Due Sept 16, 2014 at 5:00PM

Your submission must include:

1. A .pdf or .doc file clearly documenting your code, figures, and results.

2. Your MATLAB code. The code must be saved in plain text files that can be immediately run in MATLAB; include a file called main.m that runs the functions you implemented and generates the figures described in your assignment write-up. You will lose credit if your code is absent or cannot be run.

Store your write-up and code in a single directory named hw1_yourID (for example hw1_12D423222) and submit it in moodle.

Do not wait till the last minute to start the Assignment as you will require at least at least 8 hours to complete all the questions.

Late submission policy:

Before the solution key is uploaded in moodle: If your original score is S and you submitted the HW X hours after the deadline, your score will be $S \exp(-X/24)$.
After the solution key is uploaded in moodle: 0 credit.

Discerning timing dependent signals

In this assignment, we will understand how integrate and fire neurons with distinct synapses can distinguish timing dependent stimulus with very similar statistical characteristics.

Problem 1: AEF neuron driven by a synapse receiving Poisson stimulus

(a) Create a Poisson stimulus with $T = 500$ ms, $\Delta t = 0.1$ ms and $\lambda = 10/s$. Determine the time instants when stimulus arrives, t_k.   **2 points**

(b) Assume that this stimulus arrives at an AEF RS neuron (from HW1) through a synapse. Then, the total current flowing into the neuron through the synapse at any time t will depend on the stimulus arrival times prior to time t (Say $t^1, t^2, t^3, \ldots t^n$, with $t^n < t$). We will model the current at time t according to the expression

$$I_{app}(t) \quad = \quad I_0 w_e \sum_{m=1}^{n} \left[e^{-(t-t^m)/\tau} - e^{-(t-t^m)/\tau_s} \right] \tag{1}$$

This form of modeling the synaptic current is particularly amenable for hardware implementation, as it depends only on the synaptic strength and stimulus times (and not on post-synaptic potentials).

Assume that $I_0 = 1$ pA and strength of the synapse is given by $w_e = 500$. Also, $\tau = 15$ ms represents the time constant of the membrane and $\tau_s = \tau/4$ is the synaptic time constant.

Plot the response of the neuron along with the input current and stimulus to convince yourself that the neuron emits spikes only if closely spaced stimulus are present.   **8 points**

Problem 2: AEF neuron driven by multiple synapses

We would like to use this framework to understand the dynamics of a neuron that is receiving inputs from multiple synapses.

Assume we have a total of $N_s = 100$ synapses driving the neuron, whose connection strengths are gaussian distributed, with a mean strength of w_0 and standard deviation of σ_w. As you may expect, larger magnitudes of w_0 has higher probability of causing the neuron to spike. (You may use the `randn` function from MATLAB to create the strengths for the synapses).

(a) Assume $w_0 = 50$ and $\sigma_w = 5$ and create a population of N_s synapses. For each synapse, create a Poisson stimulus with $T = 500$ ms, $\Delta t = 0.1$ ms and $\lambda = 1/s$. Determine the total current flowing into the neuron based on equation (1) and plot the response of the neuron. How many spikes are issued by the neuron in the interval $[0, T]$?   **6 points**

(b) For the same stimulus in (a), plot the total current and the response of the neuron, but with a new configuration of synaptic strengths defined by a guassian distribution whose mean is $w_0 = 250$ and $\sigma_w = 25$. How many spikes are issued by the neuron in the interval $[0, T]$? **4 points**

Problem 3: Adjusting the weights to elicit a spike response

As you would have learnt from problem (3), a naive way to elicit a spike response from the neuron is to increase the strength of every synapse. But, we could actually do this in a more efficient manner.

Start with a synaptic configuration given by $w_0 = 50$ and $\sigma_w = 5$ and the same stimulus condition as in Problem 3. Look at the response of the neuron and determine the time instant, t_{max} where the neuron membrane potential was the maximum in the interval $[0, T]$.

Identify the synapse (call it S_p) that received a stimulus just prior to t_{max}. (i.e., If the kth synapse received its stimulus at time t_k, find the synapse for which the quantity $\Delta t_k = (t_{max} - t_k)$ is the minimum (and positive).)

You would expect that if you left all other synapses unaltered, but just increased the magnitude of S_p by a sufficient amount, the neuron will have a higher probability to spike. Verify if this is indeed the case.

A more general way to implement this scheme of changing the synaptic strengths in order to elicit a spike is to modify the strength of each synapse depending on its value of Δt_k. We will use the rule

$$\Delta w_k = +w_k \gamma (e^{-\Delta t_k/\tau} - e^{-\Delta t_k/\tau_s}) h(\Delta t_k) \tag{2}$$

where γ is a parameter that controls the rate of learning. $h(x)$ is the heaviside step function defined as $h(x) = 1$ for $x \geq 0$ and $h(x) = 0$ otherwise.

With this rule, synapses that have the potential to contribute the most to a neuron spike are selectively increased. It might be necessary to apply this rule to the entire population of synapses multiple times, depending on the value of the learning rate. Also, typical electronic devices that could be used as synapses have an on-off ratio of about $10 - 50$; hence, assume that the maximum synaptic strength can not exceed 500.

(a) Write a program to implement this learning rule and determine the number of iterations that are required to cause the neuron to create at least one spike for $\gamma = 1$. The output of your code should be the set of weights after training. **14 points**

(b) Plot, on the same graph, the value of Δw_k vs Δt_k for every synapse and every training iteration. **1 points**

Problem 4: Adjusting the weights to remove all spike responses

You may also realize now that a naive way to remove all spike response from the neuron is to decrease the strength of every synapse. Again, we could do this in a more efficient manner.

Start with a synaptic configuration given by $w_0 = 250$ and $\sigma_w = 25$ and the same stimulus condition as before. Look at the response of the neuron and determine the time instants, $t_{sp}^1, t_{sp}^2, t_{sp}^3 \ldots$, where the neuron spiked in the interval $[0, T]$.

For each spike, identify the synapse, (say S_q) that received a stimulus just prior to the spike. (i.e., If the kth synapse received its stimulus at time t_k, then, find the synapse for which the quantity $\Delta t_k^m = (t_{sp}^m - t_k)$ is the minimum (and positive).) By reducing the strength of the synapse S_q appropriately, you should be able to lower the probability of that particular spike. One has to do this exercise iteratively for every spike (by calculating Δt_k for each spike) to remove all the spikes issued by the neuron.

A general way to implement this scheme of changing the synaptic strengths in order to remove all spike responses is to modify the strength of each synapse depending on the value of Δt_k, (measured as the difference between stimulus time and the time when the neuron issued the next immediate spike). We will use the rule

$$\Delta w_k = -w_k \gamma (e^{-\Delta t_k/\tau} - e^{-\Delta t_k/\tau_s}) h(\Delta t_k) \tag{3}$$

where γ is the same parameter as in problem 4. With this rule, synapses that has the potential to contribute the most to a neuron spike is selectively decreased. As before, it might be necessary to apply this rule to the entire population of synapses multiple times, depending on the value of the the learning rate. You should also assume that the synaptic strength can not fall below 10.

(a) Write a program to implement this learning rule and determine the average number of iterations that are required to remove all the spikes of the neuron. The output of your code should be a set of weights that results in removing all the spike from the neuron. In this experiment, start with a new population of synaptic strengths with $w_0 = 250$ and $\sigma_w = 25$, but the same pattern of stimulus as before. **14 points**

(b) Plot, on the same graph, the value of Δw_k vs Δt_k for every sypanse and every training iteration. **1 points**

Problem 5: Discriminating stimuli with similar statistical characteristics

Now, assume that the synaptic population has a $w_0 = 200$ and $\sigma_w = 20$. Create two stimulus patterns S_1 and S_2 with $T = 500$ ms, $\Delta t = 0.1$ ms and $\lambda = 1/s$. We would like to now determine a single set of weights for the synaptic population which will elicit opposite spike responses when these stimuli are presented.

(a) Determine the response of the neuron for stimulus S_1 and S_2 for the same starting synaptic strengths w_0. **2 points**

(b) Apply the routine to remove all the spikes with stimulus S_2, starting with w_0. Call the resulting weight population w_1. **4 points**

(c) Determine the response of the neuron for stimulus S_1, but with synaptic strength w_1 from part (b) above. If there is a spike response, you have now obtained a weight configuration that allows you to distinguish S_1 and S_2. **6 points**

If there is no spike response when S_1 is presented, apply the routine to create a spike for stimulus S_1, but with synaptic strengths w_1. Call the new weight configuration w_2. Go back to step (b) and iterate starting with w_2, till you are able to obtain a single set of weights that causes a spike response for presentation of S_1 and no spike for presentation of S_2.

(d) Now, determine the weights necessary to cause a spike response for presentation of S_2 and no spike for presentation of S_1. **8 points**

This assignment is loosely based on what is known as the 'Tempotron'. For those interested in more details, refer to: *The tempotron: a neuron that learns spike timing based decisions, Robert Gutig & Haim Sompolinsky, Nature Neuroscience 9, 420 - 428 (2006).*