MATLAB Workshop

TECHNISCHE HOCHSCHULE LÜBECK

Sheet 8 – Boolean operators and flow charts

Please write all commands in the MATLAB editor into one single m-file and save it in a folder that you specifically dedicate to this workshop. If you don't know how a command is being used type "`help [commandname]`" into the command window. Comment each code line briefly to document what it is doing.

Exercise 1:

A "coincidence neuron" is a neuron that outputs a spike (an action potential) only if all their input dendrites receive a spike simultaneously (i.e., at the same point in time). The goal here is to model such a coincidence neuron with 3 independent input dendrites and one output axon. Let a spike be symbolized by a "1" and no spike with a "0".

a) Create the three input vectors symbolizing the three dendrites by using a random process that creates a spike in 10% of the instances in time. Let all three vectors have 1000 entries corresponding to 1 second of data (i.e., a sampling frequency of 1000 Hz).

b) Calculate the spike rate of each dendrite, i.e., the average number of spikes per second.

c) Create the output of the coincidence neuron by assuming that it only spikes when all three dendrites have a spike at the same instant in time.

d) What is the average spike rate (i.e., the number of spikes per second) of the output axon?

e) Plot the spike trains of the three input dendrites within one plot using different colors. Mark the instances in time that the output neuron spikes with 'X'-Symbols in a different color.

Exercise 2:

Take the same neuron from Exercise 1, but now assume that one of the three dendrites has an inhibitory behavior, i.e., the coincidence neuron spikes if the two excitatory dendrites have a spike at the same time, but only if at that instance in time there is not a spike from the inhibitory dendrite. What is the spike rate of the output axon now?

Exercise 3:

Before you start the actual coding of this exercise, make a flow chart about this task!

Based on the code from Sheet #8, Exercise 1 (coin-flip), modify the code such that user gets a second chance to flip the coin only if the coin flip took place in a second that is an even number, e.g., at 17:33:44, but not at 17:33:45. Use the command `clock` to access the exact time.