International Journal of Combinatorial Optimization Problems and Informatics, 12(3), Sep-Dec 2021, 8-19. ISSN: 2007-1558.

www.editada.org

A Comparative Study of Dendrite Neural Networks for Pattern Classification

Rodrigo Román-Godínez[1], Erik Zamora[1], and Humberto Sossa[1,2]

[1] Instituto Politécnico Nacional - CIC, Av. Juan de Dios Batiz S/N, Gustavo A. Madero, 07738 México City, México.
[2] Tecnológico de Monterrey, Campus Guadalajara. Av. Gral. Ramón Corona 2514 Zapopan, Jalisco. 45138, México.
rodrigo0045@gmail.com, ezamorag@ipn.mx, hsossa@cic.ipn.mx

Abstract. Dendrite neurons are an alternative for classification tasks, providing competitive results when compared to typical classification methods. Dendrite networks allow each dendrite to build a close boundary to assign each incoming pattern $\mathrm{x_i} = (\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_n})^{\mathrm{T}}$ to its respective class. Hyperboxes, hyperellipsoids and hyperspheres are novel ways for dendrite computing. In this research we test these models and some hybrid variances trained by stochastic gradient descent. Results show that hyperellipsoidal neurons work well as classifiers with low-dimensional tasks, while hyperspherical neurons score better than the others in the case of image processing. However, when hybridizing, hyperboxes show poor results but hyperellipsoid and hyperspheres obtain even better results than two layer prerceptrons for many datasets.

Keywords: Multilayer perceptron · Dendrite Morphological Neurons · Dendrite Ellipsoidal Neuron · Dendrite Spherical Neuron

Article Info
Received April 16, 2021.
Accepted June 2, 2021

1 Introduction

Many techniques for solving classification problems have been reported in literature. Among them are: classification trees, logistic regression, discriminant analysis, neural networks, random forests, nearest neighbors, support vector machines, and so on. The main goal is to associate an input pattern $\mathrm{x_i} = (\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_n})^{\mathrm{T}}$ to their corresponding class $C_i = \{\mathrm{C_1}, \mathrm{C_2}, \ldots, \mathrm{C_n}\}$ by means of a mathematical function $f: x_i \rightarrow C_i$.

Different kinds of artificial neural networks (ANN) have been used for classification problems. The perceptron [1, 30] is used since many years ago to classify patterns by dividing the feature space into hyperplanes. Stacking perceptrons, layer after layer, known as a multi-layer perceptron (MLP) can be used to approximate any function [2]. Training of this kind of machines is often based on gradient descent and back-propagation [3].

Dendrite neural networks [4, 18, 17] are an alternative approach to separate input data by employing closed boundaries such as hyperboxes, hyperellipsoids, and hyperspheres. These specialized ANN tend to solve non-convex problems without hidden layers [7].

The main purpose of this research is to evaluate the performance of three different dendrite neurons: Dendrite Morphological Neurons (DMN), Dendrite Ellipsoidal Neurons (DEN), and Dendrite Spherical Neurons (DSN) as classifiers trained by stochastic gradient descent (SGD) with random initialization.

The rest of the paper is organized as follows. Section 2 provides a summary of the works that use these dendrite neural networks. Section 3 describes the architectures of DMN, DEN and DSN chronologically. Section 4 discusses the experimental results. Finally, in Section 5, we present our conclusions and provide recommendations.

2 Related Work

In this section, we present different neural networks with max, min operation and closed decision boundaries like: stand alone DMN, DEN and DSN; hybrid architectures morphological/rank/neural network, linear morphological neural network and morphological linear neural network, that have been proposed in the literature over the last years and some of their applications.

Among ANN architectures, we focus on those that make use of dendritic processing such as morphological neurons [4,5] and [8] models that are based on lattice algebra [6] as an alternative to the classical perceptron.

Lattice-based models combine operations such as max, min, addition, and subtraction. Furthermore, there are many hybrid architectures that merge morphological neurons with linear units as described in [19] by Hernandez et al.. In this paper, authors first use a layer of a DMN as a middle layer connected to an MLP, they then invert the order. Moreover, Pessoas and Margos [15] present a morphological/rank/linear network, while Araújo proposes a hybrid morphological-linear perceptron [16].

As an alternative to DMN, Arce et al. [18] proposed changing the form of the hyperbox to obtain a smoother decision boundary by using the hyperellipsoid, authors use well-known Mahalanobis distance to determine if an input pattern belongs to a certain class or not. Each hyperellipsoid is generated by one dendrite that uses a centroid and covariance matrix. Arce et al. performed some tests for solving different classification problems, their results reported competitive when compared to other state-of-the-art classifiers [29] and [18].

Inspired by DEN, in [17], authors simplified the hyperellipsoid to attain a hypersphere to create a model called Dendrite Spherical Neuron (DSN). In this case, the covariance matrix is replaced by a radius with which the number of the parameters for dendrites can be reduced. Apart from an MLP, dendritic neurons present competitive results in pattern classification in different applications, some of which are used for motor task recognition from electroencephalographic signals [31], 3D object recognition [32], retinal vessel extraction [34], and Parkinson detection of speech signals [35] and as a facial detector for determine asymmetry level in cleft lip children [33]. Other applications can be found in [12,13,16,15]. Formerly, dendrite neurons were initialized by different methods such as HpC, dHpC, D&C, k-means, hill-climbing algorithm, simulated annealing, among others [17,20,21,22,24,10,11] and various other training methods [20,21,11,23].

In this paper, neurons are trained by SGD with random initialization for evaluating which of these neurons perform better under these conditions; thus, it can be also used in future research works.

3 Neural Models and Methods

In this section, we present a brief description of the three dendritic neural models and their decision boundaries: Dendrite Morphological Neuron (DMN) which classifies an input pattern by enclosing patterns inside of a hyperbox, Dendrite Ellipsoidal Neuron (DEN) which classifies an input pattern by enclosing patterns inside of a hyperellipsoid and Dendrite Spherical Neuron (DSN) which classifies an input pattern by enclosing patterns inside of a hypersphere. Additionaly, we describe a multi-layer perceptron.

3.1 Dendrite Morphological Neuron

Dendrite Morphological Neuron (DMN) segments the input space into hyperboxes of N dimensions where $N \in I^{+}$. After processing the data, the output $d_c(x)$ is given by the following equations:

$$d_c(x) = \underset{c}{\operatorname{argmax}}\left(h_{k,c}\,(x)\right) \tag{1}$$

$$h_{k,c}(x) = \min_{2}\left(\min_{N}(x_i - w_{min}\,, w_{max}\ - x_i)\ \right), \qquad i = \{0,1,2, \dots n\} \tag{2}$$

Here, $h_{k,c}(x)$ is the output of a dendrite, k indicates a specific dendrite and $k \in I^{+}$, c represents the class and $c \in I^{+}$, x is an input vector and $x \in N$, w_{min} and w_{max} are the weight vectors that represent the opposite vertices of the hyperbox, both $w_{min}, w_{max} \in N$. The inner min operator in Equation (2) gets the minimum value of the vectors of $x - w_{min}$ and gets the minimum value of the vectors of $w_{max} - x$; the outer min gets the minimum value of the pair values gets from de inner min. Whether an input x belongs to the class or not, If $h_{k,c} > 0$, then the input is inside the hyperbox. If $h_{k,c} = 0$, it is over the hyperbox boundary, otherwise, it does not belong to the class as it shows Figure 1.

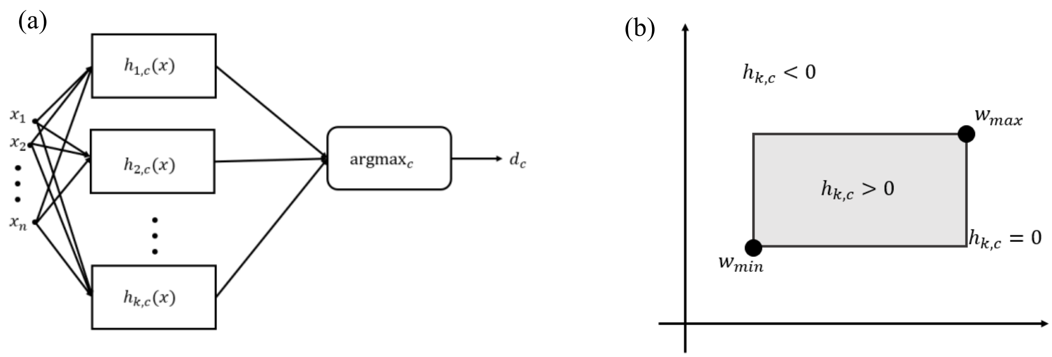

Figure 1. (a) Architecture of Dendrite Morphological Neuron. (b) Example of an hyperbox in 2D.

3.2 Dendrite Ellipsoidal Neuron

DEN architecture helps classifying patterns by enclosing the data into a close decision boundary, it differs from DMN, hyperboxes are substituted by hyperellipsoids. This neuron uses the Mahalanobis distance to determine if the output τ_j belongs to a class or not. For this, refer to the following equations:

$$\tau_j \ = \ argmin_k(\tau_j^k) \tag{3}$$

$$\tau_k^j = (x_i - \mu_k)^T \ \sum_{k}^{-1} (x_i - \mu_k) \tag{4}$$

Here, τ_j^k is the output of a dendrite, μ_k is a mean vector, and $\sum_k$ is a covariance matrix and $\sum_k^{-1}$ is the inverse of the covariance matrix associated with the k-th cluster, $K = 1, \dots, k$, and x_i is the input pattern. Dendrites in DEN, in this case, measure the distance between patterns to hyperellipsoids. A pattern is assigned to the class whose dendrite output outputs the minimum value. If $h_{k,c} < 0$, then the input is inside the hyperbox; if $h_{k,c} = 0$, it is over the hyperbox boundary. Otherwise, it is declared to be outside of the class as it shows Figure 2.

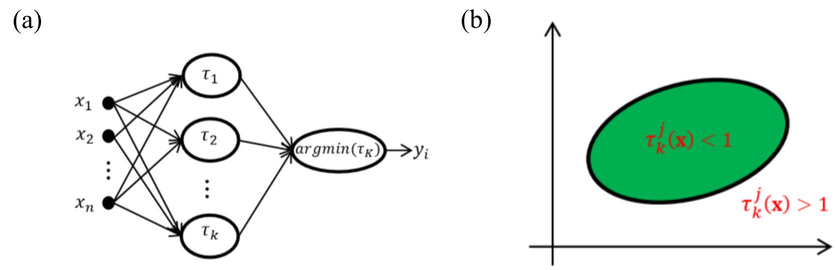

Figure 2. (a) Architecture of Dendrite Ellipsoidal Neuron. (b) Example of an hyperellipsoid in 2D

3.3 Dendrite Spherical Neuron

DSN is a simplification of DEN with respect to the decision boundary generated. In this case, a DSN forms hyperspheres for data classification. To do so, it compares the distance of every input data to the center of the hypersphere against its radius, by means of the following equations:

$$d_j(x) = max\left(h_{i,j}(x)\right), i = 1, \dots l_j \tag{5}$$

$$h_{i,j}(x) = r_{i,j} - \left\|x - c_{i,j}\right\|^2 \tag{6}$$

Here, $\|*\|$ is the Euclidean norm. $c_{i,j} \in \mathbb{R}^D$ is the centroid of the dendrite, and r is the radius. $h_{i,j}(x)$ represents the response of the dendrite. If $h_{i,j}(x) < 0$ we say that the input does not belong to the class. However, a number greater than zero is obtained, it means that the pattern is inside the hypersphere; thus, we say that this patterns belongs to the class. In the third scenario is when the pattern is on the boundary, thus $h_{i,j}(x) = 0$. Moreover, $h_{i,j}(x)$ is the output of the i-th dendrite for the j-th class; this value goes into a max function to get d_j that is the output of the j-th dendrite cluster as it shows Figure 3.

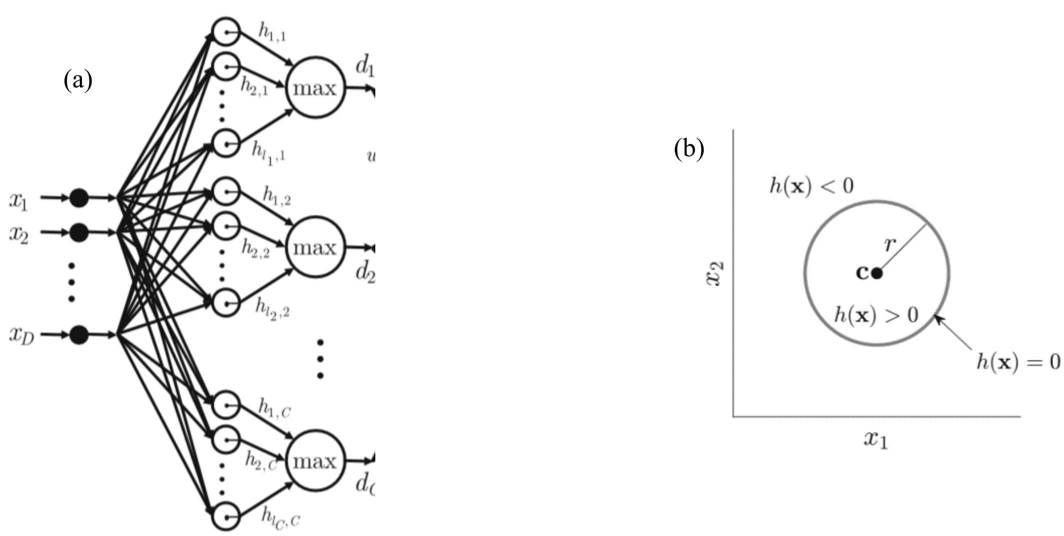

Figure 3. (a) Architecture of Dendrite Spherical Neuron. (b) Example of an hyperspheres in 2D

3.4 Multi-Layer Perceptron

An MLP is a widely studied model for solving multiple classification problems. The MLP architecture is composed by an input layer, with one or more neurons in parallel, followed by one or more hidden layers in the middle composed of perceptrons, and an output layer. The output of each layer y_m is computed as follows:

$$y_m = f(W_m y_{m-1} + b_m) \tag{7}$$

Here, y_m is the output vector of the m layer for $m = 1,2,3 \ldots M$, W_m is the weight matrix, b_m is the bias vector, and f is an activation function that could be a sigmoid, a tanh, a ReLU, among other functions.

4 Experimental Results

In this section, we present different experiments to test the performance of dendrite neurons with random initialization and SGD training instead of the different training methods proposed in literature like HpC, DHpC, D&C, and k-means, we tested both stand alone units and hybrid units for low dimensionality problems with a short training dataset (UCI real problem datasets) and high dimensionality problems with large training dataset (Image datasets: Fashion-Mnist, CIFAR10 and CIFAR100).

Four classification experiments have been used. The first two were taken from the UCI datasets; the other two are the sets of images used by ResNet50 architecture for feature extraction. The aim of the experiments is to evaluate the performance of the described dendrite neurons as classifiers and the hybrid architecture with a hidden dendrite layer and a perceptron layer as output.

The experiments were performed on a low-dimensionality problem and image datasets. Each neuron was implemented as a layer with Keras library [27] in Python. Every architecture was trained by means of gradient descent with a randomized initialization of the weights. Also as a reference, all proposals were compared with a two-layer perceptron (TLP) and a single layer perceptron (P) as benchmarks to maintain the same depth of the classifier. The hyper-parameters and a set of learning parameters that have been used to perform the test and reported in this paper were the ones to present better results among different chosen manual configurations.

We performed manual and iterative tuning for each hyperparameter for every model and every dataset. The activation function was chosen from: tanh, ReLU and Sigmoid; the learning rate was chosen from: 0.008, 0.005 and 0.001 and the number of dendrites varies from five to five to reach a hundred. We fixed the values of epoch in 100, with an adam optimizer and alternate between binary cross-entropy and categorical cross-entropy.

4.1 Datasets

In order to test and measure the classification performance of the three different dendrite neurons, besides the TLP and a simple perceptron layer (all of them have been described in section 3), a total of 10 real-world datasets obtained from the Machine Learning Repository of the UCI [28] were considered for the first and second experiments. The characteristics of these 10 databases are listed in Table 1. Besides, three image datasets for image classification: Fashion-MNIST [25] -contains 60,000 training images and 10,000 test images of fashion and clothing items, taken from 10 classes. Each image is a standardized 28×28 size in grayscale-, CIFAR10 -contains 60,000 training images and 10,000 test images of 10 different classes Each images is standardized 32x32 in color.- and CIFAR100 [26] CIFAR10 -contains 50,000 training images and 10,000 test images of 100 different classes Each images is standardized 32x32 in color.- were used in experiments number three and four were also used.

Table 1. Real-world datasets with distinct number of classes (C) and dimensionality (D). The number of patterns is N.

Dataset	*N*	*C*	*D*
Breast cancer Wisconsin	569	2	30
Glass identification	214	6	10
Heart disease Cleveland	297	2	13
Hepatitis	112	2	18
Iris data	150	3	4
Page blocks	5409	5	10
Pima Indians diabetes	768	2	8
Seeds	199	3	7
Thyroid gland data	215	3	5
Wine recognition data	178	3	13

4.2 Experiment 1

The proposed architecture for the first experiment is a two-layer network composed by a TLP, two-layer DMN (TLDMN), two-layer DEN (TLDEN) and two-layer DSN (TLDSN). Due to the low number of data the accuracy percentage reported in Table 2 was calculated by appliying ten-fold cross validation, leaving one different fold out for testing while 9 folds are used to train. Finally, we calculate an average of the ten different test segments. As can be seen, the TLP architecture outperforms the other morphological units in every dataset, except for one where TLDEN overcomes it by almost three percentage points. However, when comparing both, it seems that TLDEN shows a better classification performance when SGD is used training. According to the mean of each model, TLP persists as the better-performing model by more than 20 percentage points with respect to TLDEN. Besides, either TLP or TLDEN shows good consistency with almost five percent of standard deviation. As stated by the data in Table 2, none of the morphological units trained by SGD is comparable with TLP in terms of accuracy, except DEN at page block dataset experiment where the latter was superior by two percentage points. Besides, as can be appreciated, (DEN) provided the best performance among dendritic-based neurons. We performed a t-test to validate the statistical significance of results obtained in each dataset. In most of the cases TLP and TLDEN shows p-values<0.05 with respect to the other classifiers, while TLDMN and TLDSN shows p-values>0.05. Table 2 (c) shows the results for the Breast Cancer Wisconsin dataset.

Table 2. (a)Accuracy of experimental results for TLP, TLDMN, TLDEN TLDSN classifier with cross validation $k = 10$. The best results are highlighted in bold. (b) Number of dendrites for each model. (c) T-tes results for the Breast Cancer Wisconsin dataset.

(a)

Dataset	**TLP %**		**TLDMN %**		**TLDEN %**		**TLDSN %**	
Breast cancer Wisconsin	**96.66**	**±2.27**	74.80	±8. 70	79.08	±3.70	62.76	±7.31
Glass identification	**79.43**	**±5.78**	35.58	±6.02	44.44	± 8. 36	35.52	±12.10
Heart disease Cleveland	**88.13**	**±4.67**	54.50	±5.97	76.88	± 4.09	54.51	±8.84
Hepatitis	90.95	±8.60	81.82	±8.40	**95.07**	**±3.62**	81.82	±11.43
Iris data	**98.00**	**±3.00**	58.60	±11.80	62.66	± 8.00	96.66	±3.33
Page blocks	93.45	±2.34	89.76	±1.11	**95.66**	**± 1.08**	89.76	±1.04
Pima Indians diabetes	**76.29**	**±4.78**	65.10	±3.70	67.05	± 3.93	65.11	±5.88
Seeds	**91.42**	**±4.10**	22.85	±6.31	30.00	± 5.23	29.04	±6.88
Thyroid gland data	**98.59**	**±2.14**	69.84	±10.73	84.69	± 2.71	70.00	±12.45
Wine recognition data	**91.47**	**±9.90**	33.00	±11.72	51.07	±7.69	40.09	±12.81
Mean	**90.43**	**±4.75**	58.57	±7.36	68.66	±4.84	62.52	±8.20

(b)

	MLP	MNN	DEN	DSN
Breast cancer Wisconsin	64	30	80	40
Glass identification	100	5	95	50
Heart disease Cleveland	15	90	95	90
Hepatitis	100	75	95	80
Iris data	10	80	35	80
Page blocks	25	40	85	40
Pima Indians diabetes	50	10	100	95
Seeds	60	40	95	35
Thyroid gland data	30	100	30	35
Wine recognition data	20	20	95	35

(c)

	MLP	MNN	DEN	DSN
MLP	1	4.56e-09	5.87e-15	1.07e-07
MNN	4.56e-09	1	3.29e-07	0.996
DEN	5.87e-15	3.29e-07	1	2.98e-06
DSN	1.07e-07	0.99	2.98e-06	1

4.3 Experiment 2

For the second experiment reported in Tables 3 and 4, we introduce and evaluate three hybrid models with two layers. First in Table 3 the hidden layer of morphological with DMN, DEN, and DSN architectures for each model, and as output layer a classical perceptron layer for classification. In the other hand, in Table 4 the hidden layer has perceptrons while the morphological units form the output layer. As in the first experiment, performance of the different proposals were measured by ten-fold cross-validation leaving one different segment out for testing while 9 segmets are used to train, finally we calculate an average of the ten different test segments. The same data were used to evaluate the hybrid architectures We performed a t-test to validate the statistical significance of results obtained in each dataset. In most of the results of the t-test does not shows significant statistical difference according to the values of p-value>0.05. Table 3 (c) shows the results for the Pima Indians diabetes dataset.

Table 3. Accuracy of experimental results for hybrid classifiers with DMN, DEN and DSN as middle layer and perceptron P as output layer with cross validation $k = 10$. The best results are highlighted in bold. (b) Number of dendrites for each model. (c) T-tes results for the Pima Indians diabetes dataset.

(a)

Dataset	DMN-P %		DEN-P %		DSN-P %	
Breast cancer Wisconsin	**94.90**	**±3.88**	90.68	±4.00	62.72	±4.96
Glass identification	36.50	±6.37	**41.11**	**±5.77**	35.99	±3.85
Heart disease Cleveland	57.40	±3.39	**69.64**	**±5.79**	54.43	±9.71
Hepatitis	81.79	±7.17	**91.55**	**±1.70**	81.84	±10.44
Iris data	96.00	±7.99	65.33	±9.79	**97.33**	**±3.26**
Page blocks	91.60	±1.44	**95.19**	**±1.20**	89.76	±0.92
Pima Indians diabetes	**65.11**	**±5.96**	62.23	±5.01	65.11	±3.62
Seeds	**30.42**	**±10.44**	26.66	±7.73	26.19	±6.19
Thyroid gland data	69.71	±9.20	**80.40**	**±6.90**	69.82	±13.39
Wine recognition data	39.80	±8.50	**62.32**	**±13.90**	39.90	±10.40
Mean	66.32	±6.43	**68.51**	**±6.17**	62.30	±6.67

(b)

	DNN_P	DEN_P	DSN_P
Breast cancer Wisconsin	10	85	70
Glass identification	85	65	60
Heart disease Cleveland	20	100	45
Hepatitis	45	95	20
Iris data	45	10	90
Page blocks	40	75	75
Pima Indians diabetes	70	75	45
Seeds	30	75	60
Thyroid gland data	15	30	5
Wine recognition data	60	65	50

(c)

	MNN_P	DEN_P	DSN_P
MNN_P	1	0.08	0.99
DEN_P	0.08	1	0.08
DSN_P	0.99	0.08	1

Table 3 shows an increment in some of the accuracy percentages. Besides, it seems that they have a competitive score between them but still below the percentage of TLP. Again, DEN overcomes DMN and DSN in six of the datasets, the performance is close for every dataset with slightly better performance score. As can be appreciated, when using morphological units in the output layers, the classifiers reach competitive results whit some scores better than a TLP.

On the other hand, as can seen from Table 4, the best result was obtained with a layer of perceptrons is used as a hiden layer and DEN is used as the output. We performed a t-test to validate the statistical significance of results obtained in each dataset. In most of the results of the t-test shows significant statistical difference according to the values of p-value<0.05. Table 3 (c) shows the results for the Glass identification dataset.

Table 4. Accuracy of experimental results for hybrid classifiers with a layer of perceptrons P in the middle layer and DMN, DEN or DSN as output layer with cross validation $k = 10$. The best results are highlighted in bold. (b) Number of dendrites for each model. (c) T-tes results for the Glass identification dataset.

(a)

Dataset	**P-DMN %**		**P- DEN %**		**P- DSN %**	
Breast cancer Wisconsin	97.01	±1.76	**98.42**	**±1.65**	97.32	±2.74
Glass identification	73.36	±7.28	70.56	±4.97	**87.87**	**±3.01**
Heart disease Cleveland	86.79	±5.58	89.10	±2.31	**89.44**	**±4.08**
Hepatitis	88.71	±4.76	95.73	±2.67	**96.50**	**±2.18**
Iris data	97.99	±2.66	**98.66**	**±1.63**	97.26	±1.13
Page blocks	94.60	±0.32	95.06	±0.86	**96.12**	**±0.30**
Pima Indians diabetes	76.04	±3.80	**97.13**	**±1.39**	83.33	±2.36
Seeds	92.85	±3.01	**95.71**	**±0.95**	93.33	±3.15
Thyroid gland data	98.13	±0.93	**100.00**	**±0.00**	99.06	±1.13
Wine recognition data	71.36	±5.87	**97.19**	**± 3.04**	94.93	±2.13
Mean	87.68	±3.59	**93.75**	±1.94	93.65	±2.27

(b)

	P-MNN	P-DEN	P-DSN
Breast cancer Wisconsin	60	70	35
Glass identification	95	70	85
Heart disease Cleveland	90	85	90
Hepatitis	90	40	20
Iris data	20	5	5
Page blocks	30	95	100
Pima Indians diabetes	70	85	95
Seeds	35	5	35
Thyroid gland data	80	10	40
Wine recognition data	95	85	60

(c)

	P_MNN	P_DEN	P_DSN
P_MNN	1	6.87e-08	0.01
P_DEN	6.87e-08	1	1.27e-10
P_DSN	0.01	1.27e-10	1

4.4 Experiment 3

The third experiment consisted of a feature extraction stage, which is performed by a ResNet-50 plus a dendrite classifier, whereas, in the first experiments, the TLP model was clearly dominant, this time, the best classification score was reached by DSN with a 91.49%, 75.22% and 41.47% for Fashion-MNIST, CIFAR10, and CIFAR100 datasets, respectively as depicted in Table 5.

It is worth mentioning that for the DEN classifier, it was necessary to reduce the dimensionality of the feature vector due to the requirement of a high number of parameters. Thus, we proposed two different options: firstly, by using an extra convolutional layer, and secondly, by using a perceptron layer. In the both cases, the accuracy was competitive for Fashion-MNIST and CIFAR10, but not for CIFAR100, where the score is far below the results reported for DSN.

Table 5. Accuracy of experimental results over image datasets with ResNet-50 as feature extraction and perceptron (P), morphological neuron (DMN), ellipsoidal neuron (DEN) and spherical neuron (DSN) as classifier. The best results are highlighted in bold.

	Fashion-MNIST %	CIFAR-10 %	CIFAR-100 %	Mean %
ResNet-50 + P	90.90	73.19	37.43	67.17
ResNet-50 + DMN	90.71	72.04	30.18	64.31
ResNet-50 + conv + DEN	90.89	72.74	27.21	63.61
ResNet-50 + P + DEN	91.43	72.32	36.48	66.74
ResNet-50 + DSN	**91.49**	**75.22**	**41.47**	**69.39**

4.5 Experiment 4

The last experiment uses the same datasets: Fashion-MNIST, CIFAR10 and CIFAR100. In this experiment, feature extraction is performed again with ResNet-50 and then the same hybrid architecture used in experiment 2. The results depicted in Table 6 show that DEN and DSN perform poorly when classifying images, whereas DMN outperforms by far them by obtaining an accuracy scored of 91.13% for Fashion-MNIST, 71.39% for CIFAR10, and the lowest score for CIFAR100 at nearly 23.45%. The first two of these results are competitive with the other accuracy scores reported in this paper; nonetheless, for CIFAR100, the score is more than 10 points below most of the scores, as shown in Table 5. In general when using morphological units as final layer for image classification shows better performance than its hybrid counter part, with the unit formed by perceptron and dendritic spherical neuron with the better scores for this datasets.

Finally, Table 7 the results obtained when ResNet-50 is combined with layer of perceptrons and either DMN, DEN or DSN. As can be appreciated from this table, ResNet-50+P+DSN provides the best performance.

Table 6. Accuracy of experimental results over image datasets with ResNet-50 as feature extraction and hybrid architecture using morphological neuron (DMN), ellipsoidal neuron (DEN) and spherical neuron (DSN) as middle layer and perceptron (P) as output layer. The best results are highlighted in bold.

	Fashion-MNIST %	**CIFAR-10 %**	**CIFAR-100 %**	**Mean %**
ResNet-50+DMN+P	**91.13**	**71.39**	**23.45**	**61.99**
ResNet-50+DEN+P	28.43	18.00	2.40	16.27
ResNet-50+DSN+P	10.00	10.00	1.00	7.00

Table 7. Accuracy of experimental results over image datasets with ResNet-50 as feature extraction and hybrid architecture using morphological neuron (DMN), ellipsoidal neuron (DEN) and spherical neuron (DSN) as final layer and perceptron (P) as hidden layer. The best results are highlighted in bold.

	Fashion-MNIST %	**CIFAR-10 %**	**CIFAR-100 %**	**Mean %**
ResNet-50+P+DMN	89.10	69.88	33.46	64.14
ResNet-50+P+DEN	84.81	72.25	35.91	64.32
ResNet-50+P+DSN	**91.29**	**75.22**	**41.04**	**69.18**

5 Conclusions and Directions for Further Research

We presented an experimental study where different dendritic processing based ANN such as DMN, DEN, and DSN for pattern classification, when trained by means of SGD. Furthermore, we proposed and tested four hybrid configurations plus others two described in [16] using real-world datasets and images. Standalone dendrite neurons trained by SGD seem to perform poorly on low dimensional data, although DSN obtained the best scores for image classification. On the other hand, hybrid architectures tend to provide competitive results with MLP-based architectures when morphological neurons are at the final layer.

DEN based architectures tend to showcase a reasonably good performance in solving low dimensionality problems due to the smooth edges, unlike DMN, and has two axes of different measures, unlike DSN.

Notably, this is not true when tested for images where feature extraction is performed by a deep model such as ResNet-50 where dendritic neurons are in the hidden layer and do not generate competitive results. On the other hand, when using morphological units as output layer, hybrid units provide competitive scores for image classification. In conclusion, hybrid neurons seem to be an alternative for classification task with morphological neurons as output layer, but not when they are used as the hidden layer. The difference in the performance could be because of the characteristics of the input data on the classifier stage, even though the first two experiments have lower dimensionality, ResNet-50 have proved to obtain better input data for the classifiers.

Future works should involve an extended study of dendrite ANN with different initialization algorithms, also experiments as a feature extractor on deep models to understand what kind of visual features can be acquired from the images. Moreover, the study must be extended with other new architectures such as DMN, as proposed in [36], and hybrid morphological/linear perceptron trained by extreme learning machine, as described in [9]. Both mentioned that SGD training is feasible.

6 Acknowledgements

H. Sossa and E. Zamora would like thank the Instituto Politécnico Nacional and CONACYT for the economic support to carry out this research under projects: SIP 20200630, SIP 20210788, SIP 20210316, CONACYT 65 (Fronteras de la Ciencia) and CONACYT FORDECYT PRONACES 6005. R. Román-Godínez acknowledges CONACYT for the scholarship granted to him for his postgraduate studies.

References

1. Rosenblatt F. (1957). "The Perceptron—a perceiving and recognizing automaton", Cornell Aeronautical Laboratory, Report 85-460-1.
2. Moshe L. et al. (1993). "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function", Neural Networks, 6(6):861–867.
3. Rich C et al. (2000). "Overfitting in neural nets: Back propagation, conjugate gradient, and early stopping", In Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS'00), MIT Press, Cambridge, MA, USA, pp. 386–387.
4. Ritter GX et al. (2003). "Morphological perceptrons with dendritic structure." ,In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, Volume 2, 1296—1301, May 2003.
5. Ritter GX and Sussner P (1996). "An introduction to morphological neural networks.", In: Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 1996, 709—717.
6. Ritter GX and Urcid G (2003). "Lattice algebra approach to single-neuron computation", in IEEE Transactions on Neural Networks 14(2):282-295.
7. Ritter GX and Urcid G (2007). "Learning in Lattice Neural Networks that Employ Dendritic Computing.", In: Kaburlasos V.G., Ritter G.X. (eds) Computational Intelligence Based on Lattice Theory. Studies in Computational Intelligence 2007, vol 67. Springer, Berlin, Heidelberg.
8. Sussner P (1998). "Morphological perceptron learning.", in: Proceedings of the IEEE International Symposium on Intelligent Control, Gaithersburg, MD, 1998, pp. 477–482.
9. Sussner P and Campiotti I (2020). "Extreme learning machine for a new hybrid morphological/linear perceptron.", Neural Networks 123:288-298.
10. Sussner P and Esmi E (2011). "Morphological perceptrons with competitive learning: Lattice theoretical framework and constructive learning algorithm.", Information Sciences 181(10):1929– 1950.
11. Sussner P and Esmi E (2009). "Introduction to morphological perceptrons with competitive learning.", in: Proceedings of the International Joint Conference on Neural Networks, Atlanta, GA, 2009, pp. 3024–3031.
12. Araújo RA et al. (2017). "A morphological neural network for binary classification problems.", Engineering Applications of Artificial Intelligence 65:12 – 28.
13. Araújo RA (2012). "A morphological perceptron with gradient-based learning for Brazilian stock market forecasting", Neural Networks 28:61–81.
14. Araújo RA and Sussner P (2010). "An increasing hybrid morphological-linear perceptron with pseudo gradient-based learning and phase adjustment for financial time series prediction.", in: Proceed ings of the 2010 IEEE World Congress on Computational Intelligence, Vol. IJCNN, Barcelona, Spain, 2010, pp. 807–814.
15. Pessoa LFC and Maragos P (2000). "Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition.", Pattern Recognition 33:945–960.
16. Araújo RA and Sussner P (2010). "An increasing hybrid morphological-linear perceptron with pseudo gradient-based learning and phase adjustment for financial time series prediction.", in: Proceed ings of the 2010 IEEE World Congress on Computational Intelligence, Vol. IJCNN, Barcelona, Spain, 2010, pp. 807–814
17. Gómez-Flores W and Sossa JH (2020), "Towards Dendrite Spherical Neurons for Pattern Classification", In: Figueroa Mora K., Anzurez Marín J., Cerda J., Carrasco-Ochoa J., Martínez-Trinidad J., Olvera-López J. (eds) Pattern Recognition, MCPR 2020. Lecture Notes in Computer Science, vol 12088. Springer, Cham.
18. Arce F et al. (2017). "Dendrite Ellipsoidal Neuron", In: 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 795-802.

19. Hernández G et al. (2018). “Morphological-Linear Neural Network”, 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, 2018, pp. 1-6,
20. Zamora E and Sossa JH (2017). “Dendrite morphological neurons trained by stochastic gradient descent.”, Neurocomputing, Volume 260, 2017, Pages 420431.
21. Sossa JH and Guevara E (2014). “Efficient training for dendrite morphological neural networks.”, Neurocomputing, 131:132-142.
22. Arce F et al. (2018). “Dendrite ellipsoidal neurons based on k-means optimization.” Evolving Systems 1–16.
23. Arce F et al. (2018). “Differential evolution training algorithm for dendrite morphological neural”, networks, Applied Soft Computing 68:303-313.
24. Arthur D and Vassilvitskii S (2007). “K-means++: the advantages of careful seeding.”, In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia (2007)
25. Xiao H et al. (2017). “Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms.”: 2017, arXiv:1708.07747 (Preprint). https://github.com/zalandoresearch/fashion-mnist
26. Krizhevsky A and Hinton G (2009). “Learning multiple layers of features from tiny images.”, Technical Report (2009) https://www.cs.toronto.edu/ kriz/cifar.html
27. Chollet F et al..: Keras (2015). https://keras.io
28. Dua D and Graff C. UCI machine learning repository (2017). http://archive.ics.uci. edu/ml
29. Arce F et al. (2019). “Dendrite Ellipsoidal Neuron Trained by Stochastic Gradient Descent for Motor Imagery Classification.”, In: Carrasco-Ochoa J., Martínez-Trinidad J., Olvera-López J., Salas J. (eds) Pattern Recognition. MCPR 2019.: Lecture Notes in Computer Science, vol 11524. Springer, Cham.
30. Zhang GP (2000). “Neural networks for classification: a survey”, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 30, no. 4, pp. 451-462, Nov. 2000.
31. Antelis JM et al. (2018). “Dendrite morphological neural networks for motor task recognition from electroencephalographic signals.”, Biomedical Signal Processing and Control, Volume 44, 2018, Pages 12-24.
32. Sossa JH and Guevara E (2013) “Modified Dendrite Morphological Neural Network Applied to 3D Object Recognition.”, In: Carrasco-Ochoa J.A., Martínez-Trinidad J.F., Rodríguez J.S., di Baja G.S. (eds) Pattern Recognition. MCPR 2013. Lecture Notes in Computer Science, vol 7914. Springer, Berlin, Heidelberg.
33. Cortés G et al. (2019). “Asymmetry Level in Cleft Lip Children Using Dendrite Morphological Neural Network.”, In: Carrasco-Ochoa J., Martínez Trinidad J., Olvera-López J., Salas J. (eds) Pattern Recognition. MCPR 2019. Lecture Notes in Computer Science, vol 11524. Springer, Cham.
34. Vega R et al. (2015). “Retinal vessel extraction using Lattice Neural Networks with dendritic processing, Computers in Biology and Medicine.”, Volume 58, 2015, Pages 20-30, ISSN 0010-4825,
35. Gutierrez-Loaiza LD and Alfonso-Morales W (2020). “Morphological Neural Networks for Parkinson Detection through Speech Signals,” 2020 IEEE Colombian Conference on Applications of Computational Intelligence (IEEE ColCACI 2020), Cali, Colombia, 2020, pp. 1-6.
36. Gómez-Flores W and Sossa JH (2021). “Smooth dendrite morphological neurons.”, Neural Networks, Volume 136, 2021, Pages 40-53.