Learning and Nonlinear Models - Journal of the Brazilian Computational Intelligence Society, Vol. 11, Iss. 1, pp. 48–55, 2013.

A SWARM-BASED EVOLUTIONARY MORPHOLOGICAL APPROACH FOR BINARY CLASSIFICATION PROBLEMS

Ricardo de A. Araújo[1,2], Adriano L. I. Oliveira[1], Silvio Meira[1]
[1] Informatics Center (CIn), Federal University of Pernambuco (UFPE), Recife, PE, Brazil
[2] Informatics Department, Federal Institute of Sertão Pernambucano, Ouricuri, PE, Brazil
ricardo.araujo@ifsertao-pe.edu.br or {raa,alio,srlm}@cin.ufpe.br

Abstract – In this work we propose a swarm-based evolutionary morphological approach to deal with binary classification problems. It consists of a hybrid neuron based on principles of mathematical morphology and lattice theory, referred to as dilation-erosion-linear perceptron (DELP). We also present a swarm-based evolutionary learning process, called DELP(PSO), using a particle swarm optimizer (PSO) to design the DELP model, due to some drawbacks from gradient estimation of morphological operators in the classical learning process of the DELP. Besides, we conduct an experimental analysis using two relevant binary classification problems and the obtained results are discussed and compared with those obtained by established techniques in the literature.

Keywords: Dilation-Erosion-Linear Perceptron, Evolutionary Learning, Particle Swarm Optimizer, Binary Classification Problems, Mathematical Morphology, Lattice Theory.

1 Introduction

The classical perceptron is the most known neuron model proposed in the literature [1, 2]. It is inspired by the concept of biological neurons and it is able to solve linear classification problems [1, 2]. There is a particular class of artificial neurons based on the framework of mathematical morphology (MM) [3, 4] and lattice theory [5–7], called morphological perceptrons (MPs) [8, 9], which have been successfully applied as solution of linear and nonlinear problems [9–27].

Among several MPs proposed in the literature, the dilation-erosion-linear perceptron (DELP) [27], a hybrid perceptron which employ dilation and erosion operators from MM combined with a finite impulse response linear operator, was recently proposed to deal with classification problems. The classical DELP learning process, referred to as DELP(BP), employs a gradient steepest descent method using the back propagation (BP) algorithm [27]. The main drawback of the classical DELP learning process is the need of a systematic methodology to overcome the non-differentiability problem of dilation and erosion operators [27]. This is because, in some situations, this scheme can lead to abrupt changes and compromising the numerical robustness of the gradient estimation, so that makes the learning process unstable [27].

In this sense, this work presents a swarm-based evolutionary morphological approach to deal with binary classification problems. The proposed approach employs the dilation-erosion-linear perceptron (DELP) [27] with a swarm-based evolutionary learning process, called DELP(PSO), using a particle swarm optimizer (PSO) [28, 29]. An experimental analysis is conducted with the proposed approach using two relevant binary classification problems (Ripley's Synthetic [30] and the Wisconsin Breast Cancer [31]) and the obtained results are discussed and compared with those obtained by established techniques in the literature, where it is possible to notice that the DELP model designed by a swarm-based evolutionary approach can be used as an accurate binary classifier.

This paper is organized as follows. Section 2 presents the DELP fundamentals and describes the proposed evolutionary learning process. In Section 3 we present the simulations and the experimental results with the proposed model, as well as a comparison between the obtained results with those given by the models previously presented in literature. At the end, in Section 4, we present the conclusions of this work.

2 The Dilation-Erosion-Linear Perceptron

The dilation-erosion-linear perceptron (DELP) consists of a linear combination of a nonlinear operator (dilation and erosion operators) and a linear operator (finite impulse response). Next we present the definition of the DELP.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ a real-valued input signal inside a d-point moving window and let y the output of the DELP. Then, the DELP is defined by a hybrid morphological-linear system with local signal transformation rule $\mathbf{x} \rightarrow y$, given by

$$y = \lambda\alpha + (1-\lambda)\beta, \qquad \lambda \in [0, 1], \tag{1}$$

where

$$\beta = \mathbf{x} \cdot \mathbf{p}^T = x_1 p_1 + x_2 p_2 + \ldots + x_d p_d, \tag{2}$$

and

$$\alpha = \theta\varphi + (1-\theta)\omega, \qquad \theta \in [0, 1], \tag{3}$$

in which

$$\varphi = \delta_{\mathbf{a}}(\mathbf{x}) = \bigvee_{i=1}^{d} (x_i + a_i), \tag{4}$$

and

$$\omega = \varepsilon_{\mathbf{b}}(\mathbf{x}) = \bigwedge_{i=1}^{d} (x_i +' b_i), \tag{5}$$

where term d denotes the dimensionality of the input signal ($\mathbf{x}$), terms $\lambda, \theta \in \mathbb{R}$ and $\mathbf{a}, \mathbf{b}, \mathbf{p} \in \mathbb{R}^d$. The vector $\mathbf{p} \in \mathbb{R}^d$ represents the coefficients (weights) of the linear operator. The term β represents the output of the linear operator. The term α represents the convex combination of the morphological operators of dilation and erosion (the mixture term is defined by θ). The terms φ and ω represent the output of morphological operators of dilation and erosion, respectively. The vectors $\mathbf{a}$ and $\mathbf{b}$ represent the structuring elements (weights) of the dilation ($\delta_{\mathbf{a}}(\mathbf{x})$) and erosion ($\varepsilon_{\mathbf{b}}(\mathbf{x})$) operators employed into the nonlinear module of the DELP. Terms $\bigvee$ and $\bigwedge$ represent the supremum and the infimum operations. Note that the output y is given by a convex combination of the linear operator and another convex combination of morphological operators of dilation and erosion (the mixture term is defined by λ). The main differences between "$+'$" and "$+$" are given by the following rules:

$$(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty, \tag{6}$$

and

$$(-\infty) +' (+\infty) = (+\infty) +' (-\infty) = +\infty. \tag{7}$$

2.1 The Proposed Swarm-based Evolutionary Learning Process

Note that the DELP model requires the setting of the parameters **a**, **b**, **p**, λ and θ. Therefore, the weight vector to be used in the training process is given by

$$\mathbf{w} = (\mathbf{a}, \mathbf{b}, \mathbf{p}, \lambda, \theta). \tag{8}$$

The proposed swarm-based evolutionary process, called DELP(PSO), employees a particle swarm optimizer (PSO) [28, 32, 33], which uses the idea of the social behavior (swarm) that a population of individuals (referred to as particles) adapts to its environment, to adjust the weights of the DELP model according to an error criterion until convergence or until the end of PSO generations. Each i-th particle from swarm at generation g represents a candidate weight vector (denoted by $\mathbf{w}_i^{(g)}$) for the DELP model. The scheme to adjust the weight vector is initially to define a fitness function $fit(\mathbf{w}_i^{(g)})$ (which must reflect the solution quality achieved by the parameters configuration of the system), given by

$$fit(\mathbf{w}_i^{(g)}) = \frac{1}{M} \sum_{j=1}^{M} e^2(j), \tag{9}$$

where M is the number of input patterns and $e(j)$ is the instantaneous error, given by

$$e(j) = d(j) - y(j), \tag{10}$$

where $d(j)$ and $y(j)$ are the desired output signal and the actual model output for the training sample j, respectively.

The PSO procedure starts with its parameters definition. There are four initial parameters to be defined [28,32,33]: i) S is the swarm size, ii) c_1 and c_2 represent the acceleration coefficients to control the velocity change of a particle in a single iteration, iii) ι is the inertia weight, where its value is typically set to vary linearly from 1 to near 0 during the optimization process, and iv) ν_{min} and ν_{max} represent the lower and upper bounds of the particles velocities. In our simulations, the swarm comprises ten particles ($S = 10$).

The first step is to build a set of candidate particles, or search space points. In this way, the i-th swarm particle of the g-th PSO generation has a current position ($\mathbf{w}_i^{(g)} \in \mathbb{R}^n$) and a velocity ($\nu_i^{(g)} \in \mathbb{R}^n$) in the search space, respectively given by

$$\mathbf{w}_i^{(g)} = (w_{i,1}^{(g)}, w_{i,2}^{(g)}, \ldots, w_{i,n}^{(g)}), \tag{11}$$

and

$$\nu_i^{(g)} = (\nu_{i,1}^{(g)}, \nu_{i,2}^{(g)}, \ldots, \nu_{i,n}^{(g)}), \tag{12}$$

in which $i = 1, 2, \ldots, S$ and $g = 1, 2, \ldots$. Recall that term n represents the dimensionality of the DELP model weight vector, which is given by $3d + 2$.

Then, the PSO starts a loop containing some steps to minimize the fitness function $fit : \mathbb{R}^n \to \mathbb{R}$. The next steps are used to update particles velocities ($\nu_i^{(g)}$) and current positions ($\mathbf{w}_i^{(g)}$), as well as particles personal best position ($\mathbf{p}^{+}{}_i^{(g)}$) and swarm best position $\mathbf{p}^{*}{}_i^{(g+1)}$.

The particles velocities can be updated by

$$\nu_i^{(g+1)} = \iota \nu_i^{(g)} + c_1 r_1 (\mathbf{p}^{+}{}_i^{(g)} - \mathbf{w}_i^{(g)}) + c_2 r_2 (\mathbf{p}^{*}{}_i^{(g)} - \mathbf{w}_i^{(g)}), \tag{13}$$

where $r_1, r_2 \sim U(0,1)$ are elements from two uniform random sequences in the interval [0,1]. Terms $\mathbf{p^+}_i^{(g)} \in \mathbb{R}^n$ and $\mathbf{p^*}_i^{(g)} \in \mathbb{R}^n$ represent, respectively, the particle personal best position and the swarm best position, which will be formally defined below. The values of each component in every $\nu_i^{(g+1)}$ vector can be clamped to the range $[\nu_{min}, \nu_{max}]$ in order to reduce the likelihood of particles leaving the search space. This mechanism does not restrict the values of $\mathbf{w}_i^{(g+1)}$ in the range of $\nu_i^{(g+1)}$, it only limits the maximum distance that a particle will move during each iteration [28, 32, 33].

The particles position can be updated by

$$\mathbf{w}_i^{(g+1)} = \mathbf{w}_i^{(g)} + \nu_i^{(g+1)} \tag{14}$$

The personal best position of each particle ($\mathbf{p^+}_i^{(g)}$), can be updated by

$$\mathbf{p^+}_i^{(g+1)} = \begin{cases} \mathbf{p^+}_i^{(g)} & \text{if } fit(\mathbf{w}_i^{(g+1)}) \geq fit(\mathbf{p^+}_i^{(g)}), \\ \mathbf{w}_i^{(g+1)} & \text{otherwise.} \end{cases} ; \tag{15}$$

Note that $\mathbf{p^+}_i^{(0)} = \mathbf{w}_i^{(0)}$.

The swarm best position (found by any particle during all previous iterations), denoted by $\mathbf{p^*}_i^{(g)}$, can be updated by

$$\mathbf{p^*}_i^{(g+1)} = \arg\min(fit(\mathbf{p^+}_i^{(g+1)})) \quad \text{with} \quad i = 1, 2, \ldots, s \tag{16}$$

where $\arg\min(\cdot)$ denotes the minimum argument. Note that $\mathbf{p^*}_i^{(0)} = \mathbf{p^+}_i^{(0)}$.

Further details of PSO procedure can be found in [28,32,33]. Figure 1 presents the proposed DELP(PSO) steps. Two stopping criteria are used in the proposed DELP(PSO): i) The maximum generation number (PSO_{gen}), and ii) The decrease in the training error process training (Pt) [34] of the fitness function.

```
begin DELP(PSO)
    Initialize PSO parameters (s, c1, c2, ι, ν_min, ν_max) according to [28,32,33];
    Initialize the stop condition;
    g = 0;            // g: actual generation
    for i = 1 to S do
        create particle w_i^(g);
        initialize DELP parameters with the values supplied by w_i^(g);
        calculate y and the instantaneous error for all input patterns;
        evaluate the particle fit(w_i^(g)) using the Equation 9;
    end
    repeat
        for i = 1 to S do
            Update the current particle velocity ν_i^(g+1) using Equation (13);
            Update the current particle position w_i^(g+1) using Equation (14);
            Initialize DELP parameters with the values supplied by w_i^(g+1);
            Calculate y and the instantaneous error for all input patterns;
            Evaluate the candidate particle fit(w_i^(g+1)) using the Equation 9;
            Update the particle personal best position p+_i^(g+1) using Equation (15);
        end
        Update the swarm best position p*_i^(g) using Equation (16);
        g = g + 1;
    until not stop condition ;
end
```

Figure 1: The proposed DELP(PSO) steps.

3 Simulations and Experimental Results

The well-known Ripley's synthetic and Wisconsin breast cancer classification problems were used as a test bed for the evaluation of the proposed model. To assess the classification performance we use the percentage of misclassified patterns (PMP) [9] metric. Also, we use the percentage gain (PG) metric, in terms of the PMP obtained using DELP(PSO) model and using other models investigated in this work, which is given by

$$PG = 100 - 100\frac{PMP_{\text{delp}}}{PMP_{\text{model}}}, \tag{17}$$

where PMP_{delp} represents the PMP obtained using the proposed DELP(PSO) model and PMP_{model} represents the PMP obtained using the investigated model.

It is worth mentioning that the data was normalized to lie within the range $[0, 1]$ according to Prechelt [34]. The entries of the DELP weight vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{p}$ are randomly initialized within the range $[-1, 1]$. The initial DELP mixture coefficients λ and θ are randomly chosen in the interval $[0, 1]$. It is worth mentioning that two stop conditions are used into the learning process: i) The maximum epoch number equals to 10^4, and ii) The decrease in the training error process training (Pt) of the cost function equals to 10^{-6}.

In order to establish a fair performance comparison, results with the following classification models were examined in the same context and under the same experimental conditions: multilayer perceptrons (MLP) [1,2], morphological-rank-linear neural network (MRLNN) [35], morphological perceptron with competitive learning (MP/CL) [9], single layer morphological perceptron (SLMP) [36], fuzzy lattice neural network (FLNN) [13], fuzzy lattice reasoning (FLR) [37], k-nearest neighbors (KNN) [38], decision tree (DT) [39, 40], support vector machine (SVM) [2] and dilation-erosion-linear perceptron with gradient-based learning, that is, the DELP(BP) [27].

In all experiments we used the MLP model with sigmoidal processing units and a single hidden layer. For its learning process we used the Levenberg-Marquardt [41] algorithm using the following stopping criteria [34]: i) The maximum epoch number equals to 10^4; ii) The decrease in the training error process training (Pt) of the cost function equals to 10^{-6}. Also, for the MRLNN model we used the same parameters suggested by [35] with a single hidden layer, and for its learning process we used the generalized back propagation (GBP) [35] algorithm with learning rate equals to 0.01, scale factor equals to 0.001 and using the same stopping criteria of the MLP model. It is worth mentioning that for both MLP and MRLNN models, we applied the 10-fold cross validation to determine the number of hidden processing units (5, 10, 15, 20, 25 or 50). For the MP/CL model we used the same design process and parameters definition suggested by [9]. For the SLMP model we used the same design process and parameters definition suggested by [36]. For the FLNN model we used the same design process and parameters definition suggested by [9, 13]. For the FLR model we used the same design process and parameters definition suggested by [9, 37]. For the KNN model we used the 10-fold cross-validation to determine the best value of k (1,2,. . .,20) in terms of the mean error rate on the validation set, as suggested by [9]. For the DT model we used the criterion for choosing a split by Gini's diversity index, as suggested by [9]. At the end, for the SVM model we used linear (SVM-L), polynomial (SVM-P), quadratic (SVM-Q) and rbf (SVM-RBF) kernels with the least squares method to find the separating hyperplane, as defined in [2]. For the DELP(BP) model we used the same design process and parameters definition suggested by [27].

3.1 Ripley's Synthetic Problem

The Ripley's synthetic problem [30] consists of samples from two classes. Each sample has 2-dimensional features vector. The data are divided into training and test sets. The training set consists of 250 samples, while the test set consists of 1000 samples. It is worth mentioning that, for both training and test sets, we have the same number of samples belonging to each of the two classes, characterizing a balanced binary classification problem in $\mathbb{R}^2$. The Table 1 presents the experimental results of the test set obtained by the models presented in literature, as well as those achieved by the proposed DELP(PSO) model.

Table 1: Percentage of misclassified patterns of the test set for the Ripley's synthetic problem.

Model	PMP (%)
MLP	9.30
MRLNN	9.50
MP/CL	10.20
SLMP	16.90
FLNN	14.20
FLR	15.30
KNN	9.60
DT	13.30
SVM-L	11.60
SVM-P	9.10
SVM-Q	9.40
SVM-RBF	8.30
DELP(BP)	8.30
DELP(PSO)	8.30

According to Table 1, it is possible to notice that the best model found in the literature is the SVM-RBF and DELP(BP) (with $PMP = 8.30\%$). However, a slightly inferior classification performance can be achieved using SVM-P, MLP, SVM-Q, MRLNN, KNN and MP/CL models. It is worth mentioning that the proposed DELP(PSO) model obtained good classification performance, having the same PMP value obtained by the DELP(BP). The Table 2 presents the PG (test set), in terms of the PMP obtained using DELP(PSO) model and using other models investigated in this work.

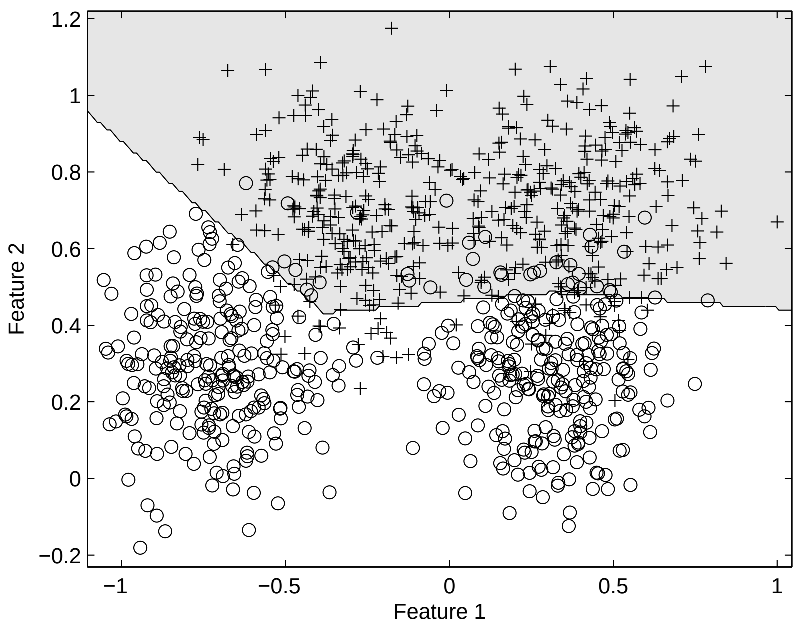
According to the Table 2, without relying on the results obtained with SVM-RBF and DELP(BP) (where the proposed DELP(PSO) model achieved the same classification performance), it is possible to notice that the proposed DELP(PSO) model

Table 2: Percentage gain (test set) of the proposed DELP(PSO) model regarding MLP, MRLNN, MP/CL, SLMP, FLNN, FLR, KNN, DT, SVM-L, SVM-P, SVM-Q, SVM-RBF and DELP(BP) models for the Ripley's synthetic problem.

	PG (%)
DELP(PSO) / MLP	10.75
DELP(PSO) / MRLNN	12.63
DELP(PSO) / MP/CL	18.63
DELP(PSO) / SLMP	50.89
DELP(PSO) / FLNN	41.55
DELP(PSO) / FLR	45.75
DELP(PSO) / KNN	13.54
DELP(PSO) / DT	37.59
DELP(PSO) / SVM-L	28.45
DELP(PSO) / SVM-P	8.79
DELP(PSO) / SVM-Q	11.70
DELP(PSO) / SVM-RBF	0.00
DELP(PSO) / DELP(BP)	0.00

obtained improvement greater than 8% over the results achieved using MLP, MRLNN, MP/CL, SLMP, FLNN, FLR, KNN, DT, SVM-L, SVM-P and SVM-Q models. The decision surface generated by the proposed DELP(PSO) model for the Ripley's synthetic problem is depicted in Figure 2.

Figure 2: Decision surface produced by the proposed DELP(PSO) model for the Ripley's synthetic problem.



3.2 Wisconsin Breast Cancer Problem

The Wisconsin breast cancer problem [31] consists of samples from two classes representing malignant and benignant breast cancer. The data are divided into training and test set, where we used the same partitioning scheme suggested by [9] (the first 249 samples of the benignant class and the first 148 samples of the malignant class are used in the training set and the rest of the samples from both classes are used in the test set). Each sample has 30-dimensional features vector. The Table 3 presents the experimental results of the test set obtained by the models presented in literature, as well as those achieved by the proposed DELP model.

According to Table 3, we can verify that the best model found in the literature is the DELP(BP) (having $PMP = 1.40\%$). However, a slightly inferior classification performance can be found using SVM-L, SVM-Q, MRLNN, SVM-RBF, FLR, MP/CL and MLP. It is possible to notice that the proposed DELP(PSO) model obtained good classification performance (with $PMP = 1.05\%$), overcoming the best model found in the literature. The Table 4 presents the PG (test set), in terms of the PMP obtained using DELP(PSO) model regarding the PMP obtained using other models investigated in this work.

According to the Table 4, we can see that the proposed DELP(PSO) model obtained improvement greater than 25% over

Table 3: Percentage of misclassified patterns of the test set for the Wisconsin breast cancer problem.

Model	PMP (%)
MLP	4.55
MRLNN	2.10
MP/CL	4.20
SLMP	11.89
FLNN	5.59
FLR	3.50
KNN	5.94
DT	8.74
SVM-L	1.75
SVM-P	10.84
SVM-Q	1.75
SVM-RBF	3.15
DELP(BP)	1.40
DELP(PSO)	1.05

Table 4: Percentage gain (test set) of the proposed DELP(PSO) model regarding MLP, MRLNN, MP/CL, SLMP, FLNN, FLR, KNN, DT, SVM-L, SVM-P, SVM-Q, SVM-RBF and DELP(BP) models for the Wisconsin breast cancer problem.

	PG (%)
DELP(PSO) / MLP	76.92
DELP(PSO) / MRLNN	50.00
DELP(PSO) / MP/CL	75.00
DELP(PSO) / SLMP	91.17
DELP(PSO) / FLNN	81.22
DELP(PSO) / FLR	70.00
DELP(PSO) / KNN	82.32
DELP(PSO) / DT	87.99
DELP(PSO) / SVM-L	40.00
DELP(PSO) / SVM-P	90.31
DELP(PSO) / SVM-Q	40.00
DELP(PSO) / SVM-RBF	66.67
DELP(PSO) / DELP(BP)	25.00

the results achieved using MLP, MRLNN, MP/CL, SLMP, FLNN, FLR, KNN, DT, SVM-L, SVM-Q, SVM-P, SVM-RBF and DELP(BP) models.

4 Conclusion

In this paper we presented a swarm-based evolutionary morphological approach for dealing with synthetic and real-world binary classification problems. It is composed of a hybrid neuron based on principles of mathematical morphology and lattice theory, referred to as dilation-erosion-linear perceptron (DELP), with a swarm-based evolutionary learning process, called DELP(PSO), using a particle swarm optimizer (PSO) to design the model.

The classification performance of the proposed DELP(PSO) model was assessed in terms of well-known models presented in the literature (MLP, MRLNN, MP/CL, SLMP, FLNN, FLR, KNN, DT, SVM-L, SVM-P, SVM-Q, SVM-RBF and DELP(BP)) and using the percentage misclassified patterns metric. Besides, two relevant binary classification problem were investigated in this work: Ripley's Synthetic and Wisconsin Breast Cancer. The experimental results demonstrated similar performance (for the Ripley's problem) and better performance (for the Wisconsin Breast Cancer problem) of the proposed DELP(PSO) model in comparison to the better models found in the literature. In other words, the proposed DELP(PSO) model with swarm-based evolutionary learning succeeded to solve the aforementioned classification problems, exhibiting very satisfactory classification results.

Further studies must be developed to better formalize and explain the properties of the proposed DELP(PSO) model and to determine its possible limitations with other binary classification problems. Further studies, in terms of convergence analysis, must be done in the swarm-based evolutionary learning process. Finally, a particular study about the computing complexity and CPU time of the proposed DELP(PSO) model must be done in order to establish a complete cost-performance evaluation of the proposed model.

Acknowledgment

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037-1.03/08.

References

[1] S. Haykin. *Neural networks: A comprehensive foundation*. Prentice Hall, New Jersey, 1998.

[2] S. Haykin. *Neural Networks and Learning Machines*. McMaster University, Canada, 2007.

[3] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[4] P. Maragos. "A Representation Theory for Morphological Image and Signal Processing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 586–599, 1989.

[5] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.

[6] C. Ronse. "Why Mathematical Morphology Needs Complete Lattices". *Signal Processing*, vol. 21, no. 2, pp. 129–154, October 1990.

[7] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, New York, NY, 1994.

[8] G. X. Ritter and G. Urcid. "Lattice Algebra Approach to Single-Neuron Computation". *IEEE Transactions on Neural Network*, vol. 14, no. 2, pp. 282–295, March 2003.

[9] 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, 2011.

[10] G. X. Ritter, P. Sussner and W. B. Hacker. "Associative Memories with Infinite Storage Capacity". In *InterSymp' 97, 9th International Conference on Systems Research Informatics and Cybernetics*, pp. 281–293, Baden-Baden, Germany, 1997. Invited Plenary Paper.

[11] G. X. Ritter, P. Sussner and J. L. D. de Leon. "Morphological Associative Memories". *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 281–293, 1998.

[12] P. Sussner. "Kernels for Morphological Associative Memories". In *Proceedings of the International ICSA/IFAC Symposium on Neural Computation*, pp. 79–85, Vienna, September 1998.

[13] V. Petridis and V. G. Kaburlasos. "Fuzzy lattice neural network (FLNN): a hybrid model for learning". *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 877–890, sep 1998.

[14] V. G. Kaburlasos and V. Petridis. "Fuzzy lattice neurocomputing (FLN) models". *Neural Networks*, vol. 13, no. 10, pp. 1145–1170, 2000.

[15] M. A. Khabou and P. D. Gader. "Automatic target detection using entropy optimized shared-weight neural networks". *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 186–193, January 2000.

[16] A. K. Hocaoglu and P. D. Gader. "Domain learning using Choquet integral-based morphological shared weight neural networks". *Image and Vision Computing*, vol. 21, no. 7, pp. 663–673, 2003.

[17] P. Sussner and M. E. Valle. "Grayscale Morphological Associative Memories". *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 559–570, may 2006.

[18] P. Sussner and M. E. Valle. "Implicative Fuzzy Associative Memories". *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 793–807, 2006.

[19] R. de A. Araújo, F. Madeiro, R. P. de Sousa, L. F. C. Pessoa and T. A. E. Ferreira. "An Evolutionary Morphological Approach for Financial Time Series Forecasting". In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2467–2474, 2006.

[20] R. de A. Araújo, F. Madeiro, T. A. F. Ferreira, R. P. de Sousa and L. F. C. Pessoa. "Improved Evolutionary Hybrid Method for Designing Morphological Operators". In *Proceedings of the IEEE International Conference on Image Processing*, pp. 2417–2420, 2006.

[21] R. de A. Araújo, F. Madeiro, R. P. de Sousa and L. F. C. Pessoa. "Modular Morphological Neural Network Training via Adaptive Genetic Algorithm for Designing Translation Invariant Operators". In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pp. 873–876, May 2006.

[22] P. Sussner and M. E. Valle. "Morphological and Certain Fuzzy Morphological Associative Memories for Classification and Prediction". In *Computational Intelligence Based on Lattice Theory*, edited by V. G. Kaburlassos and G. X. Ritter, volume 67, pp. 149 – 173. Springer Verlag, Heidelberg, Germany, 2007.

[23] M. E. Valle and P. Sussner. "A General Framework for Fuzzy Morphological Associative Memories". *Fuzzy Sets and Systems*, vol. 159, no. 7, pp. 747–768, 2008.

[24] A. M. Silva and P. Sussner. "A Brief Review and Comparison of Feedforward Morphological Neural Networks with Applications to Classification". In *Proceedings of the International Conference on Artificial Neural Networks*, pp. 783–792, 2008.

[25] P. Sussner and E. L. Esmi. "An Introduction to Morphological Perceptrons with Competitive Learning". In *Proceedings of the International Joint Conference on Neural Networks*, pp. 3024–3031, Atlanta, GA, 2009.

[26] P. Sussner and E. L. Esmi. "Constructive morphological neural networks: some theoretical aspects and experimental results in classification". In *Constructive Neural Networks*, edited by J. Kacprzyk, Studies in Computational Intelligence, pp. 123–144. Springer Verlag, Heidelberg, Germany, 2009.

[27] R. de A. Araújo, A. L. I. Oliveira and S. Meira. "A Hybrid Neuron with Gradient-based Learning for Binary Classification Problems". *Encontro Nacional de Inteligência Artificial - ENIA*, 2012.

[28] R. C. Eberhart and J. Kennedy. "A new optimizer using particle swarm theory". In *Proceedings of the International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.

[29] J. Vesterstroem and R. Thomsen. "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems". In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pp. 1980–1987, 2004.

[30] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, United Kingdom, 1996.

[31] A. Asuncion and D. J. Newman. "UCI Machine Learning Repository", 2007.

[32] R. C. Eberhart and X. Hu. "Human tremor analysis using particle swarm optimization". In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1927–1930, Washington, USA, 1999.

[33] F. Vandenbergh and A. P. Engelbrecht. "A Cooperative Approach to Particle Swarm Optimization". *IEEE Trans. Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[34] L. Prechelt. "Proben1: A set of Neural Network Benchmark Problems and Benchmarking Rules". Technical Report 21/94, 1994.

[35] L. F. C. Pessoa and P. Maragos. "Neural networks with hybrid morphological rank linear nodes: a unifying framework with applications to handwritten character recognition". *Pattern Recognition*, vol. 33, pp. 945–960, 2000.

[36] P. Sussner. "Morphological Perceptron Learning". In *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 477–482, Gaithersburg, MD, 1998.

[37] V. G. Kaburlasos, I. N. Athanasiadis and P. A. Mitkas. "Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation". *Int. J. Approx. Reasoning*, vol. 45, no. 1, pp. 152–188, May 2007.

[38] L. Devroye, L. Gyorfi and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

[39] L. Breiman, J. Friedman, R. Olshen and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[40] F. Esposito, D. Malerba and G. Semeraro. "A Comparative Analysis of Methods for Pruning Decision Trees". *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 476–491, 1997.

[41] M. Hagan and M. Menhaj. "Training feedforward networks with the Marquardt algorithm". *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, November 1994.