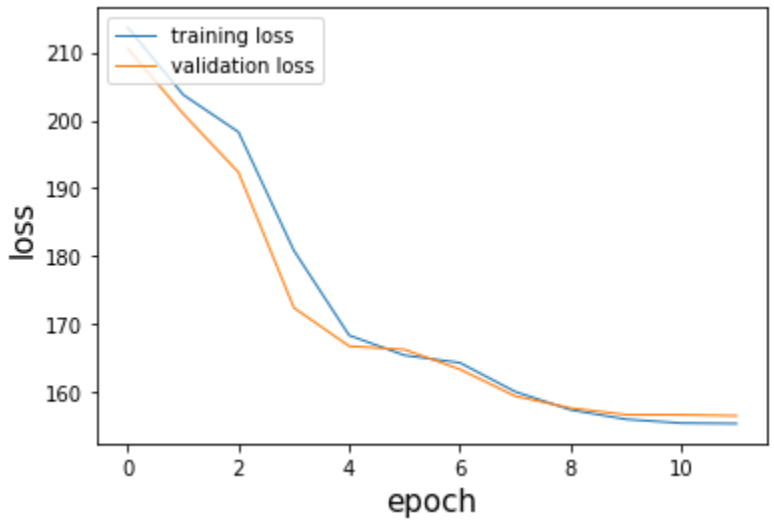
Best model Results:

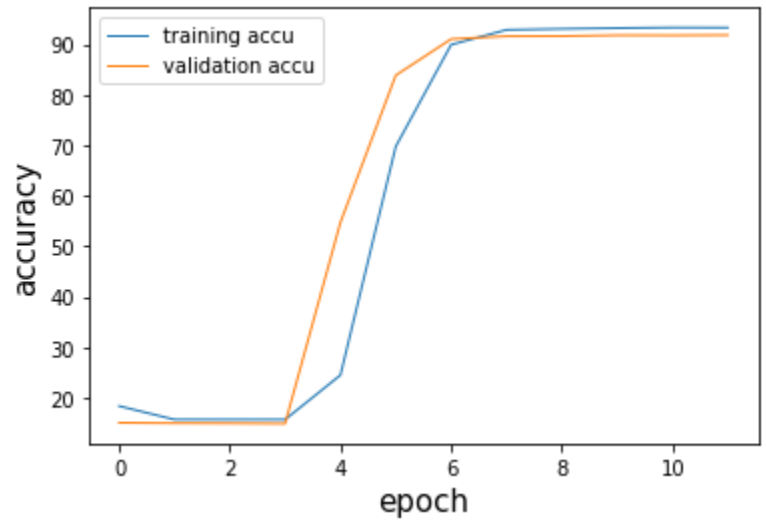
Hyperparameters settings:

Model = [784,128, 64, 10], loss_func = nn.CrossEntropyLoss, optimizer = optim.Adam

Lr = 0.01, Epochs = 20 With N = 5 for early stopping, step_size = 5, gamma = 0.1 for StepLR.

1. Accuracy & Loss of test data





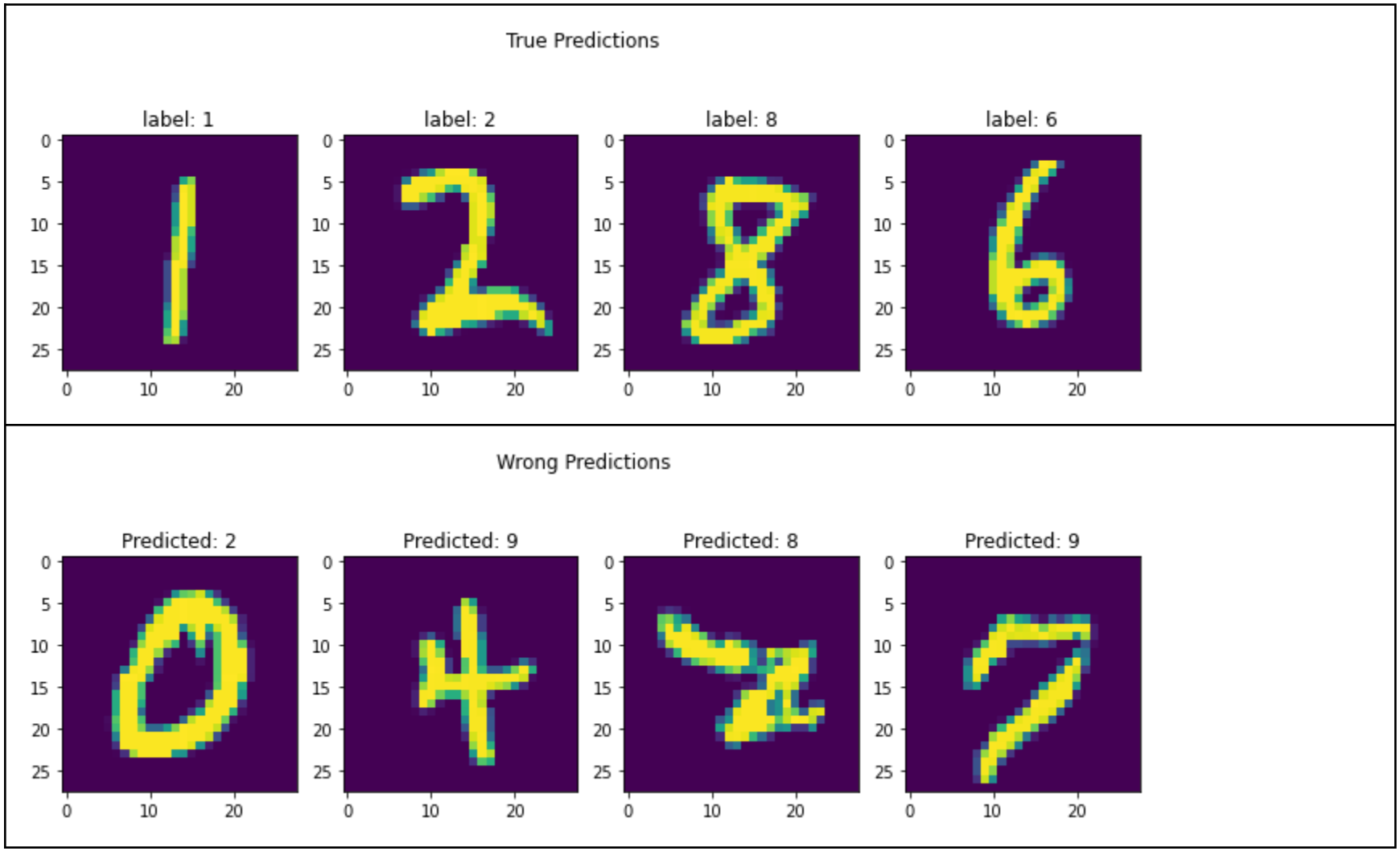
Training Accuracy: 93.3
Validation Accuracy: 91.8
Testing Accuracy: 67.3

2. Confusion matrix

3. Precision, Recall and F1_score

	precision	recall	f1-score	support
0	0.000	0.000	0.000	980
1	0.964	0.980	0.972	1135
2	0.787	0.947	0.860	1032
3	0.775	0.944	0.851	1010
4	0.000	0.000	0.000	982
5	0.710	0.943	0.810	892
6	0.631	0.977	0.767	958
7	0.000	0.000	0.000	1028
8	0.859	0.954	0.904	974
9	0.372	0.969	0.538	1008
accuracy			0.673	9999
macro avg	0.510	0.671	0.570	9999
weighted avg	0.514	0.673	0.573	9999

4. 4 correct and 4 wrong predictions



TASK 2: Hyperparameter settings:

1. Architecture:

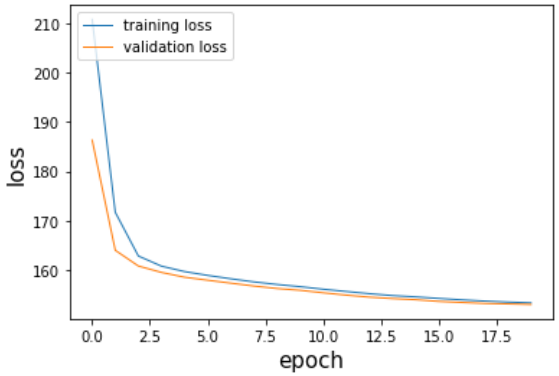
Hyper parameter					
Lr	**Optimizer**	**Loss**	**Epochs**	**Val % of train data**	**N for early stopping**
0.001	Adam	Cross Entropy	10	0.2	5

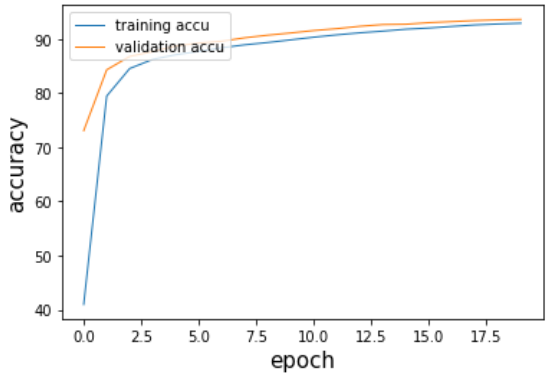
- In all these architechures ReLU activation is used in Convolution layers and Sigmoid in FC layers.

Arch. No.	Conv layers + FC layers	Architecture
1	2+1	

in_channels	out_channels	kernel_size	stride	output
1	16	5	2	12x12
16	8	3	1	10x10

in_dim	out_dim
800	10



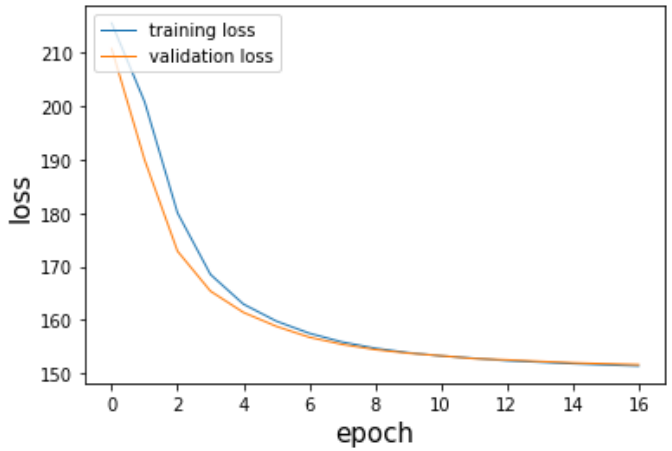


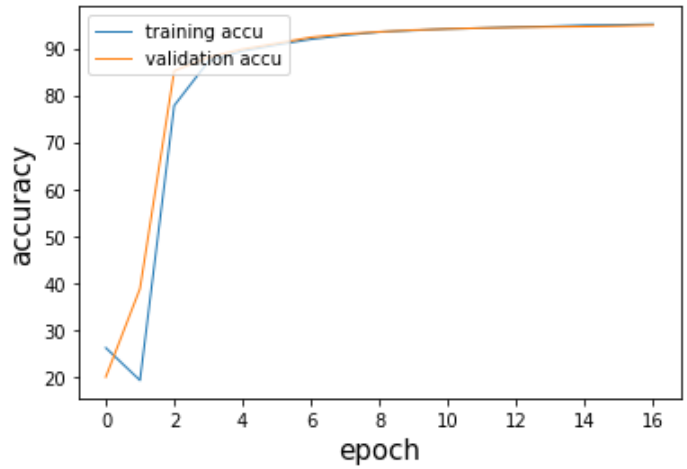
Training Accuracy: 92.1
Validation Accuracy: 93.6
Testing Accuracy: 67.9

2	3+2	

in_channels	out_channels	kernel_size	stride	output
1	32	5	2	12x12
32	16	3	1	10x10
16	8	3	1	8x8

in_dim	out_dim
512	256
256	10



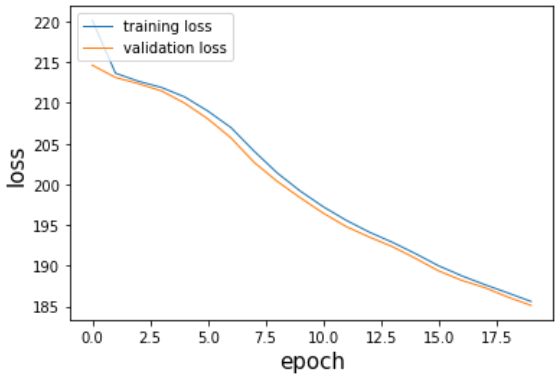


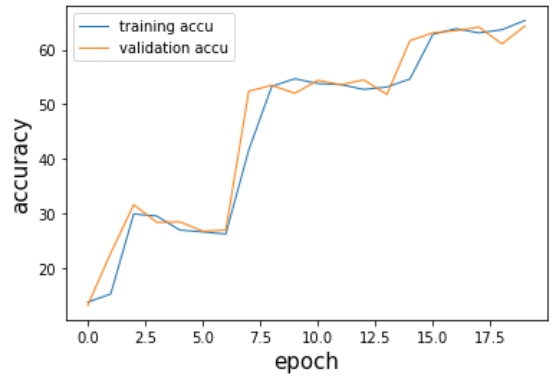
Training Accuracy: 95.1
Validation Accuracy: 94.9
Testing Accuracy: 69.0

3	4+2	

in_channels	out_channels	kernel_size	stride	output
1	32	5	2	12x12
32	16	3	1	10x10
16	8	3	1	8x8
8	4	3	1	6x6

in_dim	out_dim
144	72
72	10



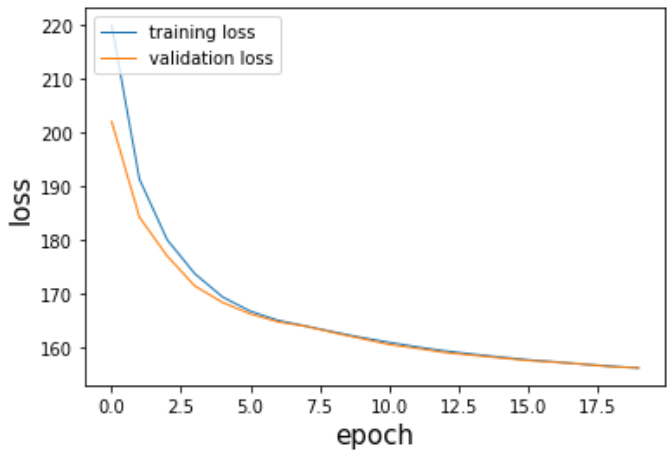


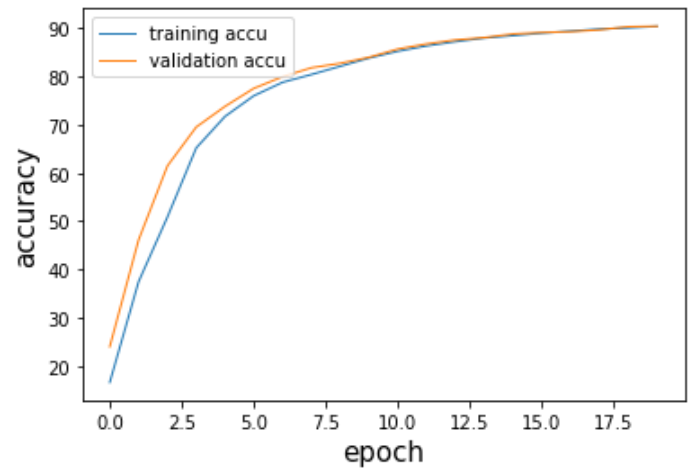
Training Accuracy: 65.4
Validation Accuracy: 64.4
Testing Accuracy: 47.0

4	5+1	

in_channels	out_channels	kernel_size	stride	output
1	32	5	2	12x12
32	16	3	1	10x10
16	8	3	1	8x8
8	4	3	1	6x6
4	2	3	1	4x4

in_dim	out_dim
32	10

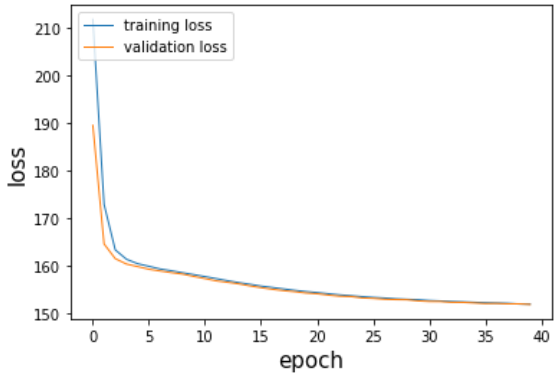
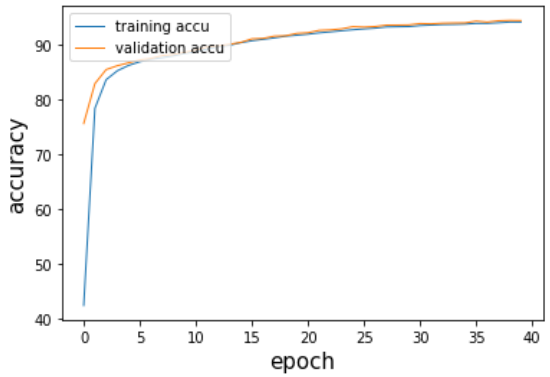
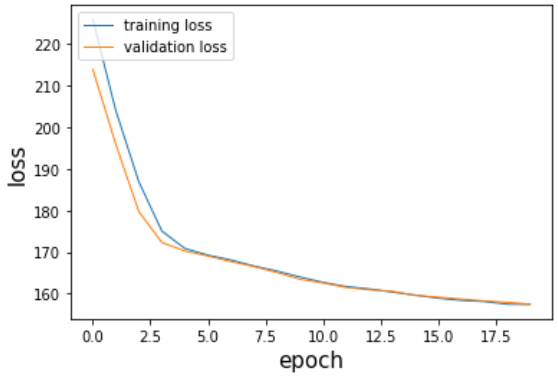
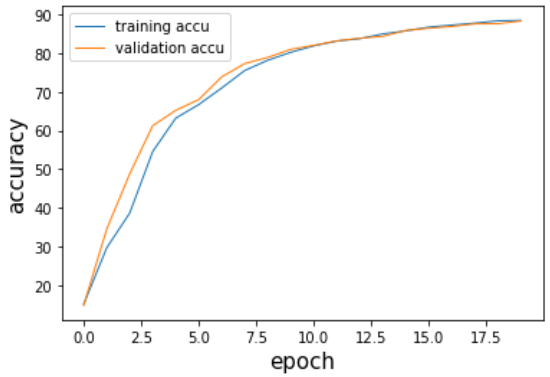




Training Accuracy: 90.3
Validation Accuracy: 90.3
Testing Accuracy: 65.8

- Architecture 1 has not overfitted so, it can be trained for some more epochs.
- Architecture 2 stopped after 16 epochs because of early stopping.
- Architecture 3 is not fully trained.
- Architecture 4 is fully trained.

2. Epochs

Architecture 1 and 3 are trained for 40 epochs further.	
Architecture 1 with N = 5 for early stopping	Architecture 3 with N = 5 for early stopping
  **Training Accuracy: 93.8** **Validation Accuracy: 94.2** **Testing Accuracy: 68.4**	  **Training Accuracy: 88.5** **Validation Accuracy: 88.3** **Testing Accuracy: 64.8**

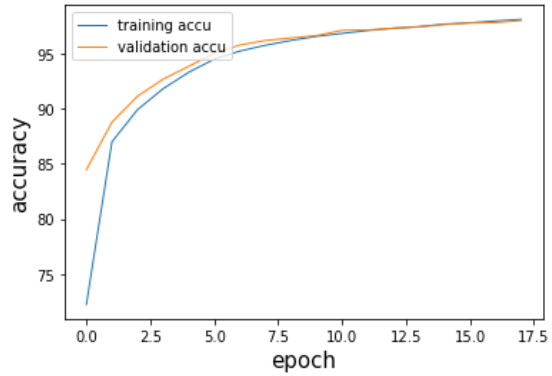
3. Rate Decay (Schedular)

Architecture 1 with epochs = 50, Lr = 0.001, activation function in convolution block is ReLU.				
scheduler	Training Accuracy	Validation Accuracy	Testing Accuracy	Remarks
No scheduler	93.8	94.2	68.4	
StepLR, step_size=10, gamma=0.5	89.9	89.1	64.9	The learning rate decreased and the model took small steps. Early Stopped.
Epochs = 70, StepLR, step_size=20, gamma=0.5	93.2	93.1	67.9	Early stopped after 26 epochs.

4. Activation Functions

Architecture 1 with epochs = 50, Lr = 0.001. No Scheduler				
	Training Accuracy	Validation Accuracy	Testing Accuracy	Remarks
Leaky ReLU	92.2	92.0	67.1	Early stopped: 15 epochs
ELU	94.3	94.1	68.7	Early stopped: 27 epochs

5. Batch Normalization

Architecture 1 with epochs = 50, Lr = 0.001. No Scheduler. ELU Activation.				
		Training Accuracy	Validation Accuracy	Testing Accuracy
BatchNorm layer		**98.1**	**97.9**	**69.9**
● The model early stopped after 18 epochs. ●				

6. Dropout

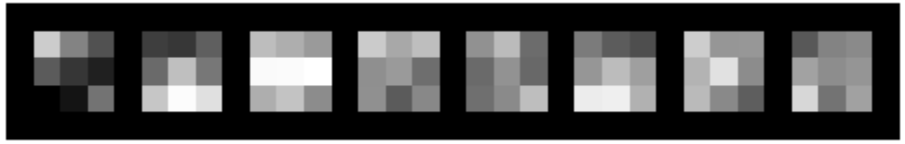
Architecture 1 with epochs = 50, Lr = 0.001. No Scheduler. ELU Activation. BatchNorm after each Conv layer.			
	Training Accuracy	Validation Accuracy	Testing Accuracy
Dropout = 0.25 before the last layer	96.7	96.5	87.1
Dropout = 0.25 after the first layer	94.9	94.6	69.1

Dropout = 0.1 before the last and after the first layer	94.5	94.1	68.8

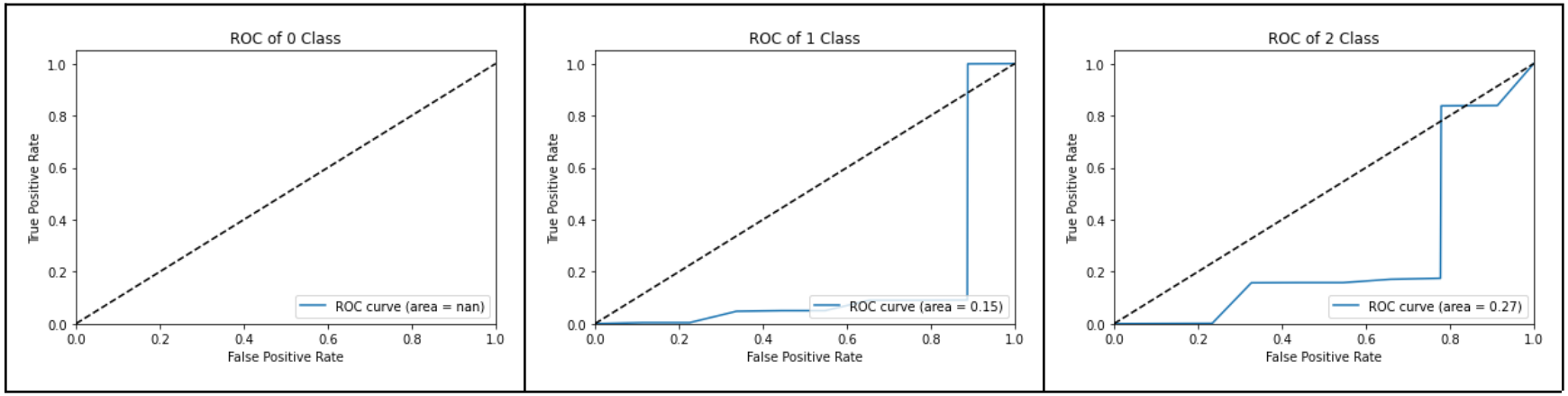
Best model Results:

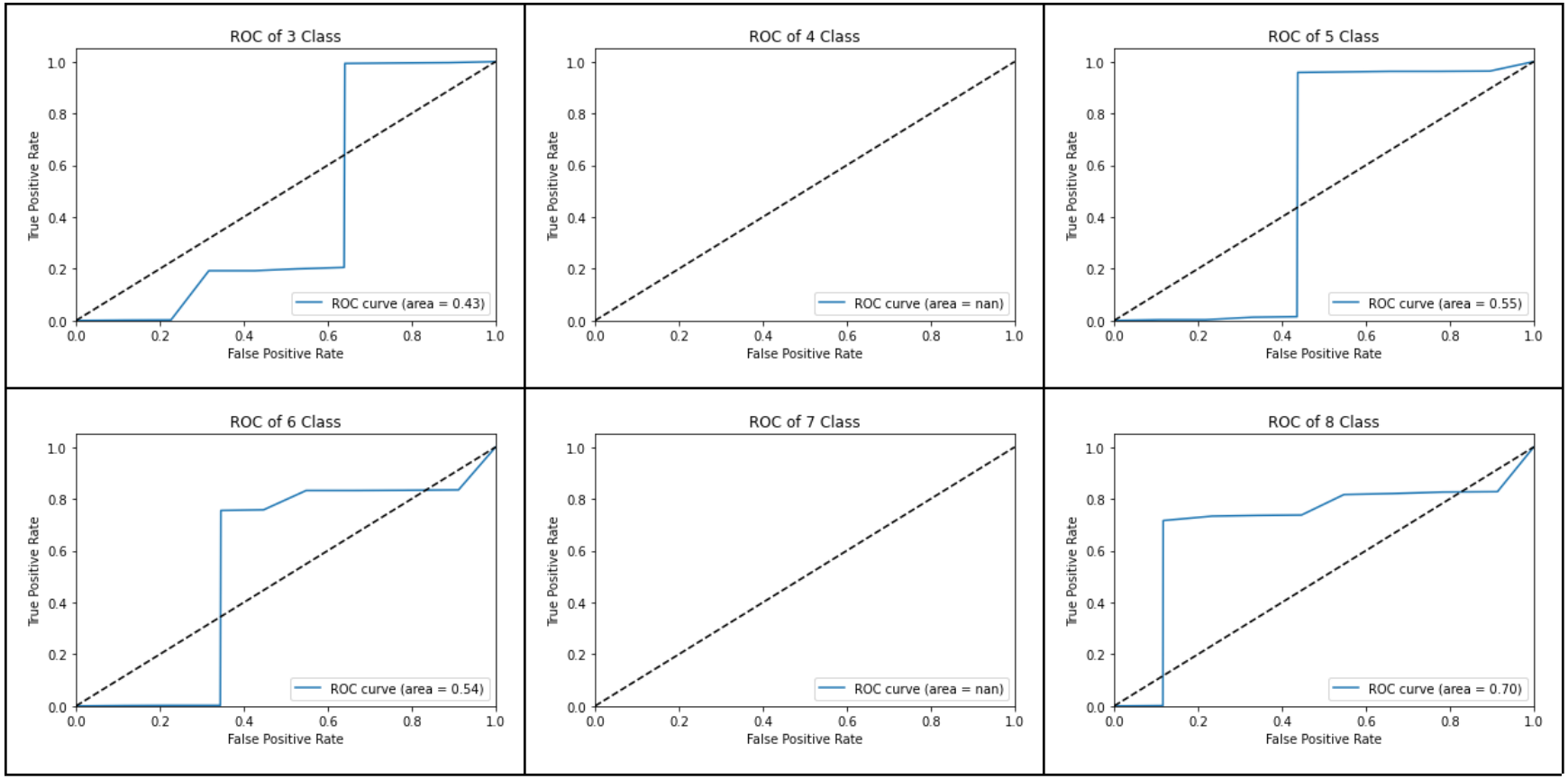
Architecture 1 with epochs = 50, Lr = 0.001. No Scheduler. ELU Activation.

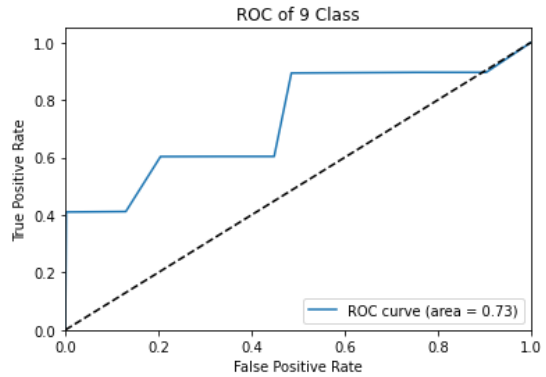
1. **Plot Learned Filters**



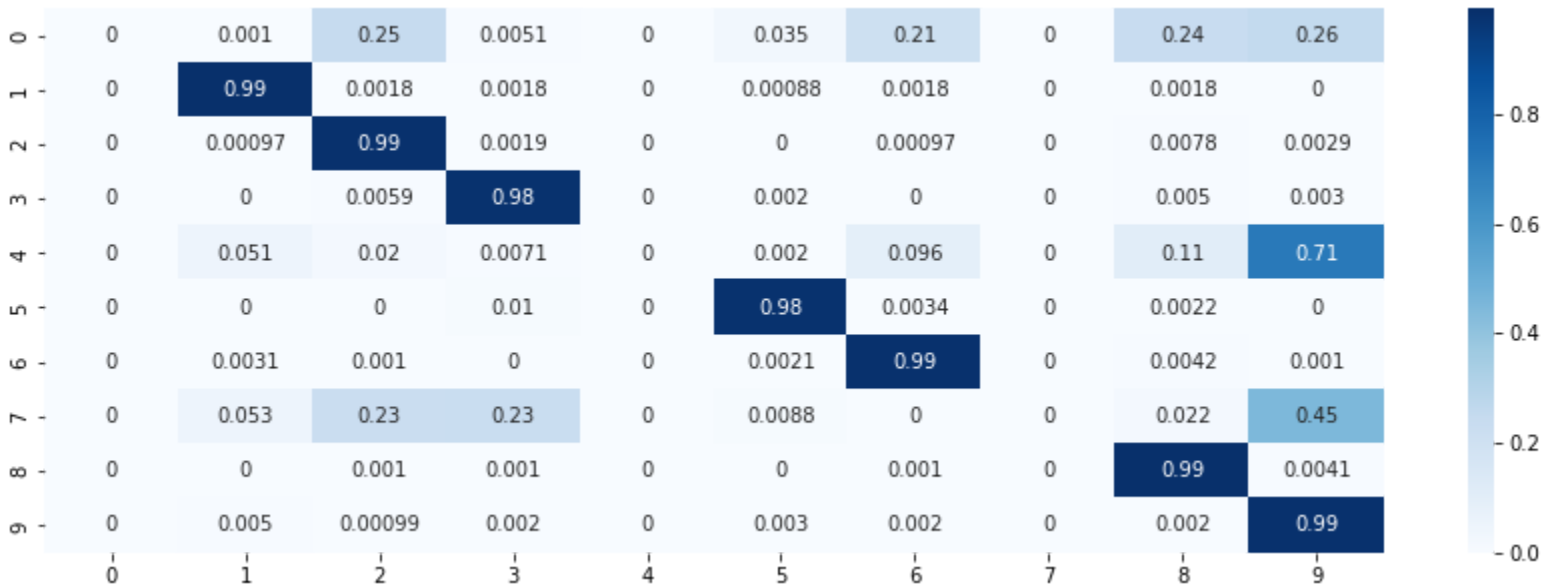
2. **ROC curves**




ROC of 3 Class
ROC curve (area = 0.43)
ROC of 4 Class
ROC curve (area = nan)
ROC of 5 Class
ROC curve (area = 0.55)
ROC of 6 Class
ROC curve (area = 0.54)
ROC of 7 Class
ROC curve (area = nan)
ROC of 8 Class
ROC curve (area = 0.70)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0



3. Confusion matrix



4. Precision, Recall and F1_score

	precision	recall	f1-score	support
0	0.000	0.000	0.000	980
1	0.908	0.992	0.948	1135
2	0.663	0.985	0.793	1032
3	0.788	0.984	0.875	1010
4	0.000	0.000	0.000	982
5	0.943	0.984	0.963	892
6	0.753	0.989	0.855	958
7	0.000	0.000	0.000	1028
8	0.715	0.993	0.831	974
9	0.410	0.985	0.579	1008
accuracy			0.692	9999
macro avg	0.518	0.691	0.584	9999
weighted avg	0.518	0.692	0.585	9999

5. 4 correct and 4 wrong predictions