Computational Neuroscience Assignment 2 (2019)

MPhil in Computational Biology

March 6, 2019

If there are errors found, I will update the assignment on the web at `http://www.damtp.cam.ac.uk/user/sje30/teaching/cn`

Due date: 2019-04-05 23:45

Please submit your report to the MOODLE website as a PDF, anonymised as before.

Your report must be a maximum of fifteen pages, excluding the appendix. Your appendix should contain only a copy of your code.

This assignment is worth 50% of your overall mark for this module.

Items marked **(Advanced:)** should be attempted if you have time and are confident with your work on the rest of the assignment.

[Thanks to Peter Dayan and Larry Abbott for providing these questions `http://www.gatsby.ucl.ac.uk/~dayan/book/exercises.html`. Chapter numbers and equation numbers refer to the Theoretical Neuroscience (TN) book – copies available in the BGML.]

1 Unsupervised learning [TN 8.4, 8.5; 10 marks]

1.1 Multiplicative normalisation

Construct two-dimensional input data sets similar to those shown in figure 1 and use them to train a two-input, one output linear network using correlation- and covariance-based Hebbian learning rules with multiplicative normalization. Compare the final outcome for the weights with the principal components of the data when the mean of the input distribution is zero and when it is nonzero.

1.2 Subtractive normalisation

Repeat question 1.1 for a data set with zero mean, but using subtractive normalization and saturation. How does subtractive normalisation compare with divisive normalisation?

Start with initial values for the weights that are chosen randomly over the full range from 0 to their saturation limit. When does this algorithm produce a weight vector aligned with the principal component axis of the input data set, and when does it fail to do so? Why does the weight vector sometimes fail to align with the principal component axis?

[Getting started: Here is some R code to generate simulated data from a two-dimensional Normal distribution:]

```
n <- 2000
rho <- 0.7
x <- rnorm(n, mean=0, sd=1)
y <- rnorm(n, mean=rho*x, sd=sqrt(1-rho^2))
plot(x,y,asp=1)
var(x,y)                              #should be close to rho.
```

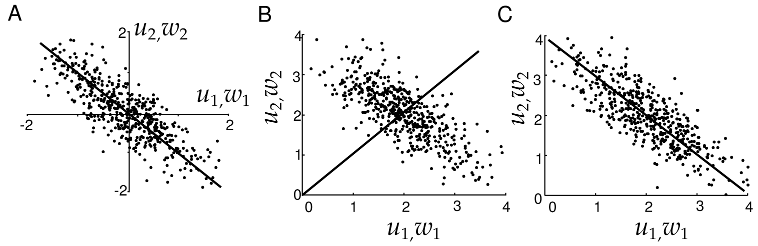

Figure 1: Examples of unsupervised Hebbian learning (same as figure 8.4 of TN).

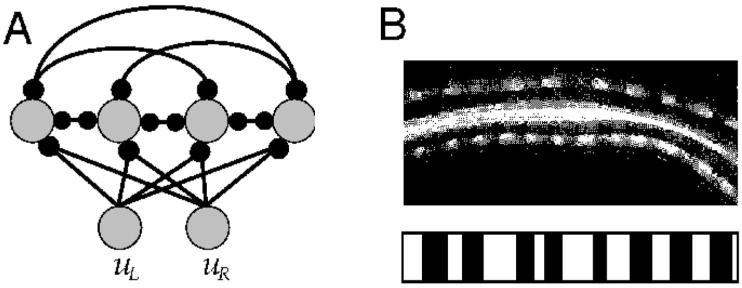

Figure 2: Ocular dominance model: simulate the system in A with two inputs and 512 cortical outputs. (This is figure 8.7 of TN).

2 (Advanced:) Ocular dominance columns [TN 8.3; 15 marks]

Simulate the ocular dominance model of figure 2 using a subtractively normalized version of TN equation 8.31 (i.e. TN equation 8.14) with saturation limits at 0 and 1, and cortical interactions generated from

$$\mathbf{K}_{aa'} = \exp\left(-\frac{(a-a')^2}{2\sigma^2}\right) - \frac{1}{9}\exp\left(-\frac{(a-a')^2}{18\sigma^2}\right)$$

where $\sigma = 0.066$ mm. Use 512 cortical cells with locations a spread evenly over a nominal 10 mm of cortex, and periodic boundary conditions (this means that you can use Fourier transforms to calculate the effect of the cortical interactions). Also use the discrete form of TN equation 8.31, i.e.:

$$\mathbf{W} \to \mathbf{W} + \varepsilon \mathbf{KWQ}$$

with a learning rate of $\varepsilon = 0.01$. Plot $\mathbf{w}_-$ as it evolves from near $\mathbf{0}$ to the final form of ocular dominance. Calculate the magnitude of the discrete Fourier transform of $\mathbf{w}_-$. Repeat this around 100 times, work out the average of the magnitudes of the Fourier transforms, and compare this to the Fourier transform of $\mathbf{K}$. How might you alter stripe width in this model?

3 The elastic net [15 marks]

Simulate the elastic net method for solving the travelling salesman problem (TSP; Durbin and Willshaw, 1987), using your own implementation of the algorithm. Your report should show examples of the elastic net at several points during a simulation. How good is your method at the TSP? You are free to compare your method with other non-neural algorithms, or use an external source such as `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95` for comparison. [10 marks]

(Advanced:) Use the elastic net model to simulate the development of ocular dominance stripes in a two-dimensional model of visual cortex. What influences stripe width? [5 marks]

4 Cart pole balancing problem [10 marks]

Implement the cart-pole balancing system for reinforcement learning (Barto et al 1983). Show how the system performance evolves over time. How sensitive are your results to variations in the key parameters of the model?

Does adding an extra layer of weights improve performance (Anderson 1987)?

References

References available from `https://paperpile.com/shared/Bv9UOO`

Anderson CW (1987) Strategy Learning with Multilayer Connectionist Representations. In: Proceedings of the Fourth International Workshop on MACHINE LEARNING (Langley P, ed), pp 103114. Morgan Kaufmann.

Barto AG, Sutton RS, Anderson CW (1983) Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Trans Syst Man Cybern SMC-13:834846

Durbin R, Willshaw D (1987) An analogue approach to the travelling salesman problem using an elastic net method. Nature 326:689–691.