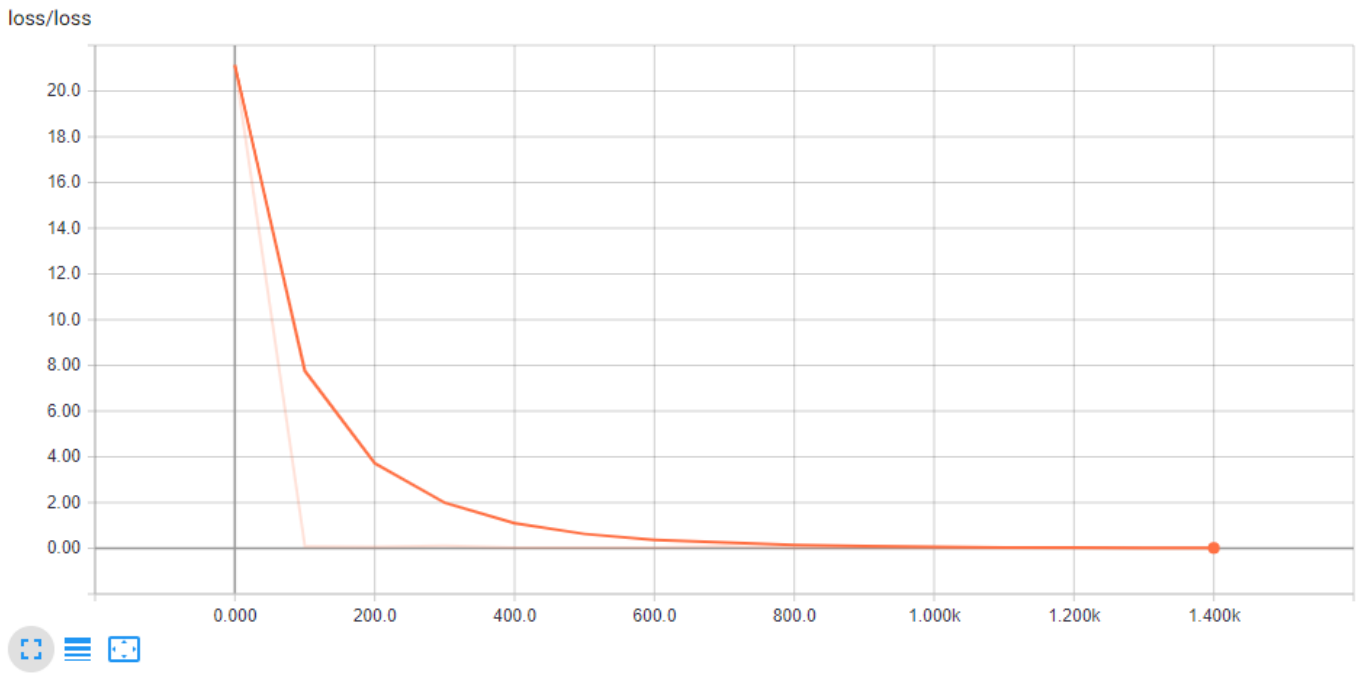
Name: **Shaik Masihullah**
College: **Indian Institute of Information Technology Sricity**
Email ID: **masihulla17@gmail.com**

Results Observed:

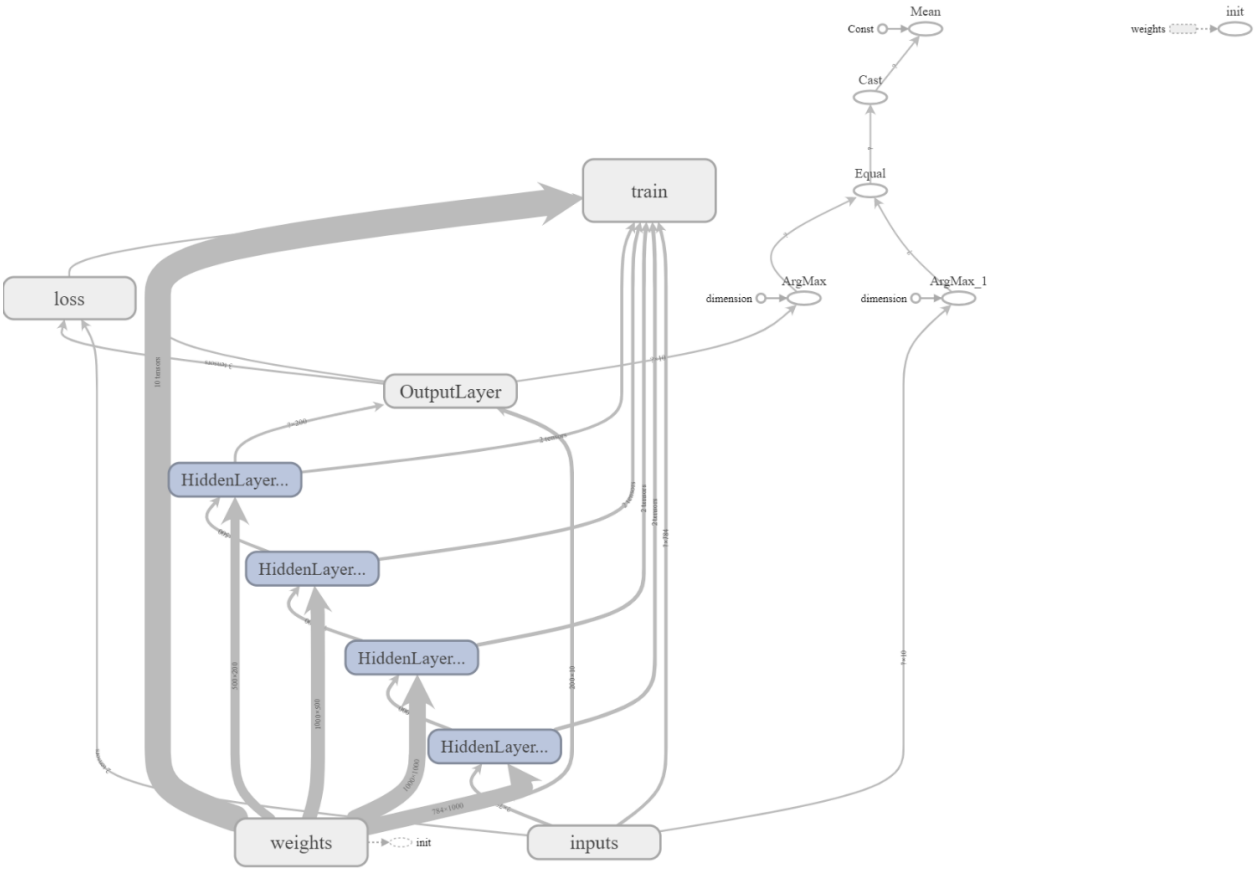
- ➔ Trained on MNIST handwritten digits dataset.
- ➔ Used Tensorflow, numpy, matplotlib, math modules and tensorboard for visualization.
- ➔ Used Gradient Descent Optimizer and softmax cross entropy to calculate loss.
- ➔ Weight pruning performance : `97.08999991, 96.93999887, 94.73999739, 89.93999958, 64.20999765, 44.5600003 , 18.32000017, 11.80000007, 9.23999995, 8.69999975`
- ➔ Unit pruning performance : `97.08999991, 82.52999783, 22.20000029, 12.68000007, 11.7399998 , 9.88000035, 9.84999985, 9.66000035, 8.64000022, 6.47`
- ➔ For pruning values : 0, 25, 50, 60, 70, 80, 90, 95, 97, 99

Data from Tensorboard :

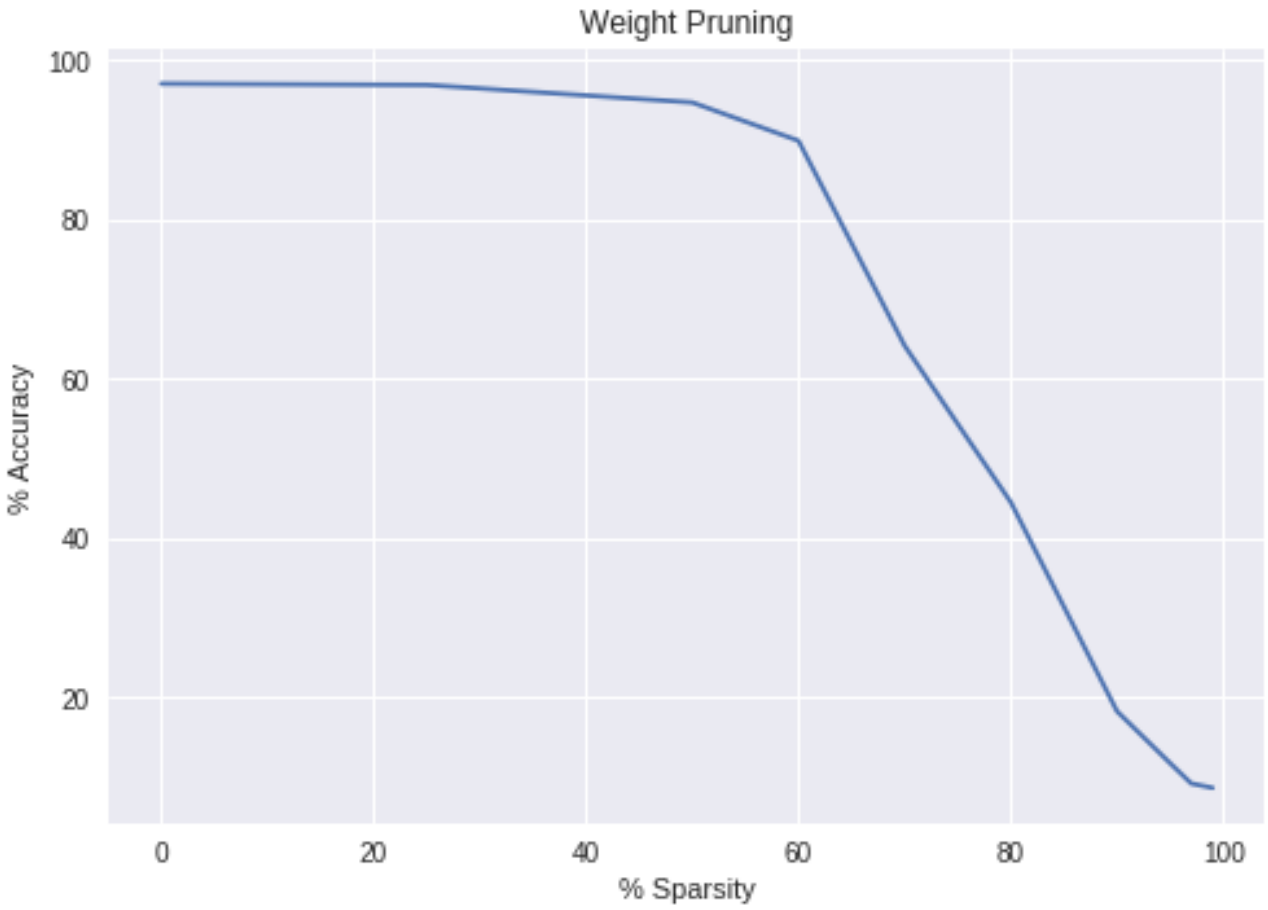
Loss :

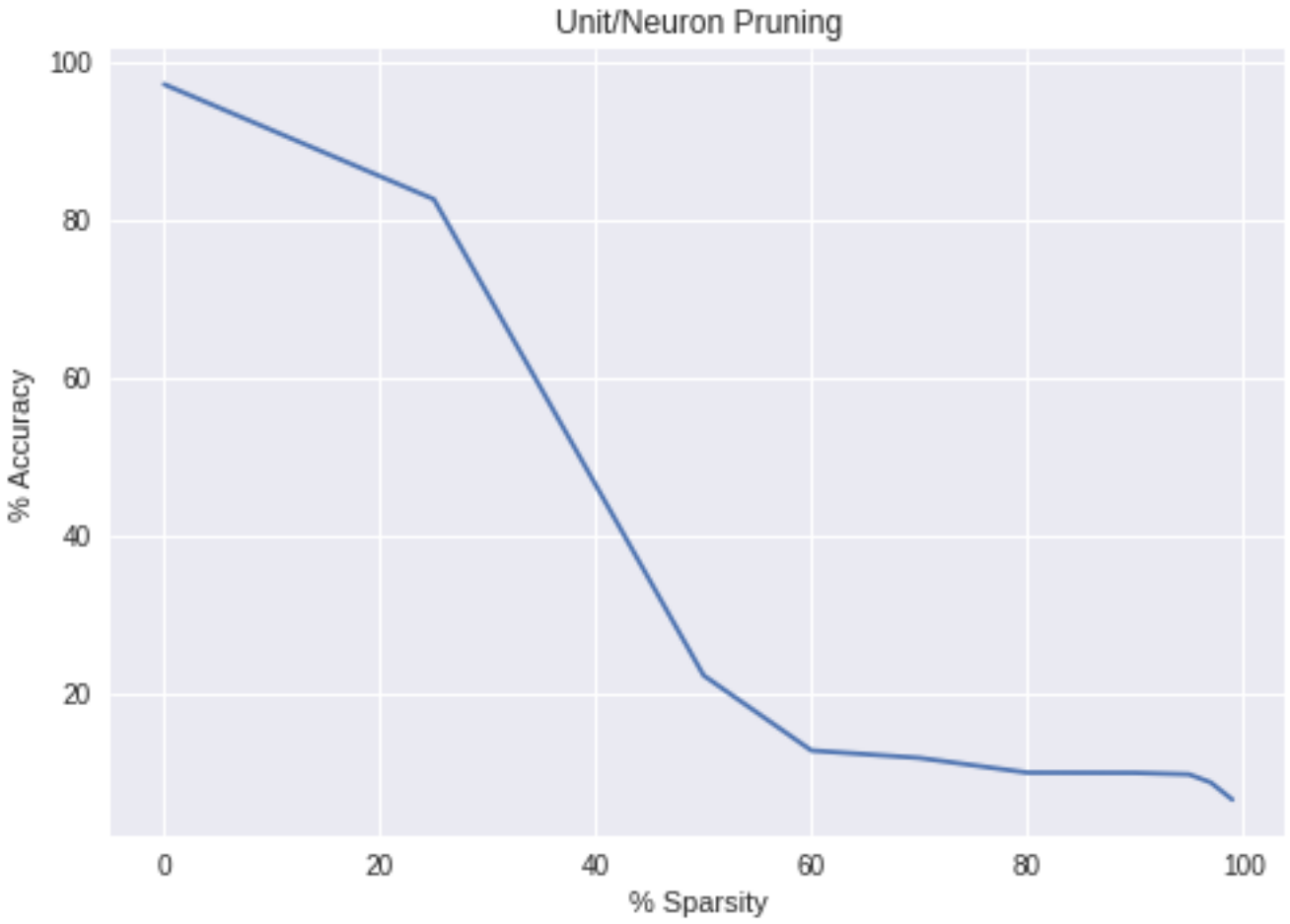


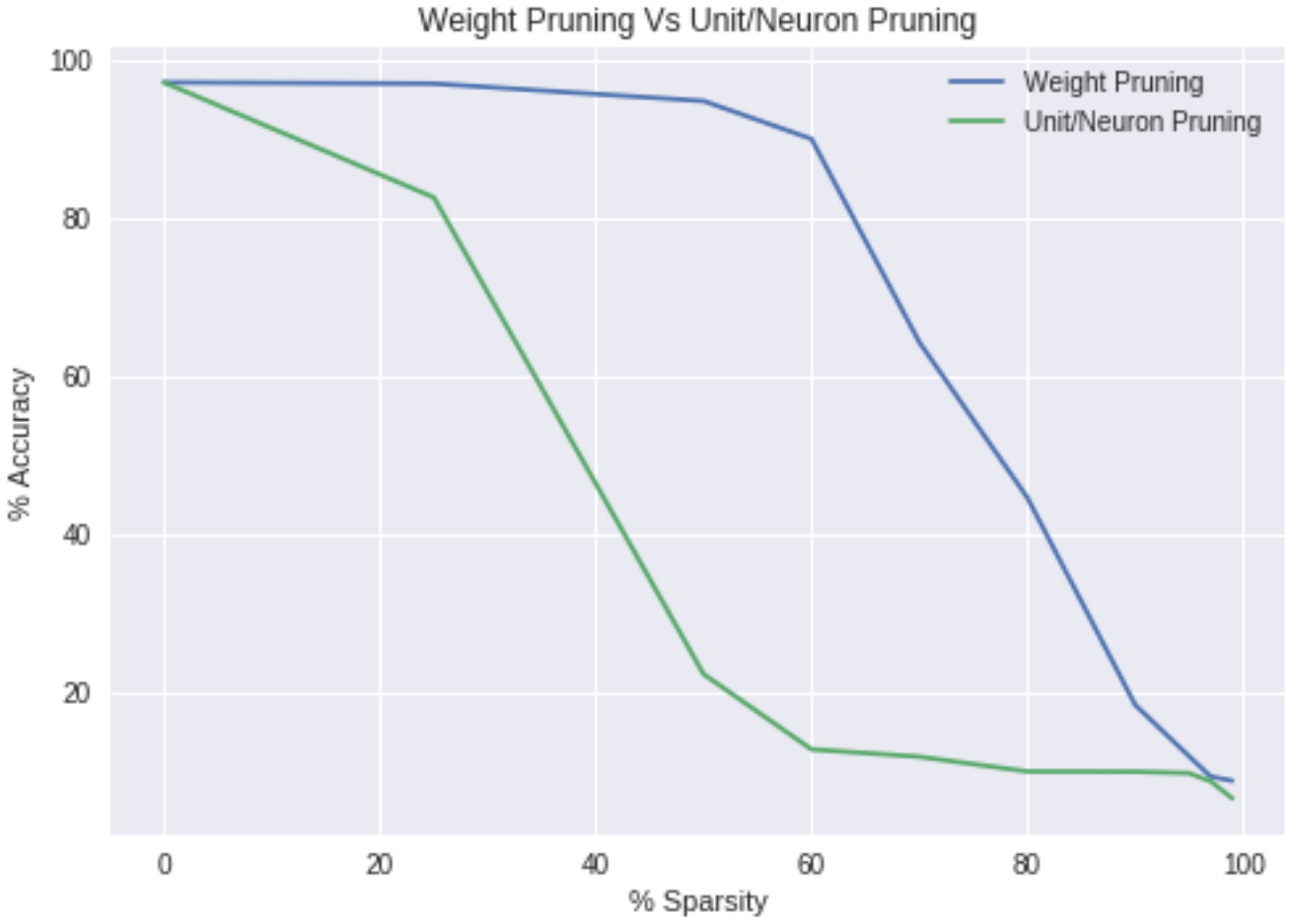
Model :



%Sparsity vs %Accuracy :




Unit/Neuron Pruning
100
80
60
40
20
% Accuracy
0
20
40
60
80
100
% Sparsity


Weight Pruning Vs Unit/Neuron Pruning
Weight Pruning
Unit/Neuron Pruning
100
80
60
40
20
% Accuracy
0
20
40
60
80
100
% Sparsity

Interesting insights I found:

1) Weight pruning is performing better than the unit/neuron pruning.
2) In very rare cases, unit/neuron pruning overtake the weight pruning.
3) Pruning can be done only up to certain percentage for better performance.
4) After a certain % of sparsity (most probably between 60% - 70%), performance accuracy reduced rapidly.
5) Although pruning reduces some accuracy in prediction, but it worth it, because of the increase of the speed in prediction.

Do the curves differ?

Yes, both the curves differ a lot. This is because unit pruning zero out, more number of neurons than weight pruning which leads to the bad performance. And also in unit pruning, we zero out a complete column of neurons in the weight matrix which contributes to the loss in performance. But in weight pruning this is not the case, in that only a few selected locations in the weight matrix are made zero. So, weight pruning performs better than the unit pruning.

Do you have any hypotheses as to why we are able to delete so much of the network without hurting performance?

Here we are deleting the neurons whose contribution towards the prediction is significantly low, so even after making these neurons zero, it doesn't hurt much of the performance. But as the % of sparsity increases, this hurts the performance, as the neurons which contribute more towards the prediction may get deleted.