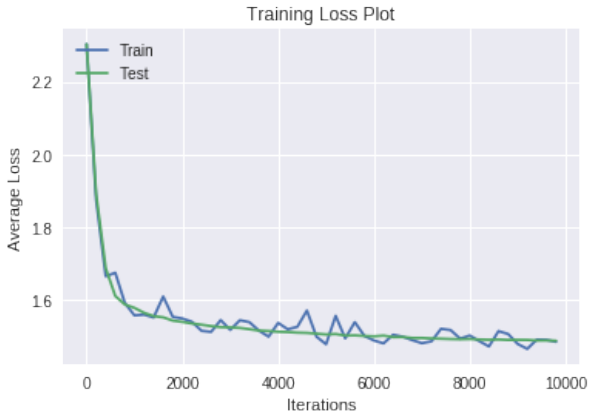
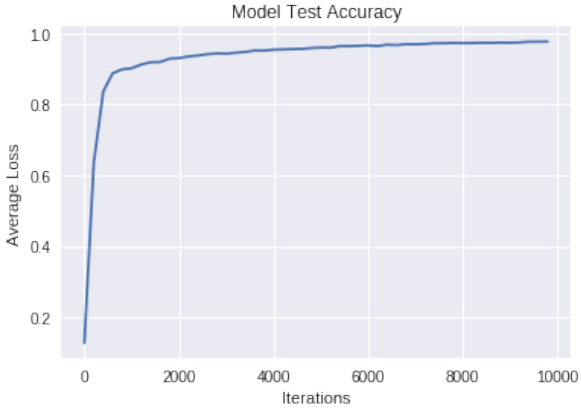
Deep Learning For Computer Vision
Assignment 2: Convolutional Neural Networks
Author: G Pradeep EE16B050

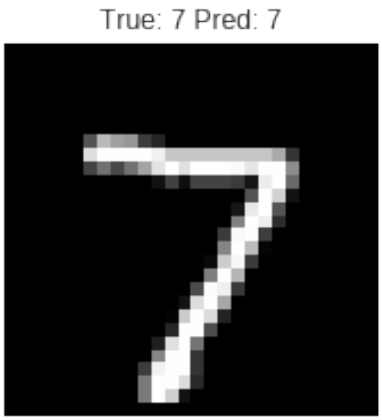
1. MNIST Classification Task:

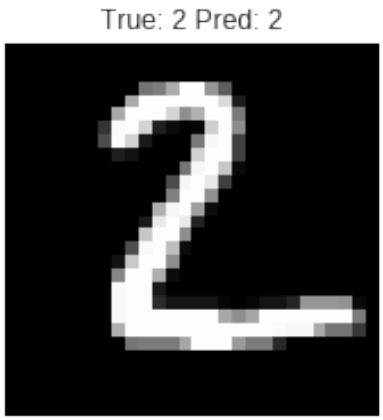
Note that image normalization converts the image to a [0,1] range. All experiments have been performed with this in mind.

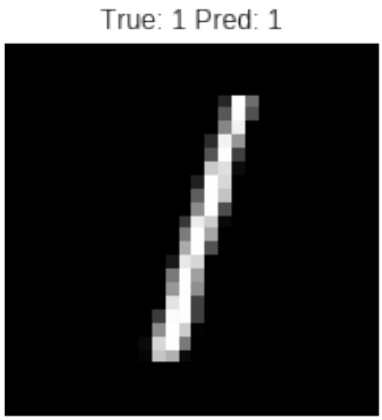
Model 1:
Accuracy = 0.9753
Learning Rate = 5e-4
Epochs = 10 using Adam Optimizer
train_batch = 64
test_batch = 10000





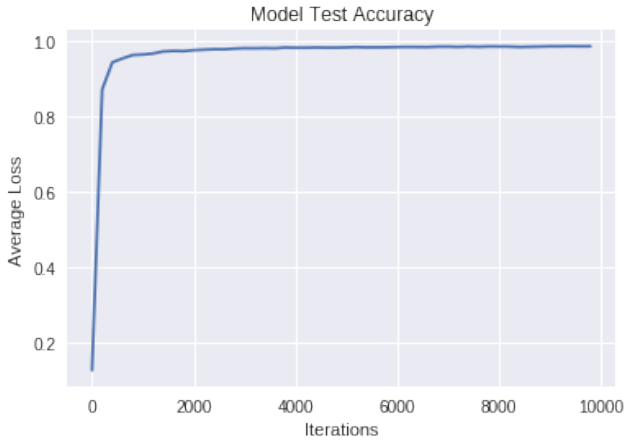


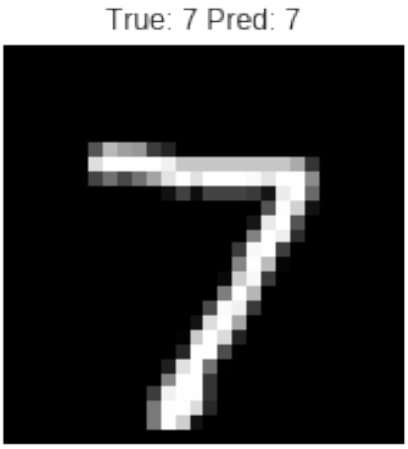


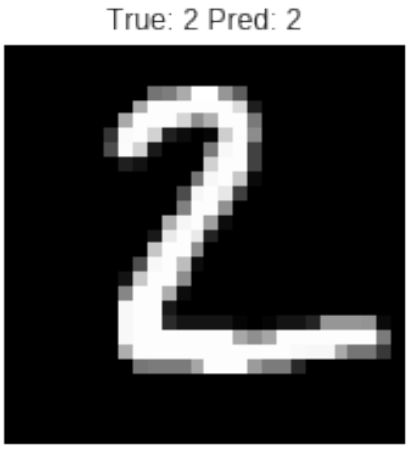


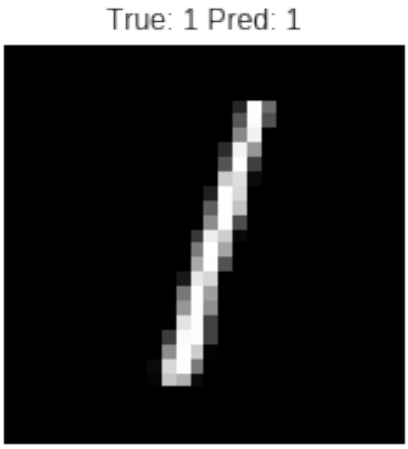
Model 2: With BN
Accuracy = 0.9869
Learning Rate = 5e-4
Epochs=10 using Adam Optimizer
train_batch = 64
test_batch = 10000

The usage of BN highly improves training speed. The classification accuracy is also bettered by a percent which is quite significant.









The Model 1 is used in all further experiments.

Dimensions:

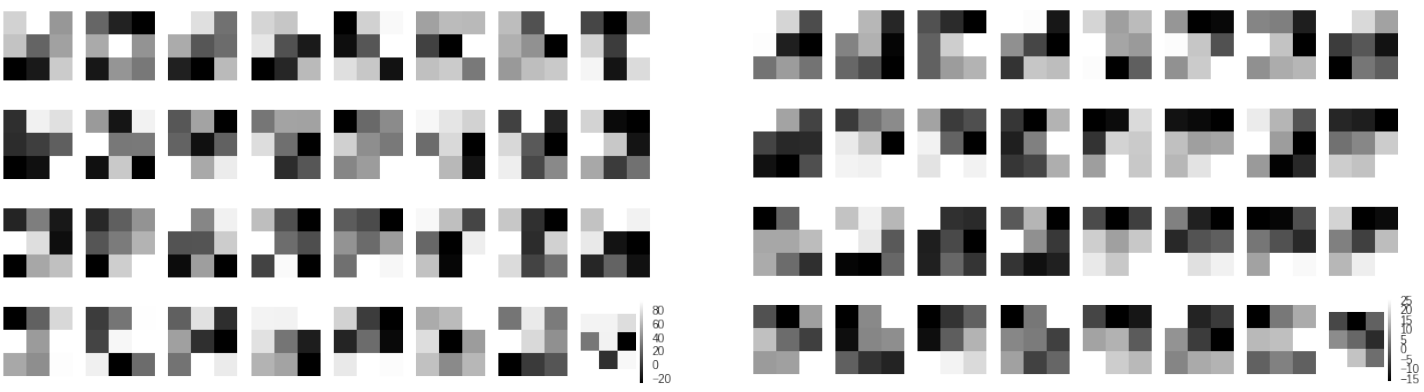
Input	(28,28)
Conv1	32 of (28,28)
ReLU1	32 of (28,28)
Maxpool1	32 of (14,14)
Conv2	32 of (14,14)
ReLU2	32 of (14,14)
Maxpool2	32 of (7,7)
FC1	1568
FC2	500
Softmax	10
Output	1

Neurons:
Conv1: 288
Conv2: 288
FC1: 1568
FC2: 500
Output: 10
Total: 2654 = 576(Conv) + 2078(FC)

Parameters:
Conv1: 32*3*3 weights + 32 biases = 320
Conv2: 32*3*3 weights + 32 biases = 320
FC1: 1568*500 weights + 500 biases = 784500
FC2: 500*10 weights + 10 biases = 5010
Total: 790150 = 640(Conv) + 789510(FC)

2. Visualising the network:

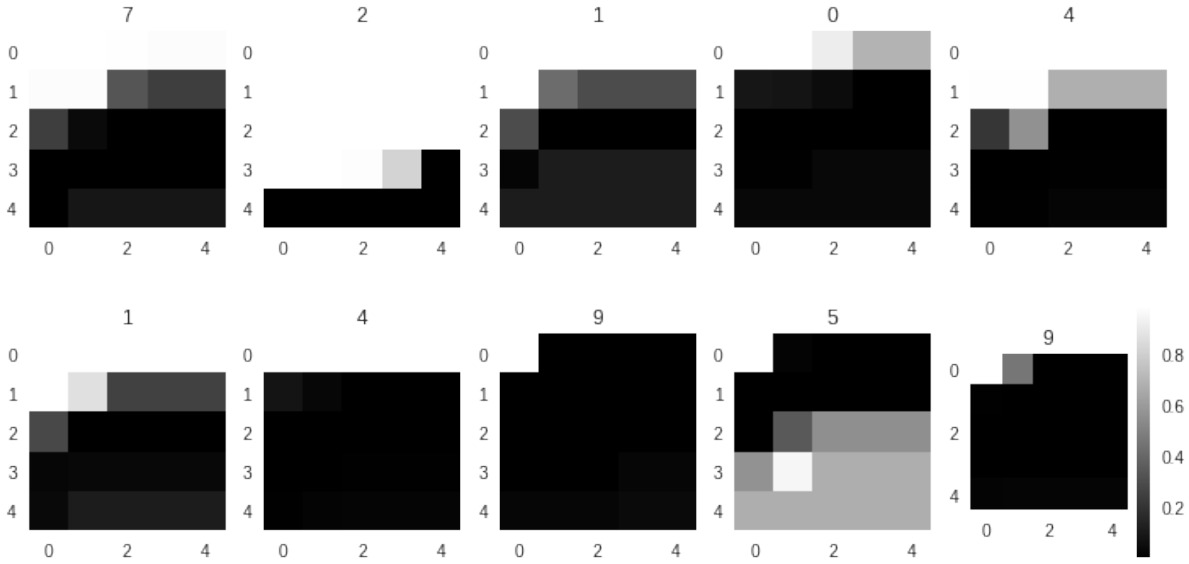
It is very apparent that the feature maps in Conv 1 are quite crude compared to Conv2. In Conv2, we start seeing maps that are reminiscent of corner, edge and diagonal line detectors. As we go deeper, further complex features that are combinations of these maps are expected.



Conv1 and Conv2 Feature Maps

Occlusion:

The image was convered with blocks of (12,12) pixels with a stride of (4,4) pixels. The numbers on the top denote the true labels of the randomly chosen images. The white maps show occluded regions still giving a high probability and the black maps, the vice-versa.
Note the plot for 2. The features learnt are present significantly in the lower strip of the image which corresponds to the horizontal bar at the bottom of the number 2. Almost all other categories reveal robust, meaningful feature learning.



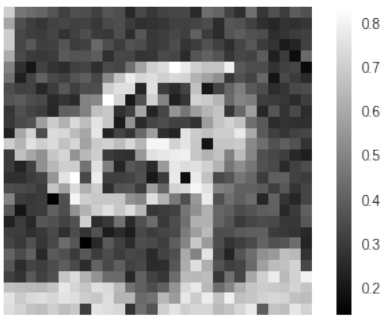
3. Adversarial Training:

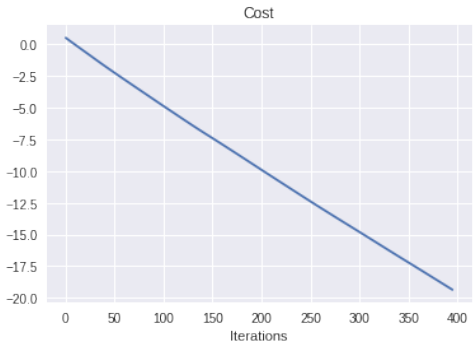
Non Targeted training:
Learning Rate = 5e-4
Iterations = 200

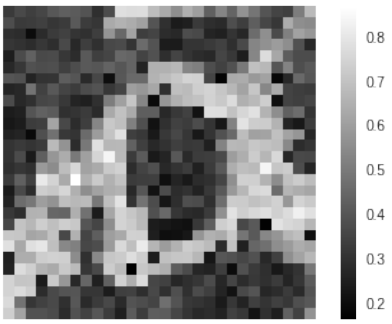
The images were initialized from a Gaussian distribution with a mean 0.5 and standard deviation 0.05
.
The images and the plots for 9, 0 and 5 respectively are as shown below.

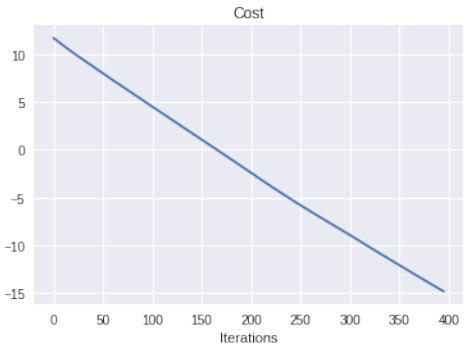
The network predicts very high confidence measures for these generated adversaries. The cost is always decreasing during the training.

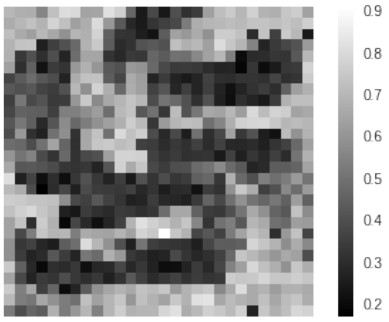
Mostly the generated images look like the intended number. But occasionally, as in the case of 5 here, it is not the observed trend. Given that the MNIST dataset doesn't have all the examples completely centred and identical, one possible reason may be the inconsistencies arising from the training data with respect to the orientation and location of the pixels. The method used bsaically optimizes the logit class score. It is definitely possible that there exists a similar, but a different shape, another optimimum, that also minimizes the logit score.

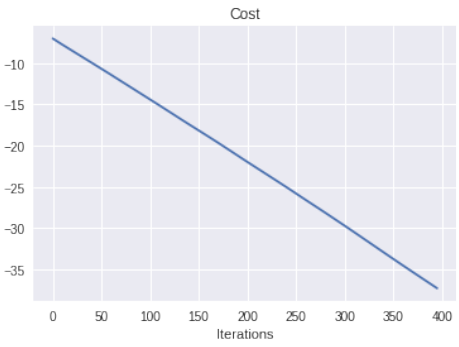




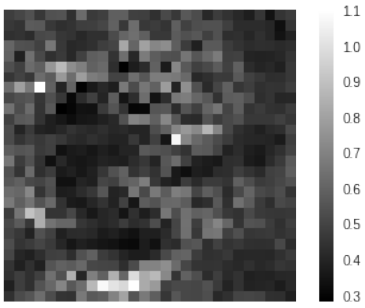


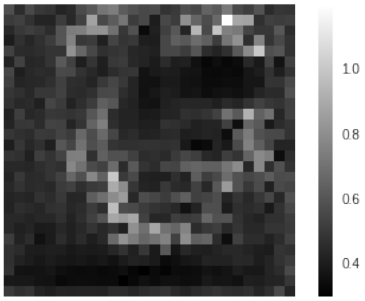


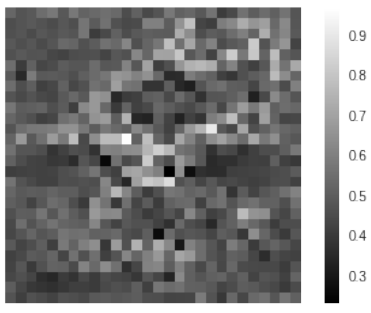




Targeted Attacks:
Learning Rate = 0.001
Iterations = 400
b=1
The initial image belonged to the class 2. The generated images for classes 3, 6 and 8 are shown. The images resembles the numbers for all the classes. Note the absence of falsely activated regions as in the previous case. The images are smoother compared to earlier.



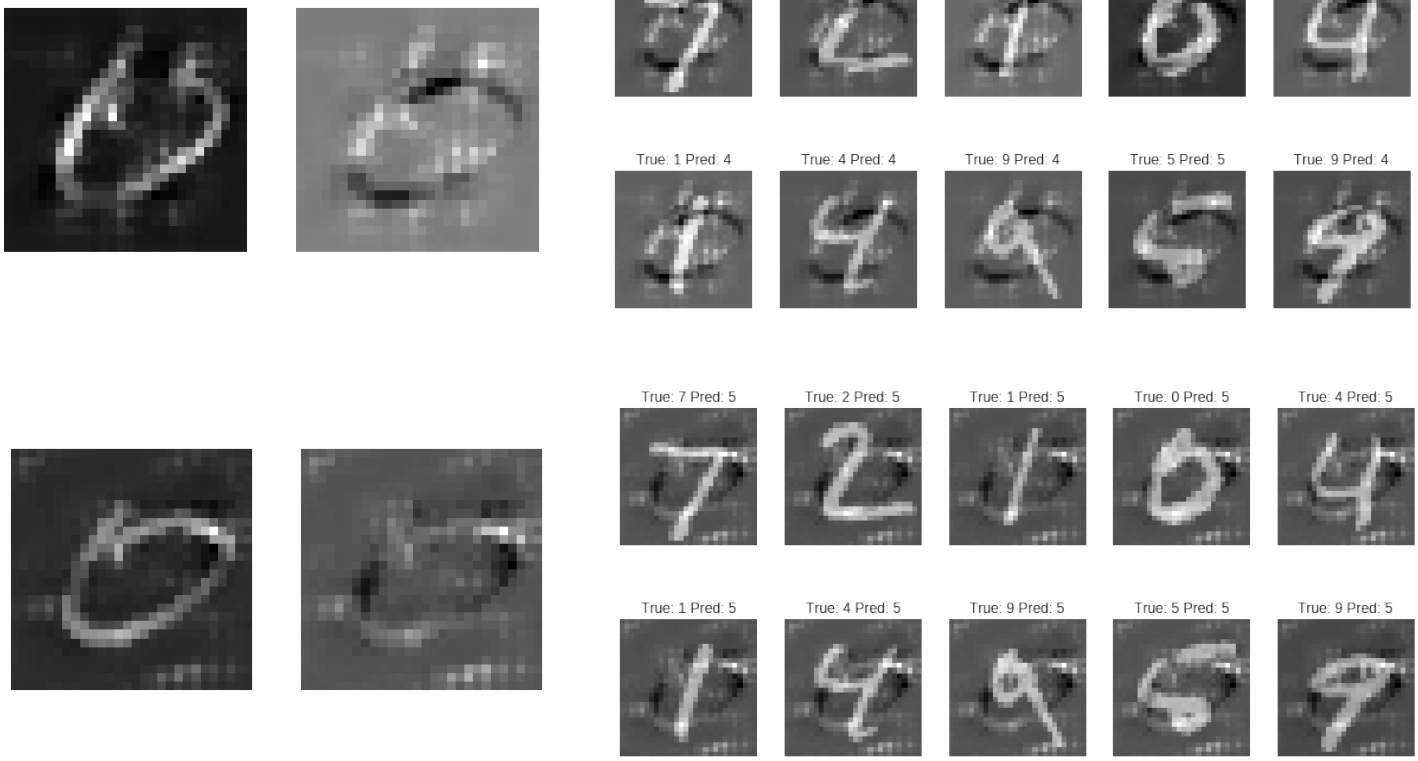




Adding calculated noise:
Learning Rate = 0.0005
Iterations = 400

The inital image belonged to the category 0. They were added with noise optimized for the categories 4 and 5 respectively. The left half shows the adversarial iage and the corresponding noise added. The right half shows the true and predicted categories when the calciulated noise is added to actual images.

Surprisingly, in most cases, the noise calculated on one class, also performs well on actual images from other classes in misclassification rates.



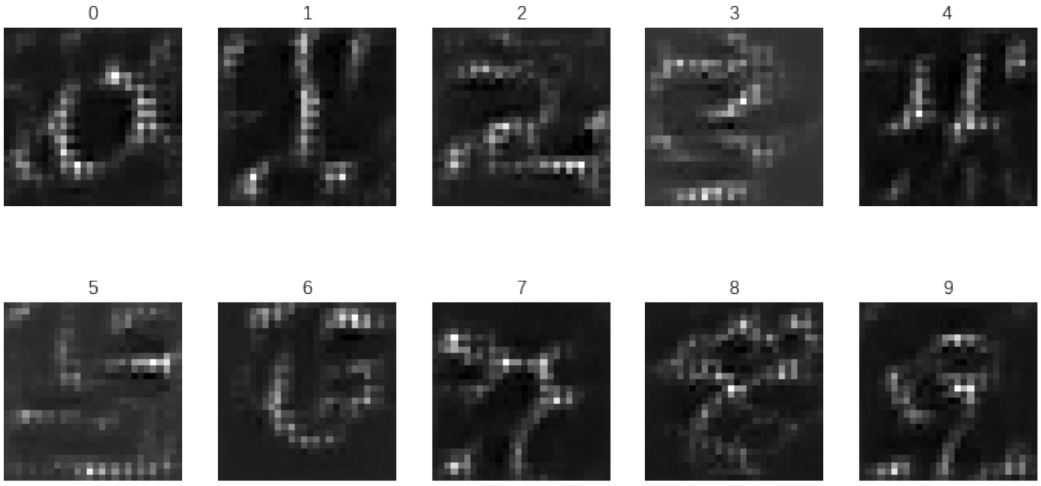
4. Optional Exercises:

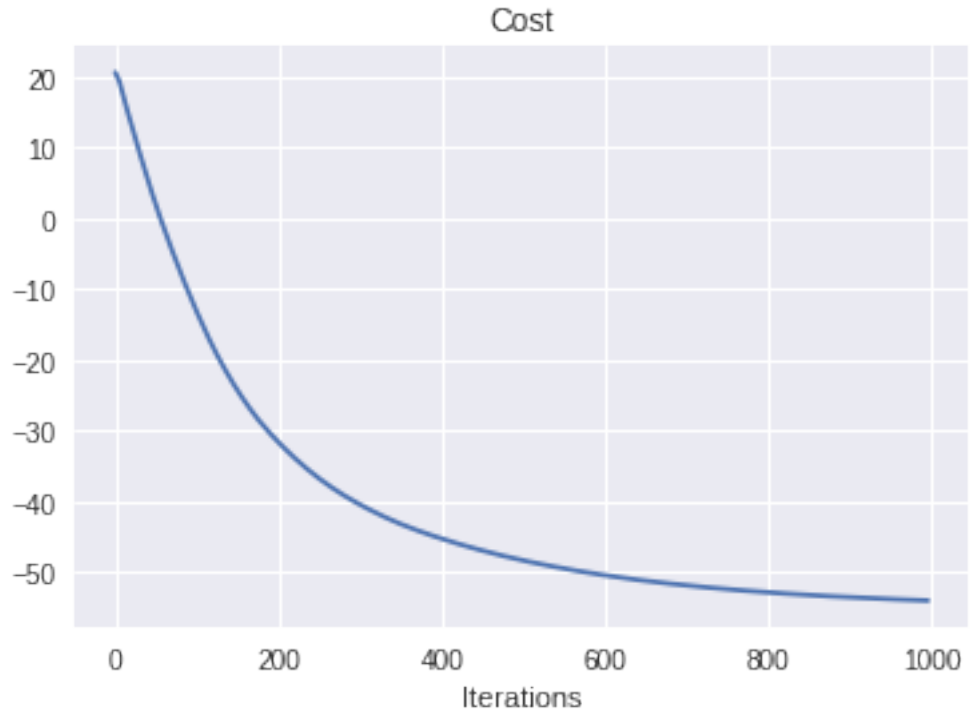
Optimization Over Images:
Learning Rate=0.05
Iterations = 200

Images are trained starting from a zero image to maximise the output confidence. In this experiment, we also use a L2 regularizer over the image pixels to improve performance. The combined loss function is optimized.

The first plot shows the generated images. The generated images are quite close to the actual digits in themselves. The jitter in the bulk of the digits in this method is a notable feature that should be explored. The regularization could be responsible for suppressing down those pixel values.

The training cost falls down and converges.





The maximisation of any intermediate activation or feature map, as observed, reduces or increases the logit scores of classes. Some features score negatively in a class's logit score(since it is not found in that digit) and the presence of some others improve the score for that class.

Defense Against Adversarial Attacks:

Of all the methods studied today, most adversarial images have jittery noise. Simple methods of defense would be blurring or thresholding images before feeding into the network to remove off the noise. Other methods would be training the networks with adversarial examples to make them robust. GANs and other generative models can be used for this purpose.