Principles of Brain Computation, SS17

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

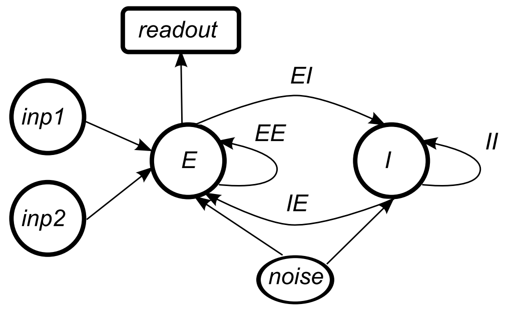
May 18, 2017

Last Name	First Name	Matrikelnmr	Team members

Handed out on 18.5.2017
Hand in on 1.6.2017

Task 4: LSMs (20 Points)

Use this page as the cover sheet of your submission.



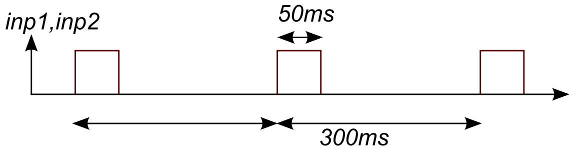


Figure 1: Top: Network structure. Bottom: Schema of input for each of the two input neurons. A pattern is presented every 300 ms and lasts for 50 ms. In task 4A, input neurons can be active at that time (probability of being active is 0.5, neurons are active independently). The readout should compute the XOR of the previous pattern. In task 4B, exactly one input neuron is active at that time (active neuron chosen with equal probability).

In this exercises you will perform simulations in nest that investigate the liquid state machine approach.

Liquid state machine

Create a network that consist of an excitatory pool of 1000 LIF neurons (E-pool) and an inhibitory pool of 250 LIF neurons (I-pool), see Fig. 1. Most parameters are already given in the provided code template. Set the bias current of all the neurons to $I_e = 14.5$ pA . All the synaptic weights are distributed normally with the given mean and a standard deviation of 70% of the mean. All the synapses should be dynamic (parameters are given in the code). Create recurrent connections within the E-pool (EE syns) with constant in-degree of 2 and mean synaptic weights $J_{EE} = 50$ pA. The E-pool is connected to the I-pool with constant in-degree of 2. These mean synaptic weights J_{EI} is 250.0 pA. The I-pool is connected to the E-pool with a constant in-degree of 1 and a mean synaptic weight of $J_{IE} = -200$ pA. Finally, the I-pool has recurrent connections with a constant in-degree of 1 and a mean synaptic weight of $J_{II} = -200$ pA.

Noise spikes are generated and connected to each neuron in the E and I-pools with a synaptic weight $J_{noise} = 5.0$ pA (parameters given in template). All synapses should have a synaptic delay with a mean of 10 ms and a standard deviation of 20 ms clipped at 3 ms and 200 ms at the lower and higher end.
To set the bias of the neurons, you can use

```
nest.SetStatus(nodes, {'I_e': <bias current value>})
```

To connect neuron pools with normally distributed weights and delays, you can use

```
nest.Connect(source_nodes, target_nodes,
<connection rule dict>,
{'model': <synapse model>,
    'weight': {'distribution': 'normal',
        'mu': <mean weight value>,
        'sigma': <std of weight values>},
    'delay': {'distribution': 'normal_clipped',
        'mu': <mean delay>,
        'sigma': <std of delays>,
        'low': <minimum allowed delay>,
        'high': <maximum allowed delay>}
})
```

The template script already contains a function that generates the spikes for two input neurons. Each of these input neurons makes 100 connections to neurons in the E-pool (connection rule `fixed_outdegree`). Synaptic weights of these connections should be distributed uniformly in $[125, 375]$ pA. These synapses should be static (model `static_synapse`). The delays should be distributed as for the network synapses (described above). The input stimulus is shown in Fig. 1. Consider n input patterns $(x_1^1, x_2^1), (x_1^2, x_2^2), \ldots, (x_1^n, x_2^n)$ with $x_i^j \in \{0, 1\}$. At time $t = i \cdot 300$ ms, pattern (x_1^i, x_2^i) is presented to the network. If $x_j^i = 1$, then input neuron j spikes at 200 Hz for a duration of 50 ms. In task 4A, each x_j^i is 1 with probability 0.5 (all x_j^i are drawn independently). In task 4B, either $x_1^i = 1$ or $x_2^i = 1$ (exactly one of them is 1) with equal probability for the two cases.

When setup correctly, the average rates in the E and I-pool will be around 20Hz and 30Hz respectively (Perform some initial short simulations and check these rates. You may also want to plot the spiking activity in the network to get a feeling for the circuit dynamics).

Task 4A, XOR computation (10 points): Record the spikes of 500 neurons in the E-pool and extract the liquid states (code is provided in the template script) 10 ms after a pattern presentation (i.e., at times 360 ms, 660 ms, 960 ms etc.). Use linear regression to train a linear readout (code provided) on 80% of the gathered liquid states to compute $XOR(x_1^i, x_2^i)$ of the previous input pattern (x_1^i, x_2^i). Then test on the remaining 20% of states (code is provided that splits the data randomly). Repeat training for 20 randomly chosen partitions into training and test sets, compute the mean and STD of the misclassification rate. Note: For several training sessions of the readout, one needs to run the circuit only once.

Task 4B, Fading memory (10 points): Record the spikes of 500 neurons in the E pool and extract the liquid states at a delay Δt after pattern presentations (i.e., at times $350 + \Delta t$, $650 + \Delta t$, $950 + \Delta t$ etc.). Train a linear readout on 80% of the gathered liquid states to classify which of the two input channels was active in the previous pattern. Then test on the remaining 20% of states (code is provided that splits the data randomly). Repeat training for 20 randomly chosen training/test sets, compute the mean and STD of the misclassification rate. Do this for a number of delays Δt and plot the misclassification rate in dependence on the delay. Note: For different delays, one only needs to run the circuit once. Then extract the liquid states at various times from the spike times of recorded neurons.

Notes: One needs many examples to achieve acceptable performance. Around 200 s of simulation time should work fine. But if this takes too long, you can also work with less simulation time. You can also try to find a good regularization parameter and train the readout with regularized least squares (code provided) in order to avoid overfitting.

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 ex4 submission" in the subject of the email**) until 8am on the day of submission. Write readable code with informative comments. 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.