# Report 6, Seqian Wang

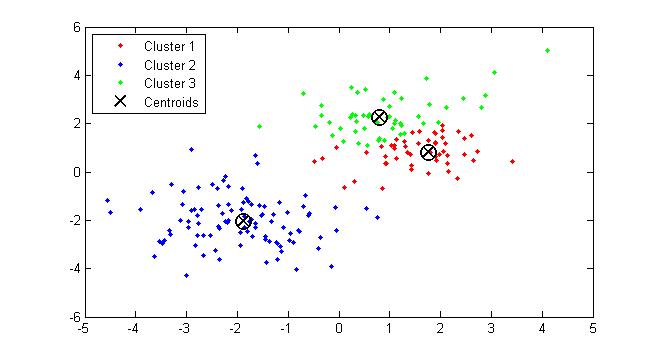
In collaboration with Sulantha Mathotaarachchi and Maxime Parent

# Part 1

## Part 1a: 2 Distributions, 3 Clusters

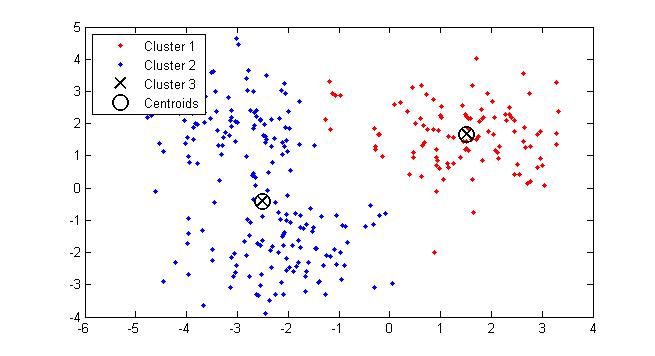
% Comment out one of the distribution in Matrix X in k-Means clustering code (line 1-3)

With 2 distributions but 3 clusters, the k-means erroneously splits one of the distribution into two clusters (in the case below, red and green).



## Part 1b: 3 Distributions, 2 Clusters

% Change kmeans second input from 3 to 2 (line 6)



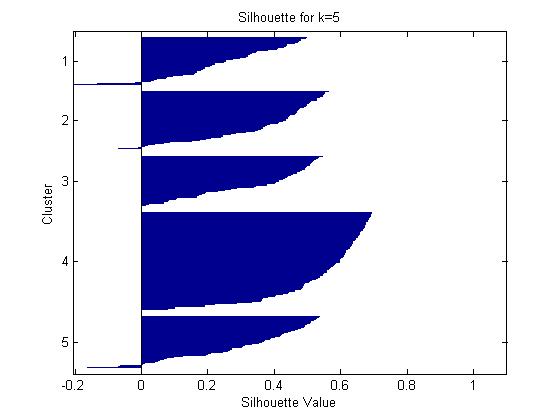
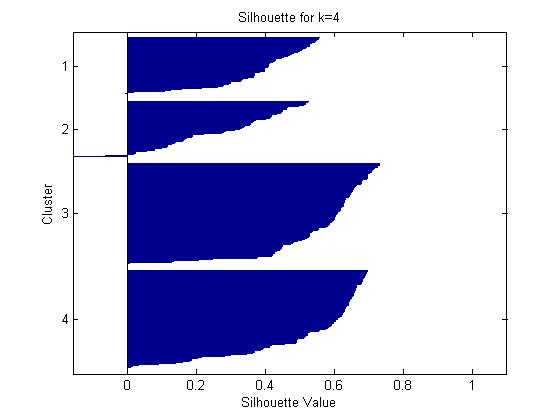
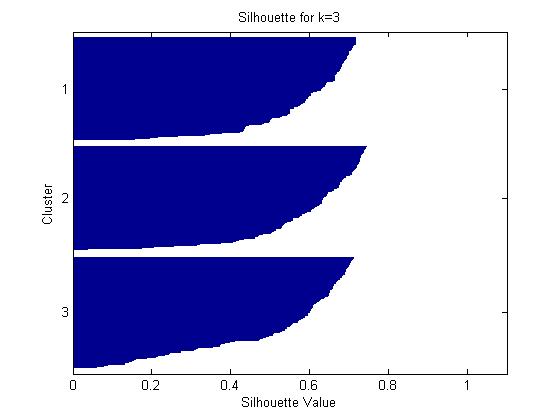
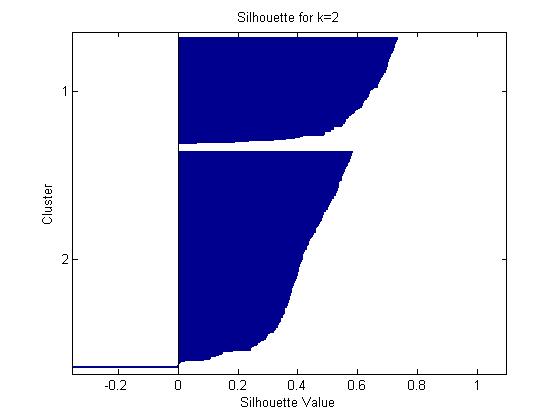
Conversely, a k too low will merge two distributions together. In this case, the blue cluster includes two distributions, one which is in the negative y values and one in the positive y values.

## Part 1c: 3 Distributions, 2 to 5 Clusters

% Created a for loop that calculates k-means with different k

Since the outcome of a k-means clustering can vary from run to run (random starting points), we increased the k-means replications to 100 to identify the global minimum.

A silhouette plot shows how distant each point in a given cluster is to points in neighboring clusters (+1 is very distant, 0 shows no distinction with neighboring clusters, -1 suggests that the point is possibly erroneously classified). Compared to the other silhouettes plot, the one with k=3 does not have negative values, has higher distance values, and the silhouette is “broad”, suggesting that many of the points are properly assigned.

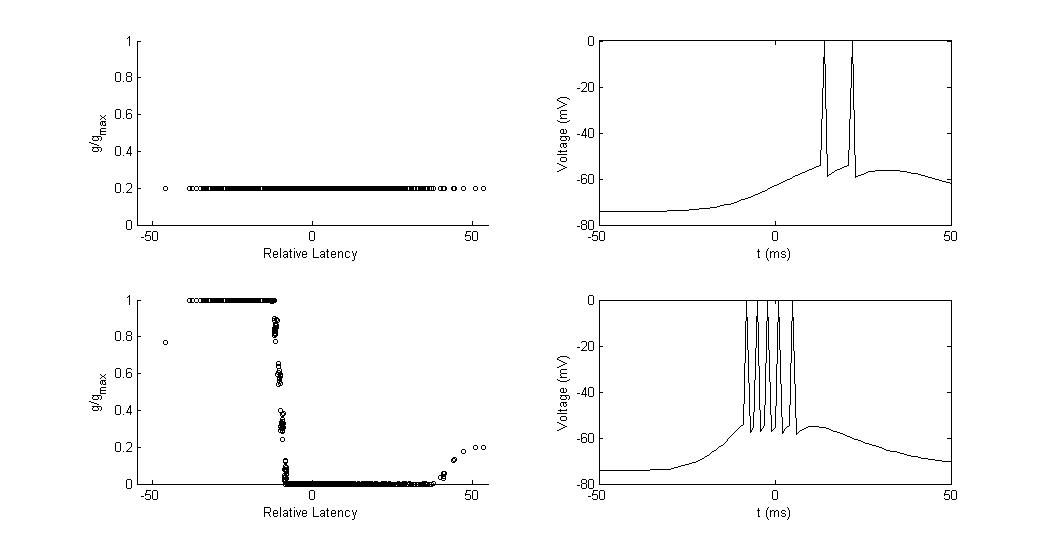
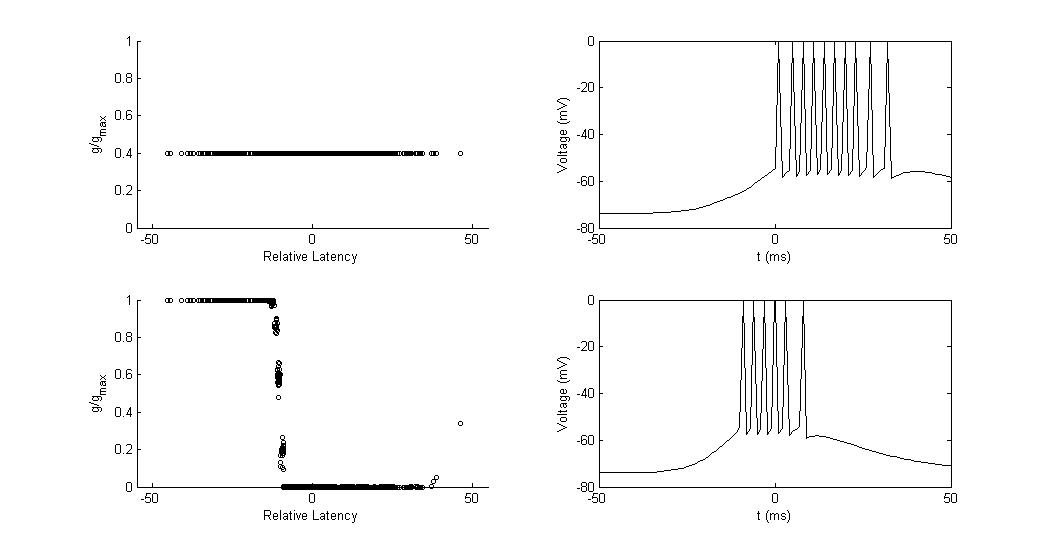


# Part 2

## Part 2a: Reduced Latency and Sharpened Response

% On line 5 of imSTDPlatencies, “g = ones(N,1).\*0.003;”, change 0.003 to 0.006

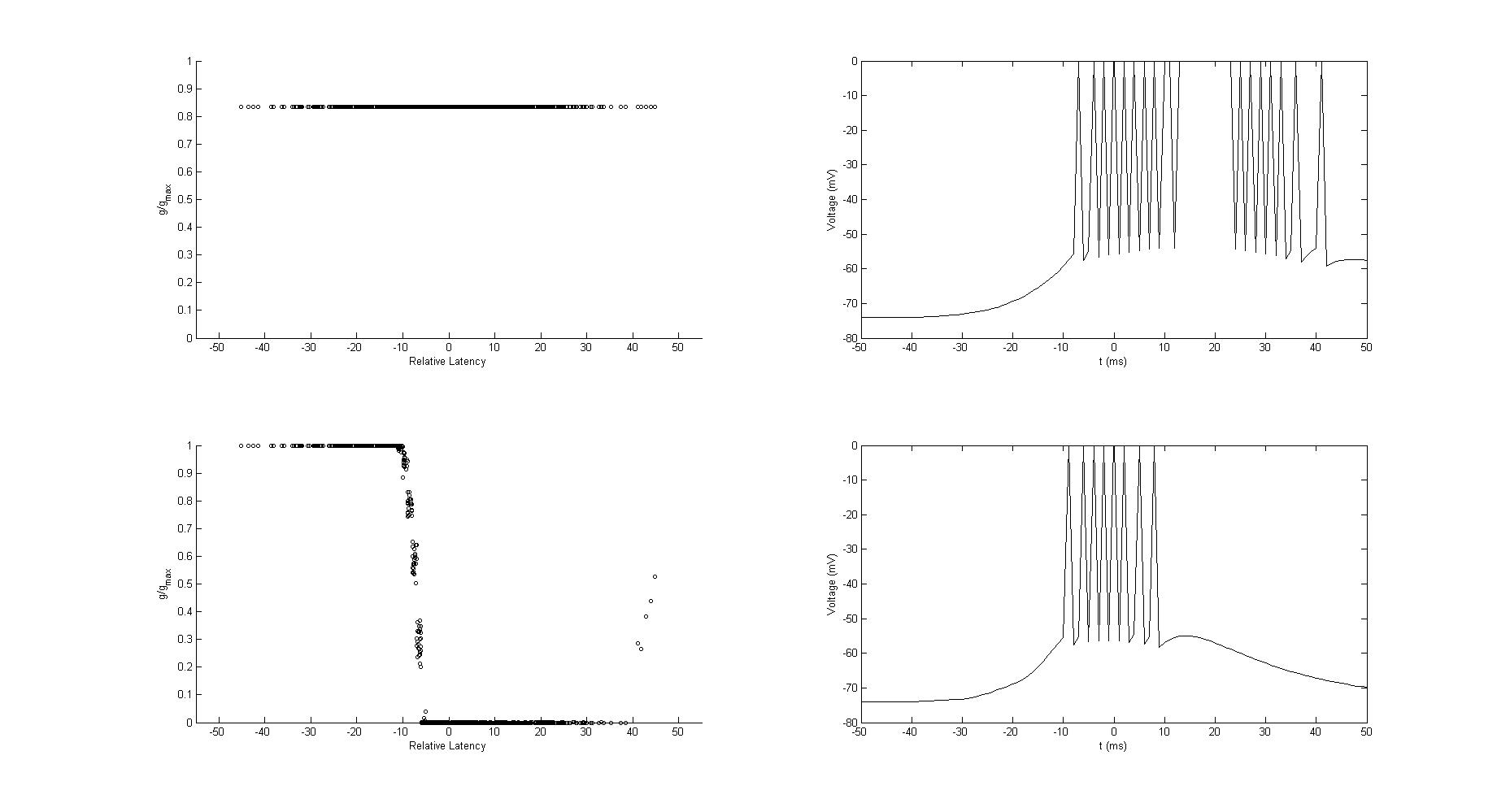
Before (g = 0.003) After (g=0.006)



After doubling the initial conductance, the greatest change is in the initial postsynaptic response to excitatory input. First, a lot more action potentials are fired (synapses are strong enough to drive postsynaptic responses). Second, the response occurs much earlier, with a relative shorter latency. Finally, after the STDP, all the low-latency input neurons have maximal conductance.

## Part 2b: Higher Stability

% On line 5 of imSTDPlatencies, “g = ones(N,1).\*0.0125;” instead of 0.003 (gmax is 0.0150)



Even with a super high initial synaptic conductance, a STDP forces a stable distribution and ensures that the postsynaptic neuron remains sensitive and reliant on presynaptic action potentials. Indeed, the conductance of high-latency neurons are severely decreased following the STDP.

## Part 2c: Unsupervised Learning and Classification

Unsupervised learning includes clustering and/or data classification. Here, the STPD segregates synaptic inputs into strong and weak groups and affect their conductance. The synapses that fire in a correlated way and that succeed at evoking a postsynaptic action potential within the STDP time function will be classified differently from those who don’t, and thus perform unsupervised learning or classification. When the neurons are not correlated, the conductance is effectively random. When a group is while another isn’t, the correlated group will gain maximum conductance, while the other will lose conductance. Finally, when two sets or groups of neurons are correlated within each group but not to the others, a competition occurs, and the winner takes all, forming strong connections with the postsynaptic neuron.

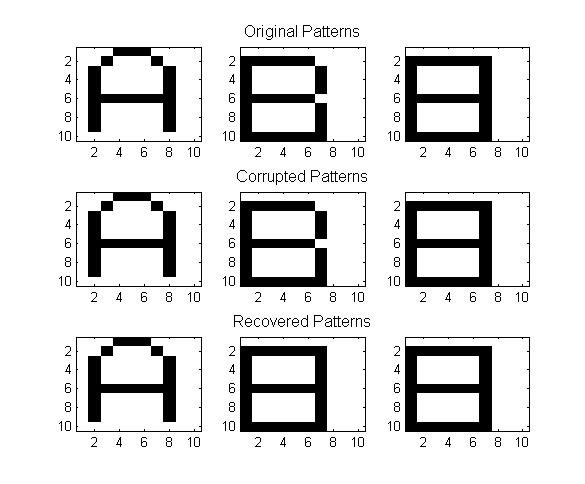
# Part 3

## Part 3a: Local minima and ghost states

With a network size of 100 and a noise level below 20, the network successfully converges back to the original patterns. With a noise level above 70, the recovered patterns are the inverse of the original ones (ghost states). This is because noise generation becomes saturated beyond a certain level. Between noise level 20 and 70, the recovered patterns are most of the time different from the original patterns. From the corrupted patterns, the network will stabilize around a minimal state. In these cases, the local minimum reached (recovered pattern) is the closest to the corrupted pattern, but distant from the original pattern.

## Part 3b: Memory Alteration

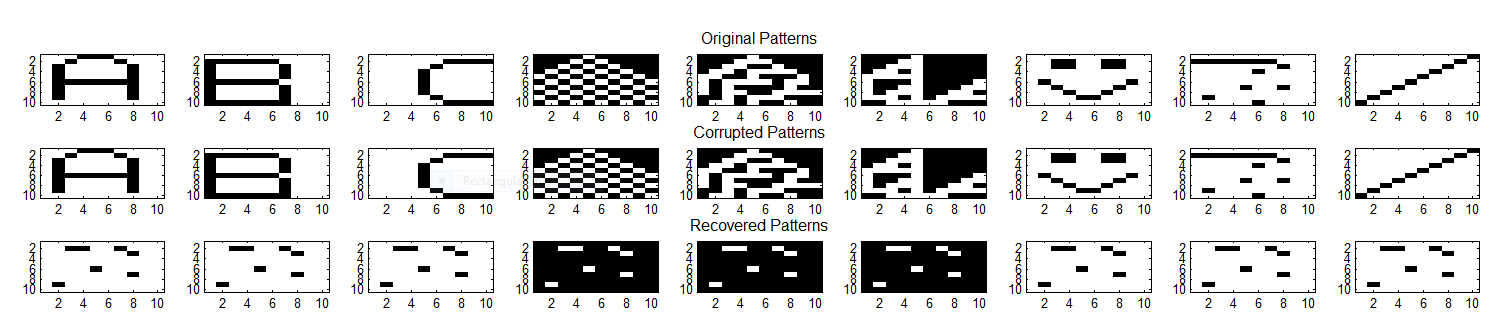
% Changed C into an 8, similar to B pattern



By having the third memory very similar to the second one, the letter B converges to the newly available attractor state (number 8), even with no noise level (as shown above). This is because the new attractor state is more stable and “attractive” than letter B pattern.

## Part 3c: Memory Addition

The script hopfield\_net.m returns an error, because the number of combinations to be drawn (512, line 92) exceed the allowance number of subplots (81, line 93). By commenting out the combinations plot section (line 90 to 104), we can generate the following patterns, suggesting that the recovered pattern is a global minimum that attracts all other forms, even with no noise for the corrupted pattern.

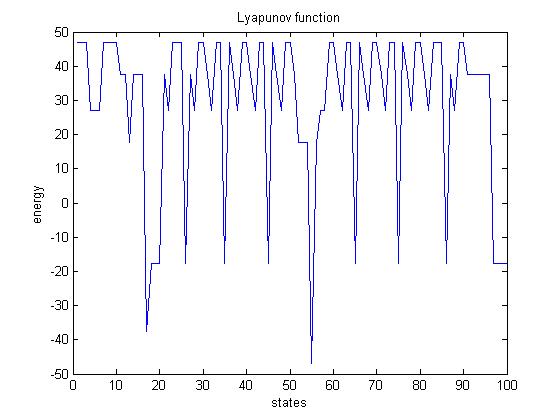


As for the network storage limit, we know that its maximum (p) follows this equation:

where N is the network size. Since our network has 100 neurons, we can only have 12.5 patterns stored at a given time.

## Part 3d: Network energy convergence

% plot(states'\*weights);



With a neural network of 100, a noise of level 20 and the provided memories (unmodified mem\_ABC.txt), the figure above shows a global minimum and convergence at state 55.

The Hopfield network relies on attractor states and energy levels to categorize memories. The network measures the similarity between given memories by converging similar ones towards the same state, thus clustering and classifying the various inputs into groups.