Multi Neuron Spike Train Data Simulator

V. Raajay

May 31, 2010

Contents

1 Introduction **2**

2 The Simulation Model **2**

3 Simulator Description and Usage **4**

3.1 Contents 4

3.2 Embedding Connections through Probabilities 5

3.3 Parameters of Simulation 5

3.4 Running the Executable 5

1 Introduction

This program is used to simulate the spiking activity of a group of neurons. Each neuron is modeled as a non-homogeneous random process where the inter-spike interval of every neuron is one the following distributions : a. Poisson b. Gamma

Strong excitatory or inhibitory connections among neurons can be specified by the user in terms of conditional probability. For example, if we want a strong connection between Neuron A and Neuron B (ie. firing of Neuron A triggering (or inhibiting) the firing of Neuron B after a time t), we specify it by a parameter $P(B/(A,t)) = p$. (For excitatory connections the value of p is much higher than random chance and for inhibitory connections it is low.)

The simulation model used is also capable of handling higher order interactions (not just pairwise) among neurons. Here we can include the effect of firing of more than one neuron on the firing rate of another neuron. For example, we can set probabilities like $P(C/(A, t_1, B, t_2)) = p$. This means that the probability of firing of neuron C, t_1 units after firing of A *and* t_2 units after firing of B is p.

2 The Simulation Model

Let N denote the number of neurons in the network and let $Z_i(t)$ denote the spiking process of the i^{th} neuron. Each Z_i is taken to be an inhomogeneous Poisson process whose rate is constant on each interval $[n\Delta T,\ (n+1)\Delta T)$, $n = 0, 1, \cdots, L$ where L represents the total time duration in units of ΔT. We normally take $\Delta T = 1ms$. Let $\lambda_i(k)$ denote the rate of $Z_i(t)$ for $t \in [(k-1)\Delta T,\ k\Delta T)$.

Let Y_{ik}, $k = 1, 2, \cdots T$, $i = 1, 2, \cdots, N$, be binary random variables defined as: $Y_{ik} = 1$ if there is a spike event of Z_i in the interval $[(k-1)\Delta T,\ k\Delta T)$; and $Y_{ik} = 0$ otherwise.

If we are given $\lambda_i(k)$ for all i, k then we can simulate the Z_i processes as follows. For each k, $k = 1, 2, \cdots$, we do the following: For $i = 1, 2, \cdots, N$, we generate random variables, ξ_i, exponentially distributed with parameter $\lambda_i(k)$; then, for each i, if $\xi_i < \Delta T$ then we put a spike for neuron i at time $(k-1)\Delta T + \xi_i$ else we put no spike for i in the interval $[(k-1)\Delta T,\ k\Delta T)$

The set of neurons are interconnected through synapses and the connection from i to j is characterized by a weight parameter w_{ij} and a delay parameter τ_{ij}. We take it that the delay τ_{ij} is specified in units of ΔT. For high order connections (say third order involving neurons i,j,l) we have a weight parameter w_{ijl} and delays τ_{il} and τ_{jl}. For only pairwise interactions, parameters of the form w_{ijl} are zero. The firing rates, $\lambda_i(k)$, of neurons are influenced by the inputs received from other neurons. Let Y_{ik}, $k = 1, 2, \cdots T$, $i = 1, 2, \cdots, N$, be binary random variables defined as: $Y_{ik} = 1$ if there is a spike event of Z_i in the interval $[(k-1)\Delta T,\ k\Delta T)$; and $Y_{ik} = 0$ The firing rates at k^{th} time-step are computed as

$$\lambda_i(k) = \frac{\lambda_m}{1 + \exp(-\theta_i - \sum_{j=1}^{N} w_{ji} Y_{j(k-\tau_{ji})} - \sum_{j=1}^{N} \sum_{l=1}^{N} w_{jli} Y_{j(k-\tau_{ji})} Y_{l(k-\tau_{li})} - ...)}. \quad (1)$$

Here θ_i is a parameter that fixes the nominal firing rate for the neuron i. (This is the firing rate when the neuron receives no input). The constant λ_m denotes the maximum possible firing rate attainable and its value is fixed so that the probability of atleast one spike in an interval of length ΔT is 0.99 and this value is the solution of the equation

$$1 - \exp(-\lambda_m \Delta T) = 0.99 \tag{2}$$

To get a specific network, the user specifies, for all i, the nominal firing rate, λ_0^i, of neuron i and the strengths and delays of all interconnections. Given the value of λ_0^i, the value of θ_i is fixed as

$$\theta_i = -ln\left(\frac{\lambda_m}{\lambda_0^i} - 1\right) \tag{3}$$

The strengths of interconnections (or synapses) are specified in terms of conditional probabilities. Let p_{ij} denote the strength of connection from i to j and it is taken to be the conditional probability that there is atleast one spike from j in an interval $[(k-1)\Delta T,\ k\Delta T]$ given that there is atleast one spike from i in the interval $[(k - \tau_{ij} - 1)\Delta T,\ (k - \tau_{ij})\Delta T]$ and that all other input to j is zero. Here τ_{ij} is the delay associated with this connection. Similarly, for higher order connections let p_{ijl} denote the strength of the connection from i and j to l. Given p_{ij}. we can calculate w_{ij} as

$$w_{ij} = -\theta_j - ln\left(\frac{\lambda_m}{\lambda'} - 1\right) \tag{4}$$

and given p_{ijl}, we can calculate w_{ijl} as

$$w_{ijl} = -\theta_j - ln\left(\frac{\lambda_m}{\lambda'} - 1\right) - w_{il} - w_{jl} \tag{5}$$

where λ' is the solution of the equation

$$1 - \exp(-\lambda' \Delta T) = p_{ij} \tag{6}$$

To use the simulator, we specify the nominal firing rate of each neuron and the strengths (in terms of conditional probability as explained above) and delays, τ_{ij} (in units of ΔT) for all connections. Then we can determine the parameters θ_i and w_{ij}. Then for each k, we obtain $\lambda_i(k)$ for all i and this is used to simulate the Z_i processes as explained earlier. We normally specify a network which has many random interconnections (i.e., with the strengths being set randomly) and some specific connections to constitute the patterns or microcircuits by giving high strength for these connections.

We note here that the nominal firing rate as well as the effective conditional probabilities in our system would have some small random variations. As explained above, we fix θ_j so that on zero input the neuron would have the nominal firing rate. However, all neurons would have synapses with randomly selected other neurons and the strengths of these synapses are also random. Hence, even in the absence of any strong connections, the firing rates of different neurons keep fluctuating around the nominal rate that is specified. Since we choose random connections in such a way that in an expected sense the input into a neuron is zero, the average rate of spiking should be close to the nominal rate specified. We also note that the way we calculate the effective weight for a given conditional probability is also approximate and we chose it for simplicity. If we specify a conditional probability for the connection from A to B, then, as given by (4)–(6), the weight of the connection is fixed so that the probability of B firing at least once in the next ΔT interval given that A has fired in an appropriate interval earlier is equal to this conditional probability *when all other input into B is zero.* But since B would be getting small random input from other neurons also, the effective conditional probability would also be fluctuating around the nominal value specified.

We have also used this simulator for generating spike trains where inter-spike intervals are (non-homogeneous) Gamma distributed instead of being exponential as in case of Poisson. (Our notation is: Gamma distribution with parameters α and β has density given by $f(x) = x^{(\alpha-1)}\beta^{\alpha}\exp(-\beta x)/\Gamma(\alpha)$). The changes needed in the simulator are as follows. We simulate the $Z_i(k)$ by generating ξ_i as before; but now the ξ_i has Gamma distribution with $\alpha = 2$ and $\beta = \lambda_{ik}$. Thus now, the spiking processes of neurons are such that inter-spike intervals are Gamma distributed with a time-varying rate parameter. The λ_{ik} are updated, as before, using eq. (1). But now λ_m is a solution of

$$1 - \exp(-\lambda_m \Delta T)(1 + \lambda_m \Delta T) = 0.99 \tag{7}$$

The θ_i are determined as before using the above λ_m. The weights w_{ij} are computed using eq. (4) as before; however the λ' in this equation is determined as the solution of

$$1 - \exp(-\lambda' \Delta T)(1 + \lambda' \Delta T) = p_{ij} \tag{8}$$

With these changes, we can still specify the connection strengths as conditional probabilities (p_{ij}) and with the weights determined as above we will have spike trains with the required embedded connection strengths. A minor difference is that the earlier equations, namely, eqs. (2) and (6) have simple closed-form solutions. Now we solve eqs. (7) and (8) using Newton-Raphson method.

3 Simulator Description and Usage

3.1 Contents

The directory contains the executable (*neuronSpikeSimulator*) and the following files:

- *inputfile.txt* : Used to set the parameters of the simulator
- *episodeFile.txt* : Used to represent the connections (in terms of probabilities) the user wants to embed.
- *stimulusFile.txt* : A text file that can be used to provide external stimulus to some neurons (Analogous to external stimuli provided to neurons through electrodes in a MEA). Each line in this file is of the form *(Neuron-ID, time of spiking)*. An entry of the form (A, t_1) means that the user wants the neuron A to fire unconditionally at time t_1.

3.2 Embedding Connections through Probabilities

As described in the 'Introduction' Section, strong connections among neurons can specified by means of conditional probabilities. The user can input these probability values through entries in *episodeFile.txt*.

The format of *episodeFile.txt* is as follows. The first line contains the number of connections (say m) the user wants to specify. Each of the next m lines represents one of the m connections. A line, that represents the connection $P(C/(A, t_1, B, t_2)) = p$, will be of the form

$$\boxed{3\ C\ A\ t_1\ B\ t_2\ p}$$

Here, 3 represents the number of interacting neurons, C is the neuron whose spiking is affected, A and B are the neurons that have third-order influence on C with time delays t_1 and t_2 respectively and p is the required probality.

3.3 Parameters of Simulation

The parameters to simulation are entered in the file *inputfile.txt*. The paramters and their description are as follows:

- *numberOfNeurons* : It denotes the total number of neurons whose spiking activity has to be simulated. The neuron ID's are integers ranging from one to *numberOfNeurons*.
- *tUpdate* : It represents the value ΔT (discussed in previous section) in seconds.
- *simulationTime* : Represents the total time duration (in seconds) of the spiking simulation. The spiking activity of neurons is simulated from zero to *simulationTime* seconds.
- *spikeDistribution* : Represents the distribution of the inter-spike intervals of a single neuron. It can be either *poisson or gamma* in this implementation. Other various distributions will be implemented further.
- *randomFrequency* : It represents the normal independent spiking rate (in Hertz) of a single neuron.
- *percentageConnections* : Apart from the user input connections, the simulator also provides the option of introducing pairwise random connections among neurons. *percentageConnections* represents the percentage of such connections.
- *pRandHigh, pRandLow, delayRandHigh, delayRandLow* : The delays and the probability values for random connections are chosen uniformly in the ranges [*delayRandLow,delayRandHigh*] and [*pRandLow,pRandHigh*] respectively.

3.4 Running the Executable

To simulate the spiking activity of the neurons we run the executable ('neuronSpikeSimulator' in linux and 'neuronSpikeSimulator.exe' in windows) from the same directory as the other files. The executable reads the probabilities and parmater information from the respective files, generates the spike train and dumps the output to a file *stream.txt*. Each line in *stream.txt* represents the spiking of a single neuron. It is of the form *(Neuron-ID,time of spiking)*. Note that, the lines are in order of the spiking times of neurons.