Part 1 ‚Machine Learning: k-Means Clustering (3 points)

Run the k-Means clustering code using kMeans;

This code makes three two-dimensional distributions and clusters them with k=3.

A. Make two distributions but cluster with k=3

Investigate how the k-Means algorithm behaves when k is erroneously set too high.

Report and describe.

When k is too high, there are too many centers where there should be none.

B. Make three distributions but cluster with k=2

Investigate how the k-Means algorithm behaves when k is erroneously set too low.

Report and describe.

When k is too low, the algorithm clumps the points into only two sets where there

clearly should be three.

C. Make three distributions, let your code determine k

Use e.g. the silhouette measure to determine the ideal k, trying k=[2,3,4,5].

For a starting point, see: http://www.mathworks.com/help/stats/k-means-clustering.html

You may want to use the replicates option for this section.

From my calculations, I found that the optimal k=3.

Part 2 ‚Learning in the Brain: STDP (3 points)

Simulate Fig 4 in Song et al 2000 and Fig 1 in Song et al 2001 using

Song2000\_F4;

Song2001\_F1;

respectively.

A. Song et al say STDP reduces latencies and sharpens responses

Our Song2000\_F4 simulation code shows that STDP reduces latencies

but it does not actually sharpen the response.

This is because the starting setting of the synaptic

conductance g is too low, so synapses are initially too

weak to properly drive the postsynaptic neuron.

Find this setting for initial g in simSTDPlatencies, double it,

and explain what is different.

A collage of diagrams

Description automatically generated

Figure 1 Double g

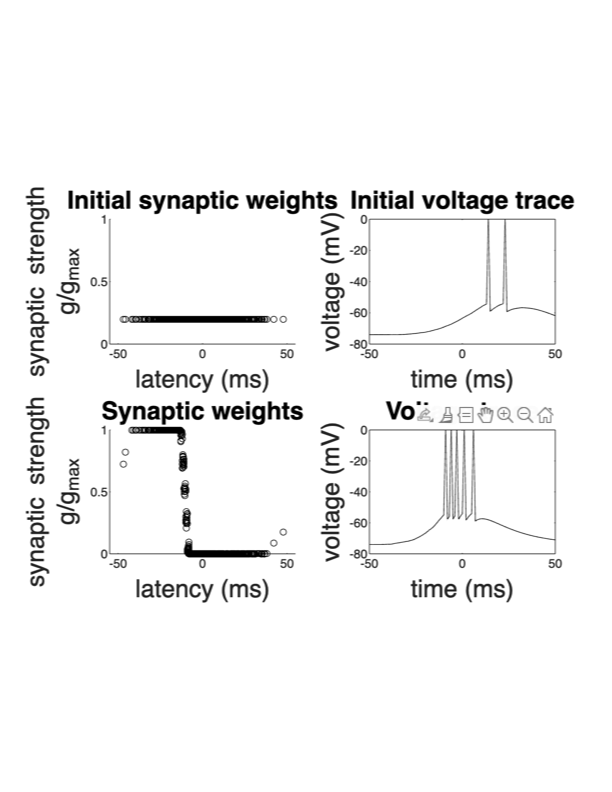


Figure 2 Original g

When the initial g (conductance) was doubled, there were 6

spikes in the voltage trace instead of five. There were also

10 spikes in the initial voltage trace instead of 2.

B. STDP provides a degree of stability

Keep increasing the starting value of the synaptic conductance

g to something unreasonably large, as indicated by the postsynaptic

neuron saturating (reaching depolarization bloc).

Run the Song2000\_F4simulation and compare the before/after activation pattern.

Show the resulting graphs. Explain how STDP can provide a degree of stability

to a neuron.

A collage of diagrams

Description automatically generated

Figure 3 10000000000000 g

By making the conductance unreasonably high, such as increasing it by a factor of 10 or 100 or even higher, the initial voltage trace becomes longer and longer and does not stop. Conversely, with a high conductance factor of 10000000000000 of the original conductance, there are initially way too many spikes in the voltage trace, but the spikes are eventually reduced to eight (more than the initial 5 but not needlessly excessive)

which demonstrates the role of STDP in preventing overexcitability in neurons by reducing post-synaptic receptiveness in response to overactive pre-synapse action potentials.

C. Without instruction, STDP can pick up on correlations in inputs

The simulation code Song2001\_F1 recapitulates the findings in

Fig 1 of Song et al 2001. Run the code. Explain how this is

unsupervised learning and classification.

Note: the outcome in the bottom right panel is random and so

you may get quite different results if you run it a few times.

In the system where half of the neurons are correlated and the other half

aren't, the correlated neurons always maximize in strength while the uncorrelated ones always minimize. This classification of neurons occurs based on the pattern of how they are connected, and which doesn't require outside input which is why it is unsupervised. The system where both halves are uncorrelated shows that no classification can occur if there are no similarities. The system where both halves are correlated shows how that classification can be

random if there are not significant differences between the input data for neurons.

Part 3 An Associative Memory: the Hopfield Network(4 points)

Run the network using: `hopfield\_net(100,'mem\_ABC.txt',10,1,1);`

This sets the network size to 100. The code will pick the memories

from the specified text file mem\_ABC.txt, from which it will create

the attractor states. The number 10specifies the amount of noise it

will use to corrupt the memory states with. The number 1after that is

a Boolean flag that tells the code to rescale the image bit values

in the text file from [0,255] to [-1,+1], so that you can import

images from e.g. ImageJ, or other image processing software packages,

which do not normally operate with negative pixel values. The last number

1 is also a Boolean flag if set to True (to value 1), it indicates

that you want to see the asynchronous updating of the state vector as

it converges to a memory. Set it to False (value zero) to make the code run faster.

A. Experiment with noise levels to find local minima and ghost states

Alter the noise levels until the network no longer correctly recalls

the stored memories. What does the network converge to instead? Try

several times with different noise levels until you obtain other

attractor states: document these states and explain why they happen,

and what they are.

For low noise, the original patterns are recovered. For high noise, the inverse patterns are recovered for the corresponding letters. This occurs because high noise virtually inverts the current colors which makes the inverse patterns more likely.

A group of black and white letters

Description automatically generated

Figure 4 Noise of 20

A group of black and white symbols

Description automatically generated

Figure 5 Noise of 50

A group of black and white symbols

Description automatically generated

Figure 6 Noise of 90

B. Alter one memory

Alter the text file so that e.g. the letter ‘C’ is no longer one of the attractor states, but store some other state, e.g. another character.

Next try altering this memory so that is like one of the other two stored attractor states.

What happens? Explain why.

A group of black and white letters

Description automatically generated

Figure 7 Partial States

A group of black and white letters

Description automatically generated

Figure 8 Similar Letters

Making the 'C' like an 'E' which is like the 'B' frequently makes the 'B' turn into an 'E' and the 'E' turn into the 'B' because the energy difference between the 'B' and the 'E' is very small. I also tried turning the 'C' into an 'L' which causes many partial final states

which probably occurred because there was not a clear enough energy difference between the 'L' and the partial states.

C. The information storage limit of a Hopfield network

Add new well-separated memories to the Hopfield network, so that you have nine in total.

Analytically calculate the theoretical storage limit of the network.

Memories were adjusted and added so that there are nine total. The storage limit was not calculated.

A group of black and white squares with numbers

Description automatically generated

D. Calculate numerically the network energy as it converges

Use the Lyapunov function to show numerically how the network is attracted to

a lower energy state as it converges. Explain how the Hopfield network can be

used as a classifier to cluster data into memorized categories.

The Hopfield network can store a series of memories. The network uses

the Lyapunov energy function and the two-way connections between neurons to classify real-world data to these memories by seeing which memory is closest energy-wise to the real-world data.

Lyapunov functional analysis was not performed.