IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Auto-Association by Multilayer Perceptrons and Singular Value Decomposition

Hervé Bourlard [1,2]

IDIAP–RR 00-16

July 2000

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

1 Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), Martigny, Switzerland
2 Swiss Federal Institute of Technology at Lausanne (EPFL), Switzerland

IDIAP Research Report 00-16

Auto-Association by Multilayer Perceptrons and Singular Value Decomposition

Hervé Bourlard

July 2000

Abstract. This report is an electronic reprint (with minor extensions and adaptations) of a paper entitled "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition", by H. Bourlard and Y. Kamp, and initially published in *Biological Cybernectics*, vol. 59, pp. 291-294, in 1988. Given regular reprint requests received by the authors, and the fact that the journal is not easily accessible, it was decided to make this electronic version available on the web.

The multilayer perceptron (MLP), 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 error back-propagation algorithm commonly used for training multilayer perceptrons. Moreover, it gives a clear interpretation of the role of the different parameters.

1 Introduction

When used for speech recognition applications (or other classification problems), the emphasis is often put on the use of MLPs as discriminant pattern classifiers. Although pattern classification plays a crucial role, it is only part of the vast speech recognition task. In spite of the spectacular progress made over the last decade, unrestricted speech recognition is still out of reach, and it is suspected that part of the difficulty lies in the use of inappropriate features for recognizing speech. *A priori* phonetic knowledge seems of little practical use in this respect. The elementary sounds composing speech can indeed be described by place and manner of articulation for instance, but it seems difficult to translate this knowledge to a precise characterization at the signal level. On the other hand, one can consider that the hidden units of an MLP develop an internal representation of the input signal which is the most appropriate for the classification task. From this point of view, the MLP performs some type of feature extraction which is given by the activity levels of the hidden units. This view of an MLP as a trainable feature extractor for speech processing was described in [1], was systematically investigated in[2], and was more generally the original perspective in some of the work of Rosenblatt and his students.

In most MLP architectures used for feature extraction, the number of units on the hidden layer is smaller than on the input layer. Consequently, the hidden layer acts as a narrow-band channel and thus performs some form of dimensionality reduction. Again, if the learning procedure of the MLP is successful, one could expect that this reduction extracts the most salient features in the signal. In view of this observation, the MLP can be considered as an attractive alternative for efficient speech coding and image compression as examined in [2] and [3].

Feature extraction and dimensionality reduction can be learned in many ways but the most efficient one is to use teaching signals which are identical to the input since this avoids explicit segmentation and labeling of the signal and thus allows unsupervised training of the MLP. For this particular mode of operation, known as auto-association or identity mapping, the output layer generally does not contain any nonlinear function (at least for real valued inputs) since the output target is identical to the input pattern.

Of course, there are other techniques by which data compression and feature extraction can be achieved. Most important among these is the Karhunen-Loève or principal components transform, which is a purely *linear* method, in contrast with the nonlinear operation mode of the MLP, due to the sigmoidal function at the hidden units. In spite of this opposition, it was already anticipated in [3] that the auto-associative MLP should somehow be related to more classical techniques, the more so that a linear version of it produced results which were compatible with the nonlinear version. At this point, however, the exact nature of this relationship was not fully understood.

The purpose of this paper is to show on a rigorous basis that an auto-associative MLP with linear output units is nothing but an indirect way of performing data compression by a Karhunen-Loève transform (at best). More precisely, it will be shown that the optimal weight values can be derived by standard linear algebra, consisting in singular value decomposition (SVD) thus making the nonlinear functions at the hidden layer unnecessary. The advantages are obvious: the solution is obtained explicitly in terms of the training data, whereas the EBP algorithm generally used for training MLPs proceeds iteratively and may well miss the optimum solution since it relies on a gradient technique and 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 patterns.

Figure 1: *MLP with one hidden layer for auto-association.*

2 MLP and Auto-Association

Consider an MLP with a single hidden layer as represented in Figure 1 where p is the number of hidden units. When using this type of network to achieve dimensionality 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

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

where W_1 is the (input-to-hidden) $p \times n$ weight matrix, w_1 is a p-vector of biases and the nonlinear (typically sigmoid) function F is operated component wise. For most applications of MLPs, e.g., for classification, the values in the output layer are obtained in a similar way. However, in the case of auto-association, the output values should approximate the inputs as closely as possible. 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

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

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 $E = \sum_{k=1}^{N} \parallel x_k - y_k \parallel^2$, which corresponds to the standard 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. Given (1) and (2), 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 Figure 2:

$$B = W_1\, X + w_1\, u^T \tag{3}$$

$$H = F(B)\ , \tag{4}$$

Figure 2: *Sequence of operations in the auto-associative MLP.*

$$Y = W_2 \, H + w_2 \, u^T \tag{5}$$

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

$$E = \parallel X - Y \parallel^2 \tag{6}$$

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

3 Explicit and Optimal Solution

Using (5) the squared error norm can be rewritten as

$$E = \parallel X - W_2 H - w_2 u^T \parallel^2 \tag{7}$$

and, in view of $\parallel A \parallel^2 = tr(AA^T)$, one easily verifies that minimization of E with respect to w_2 yields

$$\widehat{w}_2 = \frac{1}{N} (X - W_2 H) u \tag{8}$$

Substituting (8) in (7) one obtains for the squared error norm:

$$E = \parallel X' - W_2 H' \parallel^2 \tag{9}$$

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 (9) shows that the product $W_2 H'$ minimizing E 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' [4, 5, 6]:

$$X' = U_n \, \Sigma_n \, V_n^T \tag{10}$$

where $U_n(V_n)$ is an $n \times n$ $(N \times n)$ matrix formed by the normalized eigenvectors of $X' X'^T$ $(X'^T X')$ associated with the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and where $\Sigma_n =$ 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 [4, 6] that the best rank p approximation of X' is given by

$$\widehat{W}_2 \, \widehat{H}' = U_p \, \Sigma_p \, V_p^T \tag{11}$$

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

$$\widehat{W}_2 = U_p \, T^{-1} \, , \qquad \widehat{H}' = T \, \Sigma_p \, V_p^T \tag{12}$$

where T is an arbitrary non-singular $p \times p$ matrix which will subsequently play an important role as a scaling matrix.

Let us pause here 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., $\mu_X = \frac{1}{N} X \, u$ and let similarly $\mu_Y = \frac{1}{N} Y \, u$ be the average of the MLP output vectors. Taking (5) and (8) into account, it follows that the optimal bias vector $\widehat{w}_2$ insures

$$\mu_Y = \mu_X \tag{13}$$

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 $\widehat{W}_2' \widehat{H}' = 0$ by (11).

If $\mu_H = \frac{1}{N} H u$ 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 $\widehat{w}_2$ is thus to reduce the training problem (9) 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 [8]. Indeed, owing to (10),

$$C_X = X' \, X'^T = U_n \, \Sigma_n^2 \, U_n^T \tag{14}$$

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 (13), (5), (11) and orthogonality properties, be rewritten as

$$C_Y = U_p \, \Sigma_p^2 \, U_p^T \tag{15}$$

and comparison of (15) with (14) terminates the proof.

It is a remarkable fact that the optimal expressions in (8) and (12), 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 following section, 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 is usually the case in an MLP.

4 Linear Hidden Units

Since $B = H$, we have to prove that $\widehat{H}'$ as prescribed by (12) can be generated in accordance with equation (3) by an appropriate choice of W_1 and w_1. Multiplying both sides of (3) by $(I - uu^T/N)$ we have thus to solve the following equation for W_1 and w_1

$$T\,\Sigma_p\,V_p^T = W_1 X' + w_1 u^T\,(I - uu^T/N) \tag{16}$$

In view of $u^T u = N$, the second term on the right-hand side vanishes, showing that w_1 is arbitrary. Next, taking (10) into account, the left-hand side can be rewritten as $T\,U_p^T\,X'$ and (16) then becomes $T\,U_p^T X' = W_1 X'$, so that

$$\widehat{W}_1 = T\,U_p^T \tag{17}$$

Finally, to find the optimal value of the bias vector w_2, it is sufficient to eliminate $H = B$ from (8), via equation (3) and to incorporate results (12) and (17). One finds

$$\widehat{w}_2 = (I - U_p U_p^T)\,\mu_X - U_p T^{-1} w_1 \tag{18}$$

Thus, for arbitrary w_1, vector $\widehat{w}_2$ should be adjusted according to (18) which, as observed before, insures $\mu_X = \mu_Y$. In summary, after SVD of X', equations (12), (17) and (18) give the optimal solutions for W_1, W_2, w_1 and w_2 of the "linear" MLP.

5 Nonlinear Hidden Units

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.,

$$F(x) \sim \alpha_0 + \alpha_1 x \qquad \text{for } x \text{ small} \tag{19}$$

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 the previous sections still produce the expression for $\widehat{H}'$ required by (12). If we take

$$\widehat{W}_1 = \alpha_1^{-1}\,T\,U_p^T \tag{20}$$

we obtain by (3),

$$\widehat{B} = \alpha_1^{-1}\,T\,U_p^T\,X + w_1\,u^T \tag{21}$$

Obviously, if we want to use approximation (19), then B should be made small by acting on w_1 and on the arbitrary scaling matrix T. This leaves still some freedom on $\widehat{w}_1$ which could e.g., be chosen equal to zero. Another interesting possibility is to force $\mu_B = \frac{1}{N}\,Bu$, the average vector of matrix B defined in (3), to be zero by selecting

$$\widehat{w}_1 = -\alpha_1^{-1}\,T\,U_p^T\,\mu_X \tag{22}$$

In both cases, $\|\,T\,\|$ should be sufficiently small but nonsingular. With $\widehat{w}_1$ as given in (22), one finally obtains

$$\widehat{B} = \alpha_1^{-1}\,T\,U_p^T\,X' = \alpha_1^{-1}\,T\,\Sigma_p\,V_p^T \tag{23}$$

and equation (4) yields $\widehat{H} = \alpha_0 uu^T + \alpha_1 \widehat{B}$, leading to

$$\widehat{H} = \alpha_0\,uu^T + T\,\Sigma_p\,V_p^T \tag{24}$$

Since $\widehat{H}'$ has been defined by $\widehat{H}' = H(I - uu^T/N)$, this gives, as desired, $\widehat{H}' = T\,\Sigma_p\,V_p^T$. As for the optimal bias w_2, it can easily be computed from (8), (12) and (24) as

$$\widehat{w}_2 = \mu_X - \alpha_0\,U_p\,T^{-1}\,u \tag{25}$$

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 expressions (12), (20), (22) and (25).

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 [7].

6 Experiments and Discussions

A simple training database was composed of 60 vectors in R^{16} (hence X is a 16×60 real matrix). These were cepstral vectors obtained from 10-ms frames of speech signal and corresponded to the mean vectors associated with the states of phonemic hidden Markov models [9]. In order to confirm the theoretical results, we determined by the SVD of X' and equations (12), (20), (22) and (25), 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 mean square error (6). Moreover, when starting the EBP training algorithm several times with random weights, it always got stuck in local minima, giving higher error values. This illustrated that the linear approach was 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 also has a sequential version [10], so this argument does not apply. Similarly, while the MLP can be implemented on fast parallel hardware, similar mappings can be made for SVD. Perhaps the only hardware-oriented argument that may favor the MLP approach is that MLP training can be done with lower precision (e.g., 16 bits for weights and 8 bits for activation), while SVD requires more precision (typically 32-64 bit floating point is used).

It is also important to remember that the theoretical developments presented in this paper are only valid for the auto-associative MLP with linear outputs and linear or nonlinear hidden units, where the number of hidden units is smaller than the number of input (and output) units. In the case where the (bottleneck) hidden layer with $p < n$ hidden units is preceded and followed by at least one additional hidden layer with $q > n$ units, this network will perform a nonlinear expansion of the input space before doing SVD in that expanded space. In this case, an explicit solution by linear algebra is no longer possible. However, this kind of nonlinear preprocessing has been shown to lead to better classification performance on some speech recognition problems [11].

Some parts of the theory developed in this paper can also be used to improve our understanding of hetero-associative MLPs used for classification and their relationships with discriminant analysis (see, e.g., [12]). However, this will never enable us to find the optimal solution for all the weights of an MLP (as done here for auto-association) except in some very particular cases where all the hidden and output units have a linear transfer function. In this case, of course, more strict mathematical treatments about the absence of local minima, the presence of saddle points, learning properties and relationships with principal component analysis is possible (see, e.g., [13, 14]).

7 Conclusion

We have investigated here the possibility to use MLP for feature extraction, related to the front-end processing of a speech recognizer. In this case, MLPs working in "auto-association" mode are usually used to extract relevant features from rough data. Such a network was studied here and it was shown that EBP can be avoided by analytically determining the optimal parameters of the network. It was proved that the optimal solution of the MLP was strictly equivalent to the standard singular value decomposition (SVD) approach and that, in this case, the nonlinearity in the hidden units is theoretically of no help. It was shown that the network actually projects the input onto the subspace spanned by the first p principal components of the input, where p is the number of hidden units.

Although this conclusion sounds a bit pessimistic, this approach has some merits: while allowing a better understanding of neural network processing and its relationships with standard signal processing techniques, it also provides us with an efficient parallel implementation of the SVD algorithm which can be integrated easily in a general neural network framework. Such a framework, as noted earlier, can be based on low or moderate precision hardware.

References

[1] Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986a). Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing. Exploration of the Microstructure of Cognition. vol. 1: Foundations*, D.E.Rumelhart and J.L.McClelland (eds.), MIT Press.

[2] Elman, J. & Zipser, D. (1988), Learning the Hidden Structure of Speech, *Journal of the Acoustical Society of America*, vol. 83, pp. 615-626.

[3] Cottrell, G.W., Munro, P.W. & Zipser, D. (1988). Image Compression by Back-Propagation: A Demonstration of Extensional Programming, in *Advances in Cognitive Science* (vol. 2), N.E. Sharkey (ed.), Norwood, (NJ): Ablex.

[4] Golub, G.H. (1968). Least squares, singular values and matrix approximations. *Applikace Matematiky*, vol. 13, pp. 44-51.

[5] Golub, G.H. & Van Loan, C.F. (1983). *Matrix computations*, Oxford: North Oxford Academic.

[6] Stewart, G.W. (1973). *Introduction to Matrix Computations.* New York, Academic Press.

[7] Delsarte, P., Kamp, Y. (1988). Low Rank Matrices With a Given Sign Pattern, *SIAM Journal of Discrete Math.*, vol. 2, pp. 51-63, 1989.

[8] Ahmed, N. & Rao, K.R. (1975). *Orthogonal Transforms for Digital Signal Processing*, New York, Springer-Verlag.

[9] Bourlard, H., Kamp, Y., Ney, H. & Wellekens, C.J. (1985). Speaker-Dependent Connected Speech Recognition via Dynamic Programming and Statistical Methods, in *Speech and Speaker Recognition*, M.R.Schroeder (ed.), Karger.

[10] Bunch, J.R. & Nielsen, C.P. (1978). Updating the singular value decomposition. *Num. Math.*, vol. 31, pp. 111-129.

[11] Nakamura, M., Tamura, S. & Sagayama, S. (1991). Phoneme Recognition by Phoneme Filter Neural Networks, *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 85-88, Toronto.

[12] Webb, A.R. & Lowe, D. (1989). Adaptive Feed-Forward Layered Networks as Pattern Classifiers: A Theorem Illuminating Their Success in Discriminant Analysis, *Neural Networks.*

[13] Baldi, P. & Hornik, K. (1989). Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima, *Neural Networks*, vol. 2, pp. 53-58.

[14] Baldi, P. & Hornik, K. (1991). Back-Propagation and Unsupervised Learning in Linear Networks, *Back-Propagation: Theory, Architectures and Applications*, Y. Chauvin and D.E. Rumelhart (eds.), Lawrence Erlbaum, NJ.