Solving AI

Kyra, Nischal

April 21, 2023

We consider rate based neural network model with populations of inhibitory and excitatory neurons. The dynamics of the network is given by:

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathbf{r} = -\mathbf{r}(t) + \mathbf{W} \cdot \phi\left(\mathbf{r}(t)\right). \tag{1}$$

The choice of nonlinearity ϕ is flexible and we should probably use ReLU or Sigmoid.

In particular, we devide the population of neurons to excitatory and inhibitory neurons and denote them with a subscript: $\mathbf{r}_{E/I}$. Practically, we will likely splid the population in a 20/80 fashion with a total of 1000 neurons.

This division also induces a division of the weight matrix $\mathbf{W}$ into block structure and we get 4 sets of matrices. The releveant one for us is the inhibitor to excitatory synapses represented by the matrix : $\mathbf{W}_{I\to E}$.

Practically, we initialize the whole matrix randomly with positive random values, and then enforce that the $I \to E$ connection is set to ~ 0. This is also the subset of connections that will be allowed some synaptic plasticty, while all other coonections basically remain forzen.

A crucial point here though: The connections are sparse and we should only connect $\sim 5\%$ of neurons presynaptically to any given neuron. We can do this by basically setting 95% of values in each row to 0. This should be fine on average. But this given rise to a subtelty in simulation: we only update the weights that were non zero at intialization. Meaning that the weights that were set to 0 initially don't change. We can do this practically by creating a dummy motif matrix $\mathbf{J}$ that is 1 where there is a connection and zero everywhere else and taking element wise product with $\mathbf{J}$ of weights during synaptic plasticity.

Finally, the synaptic plasticity dynamics a la' Spreckler is as follows for N excitatory neurons ans D inhibitory neurons:

$$\Delta\mathbf{W}_{I\to E} = \eta\left(\mathbf{r}_E \cdot \mathbf{r}_I^T - \mathbf{r}_E \cdot \rho\right). \tag{2}$$

where η is the learning rate and ρ is the desired firing rate written as a matrix of size $1 \times D$.

We should probably update the weights every 10th step of dynamics update, and run simulation to weight saturation. I also, think ρ should be rought equal to the $\mathbf{r}_{max}$ at initilization. Unsure about η though, maybe should be $0.1\times < r_0 >$.

What do you think? Hopefully this is enough to get started.