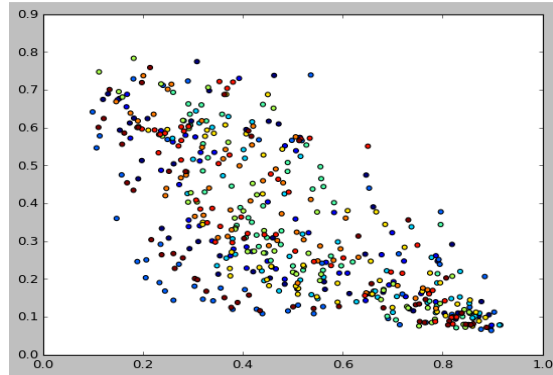
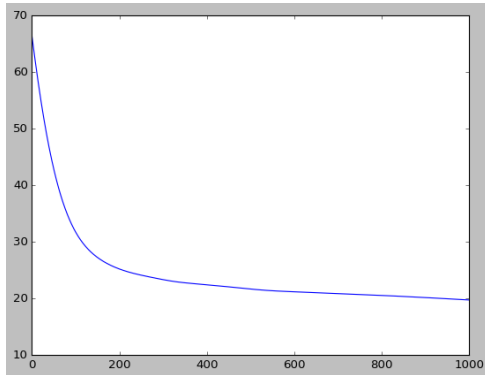
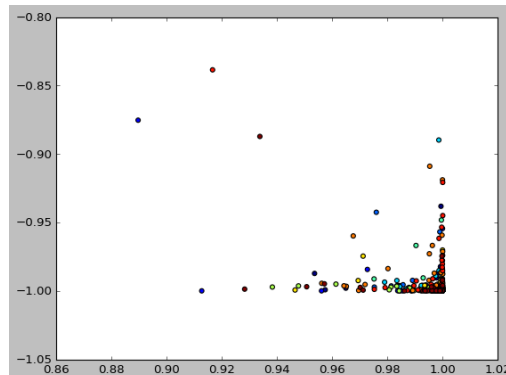
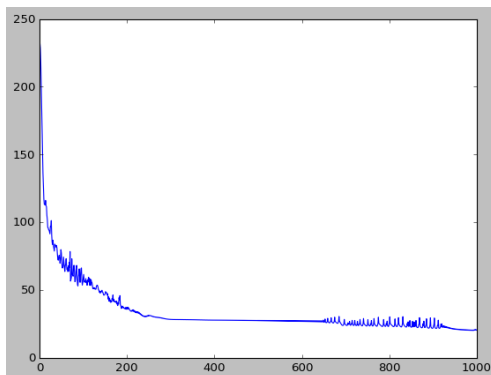


Q2) Error history and Encoded data plots:

For sigmoid:



For hyperbolic tangent:



Q3:

PART A)

- a) classification error on the test data as a function of the number of neurons/units in the hidden layer for all 3 activations:

Number of idle layer neurons	Rectifier Linear activation	Hyperbolic tangent	sigmoid
10	0.0214285714286	0.728571428571	0.0357142857143
30	0.0214285714286	0.292857142857	0.0214285714286
100	0.0214285714286	0.307142857143	0.00714285714286

b) $C = 1$, $\text{tol} = 0.0001$ and $\text{max_passes} = 1000$, 1 vs all SVM yields an accuracy of 0.985714285714.

c) 1 vs all Logistic regression gives an accuracy of 0.957142857143.

d) After applying PCA to reduce dimensions = 100:

Number of idle layer neurons	Rectifier Linear activation	Hyperbolic tangent	sigmoid
10	0.9	0.985714285714	0.971428571429
30	0.9	0.978571428571	0.985714285714
100	0.9	0.985714285714	0.978571428571

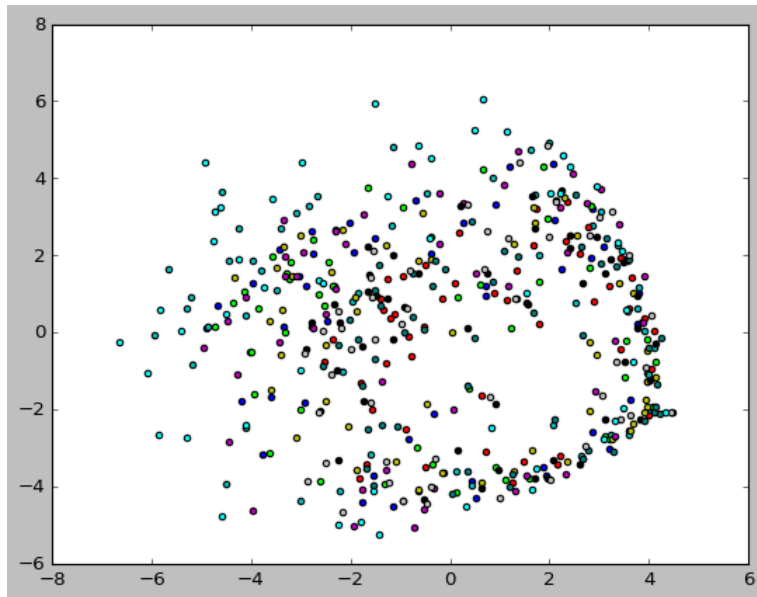
For $C = 1$, $\text{tol} = 0.0001$ and $\text{max_passes} = 1000$: SVM accuracy = 0.0214285714286

Logistic regression accuracy: 0.0142857142857

The results obtained by classification through Neural Networks, SVM and Logistic Regression have lesser accuracy with reduced dimension data.

e)

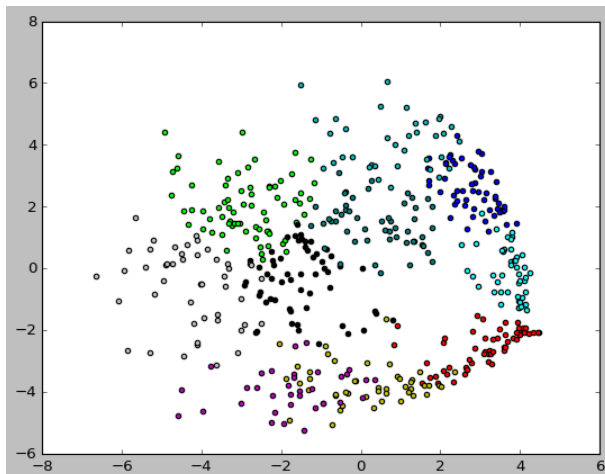
f) After applying PCA to reduce dimensions = 2, the data is not separated according to classes



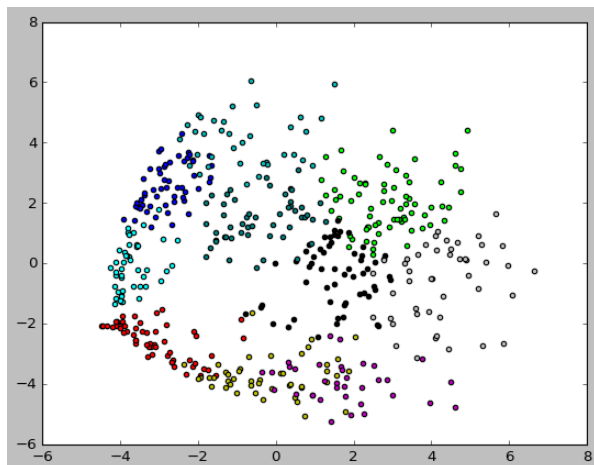
PCA is based on extracting the axes on which data shows the highest variability. Although PCA “spreads out” data in the new basis, there is no guarantee that the new axes are consistent with the discriminatory features in a classification problem.

g) Clustering Error ratio with Kmeans: 0.886

With PCA, $d = 2$:



With PCA, $d=100$:



It can be seen clearly that Kmeans fails in recovering the true clustering of the data, because Euclidean distance cannot represent similarity between image features. Also,

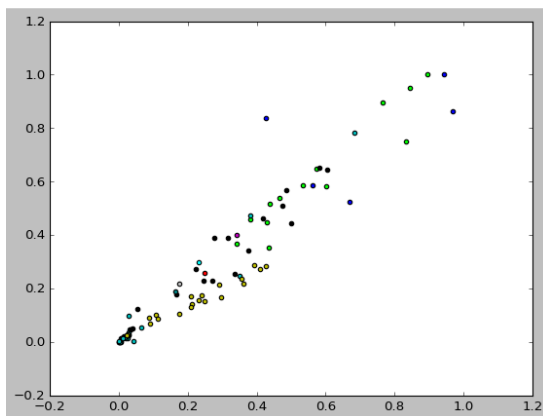
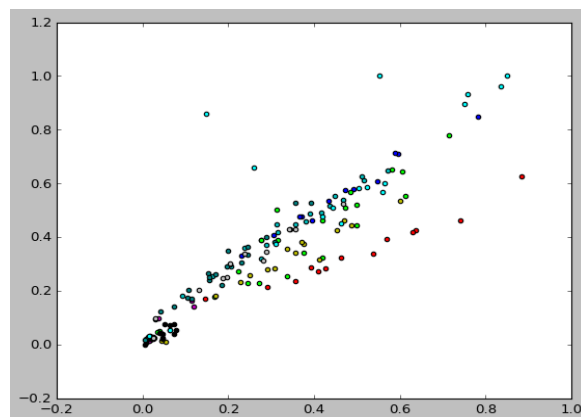
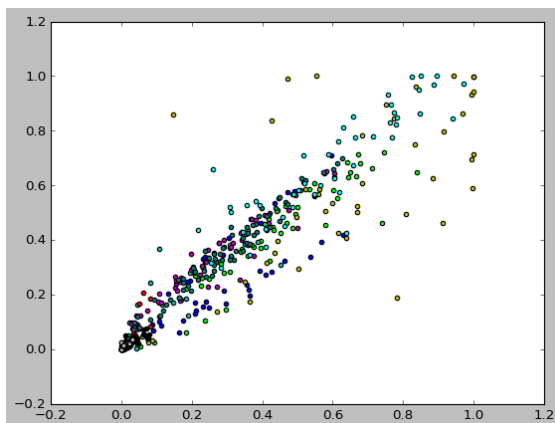
Euclidean space is *linear* which implies that small changes in the data result in proportionately small changes to the position of the cluster centroid.

h)

K	gamma = 0.0001	gamma = 0.001	gamma = 0.01	gamma = 0.1	gamma = 1	gamma = 10
10	0.946000	0.952000	0.914000	0.884000	0.930000	0.892000
30	0.996000	1.0	0.932000	0.962000	0.940000	0.904000
100	0.990000	0.996000	0.988000	0.978000	1.0	1.0

The above table has error ratios for various K and sigma. As seen from the plots below, Spectral Clustering performs no better than SVM, because RBF kernel does not give full information about the structure of data.

For K = 10, 30 and 100 respectively and gamma = 0.0001:



For $K = 10, 30$ and 100 respectively and $\gamma = 10$:

