Lecture 3 – Integrate-and-fire neuron model

Computational Neuroscience Summer Program

June, 2010

Motivation. This lecture builds on the simple model neuron that we developed in the last lacture by adding in action potentials (APs). Rather than modeling the biophysical basis of the AP, in this model we manually cause the neuron to spike when its membrane voltage reaches a threshold value.

Recap. Our working model is that neurons are like capacitors connected to resistors – the cell membrane stores charge, which can leak out through ion channels. As developed in the last lecture, the equation we'll be using for all model neurons is:

$$\tau_m \frac{dV}{dt} = E - V + R_m I_e$$

Also from the previous lecture, we can solve for $V(t)$ as follows:

$$V(t) = V_\infty + (V(0) - V_\infty)e^{\frac{-t}{\tau_m}}$$

Running a simulation – the "integrate" part of the model. Running a simulation entails computing changes in the cell's membrane voltage for each iteration of the simulation (dt ms). We can use the same equation as above, but replace $V(t)$ with $V(t+dt)$ (i.e. the voltage in the next time step of the simulation) and $V(0)$ (the starting voltage) with $V(t)$:

$$V(t+dt) = \text{(where we are going)} + \text{(distance)}e^{\frac{-t}{\tau_m}} = V_\infty + (V(t) - V_\infty)e^{\frac{-t}{\tau_m}}$$

In practice, these computations work best for small values of dt (in most cases we'll use $dt \leq 0.1$ ms).

Running a simulation – the "fire" part of the model. Now the integrate-and-fire model is almost entirely in place. The only thing we need to add is the rule that when $V(t+dt) \geq V_{thresh}$, simulate an action potential by setting $V(t) = V_{peak}$ and $V(t+dt) = V_{reset}$. V_{thresh} is generally somewhere around -55 mV, V_{peak} is around 40 mV, and V_{reset} is around -80 mV. In the next lectures we'll discuss the biophysical basis of why the neuron depolarizes (increases its membrane voltage) suddenly during the start of the action potential and why the membrane voltage becomes hyperpolarized (decreased) after the action potential is fired.

Computing firing rate. This is straightforward. We can simply count up the number of spikes that were fired during the simulation (i.e. times when $V \geq V_{thresh}$ and divide by the length of time we were simulating.

Analytic solution for firing rate. While the full integrate-and-fire simulation is often useful (and is necessary if you want to model things like spike timing), it turns out that there is an analytic method for computing the expected firing rate of the model, given a constant external current I_e. We start with the equation for finding the membrane voltage at time t:

$$V(t) = V_\infty + (V(0) - V_\infty)e^{\frac{-t}{\tau_m}}$$

We can then solve for t as follows:

$$V(t) - V_\infty = (V(0) - V_\infty)e^{\frac{-t}{\tau_m}}$$

$$\frac{V(t) - V_\infty}{(V(0) - V_\infty)} = e^{\frac{-t}{\tau_m}}$$

$$ln(\frac{V(t) - V_\infty}{V(0) - V_\infty}) = \frac{-t}{\tau_m}$$

$$\tau_m ln(\frac{V(t) - V_\infty}{V(0) - V_\infty}) = -t$$

$$t = -\tau_m ln(\frac{V(t) - V_\infty}{V(0) - V_\infty})$$

Now let's suppose our model neuron has just fired a spike in the previous timestep of our simulation. We start by setting $V(0) = V_{reset}$. We next need to know how long it is until the neuron next fires a spike (the inter-spike interval, t_{isi} – or, in other words, the time t at which $V(t) = V_{thresh}$ after starting at $V(0) = V_{reset}$. Plugging in the appropriate values, we can compute t_{isi} as follows:

$$t_{isi} = -\tau_m ln(\frac{V_{thresh} - V_\infty}{V_{reset} - V_\infty})$$

Recall that $V_\infty = E + R_m I_e$. Thus

$$t_{isi} = -\tau_m ln(\frac{V_{thresh} - (E + R_m I_e)}{V_{reset} - (E + R_m I_e)}) = -\tau_m ln(\frac{V_{thresh} - E - R_m I_e}{V_{reset} - E - R_m I_e})$$

The firing rate (r_{isi}) is the inverse of the inter-spike interval:

$$r_{isi} = (-\tau_m ln(\frac{V_{thresh} - E - R_m I_e}{V_{reset} - E - R_m I_e}))^{-1}$$

Putting it all together. To run the simulation, start with the basic model neuron equation:

$$\tau_m \frac{dV}{dt} = E - V + R_m I_e$$

Now solve for dV:

$$\frac{dV}{dt} = \frac{E - V + R_m I_e}{\tau_m}$$

$$dV = (\frac{E - V + R_m I_e}{\tau_m})dt$$

Start the simulation by setting $V(0) = E$. With each timestep set $V(t + dt) = V(t) + dV$. You'll need to re-compute dV for each time-step given $V(t)$ and $I_e(t)$ for the appropriate time t. Remember to include the rule for firing a spike (and resetting) when $V > V_{thresh}$ – otherwise the neuron won't fire spikes.

General MATLAB stuff. Set up your environment:

```
E = -70;         %mV
c_m = 10;        %nF / mm^2
r_m = 1;         %M ohm * mm^2
A = 0.025;       %mm^2
V_reset = -80;   %mV
V_thresh = -55; %mV
V_peak = 40;     %mV

dt = 0.1;        %ms
t = 1:dt:1000;   %ms
```

Now loop:

```
V(1) = E;
for i = 2:length(t)
  if V(i-1) > V_thresh
    {fire a spike}
  else
    {compute dV}
    V(i) = V(i-1) + dV;
end
```

For the problem set, you should write a function that runs the integrate-and-fire model for a given set of parameters. Your function should at return the firing rate (computed numerically, not using the r_{isi} equation). You also might want to have it return the vector of V over time, depending on how you set up your code.