

# Training Algorithm for Dendrite Morphological Neural Network Using K-Medoids

Yisti Vita Via<sup>1</sup>, Chrystia Aji Putra<sup>2</sup>, Ronggo Alit<sup>3</sup>

Informatics Engineering

Universitas Pembangunan Nasional “Veteran” Jawa Timur  
Surabaya, Indonesia

<sup>1</sup>yistivia.if@upnjatim.ac.id, <sup>2</sup>ajiputra@upnjatim.ac.id, <sup>3</sup>ronggo.if@upnjatim.ac.id

**Abstract**—Pattern classification is one of the relevant problems in Artificial Intelligence. Neural networks have been studied as one of the most successful methods for pattern classification. Classical perceptron can only solve linear classification problems. Morphological Neural Networks (MNN) is an alternative way to solve classification problems in the form of linear and nonlinear. Dendrite Morphological Neural Networks (DMNN) is introduced as an improved method of classical MNN. The important problem that occurs in the DMNN training algorithm is to cluster objects with hyper boxes and classify each in the corresponding class. This paper presents the proposed training algorithm using K-medoids clustering algorithm to create the hyper boxes in the dimensional space. K-medoids is better than other clustering methods in execution time and not sensitive to outliers. The implementation of the proposed algorithm will be involved in various simulations using artificial data sets and compared with other methods to evaluate the performance of this method in future work.

**Keywords**—*pattern classification; Neural Networks; Dendrite Morphological; K-medoids; training algorithm*

## I. INTRODUCTION

Pattern classification is one of the relevant problems in Artificial Intelligence. The purpose of pattern classification is to relate an input pattern  $x = (x_1, x_2, \dots, x_n)^T$  to its class  $C^1, C^2, \dots, C^m$  on the mathematical model. Many researchers has studied Neural Networks (NN) for pattern classification. NN is the most successful methods of a classical perceptron. The division of input space in the classical perceptron uses hyperplanes, which are determined by synaptic weights and biases as decision boundaries. So that, single layer perceptron only can be used to solve linear classification problems. Morphological Neural Networks (MNN) is an alternative way to solve classification problems in the form of linear and nonlinear. MNN allows nonlinear classification with only one neuron by dividing the same space into several piecewise lines and creates complex decision boundaries. The processing of morphological involves the operations of min and max to generate the boundaries of classification. The advantage of this process is being more easily implemented in logic devices [1].

MNN is a new kind of artificial neural networks based on mathematical morphology. Since it was shown in 1990 [2], it has significant progress both in theory and practice. MNN has the network structure and learning algorithm, which are

different from the traditional neural networks. MNN works with the input layer and the output layer. Davidson in [3] introduced Dendrite Morphological Neural Networks (DMNN) as an improved method of classical MNN belongs to the ANN approach. In [4] DMNN was re-discussed by Ritter and Sussner.

The important problem that occurs in the training process of DMNN is to cluster objects in hyper boxes. The other problem is to classify each object into the corresponding cluster. The training process method of DMNN based on this approach have been published by many researchers. Most of researcher use heuristics methods to create the hyper boxes in the dimensional space. Each hyper boxes groups classes and represented by dendrite. The use of dendrites in DMNN has several properties [5]. First, the processing of information is performed in the dendrites, so that no hidden layers are required.

Clustering is the process of grouping similar objects into clusters so that these objects are dissimilar in other clusters. [6]. In training process of DMNN, this method groups objects into hyper-boxes. One of the trendy heuristic clustering methods is K-medoids. K-medoids algorithm runs just like K-Means. K-Means clustering works iteratively to assigns a set of objects to the nearest centroid. The centroid is the mean of the coordinates of the similar objects in the cluster. In this case, K-means clustering is sensitive with the outliers but it has the efficient computational time. Different with K-medoids clustering, the representative objects called medoids are based on the centrality of the located object in a cluster. This term causes K-medoids method is less sensitive to outliers than the K-means clustering. In the other aspects, K-medoids is better in execution time and no sensitive to outliers although the complexity is high in comparison with the K-means clustering [7].

This paper proposes a new training algorithm for DMNN using K-medoids for clustering the object population in hyper-box. In addition, we will test whether K-medoid has a significant effect on the performance of the training process

The paper is organized as follows. The previous works in the area of MNN are given in section 2. In section 3, we describe the architecture of the DMNN. The K-medoids algorithm is introduced in section 4. We present the proposed

training algorithm in section 5. And finally, the conclusions and the directions for future research will be given in section 6.

## II. PREVIOUS WORK

In this section, this paper presents neural networks with morphological operations. The researchers have proposed this method over the last 25 years.

Ritter and Davidson proposed the first Morphological Neural Network (MNN) as a combination between image algebra and artificial neural networks in their seminal papers [2] and their computing capabilities discussed by Ritter and Sussner in [4]. MNN [8] are a particular class of Artificial Neural Networks (ANN) based on fundamental concepts of mathematical morphology [9]. The development of MNN has grown due to an important observation made by [10]. In this research, they described the operation of image algebra, which can compute by the typical neural networks [11].

Several MNN have been presented in the literature to solve relevant problems, such as automatic target detection [12], landmine detection [13], handwritten character recognition [14], hyper spectral image analysis [15], time series forecasting [16], pattern recognition [17], computer vision [18], vision-based self-localization in mobile robotics [19], high-frequency financial data prediction [20], image understanding [21], amongst others.

The morphological dendrite neurons were extended in [22] with the consideration that information processing not only occurs in dendrites but also in the cell body of neurons [23]. A key issue in the design of a DMNN is its training. This is in the selection of the number of dendrites and the values of synaptic weights for each dendrite.

Sossa and Guevara proposed Dendrite Morphological Neural Networks (DMNN) as a new training method based on a Divide and Conquer strategy in [24]. The application was used to recognize 3D objects using Kinect [25]. Pessoa and Maragos in [26] combine classical perceptrons with morphological/rank neurons called Morphological/rank/linear neural networks (MRL-NNs). Then a neural model that is similar to DMNN but with a linear activation function has been applied by de A Araujo [27] to regression the problem of forecasting stock markets.

The popular approach for DMNN is to cluster objects with hyper boxes and classify each in the corresponding class. DMNN involves the clustering method to solve this approach. K-medoids algorithm [28] has been widely used to group a set of objects to the cluster with similarity values. There have been some publications for developing a new algorithm of K-medoids clustering. Kaufman and Rousseeuw proposed CLARA, a PAM algorithm that applied to sampled objects instead of all objects [29]. Lucasius, Dane, and Kateman in [30] proposed a new approach of K-medoids clustering using a genetic algorithm. The performance of this method is better than CLARA but the computational increases as the increasing number of clusters. Wei et al. also compared the performance of CLARA with some other variants method for large datasets [31]. Zhang and Couloigner in [32] suggested K-medoids clustering using triangular irregular network concept to reduce

the computational time for calculating the total cost of the replacement in step of PAM method.

In this research, K-medoids will be applied to the training process of DMNN to cluster patterns with hyper boxes and classify each in the corresponding class.

## III. ARCHITECTURE OF DMNN

The architecture of a single output neuron in DMNN is illustrated in Fig. 1. The model consists of  $n$  input neurons represents number of attributes,  $k$  number of class neurons and a final selection output. All input neurons are connected to each class neuron by dendrites  $\tau_1^j, \dots, \tau_k^j$ . Each input neuron  $x_i$  has two weights connected on dendrite  $\tau_k^j$  in the class neuron,  $\omega_{ik}^0$  is the activation terminal fiber and  $\omega_{ik}^1$  is the inhibition terminal fiber. The value of these weights has to be learned from training process of DMNN using a training dataset. The output of dendrite in the class neuron is represented by  $\tau^j$ . The values of all dendrites are computed to provide the output value  $\tau^j(x)$  for each class neuron. The selection unit from the values given by the class neurons determines the class label.

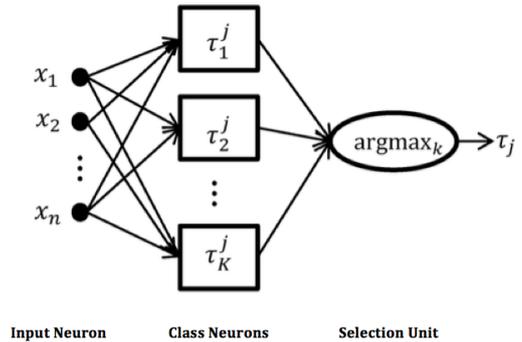


Fig. 1 Architecture of a Single Output Neuron in DMNN

The DMNN computations involve operations, min ( $\wedge$ ) or max ( $\vee$ ). These operators will be easily implemented in logic devices because the generated piece boundary for classification problems comes from comparison operators products. All the dendrites process each pattern and the largest value is then selected.

Equation (1) expresses the computation  $\tau_k^j$  for each dendrite to the corresponding class.

$$\tau_k^j = \wedge_{i=1}^n (x_i + \omega_{ik}^1) \wedge -(x_i + \omega_{ik}^0) \quad (1)$$

Where  $x_i$  is an input vector,  $n$  is the dimensional of vector. The input neurons is represented by  $i \in I$  and  $I \in \{1, \dots, n\}$ .  $\omega_{ik}^0$  and  $\omega_{ik}^1$  are the synaptic weights.

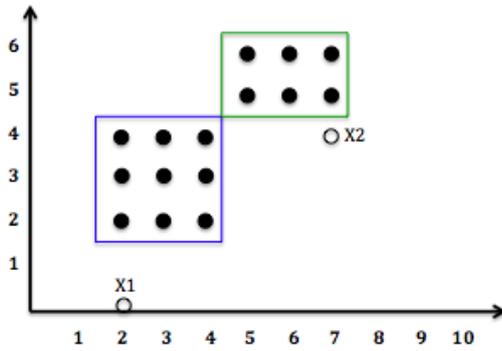


Fig. 2 Example Two Hyper Boxes in DMNN

In order to describe the training process of DMNN algorithm, Fig. 2 presents two hyper boxes that cover all patterns. Blue square represents class  $C^1$  and green square belong to class  $C^2$ . All the dendrites connect the input layer to the output neuron and represented by the weight values  $\omega_{ij}$ .

For testing, two noisy patterns,  $X1 = \binom{2}{0}$  of class  $C^1$  and  $X2 = \binom{7}{4}$  of class  $C^2$  were proposed as examples. When equation (1) is applied to the dendrite, the computations are:

$$\tau_1^1(X1) = \tau_1^1 \binom{2}{0} = [(2-4) \wedge -(2-2)] \wedge [(0-4) \wedge -(0-2)] \\ = [-2 \wedge 0] \wedge [-4 \wedge 2] = [-2 \wedge -4] = -4$$

$$\tau_1^2(X1) = \tau_1^2 \binom{2}{0} = [(2-7) \wedge -(2-5)] \wedge [(0-6) \wedge -(0-5)] \\ = [-5 \wedge 3] \wedge [-6 \wedge 5] = [-5 \wedge -6] = -6$$

$$\tau_1^1(X2) = \tau_1^1 \binom{7}{4} = [(7-4) \wedge -(7-2)] \wedge [(4-4) \wedge -(4-2)] \\ = [3 \wedge -5] \wedge [0 \wedge -2] = [-5 \wedge -2] = -5$$

$$\tau_1^2(X2) = \tau_1^2 \binom{7}{4} = [(7-7) \wedge -(7-5)] \wedge [(4-6) \wedge -(4-5)] \\ = [0 \wedge -2] \wedge [-2 \wedge 1] = [-2 \wedge -2] = -2$$

With these values, the classification is obtained:

$$\tau(X1) = \tau_1^1(X1) \vee \tau_1^2(X1) = -4 \vee -6 = -4$$

$$\tau(X2) = \tau_1^1(X2) \vee \tau_1^2(X2) = -5 \vee -2 = -2$$

Therefore  $\tau(X1) = -1$  corresponds to index of  $C^1$  and  $\tau(X2) = -1$  corresponds to index of  $C^2$  correctly.

#### IV. K-MEDOIDS ALGORITHM

K-medoids clustering involves three phases. They are as follows: initial medoids, update medoids, and assign objects to medoids.

For example that each  $n$  objects with  $p$  variables must be grouped into  $k$  clusters, where  $k < n$  and the value  $k$  is given by user. Then,  $j$ th variable of object  $i$  are defined as  $X_{ij}$  ( $i = 1, \dots, n; j = 1, \dots, p$ ). The dissimilarity measure will be determined by the value of Euclidean distance. The Euclidean distance is adopted from K-means clustering and Partitioning Around Medoids (PAM) algorithms in [35]. The value of Euclidean distance between object  $i$  and  $j$  is calculated by Equation (2).

$$d_{ij} = \sqrt{\sum_{a=1}^p (X_{ia} - X_{ja})^2} \quad i = 1, \dots, n; j = 1, \dots, n \quad (2)$$

#### Step 1: Initial medoids

- Compute the Euclidean distance between each object to all objects based on the dissimilarity measure using Equation (2).

- Calculate  $v_j$  for each object  $j$  using Equation (3).

$$v_j = \sum_{i=1}^n \frac{d_{ij}}{\sum_{i=1}^n d_{ii}} \quad j = 1, \dots, n \quad (3)$$

- Sort ascending order for  $v_j$ 's and select the first  $k$  smallest values of  $k$  objects as initial medoids.

- Assign each object  $j$  to the nearest medoid to obtain the initial cluster and calculate the sum of distances from all objects to the corresponding medoids.

#### Step 2: Update medoids

- Calculate the total distance between each object  $j$  to other objects in its cluster and find the object having the smallest total distance. This object is a new medoid in the corresponding cluster.

- Replace the current medoid in each cluster with the new medoid.

#### Step 3: Assign objects to medoids

- Assign each object  $j$  as member of the nearest new medoid and obtain again the cluster with new member of objects.

- Calculate the total of the distance from all objects to the corresponding medoids like Step 1. If the value is equal to the previous sum of distance, then stop the iterative phase. Otherwise, do the Step 2 again.

K-medoids algorithm runs just like K-means clustering when updating the medoids. The method is a local heuristic. The performance of K-medoids depends on choosing the initial medoids. This method often to select  $k$  objects as initial medoids located in the most middle.

#### V. THE PROPOSED TRAINING ALGORITHM OF DMNN

The steps of the new training algorithm are explained in this section. First step, two definitions are given.

Definition 1: Allow a set of sample objects  $x_1, x_2, \dots, x_m$ , where  $x_i = (x_{i1}, x_{i2}, \dots, x_{in}) \in R^n$  for  $i = 1, 2, \dots, m$  is an  $n$ -dimensional box. Additionally, each objects  $x_i$  belongs to one cluster  $C^k$ , for  $k = 1, 2, \dots, p$ , and  $p > 1$ .

Definition 2: The  $n$ -dimensional box  $HB^n$  contains a limited objects  $x \in R^n$ . The  $HB^n$  explains the weights  $w_{ik}^l$ ,  $l \in \{0,1\}$  where 1 represents excitation and 0 is inhibition.

The purpose of training process is to create the  $n$ -dimensional space  $HB_k^n \in R^{k \times 2n}$  for  $k \in \{1, 2, \dots, K\}$  that determine an optimum decision boundary between the classes. Equation (4) defines a hyper-box as follows:

$$HB_k^n = \{x \in R^n : w_{ik}^1 \leq x_i \leq w_{ik}^0, i = 1, \dots, n\} \quad (4)$$

K-medoids initialization is described below: generate  $k$  clusters into hyper boxes using K-medoids clustering. Step 2:

compute the error value generated by the training process of DMNN. The error value is obtained by calculating how many objects are generated after clustering in hyper box that does not match with the corresponding class. It is possible to add more than one cluster for the same class,  $q$ , for each class. Step 3: this step is repeated until all classes are clustered correctly.

The sample of four classes with two dimensions is used to describe the illustration of K-medoids initialization method.

1. One hyper-box is formed using K-medoids algorithm for  $C^1$ . Then compute the error value by calculating how many objects are generated after clustering in hyper box that does not match with the corresponding class, and iterate until  $k = 2$ . Fig. 3 shows the hipper boxes generated when  $k = 1$  and  $k = 2$ .
2. Do the same step for  $C^2, C^3$ , and  $C^4$ . Fig. 4 presents the hyper boxes generated for  $C^2$  and Fig. 5 exhibits both  $C^3$  and  $C^4$ . In the end, Fig. 6 shows all hyper-boxes, which is generated using the K-medoids algorithm.

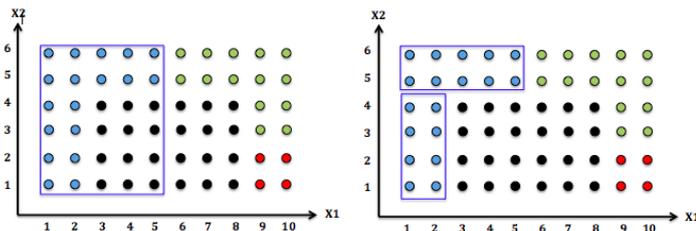


Fig. 3. The Illustration of hyper boxes that generated by K-medoids initialization method for C1.

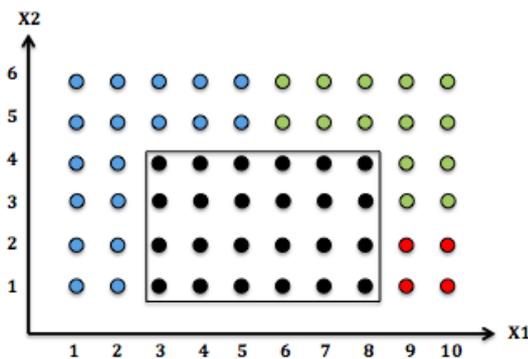


Fig. 4. The Illustration of hyper boxes that generated by K-medoids initialization method C2.

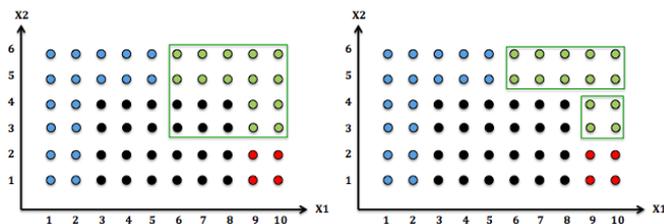


Figure 5. The Illustration of hyper boxes that generated by K-medoids initialization method for C3 and C4.

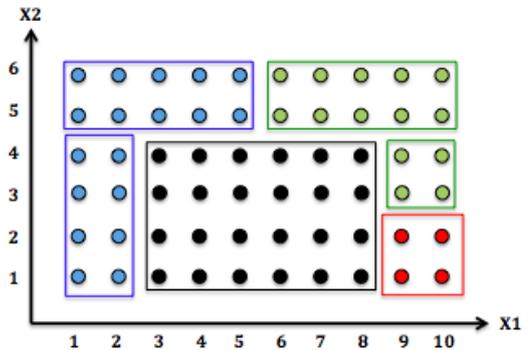


Fig. 6. The Illustration of hyper boxes that generated by K-medoids initialization method for this straightforward example

## VI. CONCLUSION

This paper proposes the training algorithm for DMNN using K-medoids clustering algorithm to generate the initial population of hyper-box. The DMNN training algorithm clusters object with hyper boxes and classify each in the corresponding class. The proposed method create the hyper boxes in the dimensional space. K-medoids is better than other clustering methods in execution time and no sense to outliers. The implementation of the proposed algorithm will be involved in various simulations using artificial data sets and compared with other methods to evaluate the performance of this method in future work.

## ACKNOWLEDGMENT

The authors would like to thank Universitas Pembangunan Nasional Veteran Jawa Timur, especially for the Informatics Engineering Department and the Faculty of Computer Science to provide financial support for this Research.

## REFERENCES

- [1] A MNN is an artificial neural network based on mathematical morphology. Since it was put forward in 1990, it has made great progress both in theory and practice.
- [2] J.L Davidson, G.X Ritter." A theory of morphological neural networks". Proc. SPIE 1215, 378–388. 1990.
- [3] J.L. Davidson, F. Hummer, "Morphology neural networks: an introduction with applications". Circuits Syst. Signal Process. 12 (2) 177–210. 1993.
- [4] G.X. Ritter, P. Sussner, "An introduction to morphological neural networks", in: Proceedings of the 13th International Conference on Pattern Recognition, vol. 4, pp. 709–717, 1996.
- [5] G. Ritter, L. Iancu, "Single layer feedforward neural network based on lattice algebra", Proceedings of the International Joint Conference on Neural Networks, Vol. 4 (2003) 2887–2892, vol.4, 2003.

- [6] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining": A survey. In H. J. Miller & J. Han (Eds.), *Geographic data mining and knowledge discovery*. Taylor & Francis, 2001.
- [7] Dr. A. Preeti Deepali, V. Shipra, "Analysis of K-Means and K-Medoids Algorithm For Big Data", *International Conference on Information Security & Privacy (ICISP2015)*, *Procedia Computer Science* 78 . pp507 – 512. 2016.
- [8] J.L. Davidson, A. Talukder, "Template identification using simulated annealing in morphology neural networks". In: *Proceedings of 2nd Annual Midwest Electro- Technology Conference*, Ames, IA, April, pp. 64–67, 1993.
- [9] J. Serra, "Image Analysis and Mathematical Morphology". Academic Press, London, 1982.
- [10] G.X. Ritter, D. Li, J. N. Wilson, "Image algebra and its relationship to neural networks". In: *Technical Symposium Southeast on Optics, Electro-Optics, and Sensors. Proceedings of SPIE*. Orlando, FL, Mar, pp. 90–101, 1989.
- [11] G.X. Ritter, J.N. Wilson, J.L. Davidson, "Image algebra: An overview". *Comput. Vis. Graph. Image Process.* 49 (3), 297–331, 1990.
- [12] M.A. Khabou, P.D. Gader, "Automatic target detection using entropy optimized shared-weight neural networks". *IEEE Trans. Neural Netw.* 11 (1), 186–193, 2000.
- [13] P.D. Gader, M.A. Khabou, A. Koldobsky, "Morphological regularization neural networks". *Pattern Recogn. Spec. Issue Math. Morphol. Appl.* 33 (6), 935–945, 2000.
- [14] L. F. Pessoa, P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition". *Pattern Recognition* 33 (6), 945 – 960. 2000. URL <http://www.sciencedirect.com/science/article/pii/S0031320399001570>
- [15] G.X. Ritter, G. Urcid, M. S. Schmalz, "Autonomous single-pass endmember approximation using lattice auto-associative memories". *Neurocomputing* 72 (10–12), 2101– 2110, 2009.
- [16] R.A. Araújo, "Evolutionary learning processes to design the dilation-erosion perceptron for weather forecasting". *Neural Process. Lett.* 37, 303–333, 2013.
- [17] Y. Jamshidi, V.G. Kaburlasos, gsaNknn: A {GSA} optimized, lattice computing knn classifier. *Eng. Appl. Artif. Intell.* 35, 277–285, 2014.
- [18] P. Sussner, E. Esmi, I. Villaverde, na, M.G., "The kosko subsethood fuzzy associative memory (ks-fam): Mathematical background and applications in computer vision". *J. Math. Imaging Vision* 42, 134–149, 2012.
- [19] E. Esmi, P. Sussner, H. Bustince, J. Fernandez, J., "Theta-Fuzzy associative memories (Theta-FAMs)". *IEEE Trans. Fuzzy Syst.* 23, 313–326, 2014.
- [20] R.A. Araújo,., Oliveira, A.L.I., Meira, S.R.L., "A hybrid model for high-frequency stock market forecasting". *Expert Syst. Appl.* 42 (8), 4081–4096, 2015.
- [21] M.G., na, D. Chyzyk, "Image understanding applications of lattice autoassociative memories". *IEEE Trans. Neural Netw. Learn. Syst.* 27 (9), 1920–1932, 2016.
- [22] G. X. Ritter and G. Urcid, "Lattice algebra approach to single-neuron computation," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 282- 295, Mar 2003.
- [23] I. Segev, "The handbook of brain theory and neural networks," M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, ch. Dendritic Processing, pp. 282-289, 1998.
- [24] H. Sossa, E. Guevara, "Efficient training for dendrite morphological neural networks". *Neurocomputing* 131, 132 – 142. URL <http://www.sciencedirect.com/science/article/pii/S0925231213010916>. 2014.
- [25] H. Sossa, E. Guevara, "Modified Dendrite Morphological Neural Network Applied to 3D Object Recognition ". *Pattern Recognition: 5th Mexican Conference, MCPR 2013, Querétaro, Mexico, , Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, Ch., June 26-29. pp. 314–324, 2013.*
- [26] L. F. Pessoa, P. Maragos, Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition. *Pattern Recognition* 33 (6), 945 – 960, 2000. URL <http://www.sciencedirect.com/science/article/pii/S0031320399001570>
- [27] de A. Araujo, R., 2012. A morphological perceptron with gradient-based learning for brazilian stock market forecasting. *Neural Networks* 28, 61 – 81. URL <http://www.sciencedirect.com/science/article/pii/S0893608011003200>
- [28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [29] L. Kaufman, and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*. New York: Wiley, 1990.
- [30] C. B. Lucasius, A. D. Dane, and G. Kateman, On k-medoid clustering of large data sets with the aid of a genetic algorithm: Background, feasibility and comparison. *Analytica Chimica Acta*, 282, 647–669, 1993.
- [31] C. P. Wei, Y. H. Lee, and C.M. Hsu, Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems with Applications*, 24(4), 351–363, 2003.
- [32] Q. Zhang and I. Couloigner, A new and efficient k-medoid algorithm for spatial clustering. *Lecture Notes in Computer Science*, 3482, 181–189, 2005.
- [33] G. Ritter, M. Schmalz, Learning in lattice neural networks that employ dendritic computing, *IEEE International Conference on Fuzzy Systems* 2006 (2006) 7–13.
- [34] M. Gori, F. Scarselli, Are multilayer perceptrons adequate for pattern recognition and verification? *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (11) , 1121–1132, 1998.
- [35] L. Kaufman, and P.J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*. New York: Wiley. 1990.