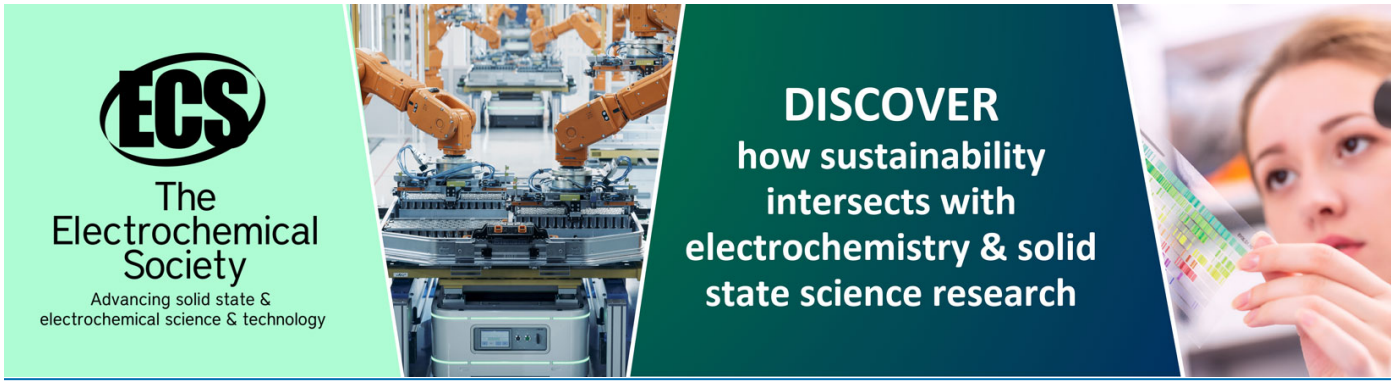
Reduction of the dimension of neural network models in problems of pattern recognition and forecasting



IOP Conf. Series: Journal of Physics: Conf. Series **929** (2017) 012038 doi:10.1088/1742-6596/929/1/012038

Reduction of the dimension of neural network models in problems of pattern recognition and forecasting

A D Nasertdinova[1], V V Bochkarev[1]

[1]Kazan Federal University, Kremlin Street 18, Kazan, Tatarstan, 420008, Russia

E-mail: voinova.anastasija@gmail.com

Abstract. Deep neural networks with a large number of parameters are a powerful tool for solving problems of pattern recognition, prediction and classification. Nevertheless, overfitting remains a serious problem in the use of such networks. A method of solving the problem of overfitting is proposed in this article. This method is based on reducing the number of independent parameters of a neural network model using the principal component analysis, and can be implemented using existing libraries of neural computing. The algorithm was tested on the problem of recognition of handwritten symbols from the MNIST database, as well as on the task of predicting time series (rows of the average monthly number of sunspots and series of the Lorentz system were used). It is shown that the application of the principal component analysis enables reducing the number of parameters of the neural network model when the results are good. The average error rate for the recognition of handwritten figures from the MNIST database was 1.12% (which is comparable to the results obtained using the "Deep training" methods), while the number of parameters of the neural network can be reduced to 130 times.

1. Introduction

Deep neural networks are widely used for solving the problems of image recognition, forecasting and classification. To describe complex dependencies, the network should include many neurons and connections. On the other hand, the amount of empirical data is always limited and using neural networks with too many parameters results in overfitting [1]. In the process of overfitting, the neural network gives good results on the examples used for training, and unsatisfactory results on new data. In recent years, a number of approaches to the problem of overfitting was proposed. The main idea is to reduce the number of parameters of a neural network. One of the most effective methods is Dropout [2]. The main idea is to randomly remove single neurons and their connections from the network during training. This prevents excessive network adaptation. Excluded neurons do not contribute to the training process at any stage of the back propagation algorithm. Therefore, dropping out at least one of the neurons is equal to training of a new neural network. This method allowed us to reduce the recognition error of handwritten numbers from the MNIST database to 0.79%. One of the drawbacks of the dropout method is that it increases training time. Dropout network training usually takes 2-3 times more time than the standard training of neural network of the same architecture.

In [3], a method is proposed to reduce the dimension of two-dimensional data for classification. This method is implemented using the so-called generalized autoencoder (GAE), which fixes the structure of the data space by minimizing the weighted distances between the restored vectors and the original ones,

for example, the method of k-nearest neighbours. It differs from the traditional auto-encoder in two aspects:

- GAE examines the compressed representation y_i for the vector x_i and builds a connection between the other data $\{x_j, x_k, ...\}$ by using y_i for reconstruction of each item in the set, not only x_i itself;
- GAE imposes a relational weight s_{ij} on the reconstruction error $\|x_j - x_i'\|$.

Therefore, the GAE fixes the structure of the data space by minimizing the weighted reconstruction error. The application of this method improved performance on the MNIST data set to 3.9%.

In this article, we propose a method of controlling overfitting by reducing the number of independent parameters of neural networks using the principal component analysis. The algorithm was tested on three sets of data: a database of images of MNIST handwritten symbols, time series of the average monthly number of sunspots and the series of the Lorentz system.

2. Use of the principal component analysis for neural networks training

The principal component analysis (PCA) is one of the main techniques for reducing the dimensionality of data and extraction of the necessary information from these data. It is assumed that the data compose a matrix (for example, they are series of measurements of several values). The PCA is based on he idea of transition to a new coordinate system in the original subspace of features where various components (columns of the matrix) will be uncorrelated. Mathematically, this is realized by means of the singular decomposition of the data matrix X, which can be described by the following formula [4]:

$$X = \sum_{r=1}^{p} \sigma_r u_r v_r^H \quad (\text{or } x_{ij} = \sum_{r=1}^{p} \sigma_r u_{ri} v_{rj}) \tag{1}$$

where u_r and v_r — orthonormal singular vectors, $\sigma_r \geq 0$ – are singular values (in fact, these are the root-mean-square deviations for new variables), and p is the matrix rank. The most informative components in such a coordinate system are those with the largest variance. Accordingly, the data matrix can be approximated in the best way (that is, with minimal loss of information) if we leave M terms in expression (1) corresponding to the largest singular values.

$$\tilde{X} = \sum_{r=1}^{M} \sigma_r u_r v_r^H \tag{2}$$

From now on, we assume that the components are ordered in descending singular values.

The main idea of the proposed method is to apply PCA to the weight matrices of the layers of the neural network. Using PCA, we obtain a network in which the weight matrices of the layers have a relatively small rank. Thus, we achieve regularization of the training of the neural network by significantly reducing the number of independent parameters of the neural network model.

Let's consider the matrix of network weights connecting the two layers consisting of N neurons, and assume that it can be shown as (2), where *M* is significantly smaller than *N*. Due to the fact that singular orthogonality and normalization conditions are imposed on the singular vectors u_r and v_r, the number of independent values will be considerably smaller than N^2. It is not difficult to calculate that the number of independent parameters for the weight matrix of the layer will be equal to:

$$2NM - M^2 \tag{3}$$

To train a neural network with weighted matrices of reduced rank, it is required to correct the scheme of the back propagation algorithm. Accordingly, new software libraries are needed to train the neural networks of this architecture. In this paper, two approaches are considered that allow us to achieve the desired structure of the weight matrices of network layers using the existing libraries of neural computing.

This problem can be solved using two methods:

- by adding a linear layer of smaller dimension between the layers of the network with nonlinear transfer functions;

- by transformation of matrixes of layer weight coefficients of already trained neural network using PCA, by removing components corresponding to small singular numbers. First of all, PCA is applied to the initial layers, after which it is required to train subsequent layers. Then, this procedure is performed for the next layer etc.

Let us consider the first method in more detail (Figure 1). There are two layers with nonlinear transfer functions in the source network (on the left). The layers are connected with a weight matrix ω. We insert an additional linear layer between the layers of the source network. In this case, the vector of output values of the first layer is successively multiplied by the matrix of weights of the linear layer $\omega^{(1)}$, and then - by the matrix of the weights of the second layer $\omega^{(1)}$. The matrix $\omega = \omega^{(2)}\omega^{(1)}$ obtained in such way will have a rank less than M. When using the criterion of a mean square error for the network training, the result will be completely analogous to the application of PCA.

The proposed methods were tested for the problem of recognition of handwritten numbers, as well as for the task of forecasting time series.

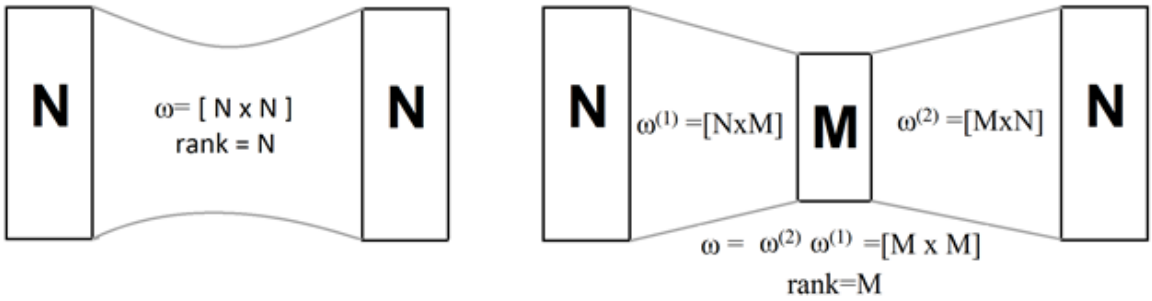


Figure 1. Structure of a neural network without an additional layer (on the left). Structure of a neural network with an additional linear layer with M neurons (on the right).

3. Recognition of handwritten number from the MNIST database

The MNIST database (available at http://yann.lecun.com/exdb/mnist) contains 60 000 black-and-white images of handwritten numbers [5]. The size of each image is 28 x 28 pixels. This database is widely used for comparative testing of training technologies and methods of pattern recognition. The data set does not require primary processing and formatting.

We trained a set of recognizers. Each recognizer checked whether a given number is shown on the successive image. The target value of the network output is 1 if the image has a specified number and is 0 if there is no specified number. The decision threshold was chosen in such a way that the probabilities of error of the first kind (it is decided that there is a number in the image when there is none) and errors of the second kind (it is decided that there is no number in the case when it exists) were equal. In this case, the quality of recognition can be quantified by a single number - the percentage of errors.

An unidirectional three-layer neural network of the form 784-800-800-1 is formed for the given task, which corresponds to 1268000 connections. We use the sigmoidal transfer function, the mean square error as the target function, the training algorithm R-prop (based on the gradient descent method) [6]. This algorithm does not require the use of one-dimensional search procedures and imposes insignificant memory requirements, works rather fast and is recommended for solving large-dimensional problems.

Let us consider the results of comparative testing taking the example of the recognizer for the number "2". The training lasted 500 epochs. The recognition error on the control set was 10.4%. This value is 90.4 times higher than the error on the training set, which indicates the presence of overfitting.

Similar calculations were performed for a neural network with an additional linear layer. To perform comparative testing, we create a neural network of the 784-800-M-800-1 form, where M is the number of neurons in the additional layer. The graph of the recognition error versus the number of neurons in the additional layer (solid lines) is shown in Figure 2. The graph shows the dashed straight lines which represent the forecast of the previous three-layer model for comparing the results. Good results are

obtained at M = 50. In this case, the probability of error on the test set is 1.13%. The number of connections between the hidden layers in accordance with the formula (3) decreases by 8.3 times.

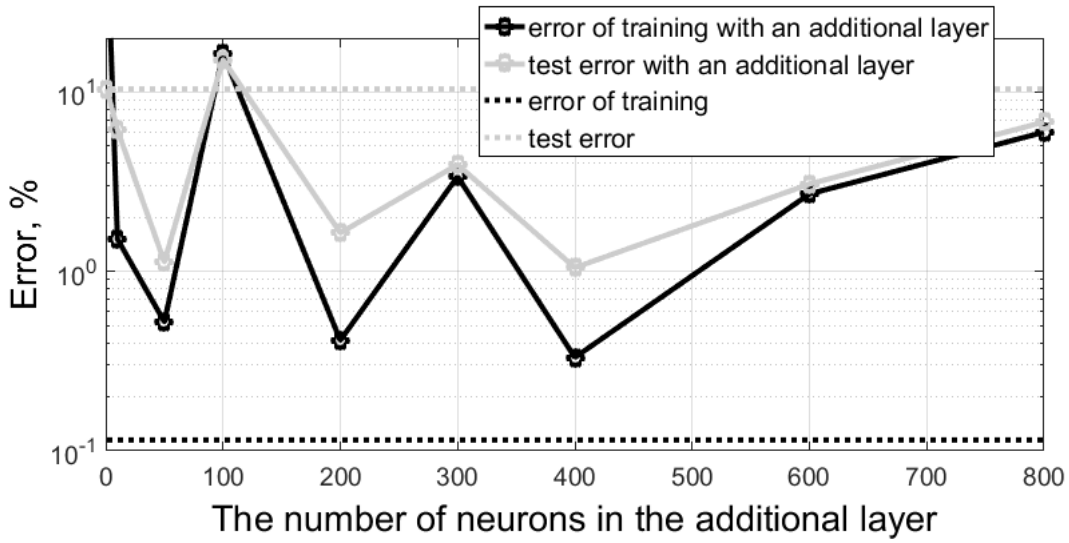


Figure 2. Probability of recognition errors of the number "2" using a neural network with an additional linear layer.

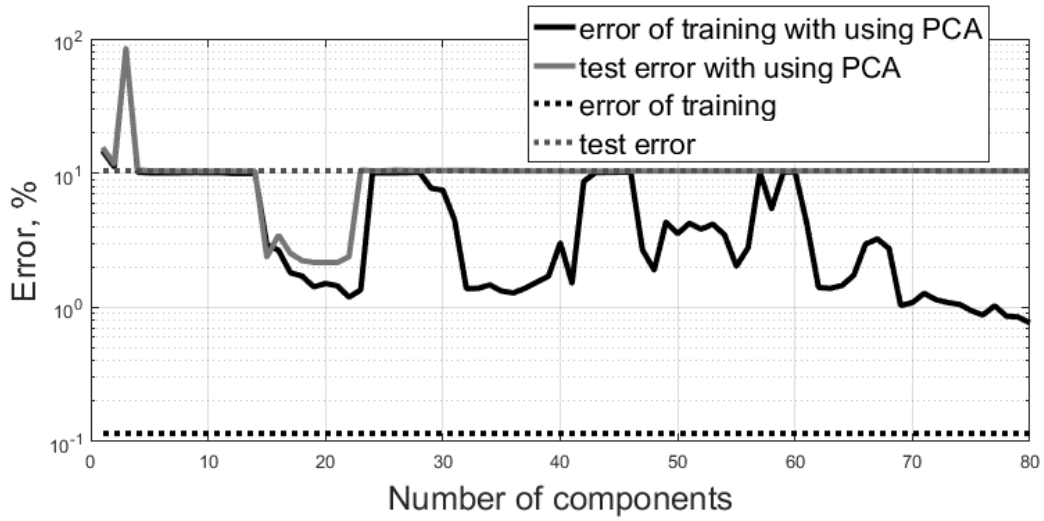


Figure 3. The probability of recognition errors of the number "2" by a three-layer neural network using the principal component analysis for adjusting the weight matrix of the hidden layer.

Also, the second method of reducing the number of independent parameters of the neural network was considered using the same example. The PCA was applied to the weighted matrix of the hidden layer (dimension - 800x800) trained by the three-layer neural network. The error probability values on the training and test set for the network with the matrix transformed by means of the PCA are shown in Fig. 3. It is interesting that by simply removing the redundant information from the network without conducting any additional training, the network results on the test set can be improved in some cases. The best result is obtained at M = 21. In this case, the probability of recognition error on the test set is 2.16%. The number of independent parameters in the weight matrix of the hidden layer can be reduced by 19.3 times.

Better results are obtained after additional training of subsequent layers. As for the given example, it is sufficient to optimize the weights of the output layer. After optimizing the weights, the frequency of recognition errors was 1.33%.

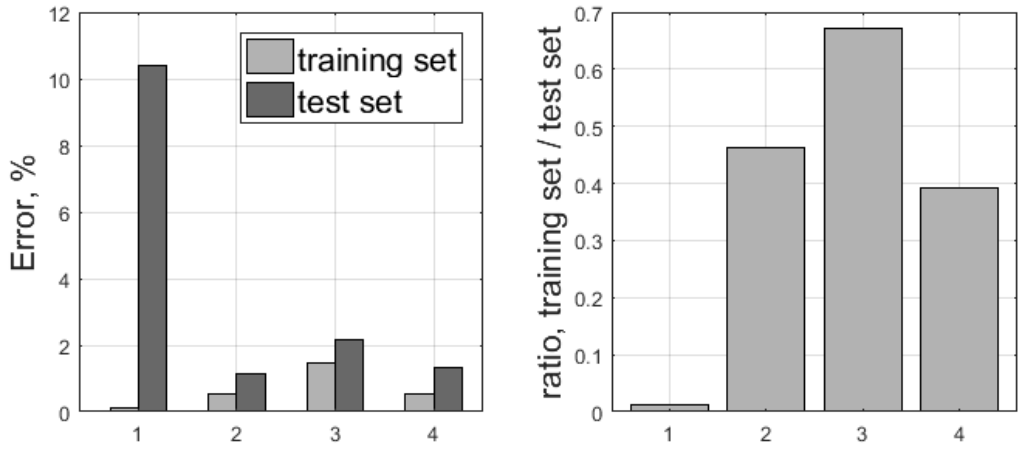


Figure 4. Comparison of recognition errors probabilities of the number "2" using different methods on the training and test set (on the left) The ratio of training error to test error (on the right). The applied method: 1 - three-layer neural network, 2 - network with an additional linear layer, 3 - correction of the weight matrix of the hidden layer using the PCA, 4 - correction of the weight matrix of the hidden layer using the PCA with further optimization of the weights of the output layer.

Comparison of the results of recognition of the number "2" by different methods is shown in Figure 2. It is necessary to pay special attention that the ratio of the error probabilities on the training and test sets in all cases of using the PCA is characterized by reasonable values in contrast to a simple three-layer network. Thus, the use of PCA allowed us to avoid overfitting.

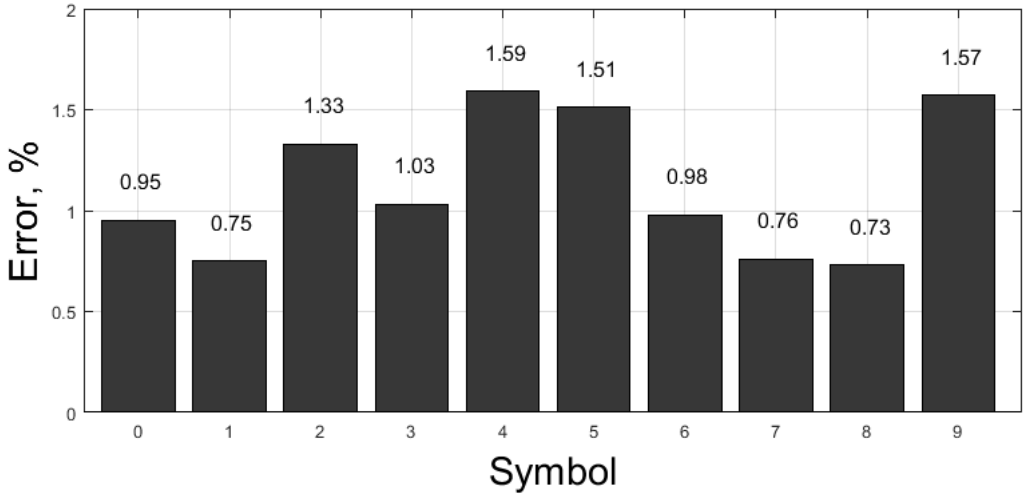


Figure 5. Percentage of recognition errors for different symbols.

The procedures described above were realised for each number from 0 to 9. The best results for each of the figures are shown in the figure 5. The best result was obtained for the figure 8, it was 0.73%. The worst result was obtained for the number 4 which was 1.59%. The average error value was 1.12%. The

greatest decrease in the number of independent parameters of the neural network is also observed for the figure 4 (by 133 times).

4. Conclusion

The article suggests a method that allows reduction of the number of independent parameters of the neural network model. Our approach is based on the use of PCA and can be implemented using existing software libraries of neural computing.

The proposed method was tested for solving the problem of recognition of handwritten numbers from the MNIST database. The average percentage of recognition errors was reduced to 1.12%, which is comparable to the results obtained using more powerful methods of in-depth training. At the same time, the number of independent parameters was reduced by 130 times compared to a conventional multilayer neural network. Good results were also obtained for the problem of time series forecasting. It should be noted that the proposed approach can be used together with the methods of in-depth training.

The research of the second author was supported by the Russian Government Program of Competitive Growth of Kazan Federal University.

References

[1] Haykin S 2008 *Neural Networks and Learning Machines* 3rd ed (New Jersey: Upper Saddle River - Pearson Education)

[2] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: A Simple Way to Prevent Neural Networks from Overfitting *J. of Machine Learning Research* **15** 1929-58

[3] Wang W, Huang Y, Wang Y and Wang L 2014 Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction *IEEE Conf. on CVPRW* **79** 496-503

[4] Gorban A and Kegl B 2007 *Principal Manifolds for Data Visualisation and Dimension Reduction,* ed Wunsch D and Zinovyev A (Springer, Berlin – Heidelberg – New York)

[5] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proceedings of the IEEE* **86(11)** 2278-2324

[6] Riedmiller M and Braun H 1993 *IEEE Int. Conf. on NN* vol 1 586-591