PROCEEDINGS REPRINT

SPIE—The International Society for Optical Engineering

Reprinted from

Applications and Science of Artificial Neural Networks II

9–12 April 1996
Orlando, Florida

Volume 2760

Improving Learning Speed in Multi-Layer Perceptrons through Principal Component Analysis

Francesco Masulli and Massimo Penna

Istituto Nazionale di Fisica della Materia and
Dipartimento di Fisica - Università di Genova
Via Dodecaneso 33 - 16146 Genova - Italy
email : {masulli| penna}@ge.infm.it

Abstract

This paper describes an application of Principal Component Analysis (PCA) to the speeding-up the learning of a Multi-Layer Perceptron (MLP). A training algorithm, called the Incremental Input Dimensionality (IID) method, is presented that is constituted by some training steps, in each of which the dimension of the principal component subspace is increased. For each training step, some training epochs (presentations of the training set), using the Back-Propagation algorithm, are performed in order to reduce the mean square error (MSE) on the test set. In this way, the last training step is performed with a subspace corresponding to the assigned reconstruction error. The performances of the MLP using IID, in the case of handwritten digit classification, are reported. For our data-base a choice of a reconstruction error rate of 5% in the IID algorithm implies a maximum dimension of the principal components subspace equal to 37. In the experiments reported in this paper, Back-Propagation using IID has turned out to be faster than standard Back-Propagation with a speed-up of about 73%. Moreover, as the IID method concerns only data representation, it can be combined with other speed-up techniques for MLP learning, and can be used by other classifiers.

Keywords: Incremental Input Dimensionality Method; Principal Component Analysis; Classification; Multi-Layer Perceptron; Learning Speedup; Handwritten Character Recognition.

1 INTRODUCTION

The feature extraction step is a key-stage in pattern recognition, especially when the pattern space shows a huge dimensionality.[1–4] The selection of optimal features, providing all the information necessary for the classification task, permits a reduction in the computational load of the classifier, in particular, in the cases where iterative learning algorithms are employed, such as the Back-Propagation algorithm for Multi-Layer Perceptrons (MLPs).[5,6]

A possible approach to the feature extraction step can be based on the use of *Principal Component Analysis* (PCA).[7–10] Principal Component Analysis is a statistical method for extracting features from high-dimensional data distributions. It is also known in signal processing as the *Karhunen-Loéve Transform* (KLT), or as the

0-8194-2141-3/96/$6.00

Hotelling Transform in image processing. It allows a linear transformation of the original pattern space into a new orthogonal space of reduced dimension, and ensures a minimal loss of information.[11,12]

In neural-network literature some methods for obtain PCA by using neural networks have been proposed.[13–15] Moreover non linear PCA networks have been proposed in order to extract higher-order statistics and add robustness to the expansion.[16–19]

In this paper, the *Incremental Input Dimensionality* (IID) method is presented. It has been derived from an article by Malki and Moghaddamjoo[20] and is based on PCA. IID includes some *training steps*, in each of which the dimension of the principal component subspace is increased. For each training step, some *training epochs* (presentations of the training set), using the Back-Propagation algorithm, are performed in order to reduce the mean square error (MSE) on the test set. The last training step is performed with a subspace corresponding to the assigned reconstruction error rate. Moreover, once the training is ended, it is possible to include the calculation of PCA into the weights of the first hidden layer of the MLP.

We apply the IID method to speed up of the training of a Multi-Layer Perceptron using the Back-Propagation algorithm, in a handwritten digits recognition task. In the experiments reported in this paper, Back-Propagation using IID (IID-BP) turned out to be faster than standard Back-Propagation (S-BP) with a speed-up of about 73% and, thanks to the smaller number of parameters of the input layer of the Multi-Layer Perceptron and to the filtering effect on the pattern noise, has led to a slightly better generalization. Moreover, as the IID method concerns only data representation, it can be used by other classifiers, and can be combined with other speed-up techniques for MLP learning, based , e.g., on adaptive learning rate, momentum,[21] second order methods,[22] data query or editing.[23]

The Incremental Input Dimensionality is presented in the next section, while the the data sets and the preprocessing of patterns are described in Section 3. In Section 4, experimental results are reported and discussed. Finally, conclusions are drawn in Section 5.

2 THE INCREMENTAL INPUT DIMENSIONALITY METHOD

Principal Component Analysis (PCA)[7,11] is a linear, orthogonal transformation of a distribution from the original space X of dimension N into a new coordinate system $U = \mathbf{u_1}, ..., \mathbf{u_N}$, in which the coordinates are uncorrelated and maximal amount of variance of the original distribution is concentrated only on a small number of coordinates. In this transformed space, one can reduce the number of variables by taking only the coordinates on which the variance is concentrated (say M), and one can minimize the loss of variance by leaving out the coordinates with small variances ($N - M$). The basis vectors of this new coordinate system are the eigenvectors of the covariance matrix $\mathbf{K_X}$ and the variance on these coordinates are the corresponding eigenvalues $\lambda_i, i \in [1, N]$. The optimal projection from N to M dimensions given by PCA is therefore the subspace of the M eigenvectors with the largest eigenvalues. The *reconstruction error* D, defined as the expectation of the difference between a pattern $\mathbf{x} \in X$ and its reconstruction after projection in the PCA subspace $\tilde{\mathbf{x}}$, is equal to[11]:

$$D = \sum_{i=M+1}^{N} \lambda_i. \tag{1}$$

Let we define also the *reconstruction precision rate* γ as :

$$\gamma = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{2}$$

and the *reconstruction error rate* as $1 - \gamma$.

The optimal dimension M^* of the subspace of characteristics depends on the redundancy of the information carried by data. Very redundant data will exhibit few high eigenvalues and many low eigenvalues. In case of classifiers based on neural networks, a choice of a dimension $M >> M^*$ would involve an increase in the duration of the training, without obtaining a corresponding improvement in generalization. On the other side, if $M << M^*$, the classifier will lack important information for its task, hence it would take a long time to organize, in a suitable way, the available information, and no acceptable generalization would be obtained.

In the following of this section we apply PCA to the speeding-up of the Back-Propagation training algorithm for Multi-Layer Perceptrons (MLPs). The proposed training algorithm, called Incremental Input Dimensionality (IID), has been derived from an article by Malki and Moghaddamjoo.[20]

The reduction of the number of characteristics obtained by using PCA, involves the decreasing of the number of parameters (weights) connecting the input layer and the first hidden layer of the neural network, thus speeding up the learning.[8] Moreover, as the characteristics obtained by PCA are uncorrelated and orthogonal to one another, the learning can be performed in separate steps, in each of which the dimension of the principal component subspace is increased, thus further decreasing the learning time, as proposed by Malki e Moghaddamjoo.[20]

The Incremental Input Dimensionality method when applied to the speed-up the Back-Propagation algorithm (IID-BP) consists of the following steps:

1. The covariance matrix $\mathbf{K_X}$ of the L vectors of the training set $\{\mathbf{x}_l \mid l = 1, ..., L\}$ (each of dimension N) is computed and then diagonalized.
2. The matrix $\mathbf{Q}$ is obtained, whose columns are the eigenvectors of $\mathbf{K_X}$, arranged according to the decreasing eigenvalues $\{\lambda_j, j = 1, ..., N\}$.
3. The value of reconstruction precision rate γ is assigned and the dimension M of the final characteristic sub-space M is obtained.
4. The eigenvalues are grouped into I clusters:
$$\{\lambda_j \mid j \in [1 + m_{i-1}, m_i]\} \quad i = 1, ..., I. \tag{3}$$
5. The data vectors are transformed into the space of characteristics:
$$\bar{\mathbf{x}}_i = \mathbf{Q}^t \mathbf{x}_i \tag{4}$$
6. A MLP is trained in I *training steps*. For each i-th training step the dimension of the characteristics space (and the input layer of the MLP) is m_i (obtained by the addition of a cluster of characteristics to the characteristics space of the previous training step), and the training is performed by Back-Propagation. The last training step ends when the MSE calculated on the test set is minimized.
7. The learned weighs $\bar{\mathbf{W}}$ are converted into the ones of the equivalent MLP $\mathbf{W}$ that will be used directly on the original data vectors $\mathbf{x}$, by setting to zero the weights corresponding to the inputs $M + 1 .. N$, and by performing the product:
$$\mathbf{W} = \bar{\mathbf{W}} \mathbf{Q}^t. \tag{5}$$

Let we denote:
$$T = \sum_{j=1}^{N} \lambda_j \text{ (trace of } \mathbf{Q}), \quad \tilde{T} = \sum_{j=1}^{M} \lambda_j \text{ , and } \quad T_1 = \sum_{j=1}^{m_1} \lambda_j. \tag{6}$$

Then the reconstruction error and the reconstruction precision rate can be expressed as:
$$D = T - \tilde{T} \quad \text{and} \quad \gamma = \frac{\tilde{T}}{T}. \tag{7}$$

A critical aspect of the IID is the selection of clusters (step 4). The first cluster must carry sufficient information, in order to initialize efficiently the learning. The remaining clusters can carry an equivalent amount of information, subject to the constraint of a small reduction between the first and the last eigenvalue of each cluster.[20] We used the following heuristics :

- For the first group m_i was selected in order to obtain:

$$\frac{T_1}{T} \geq \frac{P_1}{P}, \tag{8}$$

where $P1$ and P are, respectively, the number of weights involved with the first m_1 components, and the one with the original perceptron with N inputs.

- The number of cluster I and the dimension of each cluster were obtained by imposing

$$\sum_{j=1+m_{i-1}}^{m_i} \lambda_j \sim \frac{\tilde{T} - T_1}{I-1} \quad \text{and} \quad \lambda_{m_i} \sim 2\,\lambda_{1+m_{i-1}} \qquad i = 2\,,...,\,I. \tag{9}$$

3 DATA BASE AND PREPROCESSING

All the experiments reported in this paper were carried out on a SUN 10/50 workstation. We used a training set and a test set extracted from the NIST-3 data-base[24] containing 313389 characters coded as 128×128 binary-matrix images and labeled by the corresponding ASCII codes.

Both the training set and the test set were made up of 10,000 associative pairs of segmented handwritten digits each, obtained from disjoint groups of writers.

The preprocessing included the following steps:

1. Digit image extraction from the CD-ROM and normalization to a 32×32 binary matrix.
2. Low-pass filtering in order to remove some small spots and holes from the image.
3. Application of a shear transform to the digit image to straighten the axis joining the first upper-left point of the digit image to the last lower-right point.
4. Image skeletonization by using a thinning algorithm.[25]
5. Finally, transformation of the digit representation into a 64-element vector, each vector element representing the number of black pixels contained in adjacent 4×4 squares (local counting).

It is worth noting that the resulting digit representation exhibits sufficient degrees of invariance to both scale and small image shifts or rotations.

4 EXPERIMENTAL RESULTS AND DISCUSSION

Figure 1 shows, from upper left, the first 9 eigenvectors, associated with the higher eigenvalues, by assigning the absolute black to the largest positive value of a component and the absolute white to the largest negative value.

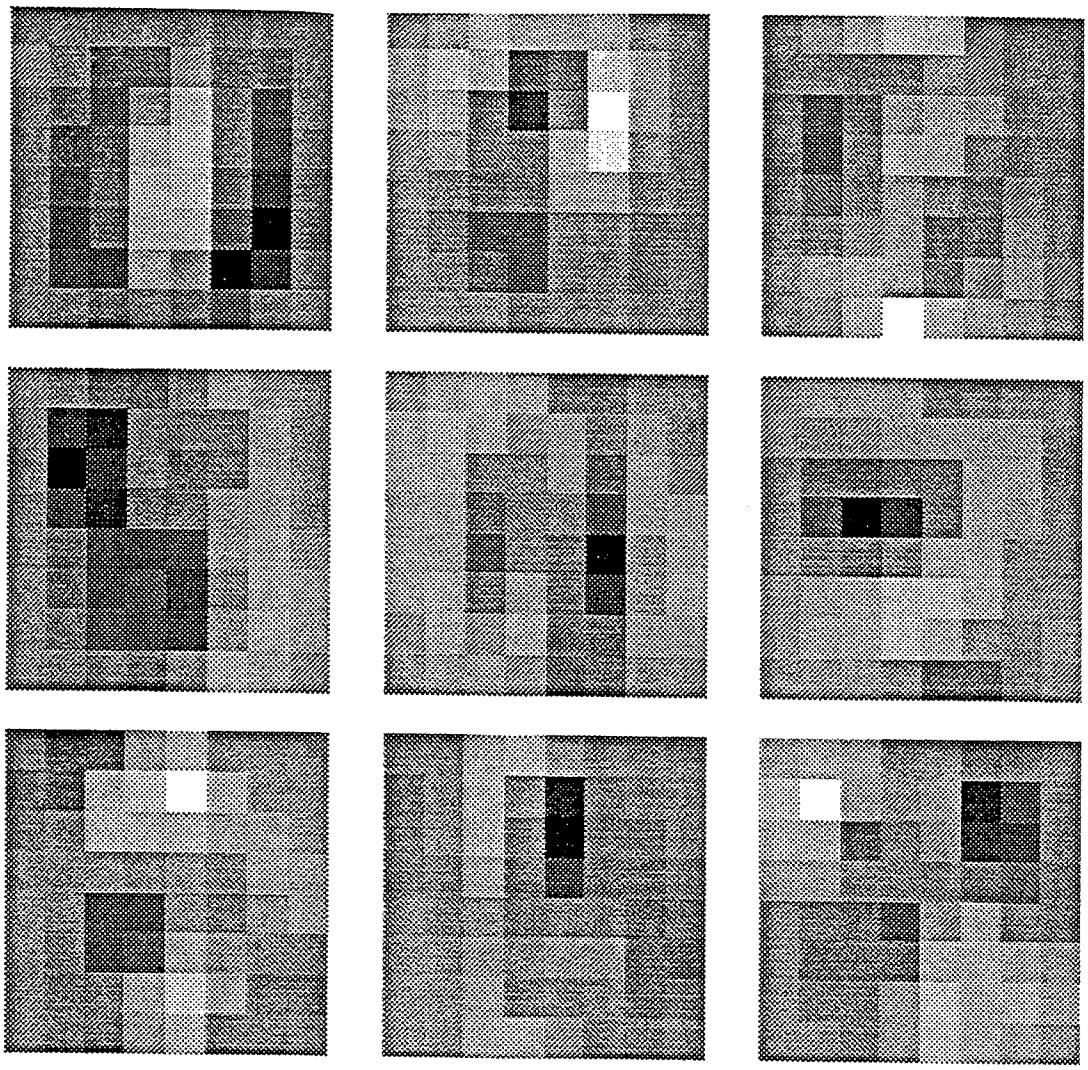

Figure 1: The first 9 principal eigenvectors.

The projection of the patterns onto the basis made up of the components of PCA, constitutes a kind of *feature detection*, where the feature extractors are obtained by a statistical analysis of the training set. In fact, it is possible to identify in such patterns precise shapes that will be enhanced by the projection of the patterns into the space of principal components: e.g., in Figure 1 the principal eigenvector enhances the circle of the the digit "O", the third eigenvector enhances the three horizontal bands of digits "3" and "5", and so on.

Figure 2 shows the behavior of the reconstruction precision as a function of the dimension of the principal component subspace for the patterns of the training set.

We compared the training time and the generalization of a MLP trained by Standard Back-Propagation (S-BP) and of a MLP trained by Back-Propagation using Incremental Input Dimensionality (IID-BP). The two MLPs were made up by 64 inputs, 32 neurons in the first hidden layer, 16 neurons in the second hidden layer, and 10 neurons in the output layer, corresponding to the 10 classes of digits. For the MLP trained with IID-BP, we used a final subspace of 37 characteristics, corresponding to a γ value of 0.95 (see Figure 2). The training with IID-BP was performed in 4 training steps, by using a subspace of 7, 12, 21, and 37 principal components, respectively.

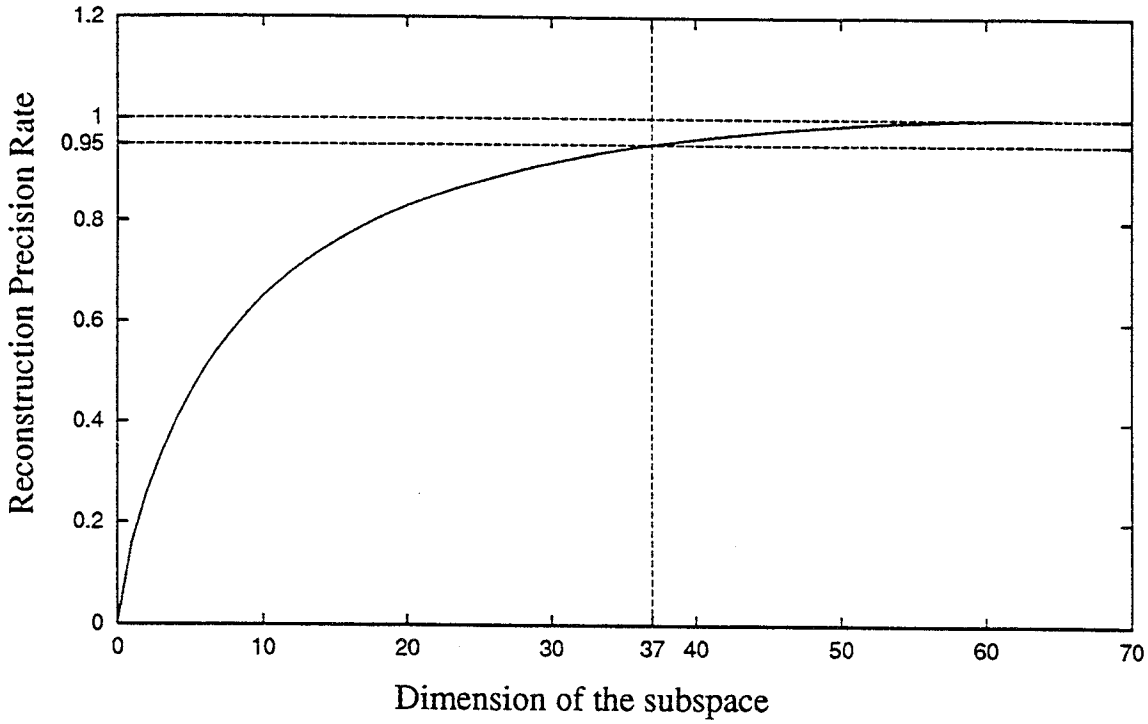

Figure 2: Reconstruction precision rate as a function of the dimension of the principal component subspace.

After training, the dimension of the input layer returned to 64, and the weight of the first layer included the calculation of the projection of inputs into the 37-dimensional subspace. In both cases the Back-Propagation used adaptive learning rate, moment and the weight were updated by epoch.[6] The stopping criteria for training was the minimum of the generalization error. A rejection decision was applied to patterns with a difference between the two higher output of the MLP smaller than .5.

Figure 3 shows examples of the original digits of the NIST-3 data-base, the features extracted as described in Section 3 (used for the training with S-BP), and the reconstruction of digits by using their 37 principal components (used for the training with IID-BP).

Table 1 gives training times and recognition rates on the test set, averaged over 10 executions, for both S-BP and IID-BP. The IID-BP method reached the best generalization results. In order to compare more precisely the execution times, we selected the 6 tests by the S-BP giving best generalization results, and the 6 tests by the IID-BP giving worst generalization results. Table 2 shows the new averaged results. The learning speed-up is 73%.

Table 3 and 4 show the confusion matrices related to the learning, respectively, by S-BP and by IID-BP. The rows refer to the labels of the patterns in the test set, the number of examples for class, and the distribution of the classification among the classes (included the rejection class) obtained by the MLP. The results are almost identical, even if IID-BP shows a slightly higher accuracy. The small improvement in generalization can be ascribed to the reduction of the number of the parameters to be learned, and to the reduction of the uncorrelated pattern noise, that is assumed to be spread over all components.

In figure 4 the MSE on the test set, for typical trainings with both S-BP and IID-BP, are reported. The reduction of relative minima on the MSE with IID-BP, can be again ascribed to the reduction of pattern noise.

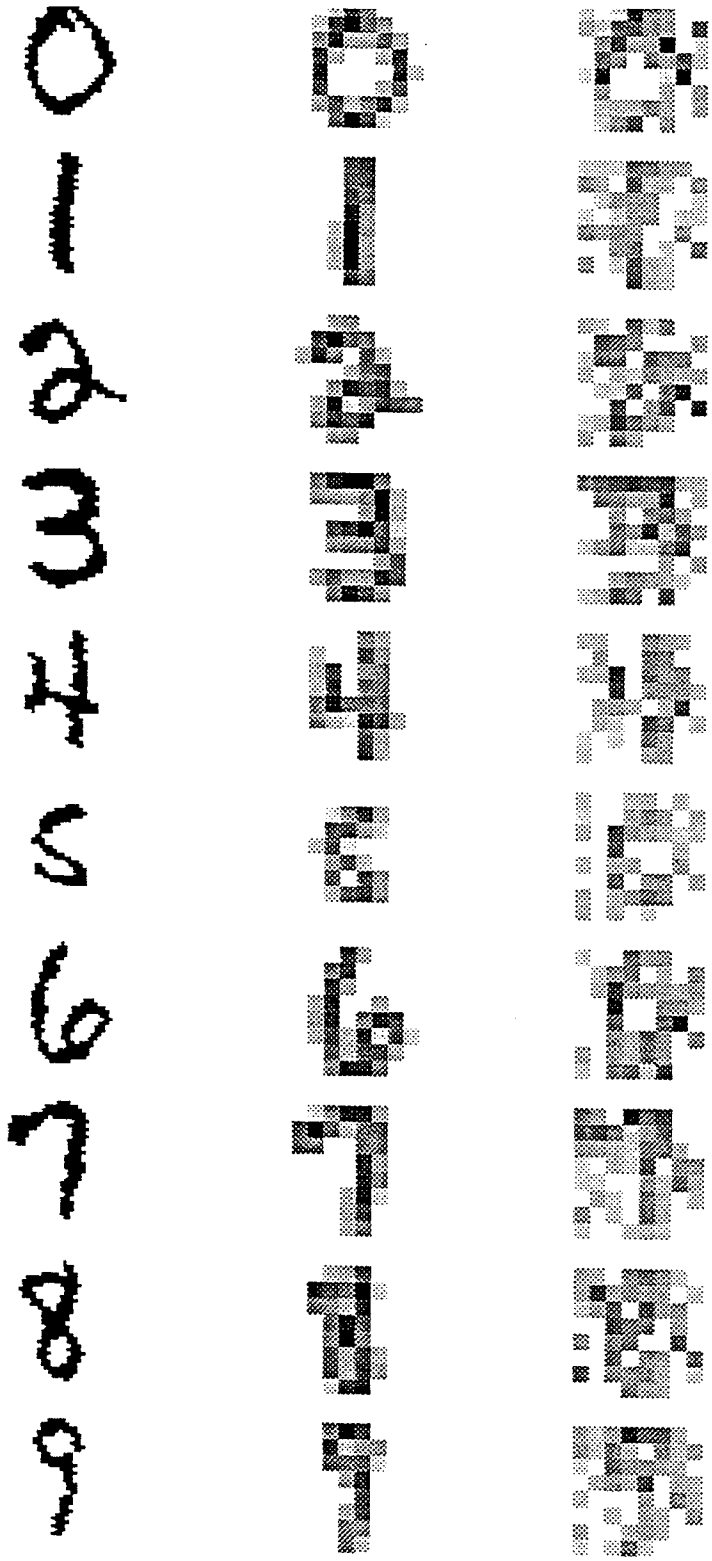

Figure 3: The three columns show some examples of the original data of the NIST-3 data-base, the results of the preprocessing (see Section 3), and of the reconstruction of digits from their projections into the 37 principal components subspace.

	Learning Time (sec)	Generalization Rate
S-BP	18290 ± 850	94.00 ± 0.04
IID-BP	11780 ± 535	94.15 ± 0.03
10 tests, $T_{S-BP}/T_{IID-BP} = 1.55 \pm 0.14$		

le 1: Learning times and recognition rates on the test set, averaged on 10 training by using S-BP and IID-BP.

	Learning Time (sec)	Generalization Rate
S-BP	19080 ± 1340	94.10 ± 0.02
IID-BP	11019 ± 690	94.10 ± 0.04
6 tests, $T_{S-BP}/T_{IID-BP} = 1.73 \pm 0.23$		

Table 2: The same as Table 1, but computed on subsets of 6 tests.

	Examples	Cl 0	Cl 1	Cl 2	Cl 3	Cl 4	Cl 5	Cl 6	Cl 7	Cl 8	Cl 9	Rej
Cl 0	1019	96.08	0.07	0.32	0.20	0.15	0.35	0.52	0.02	0.26	0.06	1.96
Cl 1	1127	0.04	97.53	0.11	0.00	0.03	0.09	0.13	0.15	0.26	0.07	1.59
Cl 2	982	0.19	0.03	90.74	1.17	1.46	0.68	0.54	0.37	0.67	0.11	4.03
Cl 3	1049	0.15	0.03	1.10	93.52	0.10	0.80	0.00	0.31	0.60	0.54	2.85
Cl 4	944	0.12	0.04	0.69	0.01	92.56	0.22	0.47	0.04	0.61	1.43	3.80
Cl 5	870	0.66	0.07	0.41	1.64	0.14	90.11	0.29	0.02	1.25	0.14	5.26
Cl 6	980	0.24	0.04	0.57	0.07	0.05	0.60	95.35	0.00	0.20	0.03	2.84
Cl 7	1035	0.05	0.34	0.23	0.09	0.25	0.02	0.00	95.66	0.25	1.26	1.86
Cl 8	1011	0.23	0.37	0.70	0.97	0.46	0.76	0.08	0.10	91.65	0.78	3.90
Cl 9	983	0.08	0.00	0.06	0.13	0.72	0.21	0.00	1.04	0.52	95.36	1.87

Table 3: Confusion matrix averaged over 10 MLPs trained independently by the S-BP.

	Examples	Cl 0	Cl 1	Cl 2	Cl 3	Cl 4	Cl 5	Cl 6	Cl 7	Cl 8	Cl 9	Rej
Cl 0	1019	96.25	0.05	0.33	0.32	0.10	0.35	0.39	0.01	0.10	0.05	2.04
Cl 1	1127	0.00	97.79	0.13	0.04	0.04	0.12	0.18	0.12	0.15	0.06	1.38
Cl 2	982	0.36	0.02	91.34	0.84	1.27	0.49	0.41	0.34	0.76	0.20	3.97
Cl 3	1049	0.12	0.06	0.80	93.98	0.03	0.75	0.02	0.30	0.53	0.40	3.01
Cl 4	944	0.13	0.06	0.58	0.03	92.85	0.20	0.14	0.08	0.47	1.38	4.08
Cl 5	870	0.64	0.10	0.31	1.36	0.13	90.10	0.20	0.02	1.09	0.13	5.92
Cl 6	980	0.27	0.08	0.59	0.08	0.10	0.38	95.42	0.00	0.20	0.00	2.88
Cl 7	1035	0.01	0.30	0.28	0.02	0.19	0.02	0.00	95.81	0.22	1.49	1.66
Cl 8	1011	0.24	0.42	0.41	0.76	0.52	0.58	0.05	0.08	91.58	0.84	4.52
Cl 9	983	0.09	0.00	0.04	0.06	0.77	0.14	0.00	1.11	0.41	95.06	2.32

Table 4: Confusion matrix averaged over 10 MLPs trained independently by the IID-BP method.

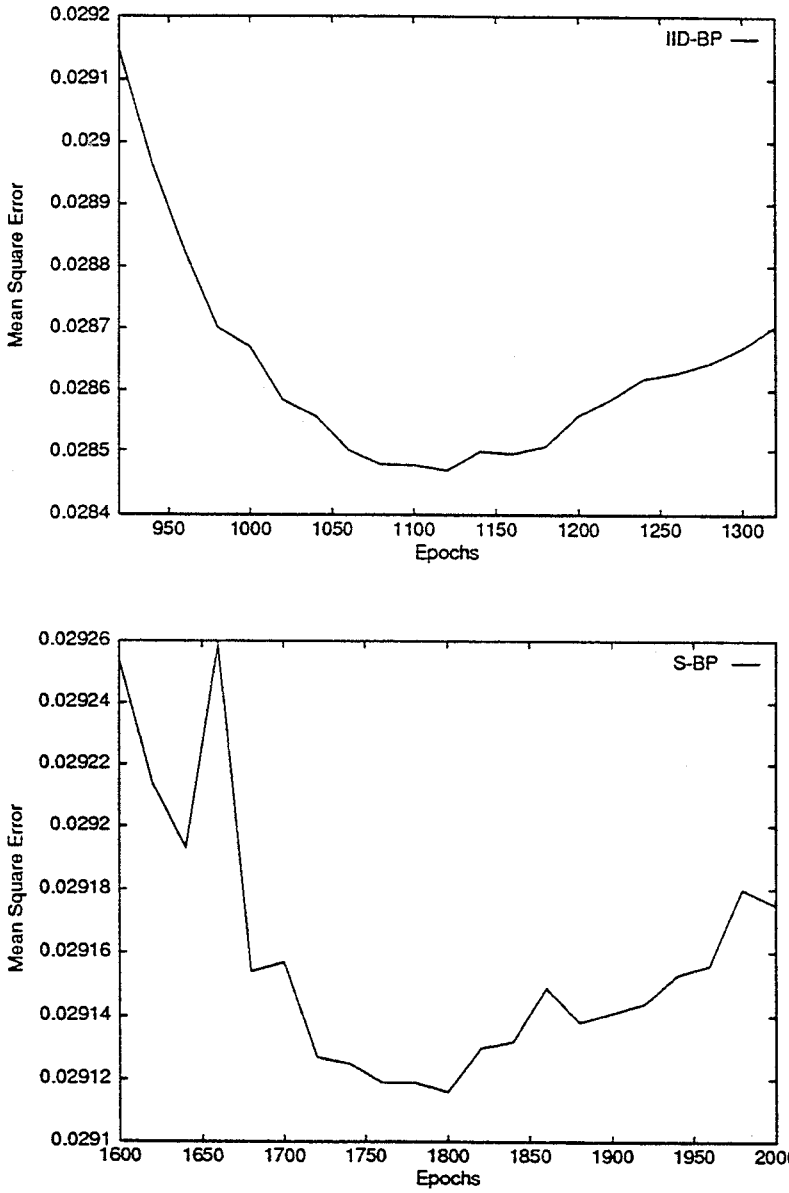

Figure 4: Mean square error versus number of epochs, near the minimum of MSE, by using IID-BP, and S-BP.

5 CONCLUSIONS

In this paper the Incremental Input Dimensionality (IID) method, based on Principal Component Analysis, has been applied to speed-up the learning of a Multi-Layer Perceptron.

The Incremental Input Dimensionality (IID) method, consists in some *training steps*, in each of which the dimension of the principal component subspace is increased. For each training step, some *training epochs*, using the Back-Propagation algorithm, are performed in order to reduce the mean square error (MSE) on the test set. In this way, the last training step is performed with a subspace corresponding to the assigned reconstruction error.

In the experiments reported in this paper, Back-Propagation using IID (IID-BP) turned out to be faster than standard Back-Propagation (S-BP) with a speed-up of about 73%, and, thanks to the smaller number of parameters of the input layer of the Multi-Layer Perceptron and to the filtering effect on the pattern noise, has led to a small improvement in generalization. Moreover, as the IID method concerns only data representation, it can be combined with other speed-up techniques for MLP learning, and can be used by other classifiers.

6 ACKNOWLEDGMENTS

This work was supported by grants from CNR-Progetto Strategico Reti Neurali, GNCB-CNR, INFM, and MURST. We thank Alessandro Sperduti and Fabrizio Vannucci for helpful discussions.

7 REFERENCES

[1] K.K. Paliwal, "Dimensionality reduction of the enhanced feature set for the HMM-based speech recognizer", *Digital Signal Processing*, vol. 2, pp. 157–173, 1992.

[2] R. Battiti, "Using mutual information for selecting features in supervised neural net learning", *IEEE Transactions on Neural Networks*, vol. 5, pp. 537–550, 1994.

[3] R. Etemad, K.and Chellappa, "Dimensionality reduction of multi-scale feature spaces using a separability criterion", *in 1995 International Conference on Acoustics, Speech, and Signal Processing. Conference Proceedings*, pp. 2547–2550, Detroit, MI, USA, 1995. IEEE, New York, NY, USA.

[4] L. Jimenez and D.A Landgrebe, "Projection pursuit in high dimensional data reduction: initial conditions, feature selection and the assumption of normality", *in 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 1, pp. 401–406, Vancouver, BC, Canada, 1995. IEEE, New York, NY, USA.

[5] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, MIT Press, Cambridge, 1986.

[6] S. Haykin, *Neural Networks : a Comprehensive Foundation*, Macmillan, New York, Toronto, 1994.

[7] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London, 1982.

[8] P.J. Grother, "Karhunen Loève feature extraction for neural handwritten character recognition", Technical Report 4824, National Institute of Standard and Technology, Gaithesburg, MD USA, 1992.

[9] J. Bigun, "Unsupervised feature reduction in image segmentation by local transforms", *Pattern Recognition Letters*, vol. 14, pp. 573–83, 1993.

[10] Ming-Wen Chang, Bor-Shenn Jeng, Dung-Ming Shieh, and Shih-Fu Shy, "Feature-based noise reduction in preprocessing for optical chinese handwritten character recognition", *in Applications of Digital Image Processing XVII - Proceedings of the SPIE*, vol. 2298, pp. 624–633, San Diego, CA, USA, 1994. SPIE.

[11] C.W. Therrien, *Decision, Estimation and Classification*, John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1989.

[12] P. Földiák, "Models of sensory coding", Technical report, CUED/F-INFENG/TR 91, Physiological Laboratory, University of Cambridge, Cambridge - U.K., 1991.

[13] E. Oja, "A simplified neuron model as a principal component analyzer", *Journal of Mathematical Biology*, vol. 15, pp. 267–273, 1982.

[14] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix", *Journal of Mathematical Analysis and Applications*, vol. 106, pp. 69–84, 1985.

[15] T.D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network", *Neural Networks*, vol. 2, pp. 459–473, 1989.

[16] E. Oja, "Neural networks, principal components and subspaces", in T. Kohonen, K. Makisara, and O. Simula, editors, *Artificial Neural Networks*, vol. 1, pp. 737–746, North-Holland, Amsterdam, 1991.

[17] E. Oja, "Beyond PCA: statistical expansions by nonlinear neural networks", in M. Marinaro and P.G. Morasso, editors, *ICANN '94. Proceedings of the International Conference on Artificial Neural Networks*, vol. 2, pp. 1049–1054, Sorrento, Italy, 1994. Springer-Verlag, Berlin, Germany.

[18] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning", *Neural Networks*, vol. 7, pp. 113–27, 1994.

[19] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks", *Neural Networks*, vol. 8, pp. 549–62, 1995.

[20] H.A. Malki and A. Moghaddamjoo, "Using the Karhunen-Loève transformation in the backpropagation training algorithm", *IEEE Transactions on Neural Networks*, vol. 1, pp. 162–165, 1991.

[21] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, and D.L. Alkon, "Accelerating the convergence of the back-propagation method", *Biological Cybernetics*, vol. 59, pp. 257–263, 1988.

[22] R. Battiti, "First- and second-order methods for learning:Between steepest descent and Newton's metod", *Neural Computation*, vol. 4, pp. 141–166, 1992.

[23] M. Plutowski and M. White, "Selecting concise training sets from clean data", *IEEE Transaction on Neural Networks*, vol. 4, pp. 305–318, 1993.

[24] M.D. Garris and R.A. Wilkinson, *NIST Special Database3 Handwritten Segmented Characters*, National Institute of Standard and Technology, Gaithesburg, MD , USA, 1992.

[25] Theo Pavlidis, *Algorithms for Graphics and Image Processing*, Springer-Verlag, 1982.