GNT short term internship Feb 2016 - June 2016

Jong Hoon LEE

2016-06-05

1 Spiking networks and predictive coding

1.1 the model

In the first part of my internship I reproduce a model that executes predictive coding using balanced spiking networks. (*Machen, Deneve, 2013, Predictive Coding of Dynamical Variables in Balanced Spiking Networks*) The basic model strategy is represented in Fig. 1. Let us consider a linear dynamical system describing the temporal evolution of a variable x:

$$\dot{x} = \lambda_s x + c(t)$$

where λ_s is the leak of the sensory integrator and c is a time varying external input.

We build a network of N neurons taking $x(0)$ and c as inputs, that can reproduce the temporal trajectory of x. Specifically, we want to read an estimate $\hat{x}$ from the network's spike train $\mathbf{o}(t) = \sum_k \delta(t - t_i^k)$ where t_i^k is the time of the $k^t h$ spike in neuron i.

We assume that the estimate $\hat{x}$ is is obtained by a weighted, leaky integration of the spike trains,

$$\dot{\hat{x}} = -\lambda_d \hat{x} + \mathbf{\Gamma o}(t)$$

where Γ contains the decoding or output weights of all neurons and λ_d is the read-out decay rate We also assume that the network minimizes the distance between x and $\hat{x}$ by optimizing over the spike times t_i^k. the decoding weights are chosen a-priori. In order to track the temporal evolution of $\hat{x}$ as closely as possible, the network minimizes the cumulative mean- squared error between the variable and its estimate, while limiting the cost in spiking. Thus, it minimizes the following cost function,

$$E(t) = \int_0^t du(\|x(u) - \hat{x}(u)\|_2^2 + \nu\|\mathbf{r}(u)\|_1 + \mu\|\mathbf{r}(u)\|_2^2)$$

where $\mathbf{r}$ is the vector of firing rates and parameters ν and μ control the cost-accuracy tradeoff. $\mathbf{r}$ is defined as

$$\dot{\mathbf{r}} = -\lambda_d \mathbf{r} + \lambda_d \mathbf{o}(t)$$

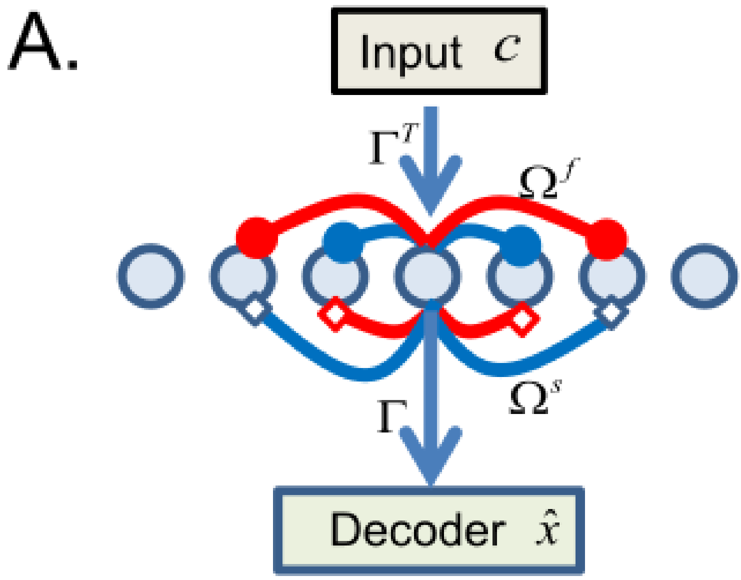

Figure 1: structure of network in original paper. The network receives input c(t) scaled by feedforward weights $\mathbf{\Gamma}^T$, which is internally processed through fast and slow recurrent connections Ω^s and Ω^f. The firing rates are then decoded in to $\hat{x}$. (*Machen et Deneve, 2013, Predictive Coding of Dynamical Variables in Balanced Spiking Networks*)

1.2 Network Dynamics

To derive the network dynamics, we will assume that a neuron can fire when this results in a decrease of $E(t)$. This prescription gives the firing rule

$$V_i(t) > T_i$$

with

$$\dot{V_i} = -\lambda_V V_i + \sum_{k=1}^{N} W_{ik} * o_k(t) + \mathbf{\Gamma}_i^T \mathbf{c}(t) + \sigma_V \eta(t)$$

where $-\lambda_V V_i$ is a leak term, $W_{ik}(t)$ is a weight matrix of connectivity filters explained below, and $\eta(t)$ corresponds to a white gaussian variant background noise. The weight matrix is defined as $W_{ik}(u) = \Omega^s_{ik} h_d(u) - \Omega^f_{ik}\delta(u)$ where $\Omega^f = \mathbf{\Gamma}^T\mathbf{\Gamma} + \mu{\lambda_d}^2\mathbf{I}$ represents a matrix of fast lateral connections. and

$$\Omega^s = \Gamma^T(\lambda_s + \lambda_d)\Gamma$$

represents slow connections.

1.3 Parameters

In this study we use $N = 100$, $totaltime = 1.2$ sec , σ_v =0.01, $\nu = 10^{-5}$, μ =10^{-6} dt =0.1msec, $\lambda_d = 0.1$ and $\lambda_v = 0.2$. The stimulus c is as plotted in Fig 2 first panel.

The connectivity matrix $\mathbf{\Gamma}$ is a matrix of size N and of value either $0.1 * j$ for excitatory neurons and $-0.1 * j$ for inhibitory neurons , where j is a weight parameter between 1 and 5. this allows us to modulate synaptic strength units between neurons. In our scenario, all neurons are connected with the same weight.

When neurons receive a stimulus their voltage changes, and multiple neurons reach a state where the voltage V_i is above the Threshold T_i. In this study we choose one of these neurons and force it to spike. After spiking, the voltage of all neurons are reset to a reset value given by Ω^f.

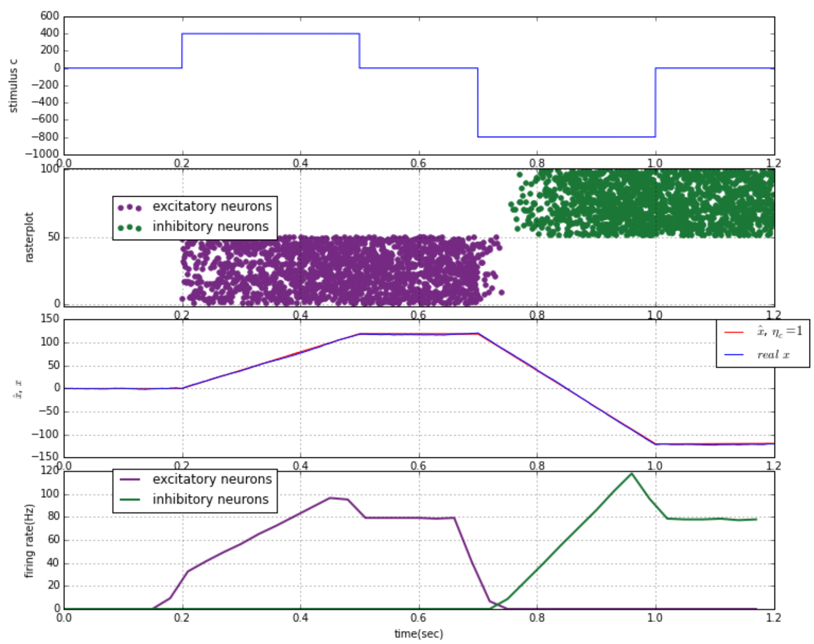

Figure 2: Stimuli(top). Rasterplot (above).Evolution of dynamic variables x and $\hat{x}$ (middle), $\sigma_V = 0.01$. Firing rates of both neuron populations(below)

1.4 Results : The perfect integrator

First we study the dynamics of a perfect integrator. In this case the leak of the sensory integrator λ_s is equal to 0. Results of this sensory integrator network is shown in figure 2. We observe that the perfect integrator increases in value when the stimulus is positive, and decreases when it is negative. We also observe that when the perfect integrator is increasing or constant with a positive value, only the excitatory neurons fire, and when it is decreasing or constant at a negative value, only inhibitory neurons fire. This is shown in more detail in the firing rates. firing rates rise for excitatory and inhibitory neurons in the presence of a positive or negative stimuli respectively.

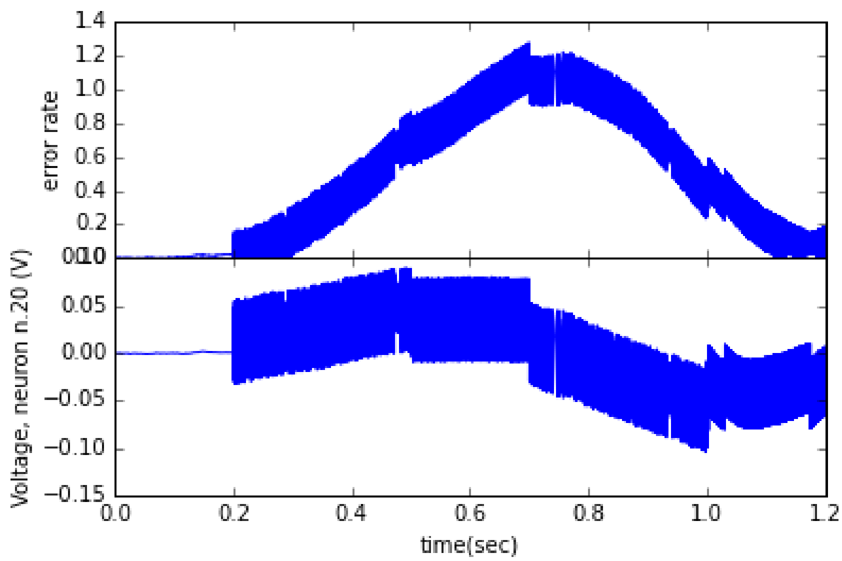

Figure 3: (above) error rate given by calculating the absolute difference between x and $\hat{x}$, (middle) the voltage of neuron 20, an excitatory neuron, (below) firing rate of the whole population of N neurons

For further analysis, we studied the error rate and the voltage evolution of a neuron. The error rate increases with time, but decreases when x goes below 0. however, it stays relatively small compared to the amplitude of dynamic variables).

1.5 Results : Sensory integration and tracking

For $\lambda_s = 0Hz$, the network is a perfect integrator of a noisy signal. However, when the dynamics of the system is larger than the decoder ($\lambda_s > \lambda_d$) then neural firing rates track the sensory signal, and model neurons have transient responses at the start or end of sensory stimulation, followed by a decay to a lower sustained rate (Fig. 9)

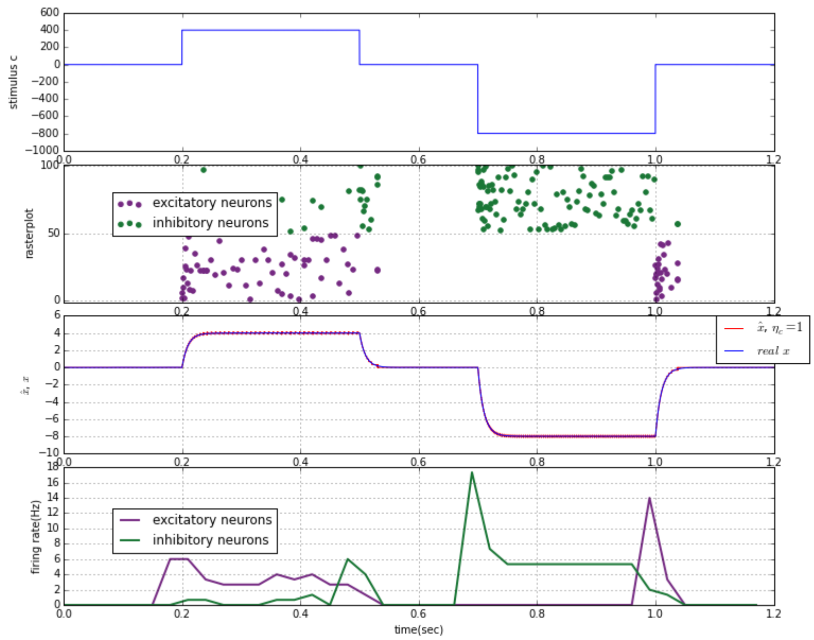

Figure 4: sensory tracking. $\lambda_s = 100Hz$

In this case fast connections alone were not enough, so we added slow connections

$$\Omega^s = \Gamma^T(\lambda_s + \lambda_d)\Gamma$$

to the weight matrix W_{ik}. With this new connection the network behavior (in error rate, voltage and firing rate) changes from the perfect integrator.

We observe in figure 4 that excitatory neurons show high activity when the integrator increases it's value to match the stimulus, then shows a lower sustained activity as long as the stimulus stays positive. when the stimulus is reverted to

0, inhibitory neurons fire. The population shows similar but opposite activity when the stimulus is negative.

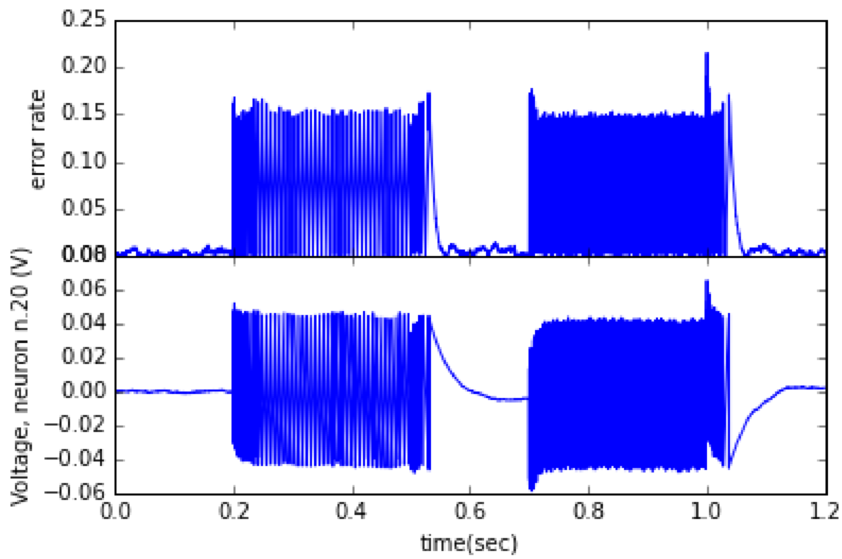

Figure 5: (above) error rate given by calculating the absolute difference between x and $\hat{x}$, (below) the voltage of neuron 20, an excitatory neuron

2 Fast connections with random connectivity

In the second part of the study we will modify the fast connectivity filter $\mathbf{\Omega}^f$ to add random connectivity between neurons. This gives the equation

$$\mathbf{\Omega}^f = \eta_c(\mathbf{\Gamma}^T\mathbf{\Gamma} + \mu\lambda_d^2\mathbf{I}) + (1 - \eta_c)\mathbf{W}_{ij}$$

where $\mathbf{W}_{ij}$ is a weight matrix following standard normal distribution with the same parameters as $\mathbf{\Gamma}$. and η_c represents the percentage of these random connectivities. When adding this random connectivity, the network shows different behavior for the same value of η_c. In both figure 6 and 8 we added 30 % of random connectivity. In figure 6 the network seems to globally predict the dynamic variable x, although with higher error rates. In figure 8 the network seeems to correctly track the variable x until at time = 0.4 seconds, where the network shows explosive behavior. We also observe that when this happens, the firing rate for both neurons are very high.

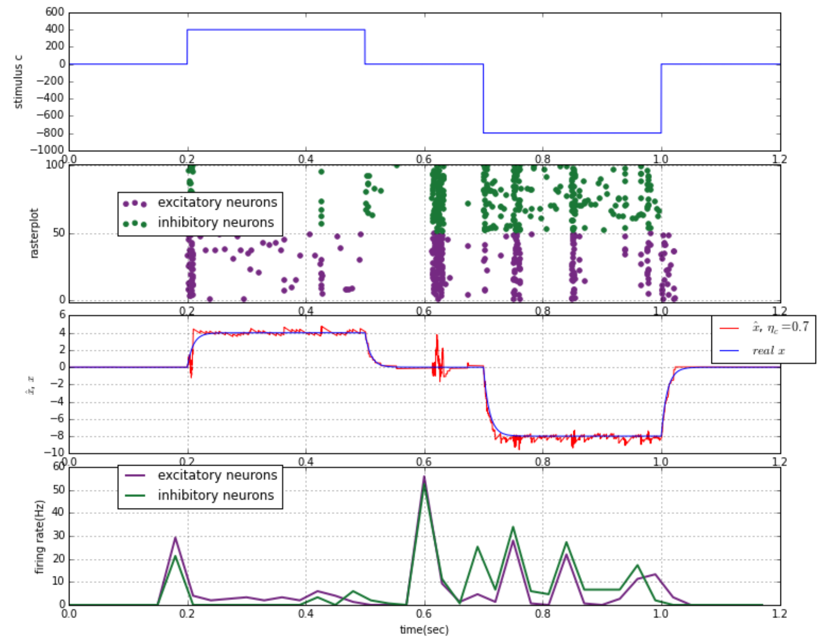

Figure 6: 30 % random connectivity: the network globally seems to predict the dynamic variable x closely but sometimes show explosive behavior (for example, at around 0.6 seconds). When this happens both excitatory and inhibitory fire at a high rate.

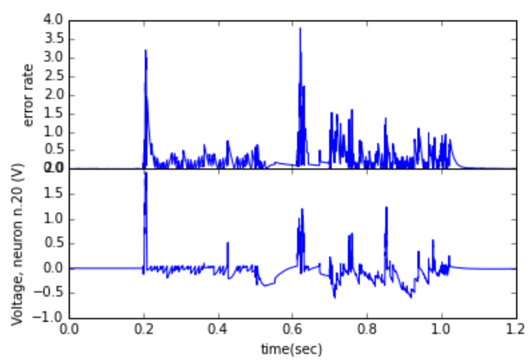

Figure 7: 30 % random connectivity: error rates and voltage of excitatory neuron 20.

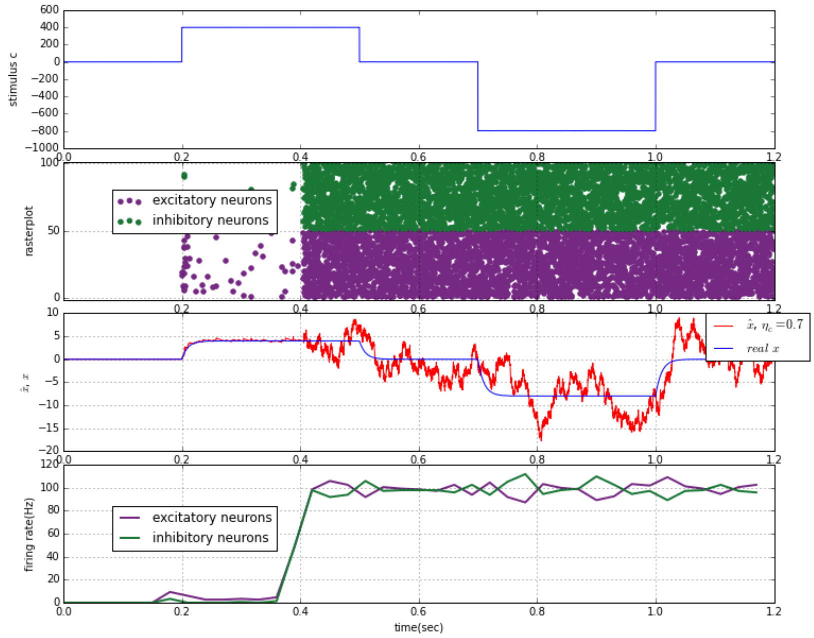

Figure 8: 30 % random connectivity, different simulation. In this simulation the network activity explodes at time = 0.4 seconds.

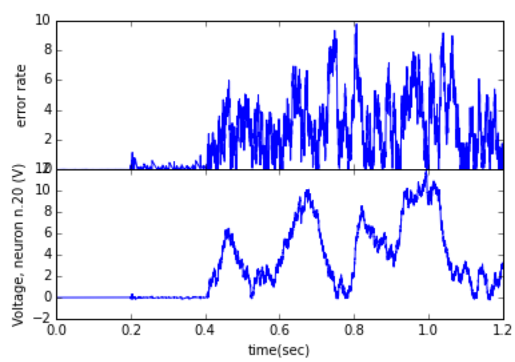

Figure 9: 30 % random connectivity: error rates and voltage of excitatory neuron 20. different simulation.

2.1 study of cost functions for different values of η_c

In this section we will study the cost function for different values of η_c. We observe in Figure 10 that for all values of η_c the cost function increases with time.

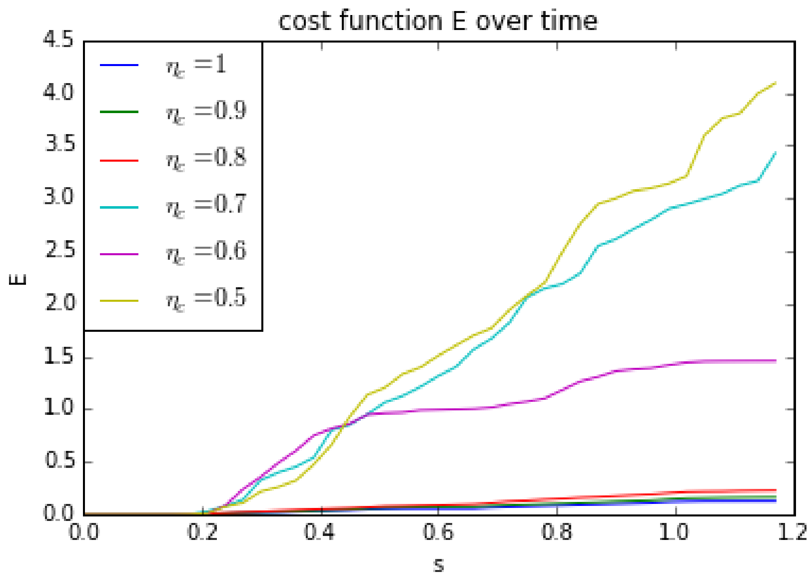

Figure 10: the evolution of the cost function E(t) for different values of η_c,

For lower values of η_c (and thus for higher random connectivity) the cost function seems to increase dramatically for η_c lower than 0.7. however, this is not always the case as we observe in figure 11: Only the cost functions for $\eta_c = 0.5$ and $\eta_c = 0.9$ shows explosive activity. It seems clear that the relationship between the cost function E and the value of η_c is not linear and subject to randomness. We observe in figure 12 that the strong increase in the cost function E corresponds to the explosive activity observed in the network. as long as E stays relatively small (as in the case of η_c the network predicts the sensory variable x.

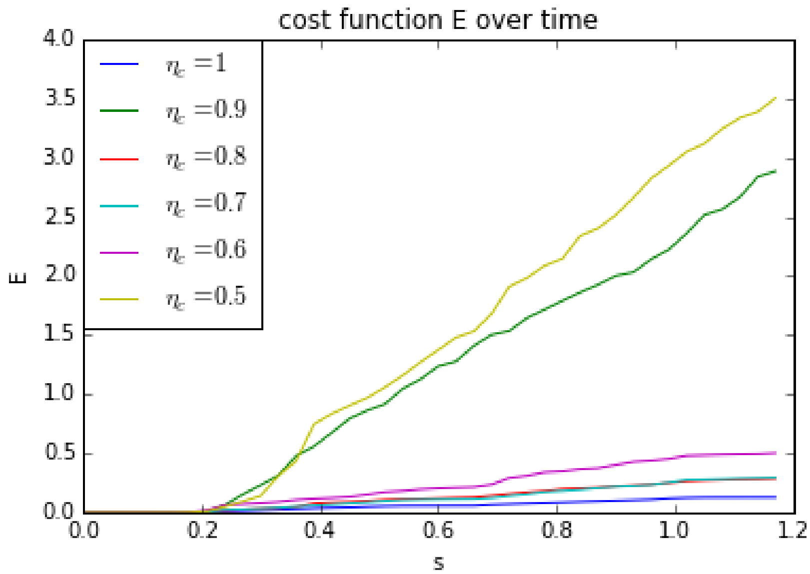

Figure 11: the evolution of the cost function E(t) for different values of η_c, different simulation

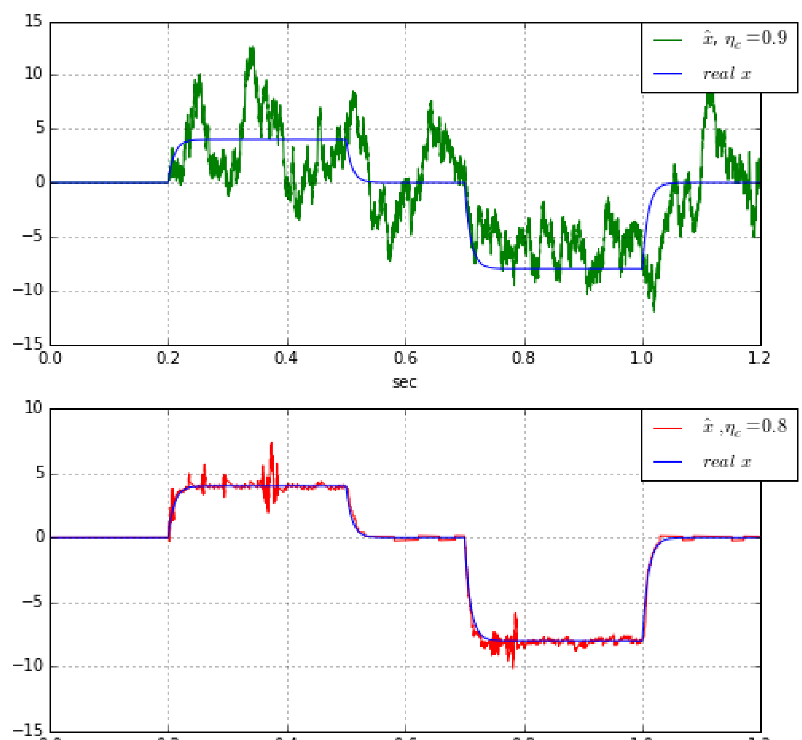

Figure 12: network activity for $\eta_c = 0.9$ (above) and $\eta_c = 0.8$ (below). The results are from the same simulation for figure 11