Appl. Math. and Comp. Sci., 1994, vol.4, No.3, 343–352

ESTIMATION OF THE HIDDEN LAYER SIZE BASED ON ANALYSIS OF NEURAL NETWORKS FOR HANDWRITTEN DIGITS RECOGNITION†

ZBIGNIEW MIKRUT*

The paper contains quantitative information about parameters generated in hidden layers of back-propagation type neural networks for handwritten digits recognition. On the basis of this information, a method of determination of the lower limit for the number of hidden layer elements for such networks is given. The method applies a suitable formula of combinatorics. The above-mentioned information is used to create an algorithm for the training set analysis aimed for more effective neural network initialization.

1. Introduction

A satisfactory solution to the problem of a suitable selection of the neural network topology for a particular application has not been found yet. As it was mentioned in a review publication (Tadeusiewicz, 1993), the topologies of networks for pattern recognition are defined by different authors in an arbitrary, subjective way. This concerns the number of hidden layers as well as the number of elements in each layer and the arrangement of connections between the layers. Each of these parameters influences considerably not only the learning process but also the results of the test set recognition. It can be inferred from these results whether the network has managed to create an appropriate internal knowledge representation during the learning process.

To determine such behaviour of the network, the term "generalization" has been introduced. This notion (for example according to (Kendall *et al.*, 1993)) should be understood as an optimum modelling of the training set (ensuring the best recognition results for the test set). The methods of reducing the number of connection weights comprise for example the application of the so-called shared weights or receptor fields (connections with a limited range), pruning (setting small weight values to zero) (Hertz *et al.*, 1991; Neural, 1991) and automatic reduction of small weights during the learning process by a suitable modification of the cost function (Hertz *et al.*, 1991). With regard to the number of elements in the hidden layer it is suggested to accept a general rule that the best model of the training set is a model creating the smallest and most compact internal representation (similarly as for the data coding or data compression) (Kendall *et al.*, 1993).

† 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

The attempts of practical application of this rule led on the one hand to the search for optimum topologies using "trial and error" method and to the statement of formulae (based on proprietary heuristics) describing the relation between the numbers of elements belonging to individual layers (Maren *et al.*, 1990). On the other hand, a number of algorithms for automatic selection of the hidden layer size have been developed for particular applications, for example cascade-correlation (Hammerstrom, 1993; Hertz *et al.*, 1991), upstart (Hertz *et al.*, 1991) or tilting algorithm (Hertz *et al.*, 1991). Most of them are really efficient for small networks (usually examples of synthesis of networks performing boolean functions with N inputs and one two-valued output are given). But still a statement is valid that no criteria providing the termination of the procedure of adding elements to the hidden layer without continuous referring to the test set (cross validation) have been worked out (Kendall *et al.*, 1993).

This paper is an attempt to work out a different approach to the problem of determining the (sub)optimum number of elements of one (and only one) hidden layer for the back-propagation type network for pattern recognition. On the basis of the results of experiments with neural networks of different topologies for handwritten digits recognition (Mikrut, 1993), the rules of creation and distribution of image features in the hidden layer will be investigated (for different hidden layer sizes). Then, using combinatorial formulae an attempt to determine a lower bound for the unknown number of neurons will be made. It seems to us that such investigations may constitute an introduction to a deeper analysis of the training set structure, which may lead to finding the optimum (for such type of problems) number of elements belonging to the hidden layer.

2. Distribution of Features in Networks with Different Topologies

The possibilities of application of back-propagation networks for pattern recognition were analysed in (Mikrut, 1993) on the basis of a classical problem of handwritten digits recognition. The input layer was a 22×22-pixel binary matrix. Each digit image was normalized to that matrix. The fully-connected networks differing with respect to the size of hidden layers (0, 4, 5, 6, 7, 10, 20) were trained by using a 250-element matrix set. The learning results were checked with the use of a 150-element test set. The simulations were performed several times and average values were introduced into the collective specifications. The results of several series of experiments are shown in Fig. 1.

All the topologies (without 484–4–10)[1] provided 100% recognition of the training set, but a particular consideration should be given to a surprisingly good result for the network without hidden layer (see also (Lisboa, 1992)). In order to analyse the rules of feature separation (internal representation), the most interesting networks with 5, 7 and 10 elements in the hidden layer were selected (see Fig. 1). By using special capabilities of the software[2] the following quantities were recorded during the network

[1] Notation: the number of elements of the input layer — hidden layer — output layer.

[2] The analysis was performed by means of the NeuralWorks Professional II+ program made by NeuralWare, run on SUN IPC+ workstation.

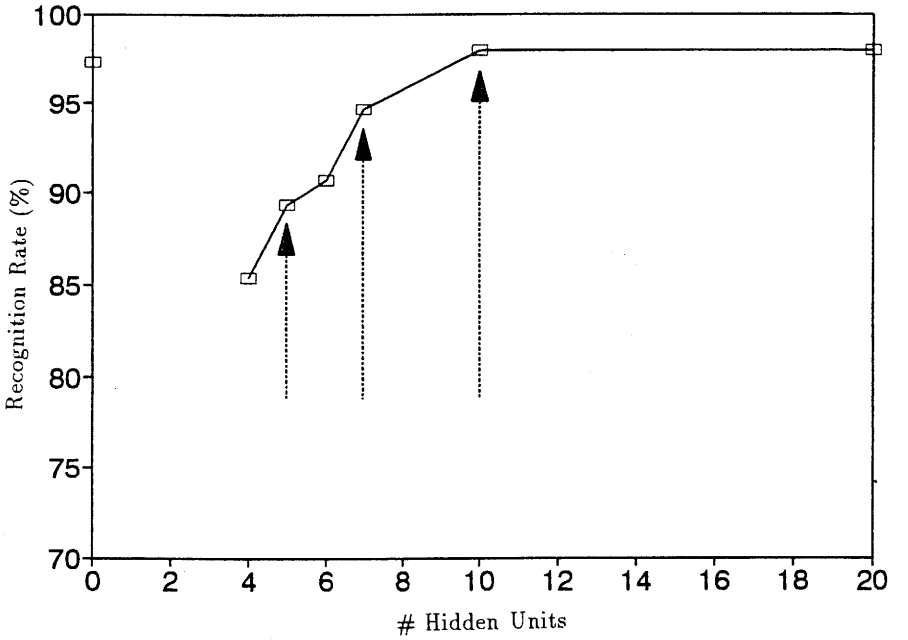

Fig. 1. Recognition rate as a function of the number of hidden units.

testing: the matrix of weights of connections of the input layer with individual elements belonging to the hidden layer, the matrix of the input signals of the hidden layer elements, the outputs of the hidden layer and the input vectors of the output layer elements.

The testing of the network was performed by inputting 10 so-called averaged images (for the training set: the sum of the matrices of a given class divided by the number of images of the given class). The quantities determined above (except for the matrix of the connection weights, which were invariable during the testing process) were recorded for each of the averaged images. The results of the "investigation" described above for the networks with 5, 7 and 10 elements in the hidden layer are collected in Tabs. 1–3.

The main object of interest were the values generated by the outputs of individual elements of the hidden layer. These values were approximated (to a reasonable degree) by limit values of a sigmoidal function, i.e. by 0 or 1 signals. In the event of the difference between the limit value and the generated value, the latter one was left unchanged. In this way a distribution of features was obtained: horizontally for the individual presented classes (digits from 0 to 9), vertically for individual elements of the hidden layer. For example taking Tab. 1 into consideration (484–5–10 network) one can say that after inputting an averaged matrix corresponding to the digit '5' — the first, second, third and fifth element of the hidden layer generate the value of 1. This means that some featuresof the '5' digit has been coded in the above-mentioned

Tab. 1. Outputs of the hidden layer of the 484–5–10 network.

	0	1	2	3	4	5	6	7	8	9		
1	1	1	0	0	1	1	1	0	0.6	1	6	5
2	0.3	1	1	1	0	1	0	1	1	1	7	1
3	1	0	1	1	0.5	1	0.6	0	1	1	6	4
4	0	1	0	1	1	0	0	1	1	1	6	5
5	0	1	1	0	1	1	1	1	1	0	7	5
	2	4	3	3	3	4	2	3	4	4	**G**	
	2	1	2	2	1	3	2	2	3	2		**U**

Tab. 2. Outputs of the hidden layer of the 484–7–10 network.

	0	1	2	3	4	5	6	7	8	9		
1	0	0	1	1	0	0	0	1	0	0	3	3
2	0	0	0	1	1	1	1	1	1	0	6	3
3	0	1	0	0	1	0	0	1	0	0	3	3
4	0	1	1	0	1	0	0	1	1	1	6	2
5	1	0	0	0	1	0	1	0	0	0	3	3
6	1	1	1	0	0	1	1	0	1	0.6	6	4
7	1	0	0	1	1	1	0	0	0	1	5	5
	3	3	3	3	5	3	3	4	3	2	**G**	
	3	2	2	2	3	3	2	2	2	2		**U**

elements. The analysis of the table row corresponding to the first element of the hidden layer reveals that some features of the following digits: 0, 1, 4, 5, 6, 9 and to some extent 8 (value of 0.6) have been stored in that row. The global amounts of these features are summed in the 'G' row for individual digits and in the 'G' column for individual elements of the hidden layer, respectively.

Tab. 3. Outputs of the hidden layer of the 484–10–10 network.

	0	1	2	3	4	5	6	7	8	9		
1	0	0	0	0	1	1	1	1	0	0	4	4
2	1	0	1	0	0.4	1	0	1	1	1	6	4
3	0	1	1	1	0	0	0	1	1	1	6	5
4	1	0	1	1	0	1	1	0	0.7	1	7	5
5	0	0	1	1	0	1	1	1	1	0	6	6
6	0	1	0	1	1	1	0	1	1	1	7	5
7	1	1	0	1	1	0.7	0	1	0	1	7	3
8	1	1	1	0	0	0	1	1	0	0	5	4
9	1	1	0	0	1	1	1	0	1	1	7	5
10	1	0	1	1	1	0	1	1	1	0.6	8	3
	6	5	6	6	5	7	6	8	7	7	**G**	
	5	4	5	5	4	4	4	5	4	4		**U**

Information concerning the actual utilization of generated features by the output layer elements was contained the distribution of features. The information was extracted from the input vectors of the output layer elements. The darker the table field, the greater the weight of a particular connection, which means the greater influence of a particular feature on the final result of the recognition process. After finding the column corresponding to the '5' digit (as in the previous example) it can be noted that the features generated by the first, second and fifth element influence the recognition result. The feature causing the generation of 1 as the output of the third element of the hidden layer is not taken into account – the weight of the connection between this element and corresponding element of the output layer was set to zero during the learning process. Similarly as in the previous example, the corresponding sums of features utilized in the recognition process are collected in the 'U' row and 'U' column of the table.

One of more important conclusions which can be drawn from the analysis of the tables presented for all the tested networks is the fact of generating of *binary signals* as the outputs of the hidden layer elements. For the 484–5–10 network only 4 outputs out of 50 are not binary. For the 484–7–10 and 484–10–10 networks the percentages of

non- binary outputs are even smaller. They are equal to 1/70 and 4/100, respectively. Additionally (it can be induced from Tabs. 1 through 3), these features are not taken into account during the recognition process.

3. Estimation of Lower Limit of the Hidden Layer Size

Three conclusions can be drawn from Tabs. 1, 2 and 3. Firstly, not all the generated features are utilized in the recognition process. Secondly, the more elements situated in the hidden layer, the more (on the average) features generated and more features utilized in the recognition process. Thirdly, the matrices of the weights of connections between individual elements of the hidden layer and the input layer are capable of storing a particular number of features (depending on the kind of the problem being solved) and the features must be divided consistently into the elements of the hidden layer. This number is about six, but the features are obviously different (in the sense of planar distribution of the significant weights of connections) for the networks with different sizes of the hidden layer. These trends are illustrated by Tab. 4. That table groups the average values of features, generated and utilized during the recognition process. These values are computed for the elements of the hidden layer and for the classes of objects undergoing recognition.

Tab. 4. The average values of the features generated in various networks.

Number of hidden units	Number of features (avg): generated by one hidden unit	utilized by output layer	received by one output unit	utilized by one output unit
5	6	4	3	2
7	4.5	3	3	2
10	6	4	6	4

On the basis of the results presented above it can be assumed that the output layer generates the recognition result on the basis of a few features only. It can be inferred from Tab. 4 that there may be from 2 to 4 features. After the application of a formula of counting theory, giving the number of combinations of R elements taken from the set of size N

$$C = \frac{N!}{R!(N-R)!} \tag{1}$$

where C (the number of combinations) corresponds to the number of elements of the output layer, N is the number of elements of the hidden layer and R is the number of features taken into consideration, the R-th order equation can be obtained, where N is unknown:

$$N^R + aN^{(R-1)} + bN^{(R-2)} + \ldots + C = 0 \tag{2}$$

Table 5 contains the roots of eqn. (2) for selected numbers of features ($R = 2, 3, 4, 5$) and for several sizes of the output layer of the network ($C = 3, 5, 7, ..., 50$). The minimum values in each row are denoted by using grey colour. These values are rounded up and given in an additional column located on the right. They constitute estimated lower limits on the hidden layer size for the networks with different sizes of the output layer.

Tab. 5. Computation of the number of hidden units – the roots of eqn. (2).

	R				
C	2	3	4	5	
3	3.0	3.7	4.6	5.5	**3**
5	3.7	4.2	5.0	5.9	**4**
7	4.3	4.6	5.3	6.1	**5**
10	5.0	5.0	5.6	6.4	**5**
11	5.2	5.1	5.7	6.4	**6**
15	6.0	5.5	6.0	6.7	**6**
17	6.4	5.7	6.1	6.8	**6**
20	6.8	6.0	6.3	7.0	**6**
23	7.3	6.2	6.5	7.1	**7**
25	7.6	6.4	6.6	7.2	**7**
30	8.3	6.7	6.8	7.3	**7**
40	9.5	7.2	7.2	7.6	**8**
50	10.5	7.7	7.5	7.9	**8**

It can be noticed that a significant increase in the number of output elements (from 3 to 50) corresponds to insignificant changes in the hidden layer size (from 3 to 8). Additionally, these values depend on the number of features taken into

consideration during the recognition process only to an insignificant degree (see Tab. 5, individual rows showing the changes of the number of hidden layer elements).

4. Further Research Fields

The results of practical simulations of back-propagation type networks revealed that the hidden layer generates binary signals as its outputs. The features coded in the matrices of weights of connections between the hidden layer elements and the input layer are distributed in a rather uniform way among individual elements of the hidden layer, but not all the elements are utilized in the further recognition process. It has been shown on the basis of the analysis of average numbers of features used for the recognition and using a suitable formula of combinatorics, how the lower limit on the hidden layer size can be estimated.

A lower limit on the area of potential optimization of the number of the hidden layer elements has been found in this way. Of course, the real size of the hidden layer always depends on the task realized by the neural network.

The task is realized by different networks with different accuracy. The final results depend on the following factors, selected on the basis of performed handwritten digits recognition experiments (Mikrut, 1993):

- selected (usually in an arbitrary way) network topology (see Fig. 1),
- initial distribution of the weights of connections obtained by means of an initial randomization,
- the order of presentation of particular elements of the training set (which is usually also chosen at random).

The influence of the last two factors can be observed in Fig. 2 which shows several typical graphs of the test set recognition level obtained during the 484–5–10 network learning. The experiment conditions differed only in that while some of them were conducted for the networks with the same initial weights and different order of the training set presentation, the rest of them were conducted with inverted conditions. It is obvious that three factors listed above *act together* and cause the separation of *different features* on the hidden layer elements, which in turn results in better or worse recognition of the test set.

It appears that on the basis of the analysis of results collected in the same form as in Tabs. 1, 2 and 3 and the topological distribution of features among the hidden layer elements an attempt to create an algorithm for the analysis of training sets can be made. The aim of the analysis will be the separation of suitable feature sets and a consistent distribution of features among the hidden layer elements. Such work would be aimed at the determination of the (sub)optimum size of the hidden layer and the preliminary initialization of the network's weights in a way providing only their precise adjustment during the learning process.

The algorithm suggested for such a task would consist of the following steps:

1. Division of the whole training set into classes and creation of their one-element

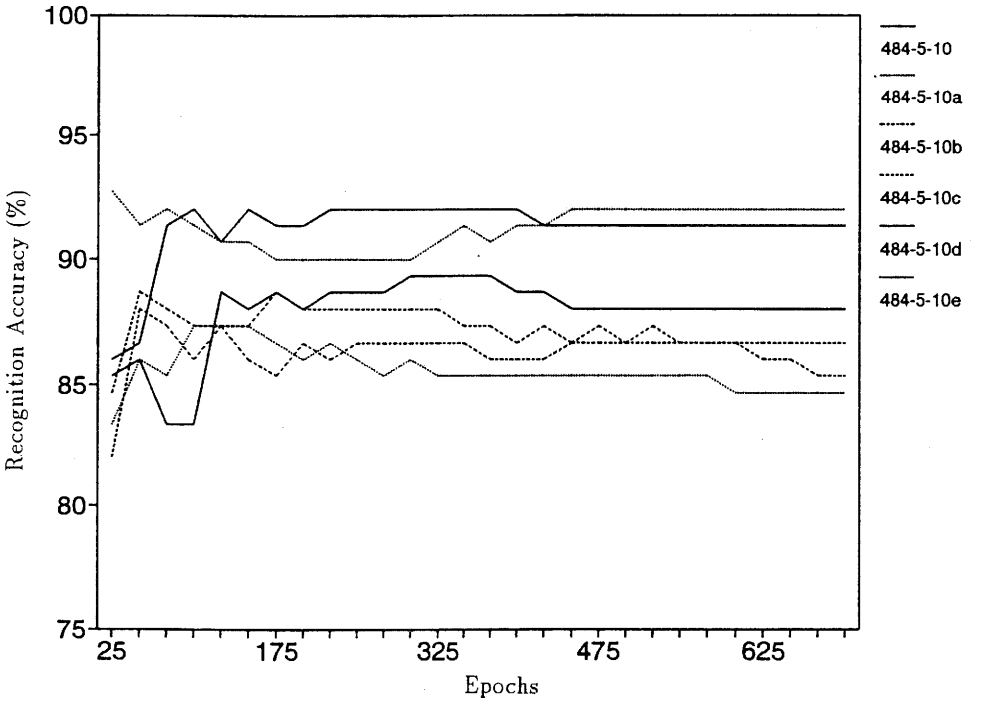

Fig. 2. Recognition rates obtained during the 484–5–10 network training.

representation. One of simple formulae which can be utilized here is as follows:

$$class\_representation = \sum object\_of\_particular\_class - \frac{\sum objects\_of\_other\_classes}{number\_of\_other\_classes}$$

The representations of classes will be equivalent to unique features of particular classes and in the neural network nomenclature they will create the weights distributions, approximately corresponding to these features.

2. Creation of generalized training images, e.g. by simple averaging of the sets corresponding to the objects of individual classes.
3. Iterative generation of features, combined with the distribution of features:
   a) determination of the basic class representation,
   b) "putting" (overlaying) successive class representations on the previously determined (e.g. by means of averaging),
   c) testing of representations obtained in the described way by training images in order to determine the real values of generated outputs,
   d) moving to the following basic class representation (see 3a) when the output value drops below a preset level or when the preset number of overlays is reached.

In Step 3b of the algorithm information about the minimum size of the hidden layer (2) should be utilized, and that means assuming a suitable number of overlays – according to the values determined in the same way as in Tab. 5. For example, considering the problem of recognition of 10 classes ($C = 10$) the first thing to analyse in Step 3b will be the combination of the second and the third overlays of the class features (including the basic class). The maximum values of the outputs during the testing should be selected (Step 3c). If it is impossible to obtain consistent distributions of features (Step 3b), the number of combinations should be increased gradually. This involves automatically an increase in the number of the hidden layer elements – for example, if $R = 4$ and $C = 10$, then $N > 6$ (see the corresponding row of Tab. 5).

If the algorithm is successful, the weights of the hidden layer elements should be initialized after an appropriate normalization with the values obtained in Step 3b. The weights between the hidden layer and the output layer should realize the combinations of features which have been planned for the correct recognition. The outline of the algorithm presented above is the subject of research being conducted.

References

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

Hertz J., Krogh A. and Palmer R.G. (1991): *Introduction to the Theory of Neural Computation.* — Reading: Addison-Wesley.

Kendall G.D., Hall T.J. and Newton T.J. (1993): *An investigation of the generalisation performance of neural networks applied to lofargram classification.* — Neural Computing & Applications, v.1, No.2, pp.147–159.

Lisboa P.J.G. (1992): *Single layer perceptron for the recognition of hand-written digits.* — Int. J. of 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.

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

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.

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

Tadeusiewicz R. and Mikrut Z. (1994): *Neural Networks applied to visual pattern recognition – a comparative study.* — Appl. Math. and Comp. Sci., (In this issue).

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