



Dendrite Ellipsoidal Neuron Trained by Stochastic Gradient Descent for Motor Imagery Classification

Fernando Arce¹✉, Omar Mendoza-Montoya², Erik Zamora¹,
Javier M. Antelis², Humberto Sossa^{1,2}, Jessica Cantillo-Negrete³,
Ruben I. Carino-Escobar³, Luis G. Hernández², and Luis Eduardo Falcón²

¹ Instituto Politécnico Nacional - CIC,

Av. Juan de Dios Batiz S/N, Gustavo A. Madero, 07738 Mexico City, Mexico
fernando.arce.vega@gmail.com, ezamorag@ipn.mx, hsossa@cic.ipn.mx

² Tecnológico de Monterrey Campus Guadalajara, Av. Gral Ramón Corona 2514,
45201 Zapopan, Jalisco, Mexico

{omendoza83,luis.g.hernandez,luis.eduardo.falcon}@tec.mx,
mauricio.antelis@itesm.mx

³ Division of Medical Engineering Research, Instituto Nacional de Rehabilitación,
14389 Mexico City, Mexico

{jcantillo,ricarino}@inr.gob.mx

Abstract. Dendrite ellipsoidal neurons are a novel and different alternative for classification tasks, giving competitive results compared with typical classification methods. Based on k-means++ algorithm, the network allows each dendrite to build a hyperellipsoidal in order to assign each incoming pattern $x_i = (x_1, x_2, \dots, x_n)^T$ to its respective C class. The main disadvantage of this training algorithm is the lack of accuracy in high dimensional datasets. In this research, we solved this problem by training the dendrite ellipsoidal neuron using stochastic gradient descent. Furthermore, electroencephalography data were acquired during two mental conditions (imaginary movements of the left and right hand) in order to test the new training algorithm. The proposed algorithm outperformed the accuracy acquired by a dendrite ellipsoidal neuron based on k-means++ obtaining 76.02% and 62.77%, respectively. Also, the algorithm was compared with multilayer perceptrons and support vector machines which are some of the most common classifiers used to detect motor-related information in brain signals. These achieved an accuracy of 72.38% and 65.81%, respectively.

Keywords: Dendrite Ellipsoidal Neuron · Motor Imagery ·
Electroencephalography · Multilayer perceptrons · Support vector
machines · Stochastic Gradient Descent · k-means++

1 Introduction

The main objective about pattern classification is to establish a mathematical function that associate input patterns $x_i = (x_1, x_2, \dots, x_n)^T$ to their

corresponding classes C^1, C^2, \dots, C^j . This assignation must be as strong as possible to reduce potential variations in incoming data and must be capable to find elemental relationships between patterns.

Since 60 years ago, many types of neural networks have been presented for solving classification tasks and the most common approach is a group of classical perceptrons which create hyperplanes to divide and associate data by using synaptic weights, biases and activation functions.

In a common neural network, each neuron can divide the input search space into two parts. Thereby, appending more neurons in a single layer, the network has the capacity to learn any complex function [1].

Other type of neural networks less known are Dendrite Morphological Neural Networks (DMNN) which separate data employing hyperboxes. These neurons group patterns using minimum or maximum operators to generate the piecewise boundaries for classification tasks. DMNN have the advantage of being easily implemented in logic devices.

This research proposes an improvement to a specific type of morphological neural networks called Dendrite Ellipsoidal Neuron (DEN) trained with k-means++ algorithm [2,3]. DEN has shown good performance in low dimensional datasets, requiring few training parameters and it is easy to implement in logic devices. Although DEN has shown to be efficient, it also has poor performance with high dimensional datasets. All these DEN advantages motivated us to explore new ideas in order to improve DEN accuracy in high dimensional datasets.

In this paper, we trained a DEN using Stochastic Gradient Descent (SGD) [4] implemented as a neural network layer of the Keras library [5] in Python. Furthermore, in order to test the proposed training algorithm in a high dimensional dataset, Electroencephalography (EEG) data were acquired from eight able-bodied subjects for classifying Motor Imagery (MI) of the hands into binary classes (Left vs Right). Contributions of this research are:

- This is the first time that a DEN is trained by SGD.
- Through a series of experiments, we show that the new training algorithm outperforms the actual DEN accuracy for our dataset, and the accuracy achieved by some of the most common classifiers for MI.

The rest of the paper is structured as follows: Sect. 2 provides a chronological list of publications related to previous literature with a comprehensive explanation. Section 3 describes the methods and materials used to obtain and characterize the EEG signals. Section 4 shows the DEN and the proposal architecture. Section 5 describes the general details of the used classifiers and the experimental results. In Sect. 6, we give our conclusions and future work.

2 Related Works

Morphological Neural Networks MNN were originally besought by Ritter and Davidson as a combination between neural networks and image algebra [6–9].

After this, Arbib published a book where it was taken into account that biological neurons process the information not only in the neuron cells but also in the dendrites [10]. Other related works resumed this research [11]. And several approaches have been proposed based on heuristics to manipulate hyperboxes not taking into account the dendrite fitness function.

All these techniques create hyperboxes to divide the input space into rectangular segments. In addition, in 2017 we presented DEN which changes the operations performed for MNN dendrites by using the Mahalanobis distance [12]. The main advantage of the ellipsoidal model is that it creates smoother decision boundaries and not rectangular regions.

Some other similar approaches to the k-means++ [13] clustering algorithm and the Mahalanobis distance [12] are elliptical k-means clustering algorithms, Gaussian Mixture Models (GMM), and classifiers based on the Mahalanobis distance.

Authors in [14, 15] employed elliptical k-means clustering algorithm to discriminate between human and nonhuman faces. For this, they altered k-means by modifying the normalized Mahalanobis distance to achieve six face pattern clusters.

GMM is a probabilistic technique focused to approximate almost any continuous density by using an enough number of Gaussian [16].

3 Methods and Materials

This section describes the experiments carried out to obtain EEG signals from subjects whom performed MI of both hands and the preprocessing and feature extraction procedure.

3.1 Experiment Setup

For this study, eight healthy people (three males and five females) aged 25 to 30 participated in an experiment designed to obtain EEG recordings for two mental tasks:

1. Imagined movements of the left hand and,
2. Imagined movements of the right hand.

These experimental conditions consisted of flexion and extension of the fingers of the right and left hand mentally without performing the actual movements. A graphical user interface developed by our team provided the instructions of the experiment and indicated when the subject had to carry out the mental imaginations.

The experiment was divided into 16 blocks of a duration of 28 s each. A block started with a fixation cross shown on the screen for 5 s, followed by a visual cue of the action to be performed by the participant (3 s). White arrows and a sphere that moved from the center of the screen to the left or right side of the monitor represented the different types of tasks, Fig. 1 (Right). Then, the participant had

to execute for 15 s the imagined movement specified by the interface. Finally, the word “Rest” was shown on the screen for 5 s, indicating that the subject could relax or move freely until the beginning of the next block. The software selected the task of each block randomly. Also, both conditions were balanced, i.e., the subject performed eight times the “left” task and eight times the “right” task. In total, an experiment lasted around seven and a half minutes. Fig. 1 (Left) illustrates the different stages of this paradigm.

During the experiment execution, a g.USBamp amplifier recorded EEG signals from 12 active electrodes at a sampling rate of 256 Hz (g.tec medical engineering GmbH, Austria). Data were band-pass filtered from 0.1 to 100 Hz, and a built-in notch filter removed the power supply noise. According to the international 10/20 system, the electrode positions used in this experiment were FC3, FCz, FC4, C3, Cz, C4, CP3, CPz, CP4, P3, Pz, and P4. This arrangement was selected to cover scalp locations that are close to the motor cortex. Additionally, the ground electrode was located at AFz, and the reference electrode was placed over the right earlobe.

3.2 Preprocessing and Feature Extraction

The Common Spatial-Pattern (CSP) algorithm was used to characterize the brain activity of both experimental conditions. This algorithm finds linear combinations of the original EEG signals (or band-limited components of the EEG) so that the variances of the new signals of one condition are maximized, whereas the variances of the signals of the other condition are minimized. In this way, if the log-variances of the signals in the projected space are used as features, the separability between conditions is optimal. In this study, the CSP algorithm was applied over band-limited components extracted by a filter bank. This strategy is commonly known as Filter Bank Common Spatial-Pattern (FBCSP).

In the preprocessing stage, a filter bank of gaussian bandpass filters with a bandwidth of 4 Hz extracted 22 components from the EEG signals (4, 5, 6, . . . , 25 Hz). Then, the data was separated into epochs or trials of 1 second of time samples. Trials contaminated by visual or muscular artifacts were identified and rejected from this study. Finally, the CSP algorithm was used to compute a new set of signals for each frequency component to increase the separability between conditions. For each band, the three best spatial filters that maximize the variances of the “Left” conditions were calculated. Likewise, the three best spatial filters that maximize the variances of the “Right” conditions were also computed, associated to a class label $y \in \{\text{Left}, \text{Right}\}$. The features used in the classification stage were the log-variances of these time-series. In total, each trial consisted of 132 new projected signals, $x \in \mathbb{R}^{132 \times 256}$.

4 Den Architecture

DEN has the same structure as any other neural network architecture: an input, a hidden and an output layer, Fig. 2.

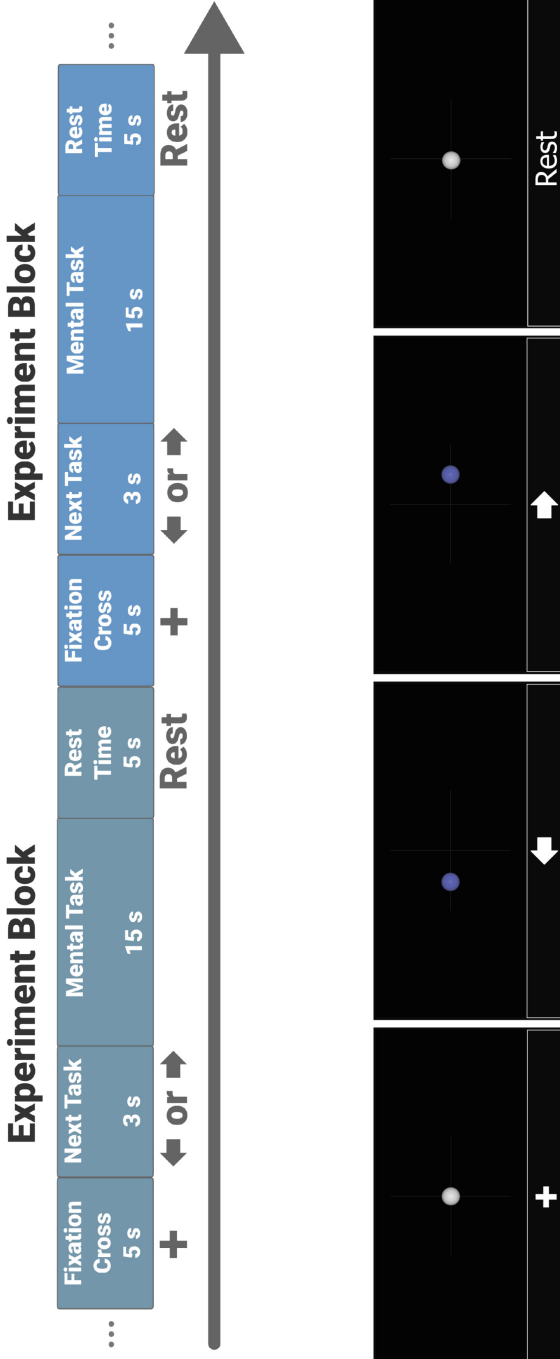


Fig. 1. Left: The experiment starts with **Fixation Cross**, which indicates the beginning of the experiment. After this, **Next Task** just indicates the action that will be performed. **Mental Task** shows the mental task needed to be carried out. And **Rest Time** indicates relaxing time. Right: Visual cues. + Fixation Cross. ← Fixation Cross. → Imagined movements of the left upper limb. → Imagined movements of the right upper limb. **Reset**.

The input layer receives the incoming $x_i = (x_1, x_2, \dots, x_n)^T$ patterns. The hidden layer calculates the Mahalanobis distance between the input patterns and all the hyperellipsoids placed by the dendrites with Eq. 1. Lastly, the output layer assigns patterns to their nearest dendrites which are related to their corresponding C^1, C^2, \dots, C^j classes with Eq. 2:

$$\tau_K = [x_i - \mu_k]^T \sum_k^{-1} [x_i - \mu_k], \quad (1)$$

$$y_i = \operatorname{argmin}(\tau_K), \quad (2)$$

where x_i is a n dimensional vector, τ_K is a vector with k Mahalanobis distances and y_i is the output vector of each x_i pattern. \sum_k^{-1} is the covariance matrix and μ_k is the centroid vector both related with the k hyperellipsoids.

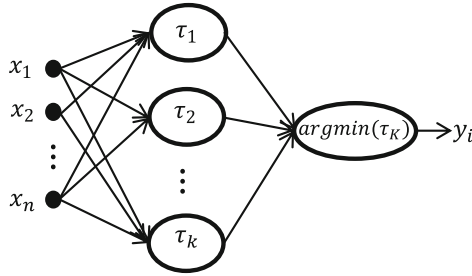


Fig. 2. DEN architecture with an input, a hidden and an output layer.

Once we experimentally observed that DEN has a good performance with small dimensional datasets but not with high dimensional datasets, we occurred to the task of setting the hyperellipsoids by using SGD as an optimization method [4].

To do this, we first implemented the hidden layer (Eq. 1) in a Keras custom layer [5]. Keras computes gradients by using automatic differentiation which automatically calculate the function derivatives of a computer program [17]. And then, we removed the output layer (Eq. 2) which could be replaceable by one or more neurons.

5 Classifiers and Results

This section shows the general details of the used classifiers for the MI classification task and the results achieved by them.

The first technique was a Support Vector Machine (SVM) [18] which is one of the most widespread methods for Brain-Computer Interface (BCI) based on EEG, as previously mentioned. It is formed by two layers: depending on the kernel, the first layer is employed for feature extraction and the second layer creates a hyperplane to separate patterns into two different classes. The goal of the second layer is to create a hyperplane with optimal margins among the support vectors.

In the experiment, we implemented the SVM with a Radial Basis Function (RBF) kernel; so commonly utilized on BCI based on EEG [19,20]. And we selected the γ gamma and the C compensation factor by doing a grid sweep in order to choose the best parameters.

The second classifier was a Multilayer Perceptron (MLP) [21]. This was implemented with two hidden layers, each layer with 100 ReLU neurons and an output layer with a sigmoid neuron (σ). To decrease the overfitting problem, we applied dropout with a rate of 0.2 between each layer.

The last classifiers were DEN and DEN trained by SGD (DEN_SGD). DEN_SGD architecture was composed by three hyperellipsoids with sigmoid activation functions in the input layer and a sigmoid neuron in the output layer, Fig. 3.

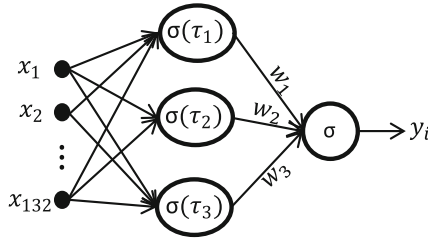


Fig. 3. DEN_SGD architecture for EEG classification task.

Table 1 presents the accuracy achieved by the four classifiers. It can be appreciated that the SVM always obtained a 100% of accuracy in the training stage. However, it computed a 65.81% in testing. As can be seen, the SVM has a high overfitting problem.

Table 1. Experimental results acquired by the SVM, MLP, DEN and SGD_DEN classifiers using our EEG dataset.

Participant	SVM		MLP		DEN		SGD_DEN	
	Train	Test	Train	Test	Train	Test	Train	Test
P1	100.00	62.99	96.73	70.66	76.02	62.76	91.48	74.50
P2	100.00	64.45	99.42	68.49	83.82	62.70	91.09	72.86
P3	100.00	72.97	98.51	78.96	80.79	64.19	93.46	81.27
P4	100.00	63.24	96.47	68.28	83.36	64.02	94.81	72.60
P5	100.00	62.46	97.74	68.61	92.61	63.66	90.52	73.65
P6	100.00	64.35	98.40	71.54	84.28	63.55	90.94	75.44
P7	100.00	75.59	96.84	82.53	70.34	63.50	97.94	85.03
P8	100.00	60.48	98.40	69.96	75.34	57.79	95.19	72.80
Average	100.00	65.81	97.81	72.38	80.82	62.77	93.18	76.02
STD	0.00	5.06	0.98	5.02	6.41	1.94	2.47	4.31

DEN achieved the lowest accuracy in training and in testing, 80.82% and 62.77%, respectively.

The best classifiers for this task were the MLP and the SGD_DEN. The MLP acquired an accuracy of 72.38% in testing and the SGD_DEN slightly acquired an improvement with 76.02%. Both presented the overfitting problem, but it was less with SGD_DEN.

Finally, as a comparison of the proposed method with the other classifiers in statistical terms, it was performed a paired t-test with a significance level of $\alpha = 0.05$. Table 2 gives the p-values acquired in the test. The comparisons between SGD_DEN and SVM, MLP and DEN achieved a less value than α , which indicates that for this dataset SGD_DEN has a significantly better performance.

Table 2. *P* – values of a paired t-test with $\alpha = 0.05$.

Classifiers	<i>P</i> – values
SVM	0.000000136
MLP	0.000007950
DEN	0.000028302

6 Conclusions

In this research, we have implemented SGD to train a DEN and acquired an EEG dataset from eight healthy participants to test the performance of the proposed training algorithm. The besought model achieved an enhancement of 13.25% over the DEN training algorithm and an improvement of 3.64% and 10.21% compared with the MLP and the SVM, respectively. We invite the reader to regard an improvement obtained with a shallow architecture which can be easily implemented in embedded electronic devices. Future work will be the evaluation of DEN_SGD using standard datasets and the implementation of this network to control external electronic devices.

Acknowledgments. H. Sossa and E. Zamora would like to acknowledge the support provided by CIC-IPN and M. Antelis to Tecnológico de Monterrey, in carrying out this research. This work was economically supported by SIP-IPN (grant numbers 20180180, 20180730, 20190007 and 20190166) and CONACYT grant numbers 65 (Frontiers of Science), 268958 and PN2015-873. F. Arce and O. Mendoza-Montoya acknowledge CONACYT for the scholarship granted towards pursuing their PhD and post-PhD studies, respectively.

References

1. Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **6**(6), 861–867 (1993)

2. Arce, F., Zamora, E., Sossa, H.: Dendrite ellipsoidal neuron. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 795–802, May 2017
3. Arce, F., Zamora, E., Fócil-Arias, C., Sossa, H.: Dendrite ellipsoidal neurons based on k-means optimization. *Evolving Syst.* 1–16 (2018). <https://app.dimensions.aion2019/02/18>
4. Montavon, G., Müller, K.-R.: Big learning and deep neural networks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) *Neural Networks: Tricks of the Trade*. LNCS, vol. 7700, pp. 419–420. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_24
5. Chollet, F., et al.: Keras (2015). <https://keras.io>
6. Ritter, G.X., Li, D., Wilson, J.N.: *Image algebra and its relationship to neural networks* (1989)
7. Davidson, J.L., Ritter, G.X.: *Theory of morphological neural networks* (1990)
8. Davidson, J.L., Sun, K.: *Template learning in morphological neural nets* (1991)
9. Davidson, J.L., Hummer, F.: Morphology neural networks: an introduction with applications. *Circ. Syst. Sig. Process.* **12**(2), 177–210 (1993)
10. Segev, I.: *The Handbook of Brain Theory and Neural Networks*, pp. 282–289. MIT Press, Cambridge (1998)
11. Ritter, G.X., Iancu, L., Urcid, G.: Morphological perceptrons with dendritic structure. In: *The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003*, vol. 2., pp. 1296–1301, May 2003
12. Mahalanobis, P.C.: On the generalised distance in statistics. *Proc. Natl. Inst. Sci. India* **2**(1), 49–55 (1936)
13. Arthur, D., Vassilvskii, S.: 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)
14. Sung, K.-K., Poggio, T.: Learning human face detection in cluttered scenes. In: Hlaváč, V., Šára, R. (eds.) *CAIP 1995*. LNCS, vol. 970, pp. 432–439. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60268-2_326
15. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1), 39–51 (1998)
16. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York (2006)
17. Rall, L.B.: Automatic computation of gradients, Jacobians, Hessians, and applications to optimization. In: Rall, L.B. (ed.) *Automatic Differentiation: Techniques and Applications*. LNCS, vol. 120, pp. 91–111. Springer, Heidelberg (1981). https://doi.org/10.1007/3-540-10861-0_6
18. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
19. Lotte, F., Congedo, M., Lécuyer, A., Fabrice, L., Arnaldi, B.: A review of classification algorithms for EEG-based brain-computer interfaces. *J. Neural Eng.* **4**, R1 (2007)
20. Ali, A.B.M.S., Abraham, A.: An empirical comparison of kernel selection for support vector machines. In: *Soft Computing Systems: Design, Management and Applications*, pp. 321–330 (2002)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)