

Introduction to Computational Neuroscience

Practice V: Single Neuron Model

Ilya Kuzovkin, Raul Vicente

March 14, 2014

In this session we will have a brief look on three different computational models of a neuron: McCulloch-Pitts, Integrate-and-Fire and Hodgkin-Huxley.

Please record the time you will spend of this homework and add it to the report. This is just for me to balance the amount and the difficulty level of the exercises.

Exercise 1: Logic gates (0.5pt)

On the lecture we have seen how to construct AND, OR and NOT logic gates using the the McCulloch-Pitts model of a neuron. Your task is to construct more. Please construct the following two gates:

1. NAND
2. XOR

Hint 1 For the XOR gate you will need more than one neuron.

Hint 2 Same input can go simultaneously to several neurons.

Exercise 2*: Simulate memory with $\neg S \neg R$ latch (0.5pt)

In electronics there is a circuit, which can *store* a state. This means that after we *set* it to some state, it will remain there until we *reset* it. The whole thing is called *D latch*, it has a sub-part called *SR latch*, which again has a subpart called $\neg S \neg R$ latch.

1. Watch this video <https://www.youtube.com/watch?v=PCT76PsDr6g> (until 12:26)
2. and build $\neg S \neg R$ latch using McCulloch-Pitts neurons.

Exercise 3: Integrate and Fire neuron model (1pt)

Integrate and Fire neuron accumulates voltage until it reaches the *threshold*. After that it fires and resets voltage back to initial value. In this exercise we will model behaviour of such neuron and study its properties. Follow the instructions in the `session5ex3.m` file and report all figures, essential pieces of code, answers, interpretations and conclusions you will make during the work.

Note The TODO marker will indicate the places where you have to do something: complete the code, plot and report a figure, give an interpretation, etc.

The very final result in this exercise should look something like this

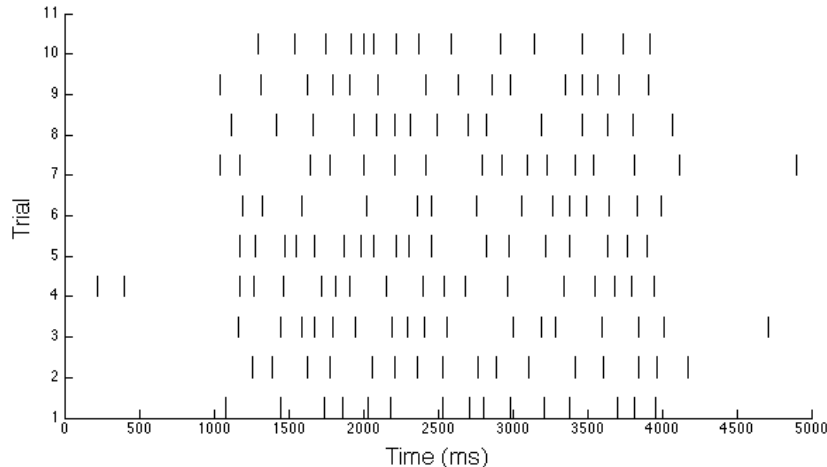


Figure 1: 10 trials of data generated using Integrate-and-Fire neuron model.

Exercise 4: Hodgkin-Huxley neuron model (1pt)

Hodgkin-Huxley model is considered to be the most important computational neuronal model in the neuroscience today. We have the model already implemented in the file `HH0.m`, study it. Follow the instructions in the `session5ex4.m` file and report all figures, thoughts, interpretations and conclusions you will have during the work.

Note The TODO marker will indicate places where you have to do something: complete the code, plot and report a figure, give an interpretation, etc.

Exercise 5*: Integrate and fire with synapses (0.5pt)

In this exercise we will play with somewhat more realistic version of integrate-and-fire model, which receives input not from constant current as we did before, but from incoming (*presynaptic*) spikes. *Temporal summation* can lead to the voltage reaching the threshold and then the *postsynaptic* neuronal response (firing) occurs. Read the tutorial¹ and study the code given there. Slightly modified code is provided to you in the file `session5ex5.m`. Your task is place incoming spikes on the line 41 of the file in such that:

1. Inside the time window from 0 to 200 ms there will be 4 incoming spikes and 1 output spike.
2. Inside the time window from 200 to 400 ms there will be 5 incoming spikes and 0 output spikes.
3. Is it possible to produce 2 output spikes with 5 incoming spikes? If yes, then produce it in time window 400 to 600 ms, if not show the maximal voltages you can achieve with 5 input spikes.

Exercise 6: Other single-neuron models (0.5pt)

Go to the Internet and find other single-neuron models that have been proposed. Describe how author came up with this model, why this model is useful and which properties does it have.

¹<http://www.dreamincode.net/forums/topic/72868-a-simple-neuron-model-the-integrate-and-fire-neuron>

Please submit a PDF report with answers to the questions and comments about your solutions. Your report should contain also the essential parts of the code you have produced. If the code is too massive you can add it to the submission and upload everything as a zip archive. But single PDF is preferred.