CSE/NB 528 Homework 3: Modeling Neurons and Networks

(Due Date: **Sunday, Feb 19, 2017 before midnight**)

Submission Procedure:
Create a Zip file called "528-hw3-*lastname-firstname*" containing the following:
(1) Document with write-up containing your answers to any questions asked in each exercise, as well as any figures, plots, or graphs supporting your answers,
(2) Your Matlab program files,
(3) Any other supporting material needed to understand/run your solutions in Matlab.

Upload your Zip file to this dropbox.

Upload your file by 11:59pm Sunday, Feb 19, 2017.
Late submission policy is here.

The first three exercises will take you from modeling simple passive membranes and integrate-and-fire neurons to modeling synapses. These exercises will provide you with a basic set of "starter code" you could use for investigating potential research questions such as temporal versus rate-based coding and synaptic plasticity.

The last exercise introduces you to stability and dynamics of recurrent networks.

The extra-credit problem tests your understanding of information theory and neural coding. (Note: You won't be penalized if you skip extra-credit problems; extra-credit points will be added after final scores have been computed).

1. (Fun with membranes; 20 points) Download and run the following Matlab/Python code for modeling a passive neuronal membrane as an RC-circuit: membrane.m (membrane.py) This code demonstrates how a membrane responds to a constant current input that is turned on for a fixed time interval and then turned off.
   a. Change the values for the membrane's resistance and capacitance (R and C), and find out how this influences the response of the membrane. Does it reach a stable value more quickly or more slowly after:
      i. multiplying R by 5
      ii. dividing C by 10
      iii. multiplying R by 10 AND dividing C by 10?
   b. An experimental method for calculating a membrane's time constant tau (when R and/or C are not known) is to start at zero and record the time at which the membrane potential V reaches a value approximately equal to 0.6321*V_peak= 0.6321*(I*R), where I is the constant injected current. Check if this method works by injecting different amounts of current I and changing the values for R and C. Once you've convinced yourself that the experimental tau appears to be identical to the theoretical tau (= RC) in all these cases, provide a theoretical justification for why this method works. (Hint: Derive a closed form exponential equation for V(t) by solving the differential equation for V and find the value of V at time t = tau).

2. (Get fired up about Integrate-and-Fire neurons; 25 points) Run the following Matlab/Python code for modeling an integrate-and-fire neuron: intfire.m (intfire.py)
   a. Vary the input current gradually from very low to high values and find out the minimum current needed to cause the neuron to spike.
   b. What is the maximum firing rate (spike count/trial duration) of this neuron and how is it related to the absolute refractory period "abs_ref" in the code?

c. Plot a graph showing input current versus the output firing rate of the neuron.
d. Instead of feeding constant input, make the current I a sinusoidal function of time: I = sin((1:tstop)*f); where f is the input frequency. Plot a graph showing the output firing rate as a function of input frequency.
e. Find the "resonant frequency" (if any) where the neuron "tracks" the input by firing exactly 1 spike for each peak in the input.

3. (Get to know an Alpha Synapse; 25 points) Here's some code for simulating an integrate-and-fire neuron receiving input spikes through an alpha synapse: alpha_neuron.m (alpha_neuron.py)
The parameter "t_peak" controls when the alpha function peaks after an input spike occurs (and hence how long the effects of an input spike linger on in the postsynaptic neuron). "t_peak" for excitatory synapses in the brain may vary from 0.5 ms (AMPA or non-NMDA) to 40 ms (NMDA synapse).
   a. Vary the value of t_peak from 0.5 ms to 10 ms in steps of 0.5 ms and observe how this influences the output of the neuron for the fixed input spike train used in this code. Plot the output spike count as a function of t_peak for the given input spike train.
   b. Fix t_peak = 0.5 ms. Change the input firing rate by varying the threshold parameter "thr" from 0.7 to 0.95 in steps of 0.05. For each value of "thr", repeat the experiment 5 times with different random input spike trains. Record both the output and the input firing rates in each case. Plot the average output firing rate as a function of average input firing rate at each value of "thr". Is the shape of this input-versus-output plot similar to or different from the plot in 2c?
   c. How would you turn this synapse into an inhibitory synapse? (Hint: See lecture slide and think about the role of the equilibrium (aka reversal) potential of a synapse). Make the necessary change to the code such that the random input spike train in 3a (stored in "spike_train") now acts through an inhibitory synapse. Add a constant current input to the neuron that is sufficient to cause it to spike with a high firing rate of about 150 Hz in the absence of inhibitory inputs. Now turn on the inhibitory input spike train and vary the value of t_peak from 1 ms to 15 ms in steps of 2 ms (with g_peak = 0.05 as in 3a). Plot the output spike count as a function of t_peak for the given input spike train.

4. (Being Nonlinear and Recurrent; 30 points) Write Matlab/Python code and answer the questions in **Exercise 4** from Chapter 7 in the textbook as described in the file c7.pdf.

   Create figures reproducing Figures 7.18 and 7.19 in the textbook using your code, and include additional example figures to illustrate the effects of varying the value of tau_I.

   (The following files implement a nonlinear recurrent network in Matlab/Python: c7p5.m and c7p5sub.m (c7p5.py for Python). (These files are for Exercise 5 in c7.pdf but you can modify them and use them for Exercise 4. For an analytical derivation of the stability matrix, see Lecture Slides and Mathematical Appendix Section A.3 in the text).

Extra Credit Problem (Information Theory and Neural Coding; 20 points)
Show that the firing-rate distribution that maximizes the entropy when the firing rate is constrained to lie in the range $0 <= r <= r_{max}$ is given by equation 4.22 in the textbook, and that its entropy is given by equation 4.23. Use a Lagrange multiplier (see the Mathematical Appendix in the textbook) to constrain the integral of $p[r]$ to one.