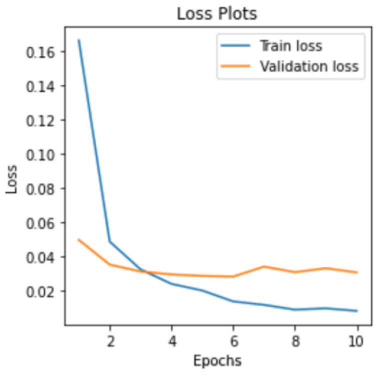
EE5179: Deep Learning for Imaging

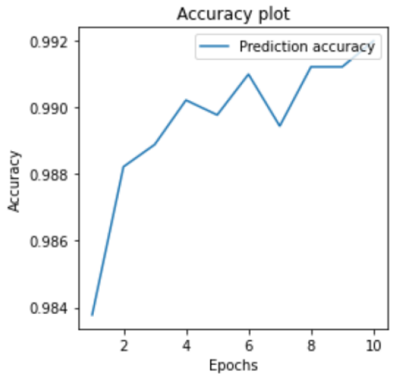
Programming Assignment 2: CNN

1. MNIST classification using CNN

Training on the entire dataset:

Training and validation error and prediction accuracy during training

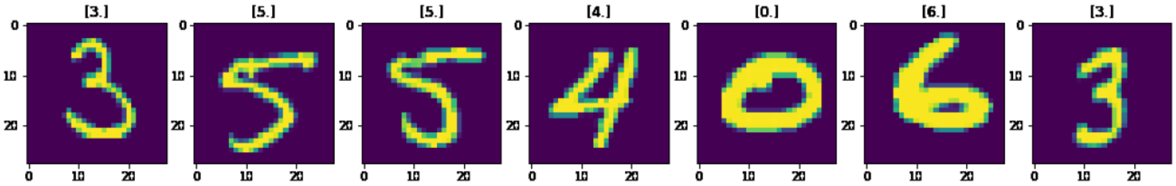




At the end of 10 epochs:

- Train Loss: 0.00845043
- Val Loss: 0.03090557
- Val Accuracy: 0.99200000

Predicted labels:



Network architecture:

- input
- conv1 (32 3×3 filters, stride 1, zero padding 1)
- 2×2 maxpool with stride 2
- conv2 (32 3×3 filters, stride 1, zero padding 1)
- 2×2 maxpool with stride 2
- fully connected layer (500 outputs)
- fully connected layer (10 outputs)
- softmax classifier

Layer 1

Layer 2

Dimensions of input and output at each layer

- Conv1: Input Size = 28x28x1, Output Size = 28x28x32
- Maxpool1: Input Size = 28x28x32, Output Size = 14x14x32
- Conv2: Input Size = 14x14x32, Output Size = 14x14x32
- Maxpool2: Input Size = 14x14x32, Output Size = 7x7x32
- FC1: Input Size = (32x7x7) = 1568, Output Size = 500
- FC2: Input Size = 500, Output Size = 10

Network parameters

- Conv1: Weights: 32x(3x3), Biases = 32, Total = 320
- Conv2: Weights: 32x(3x3x32), Biases = 32, Total = 9248
- FC1: Weights: 1568x500, Biases = 500, Total = 784500
- FC2: Weights: 500x10, Biases = 10, Total 5010

There are 799078 parameters in all- 9568 parameters in the convolution layers and 789510 parameters in the fully connected layers, there are 82.5x parameters in the FC layers when compared to the conv layers.

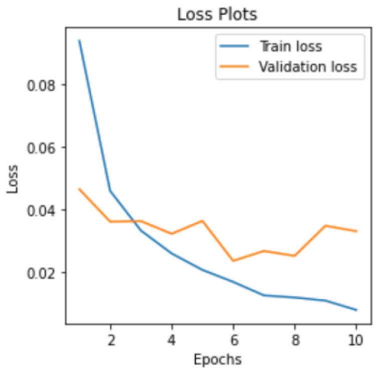
Neurons in the network

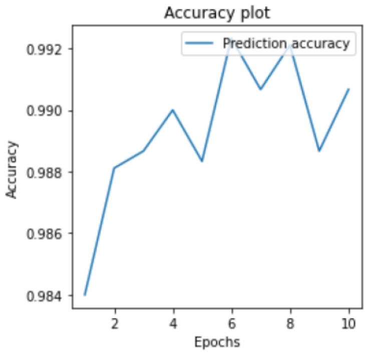
- Conv1: 32x3x3 = 288
- Conv2: 32x3x3x32 = 9216
- FC1: 500
- FC2: 10

There are 10014 neurons in all- 9504 in the conv layers and 510 in the FC layers.

Batch normalisation

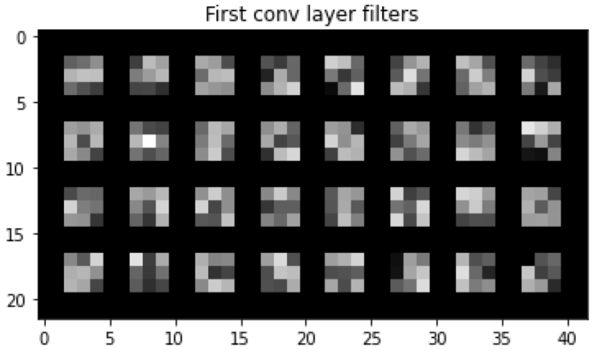
Training and validation error and prediction accuracy during training, batch trained at batch size = 64

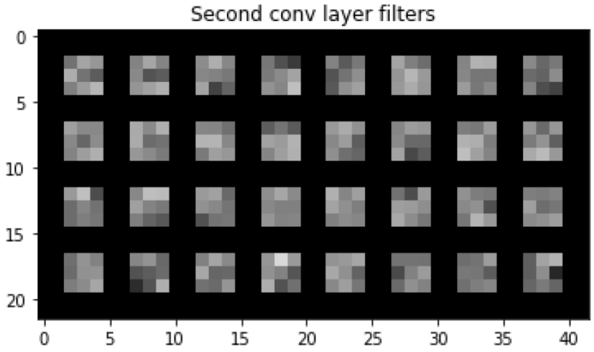




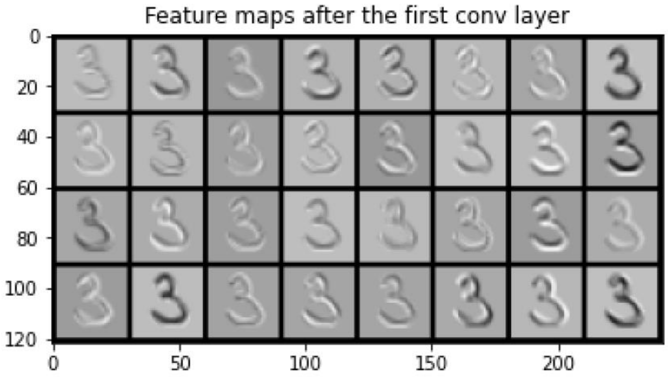
Batch normalisation does not improve the network. It takes slightly longer to train the network with batch normalisation.

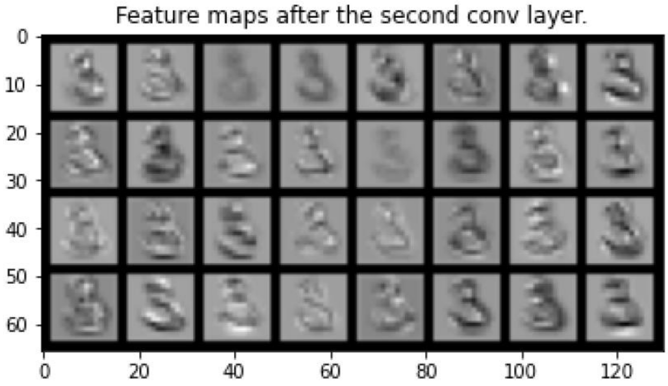
2. Visualising the Convolutional Neural Network



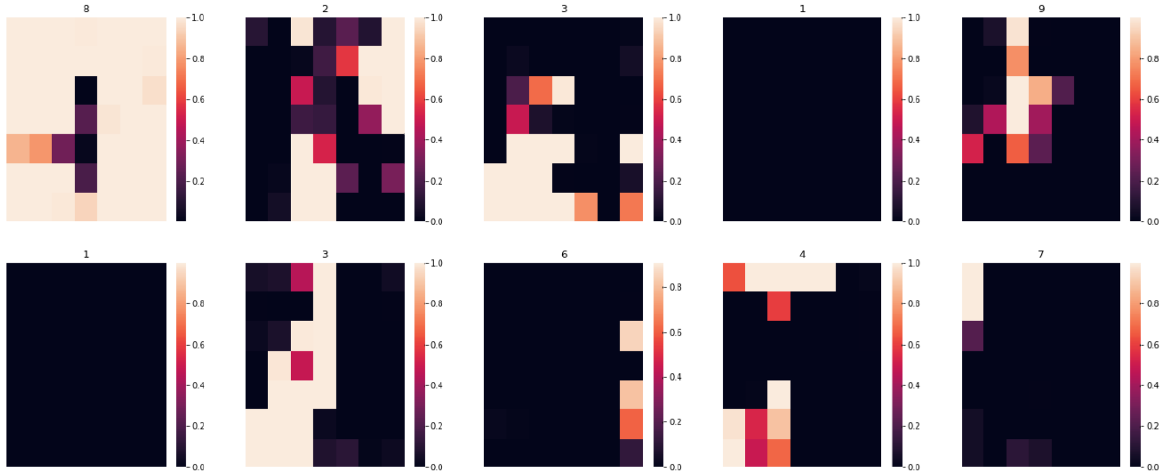


- The first conv layer filters have visible edges showing that these filters act a lot like Gabor filters
- We observe that the second conv layer filters are applied on a larger receptive field. The values in the kernel are very similar in contrast to the first conv.





- The input image is very visible in the lower conv layer
- The activations in the first layer clearly observes edges
- The second conv layer observes more than edges as seen in the activations

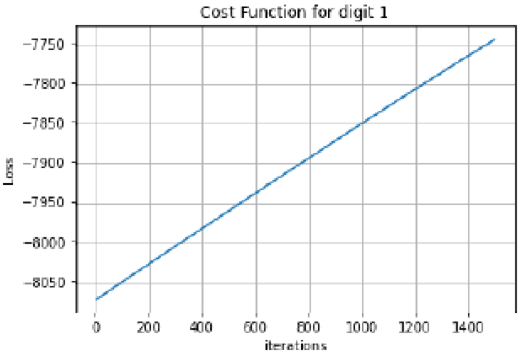


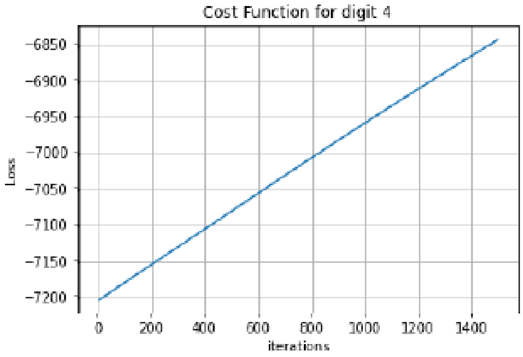
Occlusion experiments:

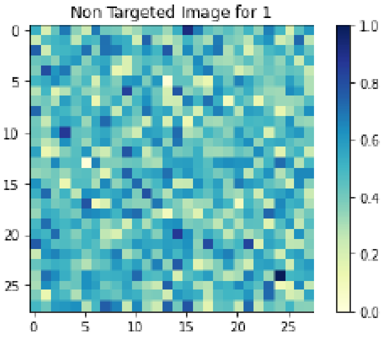
- The occlusion experiment depicts well how the prediction is accurate when parts of the image that do not include the digit are occluded. This concludes that the model is accurate.

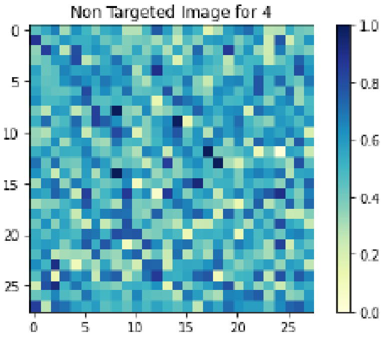
3. Adversarial Examples

Non-targeted attacks:



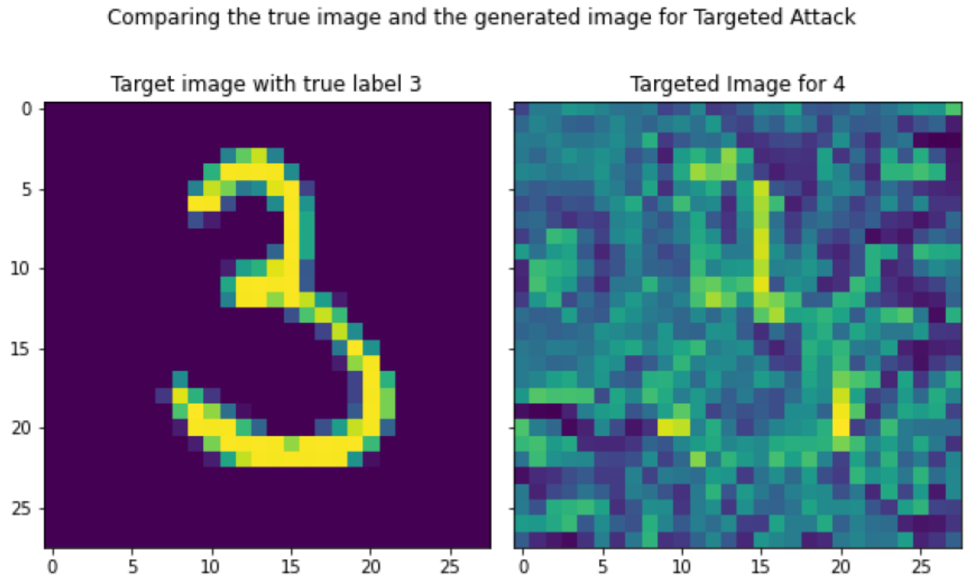






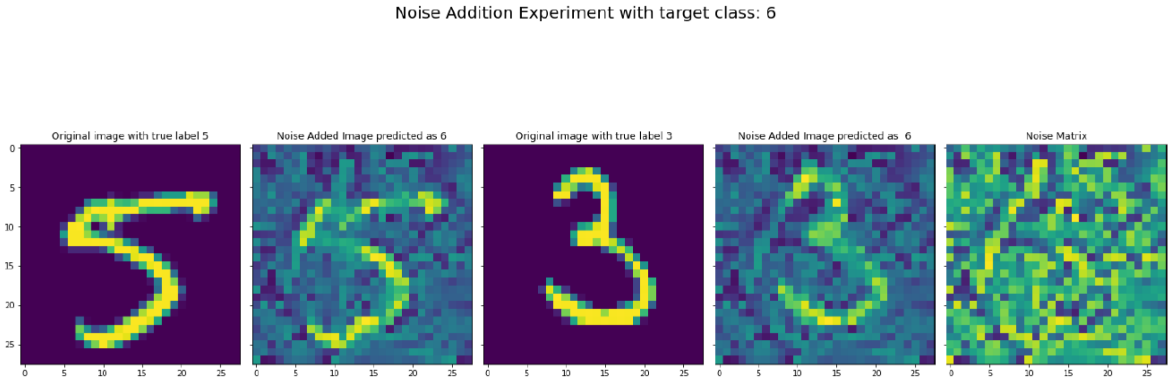
- The network predicts the digits with high probability.
- The noise images do not resemble digits

Targeted attacks:



- The image resembles a lot to the original image
- The however probability of prediction is very high
- However, we do see that the lower curve of ‘3’ has disappeared

Adding noise



- The images are yet again predicted with very high confidence
- We do observe that the noise image has a faint ‘6’ observable