ELSEVIER

Pattern Recognition Letters 18 (1997) 1323–1328

Pattern Recognition Letters

Classification of imbalanced remote-sensing data by neural networks

L. Bruzzone *, S.B. Serpico

Department of Biophysical and Electronic Engineering, University of Genoa, Via Opera Pia 11A, I-16145, Genova, Italy

Abstract

The multilayer perceptron neural network has proved to be a very effective tool for the classification of remote-sensing images. Unfortunately, the training of such a classifier by using data with very different a priori class probabilities (imbalanced data) is very slow. This paper describes a learning technique aimed at speeding up the training of a multilayer perceptron when applied to imbalanced data. The results obtained on an optical remote-sensing data set suggest that not only is the proposed technique effective in terms of training speed but it also allows classification results to be more stable with respect to initial weights. © 1997 Elsevier Science B.V.

Keywords: Artificial neural networks; Multilayer perceptron; Imbalanced data set classification; Remote sensing

1. Introduction

At present, growing interest is being devoted to the supervised classification of remote-sensing images by the neural-network approach (Bishof et al., 1992; Paola and Schowengerdt, 1995). Several neural models have been applied for this purpose (Serpico et al., 1996). The multilayer perceptron (MLP), trained by the error back-propagation (EBP) algorithm (Hertz et al., 1991), is one of the most widely used supervised classifiers. If properly trained, it provides approximations to the posterior class probabilities, given the feature vectors of the samples to be classified (Gish, 1990; Richard and Lippmann, 1991; Rojas, 1996). Such approximations can be used to apply the Bayes decision rule, which is optimal in terms of a minimum classification error (Fukunaga, 1990). The major problems related to this classifier are the duration and the reliability of the training process, in particular when the process is performed in 'batch mode' on imbalanced data (i.e., when the a priori probabilities of the various classes considered are very different). Such problems are encountered in many remote-sensing applications, and concern both cases in which only two classes are present in the considered data set and multiclass cases.

In the literature, several techniques aimed at speeding up the training of MLPs have been proposed (Vogl et al., 1988; Tollenaere, 1990; Hertz et al., 1991; Bishop, 1996). However, only few authors have addressed the problem of training an MLP by using imbalanced data sets (Anand et al., 1993, 1995). Anand et al. (1993) devised a technique suitable for two-class cases that is very useful to reduce the training time for MLPs. The same authors extended the use of this technique to multiclass cases

* Corresponding author. E-mail: lore@dibe.unige.it.

PII S0167-8655(97)00109-8

(Anand et al., 1995). However, this extension requires the definition of as many networks as the classes considered. In addition, the outputs of MLPs trained by this algorithm cannot be considered approximations for the posterior class probabilities optimized in accordance with some criterion; therefore, such an algorithm does not represent an implementation of the Bayes decision rule for a minimum-error classification.

This paper describes a two-phase technique for speeding up the training process of an MLP when it is applied to imbalanced data. Such a technique allows MLPs to be trained to obtain an estimate of the a posteriori class probabilities by minimization of the mean squared error (MSE criterion). Both two-class cases and multiclass cases are addressed. Experiments are carried out on optical remote-sensing data acquired on an agricultural area.

The paper is organized into 5 sections. Section 2 briefly describes the training of MLPs with the MSE criterion. Section 3 presents the proposed approach for both two-class and multiclass cases. Experimental results are presented in Section 4 while conclusions are drawn in Section 5.

2. Training MLPs with the mean squared error criterion

We first consider the two-class case. Let us assume to use a neural network with one hidden layer and sigmoidal non-linearities. Let us also assume that the number of samples from each of the classes is in proportion to the a priori probability of class membership. The class ω_2 is the dominant one, i.e., the number n_2 of samples of this class is much larger than the number n_1 of samples of the class ω_1. In order to make the network output provide an approximation to the a posteriori probability $P(\omega_1/x)$ of the class ω_1, given the feature vector x of a sample, optimized with respect to the MSE, we use the EBP training algorithm and MSE as a cost function (Gish, 1990):

$$E = E_1 + E_2 = \frac{1}{n}\sum_{i=1}^{n_1}\left[t^{(1)} - o\left(x_i^{(1)}\right)\right]^2 + \frac{1}{n}\sum_{j=1}^{n_2}\left[t^{(2)} - o\left(x_j^{(2)}\right)\right]^2, \qquad (1)$$

where E_l is the contribution of the class to the MSE; $n = n_1 + n_2$ is the total number of training samples; $t^{(l)}$ is the target for the samples of the class ω_l and $o\left(x_i^{(l)}\right)$ is the network output when the ith sample of the lth class is presented to the network input. The targets for the samples of the classes ω_1 and ω_2 are $t^{(1)} = 1$ and $t^{(2)} = 0$, respectively.

Let us recall the reasons why training with imbalanced data sets is slow. It is well known that the EBP algorithm is based on the gradient descent of the MSE. As shown in (Anand et al., 1993), at the beginning of the training phase, the gradient of the mean squared error calculated with respect to the network weights is dominated by the contribution related to the dominant class ω_2 ($\|\nabla E_1\| \ll \|\nabla E_2\|$); therefore, in this case $\nabla E \approx \nabla E_2$. Moreover, the angle between the two vectors ∇E_1 and ∇E_2 is larger than 90°. As a consequence, during the gradient descent, E_2 tends to decrease, whereas E_1 tends to increase rapidly. If the value of E_1 approaches its upper limit (in our case, $E_1 \approx n_1/n_2$), convergence becomes slow.

3. The proposed learning technique

3.1. The two-class case

The proposed technique divides the training process into two phases in such a way as to reach convergence, while avoiding a sharp increase in E_1.

Phase 1: The EBP algorithm is applied to the following modified cost function:

$$E' = E_1' + E_2' = \frac{n_2}{n_1}E_1 + E_2. \qquad (2)$$

The mean values of $\|\nabla E_1\|$ and $\|\nabla E_2\|$, calculated with respect to the possible values of the network weights, are about equal (a proof similar to that of Theorem 3 in (Anand et al., 1993) can be provided). Therefore, in the first phase, the problem of a sharp increase in is resolved. This phase is discontinued when the following condition is satisfied:

$$E_l' \leqslant \frac{n_2}{n} T \quad (l = 1,2;\ T < 1). \qquad (3)$$

This ensures that both the overall MSE (see Eq. (1)) and the following class-related MSEs will be less than or equal to T:

$$MSE_l = \frac{1}{n_l}\sum_{j=1}^{n_l}\left[t^{(l)} - o\left(x_j^{(l)}\right)\right]^2 = \frac{n}{n_l}E_l \leqslant T \quad (l = 1,2). \tag{4}$$

Phase 2: Unfortunately, the output of the network obtained in this way does not represent an approximation of the a posteriori probability $P(\omega_1/x)$, due to the cost modification introduced into Phase 1 (see Eq. (2)).

We can then use the network weights obtained at the end of the first phase as the initial weights of the second training phase. The latter phase is performed using training with the MSE criterion (Section 2), so that the network output may be used directly as an approximation to $P(\omega_1/x)$. At the beginning of the second phase, E_1 generally increases but, as both MSE_1 and MSE_2 (Eq. (4)) are small, the amplitudes of the variations of E_1 and E_2 are small, too (Anand et al., 1993). If E_1 does not approach its upper limit, the situation that leads to slow convergence (see Section 2) is avoided.

3.2. The multiclass case

In the multiclass case, we can consider MLPs with as many outputs as the number of classes. The mean square error to be minimized is

$$E = \sum_{l=1}^{M} E_l = \sum_{l=1}^{M}\frac{1}{n\cdot M}\sum_{k=1}^{M}\sum_{i=1}^{n_l}\left[t_k^{(l)} - o_k\left(x_i^{(l)}\right)\right]^2, \tag{5}$$

where M is the number of classes; $t_k^{(l)}$ is the target for the kth output of the network for the samples of the lth class (such a target is equal to 1 if $k = l$, and to 0 otherwise); $o_k\left(x_i^{(l)}\right)$ is the kth output of the network when the ith sample of the lth class is presented to the input; E_l is the contribution of class ω_1 to the mean squared error. In order to obtain approximations to the posterior class probabilities, the sum of the outputs of the network should be normalized to 1 (Gish, 1990).

Even though this situation is more complex, the results of the two-class case can be extended to the multiclass case as follows. Since at the beginning of the training phase the contribution of minority classes to the direction of the gradient E of the error is negligible, the mean squared error of some of such classes may increase and approach their upper limit. Consequently, training may become slow.

To extend the proposed technique to the multiclass case, the following formula of the modified cost can be used in the first phase:

$$E' = \sum_{l=1}^{M} E_l' = \sum_{l=1}^{M}\frac{n_{\max}}{n_l}E_l, \tag{6}$$

where M is the number of classes and $n_{\max}$ is the number of samples of the dominant class (i.e., the class with the largest number of samples). The first phase is discontinued when the following condition is satisfied:

$$E_l' \leqslant \frac{n_{\max}}{n}\cdot T \ (l = 1,\ldots,M;\ T < 1), \tag{7}$$

where n is the total number of training samples.

Thanks to the modification to the cost function, the mean values $\|\nabla E'\|$ of all the contributions E_l' to E' are about equal, so that the contribution of minority classes to $\|\nabla E'\|$ is no longer negligible. This balance among contributions (which is natural for training with data evenly distributed among all different classes) helps to avoid slow convergence.

The weights obtained at the end of the first phase are used as starting weights of the second training phase, which is performed by using the standard MSE criterion.

4. Experimental results

For all the experiments reported, we considered a data set with reference to an agricultural area near the village of Feltwell (UK). We selected a section (250×350 pixels) of a scene acquired by a multispectral optical sensor (an airborne thematic mapper (ATM) with eleven spectral bands (Richards, 1993)) mounted on board an aircraft. For our experiments, we considered only the six spectral bands that correspond to the bands provided by the thematic mapper sensor installed on Landsat satellites (except for the thermal band). Each pixel of the multispectral image was considered as an input pattern. The six bands utilized were associated with each pixel to form the

Table 1
Classes and related numbers of pixels in the considered remote-sensing training set

Class	Number of pixels	A priori probability
Wheat	90	0.06
Sugar beets	728	0.51
Potatoes	90	0.06
Carrots	319	0.23
Stubble	191	0.14

feature vector that was used as input to the neural classifier.

For the selected section, we prepared a reference map by using the available data on the ground truth. Such a map was used to extract the information necessary to train the neural classifier. For our experiments, we considered the following agricultural classes: wheat, sugar beets, potatoes, carrots, and stubble. The training-set pixels were obtained by sampling the related fields. In order to assess the advantages of the proposed technique in the presence of classes with low a priori probabilities, we reduced the number of samples of the wheat and the potatoes classes. This was accomplished by further subsampling the fields of these classes. Table 1 shows the numbers of samples obtained in the resulting training set.

For each experiment, we defined an MLP and compared the numbers of iterations required by the standard EBP algorithm with the MSE criterion and the proposed technique. The same learning rate was used for both training techniques (and for both phases of the proposed technique). The final convergence criterion was also the same for both training techniques, and was given by fixing the number of misclassified samples below which training can be discontinued. The intermediate convergence criterion for the proposed technique (to stop the first training phase) was given by fixing the value of the threshold T.

A fully-connected MLP with 6, 10, and 5 units in the input, hidden and output layers, respectively, was trained to classify our data set using the learning rate $\eta = 0.01$. As a final convergence criterion, training was discontinued when the number of misclassified samples was less than or equal to 64 (i.e., 4.5% of the training samples). To stop the first training phase of the proposed technique, the parameter T was set to 0.2.

Five experiments were carried out on the above data set, starting from the same five randomly generated sets of weights for both the standard and the proposed techniques. Results are given in Table 2. As can be noticed, the proposed method allowed notable speed-ups (an average of 41.5 times) with more homogeneous distributing of the number of iterations with respect to the initial weights. These improvements remain significant even if one disregards experiment 2, which is particularly unfavourable for the standard EBP algorithm.

Tables 3 and 4 show the accuracies provided at convergence for each class by both the networks trained by the standard EBP algorithm and the networks trained by the proposed technique. If one analyzes the standard deviation of the classification accuracies obtained for each single class in the five experiments, one can deduce that the proposed training technique exhibits a more stable behaviour than the standard EBP technique. In addition, the errors incurred are more evenly distributed among the classes. This was obtained thanks to the first phase of our technique which allowed the errors on all the classes to decrease in a more balanced manner. In particular, one of the two classes with the lowest a priori probabilities (i.e., wheat) was much better classified by the neural network trained by the proposed technique. On the contrary, only a slight improvement was obtained for the other minority class (i.e., potatoes). In order to interpret this situation, we applied another well-known classifier (i.e., the *k*-nearest neighbour (Fukunaga, 1990)) to the same data set. Analysis of the error matrix pointed out that

Table 2
Numbers of iterations required to reach an overall classification error less or equal to 4.5% for the five experiments carried out. The speed-ups provided by the proposed technique are reported

Experiment	Standard EBP	Proposed technique	Speed-up provided by the proposed technique
1	5851	603	9.7
2	129258	825	156.7
3	7246	603	12.0
4	4581	659	7.0
5	12637	574	22.0
Average	31915	653	41.5

Table 3
Class-by-class errors in the classification of training pixels by using the standard EBP algorithm. The overall classification error was smaller or equal to 4.5% for each of the five experiments

Experiment	Classification error (%) provided by the standard EBP				
	Wheat	Sugar beets	Potatoes	Carrots	Stubble
1	2.22	2.47	37.77	1.25	3.14
2	48.89	1.1	5.56	0.94	2.09
3	13.3	2.19	28.88	1.88	2.09
4	2.22	2.2	36.67	2.19	3.14
5	23.33	2.33	18.89	1.88	1.57
Average	17.99	2.06	25.55	1.63	2.41
Standard deviation	19.38	0.55	13.49	0.51	0.70

the potato class was quite overlapped with the sugar-beet and carrot classes. This overlapping prevented our technique from a sharp recovery of the error.

5. Conclusions

In this paper, a technique to speed up the training of MLPs in the case of imbalanced remote-sensing data sets has been presented and compared with the standard EBP algorithm. For the remote-sensing data set considered, several experiments were carried out, starting from the same randomly generated set of weights for both the standard and the proposed techniques. Results pointed out that the proposed method attained the objectives of our research by allowing notable speed-ups in comparison with the standard EBP technique for all the randomly generated weights. In addition, in the experiments performed, the proposed technique seemed to make the training process more reliable, as it provided a more stable behaviour, with respect to the initial weights, from the viewpoints of both the number of iterations required and the single-class classification accuracies; it also resulted in a better balanced distribution of the classification errors among the classes.

Finally, an MLP trained by our technique can be regarded as an implementation of the Bayes rule for minimum-error classification, as the network outputs provide approximations for posterior class probabilities.

Discussion (paper presented by Serpico)

Caelli: It just occurs to me that what you are doing may have some implications for the reformulation of the V–C dimensionality. The number of hidden units

Table 4
Class-by-class errors in the classification of training pixels by using the proposed technique. The overall classification error was smaller or equal to 4.5% for each of the five experiments

Experiment	Classification error (%) provided by the proposed technique				
	Wheat	Sugar beets	Potatoes	Carrots	Stubble
1	1.11	4.67	20.00	1.88	2.62
2	2.22	4.53	15.55	1.88	3.14
3	2.22	3.07	25.55	1.57	3.14
4	2.22	3.57	27.78	1.88	2.62
5	2.22	4.8	17.78	1.57	3.14
Average	2.00	4.13	21.33	1.76	2.93
Standard deviation	0.50	0.76	5.18	0.17	0.28

you are using and the so called V–C dimensionality of the model are relative to the data size. Have you given any thought to implications of this work to actually rethinking about the number of hidden units you use. And as I said, rethinking about V–C dimensionality?

Serpico: I did not consider these implications of this work, but I will think this over. Thank you.

Mao: The number of training patterns is fairly large in the case you presented. The output of feedforward networks can be interpreted as the a posteriori probability estimate. In the situation were you have an imbalanced population, and the training set cannot be considered as representative of the prior probabilities, do you suggest to use the measure that you propose? Suppose in some application we need the a posteriori probability estimate.

Serpico: In some sense you can use the test set and make the estimation of prior probabilities on the test set, or in a real application on the unknown set. You can apply your network and then iteratively modify the classifier in order to estimate the prior probabilities. We have applied this kind of estimation in a different problem of classification, assuming that the starting point is not very different from the final point. When you start with a classifier, there is always some underlying hypothesis about prior probabilities, at least implicitly. If the starting point is not very far from the correct values, then with some iterative technique you can try to estimate and obtain reasonable results.

Mao: If you have a larger sample size, you probably need to keep the prior distribution. You don't want to change it. Otherwise you cannot interpret your result as an a posteriori probability estimate. In your technique you basically change the a priori distribution.

Serpico: Yes, just in the first phase, while in the second phase, I use normal MSE. So, I don't change the prior probabilities in the second phase.

Acknowledgements

This research was carried out within the framework of a research project funded by the Italian Ministry of University and Scientific Research. The support is gratefully acknowledged.

References

Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S., 1993. An improved algorithm for neural network classification of imbalanced training sets. IEEE Trans. Neural Networks 4, 962–969.

Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S., 1995. Efficient classification for multiclass problem using modular neural networks. IEEE Trans. Neural Networks 6, 117–124.

Bishof, H., Schneider, W., Pinz, A.J., 1992. Multispectral classification of Landsat-images using neural networks. IEEE Trans. Geoscience and Remote Sensing 30, 482–490.

Bishop, M.C., 1996. Neural Network for Pattern Recognition. Oxford University Press, Oxford.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, New York.

Gish, H., 1990. A probabilistic approach to the understanding and training of neural network classifiers. In: Proc. 1990 Internat. Conf. Acousti. Speech Signal Process., 3–6 April, pp. 1361–1364.

Hertz, J., Krogh, A., Palmer, R.G., 1991. Introduction to the Theory of Neural Computation. Addison-Wesley, The Advance Book Program, Reading, MA.

Paola, J.D., Schowengerdt, R.A., 1995. A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification. IEEE Trans. Geoscience and Remote Sensing 33, 981–996.

Richards, J.A., 1993. Remote Sensing Digital Image Analysis, 2nd edn. Springer, Heidelberg.

Richard, M.D., Lippmann, R.P., 1991. Neural network classifiers estimate bayesian a posteriori probabilities. Neural Computation 3, 461–483.

Rojas, R., 1996. A short proof of the posterior probability property of classifier neural networks. Neural Computation 8, 41–43.

Serpico, S., Bruzzone, L., Roli, F., 1996. An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images. Pattern Recognition Letters 17, 1331–1341.

Tollenaere, T., 1990. SuperSAB: Fast adaptive back propagation with good scaling properties. Neural Networks 3, 561–573.

Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L., 1988. Accelerating the convergence of the back-propagation method. Biological Cybernetics 59, 257–263.