Exercises - Larry Abbott, June 10

1. Simulate the model
$$\frac{dx_i}{dt} = -x_i + g \sum_{j=1}^{N} J_{ij} \tanh(x_j) \tag{1}$$
for $i = 1, 2, \ldots, N$ with each element J_{ij} chosen independently from a Gaussian distribution with zero mean and variance $1/N$. Take $N = 1000$, and examine what happens for values of g in the range from 0.5 to 2. Initialize x_i for $i = 1, 2, \ldots N$ randomly in a reasonable range. Compare with reference 1, below.

2. Extend the model in (1), with $g = 1.5$, by adding an external input, so that
$$\frac{dx_i}{dt} = -x_i + g \sum_{j=1}^{N} J_{ij} \tanh(x_j) + u_i A \cos(\omega t) \tag{2}$$
with u_i, for $i = 1, 2, \ldots, N$, chosen independently from a uniform distribution ranging from -1 to 1. Look at what happens to the activity as a function of A and ω. Compare with reference 2, below.

3. Extend the model in (1), with $g = 1.5$, by adding a term
$$\frac{dx_i}{dt} = -x_i + g \sum_{j=1}^{N} J_{ij} \tanh(x_j) + u_i z \tag{3}$$
with u_i chosen as in (2) , and
$$z = \sum_{i=1}^{N} w_i \tanh(x_i)\,. \tag{4}$$
Adjust the weights w_i, for $i = 1, 2, \ldots, N$, so that z matches a target function
$$f = \cos\left(\frac{2\pi t}{50}\right)\,. \tag{5}$$
Set the initial weights randomly from a uniform distribution ranging from $-1/\sqrt{N}$ to $1/\sqrt{N}$. Also, initialize an $N \times N$ matrix with elements P_{ij}, for $i, j = 1, 2, \ldots, N$, to $P_{ij} = 0$ if $i \neq j$ and $P_{ii} = 1$ for all i. Then, adjust the weights by applying the FORCE learning rule, for all i,
$$w_i \to w_i + c(f - z)q_i\,, \tag{6}$$
where
$$q_i = \sum_{j=1}^{N} P_{ij} \tanh(x_j) \tag{7}$$

and

$$c = \frac{1}{1 + \sum_i q_i \tanh(x_i)} . \tag{8}$$

Finally, update the matrix P by

$$P_{ij} \to P_{ij} - c q_i q_j . \tag{9}$$

After a while, stop the weight adjustment and the network should produce $z = f$ autonomously. If this works, feel free to try more complex functions $f(t)$. For comparison, see reference 3.

4. Build a network of integrate-and-fire model neurons, satisfying

$$\tau_{\rm m} \frac{dV_i}{dt} = V_{\rm rest} - V_i + I_0 + s_i \tag{10}$$

for $i = 1, 2, \ldots, N$, with $N = 1000$, $\tau_{\rm m} = 20$ ms, $I_0 = 11$ mV and $V_{\rm rest} = -60$ mV. The neuron spikes when V_i reaches a threshold value $V_{\rm th} = -50$ mV and then is reset to $V_{\rm reset} = -60$ mV. Following a spike, V_i is clamped to $V_i = V_{\rm rest}$ for 5 ms. Finally, V_i is not allowed to fall below -80 mV. In other words, following an update using the above equation, if $V_i < -80$, V_i is set to -80.

The synaptic current s_i satisfies the equation

$$\tau_{\rm s} \frac{ds_i}{dt} = -s_i \tag{11}$$

for all i with $\tau_{\rm s} = 10$ ms. When some neuron j fires a spike, s_i, for all $i = 1, 2, \ldots, N$ except $i = j$, is updated by

$$s_i \to s_i + \mu + g J_{ij} \tag{12}$$

with $g = 3$ mV, $\mu = -0.5$ mV, and all the elements J_{ij} chosen independently from a Gaussian distribution with zero mean and variance 1. Initialize V_i for $i = 1, 2, \ldots N$ randomly, see what happens and then do whatever you want with the model.

5. If you have the energy, construct the model in reference 4.

References

1) H. Sompolinsky, A. Crisanti, and H. J. Sommers (1988) Phys. Rev. Lett. 61, 259.

2) K. Rajan, L.R. Abbott and H. Sompolinsky (2010) Phys. Rev. E 82:011903.

3) D. Sussillo and L.F. Abbott, (2009) Neuron 63:544-557.

4) M, Boerlin, C.K. Machens and S. Deneve (2013) PLoS Comput Biol. 9:e1003258.