A Pseudoregression Formulation of Emphasized Soft Target Procedures for Classification Problems

Soufiane El Jelali[1], Abdelouahid Lyhyaoui[2] and Aníbal R. Figueiras Vidal[1] *

[1]Univ. Carlos III de Madrid - Dept. of Signal Processing and Communications
Av. de la Universidad 30, Leganés, Madrid - Spain

[2]Univ. Abdelmalek Essaâdi- ENSA of Tangier - Dept. of Telecoms
and Electronics - LTI Lab.
B.P. 1818, Tanger Principale, Tangier - Morocco

Abstract. Replacing a hard decision by a soft targets version including an attentional mechanism provides performance advantage and flexibility to solve classification tasks. In this paper, we modify the standard emphasized soft target method by proposing two new ideas, to avoid unnecessary updating and inappropriate definition of soft targets, in order to increase designs performance. Experimental results using MLPs show the effectiveness of this approach compared with the standard ST and other methods.

1 Introduction

The training process of Neural Network (NN) classifiers minimizes a selected cost function (typically, the mean square error) that measures the difference between hard target $t(\mathbf{x}^k)$ and NN output $o(\mathbf{x}^k)$ for a given set of samples $\{\mathbf{x}^k, t(\mathbf{x}^k)\}_{k=1}^{K}$ by means a local search algorithm [1][2]. But the above procedure minimizes an approximation to the misclassification rates. On the other hand, there are algorithms that try to approximate directly the empirical misclassification rates as the original Perceptron Rule [3], but they offer a poor generalization and have a convergence problems; other approaches look for error measures which are good approximations to the misclassification error, such as Fisher discriminants [4], "decision based" algorithms [5], as well as "energy functions" [6]; and others are based on the Maximum Margin (MM) theory [7][8][9]. All them are also approximations to an exact error rate minimization, offering better or worse results depending on the problem under consideration.

A possibility for improving classifiers performance is to modify conventional training algorithms paying more effort to reduce the cost function for samples that result relevant for classification border definition; this is the well developed family of methods called Sample Selection (SS), or better, Sample Editing (SE) procedures, that are based in emphasizing erroneous samples [10][11], those samples that are nearer the classification border [12][13][14][15], or both kinds of samples [16]. In fact, samples showing a high error and those near to the

*This work has been partially supported by MEC Pjt. TEC2008-02473/TEC.

border are important for NN training to obtain a good design. It was remarked in [17][18][19] that both kinds of samples are the base to construct boosting schemes.

The idea of using Soft Targets (STs) in this context has its origin in [20], where training is not carried out at those steps for which the output is clearly enough for a correct classification. This is reasonable because we want just a correct sign of the output, and our experiments showed good performance when applying this idea. In other works [21][22][23][24] we checked that the alternative of defining a ST by means of a convex combination of the real target and the output of a previously trained auxiliary classifier, the combination parameter being a function of the auxiliary classifier error with a form which forces an emphasis for critical samples (those being important to reduce classification error) also works very well; the implicit emphasis appearing as the cause of it.

In this paper, we include basic ideas of [20] in our second ST approach, by means of modifying the emphasis function. The first change consists on canceling the second term of the convex combination for those samples that are wrongly classified and near to the classification border. The second is to avoid training when the outputs of the classifier correspond to right classifications and are further from the border than the STs.

The rest of the paper is organized as follows. In Section 2, we offer a brief summary about ST definition and we present the two new ideas to improve ST methods. Section 3 is dedicated to evaluate experimental results of the resulting approach for several classification problems. We close this paper with some conclusions and suggestions for further research.

2 Extended Soft Target approach

We define an ST $t_s(\mathbf{x})$ as a convex combination of original target $t(\mathbf{x})$ and output $\mathrm{o}_{aux}(\mathbf{x})$ of an auxiliary classifier:

$$t_s(\mathbf{x}) \;=\; \lambda(\mathbf{x})\; t(\mathbf{x}) \;+\; (\,1 - \lambda(\mathbf{x})\,)\; \mathrm{o}_{aux}(\mathbf{x}) \tag{1}$$

where $\lambda(\mathbf{x})$ $(0 \le \lambda(\mathbf{x}) \le 1)$ is a convex weight which depends on the error of the auxiliary machine. A possibility which gives good results [23][21][22][24] is

$$\lambda(|e(\mathbf{x})|) = \begin{cases} \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_1}) & \text{for} \quad |e(\mathbf{x})| \le \mu, \\ \\ \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_2}) & \text{for} \quad \mu < |e(\mathbf{x})| \le 2. \end{cases} \tag{2}$$

where μ, α_1, and α_2 are Gaussian bell parameters. $\lambda(|e(\mathbf{x})|)$ plays the role of an emphasis function which gives more importance to the more relevant samples. In addition, this emphasis function allows to reduce automatically the attention paid to the well classified samples and the outliers. Note that μ establish the value of $|e(\mathbf{x})|$ at which $t_s(\mathbf{x})$ is maximum (unity), and that α_1, α_2, control the decay from this value when samples are "clearly" well classified ($|e(\mathbf{x})| \to 0$) or give highly erroneous results ($|e(\mathbf{x})| \to 2$) by the auxiliary machine.

To improve ST MLP classifiers, we suggest the following change:

$$t'_s(\mathbf{x}) = \begin{cases} \lambda(|e(\mathbf{x})|)\, t(\mathbf{x}) \;\; \text{if} \;\; sgn(t_s(\mathbf{x})) \neq sgn(t(\mathbf{x})) \;\; \text{and} \;\; |t_s(\mathbf{x})| << 1 \\ \\ t_s(\mathbf{x}) \;\; \text{otherwise.} \end{cases} \tag{3}$$

The proposed correction affects to samples that are erroneously classified by the auxiliary machine and have small values of $t_s(\mathbf{x})$ (here, $|t_s(\mathbf{x})| \leq 0.1$). The effect of this correction is to keep a correct target from the sign point of view for those samples that can be well classified without great difficulties.

The second idea to improve the overall performance is to avoid training the final machine when the sample is correctly classified and the absolute output value is higher than the absolute value of the target. In this way, we do not force the machine to unnecessarily reduce its output when the sample is located even further from the border than the distance required by the soft target. This means that we do not spend expressive power of the machine architecture to obtain particular values that are not required, since we are applying a regression formulation just only to reduce the difficulties that hard targets create. We call this modification "pseudoregression" to express that it differs from standard regression in the above sense.

3 Experiments

3.1 Datasets

We have applied the standard and the new ST MLP schemes to six classification problems: Crabs, credit, hepatitis, ionosfera, image and ripley. Ripley [25] is a synthetic problem that has a Bayesian misclassification rate of 8%. The rest of problems are real datasets: Crabs is obtained from PRNN site [26], and the others are taken from the UCI Machine Learning Repository [27]. We will refer to them as cra, cre, hep, ion, ima and rip, respectively. The main characteristics of these problems are presented in Table 1.

Dataset	Train (+1/−1)	Test (+1/−1)	$\#dim$
cra	120 (59/61)	80 (41/39)	7
cre	414 (167/247)	276 (140/136)	15
hep	93 (70/23)	62 (53/9)	19
ima	1848 (821/1027)	462 (169/293)	18
ion	201 (101/100)	150 (124/26)	34
rip	250 (125/125)	1000 (500/500)	2

Table 1: Characteristics of the classification benchmark datasets.

3.2 Training

To design our ST classifier, we train previously a standard MLP as the auxiliary machine with original training dataset $\{(\mathbf{x}^k, t(\mathbf{x}^k);\ t(\mathbf{x}^k) \in \{\pm 1\}_{k=1}^{K}$ to generate an ST $t_s(\mathbf{x})$ using its output $o_{aux}(\mathbf{x})$ according to (1) and (2). After that, we train a final MLP with new training data $(\mathbf{x}, t_s(\mathbf{x}))$ and the standard and new ST algorithms. We will refer to these ST-classifiers as $\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$ and $\mathrm{ST_n}$-$\mathrm{MLP_{MLP}}$, respectively. All the MLP machines (both auxiliary and final) are trained with a BP (Back-Propagation) algorithm using the square error cost function and allowing a number of training epochs high enough (800 for the auxiliary MLP; 600 for ST-classifiers) to assure convergence, over a random portion of 90% of the training data set, and we use Early Stopping to select the MLP weights of the epoch that has achieved the minimum mean square error over the remaining 10% of data. Ten runs have been completed for each set of free parameter values applying a 10-fold cross-validation (CV) in order to select the non-trainable parameters. The auxiliary MLP weights are randomly initialized for each run following a uniform $[-0.1, 0.1]$ distribution, and the weights of the ST-classifiers are initially set to the final values of the auxiliary MLP having the same size. The hyperbolic tangent function is used as the output activation.

We explore the following values of the non-trainable parameters:

- Number of hidden neurons N_{T}, N_1 and N_2 of a single MLP, $\mathrm{MLP_T}$, the auxiliary MLP, and a the final machines ($\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$, $\mathrm{ST_n}$-$\mathrm{MLP_{MLP}}$) in [4 6 8 10 12 14 16].
- μ in $[10^{-2}\ 0.5\ 1\ 1.5\ 2]$, α_1 and α_2 in $[10^{-3}\ 10^{-2}\ 10^{-1}\ 1\ 3\ 5]$.

3.3 Results

Table 2 shows the results of the experiments for $\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$ and $\mathrm{ST_n}$-$\mathrm{MLP_{MLP}}$ compared with the optimal $\mathrm{MLP_T}$ and EDR (Error Dependent Repetition [10]), as well as the values of the design parameters obtained applying CV.

EDR is one of the best methods among those proposed by Cachin [10], according to his results. It consists on presenting once all the samples to the machine, and selecting maximum square error $e_{\max}$. During a given number of epochs I the machine is trained with only the samples that show a square error higher or equal than $i\,e_{\max}/I$, i being the corresponding epoch. After it, the process is repeated. We apply this algorithm to an MLP architecture with N' hidden neurons, N' being selected by CV among the same values than N_{T}, N_1 and N_2. Also, we explore I by CV in the following interval [40 50 80 100 160 200].

As previously known from experimental results, $\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$ outperforms both $\mathrm{MLP_T}$ and EDR: $\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$ provides clearly better results in cre and ima, and similar performance in the other problems. $\mathrm{ST_n}$-$\mathrm{MLP_{MLP}}$ is even better than $\mathrm{ST_s}$-$\mathrm{MLP_{MLP}}$ in cre and ima, and also in cra; slightly better in ion, and a little bit worse in rip. Consequently, we admit that the modifications we have introduced in ST procedures are useful when trying to obtain higher performance designs.

	ST$_\mathrm{s}$-MLP$_\mathrm{MLP}$	ST$_\mathrm{n}$-MLP$_\mathrm{MLP}$	MLP$_\mathrm{T}$	EDR
	$N_1/N_2/\mu/\alpha_1/\alpha_2$	$N_1/N_2/\mu/\alpha_1/\alpha_2$	N_T	N'/I
cra	97.08±0.72	**97.50±0.10**	97.40±0.40	96.46±1.46
	$6/16/0.01/3/10^{-3}$	$8/6/1/1/0.1$	10	6/40
cre	89.60±2.51	**90.10±1.77**	88.30±1.70	86.77±2.09
	$4/6/1/10^{-3}/3$	$6/6/1/10^{-3}/5$	14	14/40
hep	89.34±2.97	89.13±2.50	89.00 ±2.80	**89.95±2.15**
	$10/12/1.5/3/10^{-3}$	$12/12/2/1/3$	10	10/50
ima	90.60±1.60	**92.16±0.73**	88.50±2.40	88.78±1.63
	$10/16/1.5/1/0.1$	$10/16/0.5/10^{-3}/3$	6	12/40
ion	93.01±1.42	**93.30±1.44**	**93.30±1.60**	92.15±0.74
	$8/6/1.5/1/3$	$6/6/1.5/10^{-3}/10^{-2}$	6	4/100
rip	**90.41±0.64**	89.93±1.07	90.10±0.80	89.42±0.67
	$12/14/0.5/3/10^{-2}$	$12/14/1.5/0.1/1$	14	12/100

Table 2: Averaged percentages of correct classification (± standard deviation) of ST$_\mathrm{s}$-MLP$_\mathrm{MLP}$, ST$_\mathrm{n}$-MLP$_\mathrm{MLP}$, MLP$_\mathrm{T}$, and EDR for the different test datasets, indicating the design parameters obtained by CV.

4 Conclusion and further work

It has been proved that emphasized soft target techniques are useful to open the possibility of obtaining better performance designs for classification purposes. In this paper, we have introduced a modification of the soft target definition and a "pseudoregression" training method which demonstrate empirically relevant potential advantages with respect to standard soft target approaches.

Other modifications of soft target formulations and considering how to apply in a sequential manner these techniques, and even to use them to construct high performance ensemble classifiers, are promising avenues to extend this work.

References

[1] S. Haykin, *Neural Networks : A Comprehensive Foundation* (2nd ed.); Upper Saddle River, NJ: Prentice-Hall, 1999.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford, UK: Oxford Univ. Press, 1995.

[3] F. Rosenblatt, "The Perceptron: A Probabilistic model for information storage and organization in the brain"; Psychological Review, vol. 65(6), pp. 386-408, 1958.

[4] R. A. Fisher, "The use of multiple measurements in Taxonomic problems", Annels of Eugenics, vol. 7, Pt II, pp. 179-188, 1936.

[5] S. Y. Kung, J. S. Taur, "Decision-based neural networks with signal/image classification applications", *IEEE Trans. Neural Networks*, vol. 6, pp. 170-181, 1995.

[6] B. A. Telfer, H. H. Szu, "Energy functions for minimizing misclassification error with minimum complexity networks", Neural Networks, vol. 7, pp. 809-818, 1994.

[7] B. E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers", *Proc. 5th Workshop Comp. Learning Theory* (D. Hassler, ed.), pp. 144-152; San Mateo, CA,: ACM Press, 1992.

[8] C. Cortes, V. Vapnik, "Support Vector networks", Machine Learning, vol. 20, pp. 273-297, 1995.

[9] K. R. Müller, S. Mika, G. Rätsch, , K. Tsuda, B. Schölkopf,"An introduction to kernel-based learning algorithm", *IEEE Trans. Neural Networks*, vol. 12, pp. 181-201, 2001.

[10] C. Cachin, "Pedagogical pattern selection strategies", Neural Networks, vol. 7, pp. 171-181, 1994.

[11] P. W. Munro, "Repeat until bored: A pattern selection strategy", Advances in Neural Inf. Proc. Sys. 4 (J.E. Moody et al, editors), pp. 1001-1008; San Mateo, CA : Morgan Kaufmann, 1992.

[12] J. Sklansky, L. Michelotti, "Locally trained piecewise linear classifiers", IEEE Trans. Pattern Anal. Machine Intelligence, vol. 2, pp. 101-111, 1980.

[13] A. Lyhyaoui, M. Martinez-Ramón, I. Mora-Jiménez, M. Vázquez Castro, J. L. Sancho Goméz, A. R. Figueiras-Vidal, "Sample selection via clustering to construct Support Vector-like classifiers", IEEE Trans. on Neural Networks, vol. 10, pp. 1474-1481, 1999.

[14] E. I. Chang, R. P. Lippman, "A boundary hunting Radial Basis function classifier which allocates centers constructively", Advances in in Neural Info. Proc. Sys. 5 (S.J.Hanson et al, eds.); San Mateo, CA: Maurgan Kaufmann, 1993.

[15] M. Wann, T. Hediger, N. N. Greenbaun, "The influence of training sets on generalization in feed-forwars neural networks", Proc. Intl. Joint Conf. Neural Networks, vol. 3, pp. 137-142; San Diego, CA, 1990.

[16] P. E. Hart, "The condensed nearst neighbor rule", IEEE Trans. Info. Theory, vol. 14, pp. 515-516, 1968.

[17] J. Arenas-García, A. R. Figueiras-Vidal, A. J. C. Sharkey, "The beneficial effects of using multi-net systems that focus on hard patterns", in T. Windeatt and F. Rolli, eds., Multiple Classifier systems 4th Intl. Workshop, LNSC 2709, pp. 45-55; Surrey, UK, Springer-Verlag, 2003.

[18] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, A. R. Figueiras-Vidal, "Boosting by weighting critical and erroneous samples", Neurocomputing, vol. 69, pp. 679-685, 2006.

[19] V. Gómez-Verdejo, J. Arenas-García, A. R. Figueiras-Vidal, "A dynamically adjusted mixed emphasis method for building boosting ensembles", IEEE Trans. on Neural Networks, vol. 19, pp. 3-17, 2008.

[20] I. Mora-Jiménez, A. R. Figueiras-Vidal, "Improving performance of neural classifiers via selective reduction of target levels", Neurocomputing, vol. 72, pp. 3020-3027, 2009.

[21] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "An emphasized target smoothing procedure to improve MLP classifiers performance", Proc. 16th European Symp. Artificial Neural Networks, pp. 499-504; Bruges, Belgium, 2008.

[22] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Applying emphasized soft target for Gaussian mixture model based classification", Proc. of the Intl. Multiconf. on Computer Sci. Information Technology, vol. 3, pp. 131-136; Wisla, Poland, 2008.

[23] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Designing model based classifiers by emphasizing soft targets", Fundamenta Informaticae, vol. 96(4), pp. 419-433, 2009.

[24] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Emphasized soft target design of kernel based classifiers", submitted to IEEE trans. on Neural Networks.

[25] B. D. Ripley, "Neural networks and related methods for classification (with discussion)", J. Royal Statistical Soc. Series B, vol. 56, pp. 409-456, 1994.

[26] B. D. Ripley, Pattern Recognition and Neural Networks: www.stats.ox.ac.uk/pub/PRNN

[27] C. L. Blake, C. J. Merty, UCI Repository of Machine Learning Databases: www.ics.uci.edu/∼mlearn