2018 Exercise 1 Prof. Manevitz

You have to do the following. The grade will count as 50% of your final grade (60% if you do the extra credit). However, you **must pass** the final exam for it to count. The project is due by Dec 27 10 AM in the class. You may work in groups of 2; each one is supposed to understand and be involved in all aspects of the work.

If you turn it in by Jan 3, there will be a 3 point penalty. If you turn it in by Jan 10 there will be a 10 point penalty. If you turn it in (to my mail box) by Jan 16 there will be a 20 point (total) penalty and you will have to arrange a special meeting to judge it. It will not be accepted after Jan 16. So turn it in on time.

There are no "right" answers to this exercise. You will be judged on the quality of your implementation and your insights to the situation during an oral presentation to me.

**Part I: Simulation of Simplified Discretized Leaky Integrate and Fire Neuron**

1. **Design a simulator of a leaky integrate and fire neuron according to the guidelines below.**

Look at an integrate and fire neuron as a "bucket" having a maximal capacity, C. There are (1) pipes entering the bucket corresponding to dendrites and a hole in the bucket with (2) a small leak of a certain constant rate, l. The pipes allow only particular amounts of water (i.e. charge) to enter at a time (these are the spikes). Each spike carries a fixed amount of charge Gamma.

So in a simulation you have to keep track of the current value of C; and design updates at discrete time intervals, delta t, wherein C:= C(old) – leak x . times. delta t + Gamma .times. (the number of spikes entering the bucket during the time interval). To simplify the model connections allow a weight factor on each entry so C(new) ;= C(old) + sum ( w\_i .times. Gamma times the number of spikes entering on dendrite i ) - leak rate \* delta t.

If at the new time C is bigger than threshold; output a spike.

1. **Implement a coincidence detector with tolerance as described below**.

Suppose such a neuron has two dendrites D\_1 and D\_2 with identical input weights (say of value 1). Assume the leak is small, say 1/50 of the size of a spike. Can you design a coincidence detector beween two events, E1 and E2 where the coincidence has to be within a certain tolerance (i.e. not necessarily at exactly the same time). Assume E1 causes a spike to be generated for D1 , E2 causes a spike to be generated for D2.

1. **Do the same with the change indicated below in the encoding of the events.**
2. Now assume that when E1 occurs, a spike train of five spikes is generated over consecutive discrete time steps 1 1 1 1 1 while when E2 is generated 111 is generated. What has to change on the set up of the simulator?

Implement the above situations and discuss what happens.

**Part II**

**In this part you should design a network of such leaky integrate and fire neurons (say 100 such) .** There is a matrix of connectivity from neuron i to neuron j with a real weight value between -1 and 1.

**(a)** **You are asked to implement and explore this model with different parameters.**

Randomly choose 10% of the neurons to be input neurons; 10% to be output neurons. (Their purpose is described below.)

When a neuron fires, it sends a spike to all other neurons moderated by the weights. (If a neuron i is not connected to neuron j, then the weight is 0). If the weight is positive we call neuron i excitatory; if negative inhibitory.

Input to the network; a sequence (s1, s2, … s80) of spikes or non-spikes is input (i.e. s\_k= 1 if a spike is input, s\_k = 0 if not) to all the input neurons.

**(b)** Can you find examples of a weight matrix where the network doesn’t die out, as long as any non-zero sequence is entered? Can you find one where the network almost always dies out after sufficient time steps? Show the experiments supporting what you say.

1. **Now choose a set A of twenty distinct sequences of inputs of length 90 having two thirds 1s and one third 0s (e.g. 1 1 0 1 1 0 1 1 0 … is one such sequence; while a second set B has half 1s and half 0s. (e.g. 10 10 10 …)**

**Can you see a distinguishing difference in the network behavior between these sets**.

(Figure out some sort of visualization.)

Report on your simulations and your method. (One method might be to look at a perceptron detector which inputs the entire sequence of outputs over a time period.)

What happens if you use equal numbers of excitatory and inhibitory neurons? What if 80% are excitatory?

**Extra Credit:** Now experiment and see what you can say about generalization and about robustness of this network.

There are no "right" answers to this exercise. You will be judged on the quality of your implementation and your insights to the situation during an oral presentation to me.