Error measures based on winner-takes-all and the performance of multilayer perceptron classifier

Michel M. dos Santos
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Pernambuco, Brasil
Email: michelmozinho@gmail.com

Wellington P. dos Santos
Departamento de Engenharia Biomédica
Centro de Tecnologia e Geociências
Universidade Federal de Pernambuco
Pernambuco, Brasil
email: wellington.santos@ieee.org

***Abstract*—The squared error is a measure commonly employed for training neural networks. Alternative objective functions, directed for classification tasks, may be relevant for training neural networks classifiers. The present paper discusses why some types of objective function, used in training neural networks for classification, have shown better performance compared to the usual mean squared error (MSE). The study deals with the concept of winner-takes-all (WTA). For the condition of a trained network, it is demonstrated through a variance inequality that the application of the smooth WTA criterion (softmax) in the output estimates made by a multilayer perceptron (MLP) classifier tends to reduce the variance of such estimates. However, as the softmax approximates the abrupt WTA criterion, there is no variance reduction due to such inequality. On the other hand, considering a network to be trained, error measures based on the composition of MSE with WTA are defined to serve as objective function of particle swarm optimizer in training a MLP for solving logic gates classification. Experimental results point out that the steeper is the WTA criterion, the faster is reaching a solution for the AND gate, moreover, the functions depending on a steeper WTA were superior to the simple MSE. Curiously, in the case of XOR problem, the solution was more often reached when using the pure MSE, whereas most objective functions based on the WTA criterion were not able to achieve a solution before the maximum number of evaluations.**

I. Introduction

In a typical supervised approach for training MLP classifier, the learning algorithm consists in minimizing the MSE using gradient-based methods for adapting the network's weights [1]. In training neural networks, the MSE may not be the most appropriate measure to accomplish specific tasks. Specifically regarding the classification task, a minor MSE does not necessarily corresponds to a lower classification error [2], [3], [4]. Therefore, the optimization of error measures directed to classification tasks may be relevant for training neural networks classifiers.

Some disadvantages of the gradient-based methods are: it can easily get stuck in a local minimum of a multimodal error surface; it is not appropriated for nondifferentiable functions; it is necessary to develop a specific procedure for each type of objective. On the other hand, there is plenty of metaheuristic algorithms that perform global search, do not require gradient information and enable simple and straightforward use of diverse objective function forms without the need to develop particular training procedures [5].

Particle swarm optimization (PSO) is a population based metaheuristic method inspired by collective behaviour of bird flock, in such way it takes advantage of group information. One of the early applications of PSO was on training MLP network [6]. With regard to squared error minimization, PSO converges faster than typical backpropagation, and it achieves good performance compared to genetic algorithms, adaptive chaotic PSO and modifications of backpropagation [7], [8], [9].

In a previous work, we used two alternative objective functions, specific for classification problems, in training a MLP through PSO for solving a realistic problem of pixel classification. Such alternatives, the mean multiclass errors of type I and II, respectively, MTIE and MTIIE, were compared to MSE. The convergence was faster with multiclass errors in relation to MSE usage, when using each one of multiclass errors resulted in a higher classification accuracy with a lower number of evaluations. Moreover, the standard deviation of accuracy was expressively lower during the training process, when were used the multiclass errors [10].

Concerning representation of classification problems, the abrupt winner-takes-all (WTA) is a way for mapping the real-valued outputs to binary symbols representing classes. For the neurons in the last layer, such criterion consists in the neuron of maximum output be mapped to 1 and all other to 0 [11]. A related concept is the softmax function, a smooth version of abrupt WTA criterion, by means of which it is possible interpreting network outputs as probabilities, with the smoothness degree set by a real parameter [12], [13]. Many classification evaluation measures are dependent of the abrupt WTA criterion to discriminate classes, including the multiclass errors and the global accuracy [10]. Objective functions depending on the abrupt WTA are non-differentiable and, therefore, are not suited for using with the typical backpropagation method.

The abilities to solve classification problems (accuracy) and to produce classifiers with lower variance (precision) are the two main issues of present paper. The aim is not to assess training algorithms, but rather to discuss the impact of WTA with different degrees of smoothness on objective functions related to classification tasks. This paper discusses possible reasons why mean multiclass errors, besides other objective functions using abrupt WTA, yielded better accuracy and precision when compared to MSE. The present work involves an analytical study about variances inequality in the condition of a trained network when the smooth WTA is used or not. As also, an experimental approach is carried out to evaluate the

classification accuracy when objective functions with different degrees of smoothness in WTA criterion are employed.

The next sections are organized as follows: the section II contains some basic concepts related to MLP training via PSO for classification tasks. In section III, related works are reviewed. The section IV deal with a variances inequality analysis through derivatives. Section V defines some objective functions that are after employed in an experimental study described in section VI. Experimental evidences are presented in section VII. The conclusions are done in section VIII.

II. Basic Concepts

A. Particle Swarm for Multilayer Perceptron Training

In PSO, the i-th particle is associated to a position vector $\mathbf{x}_i \in \mathbb{R}^n$ and a velocity $v_i \in \mathbb{R}$. By applying PSO algorithm to evolve a neural network, each particle position vector contains neural network's weights and represents a point in the search space. It is considered that the network structure is fixed, and just the network weights are adapted. In this way, each position visited by a particle defines a network setting that is evaluated for all input patterns in training dataset. Thus, the error metric used to evaluate the network performance corresponds to the value of objective function for the respective particle.

During PSO evolution, the update on velocity of a given particle is influenced by global best position $\mathbf{g} \in \mathbb{R}^n$ and its own best position $\mathbf{p}_i \in \mathbb{R}^n$, according to equation

$$v_{i,j} = wv_{i,j} + c_1 r_p(p_{i,j} - x_{i,j}) + c_2 r_g(g_j - x_{i,j}), \quad (1)$$

where w is the inertia weight, c_1 and c_2 are real parameters, r_p and r_g are uniform random number in $[0, 1]$ and indexes i and j indicate, respectively, particles and coordinates in search space [14]. Moreover, positions are updated as follows

$$x_{i,j} = x_{i,j} + v_{i,j}. \quad (2)$$

Some rules of thumb can be used for limiting the particles' velocities and retaining them inside a predefined search space [10], [15].

B. Classification Representation and Multilayer Perceptron

The MLP can be used in supervised classification problems. In this way, input layer receives instances of feature vector $\mathbf{x}$, and the output layer is responsible for indicating the class. The representation of examples can be made according to **1-of-C encoding**. Formally, let $\mathbf{x} \in \mathbb{R}^N$ be an instance of the input data for the MLP, and let $\mathbf{d} = \{d_i; i = 1, \ldots, C\}$ be an indicator vector of label assigned for each class $\mathcal{C}_i$, the 1-of-C encoding is such that:

$$d_i = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{C}_i, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The output layer of the MLP have to correspond to the scheme 1-of-C. Thus, the number of neurons in the output layer is equal to the number of classes. Furthermore, the values of each neuron at output layer should be in the range $[0, 1]$. A differentiable activation function that produces values in such range is usually employed as required by the gradient based optimization methods. Nevertheless, there is a need to employ a discriminant criterion for indicating the output class. A possible choice is the **Winner-Takes-All** (WTA) criterion, according to which is assigned value of 1 (one) to the largest neuron in the output layer indicating the resulting class, and the other outputs receive 0 (zero) [11]. The WTA is related to the bayesian discriminant criterion, whether the output values can be interpreted as *a posteriori* probabilities [16].

III. Related Works

A. Two Categories of Objective Function

It is common to separate objective functions into two types: those based on quadratic error as sum and average of squared errors; and others based on Information Theory, for example, cross-entropy, maximum mutual information and Kullback-Liebler divergence [16], [11], [17].

It was proved that both types of objective functions are able to approximate the output of a MLP network to the *a posteriori* probabilities $P(\mathcal{C}_i|\mathbf{X})$, for each class $\mathcal{C}_i$ given the input random vector $\mathbf{X}$, in classification problems using the 1-of-C encoding [16], [18].

B. The Softmax Function

The softmax function was proposed by Bridle [12] as a manner to permit the interpretation of network outputs as probabilities, since the use of logistic activation by itself does not guarantee that the sum of all outputs is 1 (one). Softmax preserves the order of input values and is a differentiable indicator of maximum, that is, a smooth approximation of the WTA criterion [13]. For a set of variables $\{z_i, i = 1, \ldots, C\}$, the softmax function for each variable is defined as a normalized exponential transformation:

$$f(z_i) = \frac{exp(\alpha z_i)}{\sum_{i=1}^{C} exp(\alpha z_i)}, \quad (4)$$

in which the parameter $\alpha > 0$ is the steepness degree, that is, the larger α, the more abrupt is the function variation [11].

Commonly the softmax activation function is employed with objective functions based on Information Theory. One reason is that, in some cases, the derivative of the objective function, with respect to a given output, leads to the error, that is, the difference between the obtained output and the desired output, this fact is called "natural pairing" of objective function and output unit activation function [13], [11], [19].

C. Evaluations of Squared Error with Softmax

In the following, some works that used the soft version of the WTA are reviewed. In all of them, the logistic activation function was employed in the intermediate layers, and different activation functions were evaluated in the output layer. In summary, the results of these initiatives suggest that the application of softmax is related to higher classification accuracy, less training time and reduction of dispersion in outputs, so increasing precision.

Kline et al. [17] argue that the cross-entropy has the following advantages: non-Gaussian assumption, greater robustness to outliers, it should perform better in estimating relatively

small conditional probabilities. Furthermore, these authors provided experimental evidence that the softmax activation with cross-entropy tends to produce lower values of average absolute deviation compared to logistic activation with squared error.

A marked reduction in training time was found in [20], in the case which the logistic activation function was employed together with relative entropy, compared to the sum of squared error with linear activation. The improved performance was attributed to a steeper error surface found when using relative entropy. A similar observation was made later in [21].

In [19], three different combinations of objective and activation were compared with respect to the mean and variance of classification accuracy. From worst to best performance, the results were: logistic activation and squared error, softmax activation with squared error and softmax activation with cross-entropy.

IV. Variances Inequality

In this section, the aim is analysing some features of objective function able to influence the precision of the obtained classifier. In doing so, it is demonstrated through a variance inequality that by applying the smooth WTA in the estimates made by an already trained MLP, it tends to reduce the variance of these estimates. However, it could be seen that insofar as the abrupt WTA criterion is approximated, there is no variance reduction due to such inequality.

The starting assumption is to consider a network already trained, in which the weights of the MLP are fixed, and the only source of randomness is intrinsic to the data. This is a simplifying assumption, a more intricate approach would be dealing with the random processes involved in the stochastic optimization as made by the particle swarm technique. In this manner, it is defined $\mathbf{X} \in \mathbb{R}^N$, a random vector that determines the distribution of the realization vector $\mathbf{x}$, representing the input data of a MLP. Let $\mathbf{D} = \{D_i; i = 1, \ldots, C\}$ be a random vector representing the desired labels of each class $\mathcal{C}_i$, an indicator such that:

$$D_i = \begin{cases} 1, & \text{if } \mathbf{X} = \mathbf{x} \in \mathcal{C}_i; \\ 0, & \text{otherwise;} \end{cases} \tag{5}$$

in according to the 1-of-C scheme. Also let $\hat{\mathbf{D}} = \{\hat{D}_i; i = 1, \ldots, C\}$ be a random vector consisting in i outputs of a MLP obtained after successful training. The assumption of successful training means that $\hat{D}_i$ is an estimator of D_i, such that $E[\hat{D}_i|\mathbf{X}] \approx E[D_i|\mathbf{X}]$. Note the difference in the fact that the outputs obtained from the network are restricted to a range, that is $\hat{D}_i \in [0, 1]$, which can be assured via logistic function, while D_i can assume only the values $\{0, 1\}$. Observe also that $\hat{\mathbf{D}}$ depends on $\mathbf{X}$ too. Suppose that new data, of the same sort used for training, should be classified. The effect of using the smooth WTA criterion on the variance of $\hat{D}_i$ given such new data $\mathbf{X}$ is the focus of present study.

According to the law of total variance applied to each output estimator:

$$Var(\hat{D}_i) = E[Var(\hat{D}_i|\mathbf{X})] + Var(E[\hat{D}_i|\mathbf{X}]), \tag{6}$$

that rewriting as

$$Var(E[\hat{D}_i|\mathbf{X}]) = Var(\hat{D}_i) - E[Var(\hat{D}_i|\mathbf{X})] \tag{7}$$

and using the fact that variances are positive by definition, we obtain the inequality:

$$E[Var(\hat{D}_i|\mathbf{X})] < Var(\hat{D}_i). \tag{8}$$

Then, it is sufficient reducing the i variances $Var(\hat{D}_i)$ of the outputs produced by the MLP for also decreasing $Var(E[\hat{D}_i|\mathbf{X}])$.

Consider the Theorem 1, proposed by Tang and See [22], which provides a general inequality for variances of functions of a random variable based on derivatives.

Theorem 1. *Let f and g be absolutely continuous functions on an interval $\mathcal{L}$. Thus, $Var([f(X)]) \leq Var([g(X)])$ for any random variable X with $P(X \in \mathcal{L}) = 1$ if and only if f and g satisfy both of the following properties:*

- *$g(x) = g(y)$ implies $f(x) = f(y)$;*
- *$|f'| \leq |g'|$ almost everywhere.*

The first property of Theorem 1 means that f can be written as a function of g. The second property is true if the places of the probability space in which $|f'| \leq |g'|$ does not happen have zero measure. In the following, it is demonstrated the following proposition:

Proposition 1. *By applying the softmax function to the outputs of a MLP, the variance in each output estimator tends to be smaller or equal, that is:*

$$Var\left(\frac{exp(\alpha\hat{D}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{D}_j)}\right) \leq Var(\hat{D}_i). \tag{9}$$

In order to verify the inequality in (9), let be:

$$f(\hat{D}_i) = \frac{exp(\alpha\hat{D}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{D}_j)} \tag{10}$$

and

$$g(\hat{D}_i) = \hat{D}_i. \tag{11}$$

Regarding absolute continuity, it can be shown that if a real-valued function has a bounded first derivative on the interval $[a, b]$, then such function is absolutely continuous on $[a, b]$ (see [23, pg. 288]). From what it is verified that the functions f (10) and g (11) are absolutely continuous.

The first property required by Theorem 1 is satisfied, since f (10) is a function of g (11). For verifying the second property, consider the derivatives of $g(\hat{d}_i)$ and $f(\hat{d}_i)$, respectively, $g' = 1$ and

$$f' = \alpha\frac{exp(\alpha\hat{d}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{d}_j)}\left(1 - \frac{exp(\alpha\hat{d}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{d}_j)}\right). \tag{12}$$

For clarity, the function f' is renamed such that $f'(\hat{d}_i) = h(\hat{d}_i)$. For obtaining the point where h is maximum, it is carried out the first derivative test of h:

$$h' = \alpha(\alpha+1)\left(\frac{exp(\alpha\hat{d}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{d}_j)}\right)^3 - \alpha(\alpha+2)\left(\frac{exp(\alpha\hat{d}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{d}_j)}\right)^2 + \alpha\frac{exp(\alpha\hat{d}_i)}{\sum_{j=1}^{C} exp(\alpha\hat{d}_j)}, \tag{13}$$

taking $z = exp(\alpha\hat{d}_i)/\sum_{j=1}^{C} exp(\alpha\hat{d}_j)$ and rewriting (13), it is obtained:

$$h'(z) = \alpha z(z - \frac{1}{\alpha+1})(z-1), \tag{14}$$

an equation with roots $\{0, (\alpha+1)^{-1}, 1\}$. Moreover, h is increasing on $\left]0, (\alpha+1)^{-1}\right[$, because $h' > 0$, and decreasing on $\left](\alpha+1)^{-1}, 1\right[$, since $h' < 0$, what implies that $z = 1/(\alpha+1)$ is the maximum point of h. Replacing $e^{\alpha\hat{d}_i}/\sum_{j=1}^{C} e^{\alpha\hat{d}_j} = 1/(\alpha+1)$ in f' (12), the value at the maximum point is

$$\frac{\alpha^2}{(\alpha+1)^2}, \tag{15}$$

what for α non-negative and finite is less that $g' = 1$ □

In the condition in which the softmax function approximates the abrupt WTA criterion, that is, for $\alpha \to \infty$, the absolute continuity condition is not satisfied. However, as α is increased, the expression (15) tends to 1, closer and closer to $g' = 1$. Thus, the more softmax approaches the abrupt WTA criterion, the greater is the variance on the outputs. Then, by using the abrupt WTA criterion there is no variance reduction due to such inequality, whereas a smooth WTA criterion may reduce the variance on outputs.

The previously demonstrated inequality is referent to the condition of fixed neuron weights and randomness present only in the data. In the following sections, the neuron weights are adapted in the training process by PSO and the classifier performance is experimentally assessed.

V. Mean Squared Error with Winner-Takes-All

Let $\{\hat{d}_i, i = 1, \ldots, C\}$ be the output estimates of a MLP, then the abrupt WTA criterion can be defined according to function:

$$s_i(\hat{d}_i) = \begin{cases} 1, & \text{if } \hat{d}_i = max(\hat{d}_1, \ldots, \hat{d}_C) \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Including to notation the indices $\{j = 1, \ldots, M\}$ for a set of examples being evaluated, the mean squared binarized error (MSBE) is defined as:

$$\sum_{j=1}^{M}\sum_{i=1}^{C}(s_i(\hat{d}_{ij}) - d_{ij})^2. \tag{17}$$

In expression (17), if instead of abrupt WTA, it is used the softmax function with parameter α, the above equation is referred to as Mean Squared Error with Softmax (MSES-α). Note that, for the MSES-α the derivative in relation to each $\hat{d}_{ij}$ is proportional to α, as can be obtained after applying the chain rule. Thus, the closer to the abrupt WTA, the steeper is the error surface in the direction of each output estimate.

TABLE I. Logic Gates with Two Outputs

Input		Output					
		AND			XOR		
		n_1	n_2	class	n_1	n_2	class
0	0	0	1	$\mathcal{C}_0$	0	1	$\mathcal{C}_0$
1	0	0	1	$\mathcal{C}_0$	1	0	$\mathcal{C}_1$
0	1	0	1	$\mathcal{C}_0$	1	0	$\mathcal{C}_1$
1	1	1	0	$\mathcal{C}_1$	0	1	$\mathcal{C}_0$

VI. Experimental Setting

In the experiment, MLP networks were trained via PSO for solving generic classification problems, more precisely, the mapping of input/output represented by logic gates. The objective functions evaluated were MSE, MTIE[1], MSBE and MSES-α for different parameter values. Two neurons on the output layer were employed, for making possible to use the abrupt WTA criterion as above defined (16). The truth table for logic gates with two output was defined as follows: the first output neuron n_1 is equal to the case of a single output; while the second neuron n_2 is the negation of the first. See the Table I for AND and XOR logic gates used in the experiment.

The MLP network structure consisted in: two input nodes, two neurons with logistic activation in both hidden and output layers. In PSO, 20 particles were used, the other parameters were $w = 1$, $c_1 = 2$, $c_2 = 2$, the search space in each coordinate was limited to the interval $[-12, 12]$. The maximum velocity was kept in the range $[-1.5, 1.5]$. When some particle left the search space, it was repositioned along each coordinate in the opposite direction by the sum of an uniform random variable between 0 and 60% of the exceeded limit value. A modified version of an independent source code[2] was used as a way for verifying the reproducibility of our previously published results involving mean multiclass errors. After testing some values, MLP and PSO parameters were maintained as was originally in the published source code, since that the tested values did not change results significantly. The objective function is coded as a fitness, defined as the reciprocal of error plus a small constant and, therefore, maximized. The latter fact does not change the overall reasoning since the problem is equivalent to minimizing an error measure.

The training process was repeated 30 times for statistical evaluation. The stopping criterion for training was to solve the input/output problem represented by logic gates, or reaching 4000 evaluations of the objective function. For checking if the problem has been resolved, the abrupt WTA criterion is applied to network's outputs, regardless of objective function optimized. The objective functions were compared with respect to the number of evaluations until stopping the training. The

[1]The MTIE and MTIIE are equivalent for binary classification

[2]Original version was available at http://www.lwebzem.com/cgi-bin/courses/course_view.cgi?m=c10_m2_s3.html&user_id=&c=particle_swarm_optimization

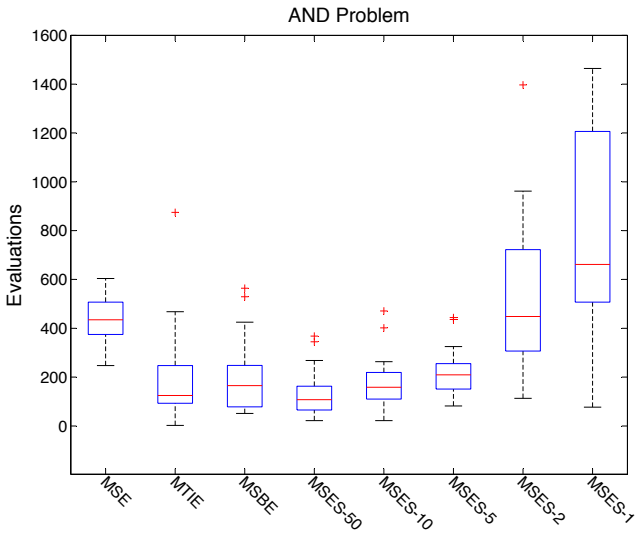

Fig. 1. Distribution of evaluations in the AND problem

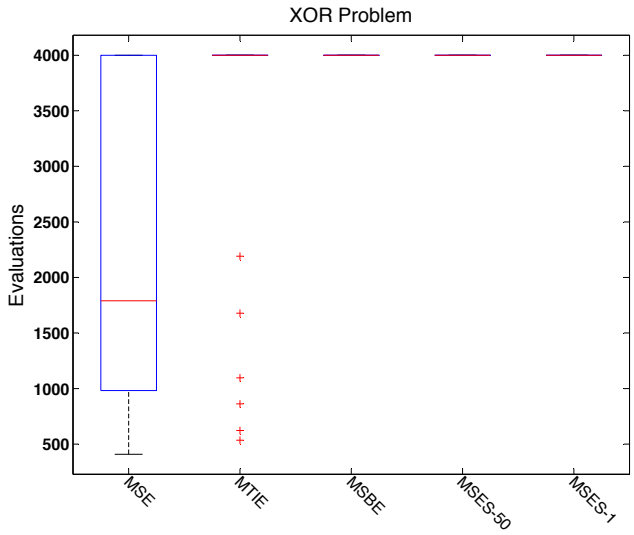

Fig. 2. Distribution of evaluations in the XOR problem

Wilcoxon's rank sum test was carried out to test the null hypothesis of that the samples represent the same distribution [24], with a significance level of 5% was used.

VII. Results and Discussion

In Figure 1 is presented the distribution of evaluations until stopping training, referring to AND problem for several objective functions. In relation to objective functions depending on abrupt WTA, the MSBE was not different to MTIE, and consistently there was no rejection of the null hypothesis with p-value $= 0.6573$. In addition, with MSBE the number of evaluations tended to be lower than with MSE only, and this difference was significant p-value $= 2.1012 \times 10^{-8}$. As regards the MSES, there was a tendency that as the parameter α was increased, a smaller number of evaluations has been spent, moreover, it is possible to note a reduction in the variance (lower interquartile range), especially from MSES-2 to MSES-5. The best performance when increasing the α values, can be explained in terms of the derivative of MSES in direction of $\hat{d}_{ij}$, that is proportional to the α parameter , indicating that the steeper the error surface, the more easily a solution was reached in the training process.

For the XOR gate, the distribution of evaluations is presented in Figure 2. Most objective functions using the WTA criterion was not able to achieve a solution before the maximum number of evaluations, except for the MTIE that reached the solution a few times. On the other hand, the MSE reached a solution in many cases. It is not clear at this time why the use of objective functions based on WTA has made more difficult solving the XOR problem than with MSE.

Compared to other works with respect to efficiency for the most common case of MSE optimization for solving the XOR problem, the version of PSO employed may not be the best method, it spent on average 2214 epochs (evaluations of training set). An earlier work employing the original PSO reported the resolution of XOR problem with just 9 epochs on average [25]. In another work, alternative PSO versions were able to solve the XOR problem in dozens of evaluations, while the basic PSO was not even able to converge [26]. In [27], a variety of backpropagation methods were related to results ranging from 24 to 3495 epochs, and the author highlighted that the stopping criterion used may greatly influence the obtained number of epochs.

VIII. Conclusion

It was studied the influence of using the WTA criterion in the classification performance of MLP network. In the situation of a trained network, with randomness present only in input examples, it was demonstrated a tendency of variance reduction on the output estimates when applying the smooth WTA criterion, but such reduction is lost insofar that the abrupt WTA is approximated. On the other hand, a network to be trained with an objective function that depends on a steep WTA criterion tends to reach a solution faster than with a simple MSE for the AND problem. For the XOR problem, the pure MSE was the best objective seeing that often it gives a solution, whereas the objective functions based on the WTA criterion remarkably failed.

Assuming that an unbiased estimator of higher variance would be preferred over a biased estimator of lower variance. This could lead to choose either an objective function that depends on a steep WTA criterion for problems like the AND gate classification, or the MSE as objective for the XOR problem. In ultimate analysis, what is the best objective function to use is still very data dependent.

Future works may evaluate other PSO versions in order to reduce the number of epochs spent. More experiments involving other datasets are also required, for verifying if results involving the WTA based errors maintain the observed pattern. The variance inequality found may be relevant in a more general context, so it must be further studied and supported by simulation examples.

Acknowledgment

The authors are grateful to CAPES-Brazil by the financial support to develop this work.

REFERENCES

[1] S. Haykin, *Neural networks*. Practice-Hall Press, New Jersey, 1999.

[2] J. B. Hampshire, A. H. Waibel *et al.*, "A novel objective function for improved phoneme recognition using time-delay neural networks," *Neural Networks, IEEE Transactions on*, vol. 1, no. 2, pp. 216–228, 1990.

[3] M. F. Møller, "Efficient training of feed-forward neural networks," Ph.D. dissertation, Aarhus University, 1993.

[4] L. M. Silva, J. M. de Sá, and L. A. Alexandre, "Data classification with multilayer perceptrons using a generalized error function," *Neural Networks*, vol. 21, no. 9, pp. 1302 – 1310, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608008000749

[5] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423 –1447, sep 1999.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov 1995, pp. 1942–1948.

[7] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 2003, pp. 110–117.

[8] E. A. Grimaldi, F. Grimaccia, M. Mussetta, and R. Zich, "Pso as an effective learning algorithm for neural network applications," in *Computational Electromagnetics and Its Applications, 2004. Proceedings. ICCEA 2004. 2004 3rd International Conference on*. IEEE, 2004, pp. 557–560.

[9] A. Cerasa, E. Bilotta, A. Augimeri, A. Cherubini, P. Pantano, G. Zito, P. Lanza, P. Valentino, M. C. Gioia, and A. Quattrone, "A cellular neural network methodology for the automated segmentation of multiple sclerosis lesions," *Journal of neuroscience methods*, vol. 203, no. 1, pp. 193–199, 2012.

[10] M. M. dos Santos, M. J. Valença, and W. P. dos Santos, "Mean multiclass type i and ii errors for training multilayer perceptron with particle swarm in image segmentation," in *Intelligent Data Engineering and Automated Learning-IDEAL 2012*. Springer, 2012, pp. 135–142.

[11] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Oxford University Press, USA, jan 1996. [Online]. Available: http://www.worldcat.org/isbn/0198538642

[12] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing - Algorithms, Architectures and Applications*, ser. NATO ASI Series, F. F. Soulié and J. Hérault, Eds. Springer Berlin Heidelberg, 1990, vol. 68, pp. 227–236. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-76153-9_28

[13] ——, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann Publishers Inc., 1990, pp. 211–217.

[14] R. C. Eberhart and Y. Shi, *Computational Intelligence - Concepts to Implementations*. Elsevier, 2007.

[15] M. Zamani and A. Sadeghian, "A variation of particle swarm optimization for training of artificial neural networks," *Computational Intelligence and Modern Heuristics*, 2010.

[16] J. B. Hampshire and B. Pearlmutter, "Equivalence proofs for multi-layer perceptron classifiers and the bayesian discriminant function," in *Proc. of the 1990 Connectionist Models Summer School*, 1990, pp. 1–17.

[17] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005. [Online]. Available: http://dx.doi.org/10.1007/s00521-005-0467-y

[18] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural computation*, vol. 3, no. 4, pp. 461–483, 1991.

[19] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 181*, vol. 185, 1997.

[20] S. A. Solla, E. Levin, and M. Fleisher, "Accelerated learning in layered neural networks," *Complex Systems*, vol. 2, no. 6, pp. 625–639, Dec. 1988. [Online]. Available: http://dl.acm.org/citation.cfm?id=65512.65513

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS?10). Society for Artificial Intelligence and Statistics*, 2010.

[22] H.-K. Tang and C.-T. See, "Variance inequalities using first derivatives," *Statistics & Probability Letters*, vol. 79, no. 9, pp. 1277–1281, 2009.

[23] J. Yeh, *Real analysis: theory of measure and integration*. World Scientific Publishing Company, 2006.

[24] M. P. Fay and M. A. Proschan, "Wilcoxon-Mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules." *Statistics surveys*, vol. 4, no. 0, pp. 1–39, 2010. [Online]. Available: http://dx.doi.org/10.1214/09-ss051

[25] J. Salerno, "Using the particle swarm optimization technique to train a recurrent neural model," in *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*. IEEE, 1997, pp. 45–49.

[26] G. Yang, D. Zhang, and X. Hu, "A learning algorithm based on pso and l-m for parity problem," in *Stochastic Control*, C. Myers, Ed. InTech, 2010, pp. 151–166.

[27] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," *Computer Science Department, Carnegie Mellon University*, 1988.