Exercise 4: Learning

Expected time: 3h

Hints

Please use the following matlab files for your exercise:

learning.m, *Layer1.m*, *Layer2.m*, *changeweight.m*, *antihebb.m*, *createInput.m*

The model is started with the *learning.m*. Type "help learning.m" for more information.

A. Oja's learning rule

Implement a standard feedforward model using the following equations:

Layer 1: $\tau \frac{\partial r_i^{L1}}{\partial t} = r_i^{Input} - r_i^{L1}$ (use *Layer1.m*)

Layer 2: $\tau \frac{\partial r_j^{L2}}{\partial t} = \sum_i w_{ij} r_i^{L1} - r_j^{L2}$ (use *Layer2.m*)

Ensure that r_i^{L1} and r_j^{L2} always remain postive. r_i^{L1} is the firing rate of cell i in Layer 1 (r_j^{L2} analogous). r_i^{Input} gives the strength of the input for the Layer 1 cell i. w_{ij} gives the strength of the synaptic feedforward weight connection from cell i of Layer 1 to cell j of Layer 2.

Use the given model and add the learning rule of Oja to change the synaptic weights:

$\tau \frac{\partial w_{ij}}{\partial t} = r_i^{L1} \cdot r_j^{L2} - \alpha (r_j^{L2})^2 \cdot w_{ij}$ (use *changeweight.m*)

α is a constant constraint factor.

1. Start the model and evaluate what happens.

2. Add Anti-Hebbian inhibition to the activation rule for the second layer.

Layer 2: $\tau \frac{\partial r_j^{L2}}{\partial t} = \sum_i w_{ij} r_i^{L1} - \sum_{y, y \neq j} f\left(c_{yj} r_y^{L2}\right) - r_j^{L2}$ (use *Layer2.m*)

with $f(x) = \log\left(\frac{1+x}{1-x}\right)$

c_{kj} is the strength of the synaptic inhibition connection from cell k of Layer 2 to cell j from the same layer.

3. Implement the anti-Hebbian weight change: $\tau \frac{\partial c_{kj}}{\partial t} = (r_k^{L2} - \gamma)^+ \cdot (r_j^{L2} - \gamma)^+ - \alpha_c (r_j^{L2} - \gamma)^+ \cdot c_{kj}$ (use *antihebb.m*)

γ is a constant threshold and α_c gives a constant factor. $(x)^+ = max(x, 0)$.

4. Start the model with the Anti-Hebbian inhibition and explain the differences to the previous model.

B. Learning rules

Implement other learning rules (please refer to the lecture, e.g. covariance rule) and evaluate their advantages and disadvantages in comparison to one another. Note, that learning rules can easily be included into the *changeweight.m* by using the OPTIONS parameter.

C. Noise

Evaluate the influence of the level of noise added to the input patch. Use the most promising learning rule from **B**. Note, that the level of noise can be directly manipulated via the options. Type "help *learning.m*" for more information.