Principles of Brain Computation, SS17

Robert Legenstein
Institute for Theoretical Computer Science
Graz University of Technology
A-8010 Graz, Austria
robert.legenstein@igi.tugraz.at

June 7, 2017

Last Name	First Name	Matrikelnmr	Team members

Handed out on 8.6.2017
Hand in on 29.6.2017: Hand in at the Sekretariat Institut für Grundlagen der Informationsverarbeitung (Inffg. 16b/first floor) from 1 pm to 2 pm.

Task 5: Competitive Learning in WTA networks (15+5* Points)

Use this page as the cover sheet of your submission.

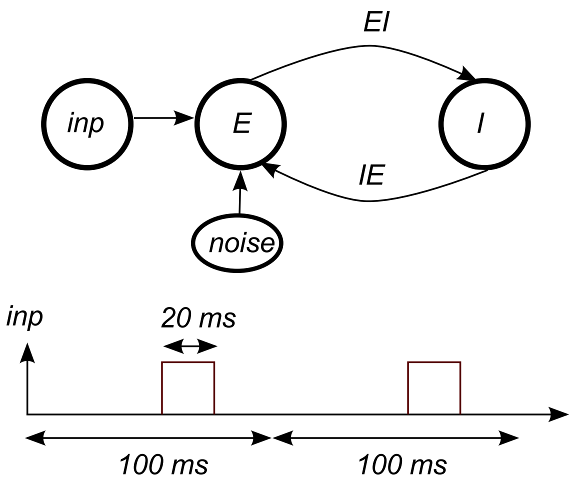

Figure 1: Top: Network structure. Bottom: Schema of input. One out of 5 possilbe patterns is presented every 100 ms and lasts for 20 ms.

This exercise sheet is based on the SEM model, a spiking neural network model which can be shown to perform probabilistic inference and learning in a mixture model. During exposure to spiking input stimuli, each neuron learns to become an expert for a particular pattern. When the pattern – or noisy variation of it – is presented again to the network, the corresponding neuron acts as a probabilistic feature detector.

Here, we do not implement the exact SEM model, but a spiking neural network in NEST with strong lateral inhibition and a SEM-like learning rule.

Network model

In the template code, a network model with lateral inhibition is implemented. The network structure is shown in Fig. 1. The network is fully implemented in the template, you just have to adjust parameters to obtain good network behavior. There are 10 neurons in the excitatory pool. These neurons receive input from 80 input neurons (Poisson generators). One pattern application lasts 100 ms. In such an application, a constant rate pattern is presented after 60 ms and it lasts for 20 ms. There are 5 possible rate patterns. The rate patterns have a Gaussian spatial profile, see the plot in the template code.

Each neuron in the excitatory pool additionally receives noise input from one excitatory and one inhibitory Poisson noise source. This has the effect that the membrane potential of the neuron varies randomly, leading to probabilistic spiking behavior. When the neurons recieve additional input from the input neuron, one of them may spike. This spike should induce several spikes in the inhibitory pool, which feed back to the E-pool.

Task 5A (5 points)

- Set the undeclared parameters (in particular the strength of the noise input and the strength of the initial input synapses, EI-, and IE connections) of the network such that one obtains this behavior. In particular, we want that at most pattern presentations, one (or a few) neuron spikes. Set `_TUNE_NETWORK` to `True` in this initial tuning.
- Describe your parameter setting and the network behavior, using plots from your simulations. Explain how it works.

Network learning without weight dependency

After the network has been set up correctly, start with learning trials, by setting `_TUNE_NETWORK` to `False`. By using `WDEP = False`, the following learning rule is applied (implemented in the template):

When a neuron spikes,

- all incoming synapses that did not spike are depressed by an ammount ηA_{-},

- all incoming synapses that did spike are potentiated by an amount $\eta(A_+ - A_-)$,

where η is a learning rate. Weigths are kept within a lower bound of 0 and an upper bound of $w_{\max}$.

Task 5B (7 points)

- Choose plasticity parameters (the ones above and the number of pattern presentations `Nep`) to obtain good competitive learning. The goal of learning is that each neuron specializes on one pattern. The optimal outcome would be that each neuron spikes ony for one specific pattern and that at each pattern presentation at least one neuron spikes.
- Describe your parameter setting and the network behavior, using plots from your simulations. Explain the results.

Network learning with weight dependency

By setting `WDEP = True` in the template code, a weight dependent plasticity rule is applied of the form:

When a neuron spikes,

- all incoming synapses that did not spike are depressed by an ammount $\eta\left(\frac{w_{ij}}{w_{\max}}\right)^{\alpha}$,
- all incoming synapses w_{ij} that did spike are potentiated by an amount $\eta\left(\frac{w_{\max}-w_{ij}}{w_{\max}}\right)^{\alpha}$,

where η is a learning rate, $w_{\max}$ is the upper bound for the weights, and α is a parameter. We use $\alpha = 0.5$.

Task 5C (3 points)

- Use this rule and adapt the plasticity parameters (η, $w_{\max}$, number of epochs), to obtain good competitive learning.
- Describe your parameter setting and the network behavior, using plots from your simulations. Describe differences to the weight-independent learning rule and why this happens.

Task 5D (5* points)

- For the weight-dependent learning rule, consider one synapse with given constant pre-post spike statistics $P(\text{pre spikes}|\text{post spikes})$. Derive the fixed-point of the weight value, i.e., that weight value where the expected weight change is 0. For given statistics, what can we say (mathematically) for the weight-independent rule?

Submit your plots and interpretations on paper, and the code by email to `robert.legenstein@igi.tugraz.at` and `m.mietschnig@student.tugraz.at` (**include "PoBC ex5 submission" in the subject of the email**) until 8am on the day of submission. All members of the same team are allowed to submit the same code. Do not send the report by email but hand in a printout.