# Neural Network Classification using Error Entropy Minimization

Jorge M. Santos[1,3], Luís A. Alexandre[2] and Joaquim Marques de Sá[1]

[1] INEB - Instituto de Engenharia Biomédica, Porto, Portugal
[2] IT - Networks and Multimedia Group, Covilhã
[3] Instituto Superior de Engenharia do Porto, Dep. Matemática, Porto, Portugal
(jms@isep.ipp.pt)

**Abstract.** One way of using the entropy criteria in learning systems is to minimize the entropy of the error between two variables: typically, one is the output of the learning system and the other is the target. This framework has been used for regression. In this paper we show how to use the minimization of the entropy of the error for classification.
The minimization of the entropy of the error implies a constant value for the errors. This, in general, does not imply that the value of the errors is zero. In regression, this problem is solved by making a shift of the final result such that it's average equals the average value of the desired target. We prove that, under mild conditions, this algorithm, when used in a classification problem, makes the error converge to zero and can thus be used in classification.

## 1 Introduction

Since the introduction by Shannon [8] of the concept of entropy, and the posterior generalization made by Renyi [7], that entropy and information theory concepts have been applied in learning systems.
Shannon's entropy,

$$H_S(x) = -\sum_{i=1}^{N} p_i log p_i \qquad (1)$$

measures de average amount of information conveyed by the random variable $x$ whose $N$ possible values occur with probability $p_i$. An extension of the entropy concept to continuous random variables $x \in C$ is:

$$H(x) = -\int_C f(x) log f(x) dx \qquad (2)$$

where f(x) is the probability density function (pdf) of the random variable $x$.
The use of entropy and relative concepts have several applications in learning systems. These applications are mostly based on finding the mutual information and the consequent relations between the distributions of the variables involved in a particular problem. Linsker [5] proposed the *Infomax* principle that consists on the maximization of Mutual Information (MI) between the input and the output of the neural network. Mutual information gives rise to either unsupervised or supervised learning rules depending

on how the problem is formulated. We can have unsupervised learning when we manipulate the mutual information between the outputs of the learning system or between its input and output. Examples of these approaches are independent component analysis and blind source separation [1, 2]. If the goal is to maximize the mutual information between the output of a mapper and an external desired response, then learning becomes supervised.
With the goal of making supervised information-theoretic learning, several approaches have been proposed:

– CIP (Cross Information Potential) - The CIP tries to establish the relation between the pdfs of two variables. These variables could be the output of the network and the desired targets or the output of each layer and the desired targets [10].
– The entropy maximization of the output of the network and simultaneously the minimization of the entropy of the output of the data that belongs to a specific class. This method was proposed by Haselsteiner [4] as a way of performing supervised learning without numerical targets.
– MEE - Consists of the minimization of the error entropy between the outputs of the network and the desired targets. This approach was proposed by Denis Erdogmus [3] and used to make time series prediction.

We made some experiments with these proposed three methods with the goal of performing supervised classification but we did not achieve good results. This lead us to develop a new approach as described next.

## 2 Renyi's Quadratic Entropy and Back-propagation Algorithm

Renyi extended the concept of entropy and defined the Renyi's $\alpha$ entropy, in discrete cases, as:

$$H_{R\alpha}(x) = \frac{1}{1-\alpha} \log \left( \sum_{i=1}^{N} p_i^{\alpha} \right) \tag{3}$$

which tends to Shannon entropy when $\alpha \to 1$. If we take the Renyi's Quadratic Entropy $(\alpha = 2)$, to continuous random variables, we obtain

$$H_{R2}(x) = -\log \left( \int_C [f(x)]^2 dx \right) \tag{4}$$

Renyi's Quadratic Entropy in conjunction with the Parzen Window probability density function estimation with gaussian kernel allows, as we will see later, the determination of the entropy in a non-parametric, very practical and computationally efficient way. The only estimation involved is the pdf estimation.
Let $a = a_i \in \mathbb{R}^m$, $i = 1, \ldots, N$, be a set of samples from the output $Y \in \mathbb{R}^m$ of a mapping $\mathbb{R}^n \mapsto \mathbb{R}^m : Y = g(w, x)$, where $w$ is a set of Neural Network weights. The Parzen window method estimates the pdf $f(y)$ as

$$f(y) = \frac{1}{Nh^m} \sum_{i=1}^{N} K(\frac{y - a_i}{h}) \tag{5}$$

where $N$ is the number of data points, $K$ is a kernel function, and $h$ the bandwidth or smoothing parameter. If we use a simple Gaussian kernel (being **I** the identity matrix)

$$G(y,I) = \frac{1}{(2\pi)^{\frac{m}{2}}} exp\left(-\frac{1}{2}y^T y\right) \tag{6}$$

then, the estimated pdf $f(y)$ using Parzen window and Gaussian kernel will be:

$$f(y) = \frac{1}{Nh^m}\sum_{i=1}^{N} G\left(\frac{y-a_i}{h}, I\right) \tag{7}$$

The Renyi's Quadratic Entropy can be estimated, applying the integration of gaussian kernels [10], by

$$\begin{aligned}\hat{H}_{R2}(y) &= -\log\left[\int_{-\infty}^{+\infty}\left(\frac{1}{Nh^m}\sum_{i=1}^{N} G(\frac{y-a_i}{h}, I)\right)^2 dx\right] \\ &= -\log\left[\frac{1}{N^2h^{2m-1}}\sum_{i=1}^{N}\sum_{j=1}^{N} G(\frac{a_i-a_j}{h}, 2I)\right] = -\log V(a)\end{aligned} \tag{8}$$

Principe [6] calls $V(a)$ the *information potential* in analogy with the potential field in physics. For the same reason he also calls the derivative of $V(a)$ the *information force* $F$. Therefore

$$\begin{aligned}F &= \frac{\partial}{\partial a}V(a) = \frac{\partial}{\partial a}\left[\frac{1}{N^2h^{2m-1}}\sum_{i=1}^{N}\sum_{j=1}^{N} G(\frac{a_i-a_j}{h}, 2I)\right] \\ F_i &= -\frac{1}{2N^2h^{2m+1}}\sum_{j=1}^{N} G(\frac{a_i-a_j}{h}, 2I)(a_i-a_j)\end{aligned} \tag{9}$$

This *information force* is back-propagated into the MLP the same way as in the MSE algorithm. The update of the neural network weights is performed using $\Delta w = \pm\eta\frac{\partial V}{\partial w}$. The $\pm$ means that we can maximize $(+)$ or minimize $(-)$ the entropy.

## 3 Supervised Classification with Error Entropy Minimization

We make use of the information-theoretic concepts, applying an entropy approach to the classification task using the entropy minimization of the error between the output of the network and the desired targets: the Error Entropy Minimization, EEM.
Let $d \in \mathbb{R}^m$ be the desired targets and $Y$ the network output from the classification problem and $e_i = d_i - Y_i$ the error for each data sample $i$ of a given data set. The error entropy minimization approach, introduced by Erdogmus [3] in time series prediction,

states that Renyi's Quadratic Entropy of the error, with pdf approximated by Parzen window with Gaussian kernel, has minima along the line where the error is constant over the whole data set. Also the global minimum of this entropy is achieved when the pdf of the error is a Dirac delta function.
Taking the quadratic entropy of the error

$$\hat{H}_{R2}(e) = -\log\left[\frac{1}{N^2 h^{2m-1}} \sum_{i=1}^{N}\sum_{j=1}^{N} G\left(\frac{e_i - e_j}{h}, 2I\right)\right] \tag{10}$$

we clearly see that this entropy will be minimum when the diferences of all the error pairs $(e_i - e_j)$ are zero. This means that the errors are all the same. In classification problems with separable classes, the goal is to get all the errors equal to zero, meaning that we don't get any errors in the classification. In classification problems with non separable classes the goal is to achieve the Bayes error.
In the following we prove that, in classification problems, imposing some conditions to the output range and target values, the EEM algorithm makes the error converge to zero. The objective is to minimize the entropy of the error $e = d - Y$ and, as stated above, to achieve the goal of $e = 0$ for all data samples.

**Theorem 1.** *Consider a two class supervised classification problem with a unidimensional output vector. $Y \in [r,s]$ is the output of the network and $d \in \{a,b\}$ the target vector or the desired output. If $r = a$, $s = b$ and $a = -b$ then the application of the EEM algorithm makes the errors on each data point be equal and equal to zero.*

*Proof.* Define the targets as $d \in \{-a,a\}$ and consider the output of the network as $Y \in [-a,a]$.The errors are given by $e = d - Y$.
If the true target for a given input $x_i$ is $\{a\}$ then the error $e_i$ varies in $P = [0,2a]$.
If the true target for a given input $x_j$ is $\{-a\}$ then the error $e_j$ varies in $Q = [-2a,0]$.
Since the minimization of the entropy of the error makes the errors all have the same value, $r$, we get $e_i = e_j = r$.
$r$ must be in $P$ and $Q$. $P \cap Q = \{0\}$ thus $r = 0$ and $e_i = e_j = 0$.

A similar proof can be made for multidimensional output vectors.

Therefore, by minimizing the Renyi's Quadratic Entropy of the error, applying the back-propagation algorithm, we find the weights of the neural network that yield good results in classification problems as we illustrate in the next section.

## 4 Experiments

We made two experiments, using multilayer perceptrons (MLP), to show the application of the EEM algorithm to data classification. The learning rate $\eta$ and the smoothing parameter $h$ were experimentally selected. However this is one subject that must be studied with more detail in our subsequent work.
In the first experiment we created a data set consisting of 200 data points, constituting 4 separable classes (figure 1).

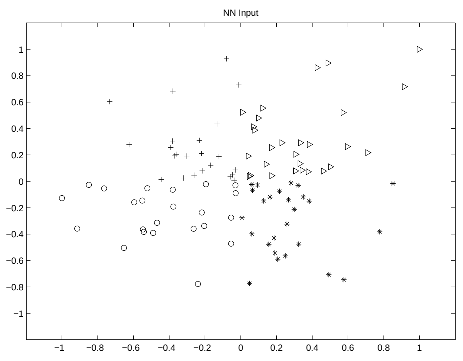

**Fig. 1.** Dataset for the first problem

Several (2;n;2) MLP's were trained and tested 40 times, 150 epochs, using EEM and also for MSE. We made *n* vary from 3 to 6. Each time, half of the data set was randomly used for training and the other half for testing. The results of the first experiment are shown in table 1.

**Table 1.** The error results of the first experiment (mean error $\pm$ std)

| | n=3 | 4 | 5 | 6 | STD |
|---|---|---|---|---|---|
| EEM | $2.43 \pm 1.33$ | $2.20 \pm 1.20$ | $2.01 \pm 1.09$ | $2.09 \pm 1.02$ | 0.18 |
| MSE | $2.93 \pm 1.46$ | $2.55 \pm 1.24$ | $2.64 \pm 1.13$ | $2.91 \pm 1.73$ | 0.19 |

In the second experiment we used the well known Fisher's IRIS data set. It consists of 3 classes, 4 numeric attributes, 150 instances. One class is linearly separable from the other two, but the other two are not linearly separable from each other.

Several (2;n;2) MLP's were trained and tested 40 times, 150 epochs, for EEM and also for MSE. We made *n* varying from 3 to 8. Each time, half of the data set was randomly used for training and the other half for testing. The results of the second experiment are shown in table 2.

**Table 2.** The error results for IRIS data set (mean error $\pm$ std)

| | n=3 | 4 | 5 | 6 | 7 | 8 | STD |
|---|---|---|---|---|---|---|---|
| EEM | $4.36 \pm 1.12$ | $4.43 \pm 1.30$ | $4.38 \pm 1.34$ | $4.30 \pm 1.16$ | $4.41 \pm 1.42$ | $4.31 \pm 1.27$ | 0.05 |
| MSE | $4.72 \pm 4.75$ | $4.75 \pm 1.27$ | $4.15 \pm 1.32$ | $3.97 \pm 1.05$ | $5.18 \pm 4.74$ | $4.65 \pm 1.32$ | 0.44 |

The results show, in almost every experiments, a small, but better, performance of the EEM algorithm. They also show, especially in the second experiment, that the variation of the error along *n* is smaller in the EEM than in the MSE (last column STD - standard deviation over the different *n* sessions). This could mean that the relation between the complexity of the MLP and the results of the EEM algorithm is not so tight as for the MSE algorithm, although this relation must be studied with more detail in our future work.

## 5 Conclusions

We have presented, in this paper, a new way of performing classification by using the entropy of the error between the output of the MLP and the desired targets, as the function to minimize. The results show that this is a valid approach for classification and, despite the small diference comparing to MSE, we expect to achieve better results in high dimensional data. The complexity of the algorithm, $(N^2)$, imposes some limitations on the number of samples in order to get results in a reasonable time. Some aspects in the implementation of the algorithm will be studied in detail in our future work: how to choose $h$ and $\eta$ and make their values adjust during the training phase to improve the classification performance. We have already tested the adjustment of $h$ during the training phase, but we did not achieved good results. We know that the variation of $\eta$ during the training process improves the performance [9]. So, we plan to adjust $\eta$ as a function of the error entropy instead of adjusting it as a function of the MSE.

## References

1. Amari, S., Cichocki A. and Yang, H.: A New Learning Algorithm for Blind Signal Separation. Advances in Neural Information Processing Systems **8**, MIT Press, Cambridge MA, (1996) 757–763
2. Bell A., Sejnowski T.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, **7-6** (1995) 1129–1159
3. Erdogmus D., Principe J.: An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems. Trans. on Signal Processing, **50-7** (2002) 1780–1786
4. Haselsteiner H., Principe J.: Supervised learning without numerical targets - An information theoretic approach. European Signal Processing Conf., (2000) n/a-n/a
5. Linsker R.:Self-Organization in a Perceptual Network. IEEE Computer, **21** (1998) 105–117
6. Principe J., Fisher J., Xu D.: Information-Theoretic Learning. Computational NeuroEngineering Laboratory, University of Florida, (1998)
7. Renyi A.: Some Fundamental Questions of Information Theory. Selected Papers of Alfred Renyi, **2** (1976) 526–552
8. Shannon C.: A Mathematical Theory of Communication. Bell System Technical Journal, **27** (1948) 379-423, 623–653
9. Silva, F. and Almeida, L.: Speeding up Backpropagation, Advanced Neural Computers, Eckmiller R. (Editor), (1990) 151–158
10. Xu D., Principe J.: Training MLPs layer-by-layer with the information potential. Intl. Joint Conf. on Neural Networks, (1999) 1716–1720