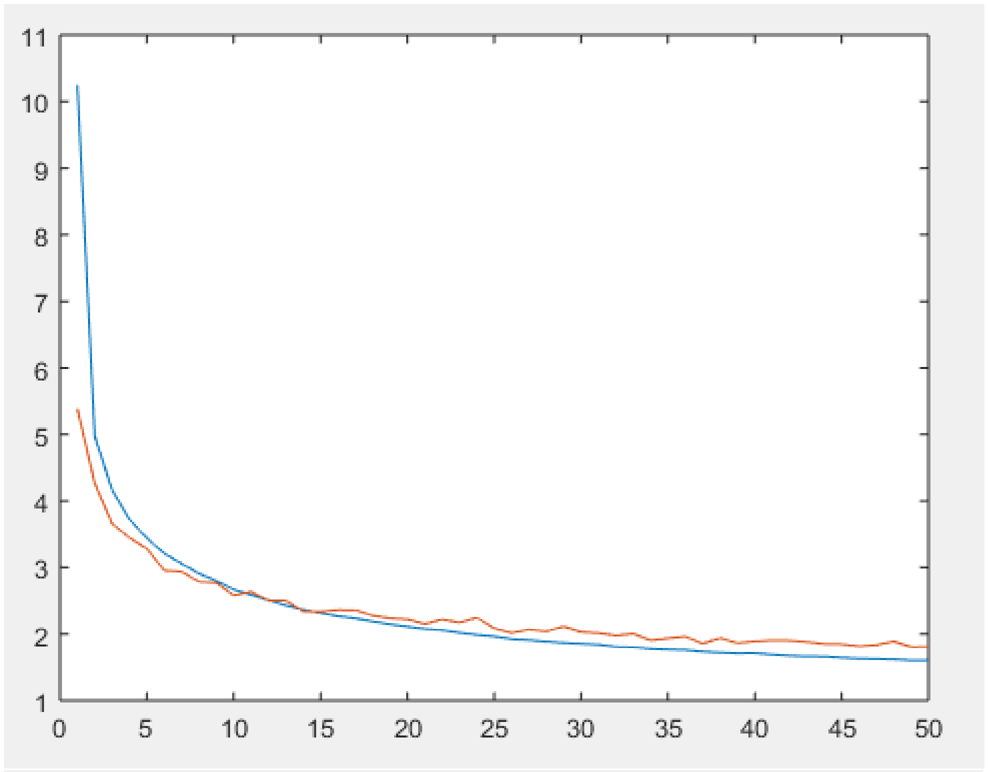
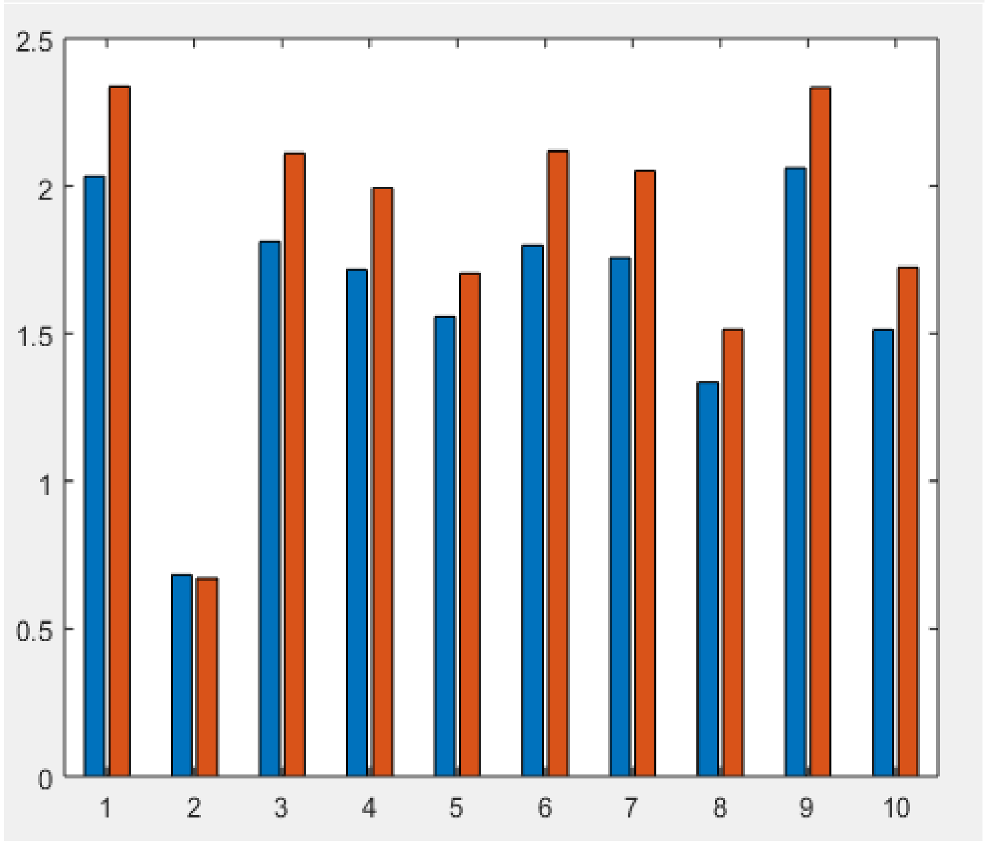
System Description:

A description of all the parameter choices I have made – learning rate, momentum, β and λ, rule for choosing initial weights

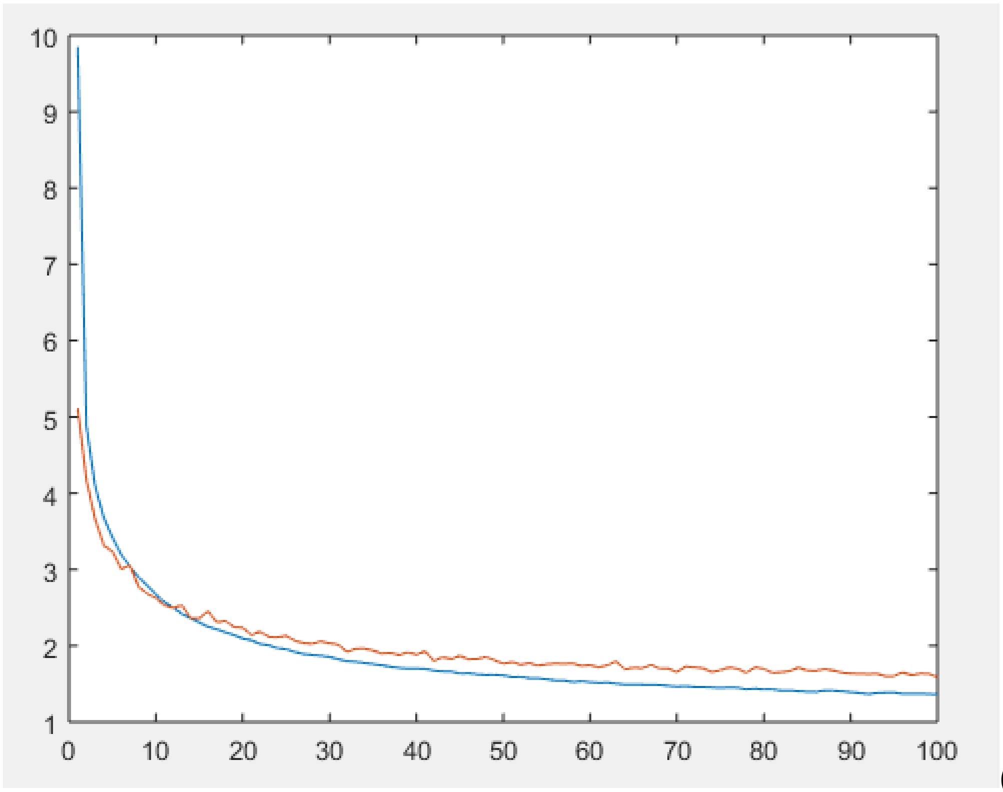
- The parameters I choose are as following, I have taken learning rate same as 0.08; which gave me best results and avoided overfitting, I have also tried increasing which overfitted my data, tried decreasing it to 0.05 which did not optimize my results.so I have stayed with 0.08.
- I have taken a stronger momentum, since my gradient keeps changing causing variation which may result in shifting outwards from the path of local minima, making convergence harder.by taking strong momentum it will smooth out the variations.so I have taken strong momentum of 0.7.
- Tried different Lagrange multiplier values and came to conclusion that its best to keep it max, so I have taken its value beta=4.
- From class lecture I got to know that proper weight decay needs least lambda value so I took it as 0.001.
- Initially I have taken Xavier initialization, later converted it into normal gaussian distribution.
- My network max converged around 15th epoch and I let it run till 100th epoch where it further decreased.

AUTOENCODER WITHOUT REGULARIZTION

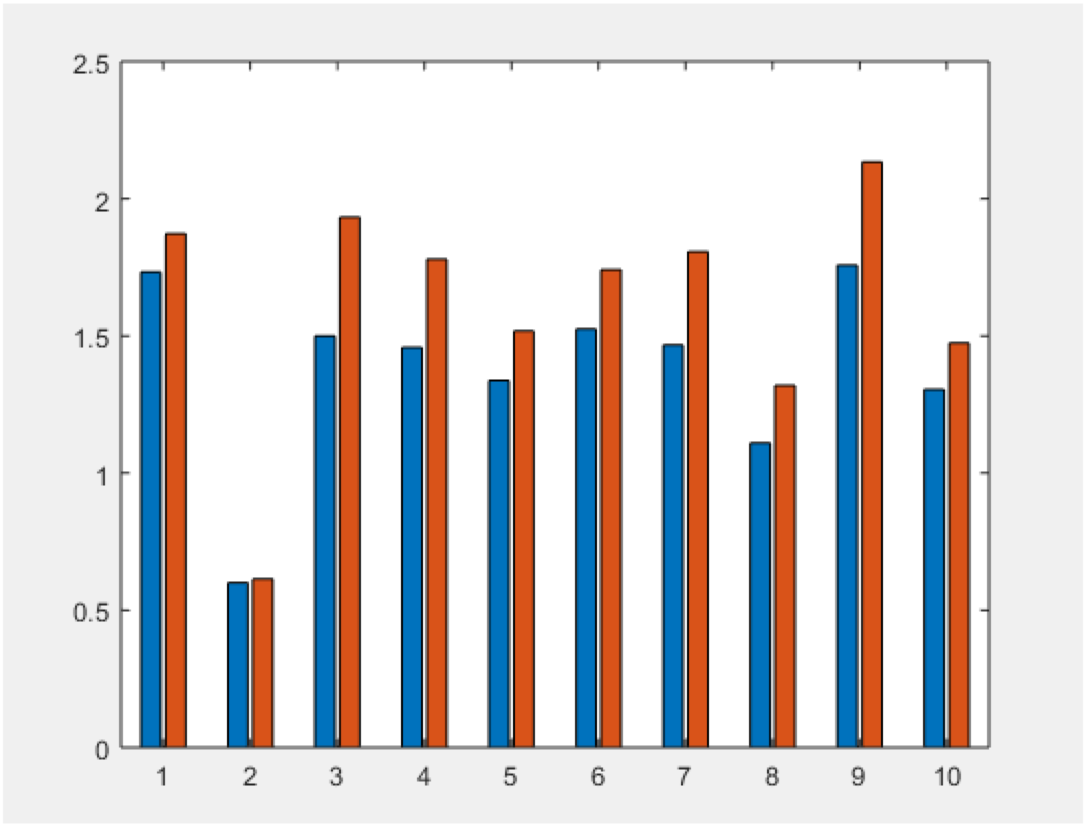


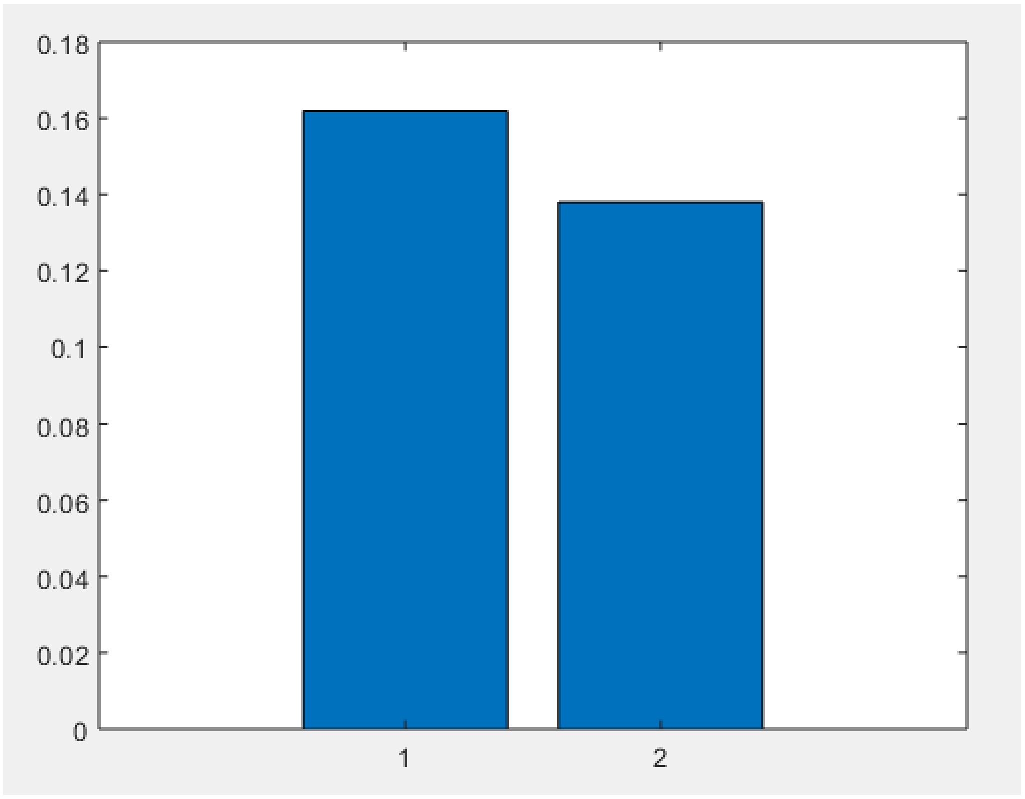


AUTOENCODER WITH REGULARIZTION



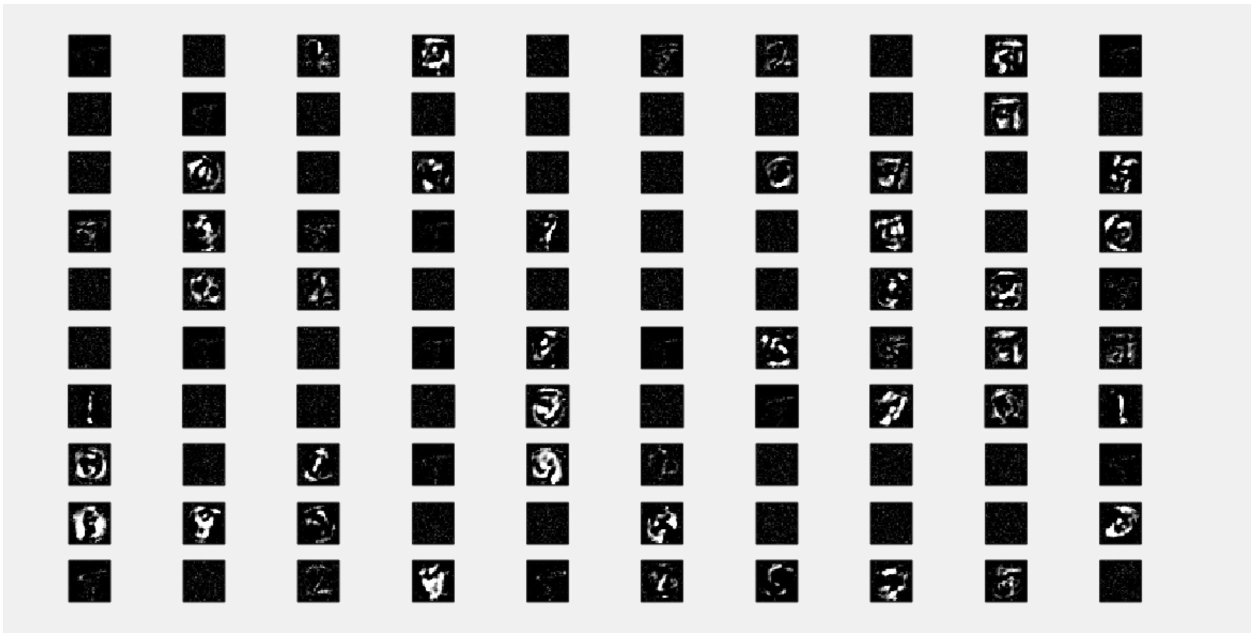
(normalized(4000))


2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10


0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
2

<u>Features:</u>

After sparsness:



Analysis of Results:

In my auto encoder training weight are adjusted in such a way that it produce perfect result in according to the training data, but when tested with test data it overfits , which can be avoided using weight decay.my test error got decreased after using regularization which is evident using the above mentioned graphs.

By using sparseness only limited features were detected as given above avoiding neurons learning all weights.