ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

Dimensionality Reduction by Local Processing

Christian Wöhler, Ulrich Kreßel, Jürgen Schürmann

DaimlerChrysler Research and Technology
PO Box 2360, D – 89013 Ulm, Germany

Joachim K. Anlauf

Rheinische Friedrich-Wilhelms-Universität Bonn,
Institut für Informatik II, D – 53117 Bonn, Germany

Abstract. In this paper we describe a novel approach towards dimensionality reduction of patterns to be classified. It consists of local processing of the patterns as an alternative to the well-known global principal component analysis (PCA) algorithm. We use a feed-forward neural network architecture with spatial or spatio-temporal receptive field connections between the first two layers that yields a transformed feature vector of significantly reduced dimension. We suggest two techniques to adapt the weights of the receptive fields: a local PCA algorithm and training by online gradient descent. Our dimensionality reduction algorithm requires computational costs that are several times smaller compared to the classical PCA approach without loosing performance in the subsequent classification process. We apply the algorithm to the problem of handwritten digit recognition as well as to the recognition of pedestrians in image sequences.

1. Introduction

When applying sophisticated classification techniques like e. g. polynomial classifiers [6] or support vector machines (SVMs) [4], one is often confronted with difficulties concerning limited computational power and memory space when processing high-dimensional patterns, which means in this context, patterns with more than about several hundred input features. Consequently, a need is encountered to reduce the number of input dimensions such that the properties of the patterns which are necessary to successfully carry out the classification task are retained, while it is desired to discard information that is irrelevant with respect to the classification task.
A well-known and widely-used classical approach towards dimensionality reduction of input patterns for classification is the principal component analysis (PCA) algorithm (for an introduction, see e. g. [1, 6]). This algorithm minimizes the reconstruction error of an input pattern for the desired reduced

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

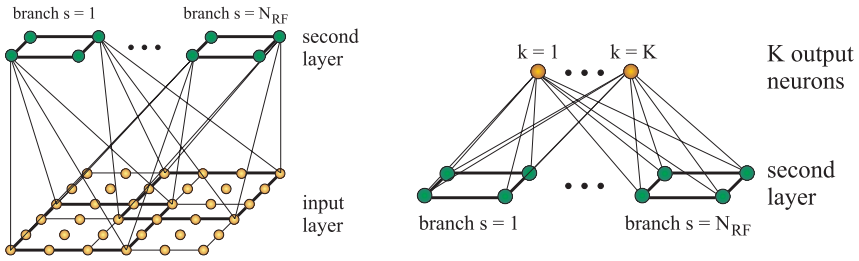

Figure 1: Left: Connections between the neurons of the input layer and the second layer. The original pattern of size 6×6 is divided into four receptive fields of size $R_x = R_y = 4$ with a distance of two pixels in each direction between the centres of two neighbouring receptive fields. For clarity, only the connections to the lower left and to the upper right receptive field are drawn. Right: Linear perceptron-like connections between the second neuron layer and the K output units for adaptation by gradient descent.

dimensionality. For high-dimensional input patterns, however, problems arise in performing the PCA algorithm as it then involves the diagonalization of large matrices, which may lead to strong difficulties that have to be overcome e. g. by sophisticated numerical or neural methods specially designed to yield only the first (largest) few eigenvalues and the corresponding eigenvectors of the matrix (see e. g. [1, 5]). Secondly, for real-time applications the processing time for global dimensionality reduction might be too costly.

2. Description of the dimensionality reduction algorithm

Our basic idea to overcome the difficulties encountered when classifying high-dimensional input patterns is local processing instead of the global PCA approach that aims at reconstructing the pattern as a whole. In this paper, we examine situations in which the position of a certain feature in the input pattern is characterized either by two indices (as it is the case e. g. for an intensity value B_{xy} in a greyscale image where x and y denote the pixel position) or by three indices (as e. g. in a temporal sequence of greyscale images where three indices are necessary to define the position of a voxel with intensity B_{xyt}). In the following, this "structural dimension" of the input pattern is denoted by d, where we have $d = 2$ for images and $d = 3$ for temporal image sequences. For the global PCA approach, the value of d is irrelevant as the order of the features is of no importance. In our approach, however, it must be taken into account as we regard local relations within limited spatial or spatio-temporal regions: the pattern is divided into overlapping d-dimensional blocks, later on called *receptive fields*, as shown in Figs. 1 and 2.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

From the neural network point of view, this idea corresponds to a network with a d-dimensional input layer and a second layer that consists of several branches. The number of branches is denoted by N_{RF}, the index of a branch by s with $1 \leq s \leq N_{RF}$. In this network, a layer 2 neuron of a certain branch is not connected to the complete input layer but only to a small d-dimensional region of it, sized $R_x \times R_y$ input neurons in the $d = 2$ case. This region is generally called the receptive field of the corresponding layer 2 neuron.
The *shared weights* principle is applied, which means that each layer 2 neuron of a certain branch is connected to its receptive field by the same configuration of weights. The configuration of the weights connected to a layer 2 neuron at position (i, j) thus only depends on the branch s in which the neuron is situated but not on its position (i, j); the weights are thus denoted by r^s_{mn}. The activation of this neuron is then given by

$$\xi^s_{ij} = g\left(\sum_{n=1}^{R_y}\sum_{m=1}^{R_x} r^s_{mn} B_{D_x(i-1)+m, D_y(j-1)+n} - \theta^s\right) \tag{1}$$

with B_{xy} as the input pattern, $g(x)$ as the activation function, and θ^s as the threshold value. The distance of the centres of two neighbouring receptive fields in the x and the y direction is given by D_x and D_y, respectively. The activation patterns $\{\xi^s_{ij}\}$ in the second network layer can now be regarded as N_{RF} filtered versions of the original input image with the N_{RF} filter kernels given by the receptive field weights $\{r^s_{mn}\}$.
In Fig. 1, the dimension of the input feature vector that may e. g. contain the grey values of an image is 36; in the second network layer, is has been reduced to $4N_{RF}$. The resulting activation patterns $\{\xi^s_{ij}\}$ in network layer 2 are further processed by a classifier. Our main task is now to adapt the weight configurations $\{r^s_{mn}\}$ of the receptive fields such that most of the information contained in the original input pattern necessary for the classification task to be performed is preserved by transformation (1). We will examine the following two approaches:

Local PCA of the receptive fields: The training set consisting of all d-dimensional overlapping receptive fields extracted from all training examples is decomposed into its N_{RF} most significant principal components, denoted by $\{P^s_{mn}\}$ with $1 \leq s \leq N_{RF}$. The zero component $\{P^0_{mn}\}$ is the corresponding average vector. The dimension $R_x \times R_y$ of such a local feature vector is generally significantly smaller – in Fig. 1, we have $R_x \times R_y = 16$ – than the dimension of the original feature vector, which is 36 in the example of Fig. 1. In the $d = 2$ example, we can now identify $r^s_{mn} = P^s_{mn}$ following (1) with a linear activation function $g(x) = x$. As for the global PCA method, the average $\{P^0_{mn}\}$ has to be subtracted from the input pattern before feeding it into the network, which leads to thresholds given by

$$\theta^s = \sum_{n=1}^{R_y}\sum_{m=1}^{R_x} r^s_{mn} P^0_{mn}. \tag{2}$$

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

This means that the vector $\left\{B_{D_x(i-1)+m,D_y(j-1)+n} - P^0_{mn}\right\}$ of pixel values in the receptive field connected to the layer 2 neuron at position (i,j) is expanded with respect to the first N_{RF} principal components calculated before. The corresponding overlaps are represented by the activations of the neurons in network layer 2. The reduced feature vector $\left\{\xi^s_{ij}\right\}$ can then be processed by any classifier.

Training of the receptive field weights by gradient descent: In a second approach, the neurons of network layer 2 are connected like a linear perceptron with the K output units representing the K classes the network is supposed to distinguish, as shown in Fig. 1. The resulting network is trained by a backpropagation-like online gradient descent method; during one training step, the corresponding weights are updated simultaneously with the receptive field weights $\{r^s_{mn}\}$ and the thresholds $\{\theta^s\}$. The training process of the resulting neural network is described in detail in [8] where the more general case of spatio-temporal receptive fields and thus $d = 3$ is examined, to which we will refer in section 3.2. We have chosen $g(x) = \tanh(x)$ as the activation function. Although transformation (1) then becomes nonlinear, it shows a linear behaviour in the limit of small weight values $\{r^s_{mn}\}$ such that the result of the local PCA described above is contained in the set of possible results of the gradient descent training process. The linear perceptron structure aims at transforming the original features into reduced features $\left\{\xi^s_{ij}\right\}$ in the second network layer that are linearly separable. Although in the cases examined later on this could never be achieved perfectly, the obtained distribution of patterns in the transformed feature space of reduced dimension turned out to be a good starting point when processing these patterns by more complex classifiers like polynomial classifiers or SVMs.

3. Application to classification tasks

3.1. Classification of handwritten digits

Our first application scenario is the recognition of handwritten digits, which is a classical character recognition problem. Our database contains 20000 examples altogether, 2000 of each of the $K = 10$ digit classes, with all examples normalized to a size of 16×16 greyscale pixels. We split the database by half into a training and a test set. Typical representatives of the training set and the decomposition into receptive fields are shown in Fig. 2. The classifier with which the transformed feature vectors of reduced dimensionality are processed is a pairwise linear polynomial classifier [4] that constructs $K(K-1)/2 = 45$ separating planes between all possible pairs of classes. Examples of receptive field weight configurations ("filter kernels") obtained by the two adaptation methods are shown in Fig. 2. The total complexity C_{tot} of one classification process, i. e., the number of floating-point multiplications and additions needed

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

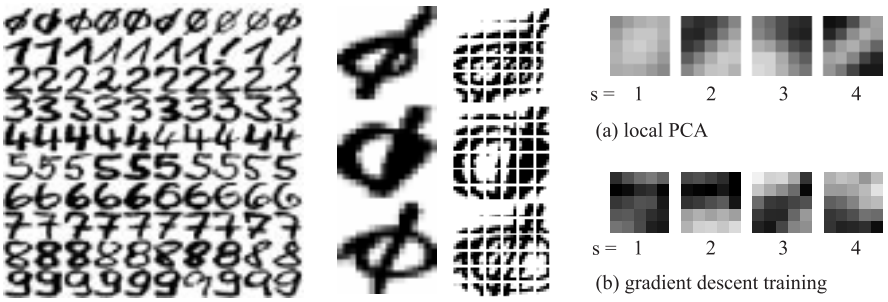

Figure 2: Left: Typical representatives of the handwritten digit database. Middle: Decomposition of digits into receptive fields of size $R_x \times R_y = 5 \times 5$ pixels with an offset of $D_x = D_y = 2$ pixels. Right: Receptive field weight configurations ("filter kernels") for the network configuration $R_x = R_y = 5$, $D_x = D_y = 2$, $N_{RF} = 4$ obtained by local PCA adaptation (a) and by gradient descent training (b). Both methods yield a low pass filter (a_1, b_1), a filter for diagonal edges (a_3, b_4), and a Laplacian-like filter (a_4, b_3). Furthermore, method (a) gives another diagonal edge filter (a_2), while method (b) produces a horizontal edge filter (b_2).

to transform and classify an input example, amounts to

$$C_{\text{tot}} = C_{\text{pre}} + C_{\text{class}} = D_{\text{red}} R_x R_y + \frac{1}{2} K(K-1)(D_{\text{red}} + 1) \tag{3}$$

with D_{red} as the dimension of the transformed feature vector. It turned out that the computational cost to calculate the tanh values can be neglected. The corresponding results are shown in Table 1 for several sizes and offsets of the receptive fields.
Table 1 shows that the computational cost of the preprocessing procedure can be reduced by about a factor of 4 or 5 compared to the global PCA approach at a comparable computational cost of the classification itself without an increase of the recognition error.

3.2. Recognition of pedestrians on image sequences

In this section, we will use our dimensionality reduction method to process spatio-temporal motion patterns of laterally walking pedestrians, i. e. more specifically, of their legs. To obtain the regions of interest containing the legs of a pedestrian, we employed the stereo detection and tracking algorithm discussed in detail in [2, 9]. The regions of interest are normalized in size (24×24 pixels) and ordered into image sequences consisting of 8 such normalized images, respectively. The structural dimension is thus $d = 3$, the actual dimension of a temporal image sequence amounts to $24 \times 24 \times 8 = 4608$. It is now very hard, if possible at all, to adapt a polynomial classifier or an SVM to feature vectors of such an immense dimension.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

R_x	D_x	N_{RF}	D_{red}	C_{pre}	C_{class}	C_{tot}	E_1 [%]	E_2 [%]
3	2	1	49	441	2250	2691	4.47	4.74
3	2	4	196	1764	8865	10629	3.90	1.98
5	2	2	72	1800	3285	5085	4.28	4.23
5	2	4	144	3600	6525	10125	3.85	2.35
5	3	4	64	1600	2925	4525	5.12	4.39
7	4	1	9	441	450	891	17.22	17.67
7	4	2	18	882	855	1737	7.38	10.36
7	4	4	36	1764	1665	3429	4.67	5.56
7	3	3	48	2352	2205	4557	3.96	4.57
7	3	4	64	3136	2925	6061	3.63	3.19
Global PCA			50	12800	2295	15095	3.74	
Pairwise lin. pol. class. on orig. digits						11565	3.67	

Table 1: Results of the classification of handwritten digits for several sizes and offsets of the receptive fields. E_1 denotes the error rate on the test set obtained with a transformation by local PCA, E_2 the one obtained by gradient-descent training of the receptive field weights. It is always $R_x = R_y$ and $D_x = D_y$.

A global PCA involves the diagonalization of a 4608×4608 matrix, which leads to strong numerical problems even if one is interested in the first few hundred principal components only, such that we did not follow this approach further. Instead, we applied our neural network concept that now becomes a time-delay neural network (TDNN) with spatio-temporal receptive fields of the type described in [8, 9]; the basic TDNN concept is described e. g. in [7]. The training set consists of 3926 pedestrian examples and 4426 non-pedestrian ("garbage") patterns. The test set contains 1000 pedestrian and 1200 garbage examples. As described in [9], a network with spatio-temporal receptive fields of size $R_x \times R_y \times R_t = 9 \times 9 \times 5$ pixels with offsets $D_x = D_y = 5$, $D_t = 1$, and $N_{RF} = 2$ branches turned out to yield by far the best performance. In the second network layer, each branch then contains 64 neurons, such that the reduced dimension amounts to $D_{\mathrm{red}} = 128$.
We adapted the receptive fields by the gradient descent method and trained a linear polynomial classifier as well as a second, third, fourth, and fifth-order polynomial SVM on the transformed 128-dimensional feature vectors. Due to the still quite large overlap between the two classes it was of no use adapting a linear SVM as most support vectors turned out to be slack variables. The SVMs of order three and higher, however, separate the two classes perfectly on the training set. Fig. 3a shows that the performance of the TDNN alone is still better than that of the linear polynomial classifier due to the temporal receptive fields between the second and the third TDNN layer (cf. [8]) offering additional degrees of freedom. The performance is then rising with increasing order of the SVM, converging for orders higher than about three.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

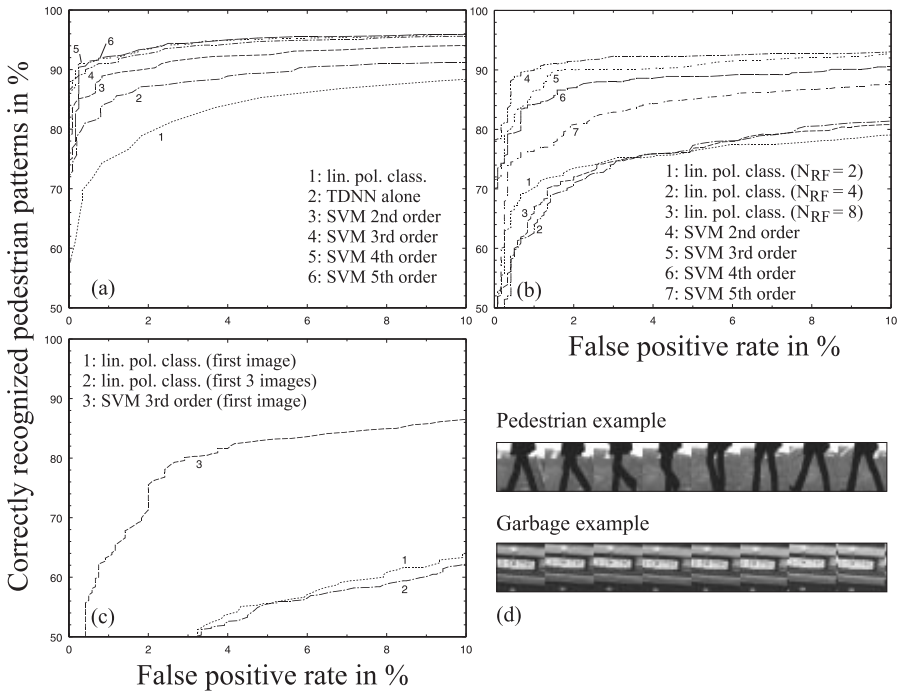

Figure 3: Recognition of pedestrians in image sequences: Rate of classification (ROC) curves on the test set for several classifiers applied to the local features produced by the gradient descent training method (a) and by the local PCA method (b). The results are compared to classifiers applied to the first few images of each sequence, respectively (c). Typical input patterns are shown in (d).

For the same network configuration the receptive fields were adapted using the local PCA algorithm with a subsequent classification stage. The method turned out to be more powerful when deriving the principal components only from the pedestrian patterns of the training set, not taking into account the garbage. The performance on the test set is somewhat lower but comparable to the one obtained by the gradient descent method (Fig. 3b). In this case, however, it decreases with rising order of the SVM, a phenomenon that may be regarded as an equivalent to the well-known "overtraining" of neural networks. Using more than the first two eigenvectors of the PCA (i. e. $N_{RF} > 2$) yields no better results at all.
In Fig. 3c, the performance of several classifiers on the first few original images of each sequence, respectively, is shown. It was not possible to adapt nonlinear classifiers to input vectors containing more data than only the first image for the reason of memory overflow. The performance is significantly lower than in Figs. 3a and 3b.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 237-244

4. Summary and Conclusion

In this paper we have described an algorithm for dimensionality reduction by local processing based on a neural network structure with spatial or spatio-temporal receptive fields. We propose two methods of adapting the weights of the receptive fields. The first one consists of performing a local principal component analysis (PCA); the resulting eigenvectors are then used as weight configurations for the receptive fields. The second method involves a backpropagation-like online gradient descent training of the weights of the receptive fields. Applying the algorithm to the problem of handwritten digit recognition as well as to the recognition of pedestrians in image sequences, we found out that compared to standard global PCA, the computational cost of dimensionality reduction could be reduced by about an order of magnitude with no loss in performance. The performance of the two described adaptation methods is comparable in both application scenarios, the gradient descent method often yielding a slightly higher recognition rate. It is, however, computationally quite expensive in the adaptation phase and convergence is not necessarily guaranteed, whereas the local PCA method has got a unique solution that is relatively easy to compute.
Dimensionality reduction by local processing is thus an algorithm that helps to cope with difficulties concerning limited computational power or memory space encountered when classifying patterns with very large input dimensions.

References

[1] K. Diamantaras, S. Y. Kung. Principal Component Neural Networks. Wiley-Interscience, New York, 1996.

[2] U. Franke, I. Kutzbach. Fast Stereo based Object Detection for Stop&Go Traffic. *IEEE International Conference on Intelligent Vehicles*, pages 339-344, Tokyo, 1996.

[3] J. A. Hertz, A. Krogh, R. G. Palmer. Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City, CA, 1991.

[4] U. Kressel. Pairwise Classification and Support Vector Machines. In: B. Schölkopf, C. Burges, A. Smola. Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, MA, 1998.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. Numerical Recipes in C. Cambridge University Press, Cambridge, 1992.

[6] J. Schürmann. Pattern Classification. Wiley-Interscience, New York, 1996.

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pages 328-339, 1989.

[8] C. Wöhler and J. K. Anlauf. A Time Delay Neural Network Algorithm for Estimating Image-pattern Shape and Motion. *Image and Vision Computing Journal*, in press.

[9] C. Wöhler, J. K. Anlauf, T. Pörtner, and U. Franke. A Time Delay Neural Network Algorithm for Real-Time Pedestrian Recognition. *IEEE International Conference on Intelligent Vehicles*, pages 247-252, Stuttgart, 1998.