Think of a title later

Ansel Blumers and Ankan Ganguly

December 13, 2015

Abstract

Abstract goes here.

Contents

1 Introduction

Write an introduction here.

2 Methods

In this paper, we used the simulated Integrate and Burst (IB) model introduced in [1]. We then implemented STDP learning and Hebbian learning.

2.1 Integrate and Burst Model

Our IB model is an implementation of equations described in [1]. We provide a detailed description of those equations here, with more attention paid to the intuition behind these equations. There are N neurons connected in an all-to-all environment. Each neuron i bursts when its membrane potential, V_i, hits the threshold V_θ. We assume that every neuron bursts for T_{burst} time. While bursting, each neuron fires four times uniformly over the burst interval before resetting to V_{reset}.

When a neuron is not bursting, its potential is governed by a typical conductance based leaky integrate and fire model with built-in inhibition:

$$\tau_V \frac{\mathrm{d}V_i}{\mathrm{d}t} = -g^L(V - V^L) - g_i^E(V - V^E) - g_i^I(V - V^I) \tag{1}$$

The leak conductance, g^L, is assumed to be a homogeneous constant input, as is the leak potential V^L. Notice that in the absence of excitatory or inhibitory conductance, neuron i will tend to V^L. Therefore we can say that the leak potential is also equal to the rest potential.

The excitation potential, V^E, and the inhibition potential, V^I, act as upper and lower bounds on untethered (What was the word for IF potential plots without a firing threshold?) potential respectively.

The excitation conductance, g^L, is defined by the activity in neighboring neurons as well by external random stimulation:

$$g^L = Ws + W_0 b$$

Where W_{ij} is the strength of the synapse from neuron j to neuron i, W_0 is the conductance strength of external synapses and b is a Poisson random variable with frequency r_{in}. r_{in} was an input parameter. In [1] r_{in} was constant, but we chose to assign r_{in} a high value and anneal it to jump-start neural activity. s_i is the activation of neuron i. It is incremented each time neuron i fires, and it decays by

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\tau s$$

The inhibitory conductance, g^I, is defined as the sum of the adaptation inhibition, g^I_{ada}, and the global inhibition, g^I_{glob}. The adaptation inhibition is the internal inhibition generated by activation of a neuron. It is defined by the same equations as s_i, but with a constant multiplier A_a and a slower time-constant τ_{ada}.

2.2 Learning

The purpose of this paper is to replicate and test some of the claims made in [1]. One of the claims made is that the model used by [1] demonstrates how STDP can contribute to the synchronous regular synfiring chains exhibited in Zebra Finch (Where? Maybe cite second paper here?). We implemented STDP, but we also implemented a pure Hebbian learning rule to demonstrate that STDP learning is not necessary for the model to exhibit synchronous regular firing chains.

2.2.1 STDP Learning

We followed [1] precisely in our implementation of STDP learning. Define

$$K(t) = \begin{cases} e^{-t/\tau_{STDP}} & \text{if } t > 0 \\ -e^{-t/\tau STDP} & \text{if } t < 0 \\ 0 & \text{otherwise} \end{cases}$$

as the STDP kernel. For every pair of neurons i, j, let $t_i < t_j$ be two times when i and j fired respectively. Then According to STDP, the weight matrix element W_{ij} should increase proportional to $K(t_j - t_i)$. Notice that the change in W according to STDP is approximately anti-symmetric. [1] strengthened the nonlinearity of STDP by making STDP growth of a synapse proportional to the strength of that synapse (with an added factor so 0 weight synapses could grow). In particular, the paper defined:

$$\Delta_{ij}^{STDP}(t) = \left(\frac{W_{ij}(t-1)}{w_{max}} + 0.001\right) * \left(x_i(t)K(0)x_j(t) + \sum_{\tau=0}^{t}[x_i(t)K(\tau)x_j(t-\tau) - x_i(t-\tau)K(\tau)x_j(t)]\right) \tag{2}$$

Where $x_i(t)$ is a binary variable taking the value 1 if neuron i fired at time t. So, at each time t, $\Delta^{STDP}(t)$ is proportional to the change in W with respect to STDP. Our implementation assumed $K(t) \approx 0$ for $t > 4$ms.

However, the key innovation of this paper was to introduce a secondary source of competition. If the total strength of all synapses into or out of a given neuron exceed a soft limit W_{max} after STDP learning, then all such synapses experience long-term depression (LTD) proportional to the amount by which the soft limit is exceeded:

$$\theta_i^{col} = \left[\sum_{j=1}^{N}(W_{ij} + \eta\Delta_{ij}^{STDP}) - W_{max}\right]^+ \tag{3}$$

θ^{col} denotes the amount by which the soft limit has been exceeded. Define θ^{row} similarly. Then,

$$\frac{\mathrm{d}W_{ij}}{\mathrm{d}t} = \eta\Delta_{ij}^{STDP} - \epsilon(\theta_i^{col} + \theta_j^{row}) \tag{4}$$

This updates weights according to STDP along with a penalty if the soft limit is exceeded. Finally, to ensure that STDP never diverges, we implement a hard limit. If any element $W_{ij}(t) > w_{max}$, where w_{max} is a constant parameter, we manually set $W_{ij}(t)$ to w_{max}.

We actually implemented the results incorrectly based on a typo in [1]. Our implementation used the following equations:

$$\theta_i^{col} = \left[\sum_{j=1}^{N} (W_{ij} + \Delta_{ij}^{STDP}) - W_{max} \right]^+ \tag{5}$$

$$\frac{\mathrm{d}W_{ij}}{\mathrm{d}t} = \eta \Delta_{ij}^{STDP} - \eta\epsilon(\theta_i^{col} + \theta_j^{row}) \tag{6}$$

As demonstrated in sections 3 and 4, this causes some significant issues with convergence. Unfortunately, we noticed the problem too late into the project to redo the simulations.

2.2.2 Hebbian Learning

Hebbian plasticity is a general family of learning rules which strengthen synapses between neurons which fire with high correlation. Hebbian learning rules are generally considered incomplete because while they explain how synapses are strengthened, they do not explain how synapses might be weakened.

One way to complete the Hebbian learning rule is with STDP. In STDP learning, highly correlated neurons experience a large change in synaptic strength, but the direction of the change (increase or decrease) depends on the relative timing of firing between the synapses.

However, the soft and hard limits imposed by the model also provide a way for synapse strength to decrease, so it should be possible to implement this model using Hebbian plasticity. To do this, we ran the exact same learning algorithm as implemented for STDP, but we modified the kernel to be

$$K(t) = \begin{cases} e^{-t/\tau_{Heb}} & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases}$$

With this new kernel,

$$\Delta_{ij}^{Heb}(t) = \left(\frac{W_{ij}(t-1)}{w_{max}} + 0.001 \right) * \left(x_i(t)K(0)x_j(t) + \sum_{\tau=0}^{t} x_i(t)K(\tau)x_j(t-\tau) \right) \tag{7}$$

gives us a Hebbian learning rule. We implemented the soft and hard synaptic weight limits as in STDP learning.

2.3 Parameter Choices

Our implementation of the IB neuron network with STDP learning matched that used in [1] in many cases. However a few parameters were chosen differently. The table below provides a list of parameters we used:

Parameters	Values
dt	2×10^{-5}s
V^E	0V
V_θ	-0.05V
τ	0.004s
N	50
η	varies
A_a	9

Parameters	Values
τ_V	0.01 F/m^2
V^I	-0.07V
V_{reset}	-0.055V
r_{in}^{start}	10000Hz
w_{max}	0.14
ϵ	varies
$\tau_{STDP} = \tau_{Heb}$	0.02s

Parameters	Values
V^L	-0.06V
W_0	5S/m^2
T_{burst}	0.006s
r_{in}^{min}	4000Hz, 6000Hz
W_{max}	0.14
A_g	4
τ_{ada}	0.015s

3 Results

We were primarily interested in two major results produced in [1]. First, we were interested the occurrence of long synfire chains with synchronous regular firing. We also investigated the paper's claim that STDP learning with $w_{max} = W_{max}$ converged to scaled permutation matrices.

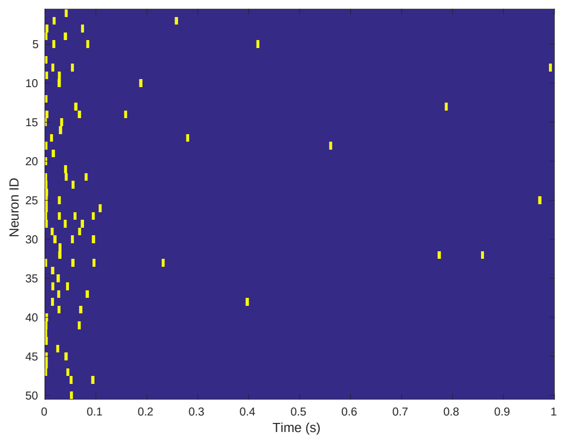
Though we tried to replicate the algorithm described in [1], we were unable to produce a stable system with the provided inputs, so we had to tune the parameters a bit. We eventually found parameters at which burst rates were steady and believable.

With the parameters we found, we were able to observe weaker versions of most of the observations in the paper. We found that weight matrices converged near permutation matrices and we were able to observe some short sequences of firing chains, but we did not see the synchronous regular behavior exhibited in the paper.

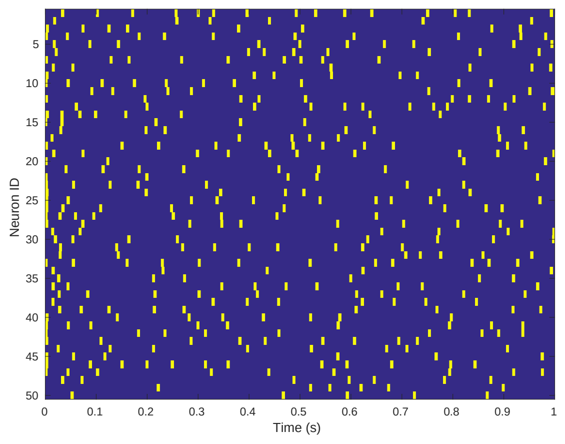
Finally, we replaced STDP learning with Hebbian learning and demonstrated that the weights of a system undergoing Hebbian learning with soft and hard limits converged as closely or better to permutation matrices than they did under STDP learning.

3.1 Parameter Tuning

Although our model was constructed to be identical to the model used in [1], we were unable to use the same parameters as the paper. As an example, the paper claimed to set r_{in}, external input, at 4 Hz. At 4 Hz, we were unable to get our system to fire at all! We assumed it was a typo (the rest of the parameters were given in terms of ms), and attempted running our simulation at 4000Hz. Even 4000Hz was not enough to get the system started, so we tried simulated annealing. The system began with 10000Hz of stimulation, which decreased steadily to 4000Hz. We did this again, but this time stopped at 6000Hz. The difference was striking (see figure 1).



(a) 4000 Hz



(b) 6000 Hz

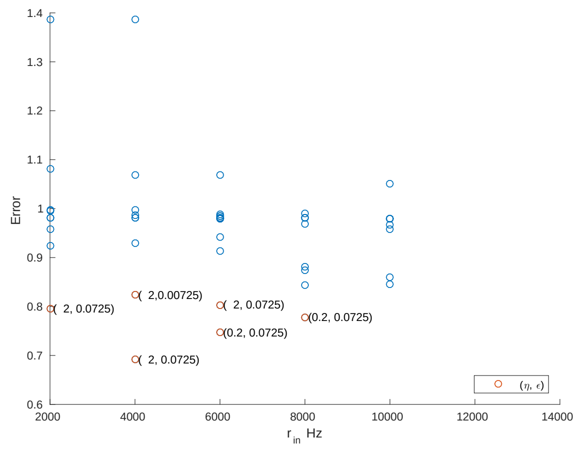
Figure 1: In figure 4 of [1], the authors displayed a burst plot in which each neuron burst every 75 ms with high synchrony and regularity. We did not observe such synchrony in either plot, but the average frequency of firing in (b) was approximately 75 ms. Notice both systems exhibited exaggerated firing in the beginning. This was a consequence of simulated annealing.

Beside r_{in}, we spent a lot of time finding appropriate values for η and ϵ.

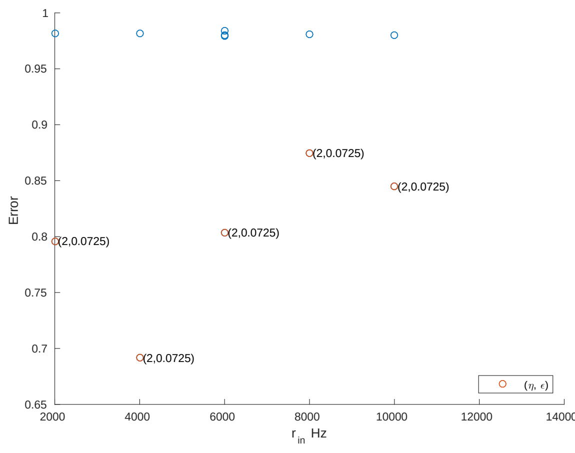
η is the learning step-size. If η is too small, the weight matrix will not converge in a reasonable amount of time. If it is too large, whenever the weight matrix overshoots the soft limit, it will do so by a large margin, which will prompt large corrective measures, so convergence to a permutation matrix (which is barely under the soft limit), will not be plausible.

ϵ is the strength of the soft limit. If ϵ is too small, then when the weight matrix passes the soft limit it will not experience very strong hLTD. This will have the effect of weakening heterosynaptic competition. If ϵ is too large, then each time the weight matrix passes the soft limit, it will experience very strong hLTD. Because the permutation matrix is on the border of the soft limit, having a large ϵ will make any permutation matrix an unstable critical point in our simulation.

To evaluate these, we ran the simulation on a large combination of parameter values for r_{in}, η and ϵ (see figure 2a). For each of these, we tested convergence using an error function (see section 3.2). Then we isolated the product $\eta\epsilon$ which seemed to yield the best results (see figure 2b).



(a) Errors of all simulations run

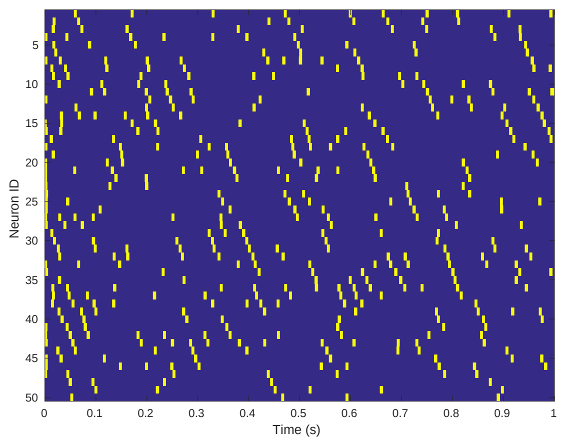


(b) Errors of all simulations with $\eta\epsilon = 0.145$

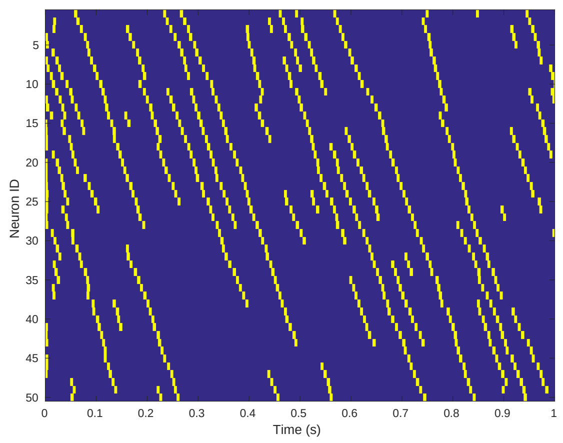
Figure 2: Recall from (6) that in our implementation, η and $\eta\epsilon$ are important tuning parameters for the rate of convergence. In all cases, we ran the algorithm for 50000 time steps with $dt = 2 \times 10^{-5}$s. Notice that all error values are very high. (a) This is a plot of every simulation we ran. The best results are labeled. (b) This is a plot of the subset of points from (a) where $\eta\epsilon = 0.145$.

The best two combinations of parameters seemed to be (r_{in}, η, ϵ) =(4000Hz, 2, 0.0725) and (6000Hz, 0.2, 0.0725). For the rest of the paper, we used these parameters unless otherwise noted.

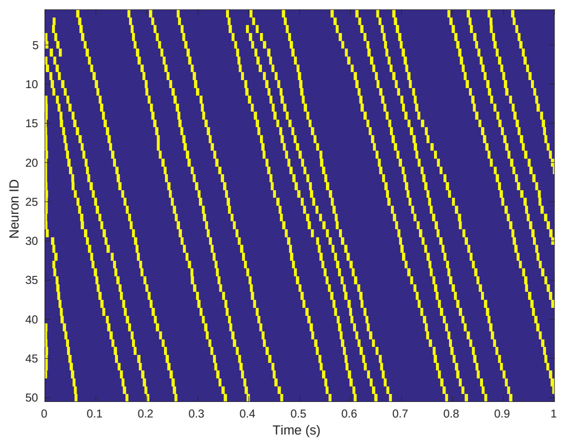
Finally, we tried running a playback sequence with a fixed permutation weight matrix. By symmetry, according to the model if one permutation matrix is a stable point, then all of them must be. Therefore we fixed W to be a permutation matrix. For convenience, we assumed this permutation matrix was a large cycle. We ran it without modification to observe burst patterns for differing values of w_{max} (see figure 3).



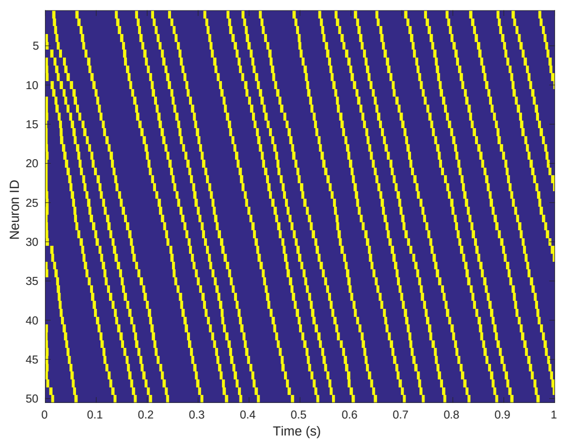
(a) Bursts without learning, $w_{max} = 0.14$.



(b) Bursts without learning, $w_{max} = 0.3$.



(c) Bursts without learning, $w_{max} = 0.5$.



(d) Bursts without learning, $w_{max} = 0.7$.

Figure 3: We ran the simulation for 50000 time steps at $dt = 2 \times 10^{-5}$s with $r_{in} = 6000$Hz. As expected, for high values of w_{max}, we witnessed high regularity, synchrony and long firing chains. At our default value of $w_{max} = 0.14$ we witnessed a few small firing chains, but much more irregular firing.

At our default w_{max} value, the IB model was unable to maintain the patterns of long sequential firing observed in [1], however larger w_{max} values did display these patterns.

3.2 Convergence and Stability

Our IB model with STDP learning very quickly developed a stable bursting rate. Given up to about 4000Hz of external stimulation, this bursting rate was essentially 0, but for more stimulation the bursting rate was consistent. The neurons in the network constructed in figure 4 of [1], on which our paper was based, bursted every 75 ms or so. From figure 4, we can see that this corresponds to an external stimulation of slightly less than 8000Hz in our model.

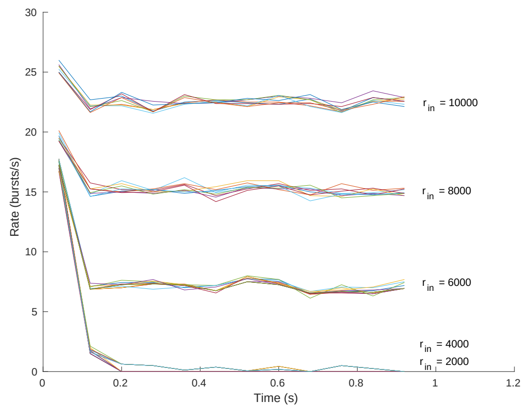
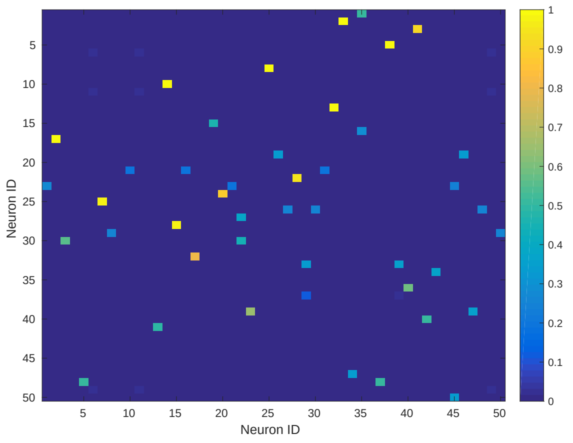
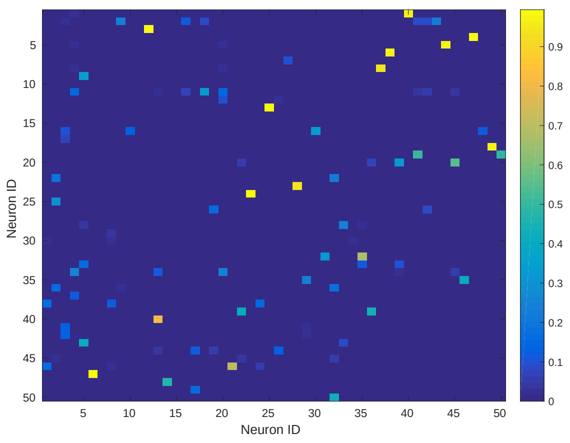


Figure 4: The plot above measures the average firing rate of a neuron in the recurrent network over time over approximately 50 simulations. Notice that the average firing rate converges cleanly to different values dependent only on r_{in}.

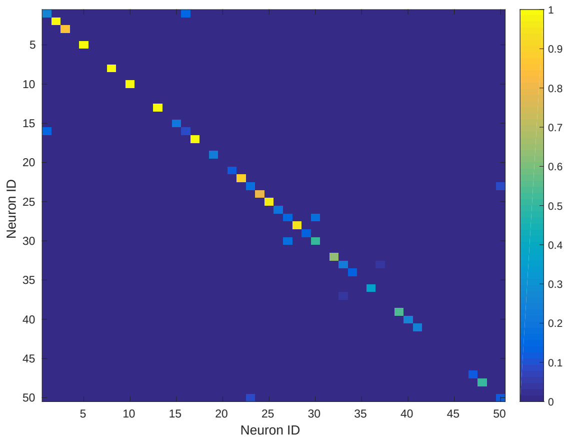
From this we concluded that our base IB model was stable. The next thing we needed to test for was convergence of our weight matrix under STDP with soft and hard limits to a permutation matrix. Recall that all permutation matrices P satisfy $PP^T = I$. So, while we can visually inspect heat maps of W to see if it's close to a permutation matrix, it is easier to look at how WW^T compares to $w_{max}^2 I$ (see figure 5).



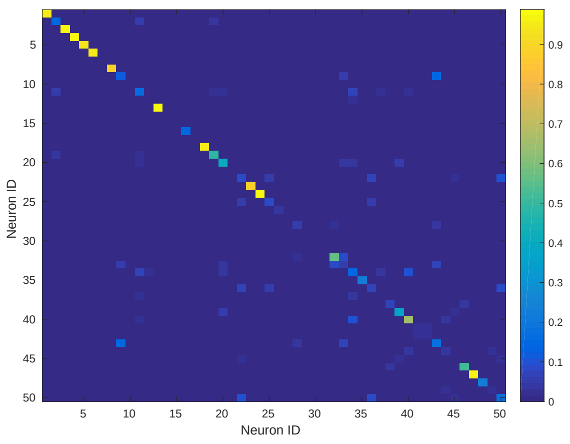
(a) Weight matrix of 4000Hz external input with annealing



(b) Weight matrix of 6000Hz external input with annealing



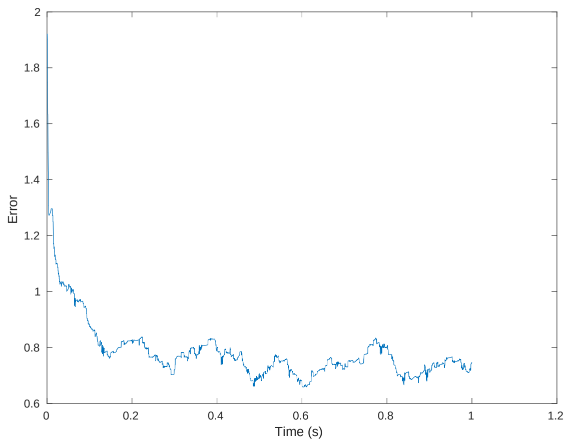
(c) WW^T of 4000Hz with annealing



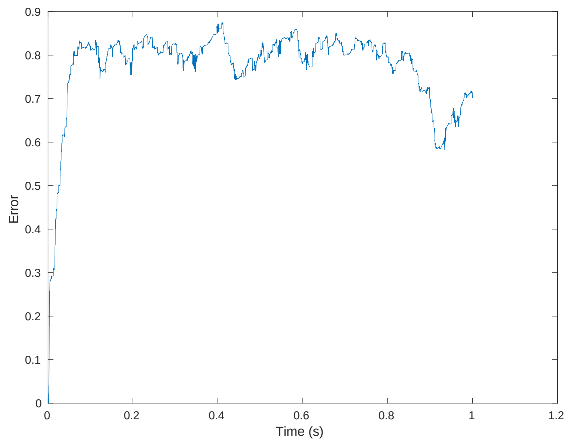
(d) WW^T of 6000Hz with annealing

Figure 5: After running each simulation for 50000 time steps with $dt = 2 \times 10^{-5}$s we plotted Here I am

Plot error function over time from normal and from permutation matrix



(a) Plot of error in weights over time starting from a full connection weight matrix.



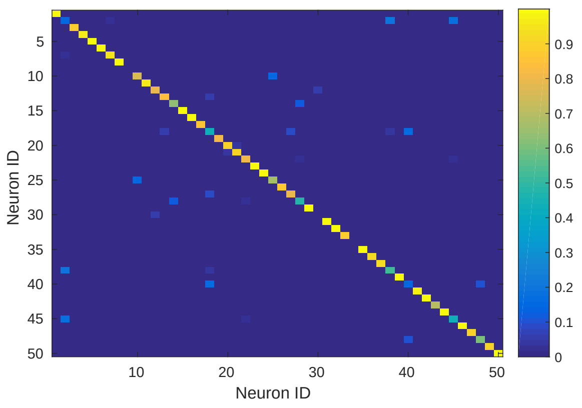
(b) Plot of error in weights over time starting from a permutation weight matrix.

Figure 6: Compare the two

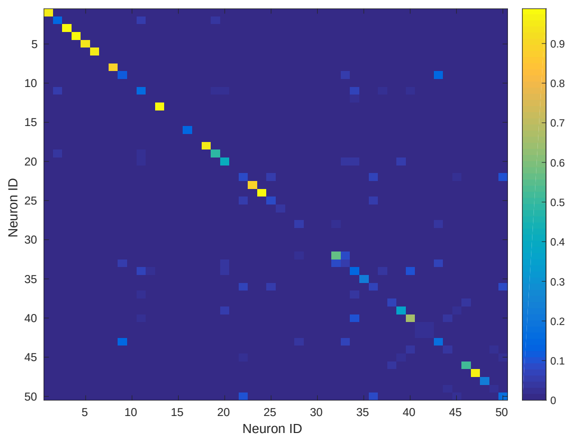
Describe why the error function converges away from (mistake, see discussion).

3.3 Hebbian Learning versus STDP

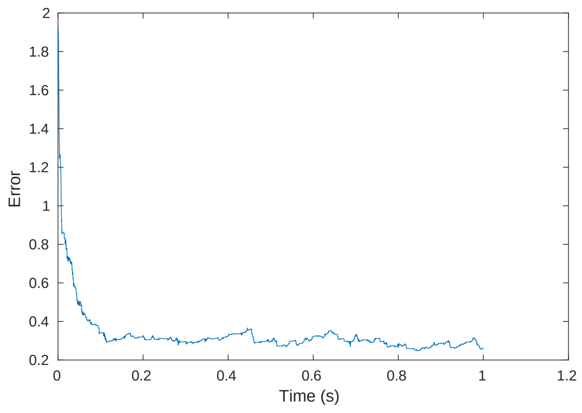
- Introduce the idea of the refutation.
- Give a theoretical description why the type of learning should be relatively unimportant.
- Compare plots (WW^T, Error vs Time, Burst History).



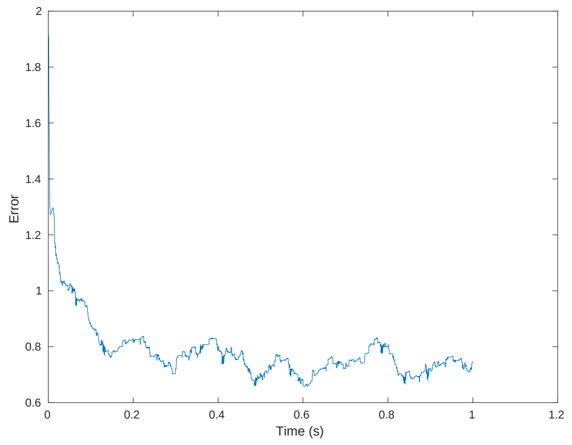
(a) Multiplied weight matrix with Hebbian learning



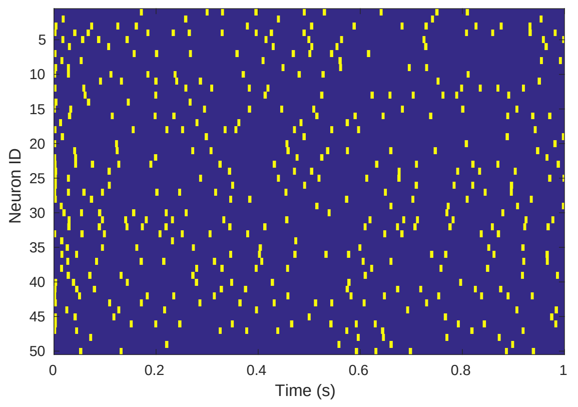
(b) Multiplied weight matrix with STDP learning



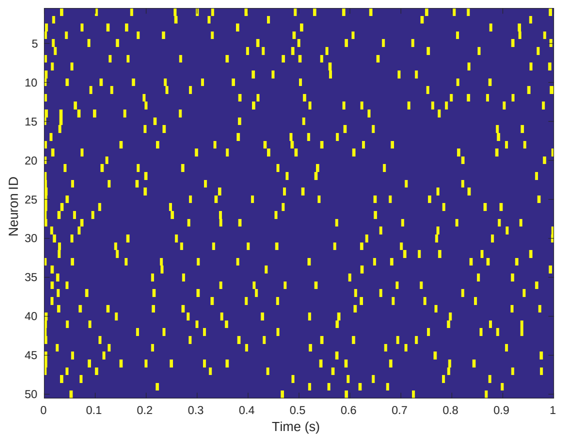
(c) Error over time of Hebbian plot



(d) Error over time of STDP plot



(e) Burst history of Hebbian plot



(f) Burst history of STDP plot

Figure 7: Compare STDP to Hebbian

4 Discussion

Further improvements that could be made to our model and where this research could be taken.

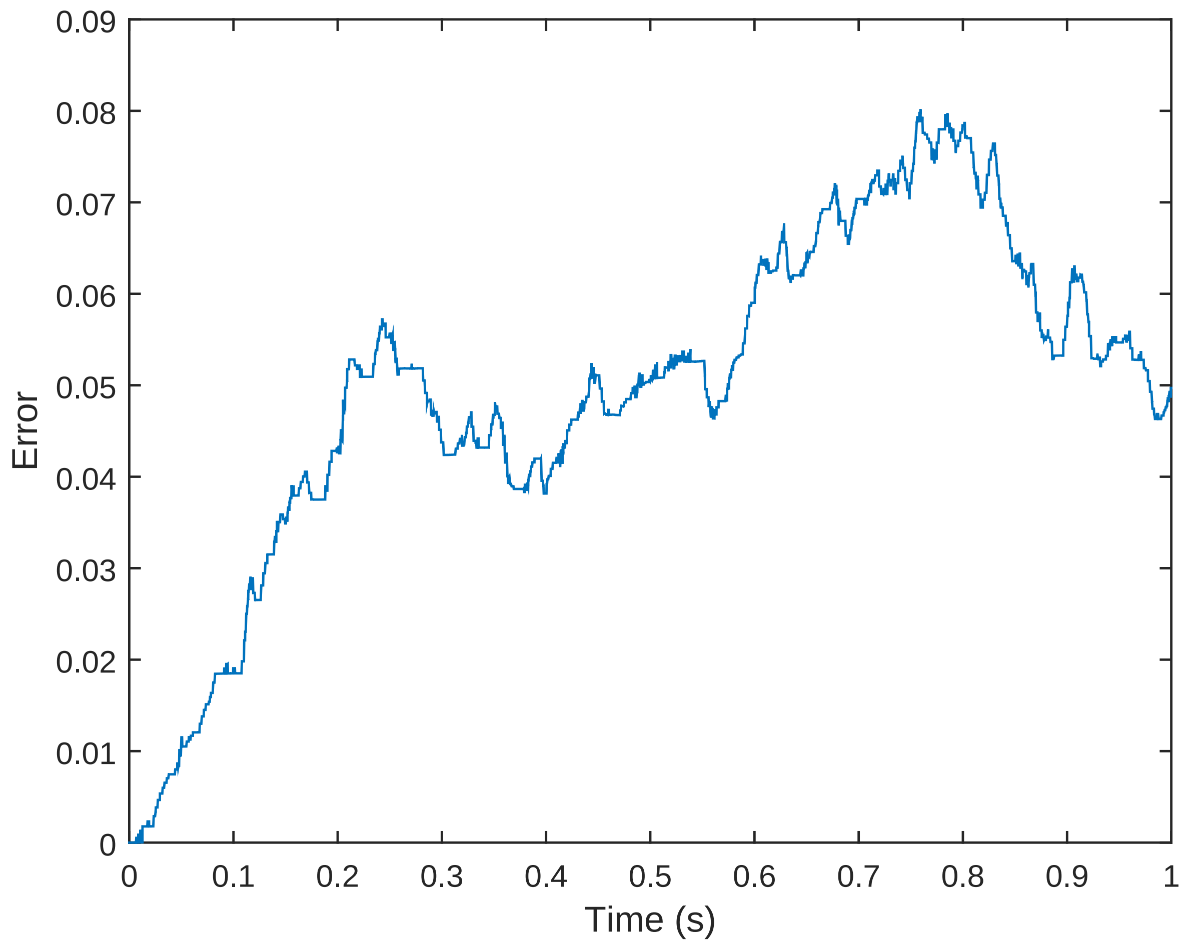


Figure 8: Here is a plot of error over time which was computed using the corrected algorithm presented in the methods section.

5 Summary

Quick summary of our results and everything.

References

[1] Ila R. Fiete, Walter Senn, Claude Z.H. Wang, and Richard H.R. Hahnloser. Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. Neuron, 65(4):563 – 576, 2010.