Biol. Cybern. 59, 291–294 (1988)

Biological
Cybernetics

Auto-Association by Multilayer Perceptrons and Singular Value Decomposition

H. Bourlard and Y. Kamp

Philips Research Laboratory, Avenue Van Becelaere 2, Box 8, B-1170 Brussels, Belgium

Abstract. The multilayer perceptron, when working in auto-association mode, is sometimes considered as an interesting candidate to perform data compression or dimensionality reduction of the feature space in information processing applications. The present paper shows that, for auto-association, the nonlinearities of the hidden units are useless and that the optimal parameter values can be derived directly by purely linear techniques relying on singular value decomposition and low rank matrix approximation, similar in spirit to the well-known Karhunen-Loève transform. This approach appears thus as an efficient alternative to the general error back-propagation algorithm commonly used for training multilayer perceptrons. Moreover, it also gives a clear interpretation of the rôle of the different parameters.

1 Introduction

Multilayer perceptrons (MLP) form a class of neural networks in which the nonlinear computing elements are arranged in a feed-forward layered structure. Since their emergence, they have been quite successfully used for a wide variety of tasks such as information processing, coding and pattern recognition, see e.g. Rumelhart and McClelland (1986) or Lippmann (1987). One of these applications consists in using the MLP to reduce the dimension of the feature space, an idea suggested by Rumelhart and McClelland (1986) and actually implemented by Elman and Zipser (1988) and Harrison (1987) in speech recognition and by Cottrell et al. (1988) for image compression. For this particular mode of operation, known as auto-association, identity mapping or encoding, the target output pattern is identical to the input pattern and, as a consequence, the output layer does generally not contain any nonlinear function, at least for real valued inputs. One can, of course, still apply the usual error back-propagation (EBP) algorithm as described by Rumelhart et al. (1986) to obtain the optimal parameter values for this case. However, the purpose of this paper is to show that, for auto-association with linear output units, the optimal weight values can be derived by standard linear algebra, consisting essentially in singular value decomposition (SVD) and making thus the nonlinear functions at the hidden layer completely unnecessary. The advantages are then obvious: the solution is obtained explicitly in terms of the training data, whereas the EBP generally used for the training of MLP proceeds iteratively and may well miss the optimum since it relies on a gradient technique and thus can get trapped in local minima. The analysis presented below offers the additional benefit that the optimal parameters are given a meaningful interpretation in terms of reconstruction of the average value and covariance of the input vectors.

2 MLP for Auto-Association

Figure 1 represents a general MLP with one hidden layer. It consists in an input layer containing n_i units and two layers with computational units: the hidden layer having p units and the output layer with n_0 units. In the standard MLP, the output of each hidden and output unit is determined by forming a weighted sum of the unit values in the preceding layer and then passing this result through a sigmoidal function, e.g.

$$F(x)=1/(1+e^{-x}). \tag{1}$$

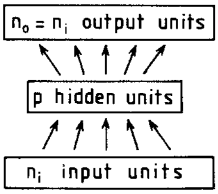

Fig. 1. MLP with one hidden layer for auto-association

When using this type of network to achieve dimension reduction by auto-association, it is desired that the input units communicate their values to the output units through a hidden layer acting as a limited capacity bottleneck which must optimally encode the input vectors. Thus, for this particular application, $n_i = n_0 = n$ and $p < n$. When entering an n-dimensional real input vector x_k $(k = 1, 2, \ldots, N)$, the output values of the hidden units form a p-vector given by

$$h_k = F(W_1 x_k + w_1), \quad k = 1, 2, \ldots, N, \tag{2}$$

where W_1 is the (input-to-hidden) $p \times n$ weight matrix and w_1 is a p-vector of biases. The nonlinear function F is operated componentwise and generates high order moments of each input vector. For most applications of MLP, as e.g. classification, the values in the output layer are obtained in a similar way and the rôle here of the nonlinear function F is to simulate the logical decision of the perceptron (Minsky and Papert 1969) which forces the output to be essentially binary (0 or 1 e.g.). For the auto-association however, the situation is different since the output values should approximate as closely as possible the inputs. Consequently, in the case of real valued inputs, the non-linearity at the output must be removed and the output values form an n-vector given by

$$y_k = W_2 h_k + w_2 \quad (k = 1, 2, \ldots, N), \tag{3}$$

where W_2 is the (hidden-to-output) $n \times p$ weight matrix and w_2 is an n-vector of biases. The problem is to find optimal weight matrices W_1, W_2 and bias vectors w_1, w_2 minimizing the mean-square error $J = \sum_{k=1}^{N} \|x_k - y_k\|^2$, which corresponds to the classical optimization criterion used for MLP training.

Let $X = [x_1, x_2, \ldots, x_N]$ be the $n \times N$ real matrix formed by the N input vectors of the training set and let $H = [h_1, h_2, \ldots, h_N]$ and $Y = [y_1, y_2, \ldots, y_N]$ be the $p \times N$ and $n \times N$ matrices formed by the corresponding vectors of the hidden and output units respectively. In view of (2), (3) the output matrix Y of the auto-associative MLP is obtained from the input matrix X as the result of the following sequence of operations illustrated by Fig. 2:

$$B = W_1 X + w_1 u^t, \tag{4}$$

$$H = F(B), \tag{5}$$

$$Y = W_2 H + w_2 u^t, \tag{6}$$

where B is a $p \times N$ real matrix and u is an N-vector of ones. With these notations, the squared error norm J can be rewritten as

$$J = \|X - Y\|^2, \tag{7}$$

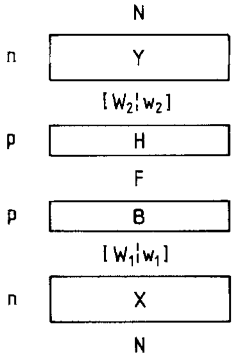

Fig. 2. Sequence of operations in the auto-associative MLP

where $\|\cdot\|$ now denotes the Euclidean matrix-norm (or Frobenius norm). The training problem is to minimize J with respect to the parameter set W_1, W_2, w_1, w_2.

3 Explicit Solution for the MLP Training

Using (6) the squared error norm can be rewritten as

$$J = \|X - W_2 H - w_2 u^t\|^2 \tag{8}$$

and, in view of $\|A\|^2 = \mathrm{tr}(AA^t)$, one easily verifies that minimization of J with respect to w_2 yields

$$\hat{w}_2 = \frac{1}{N}(X - W_2 H)u. \tag{9}$$

Substituting (9) in (8) we obtain for the squared error norm:

$$J = \|X' - W_2 H'\|^2, \tag{10}$$

where $X' = X(I - uu^t/N)$ and $H' = H(I - uu^t/N)$. In view of the fact that W_2 normally has rank $p < n$, expression (10) shows that the product $W_2 H'$ minimizing J is the best rank p approximation of X' in Euclidean norm. This is a standard problem and can be solved as follows. Consider the SVD of X' (Golub and Van Loan 1983; Stewart 1973):

$$X' = U_n \Sigma_n V_n^t, \tag{11}$$

where $U_n(V_n)$ is an $n \times n$ $(N \times n)$ matrix formed by the normalized eigenvectors of $X'X'^t(X'^tX')$ associated with the eigenvalues $\lambda_1 \geqq \lambda_2 \geqq \ldots \geqq \lambda_n$ and where $\Sigma_n = \mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_n]$ is a diagonal matrix with $\sigma_i = \sqrt{\lambda_i}$. For simplicity we will assume that X has full row rank $(\sigma_n > 0)$. It is known (Golub 1968; Stewart 1973) that the best rank p approximation of X' is given by

$$\hat{W}_2 \hat{H}' = U_p \Sigma_p V_p^t \tag{12}$$

with $\Sigma_p = \mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_p]$ and where $U_p(V_p)$ is formed by the first p columns in $U_n(V_n)$. Consequently

$$\hat{W}_2 = U_p T^{-1}, \quad \hat{H}' = T \Sigma_p V_p^t, \tag{13}$$

where T is an arbitrary non singular $p \times p$ matrix which will play an important rôle as scaling matrix in the sequel.

Let us make here a short pause to comment on the results derived so far and to point out a few interesting properties of the optimally trained auto-associative MLP.

Let μ_X denote the average of the training input vectors $x_1, x_2, \ldots, x_N$ i.e. $\mu_X = \frac{1}{N} Xu$ and let similarly $\mu_Y = \frac{1}{N} Yu$ be the average of the MLP output vectors. Taking (6) and (9) into account, it follows that the optimal bias vector $\hat{w}_2$ insures

$$\mu_Y = \mu_X \tag{14}$$

or, in other words, that the average input and output vectors are equal. Observe also that, in the very special case where all training vectors are identical, i.e. $X = \mu_X u^t$, this vector is exactly reproduced at the output ($Y = \mu_Y u^t$) since then $X' = 0$ and hence $\hat{W}_2'\hat{H}' = 0$ by (12).

If $\mu_H = \frac{1}{N} Hu$ denotes the average of the vectors at the output of the hidden units, then the definitions of X' and H' can be rewritten as $X' = X - \mu_X u^t$ and $H' = H - \mu_H u^t$ which means that they represent respectively the input and hidden unit vectors after subtraction of their average value. Consequently, the computational effect of the bias vector $\hat{w}_2$ is thus to reduce the training problem (10) to zero-average patterns.

Finally, one can show that the covariance of the output vectors $y_1, y_2, \ldots, y_N$ is the best rank p approximation of the covariance of the input vectors $x_1, x_2, \ldots, x_N$ and, in this sense, the auto-associative MLP is nothing but an indirect way of performing data compression by a Karhunen-Loève transform on zero-average data (Ahmed and Rao 1975). Indeed, owing to (11),

$$C_X = X'X''^t = U_n \Sigma_n^2 U_n^t . \tag{15}$$

On the other hand, the output covariance matrix defined as $C_Y = (Y - \mu_Y u^t)(Y^t - u\mu_Y^t)$ can, in view of (14), (6), (12) and orthogonality properties, be rewritten as

$$C_Y = U_p \Sigma_p^2 U_p^t \tag{16}$$

and comparison of (16) with (15) terminates the proof.

It is a remarkable fact that the optimal expressions in (9) and (13), as well as the preceding properties, have been obtained completely independently of the way in which H' is produced by the MLP and, more specifically, independently of the particular nonlinear function used at the output of the hidden units. In the sequel, we will first consider the case where this nonlinear function is absent which implies $H = B$. Next, we will show that this optimal situation can be approximated as closely as required even when a sigmoidal function is present at the output of the hidden units, as it is usually the case in MLP.

3.1 *No Nonlinear Function at the Hidden Units*

Since $B = H$, we have to prove that $\hat{H}'$ as prescribed by (13) can be generated in accordance with (4) by an appropriate choice of W_1 and w_1. Multiplying both sides of (4) by $(I - uu^T/N)$ we have thus to solve the following equation for W_1 and w_1

$$T\Sigma_p V_p^t = W_1 X' + w_1 u^t (I - uu^t/N) . \tag{17}$$

In view of $u^t u = N$, the second term on the right-hand side vanishes, showing that w_1 is arbitrary. Next, taking (11) into account, the left-hand side can be rewritten as $TU_p^t X'$ and (17) becomes then $TU_p^t X' = W_1 X'$, whence

$$\hat{W}_1 = TU_p^t . \tag{18}$$

Finally, to find the optimal value of the bias vector w_2, it is sufficient to eliminate $H = B$ from (9), via (4) and to incorporate results (13) and (18). One finds

$$\hat{w}_2 = (I - U_p U_p^t)\mu_X - U_p T^{-1} w_1 . \tag{19}$$

Thus, for arbitrary w_1, vector $\hat{w}_2$ should be adjusted according to (19) which, as we have seen before, insures $\mu_X = \mu_Y$. In summary, after SVD of X', (13), (18), and (19) give the optimal solutions for W_1, W_2, w_1, and w_2 of the "linear" MLP.

3.2 *The Hidden Units Contain a Nonlinear Function*

Let us now consider the case where a nonlinear function F is present at the output of the hidden units. We will not need strong assumptions about the particular form of this function except that, for small values of its argument, it can be approximated as closely as desired by the linear part of its power series expansion, i.e.

$$F(x) \sim \alpha_0 + \alpha_1 x \quad \text{for } x \text{ small} \tag{20}$$

with nonzero α_1. For the asymmetric sigmoid, $F(x) = 1/(1 + e^{-x})$, this gives $\alpha_0 = 1/2$ and $\alpha_1 = 1/4$; whereas for the symmetrical sigmoid, $F(x) = (1 - e^{-x})/(1 + e^{-x})$, one has $\alpha_0 = 0$, $\alpha_1 = 1/2$.

We will now show that, within minor modifications, the optimal values obtained in Sect. 3.1 still produce the expression for $\hat{H}'$ required by (13). If we take

$$\hat{W}_1 = \alpha_1^{-1} TU_p^t \tag{21}$$

we obtain, by (4)

$$\hat{B} = \alpha_1^{-1} TU_p^t X + w_1 u^t . \tag{22}$$

Obviously, if we want to use approximation (20), then B should be made small by acting on w_1 and on the arbitrary scaling matrix T. This leaves still some freedom on $\hat{w}_1$ which could be chosen equal to zero e.g. Another interesting possibility is to force $\mu_B = \frac{1}{N} Bu$, the average vector of matrix B defined in (4), to be zero by selecting

$$\hat{w}_1 = -\alpha_1^{-1} T U_p^t \mu_X . \tag{23}$$

In both cases, T should be sufficiently small but nonsingular. With $\hat{w}_1$ as given in (23), we finally obtain

$$\hat{B} = \alpha_1^{-1} T U_p^t X' = \alpha_1^{-1} T \Sigma_p V_p^t \tag{24}$$

and (5) yields $\hat{H} = \alpha_0 u u^t + \alpha_1 \hat{B}$, whence

$$\hat{H} = \alpha_0 u u^t + T \Sigma_p V_p^t . \tag{25}$$

Since $\hat{H}'$ has been defined by $\hat{H}' = H(I - uu^t/N)$, this gives, as desired, $\hat{H}' = T\Sigma_p V_p^t$. As for the optimal bias w_2, it can easily be computed from (9), (13), and (25) as

$$\hat{w}_2 = \mu_X - \alpha_0 U_p T^{-1} u . \tag{26}$$

Thus, in the case of a sigmoidal function at the hidden units, the optimal parameters of the MLP are given via the SVD of X' by (13), (21), (23), and (26).

It is not difficult to see that essentially the same approach can be used in the case of multiple hidden layers. The key operation remains the SVD of X' and its rank p approximation where p is now given by the last hidden layer. The freedom in the choice of the weight matrices and bias vectors becomes then even wider.

Finally, when the units on the output layer contain nonlinear functions, then of course, the approach presented above breaks down. However, even in this case, some interesting results can still be derived by analytical ways and are shown to be closely connected with low rank realizations of prescribed sign matrices (Delsarte and Kamp 1988).

4 Experiments

The training database for the experiments is composed of 60 vectors in $\mathbf{R}^{16}$ (hence X is a 16×60 real matrix). These are cepstral vectors obtained from 10-ms frames of speech signal and correspond to the mean vectors associated with the states of phonemic hidden Markov models (Bourlard et al. 1985). In order to confirm the theoretical results, we determined by the SVD of X' and Eqs. (13), (21), (23), and (26), the optimal weight matrices W_1, W_2 and biases w_1, w_2 for a rank 5 approximation (corresponding to 5 hidden units) and used these values as initialization of the EBP training algorithm. In that case, the EBP was unable to improve the parameters by reducing the error given in (7). Moreover, when starting the EBP training algorithm several times with random weights, it always got stuck in local minima giving higher error values. This illustrates thus the fact that the linear approach is definitely preferable.

One could object that the MLP and the associated EBP algorithm allow on-line learning which is an important advantage when the number of training patterns becomes large. However, the SVD algorithm has also a sequential version (Bunch and Nielsen 1978) and the argument therefore does not apply.

References

Ahmed N, Rao KR (1975) Orthogonal transforms for digital signal processing. Springer, New York Berlin Heidelberg

Bourlard H, Kamp Y, Wellekens CJ (1985) Speaker dependent connected speech recognition via phonemic Markov models. Proc ICASSP, pp 1213–1216

Bunch JR, Nielsen CP (1978) Updating the singular value decomposition. Numer Math 31:111–129

Cottrell GW, Munro PW, Zipser D (1988) Image compression by back propagation: a demonstration of extensional programming. In: Sharkey NE (ed) Advances in cognitive science, vol 2. Abbex, Norwood, (NJ) (in press)

Delsarte P, Kamp Y (1988) Low rank matrices with a given sign pattern Philips Research Laboratory, Brussels SIAM J:(to be published)

Elman JL, Zipser D (1987) Learning the hidden structure of speech. J Acoust Soc Am 83:1615–1626

Golub GH (1968) Least squares, singular values and matrix approximations. Applikace Matematiky 13:44–51

Golub GH, Van Loan CF (1983) Matrix computations. North Oxford Academic, Oxford

Harrison TD (1987) A Connectionist framework for continuous speech recognition. Cambridge University Ph. D. dissertation

Lippmann RP (1987) An introduction to computing with neural nets. IEEE ASSP Magazine, pp 4–22

Minsky M, Papert S (1969) Perceptrons. MIT Press, Cambridge

Rumelhart DE, McClelland JL, and the PDP Research Group (1986) Parallel distributed processing. Exploration in the microstructure of cognition. vol 1–2. MIT Press, Cambridge

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClellan JL (eds) Parallel distributed processing. Exploration in the microstructure of cognition, vol 1. Foundations. MIT Press, Cambridge

Stewart GW (1973) Introduction to matrix computations. Academic Press, New York

Received: December 3, 1987

H. Bourlard
Philips Research Laboratory
2, Avenue Van Becelaere, Box 8
B-1170 Brussels
Belgium