Adaptive Synaptogenesis Neural Networks: Creating Factorial Encodings Using Inhibitory Neurons

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Cooper Scher
Spring, 2022
Technical Project Team Members

Rosanne Vrugtman, Department of Computer Science

Adaptive Synaptogenesis Neural Networks: Creating Factorial Encodings Using Inhibitory Neurons

CS4991 Capstone Report, 2023

Cooper Scher

Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
cms4ub@virginia.edu

ABSTRACT
Previous adaptive synaptogenesis neural network models have failed to achieve proportionality between input patterns probabilities and output neural coding probabilities, which would optimize information storage, model size, and energy used. Using human brain learning as a guide–due to its ability to optimally store and process information using factorial encoding–an inhibitory interneuron was added to the adaptive synaptogenesis algorithm. An in-silico model of a simple unsupervised learning problem was created in MATLAB to test the ability of the modified algorithm to create distributed codes. After repeatedly simulating the network and testing for optimal parameter combinations, the inhibitory neuron was able to be integrated into the adaptive synaptogenesis algorithm and achieve factorial encoding in both simple and more complicated input worlds created for this simulation. However, the generality of this algorithm still needs to be tested with even harder input environments and datasets to ensure robustness.

1. INTRODUCTION
The goal of this research is to create a distributed code, also known as factorial encoding, which is a highly desired property in neural networks since it optimizes the minimum network size needed to represent an input. Factorial encoding is an encoding of input patterns from an environment where the probability of a given input pattern can be determined simply by the product of the probabilities of each of the encodings it belongs to. In other words, the goal is to try to find a set of conditionally independent encodings to proportionally allocate encodings to input patterns with respect to their probabilities.

This has a few important implications. First, a sufficiently decorrelated encoding will allow for more optimal representation of data. For segmentation problems where a sample space needs to be broken down into components, correlated inputs will be represented in fewer bits. Thus, a sample space is optimally broken down to independent features that combine to recreate inputs.

Second, distributed codes represent a simpler encoding of the sample space. A set of fully independent encodings will require fewer bits to encode a given space than ones where there is correlation between the encodings. For example, 2 fully independent bits can encode up to 4 possibilities while 2 fully correlated bits can only encode 2 possibilities since the value of one bit is dependent on the other.

2. RELATED WORKS

The current work was developed by modifying the model and algorithms in Thomas, et al. (2015). This model consists of three stages: associative modification, synaptogenesis, and shedding. During training, the model is given each of the inputs in a randomly-selected order and performs associative modification—the process of modifying the weights of synapses, or connections, between the input and output neurons based on how actively they are used. Synaptogenesis and shedding are the processes of randomly adding and removing inactive synapses, respectively. This model succeeded in developing a neural network that accurately classified input patterns into larger categories as part of a segmentation task; however, many individual neurons were not category-exclusive meaning that the model did not achieve a distributed encoding.

Factorial encoding as the concept introduced in this work is described in Schmidhuber (1992). This work first presented a method for achieving a distributed code for binary representations of inputs using predictability minimization. With a 3-layer back-propagation neural network and logistic activation functions, Schmidhuber was able to find factorial codes efficiently for simple, small-dimensioned datasets. However, this work has the drawback of being based on gradient descent methods which are subject to local minima if improperly parameterized.

3. PROJECT DESIGN

The neural network model in this work is a modified version of the model presented in Thomas et al. (2015), an unsupervised adaptive synaptogenesis neural network algorithm. In particular, three modifications were made to the previous algorithm and tested on the same datasets used in Thomas et al: 1) an inhibitory interneuron was added to the network; 2) associative modification was changed to occur after each new input is presented to the network for connected synapses and is now normalized; and 3) the synaptogenesis rule was changed to have a banded cutoff versus a single threshold. Additionally, both synaptogenesis and shedding now occur after each input, or exemplar, is presented to the model.

3.1 Inhibitory Interneuron

An inhibitory interneuron was added to the model to increase category-exclusivity of output layer neuron codes. The inhibitory neuron functions by limiting the number of neurons that can fire for a given exemplar. When an exemplar is presented to the network, each output neuron generates an excitation based on the weighted sum of its connections with the exemplar input. In the Thomas, et al. (2015) model, any neuron with an excitation higher than the firing threshold would fire in response to the exemplar. With the interneuron, only a top percentile of neurons by total excitation will fire in response to the given input. This ensures that only the neurons that are strongly associated with given categories and subspaces of the input dataset are firing to increase exclusivity. The interneuron was implemented programmatically by changing the algorithm for neuron fires to only allow neurons within the top 10% to fire.

3.2 Associative Modification Changes

The associative modification timescale was changed in the new model to occur during every exemplar presentation. Originally, this occurred after each time every exemplar was presented in a random order, known as an epoch. Also, the associative modification rule was changed to only act on synapse weights that connect the given input to a given output neuron normalized by the excitation. This had the effect of limiting the growth of synaptic weights and firing given a desired firing rate for every neuron.

3.3 Synaptogenesis and Shedding Changes

Like associative modification, both the synaptogenesis and shedding rules are now applied on each presentation of an exemplar. This is important for making the changes to connections more continuous and responsive to the inputs for faster times to convergence. Additionally, a new synaptogenesis cutoff rule was created for the model. The old rule functioned as a threshold equal to the desired firing rate of all neurons. If a neuron exceeded the firing rate, synaptogenesis would shut off for the neuron until it fell below the firing rate. In the new rule, there are two thresholds. The firing threshold is maintained but a lower threshold is also introduced such that neurons that fall below the firing rate do not restart synaptogenesis until the firing rate of the neuron falls below the lower threshold. This formulation helped to create faster convergence since it relaxed the requirement that all neurons fire at the same rate.

3.4 Implementing and Testing the Model

The model was coded in MATLAB. The neural network model was tested using the same datasets as Thomas, et al. (2015) to allow for a clear comparison between model performance and properties. The A, B1, B2, B3 datasets were all replicated in MATLAB.

4. RESULTS

The model maintained the proportional allocation of output neurons relative to input category probabilities as seen in Figure 1, like previous models. However, in the current model all neurons had a fired at the same rate, meaning that neurons were firing to subsections of the categories and mapping out independent sections of the categories.

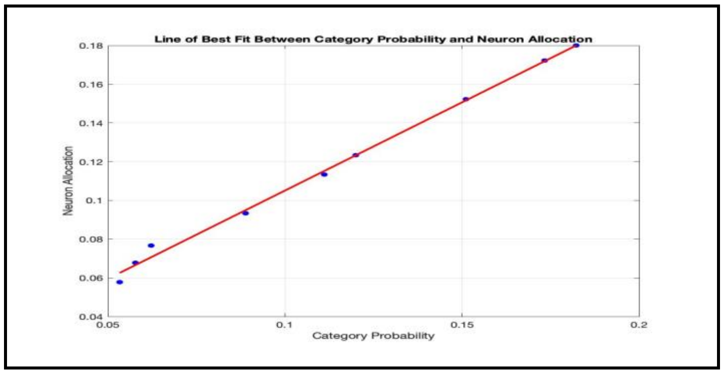

Figure 1: Scatterplot of Allocation Results with a Line of Best Fit

The scatterplot with a line of best fit shown above was created between the output neuron allocation and the input category probabilities of exemplars, showing proportional allocation (R^2 = 0.998).

The model was also able to achieve category exclusivity for the overwhelming majority of neurons in the model. While some neurons in highly correlated categories did fire to multiple categories, this was limited to less than 2% of the network. Thus, as seen in Figure 2, the model was able to approximate a distributed code.

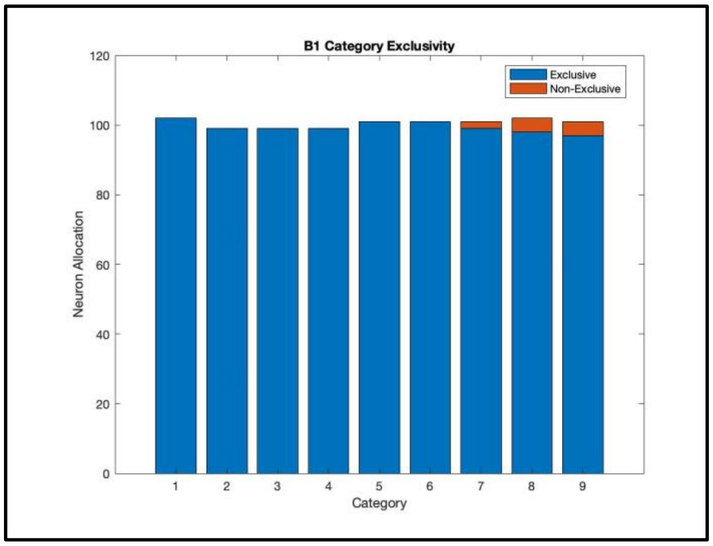

Figure 2: Bar Graph of Category Exclusivity

The exclusivity of neuron firing for the nine categories was determined for the B1. Notably, the B1 dataset on the left had high levels of exclusivity with only 1.33% of the neurons firing to multiple categories. The only categories to observe non-exclusive firing are categories 7-9, which have much higher levels of correlation than the other categories. These results strongly suggest that a distributed code was achieved since neurons are firing exclusively to different subcomponents of each category.

5. CONCLUSION

Achieving a distributed code for the adaptive synaptogenesis neural network model is an important step in the larger goal of developing an in-silico machine learning model that possesses the properties of human brain learning. Factorial encodings have many useful properties for efficiently and optimally coding inputs for many types of segmentation and mixing problems. When incorporated into larger models with more layers and supervised components, this unsupervised model, as a layer, will help to provide conditionally independent input encodings that are more statistically meaningful for downstream layers.

6. FUTURE WORK

While this model was tested on datasets that exposed the network to issues such as correlations between categories, noise, and varying category probabilities, there still needs to be more testing with more complicated and practical datasets.

Additionally, the model could be incorporated into a larger supervised learning task as a preprocessing step to provide a supervised layer better features to utilize. Finally, there are unexplored connections between parameters that still need to better understood such as the effects of a varying or diminishing synaptogenesis rate to speed up convergence.

7. ACKNOWLEDGMENTS

Dr. William B. Levy ideated and guided me through the completion of this project.

REFERENCES

Thomas, B. T., Blalock, D. W., & Levy, W. B. (2015). Adaptive synaptogenesis constructs neural codes that benefit discrimination. *PLOS Computational Biology*, *11*(7). https://doi.org/10.1371/journal.pcbi.1004299

Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural Computation*, *4*(6), 863–879. https://doi.org/10.1162/neco.1992.4.6.863