BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 3
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0030

Multi-Activation Dendritic Neural Network (MA-DNN) Working Example of Dendritic-Based Artificial Neural Network

Konstantin Tomov[1], *Galina Momcheva*[2]

[1]High School of Mathematics "Dr Petar Beron", 9009 Varna, Bulgaria
[2]Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
E-mails: kosio.tomov@gmail.com gmomcheva@math.bas.bg

***Abstract*: *Throughout the years neural networks have been based on the perceptron model of the artificial neuron. Attempts to stray from it are few to none. The perceptron simply works and that has discouraged research around other neuron models. New discoveries highlight the importance of dendrites in the neuron, but the perceptron model does not include them. This brings us to the goal of the paper which is to present and test different models of artificial neurons that utilize dendrites to create an artificial neuron that better represents the biological neuron. The authors propose two models. One is made with the purpose of testing the idea of the dendritic neuron. The distinguishing feature of the second model is that it implements activation functions after its dendrites. Results from the second model suggest that it performs as well as or even better than the perceptron model.*

***Keywords*: *Activation function, Neural network, Artificial intelligence, Dendritic neuron, Neuron model.*

1. Introduction

The structure of artificial neural networks has the goal of mimicking the structure of the biological neural network. Recent discoveries in the space of neuroscience have highlighted the importance of dendrites in the structure of the neuron. For a review of existing hypotheses for the role of dendrites in the computational function of biological neurons see [1-3]. These discoveries have led the field in a new direction of implementing the dendrite into the structure of the artificial neuron.

1.1. Artificial neural networks

Researchers have always been interested in the unfathomable abilities of the brain to learn and interpret foreign information. This interest in turn falls onto the structure of the brain and the billions of neurons and connections between them. In recent years, researchers have begun to try and emulate these capabilities of the brain by creating an artificial version of it. The artificial versions are called artificial neural networks, and they try to emulate the functionality of the brain by mimicking its structure. In recent years, the processing power of computers has tremendously increased, and so artificial intelligence has taken a great leap forward.

1.2. Biological neuron

The brain is a nervous system composed of cells called neurons and connections between them. Neuron cells transmit information, they communicate with other neuron cells, muscles, or gland cells. The human brain contains on average around 100 billion neurons each of which is connected with up to 10000 other neurons. Estimates of the human brain's memory capacity vary wildly from 1 to 1000 terabytes [4]. The primate brain is also more densely packed with neurons and the human brain is the biggest among primates [5]. Although the process by which the brain learns is not fully understood it is certain that these metrics play an important role, especially the number of neurons and their interconnectivity.

The typical neuron contains a soma (a body that contains the cell nucleus and most of the organelles), dendrites (relatively short and thick protrusions that branch close to the soma and form input contacts with other neurons) and an axon (a thin long outgrowth that branches away from the soma. It forms the output connections of the neuron), as shown in Fig. 1. It is noteworthy that on average dendrites receive more than 90 percent of all synaptic input to the nerve cell [6]. The neurons possess the so-called excitable membrane: an uneven distribution and permeability of ions create an electrochemical gradient at rest. The inner membrane surface is negatively charged relative to the external. Changes in the ionic permeability cause voltage gradient fluctuations (hyperpolarization and depolarization) across the membrane. These phenomena are facilitated by the presence of special classes of ion channels with modifiable permeability in the membrane named gated ion channels. They resemble a protein pore in the membrane with a gating mechanism that is either open, allowing the passage of certain ion types (K^+-channels, Na^+-channels, Ca^{2+}-channels), or closed. Depending on how the gate is controlled, we distinguish ligand-gated (controlled by a chemical signal), voltage-gated (open when there is a certain change in the transmembrane potential), and mechanically gated. The active electrical properties of the excitable membrane depend on the voltage-gated channels. The changes in permeability are countered by existing active transport mechanisms (ionic pumps) that redistribute the ions across the membrane against their electrochemical gradients. That is why most of the evoked changes in transmembrane potential are localized and attenuated rapidly along the membrane. When the depolarization however (the reduction of the voltage difference) crosses a certain threshold in sensitive areas of the neuronal surface (the axonal hillock at the base of the axon), a rapidly propagating spike of depolarization of the axon is initiated – Action Potential (AP). The AP travels along the axon and affects neighboring neurons via specialized contacts called synapses. The opposite voltage fluctuations (hyperpolarization, i.e., the voltage gets more negative) reduce the probability of generating an AP. The AP initiation is influenced by thousands of synapses residing on the neuronal surface (more than 90% in the dendrites) that evoke local voltage fluctuations in the postsynaptic membrane (in most cases via chemical signaling). The evoked postsynaptic potentials are usually short-lived depolarization in the excitatory synapses (Excitatory PostSynaptic Potential – EPSP) or hyperpolarization in the inhibitory (Inhibitory PostSynaptic Potential – IPSP). Utilizing these mechanisms neurons transmit, integrate, process, and transfer information one to another.

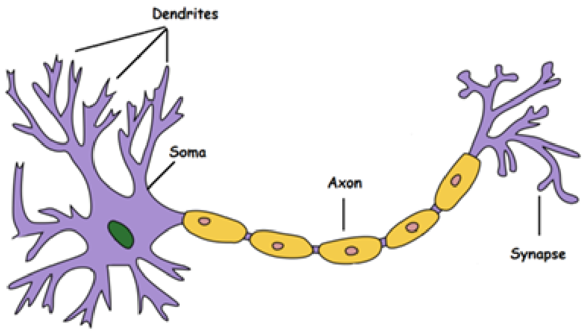

Fig. 1. A depiction of a neuron with labeled dendrites, soma, axon, and synapses

1.3. Artificial neuron (perceptron)

One of the first attempts to recreate the function of the biological neuron was the perceptron (McCulloch-Pitts neuron). It was first described theoretically by McCulloch and Pitts [7] in 1943 and simulated for the first time by Rosenblatt [8] in 1957. Since its inception, the perceptron artificial neuron model or models following its paradigms have continued to be the most used artificial neuron models. These models have been the go-to artificial neuron models for years on end, but they lack one key aspect – they do not include dendrites into their structure. Attempts to create models that stray from the simple perceptron paradigms have been made, but they are few and from our experience, literature on the subject is scarce. One model differing from the standard perceptron paradigm that has garnered a lot of success is the convolutional model. It, however, has specific connectivity patterns that are not in the scope of our paper.

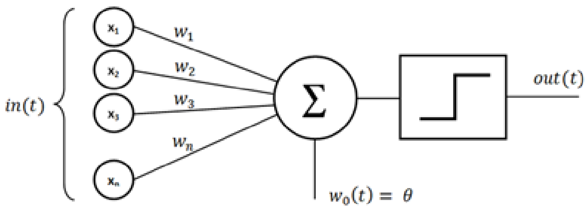

Fig. 2. A representation of a single perceptron. Its input layer, weights, sum function and activation function are shown

The perceptron is a single layer artificial neural network that mimics the biological neuron. It consists of four parts: input values, their weights (and biases), a net sum, and an activation function. The perceptron takes the input values, multiplies each of them with a weight and adds a bias. Then the results from these multiplications are summed, and the sum is then passed through an activation function. The perceptron and the inputs are connected via the weights that act as a strength modifier of a particular connection. The structure of the perceptron is presented in Fig. 2: the input values and their weights comprise a basic representation of the biological dendrites, the net sum acts as the soma and the activation function as the output of the perceptron (axonal hillock and axon). The perceptron learns by fine-tuning its weights to produce a result closer to the one wanted.

The perceptron on its own is not very powerful and can solve only linearly separable problems. A simple problem the perceptron cannot solve is the XOR problem [9]. Therefore, more sophisticated methods like the multi-layer perceptron networks are required. Multi-layer perceptron networks are composed of an input layer, hidden layers, and an output layer (Fig. 3). The input layer consists of perceptrons that take in the data that the neural network will process. The hidden layers are composed of many perceptrons connected to the previous layer. The output layer of perceptron also receives connections from the previous layer and its output is the result that the network produces. The multi-layer perceptron model can learn to solve harder problems such as the XOR problem, classification of information, speech recognition, image recognition, translation, and many others. For some of these more complex tasks the multi-layer networks include additional operations: pooling, flattening, dropout, but these operations are not in the scope of this paper.

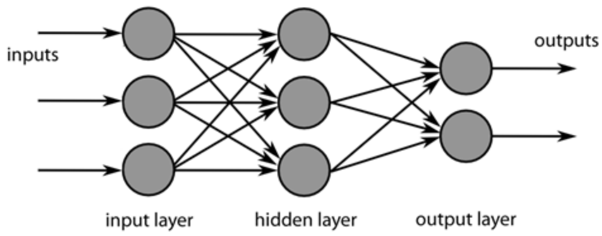

Fig. 3. Depiction of a multi-layered neural network

1.4. Dendritic neuron

Experimenting with different structures of the artificial neuron is one of the many possible ways to improve the existing models. One approach that is currently still being experimented on is the implementation of the dendritic neuron structure; however, reports in the specialized literature are still limited. The main idea behind it is the fact that the perceptron neural model does not consider the existence of the dendrites in a standalone manner in the biological neuron. As stated above dendrites receive more than 90 percent of the nerve cell input [6]. Additionally, the structure of the dendritic tree (number and length of dendrites, and branching pattern) is tightly related to the function of the neuron and pathological and functional changes are related to changes in the dendritic tree. These observations motivate us to believe that dendrites play a vital role in the learning process of the brain and should be more prominently included in the artificial neurons. Throughout the paper, the term "dendritic neuron" is used for the model of a neuron that implements dendrites.

The dendritic neuron differs from the perceptron in that it incorporates dendrites into its structure. Each dendritic neuron has several dendrites. Every one of the inputs can be connected to none, one or many of the dendrites of the neuron. Each dendrite calculates a value based on the inputs connected to it. In most cases, the value is computed by multiplying the inputs. All the values computed by the dendrites of the neuron are then summed and passed through an activation function. The output of the activation function is the result of the dendritic neuron (Fig. 4).

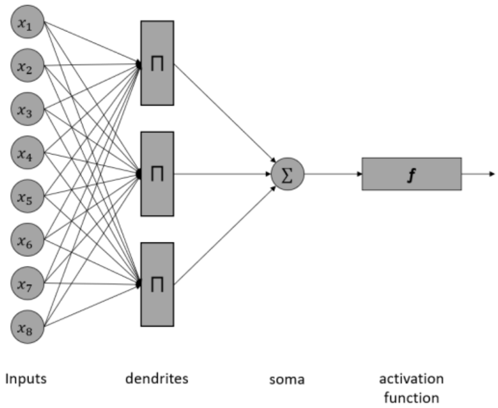

Fig. 4. A representation of a dendritic neuron with 8 inputs and 3 dendrites and full connectivity to the input layer. The values from the inputs are multiplied in every dendrite. The values of the dendrites are then summed and passed through an activation function

The dendritic neuron on its own cannot solve complex problems. It can be used to solve more complex tasks, in the same manner as the perceptron in multi-layer networks. It can also be used in conjunction with perceptrons in the same multi-layer network. Again, multi-layer networks with dendritic neurons can include layers different from dendritic neuron layers.

2. Related work

2.1. Biological aspects of dendritic computation

In the 1940s-50s Hodgkin and Huxley demonstrated brilliantly that neurons are transmitting signals composed of modulated membrane ion permeability that manifests as AP. At this time, the nature of computation occurring inside the nervous system and its fundamental unit has not been understood. It was assumed that since the signal transfer from cell to cell occurs mostly on dendritic synapses, the dendrites serve to collect signals, which are then integrated linearly in the neuronal soma, to finally initiate (or not) a response in the axonal hillock in the form of AP. This view has been challenged by the pioneering work of Wilfrid Rall, who in 1959 showed that the dendrites' conductance follows the cable law, and passive currents are rapidly attenuated along the dendritic tree and its membrane also possesses active properties [6]. This sparked serious interest in the active and passive integrative properties of the dendrites. With recent advances in electrophysiology, cell biology, imaging, and computation, a lot of data on the biological, molecular, and biophysical properties of the dendrites have been gathered that support the hypothesis that the dendritic branches operate as computational subunits, capable of generating dendritic spikes, and at the same time may act as passive filters, attenuating or augmenting currents initiated by synapses (locally or stemming from more distant branches) [1]. A detailed summary of the current knowledge about the properties and the physiology of the biological dendrites and the related hypotheses about the nature of biological dendritic computation may be found in [10] and [6].

The most important features of the dendritic tree may be summarized as follows:

- The dendritic tree consists of converging segments that receive input from multiple synapses with diverse molecular makeup and modality (excitatory or inhibitory).
- The structure and properties of the dendritic membrane differ from the somatic and axonal membrane.
- The dendritic segments between two branching points have a relatively constant diameter that increases as the tree converges on the soma.
- The propagation of the local passive or active currents between segments is modulated by synaptic activity as well as by back-propagating APs.
- The tree architecture significantly influences the passive (filtering) and active (generation of dendritic spikes) properties of the dendritic segments.
- The integration of the passive and/or active responses of the individual segments determines the effect of the entire subunit of the dendritic tree on the ability of the neuron to generate APs or change the frequency of AP generation.
- Back propagating APs from the axonal hillock strengthen synapses with an activity that is phase-locked with the firing of the postsynaptic neuron and conversely weaken synapses that are out of phase.
- The number and distribution of synapses, and their strength can be modified by internal cellular mechanisms.

Based on the current understanding of the functional and structural properties of the biological dendrite, a novel view arises that the neuron may be treated as a multilayer network where linear integration occurs at the level of individual branches and the multiple branch response is a result of nonlinear activation at the nodes of conversion [1].

2.2. Existing dendritic models

In the field of the dendritic neuron model, one structure has garnered a great deal of attention. It is frequently used in research papers on the dendritic neuron [11, 12] and is seen under many names. It consists of four layers namely a synaptic, dendritic, membrane and somatic layer. In the next paragraphs, we go over the structure of this model.

The synaptic layer receives inputs and sends signals to the dendritic layer. After a certain threshold is passed the synapse fires. To simulate this the following sigmoid function is used.

(1) $$\mathrm{Y}_{i,m} = \frac{1}{1+e^{-k(w_{im}x_i - q_{im})}},$$

where x_i is the input and $Y_{i,m}$ is the output of the *i*-th synapse in the *m*-th branch of dendrites. The parameters *k*, w_{im} and q_{im} need to be tuned if we want the synapse to have an adaptive function [13]. A threshold is calculated for the synapse in the following way:

(2) $$\theta_{im} = \frac{q_{im}}{w_{im}}.$$

After the activation of the synapse, it can adopt one of four states according to the ranges of w_{im} and q_{im}.

- Direct-connecting state – if the value of the input x_{im} is greater than θ_{im}, the value of the output approximately equals 1; otherwise, it equals 0.
- Opposite-connecting state – if the value of the input x_{im} is less than θ_{im}, the value of the output approximately equals 1; otherwise, it equals 0.
- Constant-1 state – the value of the output is always 1.
- Constant-0 state – the value of the output is always 0.

The outputs of the synaptic layer for each dendrite are multiplied and a value is computed. The dendritic layer is summarized by the following equation:

(3) $$Z_m = \prod_{i=1}^{I} Y_{im}.$$

The membrane layer takes the outputs from the dendritic layer and linearly sums the values. The membrane layer can be summed up by the following equation:

(4) $$V = \sum_{m=1}^{M} Z_m.$$

The somatic layer receives the signal from the membrane layer. The signal is then passed through an activation function,

(5) $$O = \frac{1}{1+e^{-k_{\text{soma}}(V-\theta_{\text{soma}})}}.$$

The distinguishing features of this dendritic neuron model are the full connectivity between the inputs and dendrites, the added complexity of the adaptive synapse, the multiplication in the dendrites and the summation in the membrane layer.

3. Proposed methodologies

We propose two dendritic neuron models. Both models differ from the existing go to neuron structures. The first model serves as a stepping-stone showing the inherent capabilities of the dendritic neuron structure. The things distinguishing it from the existing models are that the connectivity between the dendrites and the inputs is very sparse and done in a controlled manner, and that adaptive synapses are not used. The second model is more complex and could see more applications. The things distinguishing it from the existing models are the implemented activation functions for every dendrite branch, and that adaptive synapses are not used.

3.1. Undecorated dendritic neuron model

The undecorated dendritic model is greatly inspired by the paradigm reviewed in the related work chapter. The purpose of our model, however, is to show the dendritic neuron model idea in its most basic form. There are no complications to this model, such as adaptive synapses. It is worth being noted that although models proposed in [11, 12], through the implemented pruning, may achieve a structure, at some point in their training process, similar to that of our proposed model. That, however, cannot be for certain; it is not controlled and was not the goal of the authors. What is certain is that our model will always have that structure, as it was hardcoded. The idea behind the model is to better compare the dendritic neuron idea to the existing perceptron. Making this comparison gives us the opportunity to figure out if the dendrites really play that big of a role in the structure of the neuron, without other complications.

The model works with four different types of trainable parameters of which two are weights and two are biases. A set of weights and biases is used as a connection between the inputs and the dendrites. For every neuron, a weight and bias are trained. All inputs for an undecorated dendritic neuron are multiplied by the weight and then the bias is added. After this, the computed values are multiplied in the dendrites. Every neuron also has a set of weights and a single bias for its dendrites. A single weight for every dendrite and a bias for the dendrites are trained; let us call these biases and weights dendritic. The computed values in the dendrites are multiplied by the dendritic weights and then the dendritic bias is added. After that, the values for all dendrites in a single neuron are summed and passed through an activation function. The trainable parameters per neuron are numDendrites+1 weights and two biases. By swapping a perceptron with an undecorated dendrite we use fewer trainable parameters as the number of trainable parameters per perceptron depends on the number of inputs, and in most cases, the used number of dendrites will be less than the number of inputs. The connections between inputs and dendrites in Fig. 5 represent multiplication, not a trainable parameter. The training algorithm is backpropagation.

Although inspired by the model in the related work chapter, our model differs greatly. The first difference between the models is that all input values coming into our dendritic layer can be connected to exactly one dendrite. That is, the layers are not fully connected; they are connected very lightly. By swapping a perceptron layer with an undecorated dendritic layer, we keep the number of connections the same and lower the number of trainable parameters. This is an important fact that shows that any differences in accuracy purely stem from the innate essence of the dendritic neuron and not a larger number of parameters. The inputs are spread out amongst the dendrites as equally as possible. All dendrites will have either $\lfloor\frac{\mathrm{numInputs}}{\mathrm{numDendrites}}\rfloor$ or $\lceil\frac{\mathrm{numInputs}}{\mathrm{numDendrites}}\rceil$ inputs connected to them, as seen in Fig. 5. Another difference between both models is that our model does not search for extra complexity. No adaptive synapses are used. These differences between the models are made with the goal of creating a dendritic neuron model that is void of extra complexity. This is done to better compare the perceptron model with the true essence of the dendritic neuron.

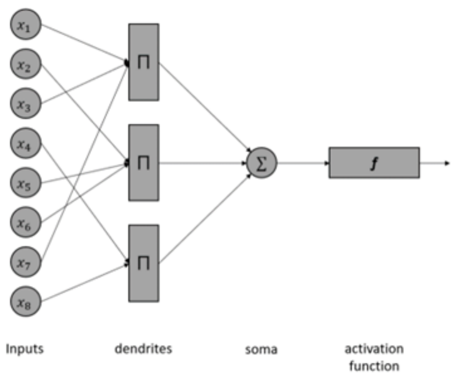

Fig. 5. A representation of the undecorated dendritic neuron with eight inputs and three dendrites

3.2. Multi-activation dendritic neuron model

The crucial thing distinguishing this model from others is that in its structure, after every dendrite there is an activation function. These activation functions bring extra complexity to the model and act as a way to make the dendrites more influential. These activation functions can differ and are independent of each other. Another noteworthy thing is that the dendrites are fully connected to the layer preceding the multi-activation neuron. For a better understanding of the structure of the model see Fig. 6.

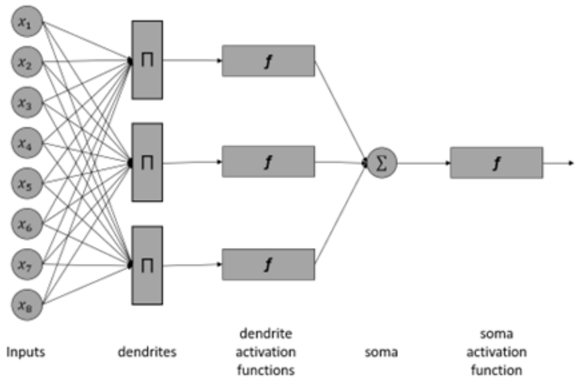

Fig. 6. A representation of the multi-activation dendritic neuron model with eight inputs and three dendrites. Note that all of the activation functions can be the same, different, and some can be the same while others are different

3.2.1. Structure of the dendrites

All neurons from the layer before a multi-activation dendritic neuron layer are connected to every dendrite of every neuron in the multi-activation dendritic neuron layer. In other words, this layer is fully connected to the preceding layer. This connectivity provides us with many tunable parameters.

The dendrites compute their value in the following manner. Every value from the previous layer is multiplied with a trainable weight. A sum of all of these multiplications is computed for every dendrite. This sum is then passed through an activation function. Different dendrites can have different activation functions. There is no limit on the number of activation functions but the number of dendrites with each activation function stays as close to equal as possible.

3.2.2. Structure of the soma

The soma of the model takes the computed values by the activation functions for each dendrite and sums them. This sum is then passed through another activation function. The computed value is the output of the neuron.

4. Results

As the models are very different in structure they are tested individually. This is done on well-known public datasets. Testing on some of the datasets is not done purely to achieve accuracy, but on others it is oriented around delving deeper into the capabilities of the dendritic neuron and bringing them to light.

4.1. Cross-validation

The datasets used are the Fashion MNIST [14], Iris [15], Breast Cancer Wisconsin (Diagnostic) [16] and the CIFAR-10 [17] datasets. The premise of the paper is not to achieve the most optimized and well performing architecture on the datasets, but to test the proposed models against the perceptron model in a variety of neural network architectures. For each dataset we picked the customary data split and used it throughout testing. Details about the splits can be found in Table 1.

Table 1. Ratios of the train to validation data split for the different datasets used

Datasets	Train to validation ratio
Fashion MNIST	6:1
CIFAR-10	5:1
Iris	9:1
Breast Cancer	4:1

4.2. Results on the undecorated dendritic neuron model

The proposed dendritic neuron model has been tested on the Fashion MNIST, CIFAR-10, and the Breast Cancer Wisconsin (Diagnostic) datasets. The data in the Breast Cancer Wisconsin (Diagnostic) dataset is closely tied to research based around mammogram images. Research on this topic can also be found [18]. Tests have been made on models where dendritic neuron layers are inserted into perceptron based neural networks, on models where perceptron layers are completely swapped out for the dendritic neuron layers, and on neural networks implementing other types of layers. The models chosen for the final ranking are the best performing ones in accuracy among the tested models. All models incorporating perceptrons were also been tested in the same structure but with some or all perceptron layers swapped out for dendritic neuron layers. Results on the Fashion MNIST dataset do not indicate an edge of neither the perceptron nor the dendritic model. Results from the CIFAR-10 dataset indicate that the perceptron models perform better. This could be a byproduct of the fact that every input to the dendritic neuron connects to only one dendrite and with that, some learning capabilities could be lost. The best-performing architecture on the Breast Cancer Wisconsin (Diagnostic) dataset used no perceptron layers, only dendritic layers. The lead it has over other architectures on the dataset is non-negligible and very promising for the field. It also achieves a loss two times smaller than the next best-performing architecture. The used loss function is sparse categorical cross-entropy. For the achieved results, see Table 2.

Table 2. Results from testing the undecorated dendritic neuron model. CNN is a convolutional neural network, UD is the undecorated dendritic neuron, DNN is a deep neural network

Dataset	Architecture	Accuracy
Fashion MNIST	CNN + UD	92.33
	CNN + perceptrons	92.22
CIFAR-10	CNN + perceptrons	74.62
	CNN + perceptrons + UD	73.22
Breast Cancer Wisconsin	UD	99.12
	Perceptrons	98.25

The architectures of the best performing models are shown in Figs. 7-12. The layers are described as follows: convolutional layers by the number of filters, kernel size, and activation function, pooling layers by kernel size, dropout layers by drop rate, dense perceptron layers by number of perceptrons and activation function, and undecorated dendritic layers by number of neurons, number of dendrites per neuron, and activation function.

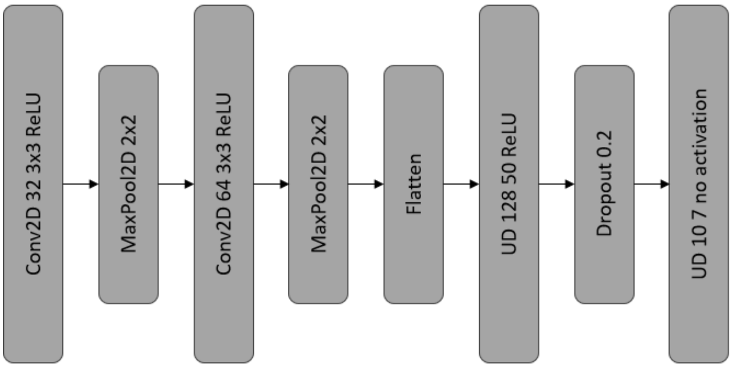

Fig. 7. Fashion MNIST CNN + UD architecture

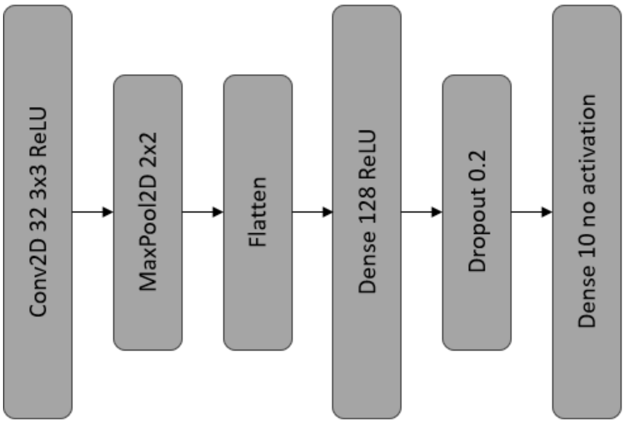

Fig. 8. Fashion CNN + perceptrons architecture

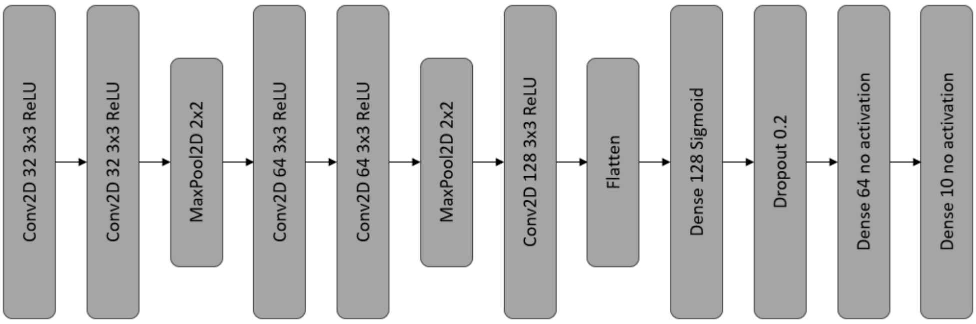

Fig. 9. CIFAR-10 CNN + perceptrons architecture

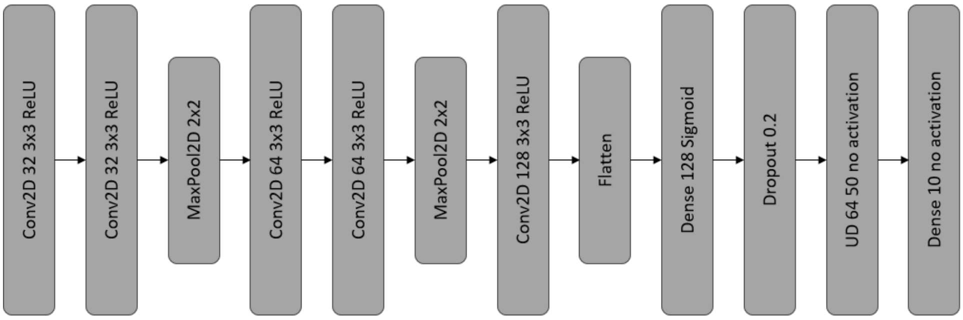

Fig. 10. CIFAR-10 CNN + perceptrons + UD architecture

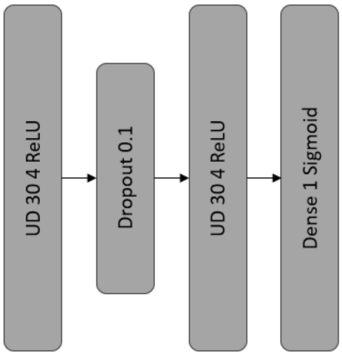

Fig. 11. Breast Cancer Wisconsin UD architecture

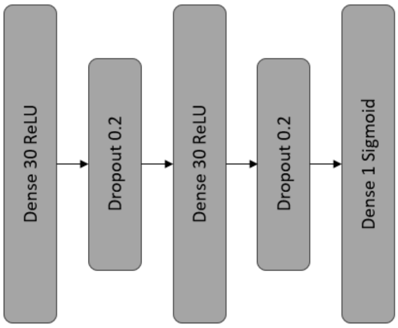

Fig. 12. Breast Cancer Wisconsin perceptrons architecture

The models shown in Fig. 7 and Fig. 8 are different, but they are compared with each other. This is because they are the best performing ones from all that have been tested. The architecture shown in Fig. 7 has also been tested with perceptrons but it achieved a lower accuracy that did not make the list of best performing neural networks and in turn is not shown in Table. 2.

These results give us more ground to believe the dendritic neuron model is inherently better than or at least as good as the perceptron neuron model. More testing is still needed, as the idea behind the model was to serve as a representation of the dendritic neuron structure in its simplest form as an effort to better compare it to the perceptron model.

4.3. Results on the multi-activation dendritic neuron model

Testing has been done on the Iris, Fashion MNIST, and Breast Cancer Wisconsin (Diagnostic) datasets. As the testing of the undecorated dendritic model the architectures are made up of perceptron layers, multi-activation dendritic neuron layers and others. All architectures including perceptrons have also been tested in the same structure but with some or all perceptron layers swapped out for multi-activation dendritic neuron layers. For the models utilizing a sigmoid activation function, the function $\mathrm{sigmoid}(x) = \frac{1}{1+\exp(-x)}$ is used.

4.3.1. Hyperparameter tuning

As the structure of this model is more complex and has tunable parameters, we need to find the most fitting values for them. The best-performing hyperparameters for the model have been established during training on the Fashion MNIST dataset. The hyperparameters tested for have been the number of dendrites, the number of activation functions, and the activation functions themselves.

With the increasing number of dendrites per neuron, the accuracy of the model has increased, and the loss decreases. This is observed up until 10 dendrites per neuron, after this bound is crossed the trainable parameters become too many and the architecture begins to express randomized behavior that stabilizes too slowly for research purposes.

The number of activation functions used in the field is quite small, but the best performance has been achieved using 2 or 3 activation functions. There is room for more experimentation as there are no bounds set for the number of activation functions or their kind in the model.

Results indicate that models utilizing the softmax and sigmoid functions in the dendrites and the softmax function in the soma perform best overall. Good results have also been achieved on neurons utilizing only the ReLU function.

4.3.2. Results on the Fashion MNIST dataset

As seen in Table 3, results from the fashion MNIST dataset show a big lead for the models utilizing the multi-activation dendritic neurons. Networks utilizing convolutional neural network paradigms perform better overall, as expected. An interesting result is that the best-performing architecture has been achieved by swapping out all perceptrons for the proposed dendritic neurons. It has a slightly larger loss, but it achieves the best accuracy overall. Results from testing on networks that do not utilize convolution again show a lead for the architectures utilizing dendritic neurons. When utilized, the softmax functions Ha a dimension of 10 as that is the number of classes in the dataset. The loss function used was binary cross entropy.

An interesting observation is that architectures that use multi-activation dendritic neurons that use the ReLU and the softmax or sigmoid function have a significantly higher loss but still achieve great accuracy. This could be explained by the fact that the value of the ReLU function can be a very large number and it can shadow the value of the softmax and sigmoid functions, which in turn messes with the training process.

Table 3. Best-performing architectures from testing on the Fashion MNIST dataset

Structure	Number of dendrites per neuron	Used activation functions for dendrites	Loss	Accuracy
CNN + dendritic neurons	10	Softmax and sigmoid	0.0527	0.9173
CNN + perceptrons	-	-	0.0224	0.9143
DNN with only dendritic neurons	10	Softmax and sigmoid	0.1133	0.9035
DNN with only dendritic neurons	10	ReLU	0.1230	0.9018
DNN with only dendritic neurons	4	ReLU	0.1276	0.8989

The architectures of the models in Table 3 are shown in Figs 13-17. The same pattern for describing the layers is used. The difference is that undecorated dendritic layers are not used; the used dendritic layers here are multi-activation dendritic neuron layers or MaDN layers. They are described by the number of neurons, the number of dendrites per neuron, the activation functions for the dendrites, and on a new line the activation function of the neuron.

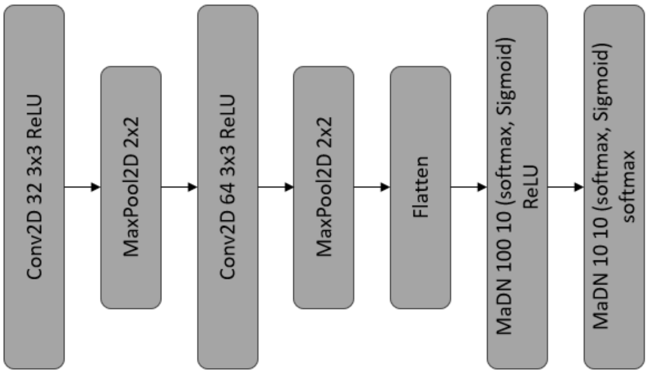

Fig. 13. CNN + dendritic neurons architecture

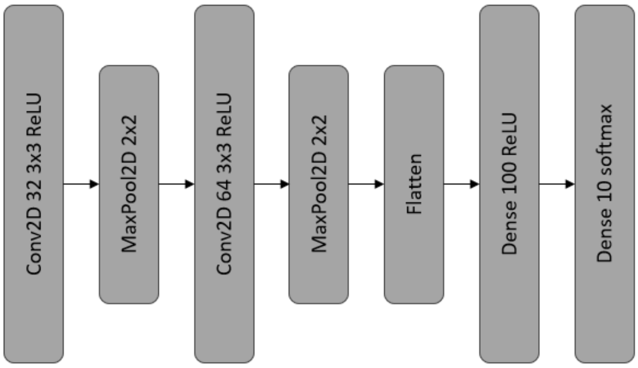

Fig. 14. CNN + perceptrons architecture

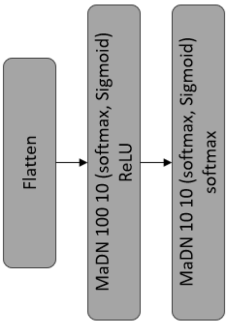

Fig. 15. DNN with only dendritic neurons,10 dendrites with softmax and Sigmoid activation architecture

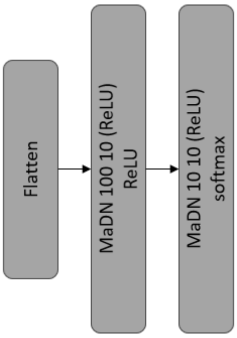

Fig. 16. DNN with only dendritic neurons, 10 dendrites with ReLU activation architecture

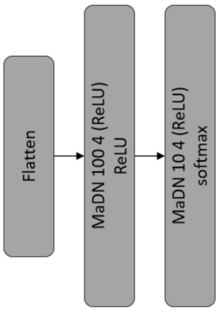

Fig. 17. DNN with only dendritic neurons, 4 dendrites with ReLU activation architecture

4.3.3. Results on the Iris dataset

This dataset is very simple, so it is very easily fully learned by networks in turn experimenting on this dataset is used as a showcase that the multi-activation dendritic neuron model layer is not just a morphology of two perceptron layers. The simplest multi-activation dendritic neuron architecture that fully learns the dataset is a single layer with three neurons, each one having three dendrites with all activation functions set to sigmoid. In an effort to debunk our hypothesis that the multi-activation dendritic and two perceptron layers behave differently, we train an architecture composed only of perceptrons against the structure made up of three multi-activation dendrite neurons. It is a two-layer perceptron network where every layer is composed of three perceptrons. In testing this architecture does not always learn the dataset and in the cases in which it succeeds it does so slower.

These results reveal that there is a strong boundary between the multi-activation dendritic neuron model and a normal two-layer perceptron neuron network. The results also favor the model utilizing dendrites.

4.3.4. Results on the Breast cancer Wisconsin (diagnostic) dataset

As this dataset is also quite simple in structure, testing has been done again in a peculiar manner. The architectures tested were a perceptron based neural network with the following structure: one layer containing 20 perceptrons, a dropout layer, a perceptron layer containing 10 perceptrons and a layer containing one perceptron. The last layer utilizes the sigmoid function, and the other layers use the ReLU function. The other structure tested follow the same architecture with the difference that the perceptrons have been swapped out for multi-activation dendritic neurons. The neurons have two dendrites utilizing the ReLU function. Each architecture has been put through the training process five times. The results for each of the runs is the best results achieved throughout the learning process. Accuracy is prioritized and after that loss. The used loss function is binary cross-entropy. The results are seen in Table 4. and Table 5.

Results indicate a clear lead of the architecture utilizing dendrites into its structure. The perceptron models achieve a better loss overall but their accuracy lacks. An interesting observation made during the training process is that after the perceptron model reaches its peak accuracy and the training process continues, the loss begins to steadily increase. This phenomenon is not observed in the architecture utilizing dendrites into its structure.

Table 4. Results from testing on the Breast Cancer Wisconsin (diagnostic) dataset with a perceptron neural network

Perceptron neural network	
Loss	Accuracy
0.2683	0.9825
0.1550	0.9825
0.0815	0.9825
0.1282	0.9737
0.1369	0.9649

Table 5. Results from testing on the Breast Cancer Wisconsin (diagnostic) dataset with a multi-activation dendritic neural network

Neural network utilizing dendrites	
Loss	Accuracy
0.2730	0.9825
0.2634	0.9912
0.2806	0.9825
0.2671	0.9912
0.2671	0.9825

4.4. Comparison with existing models

A comparison between our proposed models, most importantly the multi-activation dendritic neuron model, and the ones proposed in [11, 12] cannot be made. The cited models have been tested only on existing datasets that are morphed into binary classification datasets. This has been done because the cited papers only explore single neurons. The step into a deep dendritic neural network has not been made and without it a direct comparison between the models cannot be made.

5. Conclusion and future work

Testing on all models provides extremely interesting results. These results seem to favor architectures using neurons utilizing dendrites. In other words, neurons utilizing dendrites perform better than the perceptron model that has been used for years on end. Results of the model in shown in the related work section show a lead for the dendritic neuron model. Results from the first presented model give us a base to conclude that artificial neurons utilizing dendrites have an inherent ability to learn data, which leads to them performing as well as or even better than the perceptron neuron model. This is important, as it could have been that the addition of the dendrites botched the learning capabilities of the neural networks. Results from the second presented model show that artificial neurons utilizing dendrites are not just some morphology of perceptron put under another name. The dendritic neurons once again outperform the perceptrons. Results also show that the architectures that use dendritic neurons learn information in a better manner than those that use perceptrons.

The area of research of artificial neuron models different from the perceptron is still in very early stages and the branch of artificial neurons utilizing dendrites is in even earlier stages. Literature on the subject is scarce and in turn few experiments have been carried out by researchers. This paper dives in the field and produces very interesting and promising results but further testing and experimenting are still crucial both on the proposed models and into other models.

The research team is working on different paths in searching for new architectures: dendritic computing, spiking computing, connectome and network analysis, and computational topology.

References

1. N. Spruston, G. Stuart, M. Häusser, G. Stuart, N. Spruston, M. Häusser, Eds. Principles of Dendritic Integration. – In: Dendrites. Oxford, Oxford University Press, 2016, pp. 351-398.
2. Richards, S. E. V., S. D. Van Hooser. Neural Architecture: From Cells to Circuits. – Journal of Neurophysiology, Vol. **120**, 2018, No 2, pp. 854-866.
3. Spruston, N. Pyramidal Neurons: Dendritic Structure and Synaptic Integration. – Nat. Rev. Neurosci., Vol. **9**, 2008, pp. 206-221.
4. Zhang, J. Basic Neural Units of the Brain: Neurons, Synapses and Action Potential. – arXiv preprint arXiv:1906.01703, 2019.
5. Herculano-Houzel, S. The Remarkable, Yet Not Extraordinary, Human Brain as a Scaled-Up Primate Brain and Its Associated Cost. – Proceedings of the National Academy of Sciences, Vol. **109**, 2012, pp. 10661-10668.

6. C h a v l i s, S., P. P o i r a z i. Drawing Inspiration from Biological Dendrites to Empower Artificial Neural Networks. – Current Opinion in Neurobiology, Vol. **70**, 2021, pp. 1-10.
7. M c C u l l o c h, W. S., W. P i t t s. A Logical Calculus of the Ideas Immanent in Nervous Activity. – Bulletin of Mathematical Biophysics, Vol. **5**, 1943, pp. 115-133.
8. R o s e n b l a t t, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. – Psychological Review, Vol. **65**, 1958, pp. 386-408.
9. Y a n l i n g, Z., D. B i m i n, W. Z h a n r o n g. Analysis and Study of Perceptron to Solve XOR Problem. – In: Proc. of 2nd International Workshop on Autonomous Decentralized System, 2002, pp. 168-173.
10. G. Stuart, N. Spruston, M. Häusser, Eds. Dendrites. Oxford, Oxford University Press, 2016.
11. W a n g, Z., S. G a o, J. W a n g, H. Y a n g, Y. T o d o. A Dendritic Neuron Model with Adaptive Synapses Trained by Differential Evolution Algorithm. – Computational Intelligence and Neuroscience, Vol. **e2710561**, 2020.
12. J i, J., S. G a o, J. C h e n g, Z. T a n g, Y. T o d o. An Approximate Logic Neuron Model with a Dendritic Structure. – Neurocomputing, Vol. **173**, 2016, pp. 1775-1783.
13. M a r c i e, J. T., E. von H i p p e l. The Situated Nature of Adaptive Learning in Organizations. – Organization Science, Vol. **8**, 1997, No 1, pp. 71-83.
14. X i a o, H., R. K a s h i f, V. R o l a n d. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. – arXiv preprint arXiv:1708.07747, 2017.
15. F i s h e r, R. A. The Use of Multiple Measurements in Taxonomic Problems. – Annals of Human Genetics, Vol. **7**, 1936, pp. 179-188.
16. W o l b e r g, W. H., W. N. S t r e e t, O. L. M a n g a s a r i a n. Breast Cancer Wisconsin (Diagnostic) Data Set. – UCI Machine Learning Repository, 1992.
17. K r i z h e v s k y, A., G. H i n t o n. Learning Multiple Layers of Features from Tiny Images. – University of Toronto, 2009.
18. D o n, S., D. C h u n g, K. R e v a t h y, E. C h o i, D. M i n. A New Approach for Mammogram Image Classification Using Fractal Properties. – Cybernetics and Information Technologies, Vol. **12**, 2013, No 2, pp. 69-83.

Received: 14.06.2023; Second Version: 02.08.2023; Third Verion: 07.08.2023; Accepted: 22.08.2023 (fast track)