COMP4107 – Neural Networks

Assignment 3 Answers

Members: Krystian Wojcicki, Michael Kuang

Notes:

Extra libraries required:

minisom, https://github.com/JustGlowing/minisom, pip install minisom

Depending on instillation pillow may also need to be required (should be downloaded when installing tensorflow however)

https://github.com/python-pillow/Pillow , pip install Pillow

1.

a)

The graph below shows accuracy vs training samples for the one-shot learning as described in class. As we can see the accuracy quickly decreases as the number of training samples goes up. This seems to make sense as the Hopfield network can only learn a limited number of patterns, and as more training images are used the learned patterns become more and more degenerate until the network essentially flips a coin to decide what the output is.

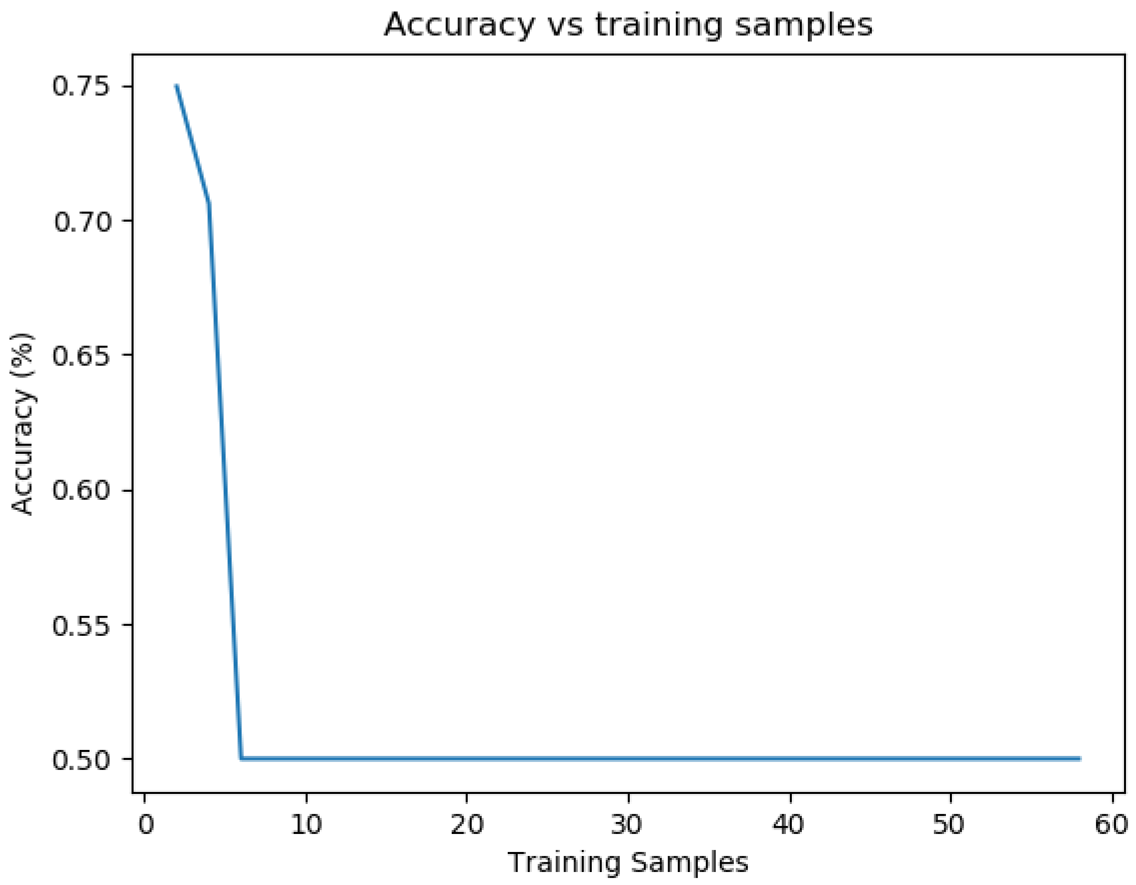


Figure 1: Comparing average accuracy over 5 runs against training samples using regular Hopfield network

b)

Using Storkey's learning method we can see that the number of patterns that can be stored is much higher as indicative of the higher accuracy and that the network does not have as big of an issue with using more training images while training.

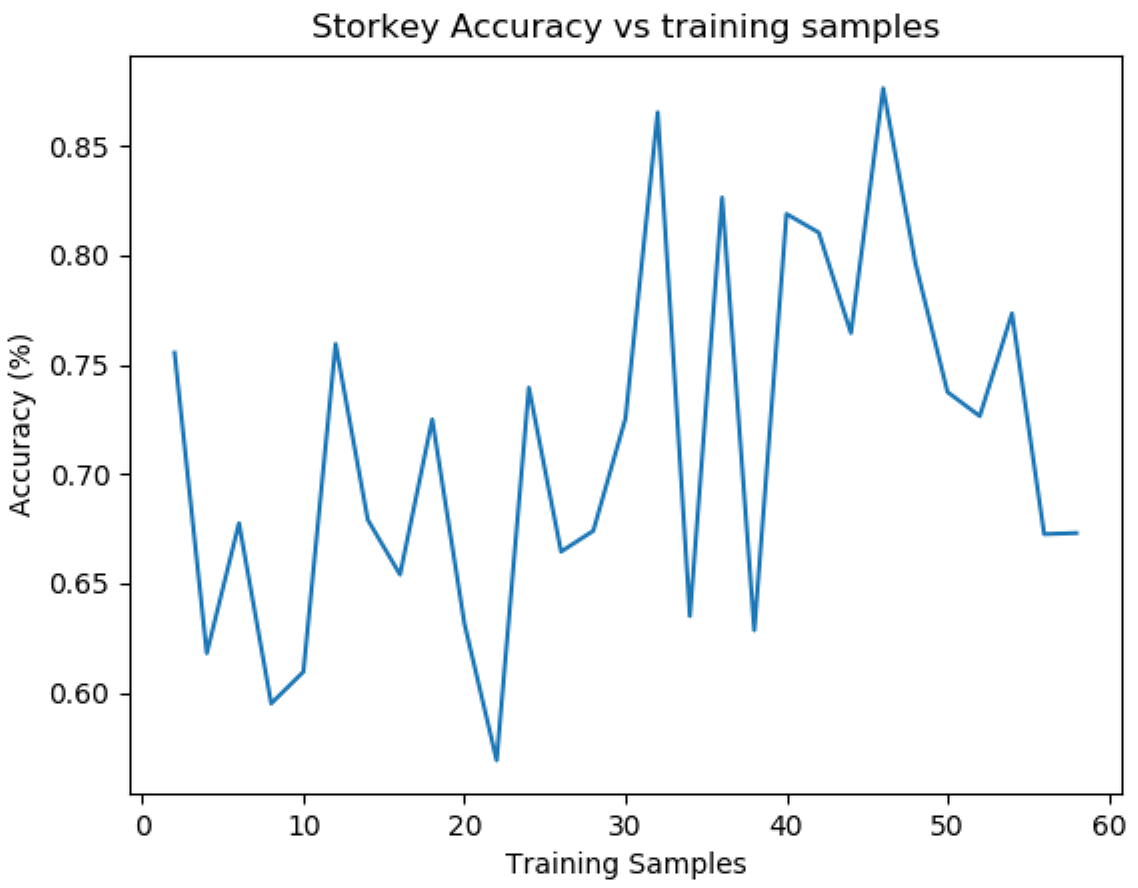


Figure 2: Comparing average accuracy over 5 runs against training samples using Storkey Hopfield network

2.

a)

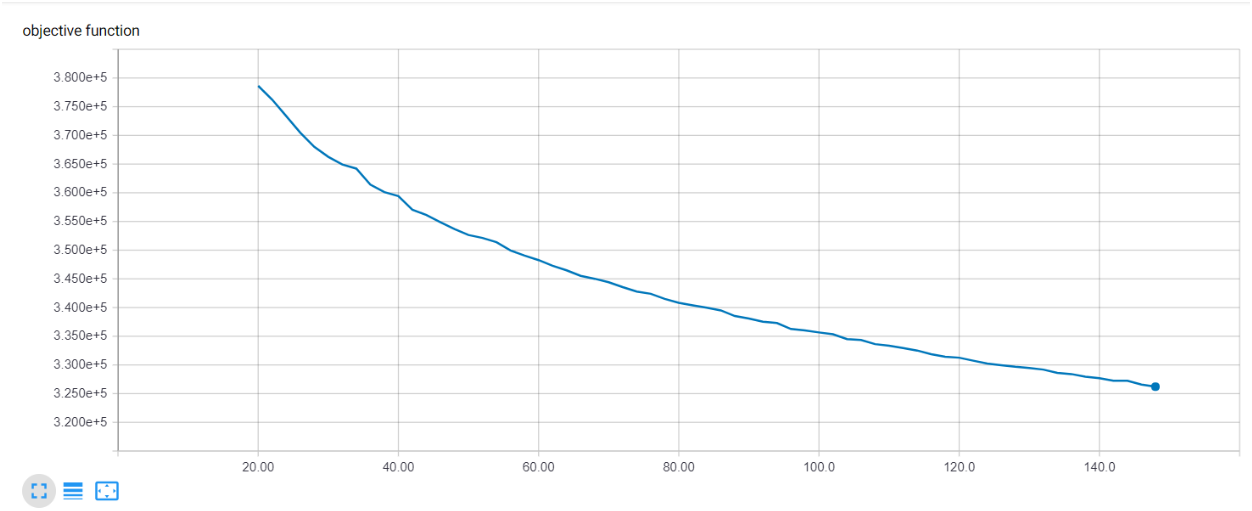
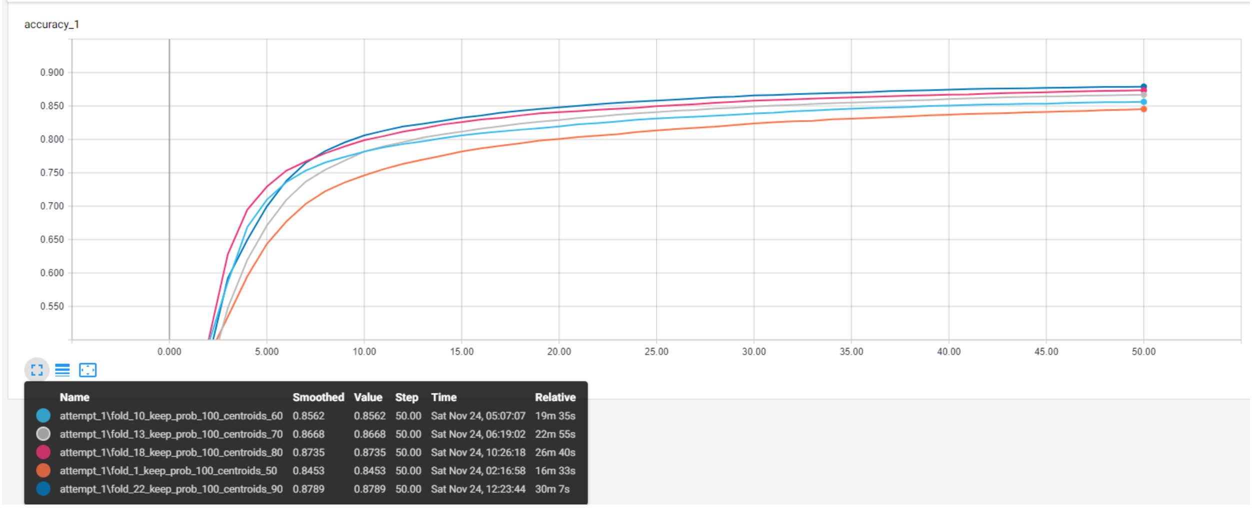


Figure 3: Elbow finding graph, y-axis denotes value of objective function, x-axis denotes number of centroids

We ran the elbow finding experiment from k=20 to k=150 with increments of 2. We see that the graph is more like a curve. At the start, it is steeper and right around 60 to 70 we see that the graph slows down. So, we think it is a reasonable assumption that choosing around 70 hidden neurons for our hidden layer is an optimal choice.

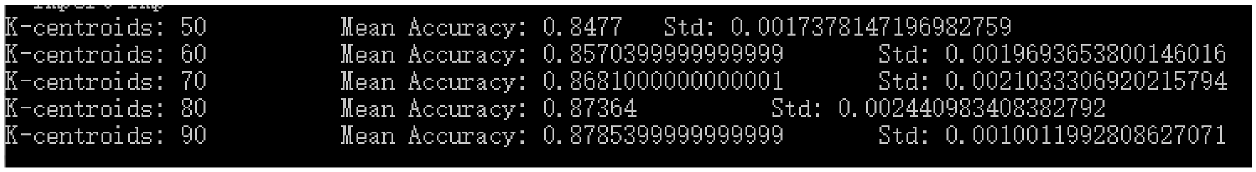
b) In our implementation of the code, you will see that we used 5-fold cross validation (CV) when we run our experiments to investigate the performance on our neural network for different hidden layer sizes and different dropout percentages. For each fold the same centroids were used.

c) We ran a 5-fold CV for hidden layer sizes of 50, 60, 70, 80, and 90 for 50 epochs each due to time constraints. As such, we did not train till convergence.



	Name	Smoothed	Value	Step	Time	Relative
●	attempt_1\fold_10_keep_prob_100_centroids_60	0.8562	0.8562	50.00	Sat Nov 24, 05:07:07	19m 35s
●	attempt_1\fold_13_keep_prob_100_centroids_70	0.8668	0.8668	50.00	Sat Nov 24, 06:19:02	22m 55s
●	attempt_1\fold_18_keep_prob_100_centroids_80	0.8735	0.8735	50.00	Sat Nov 24, 10:26:18	26m 40s
●	attempt_1\fold_1_keep_prob_100_centroids_50	0.8453	0.8453	50.00	Sat Nov 24, 02:16:58	16m 33s
●	attempt_1\fold_22_keep_prob_100_centroids_90	0.8789	0.8789	50.00	Sat Nov 24, 12:23:44	30m 7s

Figure 4: Comparing Testing Accuracy at each epoch for different hidden layer sizes with 5-fold CV



```
K-centroids: 50         Mean Accuracy: 0.8477    Std: 0.0017378147196982759
K-centroids: 60         Mean Accuracy: 0.8570399999999999        Std: 0.0019693653800146016
K-centroids: 70         Mean Accuracy: 0.8681000000000001        Std: 0.0021033306920215794
K-centroids: 80         Mean Accuracy: 0.87364           Std: 0.002440983408382792
K-centroids: 90         Mean Accuracy: 0.8785399999999999        Std: 0.0010011992808627071
```

Table 1: Accuracy mean and standard deviation of 5-fold CV for k-centroids

Figure 2 is just another graph visualization of the accuracy improvements when increasing hidden layer size. This graph only shows the training accuracies for a single fold for each hidden layer size of 50, 60, 70, 80 and 90. From Table 1, we can see that the accuracy increases as we increase the hidden layer size and this makes sense because with more centroids, the network will naturally be able to separate the inputs as it uses Euclidean distance as the metric for predicting labels. The increase in accuracy past 70 centroids is quite small indicating that ~70 hidden neurons is a good amount of neurons without causing over fitting.

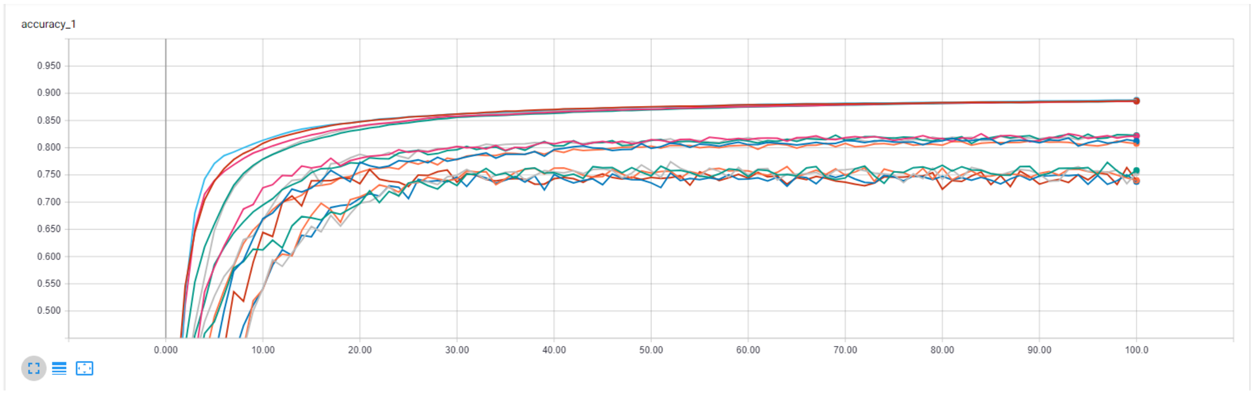
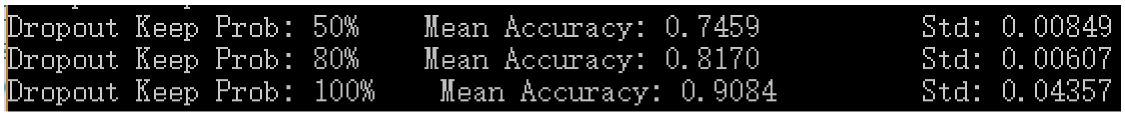


Figure 5: Comparing Testing Accuracy at different dropouts for a hidden layer size of 70 with 5-fold CV



```
Dropout Keep Prob: 50%    Mean Accuracy: 0.7459          Std: 0.00849
Dropout Keep Prob: 80%    Mean Accuracy: 0.8170          Std: 0.00607
Dropout Keep Prob: 100%    Mean Accuracy: 0.9084         Std: 0.04357
```

Table 2: Accuracy mean and standard deviation of 5-fold CV for dropouts

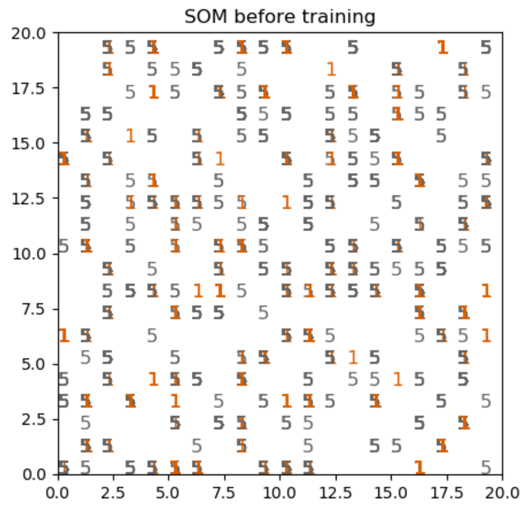
Figure 3 shows the accuracies of each 5-fold cross validation for the different dropout levels at k=70. We see from Table 2 that the lower keep probability or the more neurons we dropout at the hidden layer, the lower the accuracy becomes. This makes sense because for an RBF network, each neuron is an expert already and so the number of clusters, which is also the number of hidden layer neurons, is already indicative of the features in the underlying data. Each cluster is important because the metric for classifying the input image is based upon the Euclidean distance between the input image with each cluster. By dropping out these neurons, we lose that expert and thus we have a lower accuracy. From this we can see that dropout is not necessarily a good technique to add into an RBF network.

3.

a)

In our implementation, we used a dimension of (20,20) for our SOM, 0.9 for sigma and a learning rate of 0.25.

Plots for before training SOM and after training SOM.



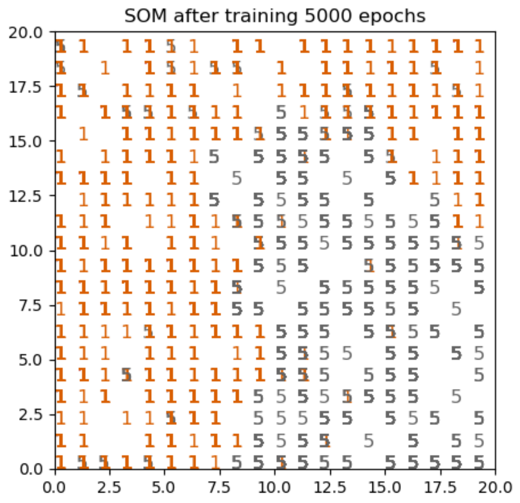


Figure 6: Showing SOM before and after training

As you can see from the before and after shots of the SOM, the 1's and 5's are mostly organized and clustered with each other.

Below are Voronoi graphs for '1' and '5' for K-means solution. We tried from k=4 to k=20 with increments of 2. In the graph, each centroid is denoted as a white circle and the background colors surrounding each centroid show the area that is closest to that centroid. For reducing the kmeans data to a 2D representation PCA was used instead of SVD.

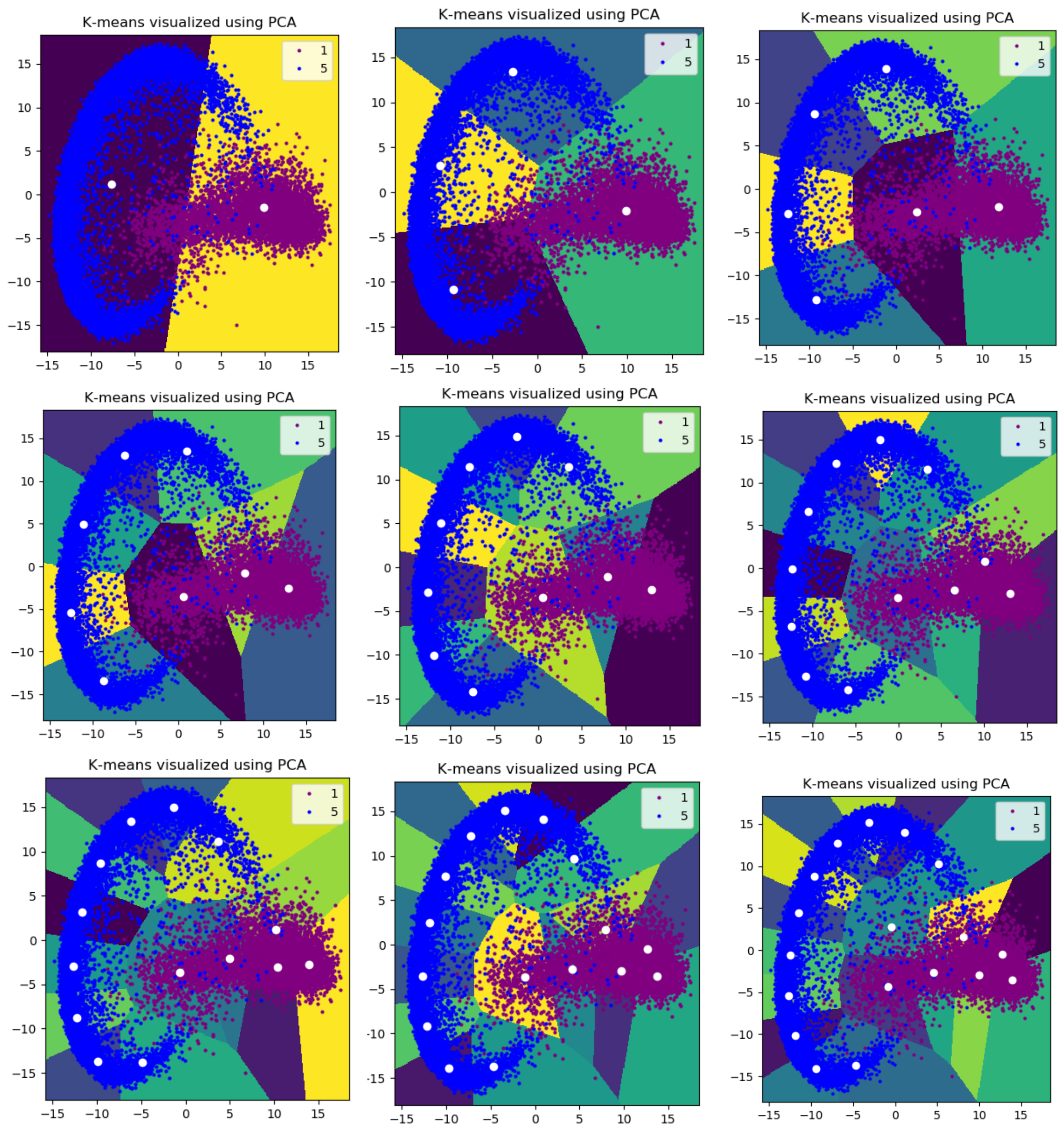


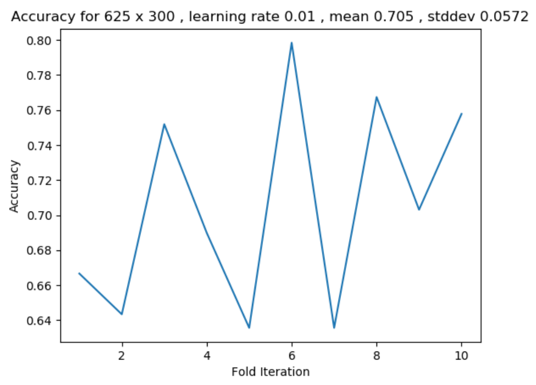
Figure 7: Showing various kmeans clusters using a Voronoi diagram, where white dots are centroids and the colored area around the centroid is the decision area

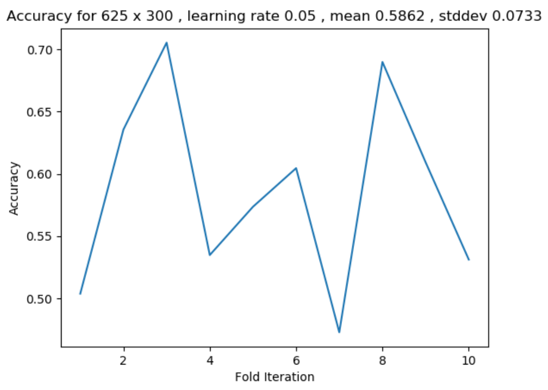
From a qualitative perspective, we see that the network will obviously have a low accuracy if k=2 as seen in the first graph, but with k=10 the graph is well separated and should have a much higher accuracy.

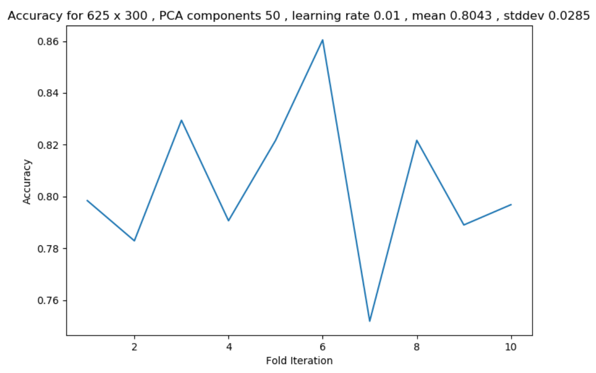
Both SOMs and K-means in general seem like decent clustering techniques. However, for K-means, we need to select an optimal k-value, otherwise the accuracy will not be good. In both cases while there are distinct regions of 1ness and 5ness the boundaries are fuzzy and nether clustering technique is truly able to separate the two classes. It seems SOM does a bit better job of separating the data.

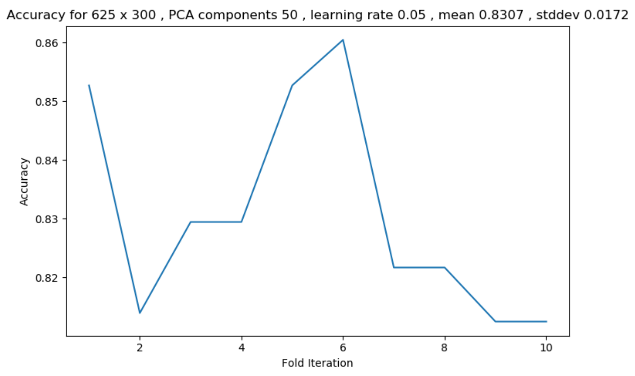
4)

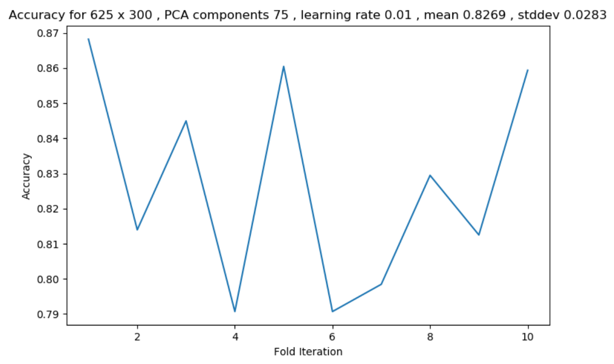
For this question the network consisted of 2 hidden layers either with 625,300 neurons or 400,200 neurons, the activation functions were regular relu not leaky relu and gradient descent was used for learning. Experimentation was done around the learning rate (0.01 and 0.05) and the number of components (50, 75, 100) used in the PCA decomposition. 10-fold experimentation was done for 100 epochs.

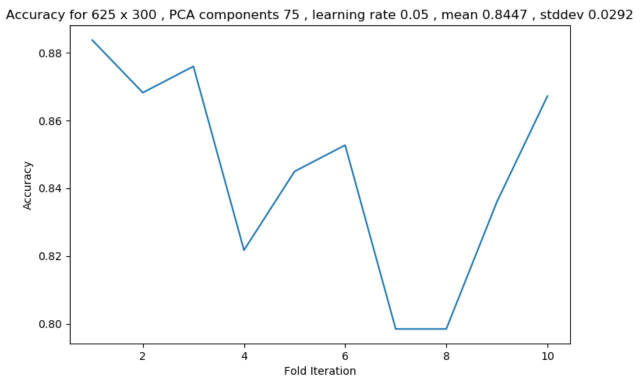


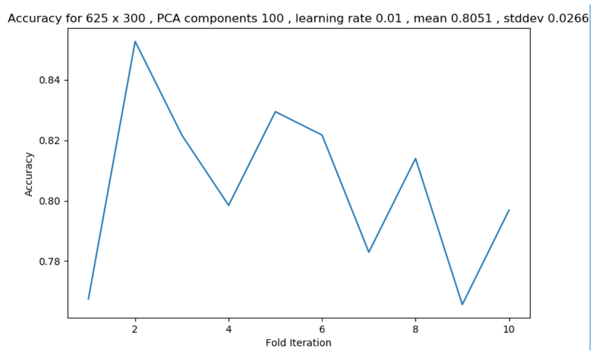












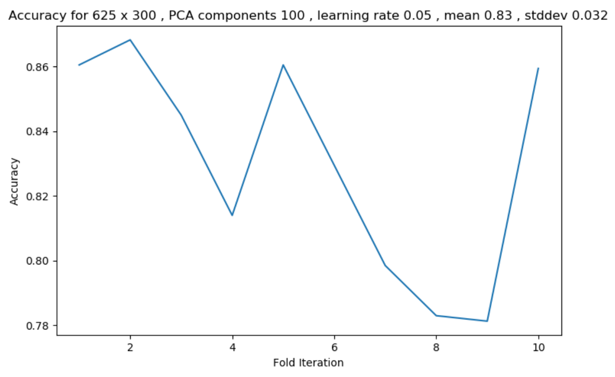
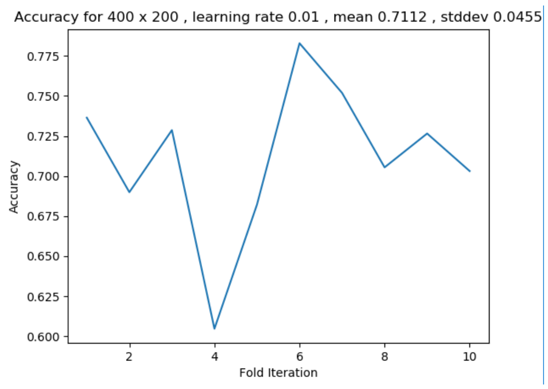
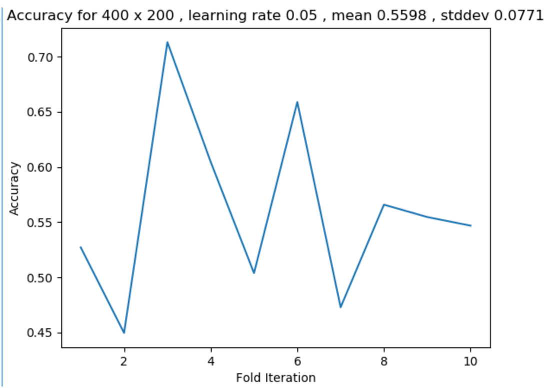
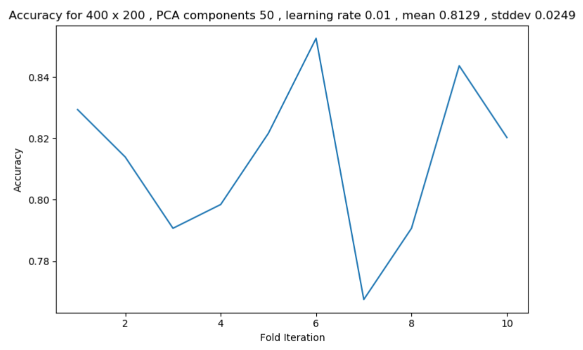


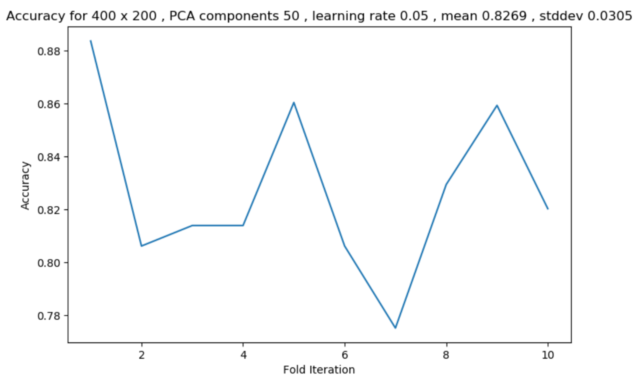
Figure 8: Showing accuracy for various PCA component sizes and learning rates with a network of size 625, 300

With a network of size 625 by 300 it is quite clear to see that doing PCA on the input greatly increases the accuracy of the network regardless of the number of components used. With respect to the optimal number of components that seems to be 75, meaning 50 is likely under fitting the data to some degree and 100 components over fits the data in both cases this causes a slight decrease in accuracy. Using a learning rate of 0.05 seems to be better as well.









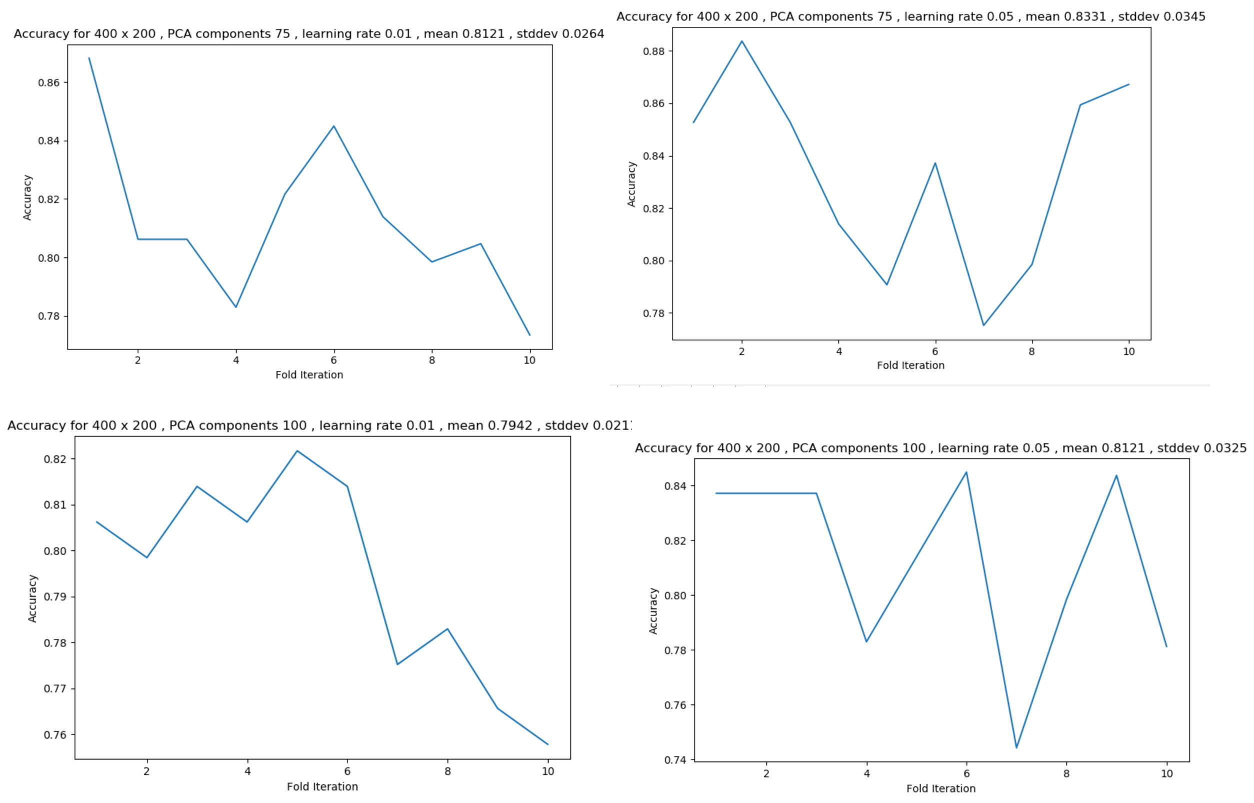


Figure 9: Showing accuracy for various PCA component sizes and learning rates with a network of size 400, 200

The results with a network of size 400,200 were quite like that of the network of size 625, 300. Using PCA greatly increased the accuracy but in this case 50, 75 components offered similar accuracy.

Overall the larger network that used PCA was more successful which makes sense as like SVD, PCA obtains the most important hidden features allowing the network to be trained on the most important part of the data.