Neural Network Classification: Maximizing Zero-Error Density*

Luís M. Silva[1,2], Luís A. Alexandre[1,3], and J. Marques de Sá[1,2]

[1] INEB -Instituto de Engenharia Biomédica, Lab. de Sinal e Imagem Biomédica, Campus da FEUP, Rua Dr. Roberto Frias, 4200 - 465 Porto - Portugal
[2] Faculdade de Engenharia da Universidade do Porto - DEEC, Rua Dr. Roberto Frias, 4200 - 465 Porto - Portugal
[3] IT - Networks and Multimedia Group, Covilhã - Portugal

Abstract. We propose a new cost function for neural network classification: the error density at the origin. This method provides a simple objective function that can be easily plugged in the usual backpropagation algorithm, giving a simple and efficient learning scheme. Experimental work shows the effectiveness and superiority of the proposed method when compared to the usual mean square error criteria in four well known datasets.

1 Introduction

The work by Príncipe and co-workers [1,2], proposes the use of information measures such as entropy as cost functions for adaptive systems, which are expected to deal better with high-order statistical behaviours than the usual mean square error (MSE). In particular, they proposed the minimization of the error (difference between the output and the target of the system) entropy. The idea is simple. Minimizing the error entropy is equivalent to minimizing the distance between the probability distributions of the target and system outputs [1]. Thus, the system is learning the target variable. The particular application to neural network classification with Rényi's entropy of order $\alpha = 2$ [3,4] and Shannon's entropy [5] has been sucessful. We propose a different but related procedure. The minimization of error entropy is basically inducing a Dirac distribution (the minimum entropy distribution) on the errors. It has been shown that under mild conditions, this Dirac can be centered at zero [3] and thus the error is made to converge to zero. For this reason, we propose to update the weights of a classification neural network by maximizing the error density at the origin. As we will see, this procedure provides a simple objective function and with no need for integral estimation as in other approaches.

* This work was supported by the Portuguese FCT-Fundação para a Ciência e a Tecnologia (project POSI/EIA/56918/2004). First author is also supported by FCT's grant SFRH/BD/16916/2004.

S. Singh et al. (Eds.): ICAPR 2005, LNCS 3686, pp. 127–135, 2005.

2 The Zero-Error Density Maximization Procedure

Consider a multi-layer perceptron (MLP) with one hidden layer, a single output y and a two-class target variable (class membership for each example in the dataset), t. For each example we measure the (univariate) error $e(n) = t(n) - y(n),\ n = 1, \ldots, N$ where N is the total number of examples. As discussed above, the minimization of the error entropy induces a Dirac distribution on the errors. It can also be seen that when encoding the classification problem such that $t \in \{-a, a\}$ and $y \in [-a, a]$ for $a > 0$, the induced Dirac distribution must be centered at the origin and thus the error is made to converge to zero [3]. Hence, adapting the system to minimize the error entropy is equivalent to adjusting the network weights in order to concentrate the errors, giving a distribution with a higher peak at the origin. This reasoning leads us to the adaptive criteria of maximizing the error density value at the origin. Formaly,

$$\mathbf{w} = \arg\max_{\mathbf{z}} f(0; \mathbf{z}) \tag{1}$$

where $\mathbf{w}$ is the weight vector of the network and f is the error density. We denote this principle as Zero-Error Density Maximization (Z-EDM). As the error distribution is not known, we rely on nonparametric estimation using Parzen windowing

$$\hat{f}(e) = \frac{1}{Nh} \sum_{n=1}^{N} K\left(\frac{e - e(n)}{h}\right) \tag{2}$$

and the Gaussian kernel

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \quad . \tag{3}$$

This is a common and useful choice, because it is continuously differentiable, an essential property when deriving the gradient of the cost function. Hence, our new cost function for neural network classification becomes

$$\hat{f}(0) = \frac{1}{Nh} \sum_{n=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{e(n)^2}{h^2}\right) \quad . \tag{4}$$

3 Backpropagating the New Criterion

3.1 Determining the gradient

As we will see, the new criterion can easily substitute MSE in the backpropagation algorithm.

If w is some network weight then

$$\begin{aligned} \frac{\partial \hat{f}(0)}{\partial w} &= \frac{1}{Nh} \sum_{n} \frac{1}{\sqrt{2\pi}} \frac{\partial}{\partial w} \exp\left(-\frac{1}{2}\frac{e(n)^2}{h^2}\right) \\ &= -\frac{1}{Nh^3} \sum_{n} K\left(\frac{0 - e(n)}{h}\right) e(n) \frac{\partial e(n)}{\partial w} \quad . \end{aligned} \tag{5}$$

Basically one has

$$\frac{\partial \hat{f}(0)}{\partial w} = \sum_n a(n)e(n)\frac{\partial e(n)}{\partial w} \tag{6}$$

with

$$a(n) = -\frac{1}{Nh^3}K\left(\frac{0-e(n)}{h}\right) \quad .$$

For the case of MSE $a(n) = 1, \forall n$. The computation of $\frac{\partial e(n)}{\partial w}$ is as usual for the backpropagation algorithm. Note that the procedure is easily extended for multiple output networks. Taking a target encoding for class $\mathcal{C}_k$ as $[-1, \ldots, 1, \ldots, -1]$ where the 1 appears at the k-th component and using the multivariate Gaussian kernel with identity covariance, the gradient is straightforward to compute

$$\frac{\partial \hat{f}(\mathbf{0})}{\partial w} = -\frac{1}{Nh^{M+2}}\sum_{n=1}^{N} K\left(\frac{\mathbf{0}-\mathbf{e}(n)}{h}\right)\sum_{k=1}^{M} e_k(n)\frac{\partial e_k(n)}{\partial w} \tag{7}$$

where M is the number of output units and $\mathbf{e}(n) = (e_1(n), \ldots, e_M(n))$. Having determined (7) for all network weights, the weight update is given, for the m-th iteration, by the gradient ascent (we are maximizing) rule

$$w^{(m)} = w^{(m-1)} + \eta\frac{\partial \hat{f}(\mathbf{0})}{\partial w} \quad .$$

3.2 Choice of η and h

The algorithm has two parameters that one should optimally set: the smoothing parameter, h, of the kernel density estimator (3) and the learning rate, η. As already seen in previous work [4,5] we can benefit from an adaptive learning rate procedure. By monitoring the value of the cost function, $\hat{f}(\mathbf{0})$, the adaptive procedure ensures a fast convergence and a stable training. The rule is given by

$$\eta^{(m)} = \begin{cases} u\,\eta^{(m-1)} & \hat{f}(\mathbf{0})^{(m)} \geq \hat{f}(\mathbf{0})^{(m-1)} \\ d\,\eta^{(m-1)} \wedge restart\ otherwise & \end{cases} \quad , 0 < d < 1 \leq u \quad .$$

If $\hat{f}(\mathbf{0})$ increases from one epoch to another, the algorithm is in the right direction, so η is increased by a factor u in order to speedup convergence. However, if η is large enough to decrease $\hat{f}(\mathbf{0})$, then the algorithm makes a *restart* step and decreases η by a factor d to ensure that $\hat{f}(\mathbf{0})$ is being maximized. This *restart* step is just a return to the weights of the previous epoch.

Although an exhaustive study of the behaviour of the performance surface has not been made yet (this is a topic for future work), we believe that the smoothing parameter h has a particular importance in the convergence success. Just as in the case of entropy, the "dilatation property" mentioned in [2] may also occur. If h is increased to infinity, the local optima of the cost function disappears, letting an unique but biased global maximum to be found. Also note that, as

training evolves, it is expected that the errors $\mathbf{e}(n)$ get concentrated around $\mathbf{0}$. Hence, we may benefit from an adaptive rule that starts with a high value of h that is decreased as training evolves. Clearly, this rule should be based on some measure of the local behaviour of the cost function or the gradient. However, we have not yet been successful with this adaptation rule and we postpone this objective as future work. The strategy was then to perform experiments with some fixed h and choose the best ones.

4 Experimental Results

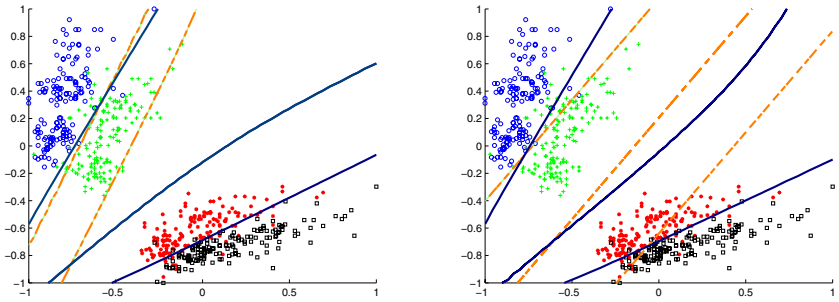
4.1 Convergence Capacity in a Vowel Discrimination Problem

In the first experiment we evaluated the convergence capacity of several MLP's (2, 6 and 10 hidden units) trained using Z-EDM and MSE cost functions, when applied to a vowel discrimination problem. The data, designated PB12, contains 608 examples produced by 76 speakers measuring the first and second formants of the vowels i, I, a and A [6]. The MLP's were trained 100 times with the whole dataset and a convergence success was counted whenever the final training error was below 9%. We varied the number of training epochs, initial learning rate η and smoothing parameter ($h = 2$ and 5) in the case of Z-EDM. Table 1 shows the convergence success rates for Z-EDM and MSE. Below these values, the mean training errors and standard deviations (over the 100 repetitions) are presented. In this Table and in the following, *hid* stands for the number of hidden units.

Table 1. Convergence success rates in 100 repetitions of different MLP's trained with Z-EDM and MSE. Below are the mean training errors and standard deviations.

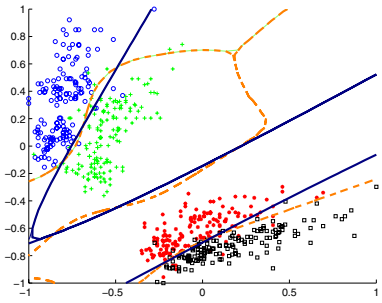
hid	2		6		10	
epochs	Z-EDM	MSE	Z-EDM	MSE	Z-EDM	MSE
200	71%	6%	100%	87%	100%	90%
	9.54(4.38)	37.9(21.1)	7.31(0.19)	9.78(8.26)	7.22(0.08)	9.11(8.88)
500	96%	21%	100%	97%	100%	99%
	7.61(2.05)	28.6(18.3)	6.62(0.28)	7.77(6.83)	6.58(0.22)	6.61(4.72)
1000	99%	38%	100%	96%	100%	100%
	7.51(2.03)	20.7(14.8)	6.07(0.21)	7.80(7.21)	6.14(0.24)	5.83(0.29)

The results of Table 1 show that the proposed method is clearly more powerful in classifying this dataset. In fact, we encounter already a very good performance for the case of 2 hidden units, while MSE has a global poor performance. By inspecting the training errors and standard deviations, we also find a higher stability of Z-EDM. We've also noted that Z-EDM was not influenced by the initial value of the learning rate, while MSE became very unstable for very high values of η. For 2 hidden units and 200 training epochs, Z-EDM preferred $h = 5$ while for higher training epochs $h = 2$ worked better. This can be related to the smoothness of the performance surface and the dilatation property mentioned

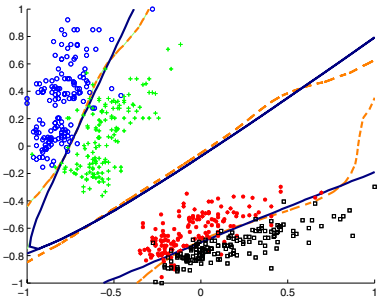


(a) $hid = 2$, $epochs = 200$

(b) $hid = 2$, $epochs = 500$



(c) $hid = 10$, $epochs = 200$



(d) $hid = 10$, $epochs = 500$

Fig. 1. Decision boundaries for PB12. Solid dark line was obtained with Z-EDM and dashed light line with MSE.

earlier. With a small h and consequently a less smoother surface, the number of training epochs (200) may not be sufficient in most cases. This can be surpassed by increasing h at a cost of biasing the optimal solution. Thus the results of Table 1 were obtained with an initial $\eta = 0.5$ and $h = 2$ except for $epochs = 200$ where $h = 5$.

Figure 1 shows decision boundaries obtained with Z-EDM and MSE in different situations. The top figures were obtained with $hid = 2$ and the bottom with $hid = 10$; the left figures used $epochs = 200$ and the right ones $epochs = 500$. The figures show evidence of the stability of Z-EDM and the poor performance of MSE for $hid = 2$. Also, we encounter a higher adaptation of MSE decision lines to the data for $hid = 10$, which can be a drawback in terms of generalization.

4.2 Evaluating the Generalization Ability

To evaluate the generalization ability of MLP's trained with Z-EDM, we conducted a train and test procedure with PB12 and three other datasets taken from the UCI repository [7]. Two of these datasets are from medical applications. WDBC is concerned with the diagnosis problem between benign and malignant

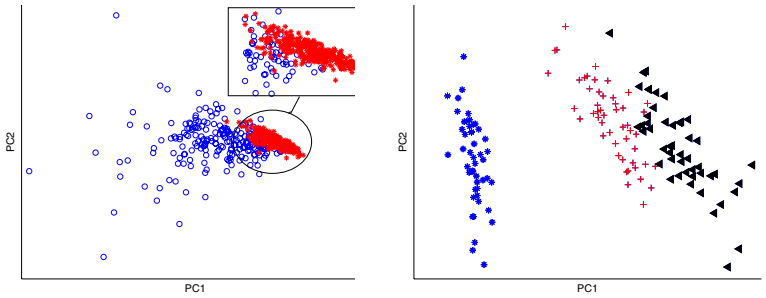


Fig. 2. Projection of WDBC (*left*) and IRIS (*right*) onto the first two principal components

Table 2. Description of the four datasets used in the experiments. The last column reports the number of training epochs used for each dataset.

Datasets	#Instances	#Features	#Classes	#Train epochs
PB12	608	2	4	500
WDBC	569	30	2	40
PIMA	768	8	2	45
IRIS	150	4	3	90

breast cancer and PIMA deals with the diagnostic of diabetes according to the World Health Organization. The fourth dataset is the well known IRIS created by R. A. Fisher. Table 2 gives a brief description of the four datasets.

Figure 2 shows WDBC (left) and IRIS (right) projected onto the first two principal components. As we can see, WDBC has a simple structure and low complexity MLP's should be sufficient to achieve good results. The projection of IRIS shows that one of the classes is linearly separable from the other two, while the latter are not. Note that this is a class structure very similar to the one encountered for PB12 (see Fig. 1).

The following procedure was performed 50 times: divide the data in two subsets, half for training and half for testing; train the network and compute the test set error; interchange the roles of the training and test sets; perform training and test again. The number of training epochs used is reported in Table 2 for each dataset. This procedure was applied to several MLP's varying the number of hidden units from 2 to 20.

Table 3 shows the mean test errors and standard deviations (in brackets). The results of PB12 confirm the previous experiments. For a low number of hidden units, MSE fails to converge in most cases, giving higher mean test errors and standard deviations. From Fig. 3(a), where a more complete set of results is presented, we can see that only for $hid = 11, 19$ and 20, MSE performs equally to Z-EDM. Thus, Z-EDM reveals more stability contrasting with the high dependency of MSE on the number of hidden units. In WDBC, Z-EDM clearly

Table 3. Test error rates (%), standard deviations (in brackets) and p-values for the Mann-Whitney test of the train and test procedure for several MLP's, trained with Z-EDM and MSE. The right column presents the best results.

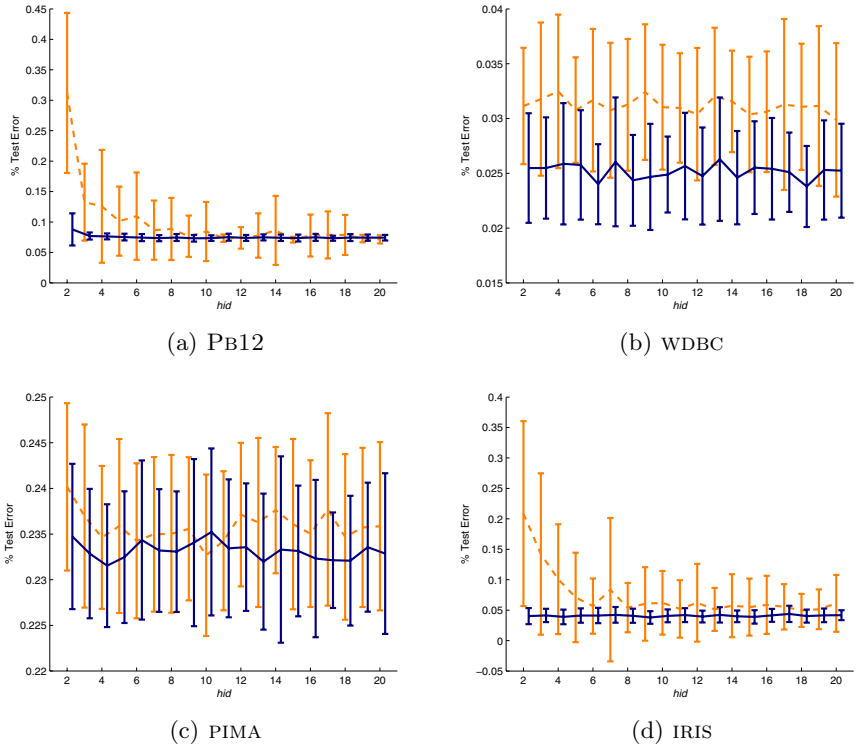
PB12	2	5	6	11	20	Best
Z-EDM	8.79(2.64)	7.53(0.56)	7.44(0.58)	7.51(0.58)	7.42(0.47)	7.32(0.53)→ $hid = 9$
MSE	31.2(13.2)	10.1(5.68)	11.0(7.18)	7.34(0.62)	7.14(0.67)	7.10(0.48)→ $hid = 15$
p-value	0.000	0.052	0.020	0.227	0.012	0.054

WDBC	2	3	4	5	6	Best
Z-EDM	2.55(0.50)	2.55(0.46)	2.50(0.55)	2.58(0.50)	2.40(0.37)	2.38(0.37)→ $hid = 18$
MSE	3.11(0.53)	3.18(0.70)	3.25(0.70)	3.08(0.48)	3.17(0.65)	2.99(0.88)→ $hid = 20$
p-value	0.00	0.00	0.00	0.00	0.00	0.00

PIMA	2	4	6	8	10	Best
Z-EDM	23.5(0.80)	23.2(0.67)	23.4(0.87)	23.3(0.66)	23.5(0.91)	23.2(0.67)→ $hid = 4$
MSE	24.0(0.91)	23.5(0.78)	23.4(0.95)	23.5(0.86)	23.3(0.88)	23.3(0.88)→ $hid = 10$
p-value	0.002	0.027	0.761	0.346	0.191	0.569

IRIS	2	6	9	13	19	Best
Z-EDM	4.02(1.32)	4.12(1.26)	3.80(1.04)	4.23(1.26)	4.15(1.13)	3.80(1.04)→ $hid = 9$
MSE	20.9(15.2)	5.67(4.54)	6.02(6.04)	5.12(3.53)	5.15(3.26)	4.96(2.72)→ $hid = 18$
p-value	0.000	0.091	0.003	0.265	0.094	0.001

outperforms MSE. All the tested MLP's for this dataset achieved better results than the ones trained with MSE. This behaviour is evident from Fig. 3(b). In what concerns PIMA, Fig. 3(c) shows that the mean test error line for Z-EDM is mostly below the one from MSE, although the differences are not as high as in the previous datasets. For example, with $hid = 6$ both methods achieve the same test error. It was also interesting to evaluate the behaviour of the train and test procedure in the IRIS dataset. As expected, we found similar results as in PB12 (see Fig. 3(d)). For all tested MLP's, MSE had difficulties in finding consistently the best solutions. This was not encountered for Z-EDM, which gave stable results along the various values of *hid*. e significance of the differences encountered in the results, we performed statistical tests of two types. The parametric t test for two independent samples and the corresponding nonparametric test of Mann-Whitney. The second one, which is a test for the equality in locations of the two samples, is preferable because it does not rely on distributional assumptions and is more robust to outliers. Nevertheless, the results found for both tests were quite similar. Hence, we opted to show the results for the Mann-Whitney test where p-values below 0.05 show evidence of different locations for the test error distributions coming from MSE and Z-EDM. Except for WDBC, where all the results of Z-EDM are significantly better, we found that when the number of hidden units increases, the differences tend to be not significant. However, we have to take some care while evaluating the p-values, because the existence of (many) strong outliers may lead to wrong conclusions (see for example in PB12 with $hid = 5$). For low complexity MLP's, MSE is clearly outperformed by



(a) Pb12 (b) wdbc

(c) pima (d) iris

Fig. 3. Errorbar plot for the test results of the four datasets. Dark solid line was obtained by Z-EDM and light dashed line by MSE. The dark line is slightly shifted to the right for better viewing. Vertical bars from the mean represent one standard deviation.

Z-EDM. This can also be seen on the right column of Table 3, where the best results are presented for each method.

5 Conclusion

We propose a neural network classification method using the error density at the origin as the adaptive criterion. This leads to a simple objective function that can be easily used with the usual backpropagation algorithm. It can be seen as a kind of weighted mean square error, but where information about the error distribution at the origin is taken into account when updating the network's weight vector. The method was evaluated in four datasets and compared to the usual mean square error. We found that Z-EDM was more stable and less dependent on the number of hidden units. The capacity of consistently finding the best solutions was higher for Z-EDM, mainly for small complexity MLP's. It also had a better performance in predicting unseen patterns. Several questions

have to be further studied, in particular the relation between the behaviour of the performance surface and the kernel smoothing parameter. This study may also provide insights for an adaptive rule for h. These experiments will be extended to a more general set of benchmark datasets to evaluate generalization ability and to make comparisons with other methods.

References

1. D. Erdogmus and J. C. Principe.: An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems. IEEE Transactions on Signal Processing, 50(7):1780–1786, 2002.
2. D. Erdogmus and J. Principe.: Generalized information potential criterion for adaptive system training. IEEE Transactions on Neural Networks, 13(5):1035–1044, 2002.
3. J.M. Santos, L.A. Alexandre, and J. Marques de Sá.: The Error Entropy Minimization Algorithm for Neural Network Classification. In Int. Conf. on Recent Advances in Soft Computing, Nottingham, United Kingdom, 2004.
4. J.M. Santos, L.A. Alexandre, and J. Marques de Sá.: Optimization of the Error Entropy Minimization Algorithm for Neural Network Classification. In C. Dagli, A. Buczak, D. Enke, M. Embrechts, and O. Ersoy, editors, Intelligent Engineering Systems through Artificial Neural Networks, volume 14, pages 81–86. ASME Press Series, 2004.
5. L.M. Silva, J. Marques de Sá, and L.A. Alexandre.: Neural Network Classification using Shannon's Entropy. In European Symposium on Artificial Neural Networks, 2005.
6. R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton.: Adaptive mixtures of local experts. Neural Computation, 3:79–87, 1991.
7. C.L. Blake and C.J. Merz.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.