Neural Nets: Problem Set 1

Due Wednesday Feb. 11.

Revised at 6:00am Feb. 9.

February 9, 2015

General guidelines for homework assignments:

- You may collaborate with others when solving the problems, but your writeup should be your own work.
- Please write the names of your collaborators (if any) on your submission.
- Explain your work as concisely as possible. You may be graded on clarity of explanation, as well as "correctness."
- When you draw graphs, make them as informative as possible by labeling the axes and by annotating important locations.
- The maximum number of points for the assignment is 20.

1. Integrate-and-fire model neuron. Consider an isolated neuron receiving input current I. Suppose that the neuron voltage $V(t)$ obeys the differential equation

$$C\frac{dV}{dt} = [I - I_\theta]^+ \tag{1}$$

   in the interval $[V_0, V_\theta]$, and $[u]^+ = \max\{u, 0\}$ is the (half-wave) rectification nonlinearity. The *threshold voltage* V_θ is assumed to be larger than the *reset voltage* V_0. If V increases and reaches the value V_θ, then V is instantaneously reset to the value V_0. This event models an action potential or spike. (In this crude model, the neuron is approximated as a capacitor for voltages between V_0 and V_θ, but nonlinear behaviors happen at the extremes of this interval.) Label your axes correctly in all graphs.

   (a) (1 pt.) Suppose that I is constant in time and larger than I_θ. Draw a graph of V as a function of t.

   (b) (1 pt.) Derive a formula for the frequency of spiking f versus the constant current I.

   (c) (1 pt.) Draw a graph of frequency of spiking f versus the constant current I.

2. Synapse as a leaky integrator. The current I sourced by a synapse in the postsynaptic neuron depends on the spike times t^a of the presynaptic neuron, and this dependence can be modeled by

$$\tau\frac{dI}{dt} + I = Q\sum_a \delta\left(t - t^a\right)$$

   where Q is the amount of charge in the postsynaptic current generated by a single spike, and $\delta(t)$ is the Dirac delta function.

   (a) (1 pt.) Suppose that the presynaptic neuron is spiking regularly in time with period $\tau/10$. Draw a graph of I versus t.

   (b) (1 pt.) Suppose that the presynaptic neuron is spiking regularly in time with period 10τ. Draw a graph of I versus t.

3. A spiking network. Now consider a network of leaky integrate-and-fire neurons communicating via leaky integrator synapses,

$$C_i \frac{dV_i}{dt} = \left[\sum_j I_{ij} - I_{\theta,i}\right]^+$$

$$\tau_{ij} \frac{dI_{ij}}{dt} + I_{ij} = Q_{ij} \sum_a \delta\left(t - t_j^a\right)$$

(Assume that the voltages V_i have the reset behavior described in Problem 1, and each neuron i has its own capacitance C_i and threshold current $I_{\theta,i}$.) Suppose that the condition $\tau_{ij} = \tau_j$ is satisfied, and that the spiking neurons all have high rates. Approximate the above equations via

$$\tau \frac{dx_i}{dt} + x_i \approx \left[b_i + \sum_j W_{ij} x_j\right]^+ \tag{2}$$

where $[u]^+ = \max\{u, 0\}$ is the (half-wave) rectification nonlinearity.

   (a) (2 pt.) Derive a formula for W_{ij}.

   (b) (2 pt.) Derive a formula for b_i.

4. Let $x_1, \ldots, x_N$ be N boolean variables, taking on the values 0 or 1.

   (a) (2 pt.) Consider the conjunction $\bar{x}_1 \wedge \bar{x}_2 \cdots \wedge \bar{x}_n \wedge x_{n+1} \wedge \cdots \wedge x_N$ in which the first n variables are negated. Express this function as an LT neuron $H\left(\sum_i w_i x_i - \theta\right)$ with weights w_i and threshold θ, where H is the Heaviside step function. Note that conjunction, or AND, is defined by

$$x_1 \wedge \cdots \wedge x_N = 1 \text{ if and only if } x_i = 1 \text{ for all } i$$

   (b) (1 pt.) Do the same for the disjunction $\bar{x}_1 \vee \bar{x}_2 \cdots \vee \bar{x}_n \vee x_{n+1} \vee \cdots \vee x_N$. Note that disjunction, or OR, is defined by

$$x_1 \vee \cdots \vee x_N = 1 \text{ if and only if } x_i = 1 \text{ for some } i$$

5. Classifying MNIST images with an LT neuron. Download and unzip the file `DeltaRule.zip`. One file is `mnistabridged.mat`, which contains a small subset of the MNIST database. The two MATLAB arrays `train` and `test` contain 5000 and 1000 images, respectively, as unsigned 8-bit integers. The arrays `trainlabels` and `testlabels` contain the labels of the images, numbers from 0 to 9. The images are stored as 784 dimensional vectors, which is convenient for numerical computations. They should be converted to 28×28 whenever visualization is necessary. For example, the command `imagesc(reshape(train(:,1),28,28))` displays the first image in the array `train`. You will probably want to use a gray scale colormap, `colormap(1-gray)`.

   (a) (1 pt.) Another file is the MATLAB code `DeltaRule.m`. This trains an LT neuron to be activated by images of "two" but remain inactive for images of other digits. Run the code. What is the final average training error of the LT neuron?

   (b) (1 pt.) Write code to run the trained LT neuron on the test set of images. What is the average error on the test set? Submit your code.

   (c) (1 pt.) What would be the error rate of the trivial recognition algorithm that always returns an output of 0?

6. Now consider training 10 LT neurons to detect the 10 digit classes. More generally, assume there are k weight vectors, which are the rows of a $k \times n$ matrix W. The desired output $\mathbf{y}$ and bias $\mathbf{b}$ are $k \times 1$ vectors, and the input $\mathbf{x}$ is an $n \times 1$ vector. The goal of supervised learning is to find W and $\mathbf{b}$ such that this approximation holds:

$$\mathbf{y} \approx H\left(W\mathbf{x} + \mathbf{b}\right) \tag{3}$$

Here the Heaviside step function H has been extended to take a vector argument, simply by applying it to each component of the vector.

(a) (2 pt.) Write the delta rule in matrix-vector format. Use $\eta > 0$ as the learning rate parameter.

(b) (2 pt.) Modify `DeltaRule.m` to train 10 LT neurons to recognize all 10 digit classes in `mnistabridged.mat`. The ith component $y_i = 1$ if the input $\mathbf{x}$ belongs to the ith digit class, while all other components are zero. We'll use the convention that the class of "zero"s corresponds to $i = 10$. (Note: You could simply add an outer loop around the `DeltaRule.m` code for the different digit classes. Don't do this, as it's more efficient and elegant to code the updates in the matrix form that you have derived.) Submit your code along with the ten learned weight vectors displayed as images.

(c) (1 pt.) If all the LT neurons were performing perfectly, only a single one would be active for each image. In reality, the number of active neurons is often greater or less than one. If we want the neurons to collectively choose a single digit class, we can replace the Heaviside step function with the winner-take-all function $\mathbf{v} = \mathrm{WTA}(\mathbf{u})$, defined by

$$v_b = \begin{cases} 1, & b = \mathrm{argmax}_a\, u_a \\ 0, & \text{otherwise} \end{cases}$$

In other words, only one of the $k = 10$ neurons remains active, the one with index equal to

$$\underset{a}{\mathrm{argmax}} \{\mathbf{w}_a \cdot \mathbf{x} + b_a\}$$

where the maximum with respect to a runs over the k neurons. Write code to evaluate the error of this "winner-take-all" classifier on the training and test sets. What are the error values? (Note in case you're worried: If there happens to be a tie, the winner-take-all operation can be defined as choosing one of the maxima at random. Ties are unlikely when using floating point numbers.)