ELSEVIER

Neurocomputing 25 (1999) 213–217

NEUROCOMPUTING

Letter

# Total least squares for block training of neural networks

Angel Navia-Vázquez*, Aníbal R. Figueiras-Vidal

*ATSC/DTC, Universidad Carlos III de Madrid, c/Butarque, 15/28911 Leganés-Madrid, Spain*

## Abstract

This paper is intended to be a contribution to the better understanding and to the improvement of training methods for neural networks. Instead of the classical gradient descent approach, we adopt another point of view in terms of block least-squares minimizations, finally leading to the inclusion of total least-squares methods into the learning framework. We propose a training method for multilayer perceptrons which combines a reduced computational cost (attributed to block methods in general), a procedure for correcting the well-known sensitivity problems of these approaches, and the layer-wise application of a total least-squares algorithm (high resistance against noise in the data). The new method, which we call reduced sensitivity total least-squares (RS-TLS) training, demonstrates good performance in practical applications.

*Keywords:* Perceptron; Noise; Total least-squares; Block training

## 1. Introduction

Although multilayer perceptrons (MLP) are powerful schemes for dealing with nonlinear decision and estimation problems, their drawbacks limit their practical application. These drawbacks include the difficulties of training, and of scaling to larger networks.

MLP training is difficult, not only due to local extrema problems, but also because of computational requirements, which can be unacceptable for recurrent structures. Block training formulations constitute an attractive way to reduce these requirements by means of a stepwise (suboptimal) layer-by-layer solution of double least-squares

* Corresponding author. Tel.: + 34-1-624-9903; fax: + 34-1-624-9430.

*E-mail addresses:* navia@ing.uc3m.es (A. Navia-Vázquez), arfv@ing.uc3m.es (A.R. Figueiras-Vidal)

0925-2312/99/$-see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0925-2312(99)00107-1

minimization problems. The authors in [1] call this method the least-squares backpropagation (LSB) algorithm, and demonstrate the performance improvements that can be obtained by using it. The procedure mainly consists of applying the inverse of the activation function to propagate errors back to the preceding layer and then solving LS problems at each layer. If we store training patterns as rows of matrices, the update equations for layer $k$ are[1]

$$\boldsymbol{Z}^{(k)} = \boldsymbol{O}^{(k-1)}\boldsymbol{W}^{(k)}, \tag{1}$$

$$\boldsymbol{O}^{(k)} = \tanh \boldsymbol{Z}^{(k)}, \tag{2}$$

where every row in $\boldsymbol{O}^{(k)}$ is a vector storing neuron outputs at layer $k$, $\boldsymbol{Z}^{(k)}$ stores state values and $\boldsymbol{W}^{(k)}$ is the weight matrix for that layer. Following this notation, $\boldsymbol{O}^{(0)}$ stores input patterns (layer 0) and $\boldsymbol{T}^{(K)}$ stores the corresponding desired outputs (i.e., an MLP with $K$ layers is considered). In the LSB algorithm, the inverse of the activity function is applied to target outputs $\boldsymbol{T}^{(k)}$ in order to compute target states:[2]

$$\boldsymbol{Q}^{(k)} = \tanh^{-1}(\boldsymbol{T}^{(k)}). \tag{3}$$

Then, the optimal weights for that layer (matrix $\hat{\boldsymbol{W}}^{(k)}$) are obtained by solving

$$\min_{\hat{\boldsymbol{W}}^{(k)}} \|\boldsymbol{O}^{(k-1)}\hat{\boldsymbol{W}}^{(k)} - \boldsymbol{Q}^{(k)}\|_2. \tag{4}$$

Once these new weights are computed, a second minimization problem is solved to obtain output target values for the previous layer ($\boldsymbol{T}^{(k-1)}$)

$$\min_{\mathrm{T}^{(k-1)}} \|\boldsymbol{T}^{(k-1)}\hat{\boldsymbol{W}}^{(k)} - \boldsymbol{Q}^{(k)}\|_2. \tag{5}$$

Thus, a single training epoch is completed after applying this three-step procedure at layers $K, K-1, \dots, 1$. Training epochs are repeated until an appropriate convergence criterion is reached.

A more detailed evaluation and study of this approach shows that sensitivity problems appear when transferring objectives for training the previous layer through the nonlinearities, as discussed in the next section. We have analyzed this problem in detail in [2,3], proposing several modifications which allow the inconvenient effects associated with block training to be limited. Here, we will apply our reduced sensitivity (RS) formulation, which yields the best general characteristics and results.

## 2. The RS–TLSB algorithm

It is not difficult to see that if the output values of each activation function are backpropagated using its inverse, sensitivity problems will appear. Specifically, the

[1] We have not included the bias terms for ease of explanation.

[2] The LSB algorithm also incorporates a rescaling procedure to ensure that $\boldsymbol{T}^{(k)}$ always lies in $(-1,1)$. We have omitted its description here because it is not vital for the understanding of the rest of the paper.

(usual) activation function has saturation regions, and its inverse works poorly in these areas (the linear regions do not cause difficulties). The results of this sensitivity problem can be different according to the particular case. For example, the LSB algorithm may converge to a suboptimal solution, related to smoothed outputs (i.e., the MLP is trained to "overgeneralize", failing to capture some local characteristics), or it may "solve" the situation using exceedingly large weight values (e.g., on the order of $10^6$), which are not acceptable for practical uses [2,4]. If we use the slope of the activation function, which is a "sensitivity" parameter, to weight the corresponding error being minimized, we compensate for the aforementioned sensitivity effect. This is what we called reduced sensitivity-least-squares block (RS-LSB) training algorithm [3].

Other problems arise when applying RS-LSB if the data are noisy since many common techniques (Moore–Penrose pseudoinverse, QR decomposition, etc.) which can be used to solve the LS equations ((4) and (5)) are not robust in the presence of noise. A singular value decomposition (SVD) can also be applied in the usual way to solve the LS minimizations; however, we can obtain an additional advantage if we use the extra information provided by SVD to obtain TLS solutions [5], accounting for errors in the input variables and finding improved solutions when dealing with noisy inputs.

When using TLS in the framework presented above to train an MLP in a blockwise manner, we are constructing a "reduced sensitivity – total least-squares block" (RS-TLSB) training algorithm (TLSB, if RS is not applied). One can expect better performance than that provided by an RS-LSB (or an LSB) when the data are noisy, maintaining keeping the advantages of LSB with respect to classical back-propagation (BP) algorithms.

## 3. Evaluation of the algorithms

In order to verify the predicted results, we applied BP, LSB, TLSB, RS-LSB, and RS-TLSB to a classical problem: the prediction of a standard series, generated by the logistic map $y_n = 4y_{n-1}(1 - y_{n-1})$, when the observations of this series are corrupted by a zero mean Gaussian noise. The selected architecture is a single hidden layer MLP with two neurons; initial weights are random in $(-0.1, 0.1)$, and the number of training patterns is 100. In spite of the simplicity of the experiment, the results are representative of the general behavior of TLS block training methods (we have obtained similar results in more complex real-world problems).

Fig. 1 shows the squared prediction error averaged over 50 experiments as a function of the signal to noise ratio (SNR). Note the clear advantage of using TLS-based training (with SVD decomposition) with respect to LS block training in the SNR margin from 5 to 20 dB. We can obtain an error reduction of several dB, which is clearly higher than the typical 0.8–1 dB obtainable in linear cases [6] (i.e., cases where a linear model can be successfully used to represent the data). The performance of RS-TLSB in this SNR region is particularly remarkable. It appears that TLS takes

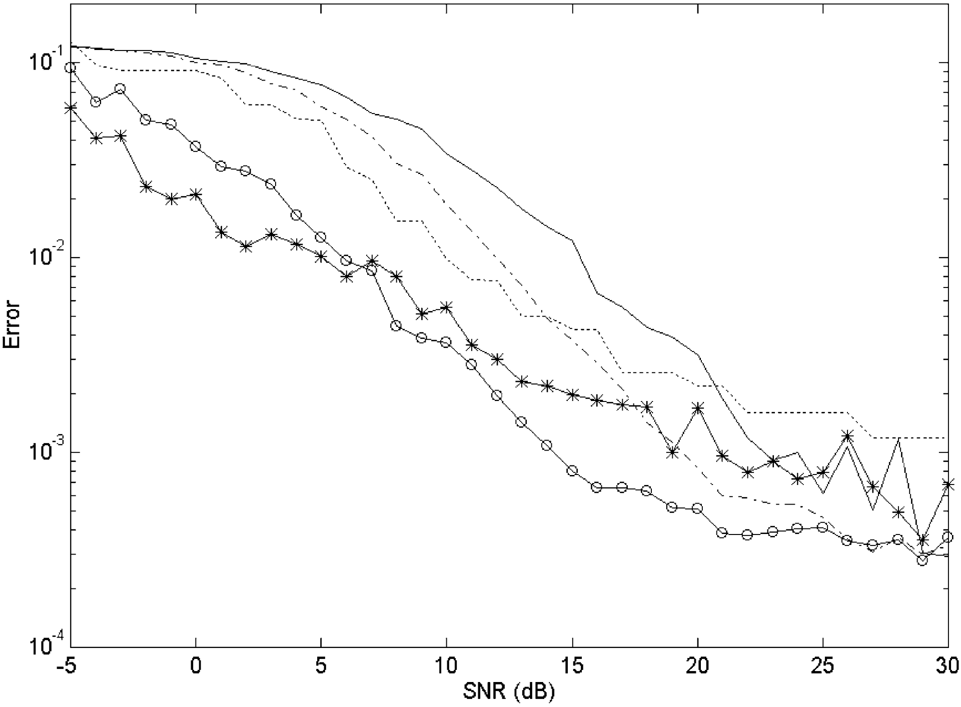

Fig. 1. Modelling error for the LSB ("- -"), TLSB ("-*-"), RS-LSB ("- · -"), RS-TLSB ("-○-"), and BP ("...") algorithms in the nonlinear prediction problem (logistic time series) when training data are embedded into Gaussian noise, with the SNR ranging from − 5 to 30 dB.

advantage of the "localization" properties found in "RS" algorithms, where each neuron focuses on a region of its input space [3].

For SNR higher than 25 dB, TLS- and LS-based methods tend to perform similarly (noise at the input is negligible and both methods yield analogous results), as expected. Finally, for very low SNR ( − 5–5 dB), TLSB is slightly better than RS-TLSB. This is mainly due to the fact that the TLS algorithm is affected by the inaccuracy in computing sensitivities, due to the presence of strong noise components in the inputs.

From the point of view of computational effort, the use of a TLS technique (SVD) introduces here an approximate factor of five with respect to an LS technique (QR decomposition); nevertheless, in our practical case, and taking into account the number of epochs needed to converge, TLSB and RS-TLSB methods required less computation than BP (RS-LSB and LSB require only 15% of that load). The increase in computational effort is justified by the improved quality of the solutions, because algorithms incorporating TLS are mainly intended for applications where computational load is not at a premium (e.g., financial prediction, credit approval, etc.), and even moderate benefits are welcomed.

## 5. Conclusions

A new MLP training method, the reduced sensitivity-total least-squares block (RS-TLSB) algorithm, has been presented and evaluated. It combines the advantages of block training (lower computational cost as compared to more classical approaches) with the improved performance achieved using sensitivity reduction, and the robustness to noise attributed to TLS techniques. We have shown by means of a simple (but representative) example that the reduction in modelling error is considerable, which justifies the extra (moderate) computational load needed to compute SVD decompositions.

Block training procedures represent also a very attractive aproach to recurrent MLP training, where the computational requirements are a major problem; we are currently developing approaches for this type of training as well.

## References

[1] F. Biegler-König, F. Bärmann, A learning algorithm for multilayered neural networks based on linear least squares problems, Neural Networks 6 (1993) 127–131.

[2] A. Navia-Vázquez, A.R. Figueiras-Vidal, Improving the performance of block Least Squares training for multilayer perceptrons, in: Proceedings of the IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks, vol. 1, M.H. Hamza (ed.), Hawaii, USA, 1996, pp. 82–85.

[3] A. Navia-Vázquez, A.R. Figueiras-Vidal, Efficient block training of multilayer perceptrons, Neural Comput. submitted for publication.

[4] A. Navia-Vázquez, A.R. Figueiras-Vidal, Block training methods for perceptrons and their applications, in: Proceedings of the SPIE Aerosense International Conference: Applications of Artificial Neural Networks II, T.S.K. Rogers (ed.), vol. 3077, Orlando, USA, 1997, pp. 600–610.

[5] G.H. Golub, A. Hoffman, G.W. Stewart, A generalization of the Eckart–Young–Minsky matrix approximation theorem, Linear Algebra Appl. 88/89 (1987) 322–327.

[6] S. Van Huffel, H. Zha, The total least squares problem, in: C.R. Rao (Ed.), Handbook of Statistics, Elsevier, Amsterdam, NL, 1993, pp. 377–408.

**A. Navia-Vázquez** is a research assistant at the Department of Communication Technologies, Universidad Carlos III de Madrid. His main interests are devoted to the study of nonlinear methods and their application to signal processing problems. His Ph.D. Thesis focused on block learning methods for neural networks and their application to nonlinear modeling and prediction.

**A.R. Figueiras-Vidal** is a full professor at the Department of Communication Technologies, Universidad Carlos III de Madrid. His research interests are focused on Digital Signal Processing, Digital Communications, Data Mining using Neural Networks, and Evolutionary Optimization. He is the author of more than 200 papers in these areas.