Nikolay N. Valov
11/27/2018
CN510
Arash Yazdanbakhsh

Assignment 5: Recurrent Competitive Fields

Introduction

This assignment looks at recurrent competitive fields based on the shunting equation from previous assignments. A recurrent neuron is defined as containing a feedback pathway that allows a cell's output to project back to its input either directly or indirectly. The competitive aspect of this network comes from the fact that there are inhibitory connections between cells in the network. For this specific network, we assume an on-center, off-surround inhibitory interactions. As stated previously, the field itself implies a conglomerate of interconnected neurons. Furthermore, the input given to these neurons will be called signal function. These functions are the way that an input current is given which creates a pattern at input offset. Once the input is turned off, the pattern is stored differently based on the signal function. The computational importance of recurrent competitive fields is their application to persistent short term memory because of the recurrence property. The signal functions explored in this assignment are: linear, slower-than-linear, faster-than-linear, and sigmoid. All of these create the same pattern at input offset, but store the information differently.

The skills that will be gained by completing this assignment primarily center on coding implementation within Matlab. The equation will not be solved at steady state due to the need to explore the time dynamics of the pattern storage. Therefore, I will need to produce a current over time within Matlab. In addition, the application of the signal function will most likely prove to be a considerable challenge as the syntax for the equation within the summation function will need to be exactly right in order to store the actual signal. Finally, being able to utilize the mesh function to explore the temporal and spatial dynamics will beneficial to analysis of the neural network.

Methods

Firstly, all equation modeling was done in matlab. The equation that models the recurrent competitive fields consists of ten STM cells. The primary equation used is based off of our previous shunting equation:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)[f(x_i) + I_i] - x_i \sum_{k \neq i} f(x_k) \qquad (1)$$

Within this equation, we have our decay term $-Ax_i$. Our inputs to the network are represented by I_i. Our A value is equal to 1 and the B value is equal to 3 for this assignment. The signal functions are represented by $f(x_i)$, specifically $f()$, and then the neuron activity (x_i) is run through the function. The summation function represents the off-surround inhibition from other neurons which will be scaled to activity.

The signal functions used for this experiment are:

$$(\boldsymbol{i}) \quad f(w) = w$$
$$(\boldsymbol{ii}) \quad f(w) = w^2$$
$$(\boldsymbol{iii}) \; f(w) = w/(F + w)$$
$$(\boldsymbol{iv}) \; f(w) = w^2/(F + w^2)$$

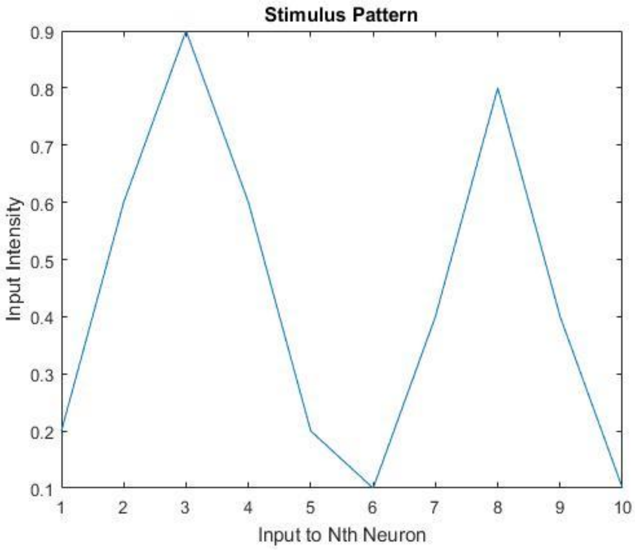
Equation i represents a linear signal function which means that it simply holds the values of x_i. Equation ii represents a faster than linear signal function which will store the squared value of x_i. Equation iii is the slower-than-linear signal function with an F-value of 0.25. Equation iv is the sigmoid signal function with the same F-value as the previous equation.

Two input patterns shown below were given to the function. The input was given for 1000 milliseconds. The equation modeling itself was run using a for-loop with a time step $dt = 0.001$ from 0 to 10 while giving input for only the first 1000ms. The euler method was used to construct the equation in order to isolate dx. The mesh function was used in order to display how the signal functions affected the pattern stored and analyze the spatial and temporal dynamics of the system. The normalized values were plotted utilizing a-for loop combined with the mesh function as well.

$$Input\ 1 = \{\, 0.2, 0.6, 0.9, 0.6, 0.2, 0.1, 0.4, 0.8, 0.4, 0.1 \,\}$$

$$Input\ 2 = \{\, 0.7, 0.6, 0.8, 0.9, 0.5, 0.3, 0.5, 0.7, 0.8, 0.4 \,\}$$

Results



Figures 1-3 are the results of utilizing input 1 from part a and plugging it into each of the signal functions to see how results are stored.

.

Figure 1) **Stimulus Pattern Storage.** This graph shows the input intensity to each neuron within the network. In this case, a ten neuron network is being utilized

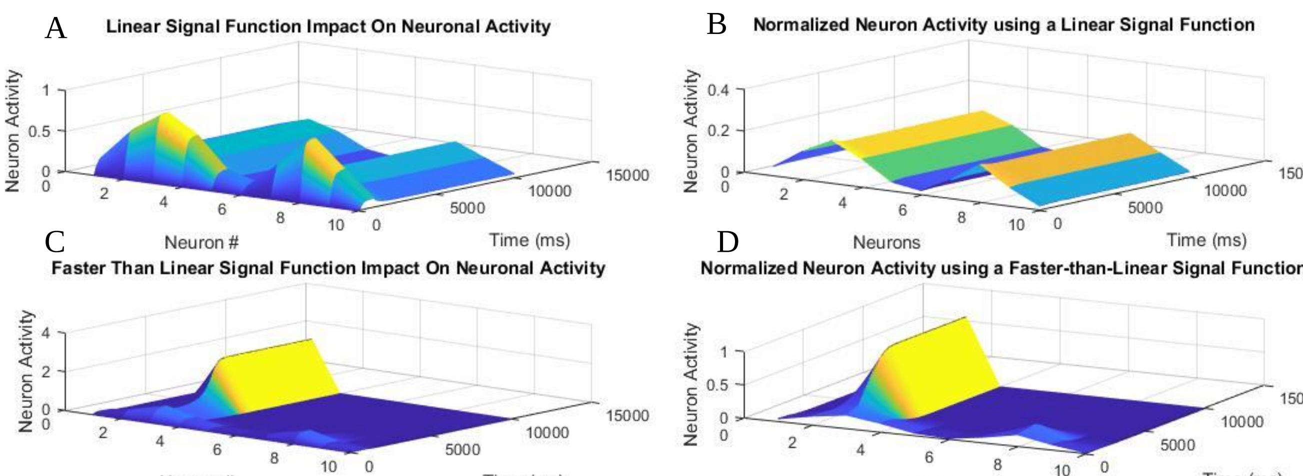


Figure 2) **Linear and Faster than Linear Signal Function Storage. (**A) Depicts Input 1 utilizing function (i), linear signal function. Shows pattern stored as the same pattern given. (B) Normalization of the neuron activity using the linear signal function shows that neuronal activity is stored as the input given after Input goes to 0. (C) Depicts the Faster Than Linear Signal Function, function (ii) and its winner-take-all storage pattern. (D) Normalization of neuron activity using a faster than linear signal function.

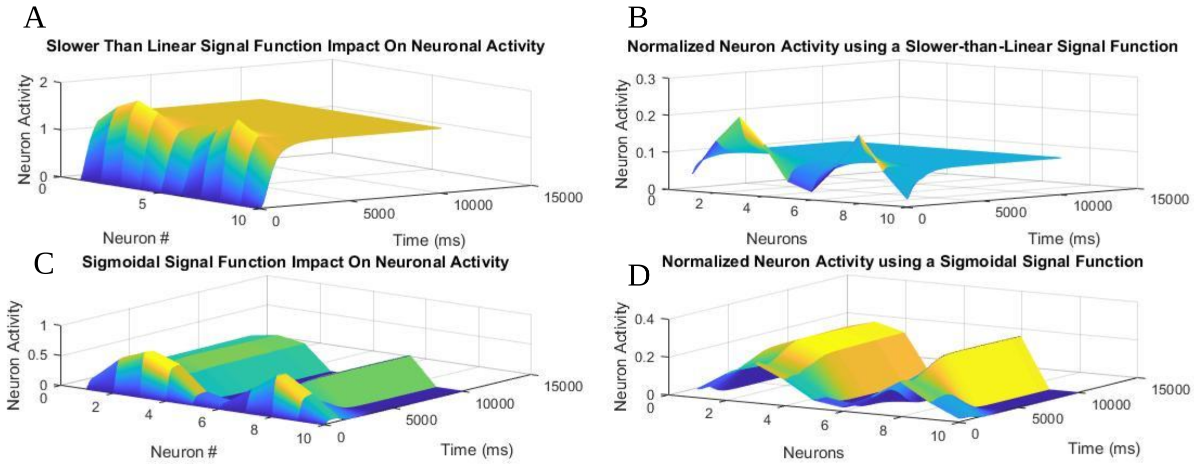


Figure 3) **Slower Than Linear Signal Function and Sigmoidal Signal Function impact on neuronal activity.** (A) Utilizing Input 1, the slower than linear signal function impact on neuronal activity is shown. The activity is stored as a uniform pattern after stimulus completion. (B) The normalized neuronal activity of the previous function. Again, showing a uniform storage patter post stimulus completion at 1000ms. (C) Sigmoidal Signal Function Impact on Neuronal Activity shows a persistent stimulus stored, above the quenching threshold. (D) Normalized values of the Sigmoidal Function.

Figure 4-6 are utilizing Input 2 from part b, but with the same parameters as discussed in the Methods.

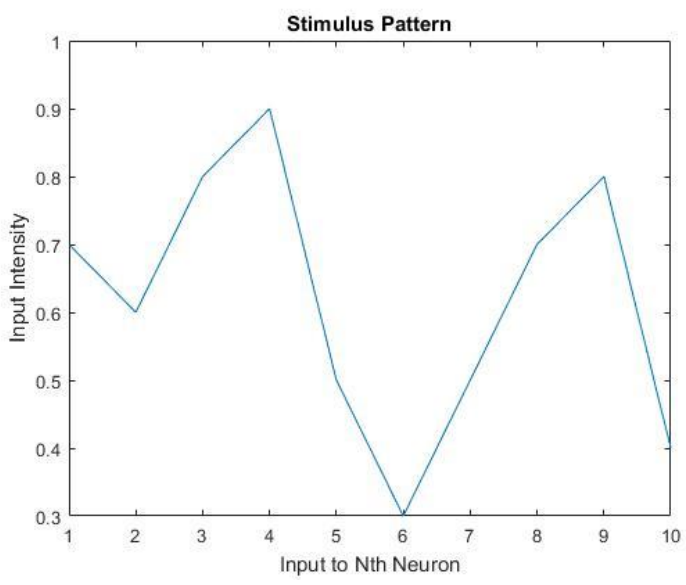


Figure 4. **Stimulus Pattern Representation.** Stimulus pattern of the second input is shown. Each neuron has its own input intensity.

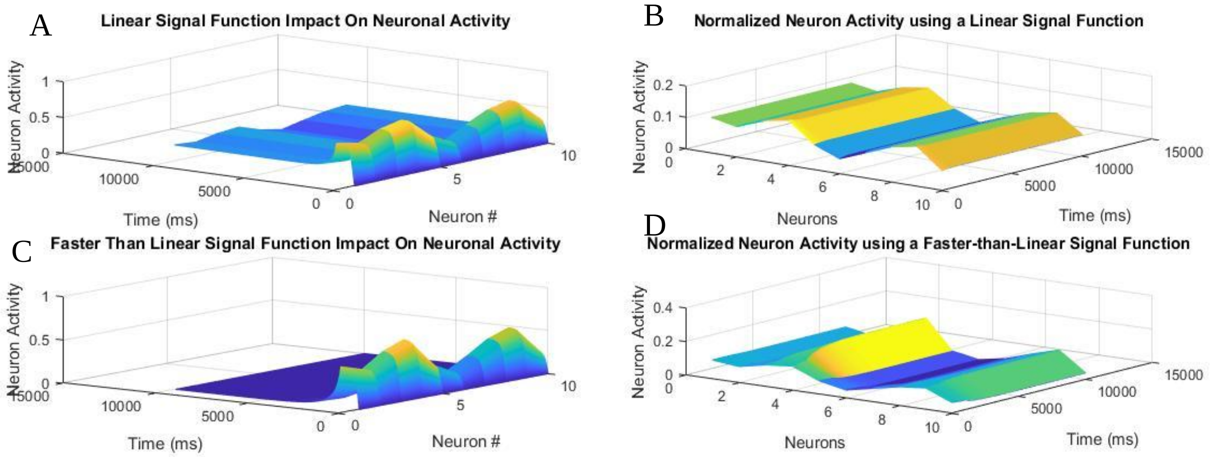


Figure 5) **Linear and Faster than Linear Signal Function Storage using Input 2.** (A)shows the linear signal function impact on neuronal activity. Due to how much stimulus is on neurons 1-3 shown in Figure 4, there is more activity on those neurons. (B) Normalized function which shows the persistent storage pattern of the linear function. (C) Unexpected result with figure representation of data. This system should have had winner-take-all storage pattern with same equation used. (D) Normalized neuron activity shows the proper Faster Than Linear storage pattern which was not shown in 5C. Shows a winner take all system.

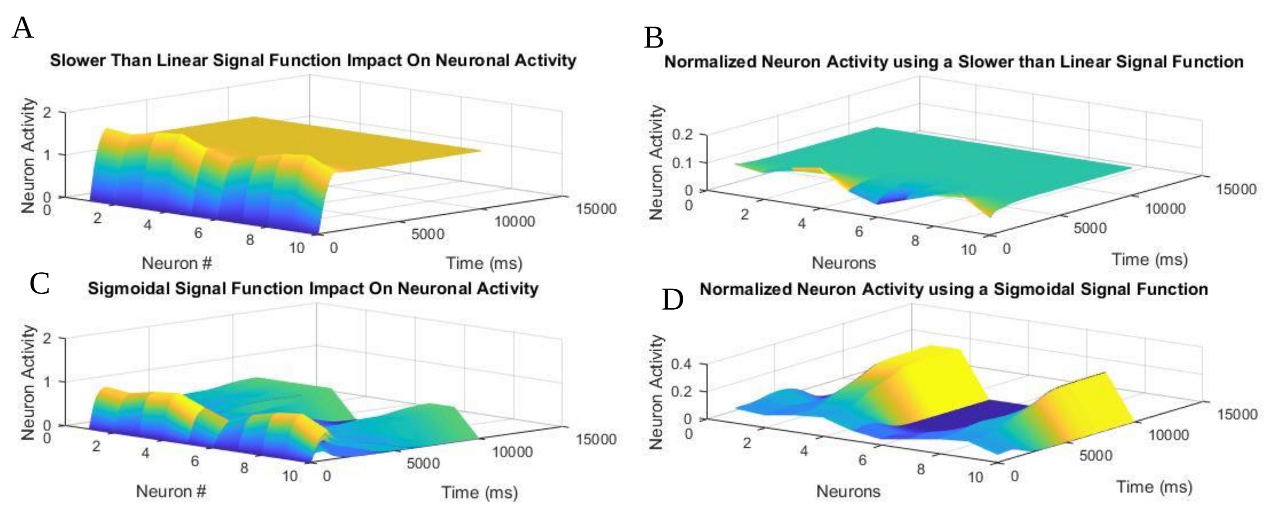


Figure 6) **Slower than Linear and Sigmoidal Signal Functions impact on neuronal activity.** (A) Shows that neuronal activity gets stored uniformly after stimulus end. (B) Normalization of neuronal activity using the slow than linear signal function. (C) Sigmoidal Signal function stores input as expected as it is above the quenching threshold. (D) Normalized activity for Sigmoidal Signal function. Shows that neuronal activity is held constant after stimulus storage.

Discussion

Firstly, the general signal function observations will be explored and analyzed based on expected results. The linear signal function was expected to store the pattern at input offset shown by Figure 1 and Figure 4 with each inputs given. Once the neurons were allowed to reach equilibrium, the stored pattern is equal to the pattern at input offset, but at a lower activity. (Figure 2A, Figure 5A) Once the linear signal function values were normalized, the true expected storage was seen where the exact input pattern is stored post-stimulus. Next, the Faster-Than-Linear signal function is expected to have a winner-take-all storage pattern after stimulus offset. (Figure 2C, Figure 5C) This is because the initial values for x_i were large enough to converge to a positive value. (Figure 2C, Figure 5C) This result is due to $x_i < x_k$ where larger x_i are increasing and smaller x_i are decreasing, leading to the competition. One issue is that storage in Figure 5C was not the expected pattern. However, Figure 5D shows the x_i values normalized, resulting in the expected winner-take-all storage pattern.

The slower-than-linear and sigmoid signal functions were explored in Figure 3 and Figure 5. In order to explain our results, the signal function (iii) technically has an additional C value that is represented as: $\frac{Cx_i}{F+x_i}$. In our case, C was assumed to be 1, hence why it was not added within the methods. The slower-than-linear signal function depends on the parameters A, B, C, F. If $\frac{C}{D} \leq \frac{A}{B}$ then all x_i will go to 0, resulting on no storage. However, what we see in Figure 3A and Figure 6A is a uniform stored pattern following stimulus offset. This is because based on our parameters where A = 1, B = 3, C =1, and F = 0.25, $\frac{C}{D} \geq \frac{A}{B}$ which means that the total energy will converge to a positive value, regardless of initial input. (Figure 3A,B & Figure 6A,B) The normalized activity versus the index of neurons shows a uniform stored pattern as well. (Figure 3B, Figure 6B) The Sigmoid Function combines properties of the previous signal functions: contrast enhances small signals, stores intermediate signals with little distortion, uniform very large signals. It also introduces Quenching Threshold which depends on the form of the sigmoid, so the parameters and the initial distribution of inputs. The quenching threshold is essentially the minimum initial activity level to avoid going to 0. In the case of inputs 1 and 2, Figure 3C and Figure 6C show that as long as the neuronal activity reaches the linear range of f(x), they will not be quenched. Therefore, smaller inputs went to 0, but the higher inputs were persistently stored. (Figure 3C & Figure 6C)

Contrasting input 1 and input 2 shows the dynamics of having different input values. In the linear signal function everything that is stored post stimulus offset is equal to the input pattern. The main difference is that neuron 1 and 2 in input 2 are more excited, resulting in the storage of that pattern with only lower values. (Figure 5A) However, in input 1, the value for the neuron 1 is = 0.2 which is stored as a lower overall input, but it still keeps the same pattern. Ultimately, there is a wider spatial distribution of activity with input 2 because of the higher input values. When contrasting the faster-than-linear signal function storage, the main difference is that there is a clear winner-take-all storage in input a. (Figure 2C) However, the input 2 pattern is slightly different, having significantly higher input values within its pattern. Therefore, there is a clear winner-take-all storage after stimulus offset, but due to the similarity of the peaks, some storage of neuron 8 and 9 also occurs. (Figure 5C) The slower-than-linear signal function has a similar result in both input 1 and 2. This makes sense as there will ultimately be uniform storage after stimulus offset. (Figure 3A and Figure 6A) Furthermore, the main difference between the two is the initial spatial distribution of inputs that is caused by the different input patterns. Finally, the sigmoidal function results show the impact of the quenching threshold. Input 1 stores the sigmoidal function with a clear differentiation between neurons 1-4 and then neurons 6-9. This is because neuron 5 and neuron 6 have an input of 0.2 and 0.1 respectively which as shown by Figure 3C do not actually pass the quenching threshold. Consequently, their input data is not stored. Contrasted with the second input where neurons 1-4 and neurons 6-8 get stored because those values were higher than the quenching threshold. This also increased the spatial distribution of storage pattern because more inputs were higher than the threshold. (Figure 6C)

Summary

We can conclude that varying the initial values of x_i will have a significant impact on the spatial distribution of the stored pattern of the sigmoidal, faster-than-linear, and linear signal functions. However, the storage pattern of the less-than-linear signal function will always result in a uniform storage output post stimulus offset regardless of the input pattern given. Furthermore, increasing the general input values within a system will result in wider spatial storage of inputs in the sigmoidal function because less neurons will have an input less than the quenching threshold.