Neural Computation (VS 265), Problem Set 1

Due date: September 8, 3:30pm

Fall 2022

General guidelines:

- We are grading problem sets anonymously. **Include your student ID in the submission, but do not include your name.**
- You may work in small groups of 2-3. Note that you are responsible for writing up and submitting your submission individually.
- You are expected to attach any code you used for this assignment but will be evaluated primarily on the writeup.

Part 1: The Membrane Equation

(a) Membrane dynamics. Simulate the membrane equation to show how the voltage across the cell membrane will change in response to a step input current, $I(t)$. Numerically simulate the solution for a duration of least 500 milliseconds with the following parameters:

- Initial condition: $V(0) = V_{rest} = -70\,\text{mV}$
- Membrane capacitance: $C = 100\,\text{pF}$
- $I(t) = \begin{cases} 100\,\text{pA} & t \geq 100\,\text{ms} \\ 0\,\text{pA} & t < 100\,\text{ms} \end{cases}$
- $G_{Leak} = 5\,\text{nS}$.

Try different values of G_{Leak} and C to explore how these parameters affect the rise time and resulting membrane voltage. Plot the results of your simulation and interpret your findings.

You may find it easiest to run this simulation using the Euler method, but you are free to use any other method. (See the handout on Simulating Differential Equations for further information).

(b) Membrane nonlinearity. Now let's examine how the membrane voltage at equilibrium behaves in response to specific ion channels opening, first in isolation and then in combination.

i. Examine the effect of a single synaptic input that opens a set of sodium channels (ΔG_{Na}). Sweep ΔG_{Na} from $0\,\text{nS}$ to $25\,\text{nS}$ and plot the resulting equilibrium membrane potential (by solving for V at $\frac{dV}{dt} = 0$) over this range. You should notice a regime where the membrane voltage can be reasonably approximated as a linear function of ΔG_{Na} – what is that regime and why?

ii. Now do the same for an inhibitory synaptic input that opens a set of potassium channels, by varying ΔG_K over the same range and superimposing on the plot above.

iii. Next, examine how the membrane responds *jointly* to both synaptic inputs. Show this as a contour plot and choose a range of values for ΔG_{Na} and ΔG_K that allows you to see the linear vs. nonlinear regimes for combined input. Explain why this is happening with reference to your plots above.

iv. Finally, in a third plot, show the effect of shunting inhibition by simulating an inhibitory synaptic input that causes chloride channels to open by some amount (say $\Delta G_{Cl} = 10\,\mathrm{nS}$) and now sweep ΔG_{Na} over the same range as above. (You may assume $V_{Cl} = V_{rest}$.) How does this compare to what you would expect from a linear superposition? (plot as a dashed line). Explain your results.

Part 2: Perceptron

As discussed in class, the Perceptron is a simple model of a neuron that assumes synaptic inputs $x_1, x_2, \ldots, x_n$ are combined linearly, and the sum is thresholded to 0 or 1 as a crude approximation to the action potential. That is,

$$u = \sum_{i=1}^{n} w_i\, x_i + w_0$$

$$y = \Theta(u) \equiv \begin{cases} 1 & u > 0 \\ 0 & u \leq 0 \end{cases}$$

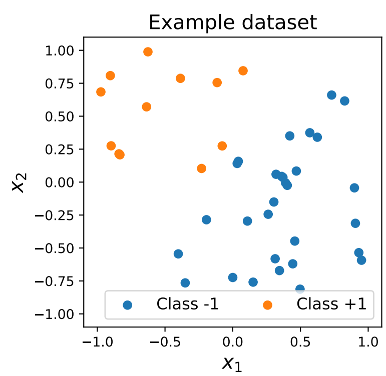
where the $w_1, w_2, \ldots, w_n$ correspond to synaptic weights and y is the output of the neuron.

Rosenblatt also defined a learning rule for updating the weights based on a set of training examples and of inputs and their desired output ('teacher signal'), $T = \{+1, -1\}$. Let $x^{(m)}_{1,2,\ldots n}$, $y^{(m)}$, and $T^{(m)}$ denote the m-th example of input, resulting neuron output, and teacher signal, respectively. The learning rule for w_k is defined as:

$$\Delta w_k = \begin{cases} 2\eta\, T^{(m)}\, x_k^{(m)} & \text{if } (2\,y^{(m)} - 1) \neq T^{(m)} \\ 0 & \text{otherwise} \end{cases}$$

where Δw_k is the incremental change to w_k and η is a learning rate.

Implement the perceptron learning rule for the dataset posted to the course website. When you plot the dataset, it should look as shown below:



Train your network until it achieves 100 percent accuracy. Make a plot of how classification accuracy changes as a function of training epochs. Finally, show the final decision boundary (i.e. the line that partitions one class from another) on top of the data.

(Optional) A limitation of the Perceptron is that it can only correctly separate data that are *linearly separable*. An example of a problem that is not linearly separable is the exclusive-or (XOR) function, defined as follows:

x_1	x_2	T
0	0	-1
0	1	+1
1	0	+1
1	1	-1

However, a multi-layer Perceptron (MLP), where the output of one layer of Perceptrons is fed as input to another layer of Perceptrons, could solve this problem. Construct (by hand for now) an example of an MLP architecture, including the weights at each layer, to compute the XOR function. The lack of a learning rule for MLP's was central to Minksy & Papert's critique and set back the field of neural network research for at least two decades.