ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

An Emphasized Target Smoothing Procedure to Improve MLP Classifiers Performance

Soufiane El Jelali, Abdelouahid Lyhyaoui and Aníbal R. Figueiras-Vidal *

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

Abstract. Standard learning procedures are better fitted to estimation than to classification problems, and focusing the training on appropriate samples provides performance advantages in classification tasks. In this paper, we combine these ideas creating smooth targets for classification by means of a convex combination of the original target and the output of an auxiliary classifier, the combination parameter being a function of the auxiliary classifier error. Experimental results with Multilayer Perceptron architectures support the usefulness of this approach.

1 Introduction

Standard Neural Network (NN) training procedures essentially consist on selecting an error objective which measures the difference between the target and the NN output and minimizing it for the available set of labelled samples by means of a local search algorithm. They are not completely fitted for designing NN classifiers because the objectives that are suitable for applying local search algorithms are not more than approximations to the adequate measure of classifiers quality, the misclassification rate. On the other hand, algorithms that try to reduce directly the empirical misclassification rate, such as the original Perceptron Rule [1], offer poor generalization and are difficult to apply in general; Fisher type formulations [2] are also approximations, as it is the valuable concept of Maximum Margin on which Support Vector Machines are based [3].

A well developed family of methods to improve the performance of NN classifiers trained with standard procedures are the sample selection and sample edition techniques, in which more effort is paid to reduce the error objective for those samples that result more important for an appropriate definition of the classification borders. It is impossible here to provide a complete overview of these techniques, but we can remark that, from their very beginning [4][5][6], these "more important" samples are those near the boundary and/or showing a high error, the relative importance of both types depending on the characteristics of the problem under analysis with respect to the absence or presence of noise[7]; [8] is also an interesting discussion. In any case, there is the possibility of exploring how much importance must be attributed to each type by using a convex combination of measures of the error and the proximity to the border for the samples, as done in [9] to construct boosting ensembles.

*This work has been partially supported by MEC Pjt. TEC2005-00992. The work of S. El Jelali was also supported by an AECI grant.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

In this paper, we explore an idea which is suggested by transductive inference principles and combines focusing on the relevant samples and transforming the classification into a regression problem. The idea consists on applying an auxiliary classifier to evaluate the error for each sample, to transform the (discrete) original target into a smoothed version by means of a convex combination of that target with the (continuous) output of the auxiliary classifier, using a combination parameter depending on the previously evaluated errors, and to train a second classifier employing the smoothed targets. In this way, we can simultaneously edit the samples and use a more adequate error objective for standard search algorithms; this last point opening the possibility of training with these algorithms even classifier schemes that cannot be directly trained in this manner, such as Gaussian Processes machines. The idea we propose is similar to that presented in [10], although particular learning mechanisms are different.

2 Definition of the smoothed targets

We will restrict our discussion here to binary problems, and we will work with Multilayer Perceptrons (MLPs), because the only objective of this paper is to show that our proposal is simple, general, and beneficial.

So, we will first train an auxiliary standard MLP, MLP_T , and we will use its output o_{aux} to define the smoothed target for the second (final) MLP, MLP_{ST} :

$$t_s(\mathbf{x}) = \lambda(|e|)\, t(\mathbf{x}) + (1 - \lambda(|e|))\, o_{aux}(\mathbf{x}) \tag{1}$$

where $t(\mathbf{x})$ is the original target(± 1), and $\lambda(|e|)$ is the convex combination weight:

$$\lambda(|e|) = \begin{cases} \exp(-\frac{(|e|-\mu)^2}{\alpha_1}), & |e| \leq \mu \\ \\ \exp(-\frac{(|e|-\mu)^2}{\alpha_2}), & \mu < |e| \leq 2 \end{cases} \tag{2}$$

e being the error corresponding to the auxiliary NN, and μ, α_1, α_2 being parameters of the Gaussian bells. MLP_T and MLP_{ST} can have different sizes. Since we are dealing with an estimation approach, we test both linear and sigmoidal output activations for MLP_{ST}.

The aspect of $\lambda(|e|)$ is shown in Fig.1. It is maximum for $|e| = \mu$, and, from this point, it decays towards well classified ($|e| \rightarrow 0$) or wrongly classified ($|e| \rightarrow 1$) cases with different rates, corresponding to parameters α_1 and α_2, respectively. Note that $\lambda(|e|)$ indicates how much of the original target we keep in smooth target $t_s(\mathbf{x})$; i.e., how much we focus on the corresponding sample. Since we can select μ, we can emphasize more those samples that show a certain "compromise" between error and proximity to the (auxiliary) border, and reduce this "emphasis" in a different manner when the error increases or decreases, (We remark that, to emphasize more the more erroneous samples, it is enough to increase μ, and α_2 if needed). This is a very flexible form and, although it is possible to use many other "reasonable" emphasis schemes, the experience indicates that the performance results do not depend on their particular aspects in a significant amount as long as they are flexible enough.

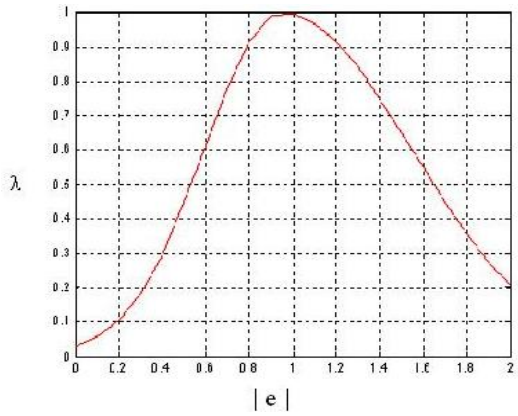

Fig. 1: Form of $\lambda(|e|)$: $\mu = 1$, $\alpha_1 < \alpha_2$.

3 Experiments

We have tested the proposed procedure with three standard benchmark datasets. The first is the synthetic bidimensional Ripley problem [11], which has a Bayesian missclassification rate of 8%. Ionosfera and Tictactoe are two real datasets from the UCI Machine Learning Repository [12].

We train both the auxiliary and MLP_{ST} machines by means of the Back-propagation algorithm, using the square error criterion, with a learning rate of 10^{-3} and applying an epoch-by-epoch 20% cross-stoping procedure, allowing a maximum number of training epochs high enough (800) to assure convergence. Ten runs have been completed for each situation, selecting also the best values of the hyperparameters:
- Number of units of MLP_T : $\text{N}_T = \{4, 6, 8, 10, 12, 14, 16\}$
- Idem MLP_{ST} : same
- μ:{0.1, 0.3, 0.6, 1, 1.2, 1.6, 2}
- α_1, α_2 :{0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 4, 5}

by means of a 20% cross-validation.

The MLP_T weigths are randomly initialized for each run following a uniform [-0.1,0.1] distribution, and the weights of the MLP_{ST} machine are initially set to the final values of the MLP_T having the same size.

Optimal values for the hyperparameters are :
- Ripley : N_T=12, N_{ST}=8, μ=1.6, α_1=0.05, α_2=4 (sigmoidal activation)
- Ionosfera : N_T=14, N_{ST}=4, μ=1.6, α_1=2, α_2=1.5 (linear output)
- Tictactoe : N_T=14, N_{ST}=4, μ=1.6, α_1=4, α_2=3 (linear output)

For the Ripley problem, highly erroneous samples help a lot to define the border, and emphasizing them allows to reduce the size of the final machine. Border samples are essential in Ionosfera. In the case of the Tictactoe, it seems to be important to pay a reduced attention just to very clearly well classified samples, and not too much is needed for wrongly classified examples.

As a reference, we will compare the results of our approach with those given by the optimal size standard MLP (which is N=14 for Ripley, N=4 for Ionosfera and N=14 for Tictactoe). Table 1 presents the results of the experiments (ten run statistics). Table 1 also includes results of using Cachin's Error Dependent Repitition (EDR), the best (on the average) method proposed in [6], as a reference. Architectures are MLPs with 14, 12, and 8 hidden neurons, respectively. Note that results are very near to those of our MLP$_{ST}$, but without creating soft targets.

Dataset	MLP	MLP$_{ST}$	EDR
Ripley	90.35±0.58	90.60±0.34*	90.25±0.40
Ionosfera	91.00±3.40	92.12±2.66	91.73±4.17
Tictactoe	68.56±4.51	71.69±4.40	71.38±4.59

Table 1: Average classification accuracy (standard deviation) for classical MLP and the MLP$_{ST}$ and the three test problems. *: Sigmoidal output activation

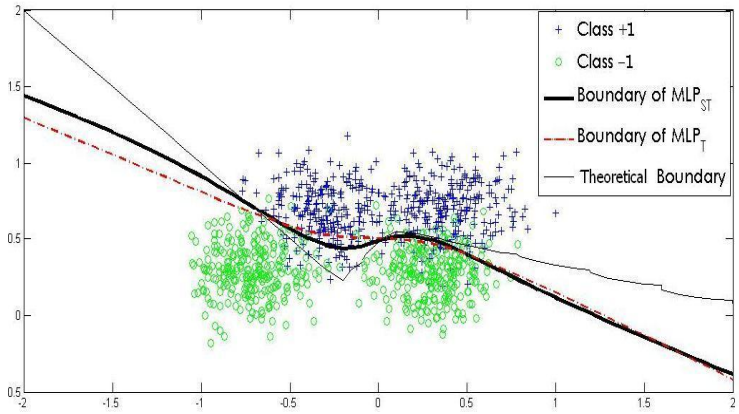

Fig. 2: Decision boundary for two dimensional problem Ripley.

In comparison with standard MLP, there is a very slight advantage of our method in Ripley, and a little bit higher for Ionosfera. Advantage is clearer for Tictactoe, although we must admit that we are far from the classification performance that can be obtained using more powerful designs [9]. In any case, improvements appear, and this is an important conclusion that support the application of the method to more elaborated schemes.

Obviously, we get these improvements by paying a higher training effort: we need 7x7x11x11 more designs (according to the number of values of the additional parameter) to carry out cross-validation, although number of training epochs is reduced (between one half and one fourth) because the initialization is done with the corresponding standard MLP weights.

In Figure 2 we can see how the MLP_{ST} classification border fits better the theoretical frontier corresponding to synthetic problem Ripley. Note that, in this case, samples are taken from Gaussian (mixture) distributions for each class; so, all of them are relevant to define the classification border; sampling effects can be reasonably compensated by means of focusing on erroneous samples. Figure 3 represents the error for the 1000 test samples when using the optimal MLP and MLP_{ST}. When applying the first machine, there are many samples with important error values, and the machine weigths have not capacity enough to reduce all these errors, originating a high number of missclassifications. But when using MLP_{ST} the implicit focusing mechanism allows the weight values to be selected just to deal mainly with relevant errors; as a consequence, it is possible to move the border towards a position at which most the samples show a very small square error, and, the number of missclassified examples decreases.

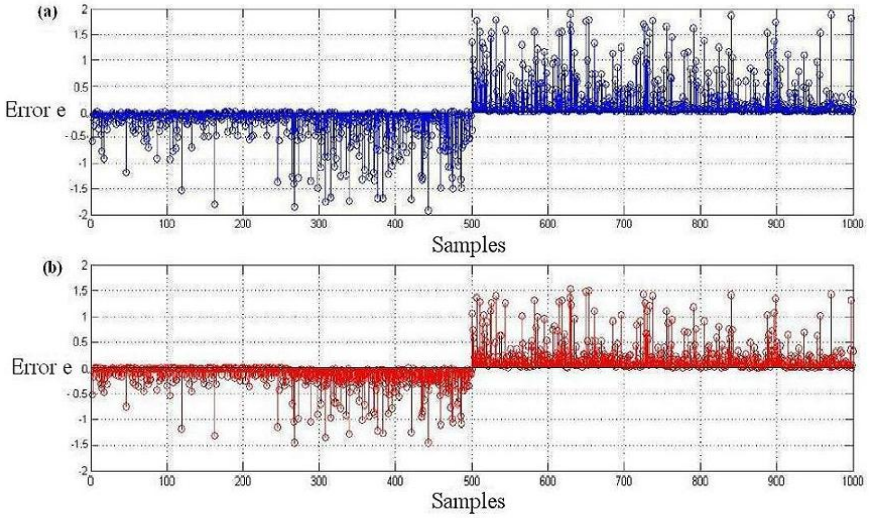

Fig. 3: Error of MLP (a) and of MLP_{ST} (b) for Ripley test samples.

Finally, sensitivity of our proposed designs is, with respect to each one of the design parameters, even more reduced than that of a traditional MLP; average error changes when each of the parameters jumps to the immediately lower and higher values are, for Ripley data: N_T: -0.17, -0.09; N_{ST}: -0.24, 0.00; μ: -0.30, -0.10; α_1: -0.13, -0.18; α_2: -0.16, -0.15; in comparison, for the MLP classifier, we have: N: +0.07, -1.02 . Of course, sensitivity increases when one considers several parameters at the same time; but the "omniscient" approach - that using the test set to select parameters (and which is not a valid design, but it can be used to estimate sensitivity with respect to design parameters) gives the results for Ripley: N_T=10, N_{ST}=16, μ=1.6, α_1=0.1, α_2=0.01; performance 90.71±0.45; and these data show that there is a very moderate sensitivity with respect to the design parameters.

4 Conclusions and further work

Smoothing classification target values in a manner which allows focusing the attention in the most important samples for learning purposes is a well-principled idea. In this paper, we propose to implement this idea by means of using the output of an auxiliary machine to construct smoothed targets by means of a convex combination of the original target with its output, the convex combination parameter being selected according to the error of the preliminary classification, just in order to allow designing an adequate focusing mechanism. A first series of experiments supports the effectiveness of the approach to improve the performance of (MLP) classifiers, and their analysis reveals that the implicit mechanisms that play a role in this improvement are consistent with the principles of the approach.

It is remarkable that, when we smooth targets, we are transforming classification into estimation problems; this fact allows to apply directly standard search algorithms to some learning classification machines that cannot use them (without employing approximations) ; this is the case of Gaussian Processes machines, in which the original classification targets cannot be considered as samples of a Gaussian variable, and of Mixture of Expert ensembles, that propose a sort of Gaussian mixture model for targets. Since there is an additional emphasis effect, not only easiness, but also some performance advantage can be expected.

Finally, let us say that it is also interesting to explore if these smoothing target principles can be used to build machine ensembles.

References

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

[2] R.A. Fisher "The use of multiple measurements in taxonomic problems"; Annals of Eugenics, **7**, Pt. II, 179-188, 1936.

[3] B.E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers"; Proc. 4th Workshop Comp. Learning Th. (D. Hassler, ed.), 144-152; San Mateo, CA: ACM Press, 1992.

[4] P.E. Hart " The condensed nearest neighbor rule", IEEE Trans. IT, **14**, 515-516, 1968.

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

[6] C. Cachin, " Pedagogical pattern selection strategies"; Neural Networks, **7**, 171-181, 1994.

[7] L. Franco, S.A. Cannas, "Generalization and selection of examples in feed-forward neural networks ", Neural Computation, **12**, 2405-2426, 2000.

[8] R. Reed, S. Oh, R.J. Marks, II, "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter", IEEE Trans. NN, **6**, 529-538, 1995.

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

[10] D. Gorse, A.J. Shepperd, J.G. Taylor "The new ERA in supervised learning", Neural Networks, **10**, 343-352, 1997.

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

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