# Honor's Thesis Prospectus

Mark Musil

November 2018

## 1 Acronyms

Dendritically Modeled Neuron – DMN. Neural Network – NN. Artificial Intelligence – AI. Machine Learning – ML.

## 2 Introduction

Current neural networks used for machine learning are only loosely based on the structure of the human brain (citation needed) but have shown great success in image recognition, functional estimation, and other machine learning tasks. The current model that is widely used - largely re-purposed and customized, although generally homogenous– is the feed forward neural network where input data flows through multiple layers of calculations until a final prediction, classification, or calculation is reached. This model uses a very simple model of the neuron as seen in Illustration 1.

Despite this success, machine learning researchers are hesitant to return to the human brain for more insights into artificial intelligence algorithms (citation needed). This hesitation from the scientific community leaves a large space for investigation into which, if any, operations performed by the human brain may be useful for artificial intelligence goals. In particular, the neuron – the unit of computation – contains many signal processing traits that have not been fully explored. In my thesis, I will be exploring the following question to help further the understanding of the processes within the neuron. Can a non-linear dendritic branch summation function be combined with a two-layer neuron model to create a computational unit that, when used to create a fully connected layer, improves performance of a CNN on the MNIST data-set?

To provide background to the first question, it is first necessary to discuss the dendritic branch summation function that will be implemented. It has long been known that stimuli input to the spine-like synapses along the dendritic branch (see Illustration 2) is not conducted directly to the soma in all cases. There are in fact several factors that impact how much – if any – of a certain stimulus reaches the soma (London and Hausser). First, the distance from the synapse to the soma can affect the net voltage change of the soma because of leakage current that exists along the dendrite. Second, if many stimuli occur at distant locations along a branch but at roughly the same time, the soma's net voltage difference will not be the sum of the voltage changes at the synapse due to each stimulus. The same 'muting' effect takes place when many stimuli take place at relatively close locations but not necessarily the same time. The aforementioned effects do not take into account inter-branch interactions and ignore many details. However they suffice to show that the classical non-leaky sum and activation model of the artificial neuron is woefully over-simplified (See figure 3).

My work will focus on using an existing model of the neuron that incorporates the dendritic transfer function, and then using that model to create a layer for an artificial neural network in order to test the effects of the novel model.

## 3 Anticipated Results

Due to the nature of neural networks it can be hard to predict what creating a new layer may do. This experiment is largely exploratory and it is not clear that it has been attempted before. One interesting new feature added in this model is the property that each dendrite sees only a portion of a neuron's total input space. However, because the neuron has multiple dendrites, it still has the chance to receive input from the entire input space. It will be interesting to see if this segmented input creates a new interaction between the input features and the final output. The linear hook implemented as the dendritic activation function provides a subtle yet significant increase in the granularity of the activation function performed by neurons. The main difference seen in the linear hook that is not seen is the simple sigmoid is linear behavior at low

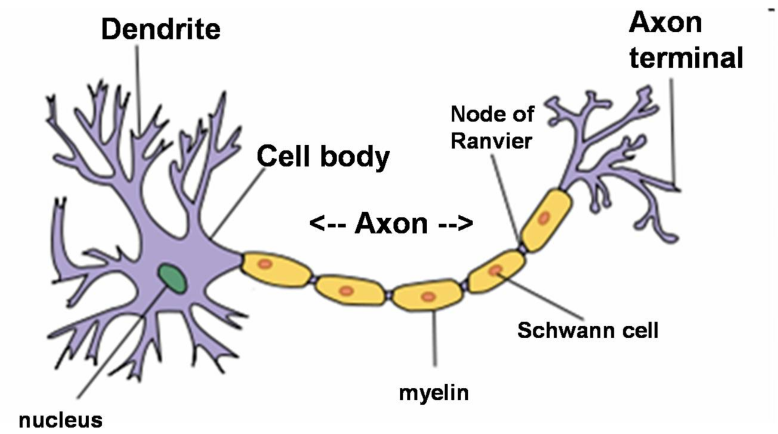


Figure 1: A simple model of the neuron. My work focuses on the synapses called spines found on the dendritic branches. Source: biology.stackexchange.com

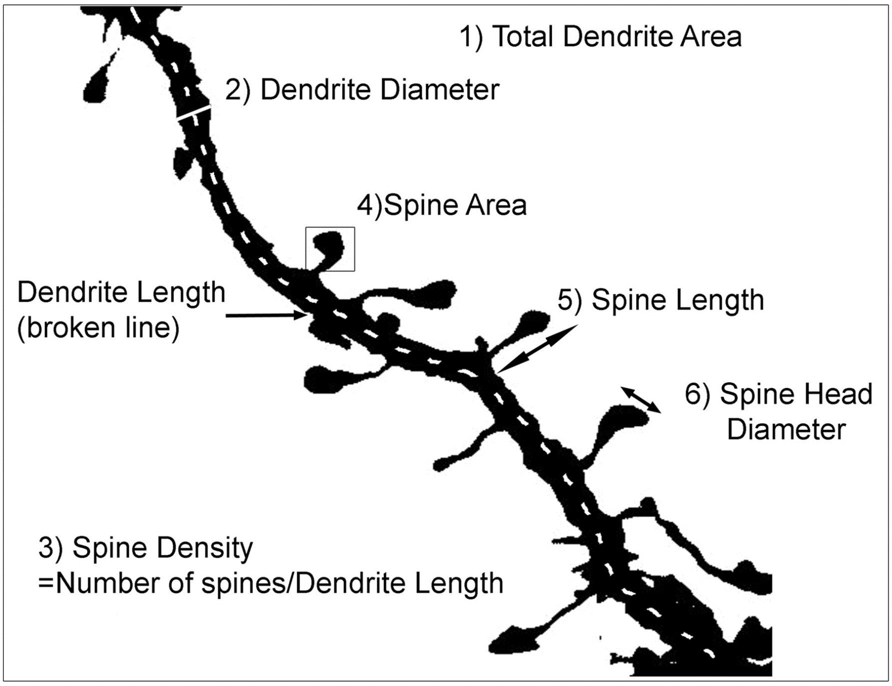


Figure 2: The focus of this thesis will be whether or not the signal processing aspects of the dendritic branch, seen here, impacts overall network processing. Source: pnas.org

activation levels. This may have a large effect on the way the weights are trained and it will be prudent to example a simple fully connected layer trained on the MNIST dataset also. The strength of this experiment is that it attempts only an incremental addition to the existing – largely effective – artificial neuron model. Many bio-inspired computing research projects make the faux-pas of adding gratuitous biological detail and overwhelm their computational resources while simultaneously having no way of knowing which features made a real contribution.

## 4 Methods

The graphical representation for the dendritically modelled neurons (which will hereafter be labeled DMNs) can be seen in figure 4. The addition of the dendritic transfer function $T(\vec{Y})$ is the most significant point to notice as is the fact that each dendritic branch does not see all of the input space but only the $D^{th}$ fraction of the input space where $D$ is the number of dendrites per neuron. The DMNs will be assembled in layers of $N$ neurons where each neuron will be fully connected to the input space but no single dendrite will see the entire input space.

The dendritic transfer function $T(\vec{Y})$ is actually split into two functions in order to aid in readability. These are $T(\vec{Y})$ - the outer function - and $g(y)$ - an inner function - In addition, the multivariate logistic-sigmoid $\sigma_{multi}$ is also defined in equation 1 and used in $T(\vec{Y})$.

$$\sigma_{multi} : R^n \to R = [1 + exp(-\Sigma X_i)]^{-1} \tag{1}$$

$$g(y) = ln(\frac{(1+exp(\alpha_L(y-b_l)))^{\alpha_L^{-1}}}{(1+exp(\alpha_U(y-b_l)))^{\alpha_U^{-1}}}) + b_L \tag{2}$$

$$T(\vec{Y_{dn}}) = g(c_d\sigma_{multi}(a_d[\vec{Y_{dn}} - b_d]) + \Sigma Y_{dn,i}) \tag{3}$$

Note that $\vec{Y}$ and $y$ are used as notation for the inputs to $g(y)$ and $T(\vec{Y})$ as opposed to the conventional $x$ because the input to the dendritic transfer function is the output of the classic fully-connected layers weight function $\vec{Y_{dn}} = \vec{X}W_d + B_{dn}$. $\vec{X}$ is the input activation of the previous layer. Note $B$ is the bias term.

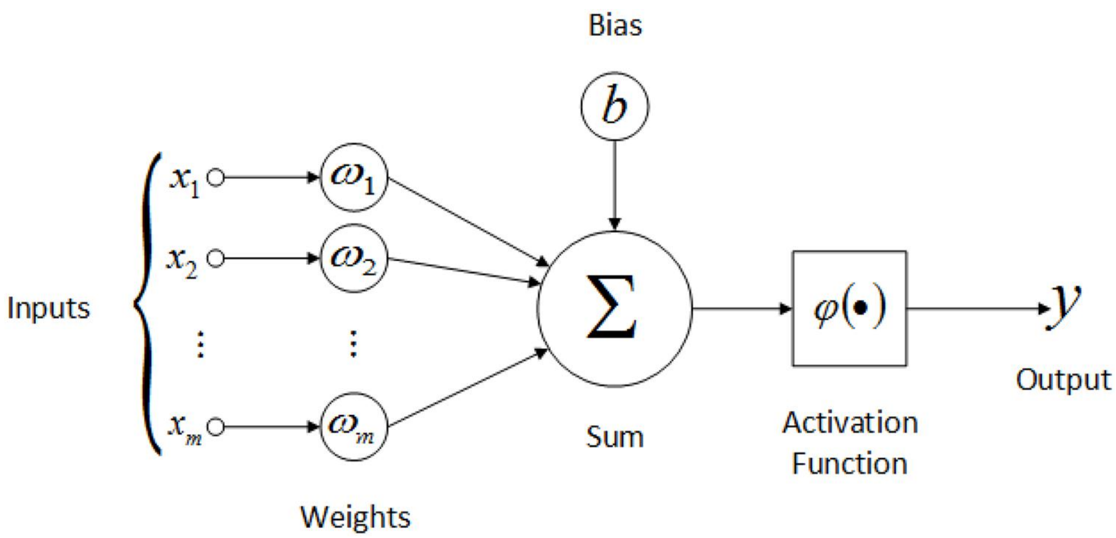


Figure 3: The artificial neuron, which assumes that all synapses are connected to the same lossless dendritic branch and are only acte upon after reaching the soma. Source scielo.br

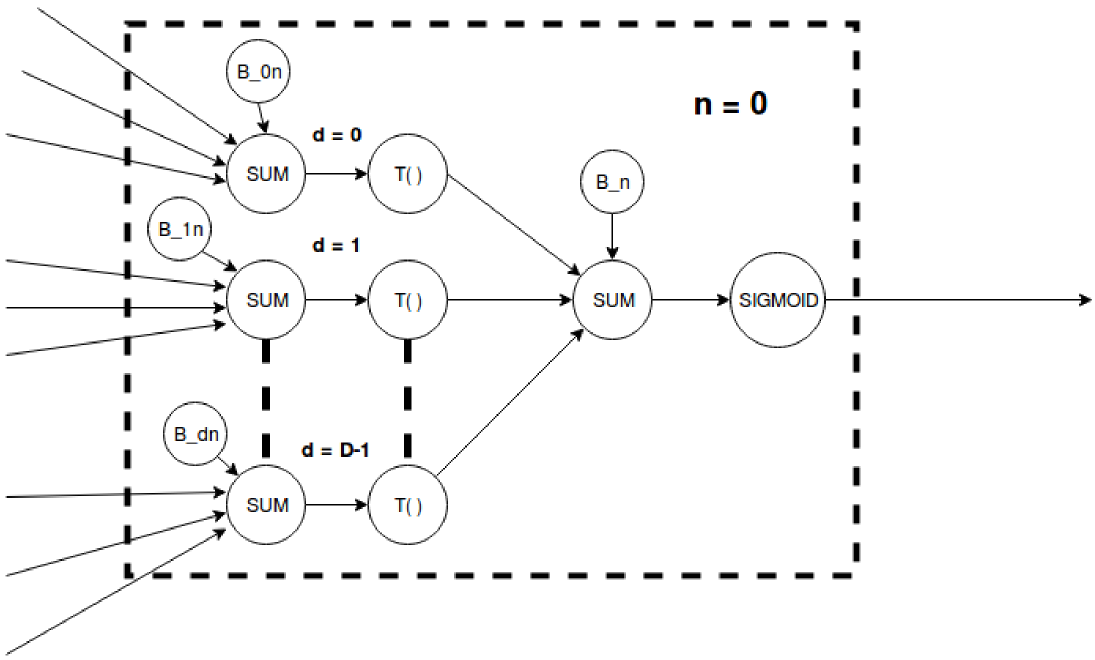


Figure 4: The schematic for the neuronal model that includes the dendritic activation mechanism. $T(\vec{X})$ is the novel function that is being implemented. Two sets of summation and a final sigmoid output are also parts of the model.

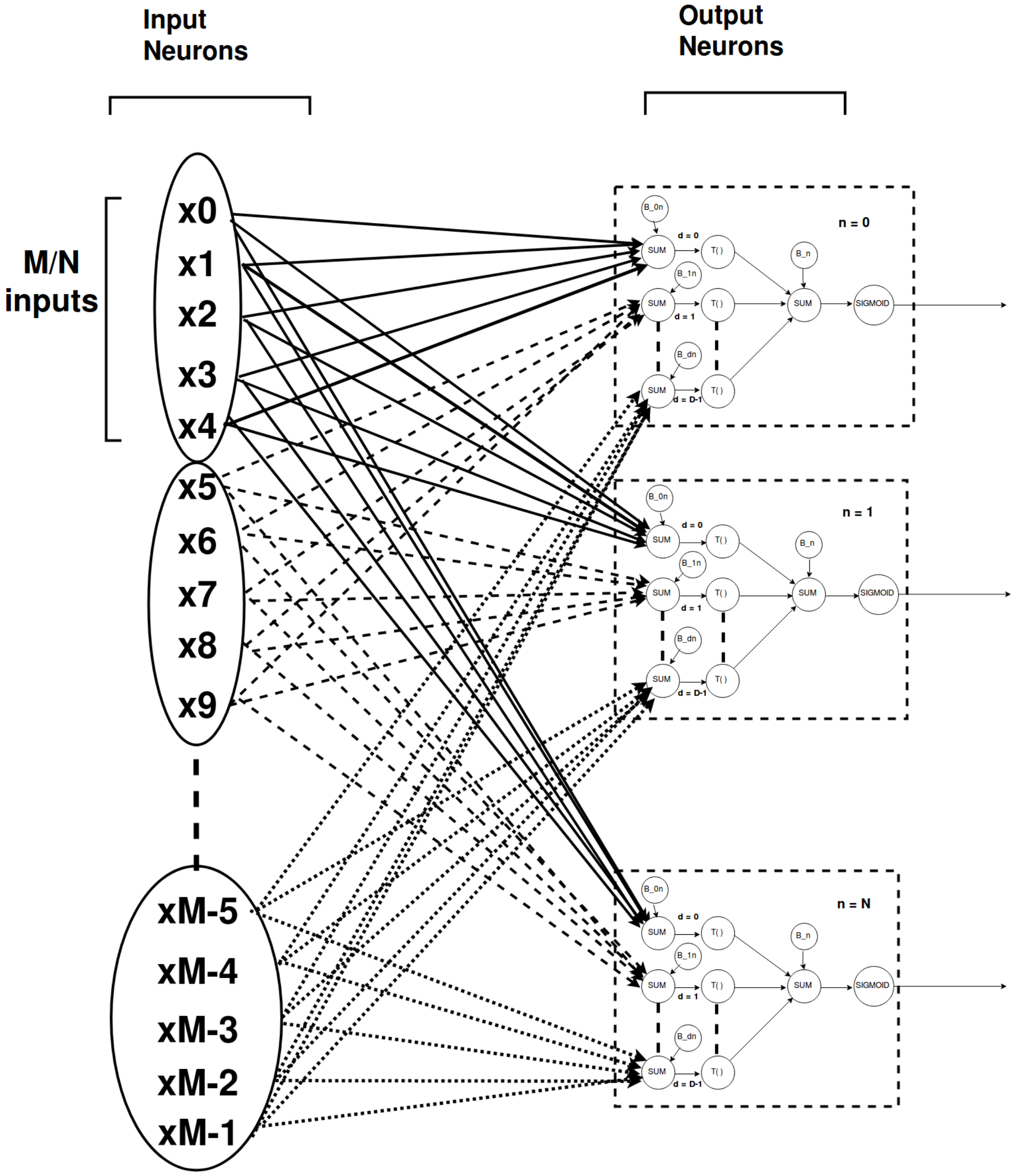


Figure 5: A limited display of the proposed fully connected layer. Note that the same group of M/D inputs connects to the first dendrite of every neuron.

The pseudo algorithm for the layer implementation is as follows. The algorithm assumes that **N** neurons and **D** dendrites have been chosen for the layer for which the computation is taking place. The subscripts $d$ and $n$ denote individuals dendrites ($d$) or neurons ($n$). $b_{dn}$ (two subscripts) is the bias for the dendrite while $b_n$ (one subscript) is the bias for the soma of a neuron.

**Data:** Set of input activations from the previous layer $\vec{X}$
**Result:** Set of output activations of the current layer $\vec{O}$
**for** $k \leftarrow 0$ **to** *N-1* **do**
  **for** $i \leftarrow 0$ **to** *D-1* **do**
    Compute $\vec{Y_{Di}} = \vec{x}W_d + b_{dn}$ ;
    Compute $T(\vec{Y_{Di}})$;
  **end**
  Compute $sigmoidInput = \sum_{n=0}^{D-1} T(\vec{Y_{Di}})$ ;
  Compute $O_k = \sigma(sigmoidInput)$;
**end**