A Brief Account on Morphological Perceptron with Competitive Layer Trained by a Certain Genetic Algorithm

Raul Ambrozio Valente
Dpto. de Eng. Elétrica - CTU
Universidade Estadual de Londrina
Londrina – PR, Brazil
Email: raulavalente@gmail.com

Marcos Eduardo Valle
Dpto. de Matemática Aplicada - IMECC
Universidade Estadual de Campinas
Campinas – SP, Brazil
Email: valle@ime.unicamp.br

***Abstract*—Lattice computing models such as the morphological neural networks and fuzzy neurocomputing models are becoming increasingly important with the advent of granular computing. In particular, the morphological perceptron with competitive learning (MP/CL), introduced by Sussner and Esmi, exhibited satisfactory classification results in some well known classification problems. On the downside, the MP/CL is subject to overfitting in which the network learns singular characteristics from the training data. In this paper, we propose a learning strategy based on a certain genetic algorithm to circumvent the overfitting problem of MP/CL. Computational experiments revealed that the novel model can achieve similar classification results but using a smaller number of hidden neurons.**

***Keywords*-neural networks; morphological perceptron; classification problem; genetic algorithm.**

I. Introduction

Broadly speaking, *neural networks* (NNs) are machines partially inspired by the human brain where the basic processing units are neurons [1]. Rigorous studies in NNs began in 1943 when the biologist McCulloch and the mathematician Pitts introduced a mathematical model of the biological neuron. Approximately 15 years after the publication of the seminal paper of McCulloch and Pitts, Rosenblatt proposed a new approach to pattern recognition called *perceptron*. Although a single perceptron is subject to restrict computational limitations, the multi-layer perceptron (MLP) with at least one hidden layer is a universal approximator [1]. In the context of classification problems, a MLP is able to approximate any compact set $X \subseteq \mathbb{R}^N$, which represents a certain class of patterns, with a given degree of accuracy $\epsilon > 0$. Broadly speaking, the set X is approximated by hyperplanes which are described by the neurons in the hidden layer. In the 1970s and 1980s, several researchers including Bryson, Werbos, Rumelhart and McClelland, developed independently the famous class of *backpropagation algorithms*, which contains many widely used algorithms for training MLP networks [2]–[5].

In the middle 1990s, Sussner and Ritter introduced the first morphological neural networks [6], [7] using the so-called *image algebra* [8], a theory that unifies several techniques for image processing, including traditional linear algebra, the minimax algebra of Cuninghame-Green [9], and mathematical morphology [10]. Usually, a morphological neural network is defined as a type of artificial neural network that performs an elementary operation of mathematical morphology at every node, possibly followed by the application of an activation function. In particular, the morphological perceptron is obtained from the perceptron of Rosenblatt by replacing the usual matrix product by lattice theoretical operations. Analogous to the MLP, a *multi-layer morphological perceptron* (MLMP) is able to approximate any compact $X \subseteq \mathbb{R}^N$ with a given accuracy $\epsilon > 0$ [11]. From a geometrical point of view, the set X is approximated by hyperboxes produced by the MLMP. As a consequence, the MLMP is close related with other hyperboxes neural networks such as the *nested generalized exemplar model* of Salzberg [12], the *fuzzy ARTMAP* of Carpenter et al. [13], and the *fuzzy min-max neural network* of Simpson [14]. More importantly, since many information granules such as fuzzy sets and their extensions, intervals, and rough sets are lattice ordered, the morphological perceptron, as well as many other morphological neural networks, may prove to be useful with the advent of granular computing [15], [16].

The first algorithms used to train the MLMP for classification consist of an incremental process in which new hyperboxes are included during the training process [6], [7], [11]. Despite some interesting properties such as fast convergence in a finite number of steps, these algorithms depend on the sequence of training data. In other words, such as many NN classifiers, the order in which the patterns are presented to the network during the training phase influences the decision surface.

Recently, Sussner and Esmi introduced the *morphological perceptron with competitive learning* (MP/CL) which arises by incorporating a winner-take-all output layer into the original morphological perceptron [17]. Furthermore, these researchers developed an algorithm for training the MLMP which, besides the fast convergence, does not depend on the sequence of training data. On the downside, due to the incremental learning process, the MP/CL is subject to overfitting the training data. In view of this drawback, in this paper we propose a genetic algorithm (GA) to train a morphological perceptron with a winner-take-all layer. Computational experiments with some well known classification problems reveals that the resulting model, referred to as the *morphological perceptron with genetic algorithm* (MP/GA), can efficiently cope with possible peculiarities of the training data. Also, in contrast to the MP/CL, the training of the MP/GA is not described by an incremental process.

The paper is organized as follows. After presenting a brief review on the MP/CL in Section II, we describe the genetic algorithm used to train the morphological perceptron with a winner-take-all layer. Section IV contains some experimental results. The paper finishes with the concluding remarks in Section V.

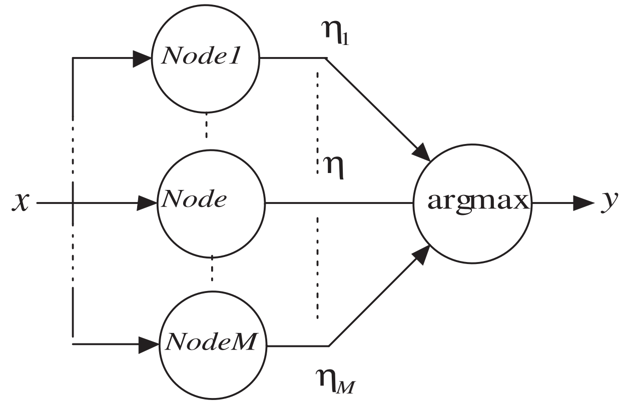


Figure 1. MP/CL network topology.

II. Morphological Perceptron with Competitive Learning (MP/CL)

First of all, recall that a NN is specified by the network topology, node characteristics, and the training rule used to determine the synaptic weight values [1]. The following subsection describes the node characteristics as well as the network topology of the MP/CL. The learning rule of the MP/CL is briefly revised subsequently.

We would like to point out that, in order to simplify the presentation, we slightly adapted the original MP/CL of Sussner and Esmi [17]. Nevertheless, our version of the MP/CL is equivalent to the original model.

A. MP/CL Topology and Node Characteristics

The MP/CL is a feed-forward NN whose topology is depicted in Figure 1. The first layer is composed of M nodes. The weights of the μ-th node are arranged in two vectors $\mathbf{a}_\mu^T = [a_{\mu 1}, \ldots, a_{\mu N}]$ and $\mathbf{b}_\mu^T = [b_{\mu 1}, \ldots, b_{\mu N}]$ of length N. The μ-th node also has a class label denoted by ℓ_μ. Hence, the parameters of the μ-th node of a morphological perceptron with competitive layer will be denoted by a triple $(\mathbf{a}_\mu, \mathbf{b}_\mu, \ell_\mu)$ in this paper.

Given a real-valued input pattern $\mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$, the individual output $\eta_\mu(\mathbf{x})$ of the μ-th node is computed as follows for all $\mu = 1, \ldots, M$:

$$\eta_\mu(\mathbf{x}) = \min\left\{ \min_{j=1:N}\{x_j - a_{\mu j}\}, \min_{j=1:N}\{b_{\mu j} - x_j\} \right\}, \quad (1)$$

where $\min_{j=1:N}\{t_j\}$ yields the smallest value of $\{t_1, \ldots, t_N\}$. A competition among the M nodes takes place in the output layer of the MP/CL. Formally, the MP/CL assigns to the input pattern $\mathbf{x} \in \mathbb{R}^N$ the class label ℓ_{μ^*}, where μ^* denotes the first index μ such that $\eta_{\mu^*} = \max_{i=1:M} \eta_\mu(\mathbf{x})$. As pointed out by Sussner and Esmi, such winner-take-all output layer can be implemented in software by simply selecting the mode with highest activation or in terms of a NN known as MAXNET [18]. Also, note that the MP/CL can be used in multi-class classification problems because we simply attribute the input pattern to the class ℓ_{μ^*} of the winner node.

Geometrically, the set of all patterns $\mathbf{x} \in \mathbb{R}^N$ such that $\eta_\mu(\mathbf{x}) \geq 0$ is an hyperbox in $\mathbb{R}^N$ whose bottom-left and top-right corners are respectively the weight vectors $\mathbf{a}_\mu$ and $\mathbf{b}_\mu$. Let us explore further the geometric interpretation of the MP/CL. To this end, let $[\mathbf{a}_\mu, \mathbf{b}_\mu]$ denote the hyperbox with bottom-left and top-right corners $\mathbf{a}_\mu$ and $\mathbf{b}_\mu$, respectively, defined by the μ-th node. Given an input pattern $\mathbf{x} \in \mathbb{R}^N$, we have $\eta_\mu(\mathbf{x}) \geq 0$ if and only if $\mathbf{x} \in [\mathbf{a}_\mu, \mathbf{b}_\mu]$. Conversely, $\eta_\mu(\mathbf{x}) < 0$ if $\mathbf{x} \notin [\mathbf{a}_\mu, \mathbf{b}_\mu]$. Intuitively, $\eta_\mu(\mathbf{x})$ measures the compatibility of the input $\mathbf{x}$ with the hyperbox $[\mathbf{a}_\mu, \mathbf{b}_\mu]$. The competitive layer attributes to $\mathbf{x}$ the class label of the hyperbox with largest compatibility.

Finally, we would like to remark that the MP/CL belongs to the class of morphological neural networks because (1) is derived from two elementary operations of mathematical morphology [17]. Precisely, the output η_μ is obtained by computing the minimum of an erosion and an anti-dilation.

B. The MP/CL Learning Rule

The learning of the MP/CL is supervised in the sense that the weights $\mathbf{a}_\mu$ and $\mathbf{b}_\mu$ as well as the class label ℓ_μ of the μ-th node are determined using a training set. Let us denote the training set by $\mathcal{S}_{tr} = \{(\mathbf{x}_1, d_1), \ldots, (\mathbf{x}_K, d_K)\}$, where $\mathbf{x}_k \in \mathbb{R}^N$ belongs to class $d_k \in \{1, 2, \ldots, C\}$, for all $k = 1, \ldots, K$.

Broadly speaking, the training of the MP/CL is constructive. It begins by determining an hyperbox that contains only points of a certain class c. In other words, it sets $\ell_1 = c$ and determines the weight vectors $\mathbf{a}_1$ and $\mathbf{b}_1$ such that the hyperbox $[\mathbf{a}_1, \mathbf{b}_1]$ contains only points of class c. Then, the points on this hyperbox are removed from the training set and the process is repeated until all training data have been processed. The reader interested in the details of the algorithm is invited to consult [17].

For the purposes of this paper, we list the following interesting properties of the algorithm proposed by Sussner and Esmi for training the MP/CL:

1) Convergence in a finite number of steps.
2) Perfect separation of the training data according to their class labels.
3) Hyperboxes with distinct class labels do not overlap.
4) Independence of the order in which the training patterns are presented to the network.

On the downside, partially due to the constructive nature of the learning rule, the MP/CL may overtit the training data. For example, the training algorithm may produce a degenerated hyperbox in which the bottom-left and top-right corners coincide. This situation will be confirmed in many computational experiments. The following section presents a certain genetic algorithm that can be used to circumvent the overfitting problem.

III. A Genetic Algorithm for Training the Morphological Perceptron with Competitive Layer

In this section, we present a learning rule for the morphological perceptron with competitive layer based on a *genetic algorithm* (GA). The resulting network will be referred to as the *morphological perceptron with genetic algorithm* (MP/GA).

Genetic algorithms, originally described by Holland in the 1960s, are probabilistic heuristics designed to optimize an objective function referred to as the *fitness function* [19]. A GA keeps a population of individuals, $\mathcal{P}(t) = \{\boldsymbol{v}_1(t), \boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_{pop\_size}(t)\}$, where t is the iteration or generation number. Each individual $\boldsymbol{v}_i$, which is represented by its chromosome, is a potential solution of the optimization problem. Inspired by natural evolution, the individuals at generation t are subject to a competition in which only fittest survive and reproduce. As a consequence, the subsequent generation $t+1$ is composed of descendants of the most apt individuals of generation t. Such as in the nature, the genetic material of the descendants are obtained from genetic operations such as crossover and mutation.

In this paper, an individual $\boldsymbol{v}_i$ corresponds to a morphological perceptron with competitive layer. Formally, we have

$$\boldsymbol{v}_i = \{(\mathbf{a}_{1i}, \mathbf{b}_{1i}, \ell_{1i}), \ldots, (\mathbf{a}_{Mi}, \mathbf{b}_{Mi}, \ell_{Mi})\}, \qquad (2)$$

where M denotes the (fixed) number of nodes and $(\mathbf{a}_{\mu i}, \mathbf{b}_{\mu i}, \ell_{\mu i})$ is the triple that characterizes the μ-th node of the MP/GA, for $\mu = 1, \ldots, M$.

Consider a training set $\mathcal{S}_{tr} = \{(\mathbf{x}_1, d_1), \ldots, (\mathbf{x}_K, d_K)\}$, where $\mathbf{x}_k \in \mathbb{R}^N$ belongs to class $d_k \in \{1, 2, \ldots, C\}$ for all $k = 1, \ldots, K$. In order to improve the generalization performance of the MP/GA, we divided the training set $\mathcal{S}_{tr}$ in two sets $\mathcal{S}_{tr}^H$ and $\mathcal{S}_{tr}^E$ such that $\mathcal{S}_{tr} = \mathcal{S}_{tr}^H \cup \mathcal{S}_{tr}^E$.

The fitness of an individual $\boldsymbol{v}_i$ is measured as the classification error on the set $\mathcal{S}_{tr}^E$. In mathematical terms, the fitness of an individual $\boldsymbol{v}_i$ is given by

$$f(\boldsymbol{v}_i) = \sum_{(\mathbf{x}_k, d_k) \in \mathcal{S}_{tr}^E} (y_k(\mathbf{x}_k) \neq d_k), \qquad (3)$$

where $y_k(\mathbf{x}_k)$ is the output of the morphological perceptron with competitive layer defined in terms of $\boldsymbol{v}_i$ after the presentation of an input $\mathbf{x}_k$. Also, we have

$$(y_k(\mathbf{x}_k) \neq d_k) = \begin{cases} 1, & y_k(\mathbf{x}_k) \neq d_k, \\ 0, & y_k(\mathbf{x}_k) = d_k. \end{cases}$$

The objective of the GA is to minimize the fitness function.

The initial population as well as some genetic operations are based on the set $\mathcal{S}_{tr}^H$. Precisely, an individual $\boldsymbol{v}_i(0)$ of the initial population $\mathcal{P}(0)$ is a morphological perceptron with competitive layer whose nodes are defined as follows: For each class $c \in \{1, \ldots, C\}$, we randomly select M_c data points $(\mathbf{x}_k, d_k) \in \mathcal{S}_{tr}^H$ such that $d_k = c$ and define the μ-th node as $(\mathbf{a}_\mu, \mathbf{b}_\mu, \ell_\mu) = (\mathbf{x}_k, \mathbf{x}_k, d_k)$. Geometrically, the μ-th node represents a degenerated hyperbox in which the bottom-left and top-right corners coincide with $\mathbf{x}_k$. Note that an individual $\boldsymbol{v}_i(0)$ has $M = M_1 + M_2 + \ldots + M_C$ nodes.

The crossover of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ simply swaps nodes between the two individuals. Formally, suppose that

$$\boldsymbol{v}_i = \{(\mathbf{a}_{1i}, \mathbf{b}_{1i}, \ell_{1i}), \ldots, (\mathbf{a}_{Mi}, \mathbf{b}_{Mi}, \ell_{Mi})\}, \qquad (4)$$

and

$$\boldsymbol{v}_j = \{(\mathbf{a}_{1j}, \mathbf{b}_{1j}, \ell_{1j}), \ldots, (\mathbf{a}_{Mj}, \mathbf{b}_{Mj}, \ell_{Mj})\}, \qquad (5)$$

have been selected for mating. Then, we randomly select one crossover point μ and define the resulting two offspring $\boldsymbol{v}_i'$ and $\boldsymbol{v}_j'$ as can be seen in (6) and (7) at the top of the following page.

Now, let us consider the case in which a node $(\mathbf{a}_\mu, \mathbf{b}_\mu, \ell_\mu)$ of an individual $\boldsymbol{v}_i'$ undergoes a mutation. In this case, we compute the activation $\eta_\mu(\mathbf{x}_k)$ given by (1) for all $\mathbf{x}_k \in \mathcal{S}_{tr}^H$ and determine an index k^* such that

$$\eta_\mu(\mathbf{x}_{k^*}) = \max\left\{\eta_\mu(\mathbf{x}_k) : \eta_\mu(\mathbf{x}_k) < 0, \mathbf{x}_k \in \mathcal{S}_{tr}^H\right\}. \qquad (8)$$

In other words, $\mathbf{x}_{k^*}$ is the pattern of $\mathcal{S}_{tr}^H$ that does not belong to the hyperbox $[\mathbf{a}_\mu, \mathbf{b}_\mu]$ but has the largest compatibility. Then, the bottom-left and top-right corners are modified if the node μ and the pattern $\mathbf{x}_{k^*}$ belong to the same class. Formally, if $d_{k^*} = \ell_\mu$, we redefine the parameters $\mathbf{a}_\mu$ and $\mathbf{b}_\mu$ as follows

$$\mathbf{a}_\mu \leftarrow \min(\mathbf{a}_\mu, \mathbf{x}_{k^*}) \quad \text{and} \quad \mathbf{b}_\mu \leftarrow \max(\mathbf{b}_\mu, \mathbf{x}_{k^*}), \qquad (9)$$

where $\min(\cdot,\cdot)$ and $\max(\cdot,\cdot)$ yield respectively the component-wise minimum and maximum of the arguments. Geometrically, the hyperbox $[\mathbf{a}_\mu, \mathbf{b}_\mu]$ is enlarged to include the pattern $\mathbf{x}_{k^*}$ if both have to the same class label.

Note that neither the crossover nor the mutation change the number M of nodes of an individual. Specifically, the

$$\boldsymbol{\upsilon}_i' = \{(\mathbf{a}_{1i}, \mathbf{b}_{1i}, \ell_{1i}), \ldots, (\mathbf{a}_{\mu i}, \mathbf{b}_{\mu i}, \ell_{\mu i}), (\mathbf{a}_{\mu+1j}, \mathbf{b}_{\mu+1j}, \ell_{\mu+1j}), \ldots, (\mathbf{a}_{Mj}, \mathbf{b}_{Mj}, \ell_{Mj})\}, \tag{6}$$

and

$$\boldsymbol{\upsilon}_j' = \{(\mathbf{a}_{1j}, \mathbf{b}_{1j}, \ell_{1j}), \ldots, (\mathbf{a}_{\mu j}, \mathbf{b}_{\mu j}, \ell_{\mu j}), (\mathbf{a}_{\mu+1i}, \mathbf{b}_{\mu+1i}, \ell_{\mu+1i}), \ldots, (\mathbf{a}_{Mi}, \mathbf{b}_{Mi}, \ell_{Mi})\}. \tag{7}$$

Experimental Results

Problem	MP/CL			MP/GA		
	E_{tr}	E_{te}	M	E_{tr}	E_{te}	M
Ripley's Problem				12.4%	9.9%	2+2
	0%	10.2%	26+29	13.6%	9.3%	3+3
Iris	0%	0%	1+7+6	7.6%	6.7%	1+2+2
Breast Cancer	0%	4.07%	34+30	6.5%	4.6%	3+3

Table I
RESULTS FROM EXPERIMENTAL TESTS USING MP/GA.

MP/GA evolves to a morphological perceptron with competitive layer with the fixed number $M = M_1 + M_2 + \ldots + M_C$ of nodes – where M_c is the number of nodes representing class c – that minimizes the classification error over the training set $\mathcal{S}_{tr}^E$.

IV. EXPERIMENTAL RESULTS

In this section we evaluate the performance of the MP/GA in some well known classification problems, namely: The synthetic problem of Ripley [20], the iris recognition problem, and the diagnostic Wisconsin breast cancer problem. Recall that the latter two are available at the machine learning database from University of California at Irvine-USA. The table I summarizes our results. A detailed discussion of the values in this table can be found in the following subsections.

We would like to point out that we used the roulette wheel as the selection process [19]. The probability of crossover and mutation have been respectively 0.3 and 0.2. Also, we considered a population of size $pop\_size = 30$ and allowed a maximum number of $t_{\max} = 100$ generations. We would like to point out that these parameters have been obtained after some preliminary experiments in which we considered, in particular, the classification error and the complexity of the MP/GA. Notwithstanding, further applications of the MP/GA require some fine-tuning of these parameters.

Finally, in agreement to some literature reporting results on the MP/CL, we used the whole training set to construct the hyperboxes as well as to evaluate the fitness of the individual. In other words, we defined $\mathcal{S}_{tr}^E = \mathcal{S}_{tr}^H = \mathcal{S}_{tr}$ in all computational experiments. Nevertheless, a set $\mathcal{S}_{te}$, referred to as the test set and different from $\mathcal{S}_{tr}$, have been also used to evaluate the performance of the NNs. The classification errors computed over the sets $\mathcal{S}_{tr}$ and $\mathcal{S}_{te}$ are denoted respectively by E_{tr} and E_{te}.

A. *Ripley's Synthetic Problem*

The synthetic problem of Ripley is composed of two classes with a bimodal distribution in $\mathbb{R}^2$ [20]. The database is composed of 250 patterns for training and 1000 for test. The class distributions allow a best-possible error rate of approximately 8%. Moreover, since the data are in $\mathbb{R}^2$, we can visualize the hyperboxes represented by the nodes of the morphological perceptrons.

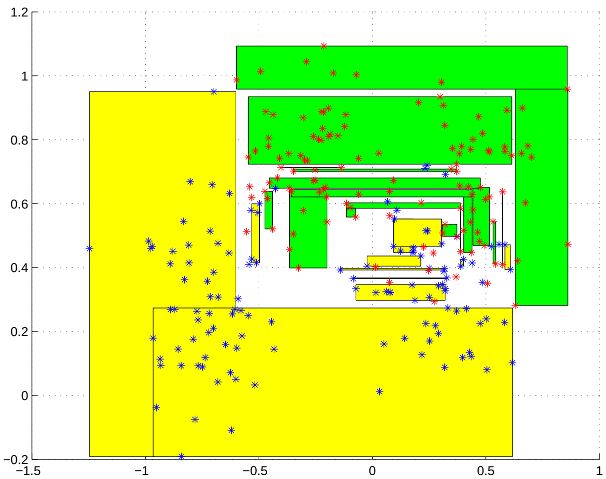
As expected, the MP/CL achieved no classification error from the training set but yielded a 10.2% error rate on the test set. The resulting network has 55 nodes in which 26 refer to one class while the remaining 29 nodes correspond to the other class. The hyperboxes produced by the MP/CL can be visualized in Figures 2a) and 3a). Since any data training point belongs to at least one hyperbox, we conclude from Figure 2 that the network have many degenerated nodes. As the reader can appreciate in Figure 3, the large number of nodes resulted some classification errors in the test set.

Figures 2b) and 3b) show the hyperboxes produced by the MP/GA with different number of nodes. First, we considered a MP/GA with only 4 nodes i.e., two hyperboxes for each class. This network achieved classification errors of 9.9% and 12.4% from the test and training sets, respectively. Similarly, a MP/GA with 6 nodes, 3 for each class, yielded respectively the errors 13.6% and 9.3% for training and testing. Note that the errors produced by both MP/GA in the test set are smaller than the error yielded by the MP/CL. Furthermore, the former required much less nodes.

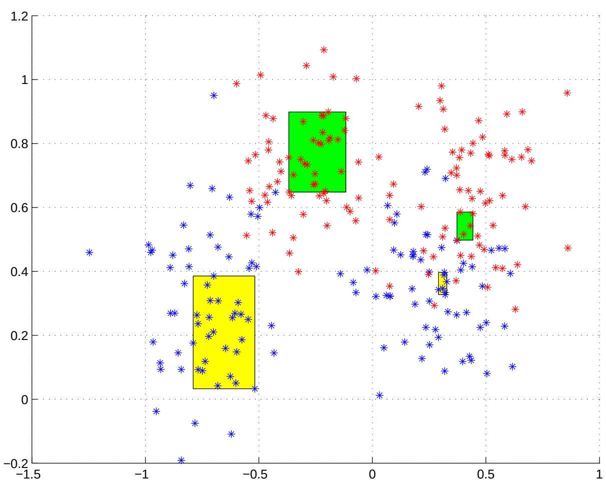
B. *Iris Recognition Problem*

The iris recognition problem proposed by Fisher in the 1930s is composed of 50 samples for each of the species of iris flowers: setosa, versicolour, and virginica. The task is to determine the specie using 4 features: height and width of sepal as well as the height and width of petal. In our experiments, the first 35th patterns of each class have been used for train while the remaining data have been used for test.

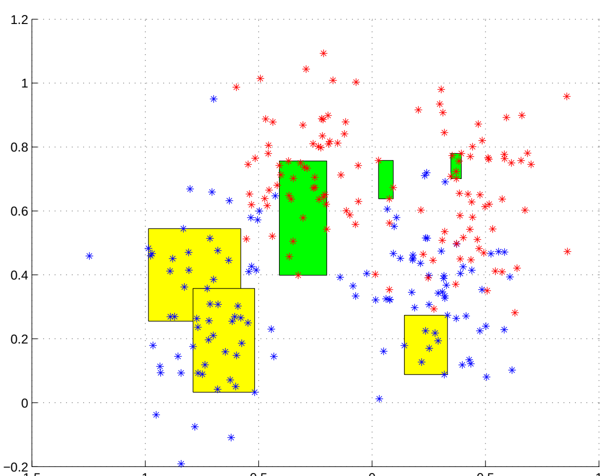
The MP/CL yielded no classification error in both training and test sets. In contrast, the MP/GA with 1 node for the first class and 2 nodes for each of the following 2 classes, produced errors at the rates 7.6% and 6.7% on the training and test sets, respectively. Nevertheless, the MP/CL has 14 nodes while the MP/GA has only 5.



(a) MP/CL - 26+29 nodes

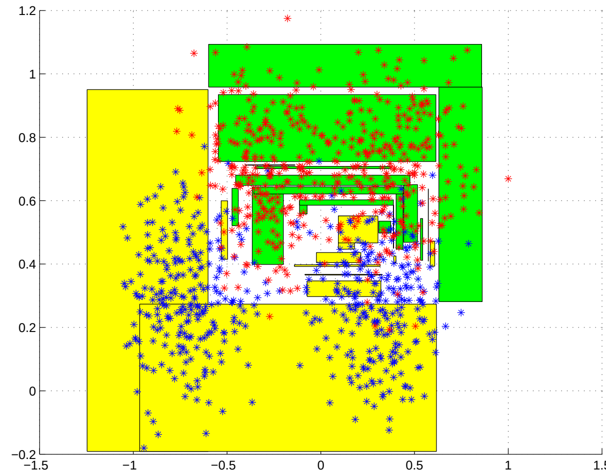


(b) MP/GA - 2+2 nodes

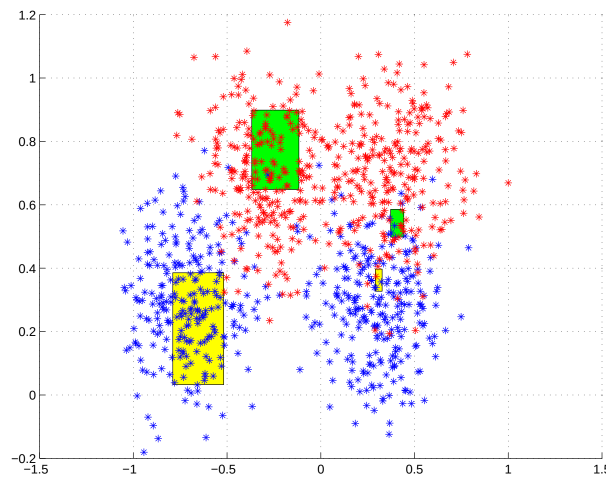


(c) MP/GA - 3+3 nodes

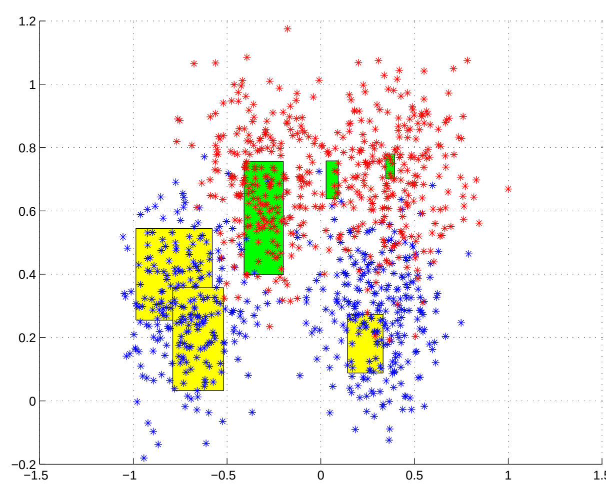
Figure 2. Hiperboxes created by MP/CL and MP/GA together with the training data.



(a) MP/CL - 26+29 nodes



(b) MP/GA - 2+2 nodes



(c) MP/GA - 3+3 nodes

Figure 3. Hiperboxes created by MP/CL and MP/GA together with the testing data.

C. Diagnostic Wisconsin Breast Cancer Problem

The data of the breast cancer diagnostic problem, created by Wolberg, Street and Mangasarian from Wisconsin University, was obtained by an autopsy of breast tissue. The set has 569 samples, where 357 correspond to benign and 212 represent malignant. In our experiments, we used the first 249 and 148 samples from benign and malignant as the training set. The remaining data have been used for test.

On one hand, the MP/CL, with 64 nodes, achieved a 4.07% classification error in the test set. On the other hand, MP/GA with only 6 nodes yielded 4.6% error in same testing data.

V. Concluding Remarks

The morphological perceptron with competitive learning (MP/CL) introduced recently by Sussner and Esmi has many interesting properties including independence of the order in which the training patterns are presented to the network. On the downside, this network is subject to overfitting. In this paper, we proposed a genetic algorithm to train the morphological perceptron with competitive layer (MP/GA) which may circumvent such drawback of the original MP/CL. Preliminary computational experiments using three well-known classification problems revealed that the MP/GA is competitive to the MP/CL in terms of the classification error over the test set but using less nodes. In the future, we plan to investigate further the use of genetic algorithm, in particular the effect of the genetic operations, in the learning phase of the morphological perceptron with a competitive layer.

Acknowledgment

This work was supported in part by CNPq under Grant 304240/2011-7, Fundação Araucária, and PROIC-UEL.

References

[1] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2009.

[2] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, estimation, and control*, 1st ed. Taylor and Francis, 1975.

[3] P. J. Werbos, “Beyond regression: New tools for prediction and analysis in the behavioral sciences,” Ph.D. dissertation, Committee on Applied Mathematics, Havard University, Cambridge, Mass, 1974.

[4] D. Rumelhart and J. McClelland, *Parallel Distributed Processing: Explorations in the Microstrutures of Cognition*. Cambridge, MA: MIT Press, 1986, vol. 1.

[5] D. Rumelhart, G. Hinton, and R. Williams, *Learning internal representation by error propagation*, ser. Parallel Distributed Processing, D. Rumelhart and J. McClelland, Eds. Cambridge, MA: MIT Press, 1988.

[6] G. X. Ritter and P. Sussner, “An introduction to morphological neural networks,” in *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, pp. 709–717.

[7] V. Kaburlasos and G. Ritter, Eds., *Computational Intelligence Based on Lattice Theory*. Heidelberg, Germany: Springer-Verlag, 2007.

[8] G. X. Ritter, J. N. Wilson, and J. L. Davidson, “Image algebra: An overview,” *Computer Vision, Graphics, and Image Processing*, vol. 49, no. 3, pp. 297–331, Mar. 1990.

[9] R. Cuninghame-Green, *Minimax Algebra: Lecture Notes in Economics and Mathematical Systems 166*. New York: Springer-Verlag, 1979.

[10] P. Soille, *Morphological Image Analysis*. Berlin: Springer Verlag, 1999.

[11] 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, 2003.

[12] S. Salzberg, “A nearest hyperrectangle learning method,” *Machine Learning*, vol. 6, pp. 251–276, 1991.

[13] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, “Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps,” *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, Sep. 1992.

[14] P. Simpson, “Fuzzy min-max neural networks – part 1: Classification,” *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 776–786, 1992.

[15] W. Pedrycz, A. Skowron, and V. Kreinovich, Eds., *Handbook of Granular Computing*. New York: John Wiley and Sons, Inc., 2008.

[16] W. Pedrycz, B. J. Park, and S. K. Oh, “The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition,” *Pattern Recognition*, vol. 41, pp. 3720–3735, 2008.

[17] P. Sussner and E. L. Esmi, “Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm,” *Information Sciences*, vol. 181, no. 10, pp. 1929–1950, May 2011.

[18] R. Lippmann, “An introduction to computing with neural nets,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-4, pp. 4–22, 1987.

[19] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1998.

[20] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.