Neural Networks Applications

13 M. M. Lukoshchik, Raif Kh. Sadikkhov
Synthesis of a Self-Organizing Neural Network in a Problem of Clustering of Gray Scale
Remote Sensing Image Data .................................................. 306

95 Ivan Budnik, El khier Bouyoucef, Abdelnasser Chebira, Kourosh Madani
A Hybrid Multi Neural Network Structure Optimization Handled by a Neurocomputer
Complexity Estimator ............................................................. 310

20 Aleksandr Doudkin, Maksim Vatkin
Neocognitron-Like Multilayer Neural Network for Halftone Images Recognition ................. 315

36 Weixue Yu, Christophe Sabourin, Kourosh Madani, Jie Yan
CMAC Neural Networks Structure Optimization for Function Approximation .................. 320

77 Mehdî Chachel Amirani, Zahra Sadeghi Gol, Ali Asghar Beheshti Shirazi
Shape-Based Image Retrieval Using a Neural Network ........................................... 326

92 Sergei Bezobrazov, Vladimir Golovko
Neural Networks and Artificial Immune Systems – Malware Detection Tool .................... 332

90 Andrii Borovyj, Volodymyr Kochan, Volodymyr Turchenko, Anatoly Sachenko, Theodore
Laopoulos
Neural-Based Prediction of Power Consumption of Data-processing Instructions .............. 337

87 Iryna Turchenko, Volodymyr Kochan, Anatoly Sachenko
Neural-Network Based Method of Control Influence Forming for Neural-Network Based
Method of Control Influence Forming for Section of Mine Ventilation Network ................ 343

99+ Dmitry Kalbukhovich, Vladimir Golovko, Andreas Paczynski
Some Approaches to the Task of Path Following Based on Neural Networks and Fuzzy Logic351

99 Nikolay Kostin, Vladimir Golovko, Vladimir Rubanov
Neural Network Methods Of Forecasting Of Economic Indicators .................................. 357

Subject Index ................................................................. 360

Author Index ................................................................. 363

Information of the Co-Organizers ......................................... 365
Neocognitron-Like Multilayer Neural Network for Halftone Images Recognition

Aleksandr Doudkin, Maksim Vatkin
United Institute of Informatics Problems of NAS of Belarus, Surganov str. 6
vatkin@isi.bas-net.by, doudkin@newman.bas-net.by, http://isi.bas-net.by

Abstract: Construction peculiarity of neural network “neocognitron” was considered as applied to image recognition task. A new structure of multilayer neural network based on its logic was developed, that was designed for classification of objects having pixel distortions on halftone images. This neural network allows significantly increase recognition accuracy.

Keywords: Neocognitron, Neural networks, Image recognition.

I. INTRODUCTION

Object recognition is one of the main stages in the image analysis. Object recognition task is stated in the following way. It is necessary to discover a map of an input pattern X to Y, where Y is a class of the input pattern. The map is defined by set of pairs (input pattern, class of pattern). The number of pairs (training samples) is significantly less than the possible amount of pairs (input, output). So training set consists of training samples.

If images are strongly distorted, it is very difficult to find informative features of the object. In common case we have distortions of two types: brightness characteristic or position of a pixel is changed.

Neural network (NN) classifiers give the most robust classification in the case when images are strongly distorted. There is a number of NN suited to this goal: a multilayered perceptron, a network of radial basis function, an ART network (Adaptive Resonant Theory), Hopfield network, Kohonen self-organising map.

Ane image pattern is the structured description of the object on the image. The most appropriate NN classifier of the image patterns is a neocognitron (Fukushima, 1983). It has hierarchical structure of the object and is designed to model human visual path of brain (Hubel, 1962).

The neocognitron provides invariant recognition, which is based on successive stages of local template matching and spatial pooling. The following ways are known for development of NN hierarchical image recognition: neocognitron modification to increase its performance (Lovell, D.R., 1992), finding additional feature invariants and changing feature extraction rules to improve recognition accuracy (Wiskott, L., 2003; Satoh, S., 1997), development of unsupervised algorithms for feature hierarchy creation (Pan, Z., 1999; Wersing, H., 2003; Behnke, S., 2003; Scalzo, F., 2005).

Let’s generalize main points of structural and algorithmic solutions in image recognition based on neocognitron NN.

1. Two types of features are considered for images: simple features that can be found in any image and complex features that are spatial combination of the simple features.

2. The features are organized in hierarchy according to complexity, where the features with the same complexity form a single hierarchical level. The simple features belong to the first hierarchical level.

3. Multi-layer NN is developed, where a number of its layers is equal to a number of the hierarchical levels. Neurons of each layer are grouped, where a number of groups is equal to a number of features of given complexity. Neurons of the group (a sublayer) detects the same local feature on the input image, so a neurons activity represents a map of a feature distribution on the image.

4. Pattern recognition is made using layer-by-layer activation from the input layer that detects the simple features to the output layer that detects all features of the pattern in a whole.

So the following tasks are solves for the neocognitron development:

1. Define feature detection method (the rule of neuron activation).
2. Define a criterion of effectiveness using of set of features on one layer. That criterion allows optimizing a number of neurons groups (sublayers).
3. Develop a hierarchy of the features and a method of combination of the features of the level to form the feature of the next hierarchy level.

In (Sadykhov, 2001) we proposed a neocognitron modification based on the mentioned principles, that classifies objects with distortions of its form. Detection of a local pattern feature is realized by a pair of inhibitory and excitatory neurons. Additional generalizing neurons are used to provide invariance of recognition of a pattern having form distortions. Inputs for generalizing neurons are activities of excitatory neurons of detection layer (fig. 1),

315
where \( W \) is a matrix of pixel values of sample pattern, \( A \) is a matrix of pixel values of pattern that has distortions of pixels of two types: changes in a brightness and a position. Matrix \( A \) has bigger size because of the changes in the of pixel positions.

Let's define parameters \( B \) and \( R \) of a range of pixel distortions. \( R \) defines a maximal possible shift of pixel positions in matrix \( A \) relatively to equivalent positions in matrix \( W \), and \( B \) is a maximal possible interpixel distance, i.e. \( w_{i,j} - a_{i+j,0,0} < B \). Let's note, that \( |\tilde{v}_{i_{1},j_{1}} - \tilde{v}_{i_{2},j_{2}}| \leq \theta \) for any two neighbouring pixels \( a_{i_{1},j_{1}}, a_{i_{2},j_{2}} \) and their shifts \( \tilde{v}_{i_{1},j_{1}} = a'_{i_{1},j_{1}} - a_{i_{1},j_{1}} \) and \( \tilde{v}_{i_{2},j_{2}} = a'_{i_{2},j_{2}} - a_{i_{2},j_{2}} \), where \( \theta = \max(|\tilde{v}_{i_{1},j_{1}}|, |\tilde{v}_{i_{2},j_{2}}|) \). It specifies uniformity of geometrical distortions, i.e. the neighbouring pixels are displaced almost equally.

Let's construct a matrix

\[
D = \begin{pmatrix}
  d_{i_{1}} & \cdots & d_{x_{1}} \\
  \vdots & \ddots & \vdots \\
  d_{i_{R}} & \cdots & d_{x_{y}}
\end{pmatrix}
\]

where

\[
w_{x_{1},y_{1}} = \begin{cases}
  0, & \min_{i,j=0,2R} (a_{x_{1}+i,y_{1}+j}) \leq w_{x_{1},y_{1}} \text{ and } \\
  \min_{i,j=0,2R} (a_{x_{1}+i,y_{1}+j}) \leq w_{x_{1},y_{1}} \text{ and } w_{x_{1}+i,j} \leq \max_{i,j=0,2R} (a_{x_{1}+i,y_{1}+j}); \\
  0, & |w_{x_{1},y_{1}} - a_{x_{1}+i,y_{1}+j}| < B; \\
  1, & \text{in other cases.}
\end{cases}
\]

Now a difference function of matrices \( A \) and \( W \) is defined as follows:

\[
fd(A, W) = \frac{100}{X \cdot Y} \sum_{x_{1},y_{1}} d_{x_{1},y_{1}}.
\]

If a training image \( A' \) is entered on each training step \( t \), we shall receive a sequence of updating matrices of the reference image \( \{W^t\}_{t=0}^{T_n} \), where

\[w_{x_{1},y_{1}}(t) + \]

is the next task.

Let's consider two matrices

II. MODIFICATION OF RULE FOR FEATURE IDENTIFICATION

As an alternative we propose a new neuron structure and an activation rule. That rule is based on a fuzzy difference function of two matrices, that is invariant to value and position changing of matrix elements. As training rule we use a fuzzy mean function that is calculated dynamically. It is also invariant to the mentioned changes.
is the number of training images and \( n \) is a number of the iteration.

Let's define the function of a dynamic average:

\[
\begin{align*}
\min_{i,j=0,2R} (a_{x+i,y+j}(t)) & \leq w_{x,y}(t) \text{ and } \\
w_{x,y}(t) & \leq \max_{i,j=0,2R} (a_{x+i,y+j}(t)); \\
|w_{x,y}(t) - a_{x+k,y+n}(t)| & < B; \\
w_{x,y}(t+1) & = \begin{cases} 
  w_{x,y}(t), & \text{in other cases}
\end{cases}
\end{align*}
\]

where for \( a'_{x,y} \) is fulfilled

\[
|w_{x,y}(t) - a'_{x,y}(t)| = \min_{i,j=0,2R} \left( |w_{x,y}(t) - a_{x+i,y+j}(t)| \right).
\]

Thus, the structure of the developed neuron can be represented in figure 2 (for case of one-dimensional data), where \([\min_k, \max_k], k = 1, ..., N, \) are defined as in the formula (4):

\[
\begin{align*}
\min_k &= \min_i (a_{x+i}), \, i = 0, \ldots, 2 \cdot R; \\
\max_k &= \max_i (a_{x+i}), \, i = 0, \ldots, 2 \cdot R.
\end{align*}
\]

As a result, the performance is raised approximately to \( k_t \approx 15N \) times for the activation of one neuron in comparison with the original neocognitron.

![Fig. 2. The neuron structure.](image)

**III. NN STRUCTURE**

The structure of the developed NN is close to the structure of neocognitron network (Fukushima, 1983), but in contrast to it generalized layers are absent (fig. 3).

![Fig. 3. NN structure.](image)

The receptor layer \( R \) transforms pixel brightnesses to values of neuron activity.

Neurons that extract the same feature on the input data are united in one group. Input data for each neuron is the local part of data from the previous layer with displacement that corresponds to neuron indexes inside of it group.

Figure 4 shows the receptor fields of neurons from one group in case of one-dimensional input data.

In the case when input data are bidimensional, neurons also are organized in bidimensional structure. Thus, one group neurons activity forms a distribution of an extracted feature in the input data.

**IV. DEFINITION OF DISTORTIONS INVARIANCE RANGES OF NEURONS**

Let \( L \) be training images of size \( X \) on \( Y \), then \( B \) is calculated according to formula (7):

\[
B = \frac{1}{X \cdot Y} \left( \frac{1}{L} \sum_{x,y} \left( \frac{1}{L} \sum_{r} a'_{x,y} - \frac{1}{L} \sum_{r} a_{x,y} \right) \right) \tag{7}
\]
Let's define $F$ as an average frequency of a brightness jump on the image, for which the neighboring pixels belong to one level, if the absolute difference of their brightness is less than $B$. Now radius $R$ of geometrical distortions we set as

$$R = [0.3/F],$$

where $[x]$ is the nearest integer of $x$.

The formula (8) describes the fact, that if the radius of distortion exceeds on 30% the average diameter of smallest object feature on image, than the image mixes up so that it is impossible to identify the feature and it is impossible to identify the image in the whole. Here the smallest feature of object should be understood as feature of object that cannot be divided into smaller features. The border of image area forming smallest feature is almost convex figure. «Almost convex figure» means that any secant drawn through its centroid crosses a figure border only twice. Diameter of the smallest feature is defined as an average length of the secants.

Thus, using training we can find neuron parameters $R$ and $B$.

V. FORMATION OF TRAINING SET FOR NN LAYER

The images for each layer are formed by the network developer according to specificity of recognized objects during neocognitron NN training. It is inefficient for automatic adaptation of recognition system to different training data and we developed an automatic algorithm for training set creation.

For each layer of NN it is possible to define the sizes of a neuron receptor field projection from some layer on an input layer of NN. In figure 5 the receptor field projection from 2-nd layer on an input layer is represented in case of one-dimensional input data.

Thus, for each training image it is possible to obtain $K$ training images for some $i$-th layer of NN:

$$K = N_{in} - N_i + 1. \quad (9)$$

where $N_{in}$ is the size of an input data and $N_i$ is the size of a projection receptor fields of $i$-th layer.

![Projection of receptor field of neuron from 2-nd layer on an input layer of a network](image)

Fig. 5. A method of construction of a receptor field projection.

Training data for $i$-th layer of NN are considered as informative, if the condition is satisfied:

$$F_i/F_k \leq 0.8 \quad \text{or} \quad F_i/F_{in} \leq 0.8, \quad (10)$$

where $F_i$ is an average frequency of brightness jump on the input image and $F_k$ is an average frequency of brightness jump on the image received in a receptor field projection window.

Having training set for each layer, it is easy to notice that the problem of training of a layer is reduced to a layer input data clustering problem.

VI. ALGORITHM OF NETWORK LAYER TRAINING

Since neurons from one group have identical weight matrices, the whole group can be represented in the form of one neuron, and a layer can be represented in the form of a single-layer NN, where each neuron corresponds to the group. For training such network we can use clustering algorithm, where each cluster corresponds to one neuron. After training a network we use such NN as a prototype for construction of a required layer in a multilayer NN, where the neurons quantity in the prototype corresponds to the quantity of the groups, and the neuron weights of the prototype correspond to neurons weights matrices of the corresponding groups.

If the training set of a layer is not great and the desirable clusters quantity is known, it is convenient to apply fuzzy C-means clustering algorithm (Bezdek, 1981).

If the clusters quantity is not known, than it can be used peak grouping clustering algorithm.

For the case, when the training set of a layer is great and the desirable clusters quantity is not known, we developed clustering algorithm that is based on modification of multi-dimensional unsupervised FOREL (FORmal Element) algorithm.
VII. NN TESTING

Training set consists of 9 images of "upwards" contact pads and 9 images of "downwards" contact pads. The example of training images is presented on fig. 6.

![Fig. 6. An example of training images.](image)

The network has three layers: the receptor layer field is equal to 32x32; the neurons receptor field size is 1x1; and the size of the output layer field is 1x1.

After training, the values of parameters R and B for each layer are the following: for first layer: R = 1, B = 39; for target layer: R = 1, B = 39. The quantity of neurons groups are: for the first layer - 61, for a target layer - 4.

A fragment of testing image is represented on fig. 7. Areas of the image, which have been classified, are circled round by a black square.

As a result from 57 contact pads it has been correctly found 46. The quantity misclassified pads was 11. Total the recognition percent was 80%, percent of mistakes was 25%.

![Fig. 7. Fragment of testing image.](image)

CONCLUSIONS

In the paper two problems were considered related to hierarchical NN neocognitron design: performance increasing and creation of unsupervised algorithm for layers training that determines weights of neurons and a number of sub-layers. The new fast activation and training rule of neuron were introduced. The rules are based on distortions invariant difference function that allows comparing matrices, which have changes of matrix element position and value. Algorithms of layer training set creation and layer training of network was developed. On the base of the algorithms full unsupervised algorithm of NN training was developed. Developed NN was approved on task of topological elements recognition.

REFERENCES