Appl. Math. and Comp. Sci., 1994, vol.4, No.3, 397–411

NEURAL NETWORKS APPLIED TO VISUAL PATTERN RECOGNITION — A COMPARATIVE STUDY †

RYSZARD TADEUSIEWICZ*, ZBIGNIEW MIKRUT*

In this paper, we discuss the principles of using back-propagation neural networks for pattern recognition. The basic information on neural networks learning is presented. A comparison is drawn between classification performance of the back-propagation network and other pattern recognition methods. Special attention is paid to preprocessing of the required data. The following problems are discussed in detail: choosing network topology, selecting proper values for learning coefficients, and estimating the quality of learning and recognition processes.

1. Introduction

Pattern recognition is most frequently referred to as an example which is well realized by means of neural networks (Tadeusiewicz, 1993). The authors' experience and the analysis of achievements of other researchers indicate that the success in this apparently easy task is not obvious or straightforward. Many papers contain identical remarks about similar difficulties and present the solutions (obtained with great effort) to the problems that have already been solved many times. Thus, we decided to collect our own and other authors' achievements and to present them in the form of one synthesized paper.

Further considerations are based on a network with the topology shown in Fig. 1. This network consists of:

- an input layer of neurons connected with pattern converter (a "retina", in practice a scanner or a "frame grabber" converter),
- one or several hidden layers used to select features that are useful in the recognition process of an input pattern,
- an output layer for signalling the recognition.

This network is capable of learning using the method of back-propagation of errors. In this method, a change of the weight coefficient value $(\Delta w_{ij}^{(k)})$ for the connection between the i-th element and the j-th neuron laying in the k-th layer is described by the following formula:

$$\Delta w_{ij}^{(k)} = \eta_1 f'(e_i) x_j \delta_i^{(k)} + \eta_2 m_{ij}^{(k)} \tag{1}$$

† This work was partially supported by the State Committee for Scientific Research under grant No. 42/8/91

* Institute of Automatics, University of Mining and Metallurgy, al. Mickiewicza 30, PL-30-059 Kraków, Poland

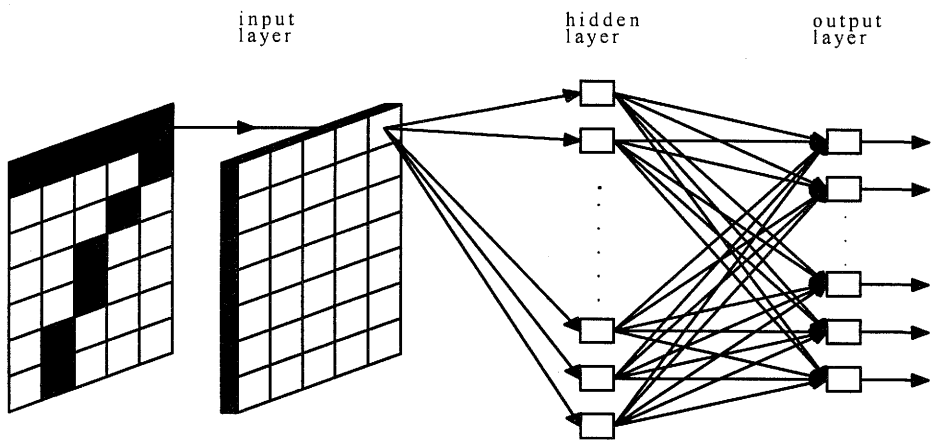

Fig. 1. Topology of the neural network for visual pattern recognition.

where $e_i = \sum_{j=1}^{n} w_{ij}^{(k)} x_j$ is the total excitation of the i-th neuron, $f(e)$ – the transfer function (typically a sigmoid), $m_{ij} = \Delta \widetilde{w}_{ij}^{(k)}$ stands for a change of the weight coefficient computed in the previous step, and the error $\delta_i^{(k)}$ is defined as follows:

$$\delta_i^{(k)} = \begin{cases} z_i - y_i & \text{for output layer neurons} \\ \sum_{l=1}^{n} w_{li}^{(k)} \delta_l^{(k+1)} & \text{for hidden layer neurons} \end{cases}$$

Other symbols are explained below:

n – the number of inputs for the considered neuron,
x_j – the j-th input signal,
z_i – the expected output signal for an output layer neuron,
η_1 – the learning rate,
η_2 – the momentum.

Besides the network proposed above other approaches to pattern recognition by using neural networks are considered in the literature, e.g. *self-organizing feature maps* by Kohonen, *Neocognitron* by Fukushima, *cellular networks* of "on center, off surround" type, *CP* (counterpropagation) networks by Hecht-Nielsen, *ART* (Adaptive Resonance Theory) by Grossberg and others. The reader can find a brief review of operating principles and topology of those networks in (Carpenter, 1989). The examples specified above are the evidence of continuous exploration of new network topologies suitable for pattern recognition. Nevertheless, it must be clearly stated that most of research and practical applications are concerned with different forms of back-propagation type network described above. Such networks, owing to their multi-functional character and apparent simplicity are most popular and most commonly used today – not only in the field of pattern recognition. Thus the rest of the paper is concerned only with this kind of network and with difficulties which are met during the attempts of its application.

2. Preliminary Stage of Pattern Recognition Process

One of more important stages of pattern recognition by means of neural networks is data preparation, which must be done in an appropriate way. The image from a camera or a scanner is entered into the computer memory by means of a suitable A/D converter and then it is processed. The sequence of processing operations usually begins with image frame analysis, image quality enhancement by means of a suitable *filtration* and preliminary *histogram analysis*. That stage usually ends with *segmentation* (separation of objects). *Transformations* utilized here are typical operations applied in pattern recognition problems and well described in the literature, e.g. (Tadeusiewicz, 1992; Pavlidis, 1982; Gonzales and Wintz, 1987; Pratt, 1978). There are a few methods of their realization.

The simplest way is to transfer the majority of problems related to preliminary pattern processing onto the neural network performing the recognition task. The role of a programmer is only to *point out* the object to be recognized. Such a method is applied in the generation of input sets for very complex networks (e.g. neocognitron (Fukushima and Wake, 1991)) capable of solving the problems of geometrical deformation of the pattern, such as object rotation, translation or scale change. However, the majority of utilized networks are not capable of solving such difficult problems and require an initial preparation of patterns to be recognized. The most widely applied method is *scaling* of the separated object in order to obtain a normalized matrix by using a sequence of appropriate geometrical transformations (sometimes a transformation to the polar coordinate system or straightening of too much slanted objects, e.g. alphanumerical characters, is used). After those transformations the object shape fills the whole matrix with preset dimensions. A final adaptation of the object to the data processing requirements is done by normalization of the matrix pixels – typically to the range of $[0, 1]$. In the most complex cases the data for the neural network may be generated on the basis of the analysis of the object's *boundary lines* or by locating particular *characteristic points* (e.g. line ending, crossing or branching) using suitable operators (e.g. convolution). The information about lines or characteristic points can be given directly in a vector form or together with a rough estimation of their position on a raster (e.g. by dividing the whole object region into several subregions and counting the number of characteristic points for each subregion separately). A more complex way of preprocessing of input data for the network is to determine some geometrical features on the basis of the whole pattern, its contour or its skeleton. Examples of such features are geometrical moments of different orders or shape factors (Tadeusiewicz, 1992; Mikrut and Tadeusiewicz, 1990). It is also possible to calculate two-dimensional Fourier transform (Maren *et al.*, 1990) or Gabor transform (Zurada, 1992) and to use the spectrum of the pattern (or its part) obtained in this way as the input of the neural network.

We applied the method of scaling and normalization of the input image (sometimes extended by "skeletonization" of some objects, for example letters and digits), with satisfactory results. Attempts at applying networks to recognition of non-scaled and non-normalized images were unsuccessful (high level of recognition errors). Attempts at using complex procedures of initial processing, in order to find specific points or lines or to identify complex geometrical features of patterns to be recognized, also (but for other reasons) demonstrated ineffectiveness. The methods applied by other authors who published the results of pattern recognition network training are collected in Tab. 1.

Tab. 1. Preprocessing of the visual data.

Problem	Preprocessing	Net input	References
Hand-written digits	binarization, size-normalization to matrix, grey-level scaling to the range of $[-1,1]$	matrix 16×16	(Le Cun *et al.*, 1990)
Hand-printed digits	size-normalization to matrix, grey-level scaling to the range of $[0,1]$	matrix 15×24	(Martin and Pittman, 1991)
Hand-printed upper-case letters	blurring through convolution with a gaussian distribution, size-normalization to matrix, grey-level scaling to the range of $[0,1]$	matrix 15×24	
Hand-written digits	linear size-normalization to matrix, grey-level scaling to the range of $[0,1]$	matrix 15×24	(Lee, 1991)
Hand-written digits	size-normalization to matrix, binarization	matrix 16×16	(Geman *et al.*, 1992)
Hand-written digits	deskewing, size-normalization, binarization, grey-level compression	vector of 28 elements	(Battiti and Colla, 1991)
Hand-written digits	skeletonization only (because digits just fitted into a rectangular frame)	vector of 96 elements	(Lisboa, 1992)
	as above and next detection of 18 features, localized in matrix quarters	vector of 72 elements	
Hand-written digits	size-normalization to matrix, grey-level scaling to the range of $[0,15]$	matrix 16×16	(Knerr *et al.*, 1991)
	skeletonization, detection of the line segments, normalization	vector of 256 elements	
Optical Char. Recognition	contour detection, smoothing, normalization to 64 points	vector of 64 elements	(Sabourin and Mitiche, 1992)
Hand-written digits	binarization and size-normalization	matrix 16×16	(Lee *et al.*, 1991)
	binary features determining: the features are based on shape analysis, existing holes and concavities, template matching	vector of 636 bin. features	
Hand-written letters	binarization and size-normalization	matrix 16×16	
	binary features determining: the features are based on shape analysis, existing holes and concavities, template matching	vector of 636 bin. features	

3. Network Topology Selection

The selection of the *network topology* has affects most significantly the learning effectiveness. The network topology can be defined by the *number of hidden layers* and the *numbers of their elements.*

The problem of the optimum selection of the network topology has been considered in many theoretical and experimental works (see Tab. 2). It is well-known that

Tab. 2. Neural networks' topologies.

Problem	Topology	Net size (numbers of layer elements)	Recog. rate	References
Hand-written digits	local, shared weights – input matrix of 16×16 elements, 4 hidden layers	16×16–4×24×24–4×12×12–12×8×8–12×4×4–10	96.6%	(Le Cun *et al.*, 1990)
Hand-printed digits	a) fully connected b) local connections between the input and the first hidden layer	15×24–150–50–(10 or 26) 15×24 –540–100–(10 or 26)	96%	(Martin and Pittman, 1991)
Hand-printed upper-case letters	c) local, shared weights	15×24–540–102–(10 or 26)	95%	
Hand-printed upper-case letters	local, shared weights applied to both layers	15×24–540–102–10	95%	(Lee, 1991)
Hand-written digits	fully connected	16×16–(from 1 to 24)–10	90%	(Geman *et al.*, 1992)
Hand-written digits	fully connected	28–10	93.4%	(Battiti and Colla, 1991)
Hand-written digits	fully connected – bit map as the input ($0 < M < 90$)	96–M–10	76%	(Lisboa, 1992)
	fully connected – features as the input ($0 < M < 90$)	72–M–10	95%	
Optical Character Recognition	fully connected – network topology depends on the height and localization of character to be recognized	64–128–36–12 64–96–60–30 64–64–36–12	96.7% (avg.)	(Sabourin and Mitiche, 1992)
Hand-written digits	fully connected – matrix as the input fully connected – vector of features	16×16–15–10 635–15–10	90.3% 92.9%	(Lee *et al.*, 1991)
Hand-written letters	fully connected – matrix as the input fully connected – vector of features	16×16–70–41 636–70–41	73.7% 88.6%	

networks with one learning layer are able to recognize patterns with linearly separated (with hyperplanes) regions of classes of objects undergoing recognition process. It refers to the regions with high degree of internal compactness and good separation between classes. It can be proven in a similar way that two-layer networks separate classes with convex hypersurfaces closing simply connected and compact regions and three-layer networks can separate regions with surfaces of an arbitrary shape. This remark, however, is not significant from the practical point of view, as for a particular recognition task it is not possible to determine *a priori* the shape of the surface separating classes of objects.

Thus, the researchers solve the problem by choosing the network topology in an arbitrary way and then, investigating its suitability, establish particular recommendations "ex post". A collection and comparison of the recommendations formulated by different researchers seem to be interesting. Experimental results presented in the literature and listed in Tab. 2 indicate that the authors were successful using networks with the number of *hidden layers* ranging from 0 to 2, where the greater number of layers was correlated with more complex connections between layers. An example of such more complex connections is the application of *local receptive fields*, limiting the possibility of linking adjacent layers to small, simply connected regions (usually rectangular in shape). This operation is done in order to decrease the total number of connections in the network (and to decrease the processing time) and assuming that during the training process, connections leading to finding local features of the analysed pattern will be generated in these regions. These local features would be used by the next network layer.

When on the defined operators a condition of identity of weights for the duration of the learning process, for all the elements lying in the considered receptor field, is imposed, we obtain another idea of the selection of connections arrangement — the method of *local shared weights* of connections (see the first four rows of Tab. 2). This method of learning allows the considered layer of the network to obtain information about the distribution of *one* feature in the image, and no information about different local features. To obtain the distribution of other features further hidden layers are applied or one hidden layer is divided into several functionally disjoint sub-layers.

A selection of the *size of a hidden layer* is a problem that is strictly connected with the network topology selection. Although some experiments prove that good learning results can be achieved by using networks without hidden layers (see Tab. 2), it is a common view that the introduction of the hidden layer and a suitable selection of the number of elements for that layer lead to *better generalization*, which means the separation of some generalized attributes (features of patterns classes). Thus, the chances of recognition of less typical patterns for a particular class can be increased.

Principles of the selection of the number of elements for the hidden layer are a subject of theoretical consideration (exemplary results are given in (Maren *et al.*, 1990)). But the analysis of the results of such theoretical investigations is disappointing: for example the optimum numbers of the elements recommended in (Maren *et al.*, 1990) (given in Tab. 2) differ sometimes by the whole order of magnitude. This is why those results, as less reliable, are not presented in this paper. It should be also mentioned that some algorithms for automatic selection of the number of elements in the hidden layer are proposed in the literature. In these algorithms, the (sub)optimum values are obtained by expanding the layer ("cascade correlation" — (Hammerstrom, 1993a) or by elimination of surplus elements and connections ("pruning" (Maren *et al.*, 1990; Orlandi *et al.*, 1991)). There are also works suggesting the use of genetic algorithms as a way of choosing the optimum topology of the neural network (including the hidden layer size) (Jones, 1993). However, on the basis of our own experience and critical analysis of the literature we are convinced that *this value should be chosen experimentally*, based on preliminary simulation of the network designed for a particular recognition task. It seems that such views are shared by many other authors (Mikrut, 1993; Geman *et al.*, 1992; Martin and Pittman, 1991).

4. Learning Parameters

The aim of neural network learning is to achieve a suitably high percentage of correct recognition of the data not presented to the network during the training process. The learning results, obtained by the authors and presented in the literature, are satisfactory. For example, the percentage of correct recognition for the task of the recognition of handwritten digits (e.g. automatic readout of postal codes) oscillates at 95% level (it can be compared with the ability of a man, which can be expressed as 96.6% correctly recognized characters (Martin, 1991)). The authors presenting the experimental results, however, have different views on the influence of various factors on the recognition accuracy. Thus various problems occurring during the planning and conducting of the experiments are listed below.

The most important factor affecting the correctness of the learning process is a suitable *choice of the training data set* and the *test set*. This reflects the views of the majority of experimenters. It is emphasized that an appropriate set:

- should have a suitable power (at least several hundred objects, usually between ten and twenty thousand);
- should be representative (containing various objects within each class);
- should have equal power within each class (to avoid preferring classes with greater power);
- should be divided at random into two parts: the training part and the test part.

The numbers of elements contained in the training and test sets in various experiments described in the literature are specified in Tab. 3.

The *selection of the learning process parameters* is another problem which can be a subject of detailed investigation. In the back-propagation learning formula (1) two parameters describing the learning process appear: the learning rate η_1 and the momentum η_2. The suitable choice of these parameters is critical, especially when the network size is large, because it influences the learning speed. Greater value of η_1 gives a greater learning speed. The upper limit is however imposed on this parameter by the liability of the network weights to oscillation. Too small values of the parameter lead to slow learning and may cause "getting stuck" in a local minimum of the error surface. The values of η_1 and η_2 are usually taken from the range of 0.1 to 0.9. Examples of these parameters' values reported in the literature are listed in Tab. 4.

Tab. 3. Training and test sets used in various experiments.

Problem	Size of the set (L – learning, T – test)	Remarks	References
Hand-written digits	L: 7291 hand-written + 2549 printed digits T: 2007 hand-written + 700 printed digits	zipcodes	(Le Cun *et al.*, 1990)
Hand-printed digits	L: 100 → 35200 T: 4000	bank checks	(Martin and Pittman, 1991)
Hand-printed upper-case letters	L: 500 → 6300 T: 2368 (generated by people different from those generating the training set)	from 110 people writing on stylus input device	
Hand-written digits	L: 30600 T: 5060	real-world financial receipts	(Lee, 1991)
Hand-written digits	L: 600 T: 600	from 12 people	(Geman *et al.*, 1992)
Hand-written digits	L: 68, 271, 2166, 6496 T: 12961		(Battiti and Colla, 1991)
Hand-written digits	L: 100· T: 50	from 15 people	(Lisboa, 1992)
Optical Char. Recognition	L: about 12000 (next: 100000) T: (?)	over 200 different font styles	(Sabourin and Mitiche, 1992)
Hand-written digits	L: 10000 (18650) T: 5000 (2711)		(Lee *et al.*, 1991)
Hand-written letters	L: 8000 T: 2865		

We conducted a research on the choice of optimum values of learning parameters (Mikrut, 1993). It has been proven experimentally that too large values of the above — mentioned parameters are less dangerous as the weight oscillations are easy to detect when monitoring the learning process. It is more difficult to determine when the training speed is clearly too small and whether it is possible to increase the parameter values. We developed and recommend our own *strategy of the selection of learning parameter values*. The strategy can be described as repeated attempts of training the network with zero value of the parameter η_2 and augmented value of the parameter η_1 until the learning process starts seeming unstable. If the value of η_1 is still being increased, it is necessary to change the value of η_2, which partially dampens the oscillations and helps to go through local minima extending (apparently) the training steps.

Sometimes, in the cases when it can be predicted that the learning process will take long, it is cost-effective to carry out more precise investigation in order to determine the relation between a standard measure of process convergence (e.g. the total

Tab. 4. Learning process parameters.

Problem	Learning rate (η_1)	Momentum (η_2)	References
Hand-printed digits	0.05	0.9	(Martin and Pittman, 1991)
Optical Character Recognition	0.5	0.5	(Sabourin and Mitiche, 1992)
Hand-written digits and letters	0.75	0.5	(Lee *et al.*, 1991)
Hand-written digits	0.1	0	(Geman *et al.*, 1992)
Pattern recognition	0.01	0.95	(Neural, 1992)
In general (for back-propagation)	0.5	0.4	(Neural, 1991)
In general (for back-propagation)	0.1	0.8	(HNC, 1991)

sum of squares in one epoch[1]) and the learning rate applied. Some exemplary results of such tests are presented in (Mikrut, 1993).

In order to accelerate the learning process and to avoid such "traps" as the one described above, the values of η_1 (and sometimes the values of η_2) are changed when the process is going on. Starting with bigger values accelerates the realization of the initial ("rough") stage of the learning process. Gradual changes of the parameters decelerate the learning process but increase its stability and accuracy, necessary at the final stages of the process. A more advanced algorithm for changes η_1 (based on the output error tracking) is described in (Neural, 1992). During the learning process, if the error is decreasing faster than that one with the preset rate — the parameter η_1 is augmented, if the error is increasing too fast — η_1 is reduced. For that algorithm it has been assumed (just for example), that the quotient of successive errors (computed for successive epochs) greater than 1.04 will result in the reduction of the learning rate by multiplying η_1 by 0.7 and cancellation of the last learning epoch (retrieval of previous weights values).

5. The End of Learning Process

One of the typical problems the researchers applying neural networks have to cope with is when to terminate the learning process. Arbitrary solutions are used most commonly: the process is terminated after a preset number of steps (or epochs). But basing on our own experience we do maintain that it is possible and purposeful to base the decision about the learning process termination on solid foundations. This possibility exists, because during the learning process some auxiliary values reflecting the learning progress can be computed without additional effort. The most common parameters are as follows:

[1] An epoch is a period corresponding to one presentation of the training sequence.

- Root Mean Square (RMS) computed according to the formula (Neural, 1991):

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - y_i)^2} \tag{2}$$

where N is the size of the output layer and other symbols are the same as in the previous formulae.

- Total sum of squares (tss) of errors in an epoch (McClelland and Rumelhart, 1987):

$$tss = \sum_{k=1}^{K}\sum_{i=1}^{N}(z_i - y_i)^2 \tag{3}$$

where K is the power of the presented set.

- An example of a normalized parameter describing also the degree of recognition reliability in classification problems is $DELTA$ index, detailed description of which is given in (Mikrut, 1993):

$$DELTA = 1 - \mathrm{pos}(\boldsymbol{Y}) - \mathrm{neg}(\boldsymbol{Y}) \tag{4}$$

where $\mathrm{pos}(\boldsymbol{Y}) = \max_{1\le i\le n} z_i\,(1 - y_i)$, $\mathrm{neg}(\boldsymbol{Y}) = \max_{1\le i\le n} (1 - z_i)\,y_i$.

When conducting experiments it is a good practice to cease the learning process every n-epochs (n – a preset number) and to test the network using both the training set and the test set. In this way one can obtain a graph of percentage of correct recognitions showing the current network status and a general trend, which is an indication of the process future. By observing the learning processes we can say that the recognition curve is not a strictly monotone function — a maximum of correct recognitions can be observed for the learning epoch, when the training set has not been recognized in 100% yet. By termination of the learning process at this moment we agree to the existence of a small (1–3%) recognition error for the *training set* and to the reduced recognition "reliability", obtaining in turn the maximum of correct recognition results for the *test set*. A detailed discussion of such a situation is presented in (Mikrut, 1993).

Table 5 contains various methods of the learning process termination used by different authors. In most cases these methods are based on the analysis of correct recognitions curve obtained for the training set (RMS value assessment belongs to the same category).

6. Additional Remarks

The tasks and solving methods described above are only a part of problems concerning the application of neural networks in pattern recognition. For example, the following two problems have not been thoroughly investigated yet. These are: the selection of the appropriate (from the task viewpoint) neuron-like element and the effect of initial weights on the learning and recognition effectiveness.

Tab. 5. Various methods of the training process termination.

Problem	Criterion	Recog. rate (for the training set)	Learning error	References
Hand-printed digits	obtaining the recognition rate of 97 – 98%	97 – 98%	~ 0.6 (Mean Square Error)	(Martin and Pittman, 1991)
Hand-written digits	stabilization of the recognition rate (for the training set)	98.9%	0.017 (MSE)	(Le Cun *et al.*, 1990)
Optical Char. Recognition	stabilization of the recognition rate (for the training set)	99.1% 97.64%	0.04 0.141 (MSE)	(Sabourin and Mitiche, 1991)
Hand-written digits	obtaining the minimum of MSE (for the test set)	?	?	(Geman *et al.*, 1992)
Hand-written digits	$RMS = 0.04$ (for the training set)	?	0.04 (Root Mean Square)	(Lisboa, 1992)

It can briefly be said that in spite of experiments in which a classical sigmoid has been applied, there are reports of good recognition effects obtained by using a sigmoid translated by $-1/2$ along Y-axis (Maren *et al.*, 1990) or a hyperbolic tangent (Le Cun *et al.*, 1990) as the neuron transfer function. Probably an improvement of the network operation was caused by the symmetry of these functions relative to the origin of the coordinate system.

It can be inferred that the initial distribution of the weights of connections affects on the learning process and recognition reliability with respect to the test set. It seems that this phenomenon is caused by the differences in generation of initial features influencing the further process of their transformation or adjustment. Some publications (Maren, 1990) suggest careful monitoring of early stages of the learning process, and in the case of lack of progress – another randomization of the weights. The selection of appropriate variability limits for the initial weights is another problem. Variability ranges proposed in professional packages for neural network modelling are equal to $[-0.1, 0.1]$ or $[-0.5, 0.5]$. When selecting the randomization range for a particular task one should follow the general principle of maintaining the initial "operating points" on the neuron transfer function within the interval of the highest derivative values (see formula (1)). When the input sums are too large, the operation moves towards the area of transfer function saturation. This causes the learning process to slow as in that area the derivative value is low.

The list of similar more or less important problems could be extended. The paper has been confined to a narrow field of pattern recognition by means of neural networks. We hope that the basic problems of neural networks application presented here will contribute to the propagation of this interesting and original pattern recognition method and will also help others to create their own applications by classifying basic ideas and analyzing typical solutions.

7. Efficiency of Pattern Recognition by Means of Networks and Other Methods

It is worthwhile to compare the effects obtained by means of neural networks with those achieved by other, "standard" methods of pattern recognition (Tadeusiewicz, 1993). Many works, both experimental and theoretical, were devoted to that problem. Such comparison is not easy as the results may be different according to the method of preprocessing of the pattern being recognized. Some reports are presented in Tab. 6.

Tab. 6. Comparison of neural networks with other pattern recognition methods.

Problem	Neural net input	Recog. rate	Pattern recog. method	Recog. rate	References
Hand-written digits	15×24 grey-level map	94.85%	Radial Basis Function — classifier	95.23%	(Lee, 1991)
			k-Nearest Neighbour method	94.86%	
Hand-written digits	16×16 bit map	90%	k-Nearest Neighbour method (Hamming metric)	90%	(Geman *et al.*, 1992)
			k-Nearest Neighbour method ("elastic matching" metric)	98%	
Hand-written digits	skeleton of the digit	76%	template matching	76%	(Lisboa, 1992)
	features determined on the basis of skeleton	95%		92%	
Optical Char. Recognition	normalized contour	96.7% (avg.)	Dynamic Contour Warping	95.9%	(Sabourin and Mitiche, 1992)
Hand-written digits	16×16 bit map	90.3%	first order Bayesian method	86.2%	(Lee *et al.*, 1991)
	vector of 636 binary features	92.9%		93.7%	
Hand-written letters	16×16 bit map	73.7%		70.6%	
	vector of 636 binary features	88.6%		90.1%	

According to the table, the *percentages of correct recognitions* obtained for the back-propagation type networks and different methods of pattern recognition are very similar, with a slight domination of neural networks. A significant exception from this rule is the result of recognition of handwritten digits, obtained using the method of the k-Nearest Neighbour (kNN) with the application of "elastic matching" (Geman *et al.*, 1992). The first order Bayes method also gave slightly better results with regard to the recognition of handwritten digits and letters, but it was not a pixel pattern, but a very long vector of different attributes that were recognized (Lee *et al.*, 1991).

A more comprehensive comparative investigation was conducted in (Lee, 1991). The results of three methods of handwritten digits recognition were compared: the k-Nearest Neighbour method, the Radial-Basis Function (RBF) algorithm and the back-propagation network. Normalized image matrices in 10-level grey scale were used as the input patterns. All the three methods gave similar percentages of correct recognitions: kNN – 94.86%, RBF – 95.23%, the network – 94.85% (see Tab. 6). The most interesting fact, however, is the comparison of the methods from the point of view of computer science: by estimating the memory utilized, learning time and recognition time. It can be inferred from such a comparison that the *neural network, which needed the least memory, was the fastest at the recognition but its learning took the longest.* The kNN algorithm was the fastest at learning, but it needed a lot of memory and recognized slowly. The RBF method learnt quicker than the network, but it also needed more memory and recognized more slowly. Taking all these factors into account it cannot be said clearly which of the algorithms appeared to be the best, nevertheless the network method was the quickest and utilized the smallest memory capacity.

One of more important problems raised by many authors making such comparisons (also by (Lee, 1991)) are mistakes made sometimes by the network (misclassifications, false-positive responses) which indicates with high confidence a different object instead of the correct one. This problem requires additional research on the network learning process and generates the need of working out a method governing the separation of features with regard to minimization of erroneous recognitions.

8. Conclusions

It should be emphasised that all the problems mentioned in this paper are open from the scientific point of view. In spite of the wide range of publications and our own considerations presented in the paper, the question of a suitable selection of the neural network and the learning strategy for the recognition task, as well as the quality of obtained results are still interesting research areas and allow the researchers to reach quite original results. It should also be mentioned that in order to carry out all the comparisons computer programs were used to *simulate* the neural network, although the neural network has a parallel computational structure with the properties (for example computational accuracy) different from the utilized simulators. Thus, the development of hardware systems for the modelling of parallel-structured learning networks for pattern recognition should result in the repetition of tests and revision of the tests results. Hardware-based recognition systems will be much more efficient, as the processing time and memory parameters of such systems become definitely better than those of conventional recognition methods, realized (by definition) on the basis of sequential computers.

References

Battiti R. and Colla A.M. (1991): *A parallel implementation of neural models for pattern recognition.* — Fourth Italian Workshop *Parallel Architectures & Neural Networks*, Vietri sul Mare, Salerno, E.R. Caianiello (Ed.): World Scientific Publishing Co, pp.115–122.

Carpenter G.A. (1989): *Neural network models for pattern recognition and associative memory.* — Neural Networks, v.2, No.2, pp.243–257.

Freeman J.A. and Skapura D.M. (1991): *Neural Networks. Algorithms, Applications and Programming Techniques.* — Reading: Addison-Wesley

Fukushima K. and Wake N. (1991): *Handwritten alphanumeric character recognition by the neocognitron.* — IEEE Trans. Neural Networks, v.2, No.3, pp.355–365.

Geman S., Bienenstock E. and Doursat R. (1992): *Neural networks and the bias/variance dilemma.* — Neural Computation, v.4, No.1, pp.1–58.

Gonzalez R. and Wintz P. (1987): *Digital Image Processing.* — Reading: Addison-Wesley

Hammerstrom D. (1993a): *Working with neural networks.* — IEEE Spectrum, July, pp.46–53.

Hammerstrom D. (1993b): *Neural networks at work.* — IEEE Spectrum, June, pp.26–32.

Jones A.J. (1993): *Genetic algorithms and their applications to the design of neural networks.* — Neural Computing & Applications, v.1, No.1, pp.32–45.

Hecht-Nielsen R. (1990): *Neurocomputing.* — Reading: Addison-Wesley.

HNC (1991): *HNC Neurosoftware Manual.* — San Diego: Hecht-Nielsen Corp.

Knerr S., Personnaz L. and Dreyfus G. (1991): *From theory to silicon: an efficient procedure for the design of neural classifiers and its application to the automatic recognition of handwritten digits.* — Proc. of *Neuronimes'91*, Nimes, France, pp.149–158.

Le Cun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W. and Jackel L.D. (1989): *Backpropagation applied to handwritten zip code recognition.* — Neural Computation, v.1, No.4, pp.541–551.

Le Cun Y., Matan O., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. and Baird H.S. (1990): *Handwritten Zip Code Recognition with Multilayer Networks.* — IEEE Communication, pp.35–40.

Lee D.S., Srihari S.N. and Gaborski R. (1991): *Bayesian and neural network pattern recognition: a theoretical connection and empirical results with handwritten characters,* In: Artificial Neural Networks and Statistical Pattern Recognition. Old and New Connections (Sethi I.K. and Jain A.K., Eds). — New York: Elsevier.

Lee Y. (1991): *Handwritten digit recognition using K nearest neighbor, radial-basis function, and backpropagation neural networks.* — Neural Computation, v.3, No.3, pp.440–449.

Lisboa P.J.G. (1992): *Single layer perceptron for the recognition of hand-written digits.* — Int. J. Neural Networks, v.3, No.1, pp.17–22.

Maren A.I., Harston C.T. and Pap R.M. (1990): *Handbook of Neural Computing Applications.* — San Diego: Academic Press, Inc.

Martin G.L. and Pittman J.A. (1991): *Recognizing hand-printed letters and digits using backpropagation learning.* — Neural Computation, v.3, No.2, pp.258–267.

McClelland J.L. and Rumelhart D.E. (1987): *Explorations in Parallel Distributed Processing.* — Cambridge: MIT Press.

Mikrut Z. (1987): *A computer system for neural network modelling – simulation of receptor+bipolar cell+neuron system.* — Eighth Polish Workshop *Biocybernetics and Biomedical Engineering*, Cracow, pp.72–74, (in Polish).

Mikrut Z. and Tadeusiewicz R. (1990): *Automatic visual identification of objects on robotized work-stands.* — Electrotechnics, v.9, No.3–4, pp.485–500, (in Polish).

Mikrut Z. (1993): *Hand-written digit recognition using neural networks with various architectures.* — Scientific Bulletins of the University of Mining and Metallurgy, Ser. *Automatics*, Bulletin No.66, pp.31-58, Cracow, (in Polish).

Neural (1991): *Neural Computing.* — Pittsburgh: NeuralWare Inc.

Neural (1992): *Neural Network Toolbox User's Guide (for use with Matlab).* — Natick: The MathWorks, Inc.

Orlandi G., Piazza F., Uncini A., Luminari E. and Ascone A. (1991): *Dynamic pruning in artificial neural networks.* — Fourth Italian Workshop *Parallel Architectures & Neural Networks*, World Scientific Publishing Co, Vietri sal Mare, Salerno, pp.199–208.

Pavlidis T. (1982): *Algorithms for Graphics and Image Processing.* — Rockville: Computer Science Press.

Pratt K. (1978): *Digital Image Processing.* — New York: Wiley & Sons.

Pawlik P. (1993): *Knowledge representation in neural network - analysis of single neuron learning results.* — Scientific Bulletins of the University of Mining and Metallurgy, Ser. *Automatics*, Bulletin No.66, pp.101–118, Cracow, (in Polish).

Raudys S. and Jain A.K. (1991): *Small sample size problems in designing artificial neural networks*, In: Artificial Neural Networks and Statistical Pattern Recognition. Old and New Connections (Sethi I.K. and Jain A.K., Eds.). — Amsterdam: Elsevier.

Sabourin M.G. and Mitiche A. (1991): *Optical character recognition by a neural network.* — Proc. *NeuroNimes'91*, Nimes, France, pp.135–148.

Sabourin M.G. and Mitiche A. (1992): *Optical character recognition by a neural network.* — Neural Networks, v.5, No.5, pp.843–852.

Tadeusiewicz R. and Flasiński M. (1991): *Pattern Recognition.* — Warsaw: Polish Scientific Publishers, (in Polish).

Tadeusiewicz R. (1992): *Vision Systems for Industrial Robots.* — Warsaw: Polish Scientific Publishers, (in Polish).

Tadeusiewicz R. (1993): *Neural Networks.* — Warsaw: Academic Publishing House, (in Polish).

Zurada J.M. (1992): *Introduction to Artificial Neural Systems.* — St. Paul: West Publishing Company,