SYA14 - Neuromorphic Computing

Lab 4

1 Objective

In this lab, we will study the learning methods for SNN. In particular, this lab is designed for the STDP learning rules.

2 Prerequisite

The following are the prerequisites of this exercise:

- Coding techniques (Lab 3).
- STDP Learning Rule
- Python

3 Ex 4.1: Software Implementation of STDP

In this exercise, we design in Python the STDP learning by following the instructions below:

- SNN configuration: fully connected, input 9 neurons (3×3) and output 2 neurons.
- There is an inhibitory connection between two output neurons for the winner-take-all mechanism.
- Neuron is Leaky Integrated and Fire (see Lab 2).

The change in weight of a synapse can be expressed as:

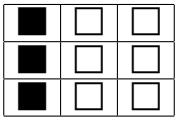
$$\Delta w = \begin{cases} A_+ e^{+\Delta t/\tau_+}, & \Delta t < 0, A_+ > 0 \\ A_- e^{-\Delta t/\tau_-}, & \Delta t > 0, A_- < 0 \end{cases} \tag{1}$$

where $\Delta t = t_{pre} - t_{post}$, denoting the time difference between presynaptic and its postsynaptic spike, A_+ and A_- denote the learning rate depending on the synaptic weight. τ_+ and τ_- are the time constants.

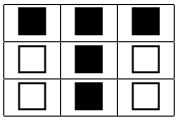
3.1 Input neurons

The input neurons can be constructed as a 3×3 shape and can be represented as a 3×3 pixel image. Please generate with rate coding for the following patterns (□ is white pixel and ■ is the black pixel):

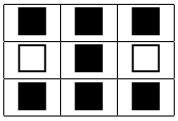
Pattern 01:



Pattern 02:



Pattern 03:



Using rate coding:

- ■: firing rate is 1.0
- □: firing rate is 0.0

3.2 Initial synaptic weights

The synaptic weights are randomized and normalized (the sum of the weights between all input neurons to an output neuron is constant).

3.3 Training rules

Train the network with STDP, one pattern 10 times. Each time train with 350 time steps. Please keep the normalization of the weight.

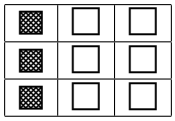
3.4 Report content

- Source code of the training program
- Plot of the weight (in 3x3 format)
- Report on training accuracy

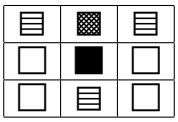
4 Ex 4.2: Validating the results

In this ex, please validate with the following patterns:

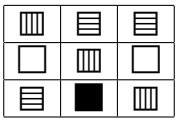
Pattern 01:



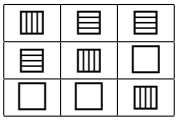
Pattern 02:



Pattern 03:



Pattern 04:



For rate coding:

- ■: firing rate is 1.0
- ▩: firing rate is 0.8
- ▤: firing rate is 0.6
- ▥: firing rate is 0.4
- □: firing rate is 0.0

5 Ex 4.3: Verilog HDL Implementation of STDP

In this part, we will design in Verilog HDL the previous STDP rule. Note that with hardware, we should approximate the value.

5.1 Report content

- Source code of the training program with Verilog HDL
- Plot of the weight (in 3x3 format)
- Report on training accuracy
- Comparison between Software (Python) and Hardware (Verilog HDL).

6 Submission format and Deadline

Your report should be prepared in English and should contain the following:

1. Your name, your ID, and the Lab #.
2. All reports
3. Submission format: soft copy.

Note: This Laboratory is designed for the book [1]

[1] Book: Neuromorphic Computing Principles and Organization 1st, Edition, ISBN-10: 3030925242, ISBN-13: 978-3030925246, Publisher: Springer, May 2022.