ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Unsupervised and reinforcement learning in neural networks
Week 6

Lecturer: Marc-Oliver Gewaltig

Assistants: Dane Corneil
Berat Denizdurduran

Kohonen maps

Exercise 1: "Blob formation"

Imagine a 1-dimensional recurrent network with M neurons and cyclic boundary conditions. The dynamics of the neurons is given by

$$y_i(t+1) = g\left(\sum_k w_{ik}x_k(t) + \sum_j B_{ij}y_j(t)\right) , \tag{1}$$

where B_{ij} are the recurrent weights. A neuron receives local excitation from itself and its d neighbours ($d \ll M$) on both sides: $B_{ij} = 1$ for $|i-j| \leq d$, and inhibition from all others: $B_{ij} = -\beta$ with $0 < \beta \leq 1$ otherwise. The activation function $g(h)$ is the Heaviside function, i. e. $g(h) = 1$ for $h \geq 0$ and $g(h) = 0$ for $h < 0$.

1.1. **In class:** Imagine that one single neuron is stimulated and therefore becomes active. This neuron will excite its neighbours and cause an "activity blob". Show that in the steady state of the network, the number N of active neurons is larger than $2d$. **Hint:** You may assume that the active neurons form a "blob" of neighbouring neurons. Consider the net input to the first neuron *outside* the blob and use the fact that this neuron is silent to find an inequality for N.

1.2. How does the strength of inhibition, i. e. the value of β, affect the number of active neurons? What can you say about how many neurons are active in the steady state in the limit $\beta = 1$? **Hint:** Consider the net input to the last neuron inside the blob to get another constraint on N.

1.3. Assume that the input to the third neuron is 1 (and there are no other external inputs to the network). Compute the first three time steps of the network dynamics for $\beta = 1$, $d = 2$ and $M = 10$. Assume that initially all neurons are silent.

Exercise 2: Batch vs. online learning

In the lecture, it was argued that batch learning and online learning are equivalent in the limit of small learning rates. Show that for competitive learning, batch and online learning are also equivalent if the learning rate is decreased in the right way.
Let $\vec{w}_0$ be the initial weight before presentation of the input patterns $\vec{x}^\mu$. For simplicity, let us consider only patterns for which the considered weight is the winner. According to the batch rule, the weight change after the first N input patterns is then given by

$$\Delta\vec{w}^N = \frac{1}{N}\sum_{\mu=1}^{N}(\vec{x}^\mu - \vec{w}^0). \tag{2}$$

Calculate the difference between the weight vector after $N+1$ and after N input patterns and show that this leads to an online learning rule in which the weight change caused by pattern $N+1$ is given by:

$$\Delta\vec{w} = \eta(N)(\vec{x}^{N+1} - (\vec{w}^0 + \Delta\vec{w}^N))\,. \tag{3}$$

How does the learning rate η depend on N?

Exercise 3: Kohonen maps

Use the Matlab or Python files in the archive “exercise6.zip” to explore the behaviour of Kohonen maps.

3.1 Starting from the original values in the function “kohonen”, what happens if you change the size of the neighborhood (i.e. changing the value sigma)? Try reducing and enlarging the neigborhood.

3.2 Again starting from the original values, what happens if you change the learning rate (smaller vs. bigger)?

3.2 What changes if you use a larger Kohonen map?

Technical Note: To use the Python version, you will need the Numpy and Matplotlib packages.