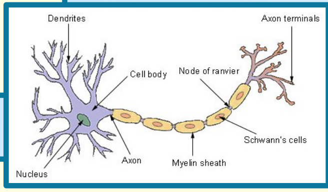
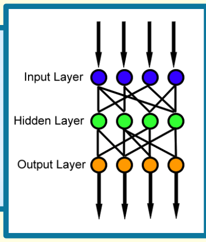
Bronzen Williams
Christopher Jimenez
Elias Corona
Lillian Reina
Mohamed Idris

Brain Dynamics: Math at the Inner Frontier



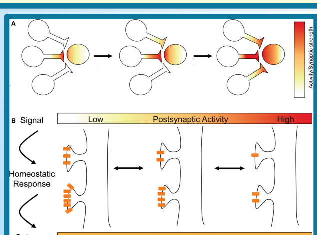
Abstract

Neurons make up a network that transmits information throughout the body. The transfer of information from one neuron to another is called a synapse. There are a couple of theories as to how the neurons fire. In this study, we have experimented with these different rules using MATLAB to write a program that simulates neurons firing to see which of these rules better improves the number of neurons that are on in the last layer. From an analysis of our heat graphs, we found that the Hebbian Rule results in more neurons on in the last layer than the Homeostatic Rule. The last thing that we wanted to test was what would happen if we combined the Hebbian and the Homeostatic Rules. We found that when we applied both to our model, the number of neurons on in the last layer was the largest and generated the smallest standard deviation. Applying both rules was the best way of optimizing of the number of neurons on in the final layer and producing the smallest standard deviation.



Introduction

- Donald Hebb discovered a theory of "cells that fire together wire together."
  - When pre-synaptic cell and post-synaptic cell fire at same time, overall synaptic strength tends to be stronger.
  - When post-synaptic cell fires before pre-synaptic cell, strength tends to be weaker.
- Forms of homeostatic plasticity are ways to stabilize properties of Hebbian plasticity.
  - Regulate neural excitement, stabilize synaptic strength, and influence rate of synapses.
- We attempted to create a program that represented a model of a neural network .
  - We first implemented Hebbian plasticity into our program, then we gathered and plotted the average grades and the standard deviations in two heat graphs based on assigned weights "a" and "b".
  - Then we implemented homeostatic plasticity also gathering and plotting our results in heat graphs.
  - Finally we implemented both rules into our program gathering and plotting our results in heat graphs.



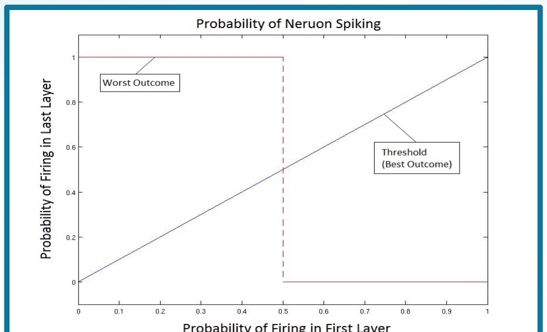
Methods and Prodedures

Using MatLab, a computer programming software, we modeled a five layer binary Feed Forward Neural Network (FFN).

By coding weights and probabilities, we were able to model how a FFN may transmit information. Weights were either one or zero, and they represented if a neuron connection fired into the next layer. The probabilities were a random string of numbers between zero and one that determined whether a weight for a particular connection was one or zero (using a threshold for the probabilities).

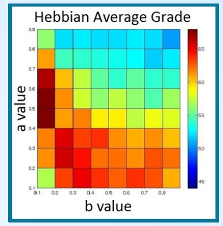
Learning rules then had to be implemented into the model. These learning rules affected the probabilities that then determined the weights, which then determined firing rates. In order to develop the learning rules fully, a method to weaken and strengthen firing rates had to be created. We achieved this by using two values, "a" and "b" between zero and one. We used "a" to weaken the probability of a weight being one, while using "b" to strengthen the probability of a weight being one. For an example of "a", if a neuron in the pre-synaptic layer was zero, but a neuron in the post-synaptic layer was one, then the probability (P) of a synapse was changed using the formula P = "a"*P effectively cutting the probability and updating it to "P*"a"". For "b", if a neuron in the pre-synaptic layer was one, and a neuron in post-synaptic layer was also one, then the probability was increased by ""b"(1-P)", making P = P+"b"(1-P). By using both the "a" and "b" values we were able to test the network model using three combinations of learning rules: Hebbian, Homeostatic, and a hybrid rule consisting of both Hebbian and Homeostatic. The script used, determined the weights based on the learning rules, tested all values of "a" and "b" between zero and one in increments of one tenth, and output heat maps that displayed average graded values, and average standard deviations.

We then observed the graded heat map matrix along-side a heat map matrix that displayed the standard deviation of grades for each combination of "a" and "b", and from this we made observations of our final results.

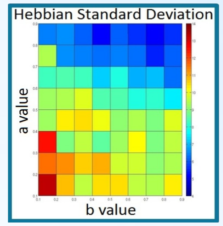


Acknowledgements

Department of Applied Mathematics
Team Members: Bronzen Williams, Christopher Jimenez, Elias Corona, Lillian Reina, and Mohamed Idris
Project Facilitators: Assistant Professor Eric Shea-Brown, Alex Cayco-Gajic, Megan Lacy, and Guillaume Lajoie
Chair of AMATH: Nathan Kutz



Above: Heat map of the average grades using the Hebbian rule. Below: Heat map of the average standard deviations using the Hebbian rule.

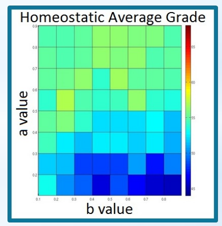


Results

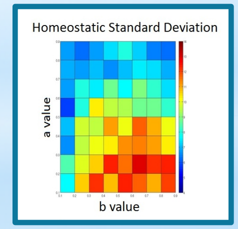
Hebbian plasticity
- The average grades tended to be highest when "a" was less than 0.6 for all "b" values.
- The standard deviations were higher when "a" was 0.5 or less for all "b" values and lower when "a" was greater than 0.6 for all "b" values.

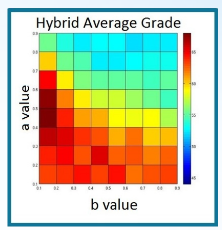
Homeostatic plasticity
- The average grades tended to be the same as the average grades using the Hebbian plasticity rule when "a" was greater than 0.6 for all "b" values.
- The average grades were lowest when "a" was less than 0.6 for all "b" values.
- The standard deviations were highest and higher than the Hebbian rule when "a" was 0.5 or less for all "b" values and lowest when "a" was greater than 0.5 for all "b" values.

Hybrid
- The average grades were highest and higher than the average grades of both the Hebbian and the Homeostatic rules' average grade when "a" was less than 0.5 for all "b" values.
- The average grades were lowest when "a" was greater than 0.5 for all "b" values.
- The standard deviations were highest and less than the average grades of both the Hebbian and the Homeostatic rules' standard deviations when "a" was less than 0.5 for all "b" values.
- The standard deviations were lowest when "a" was greater than 0.5 for all "b" values.

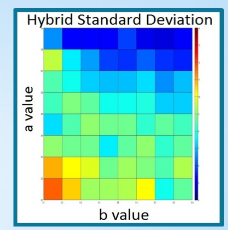


Left: Heat map of the average grades using Homeostatic plasticity. Right: Heat map of the average standard deviations using Homeostatic plasticity.





Left: Heat map of the average grade using both the Hebbian and the Homeostatic plasticity rules. Right: Heat map of the average standard deviation using both the Hebbian and the Homeostatic rules.



Conclusion

We noticed that the Hebbian plasticity and the Homeostatic plasticity both improved the Feed Forward Network in their own ways. The Hebbian rule got higher grades but had high standard deviations while the Homeostatic rule got lower grades but had low standard deviations, thus stabilizing the Feed Forward Network. Our biggest observation was that when combining both rules, our hybrid network got the highest grades out of any of the test while at the same time maintaining low standard deviations.