ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

E11-2000-269

G.A.Ososkov, A.V.Stadnik

NEURAL NETWORK APPLICATION FOR THE FACE RECOGNITION SYSTEMS

Presented at the «Modern Trends in Computational Physics»,
24–29 July 2000, Dubna, Russia

2000

1 Introduction

The problem of reliable face recognition is of importance and widely explored by various techniques (see, for example [1, 2, 3, 4]). The complexity of this problem leads to very sophisticated hierarchical algorithms. However for some simplier cases, as for frontal view face recognition systems, faster and less complex approaches should be developed. Neural networks (NN) and multilayer perceptrons (MLP), in particular, are very fast means for classifications of complex objects. However MLPs demand a prior procedure of training [5], which is usually realized by the error back propagation (EBP) algorithm and can appeare very time consuming. Besides due to tremendous volumes of data after a single face image digitalization they could not be used as a direct input for MLP. Therefore reports about MLP applications for the frontal face recognition [4] suppose considerable face data preprocessing.

In the given talk a new type of neural networks is proposed. We name them the metric NN. They are suitable for direct handling of digitized images and need much less time for training than ordinary multilayer perceptrons with EBP algorithm. Examples of efficient applications to recognizing of human faces and hand-written letters are given.

2 Problem formulation

Neural network is considered as a mean for the fast recognition of any of a substantial group of human faces. The only frontal views of face images are considered further, which are digitized (by a video-camera, for instance) and stored as 2D raster. It was found that in the majority of cases a raster with not less than 80x100 pixels of 8-bit grey level is efficient to distinguish individual features of a person. To obtain a reliable level of recognizability the NN must be trained before on a training sample of digitized face images of all persons to be recognized. After training NN must adequately recognize any of faces from the sample and undoubtly indicate any case of a "strange" face. Network must function in real circumstances when a person can slightly vary its pose, have a minor changes of the face expression, hair-dressing, make-ups, be unshaven etc.

Such the reliability and robustness requirements can be accomplished by including into the training sample more than one face image (up to 10) of the same person.

Next we will study how much the conventional MLP can satisfy such the requirements and what can be modified to fulfil them.

3 Multilayer perceptron and its training problem

A scheme of MLP is shown in Fig.1

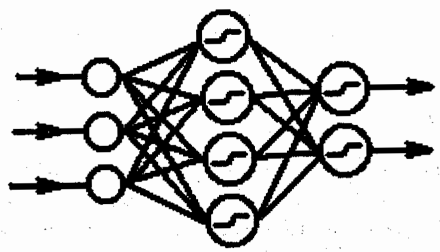


Figure 1: Scheme of a conventional feedforward three-layer neural network scheme.

A neuron is the basic element of any artificial neural network (ANN). It works as:

$$h_j = \sum_k w_{jk}^{(1)} x_k, \tag{1}$$

where x_k are input signals, $w_{jk}^{(i)}$ are the weights of synaptic connections between neurons of i and $i+1$ layers. The output signal of the j-th neuron is $y_j = g(h_j)$, where the activation function $g(x)$ is either a threshold function, or a sigmoid type function, like

$$g(x) = \frac{1}{1+e^{-x}}.$$

In the case of a threshold function and, say two classes, the perceptron attributs the vector x_i to the first class, if $\sum_j w_{ij}^{(2)} h_j \geq 0$, or to the second class, otherwise.

Such a scheme admits the following geometric interpretation (see Fig.2a): The hyperplane given by equation $\sum_k w_{ij}^{(2)} h_j = 0$, divides the space on two halfspaces corresponding to classes in question.

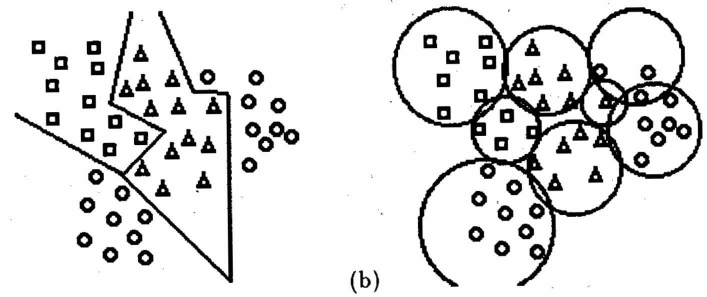
If the number of classes is more than two, then several dividing hyperplanes will be defined during the training process.

For the input vector of the classified features $\bar{X}_i$ MLP brings in correspondence an output vector $\bar{Y}_i$. The transformation $\bar{X}_i \Rightarrow \bar{Y}_i$ is completely described by the matrix of synaptic weights to be found as a solution of any concrete problem.

Let us have some training sample as a set of pairs of vectors $\{\{\bar{X}_i^{(m)}\}, \{\bar{Z}_i^{(m)}\}\}$. The MLP training is accomplished by minimization of so-called energy function

$$E = \sum_m \sum_i (Y_i^{(m)} - Z_i^{(m)})^2 \Rightarrow min$$

by weights $w_{jk}^{(i)}$ as minimization parameters. Such the EBP method is usually realized by the gradient descent method.



(a) (b)

Figure 2: a)An example of classifying into three overlapped classes by an ordinary MLP. b)Classifying of different feature vectors from the previous example into three classes by the metric ANN.

Arising hindrances:

- "Damnation of dimention" - too high number of weights, i.e. minimization parameters (proportional to number of neurons in power of layers number).
- Arbitrariness in choosing of the hidden layer neurons.
- Difficulties with selectiing of the training sample to be long enough to guarantee the correct classification.
- Unlimited time of training procedure and a lack of its convergence guaranty.

4 Proposed ANN design

We propose to replace $\sum_j w_{ij} x_j$ in (1) by

$$\sum_j (x_j - w_{ij})^2.$$

That means we introduce a new L_2^2-type metric in the feature space, i.e. we now classify different classes by covering them with a set of hyperspheres instead of hyperplanes as before (see Fig.2b).

Thus a metric neural net consists of four layers on neurons with three weight layers between them. The first input layer doest not perform any calculations, its neurons just serve to split input signals between neurons of the second layer. The second layer deals with coarsing and averaging of the input signals to transfer them to a feature space. Neurons of the third layer function similarly to Kohonen's neurons (winner-takes-all [5]), i.e. the only one neuron outputs a signal in the case of sucessful identification, when its input signal exceeds a threshold while all others gives zero outputs.

The goal of training is the determination of weight vectors of the second layer and of activation thresholds of the third layer. Neurons of the fourth layer and the third layer of weights make a decision what of faces is closest to the vector of the second weight layer, which is, in fact, an average of features of presented face modifications for the given person.

4.1 Training algorithm of the metric ANN

1. Set the initial number of neurons in the hidden layer as the number of expected classes. Equate the vector of weights of each neuron of the hidden layer with an accidentally choosen element of the class, to which this neuron is corresponded.

2. Choose randomly an input element of a pair from the training sample and check if it belongs to one of areas attributed to the class defined by the output element of this pair.

3. If yes, mark it and then go to step 2.

4. If no, then try to extend the hypersphere radius in order to be able to include this neuron in the area attributed to the given class.

5. Check if some elements of other classes belong to that extended hypersphere.

6. If no, then go to step 2.

7. If yes, than add one more neuron to the hidden layer. Equate the initial weight vector of this neuron with the running input vector. Set up the initial threshold of this new neuron of the hidden layer to a minimal value, which must guarantee that the only given input vector belongs to the new hypersphere, but no one of other elements can appeare inside of it.

8. Repeat from step 2, unless the training sample will be exhaust.

<u>What that gives:</u>

- Dynamic change of the number of hidden neurons during the training procedure.
- High speed and finitness of the trainig process.
- Potential to deal with very large number of neurons.

5 Applications and results

The metric NN described in the previous section was implemented as a C++ program. We started, first, with a simpler task to recognize letters written by computer mouse on the monitor screen. The training sample was also simplified to one of computer fonts. Results (some of them are presented in Fig.3 and 4) were so promising that stimulated our work with the frontal face recognition project.

The program was extended by modules for converting an image into 2D raster format with 80x100 pixels and then to 8000 input neurons. Besides the special means were developed to avoid the full training cycle with every adding a new face to the face bank and fulfil its local changes instead.

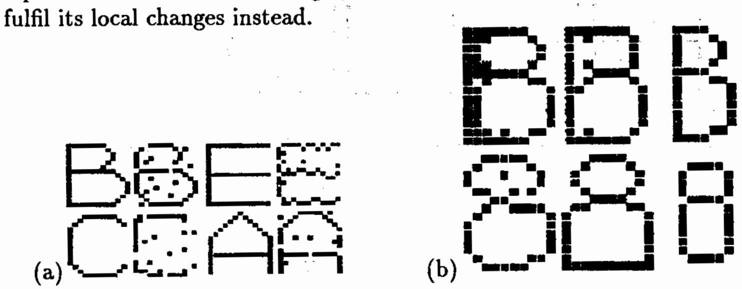


Figure 3: Two examples of successfully recognized characters. a)Set of letters. b) Second set of various writtings of two similar characters B and 8.

Two source of face images are used: the face base of the Cambrige University [6] with 10 various frontal face poses of 40 persons and, to be close to reality, face images taken by a digitized video-camera coupled with Pentium-III PC. An example of a training sample of three persons of various age and sex taken by video is shown in Fig.5. Then these images have been considerably noised or specially distorted. Results of correctly recognized images are shown in Fig.6. It was observed that the robustness of the proposed NN to noise and deliberated image distortions directly depends on the number of various modifications of the each face presented in the training sample. The Cambridge face bank was used then for the network training and testing. The recognition efficieny over 95% was achieved. To impove it and also computer memory requiremnts algorithms for the fast feature extracion are to be incorporated into the system.

6 Conclusion: advantages of the proposed approach.

The system was developed for the real time frontal view human face recognition on the low cost basis of Pentium II-like PC coupled with a digitized video-camera. The system implements a new type of the metric neural network. The tolerance ability to various image distortions was tested.

- High speed and finitness of the trainig process.
- Easyness to estimate the classification quality. It's indicated by the number of neurons in hidden layer.

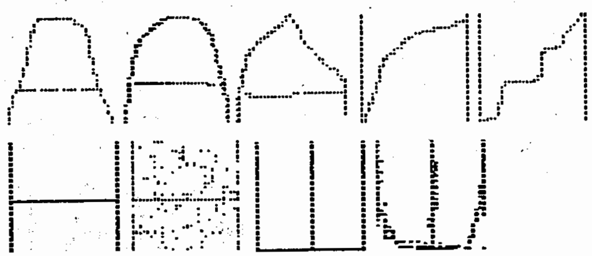


Figure 4: Imitation of handwrite characters with noise. Metric NN recognizes all characters correctly.

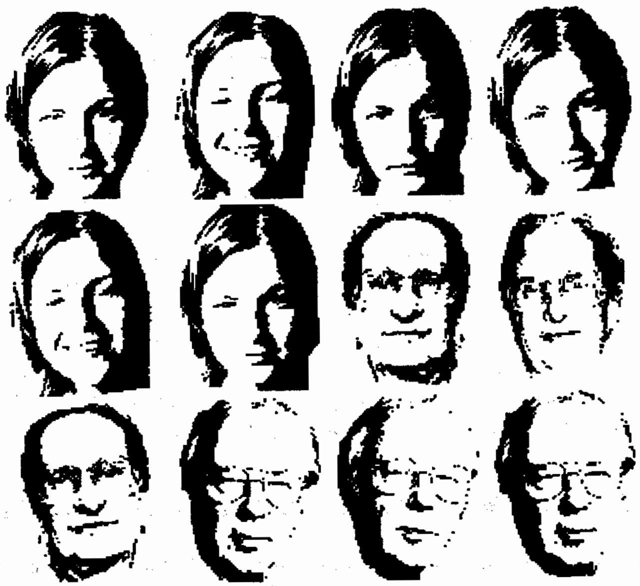


Figure 5: Training sample of frontal images of human faces.

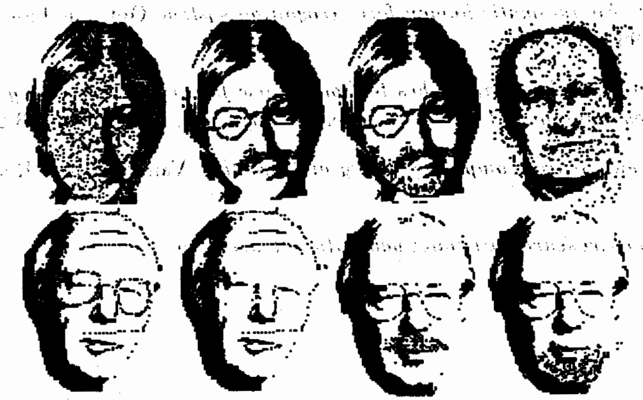


Figure 6: Noised or specially distorted faces from the previous set. All them are recognized correctly

- Ability to remove non-significant weights after training. If the radius of some hypersphere is small, then corresponding neurons may be deleted from the hidden layer without a serious violating of the classification result.
- Ability to estimate quantitatively an ambiguous situation when one input vector belongs to two different classes. For example, if an input vector activates, one neuron of the first class and two neurons of the second class, it means that probabilities of this vector to belong to those classes are correspondingly 1/3 and 2/3.
- Ability to futher improvements of the training algorithm, for instance by applying two-levels pattern recognition scheme.

7 Acknowledgements

Authors would like to thank drs V.Korenkov and J.Slawinski for involving them to the face recognition problem and permanent interest to this work and dr. L.Rozbicki for the help with the data base information.

References

[1] R.Chellappa et al, *Human and machine recognition of faces: a survey*, Proc.IEEE 83 (5) (1995) 704-740.

[2] N. Kruger et al, *Estimination of face position and pose with labeled graphs*, in: Proc. of BMVC (1996).

[3] H.Liu et al, *An automatic human face recognition system*, Opt. ans Lasers in Engeneer. 30 (1998) 305-314.

[4] k.S.Yoon et al, *Hybrid approaches to frontal view face recognition using the hidden Markov model and neural network*, Pattern Recogn. v.3, No.3 (1998)283-293.

[5] P.Wasserman, *Neural computing theory and practice*, Van Nostrand Reinhold, N.-Y.(1989).

[6] *ftp : //ftp.uk.research.att.com : pub/data/att faces.zip*

Received by Publishing Department
on November 4, 2000.