Exercises for week 3
Hebbian rules and ICA

Exercise 1: Clustering

1.1 Show for competitive learning that every prototype is at the center of its group/cluster after convergence (i.e. in the steady state).

Use the (batch) learning rule: $\Delta \vec{w}_i = \eta \sum_{\mu \in C_i} (\vec{x}^\mu - \vec{w}_i)$.

1.2 Consider now the online version of the learning rule: $\Delta \vec{w}_i = \eta(\vec{x}^\mu - \vec{w}_i)$. Calculate the fluctuation of the weight update in the steady state, i.e. its variance $\langle \Delta \vec{w}_i^2 \rangle$.

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 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

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

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:

$$\Delta \vec{w} = \eta(N) \left(\vec{x}^{N+1} - (\vec{w}^0 + \Delta \vec{w}^N) \right) = \eta(N) \left(\vec{x}^{N+1} - \vec{w}^N \right). \tag{2}$$

How does the learning rate η depend on N?

Exercise 3: Batch vs. online: case of clustering

Define the following loss function that minimizes the squared distance between each data point and its closest weight vector:

$$L(w_1, w_2 ...) = \frac{1}{2} \sum_{\mu} ||x^\mu - w_{j^*(\mu)}||^2$$

where $j^*(\mu)$ is the index of the closest weight vector for pattern μ.

3.1 Apply gradient descent (batch rule) with learning rate η.

3.2 Apply gradient descent (online rule) with learning rate η. What is the relation to competitive Hebbian learning?

3.3 How can you choose a reduction scheme for η so that the result of online is exactly the result of batch? **Hint:** get inspiration from exercise 2.

Exercise 4: Blob formation

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

$$y_i(t+1) = g\left(\sum_k w_{ik}x_k(t) + \sum_j B_{ij}y_j(t)\right), \tag{3}$$

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 $h < 0$.

4.1 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.

4.2 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.

4.3 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 5:

5.1 In class: Show that if all prototypes/weight vectors $\vec{w}_k$ are normalized, choosing the nearest prototype $\vec{w}_i$ (i.e., the prototype for which $||\vec{x} - \vec{w}_i||^2 \leq ||\vec{x} - \vec{w}_k||^2 \; \forall k \neq i$) is equivalent to choosing the neuron i with the strongest input $\vec{w}_i^T \vec{x}$.

5.2 This question invites you to rethink what you saw in class: does the result still hold true if it is the data that is normalized (i. e. $||\vec{x}^\mu||^2 = 1 \,\forall\mu$), but the weight vectors are not? Consider the case when the clustering algorithm is close to convergence. (**Hint:** See image.)