STUDIA INFORMATICA 2014
Volume 35 Number 4 (118)

Piotr CZEKALSKI, Karol ŁYP
Silesian University of Technology, Institute of Informatics

NEURAL NETWORK STRUCTURE OPTIMIZATION IN PATTERN RECOGNITION

Summary. This paper presents the analysis of the feed-forward, multilayer feed-forward network and its structure and parameters on pattern recognition effectiveness. The detailed, experimental results in Latin alphabet recognition with respect to the number of network layers, activation function and its parameters, number of connections between layers and output coding is discussed.
Keywords: neural networks, pattern recognition, OCR, optimization

OPTYMALIZACJA STRUKTUR SIECI NEURONOWYCH W ZASTOSOWANIACH ROZPOZNAWANIA WZORCÓW

Streszczenie. W artykule przedstawiono analizę zagadnienia wpływu struktury oraz parametrów wielowarstwowej sztucznej sieci neuronowej na osiągane rezultaty w zastosowaniach rozpoznawania wzorców, w szczególności optycznego rozpoznawania znaków alfabetu łacińskiego. Przedstawiono szczegółowe wyniki z przeprowadzonych eksperymentów w zakresie liczby warstw sieci, postaci i parametrów funkcji wzbudzenia neuronu, liczby połączeń oraz kodowania wartości wyjściowych.
Słowa kluczowe: sieci neuronowe, rozpoznawanie wzorców, OCR, optymalizacja

1. Multilayer feed-forward network in optical character recognition

Artificial Neural Networks (ANN) have wide industrial and research applications over the modern AI history. The common one is so called pattern recognition, particularly OCR (Optical Character Recognition) [6, 11, 12, 15, 17], Big Data analysis, speech recognition (i.e. successfully implemented in Google Android Speech Recognition), etc. [2]. The multilayer feed-forward network (MFN) is one of the most widespread and known structures of ANNs, readily implemented as being well documented and having strong mathematical fundaments along with a vast number of network training algorithms, including perhaps the most popular one – the back propagation [14]. The MFNs offers great flexibility in terms of its structure and parameters, including number of layers, activation function form and parameters, the amount of internal connections between the layers and neurons, output coding and many more as well as subject of optimization [1, 13]. They tend to be the base for many structure modifications, i.e. multistage pattern recognition [18]. This arises serious questions on the impact of particular architecture of the MFN to the quality of the classification of the network, its total error and training efficiency [5, 8]. This paper discusses the impact of the selected parameters of the network structure when applied to the OCR problem as literature lacks of comprehensive review on it. One may find ready-tuned MFNs with the configuration provided but it is usually delivered an as-is solution, with no research nor analysis what may happen if the configuration of the network is changed. This leaded authors to start research in at least limited but reasonable range of experiments, presenting an impact of the MFN construction and the parameters to the classification quality in OCR. In the chapter 2 authors presented introduction to the experiment configuration and research range. Chapter 3 discusses data sets, MFN training and quality estimation methodology during experiments. Chapter 4 along with subchapters provide detailed and summary results while chapter 5 delivers discussion and final conclusion.

The experimental results provide review on the impact of the MFN parameters regarding the OCR of the popular letters of the Latin alphabet (basic set, 26 characters), using Calibri, Arial, Tahoma, Times New Roman and Verdana fonts. There exist different approaches to the input data format, its binarization and sample probing in OCR problem. In this paper authors use very simple model where recognition subject is a monochrome bitmap, representing a pattern (here a single Latin letter) where each bitmap point presents a bipolar value (black or white).

2. MFN structure and investigation range

The chosen approach to the OCR problem fixes the inputs as required to suit the bitmap representing the single pattern (a letter) given by 12 rows, 15 columns (180 pixels total, see Fig. 1). The input data are then given by the vector of the bipolar pixel values (by rows) where white pixel equals -1.0 while the black one +1.0.

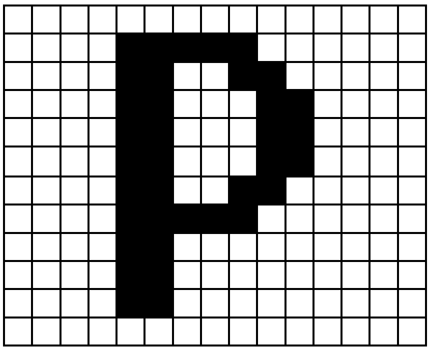

Fig. 1. Input data format
Rys. 1. Format danych wejściowych

The number of the parameters investigated by authors to find the optimal approach regarding MFN structure and parameters in OCR application includes:

- number of hidden layers {1,2},
- number of neurons in hidden layer/layers {12,25,50,75,100,150},
- cross-layer connection ratio{50%,60%,70%,80%,90%,100%}where 100% means one neuron can reach any other in the following layer,
- output format:
    - 26 outputs (one per letter),
    - 14 outputs: 1 of 13 code with 14^{th} output representing the upper / lower range,
    - 9 outputs: 1 of 7 code with two binary outputs representing a quarter,
    - 12 outputs: twice 1 of 6 code, the first one present the group, the second one position within the group.
- neuron activation functions:
    - Sigmoidal (eq. (1)) where $\beta \in \{0.5, 0.75, 1.0, 1.25, 1.5, 2.0\}$:

$$f_{sig}(x) = \frac{1}{1 + e^{-\beta x}} \tag{1}$$

    - Hyperbolic Tangent (eq. (2)) where $\beta \in \{0.01, 0.02, 0.04, 0.08, 0.15, 0.3\}$:

$$f_{\tanh}(x) = tgh(\beta x) \tag{2}$$

    - Sine (eq. (3)) where $\beta \in \{0.01, 0.02, 0.04, 0.08, 0.15, 0.3\}$:

$$f_{\sin}(x) = \begin{cases} x \le -\pi & -1 \\ x \in (-\pi; \pi) & \sin(\beta x) \\ x \ge \pi & 1 \end{cases} \tag{3}$$

    - Bipolar (eq. (4)) where $\beta \in \{0.01, 0.02, 0.04, 0.08, 0.15, 0.3\}$:

$$f_{bip}(x) = \begin{cases} x \le -\frac{1}{\beta} & -1 \\ x \in (-\frac{1}{\beta}; \frac{1}{\beta}) & \beta x \\ x \ge \frac{1}{\beta} & 1 \end{cases} \tag{4}$$

- Gaussian (eq. (5)) where $\mu \in \{2.0, 3.0, 4.0\}$ and $\sigma \in \{4.0, 4.5, 5.0, 5.5, 6.0, 7.0\}$:

$$f_{gauss}(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

3. MFN training

During all experiments underlying this paper, the classical back propagation [14, 17] method was used. The fixed parameters of the method are constant during every experiment and juxtaposed below [7]:

- training rate: 0.9,
- training rate increase coefficient: 1.03,
- training rate decrease coefficient: 0.9,
- maximum training error increase: 1.1,
- lower bound of the training rate: 0.1.

3.1. Input data – training and validation

According to the main goal of the research, the training data set should be kept simple and reliable, also not to bring extra modality. The estimated result shall be known as well. The input training and validating data sets have been integrated together and organized into 5 sets of the training points constituted of 26 the Latin capital letters each (Fig. 2).Every set has been prepared using different font (see chapter 1 for complete list). The quality of each MFN structure investigated has been performed using cross validation, repeated 5 times each. The training on MFN has been performed on 100 epochs where datasets were rotated randomly. During experiments, each network was trained with the same order of the samples to ensure comparability of the results among the experiments.

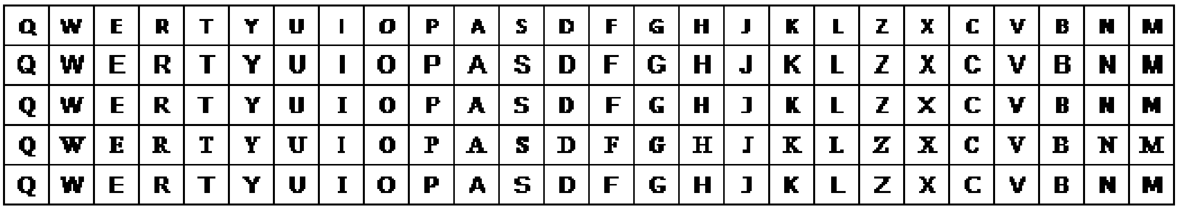

Fig. 2. Training and validating dataset
Rys. 2. Dane uczące i testowe

4. Experiments and Results

The experiments presented in the following chapters have been performed using dedicated software, developed by authors. The MFN software model and the Backpropagation method both have been written in pure C# language without usage of any external ANN-related libraries. As there are many ANN C# software libraries well-known to the authors (i.e.: FANN via C++ interop, NeuronDotNet, AForge.NET) the reason to prepare dedicated software was curiosity and willingness to ensure that 3rd party software and its optimization has no impact on the experiments and the results. The data presentation layer was also implemented in C# but includes Microsoft Excel cooperation library. The spreadsheet was chosen because it provides both numeric and visual representations of the data as well as delivers a variety of functions for data analysis.

Each MFN configuration has been trained 3 times with different, randomly generated starting weights. The same training datasets were applied to each MFN configuration to minimize the impact of the order of the dataset samples. The range of the MFN parameters investigated has been briefly presented in chapter 2. Regarding the number of hidden layers constituting the network, it may be estimated based on the nature of the problem [3] and feature extraction [4, 10, 16]. Authors of the paper approached the problem, assuming that most applied MFNs solving non-linear problems present one or two hidden layers [4, 11].

4.1. Number of Neurons when One Hidden Layer

The first investigation was performed to analyze the impact of the number of neurons [7] and amount of connections between neurons in neighbor layers, when single, hidden layer with fixed sigmoidal (eq. (1)) activation function was used. The β parameter of the sigmoidal function equal 1.5. The output was plain (non-encoded) 26 outputs, one per letter. The number of the neurons in the single, hidden layer was tested in {12, 25, 50, 75, 100, 150} and the number of connections between the neighborhood layer (connection ratio) in {50%, 60%, 70%, 80%, 90%, 100%}.Analyzing the results (juxtaposed in Table 1) one may conclude that the significantly lower number of neurons in the hidden layer (compared to the input layer) leads to low capacity of the MFN. Significantly lower performance may be observed when the number of the neurons in the hidden layer are equal 12 or 25 in case of the experiments (one may min that number of neurons in the input layer = 180).

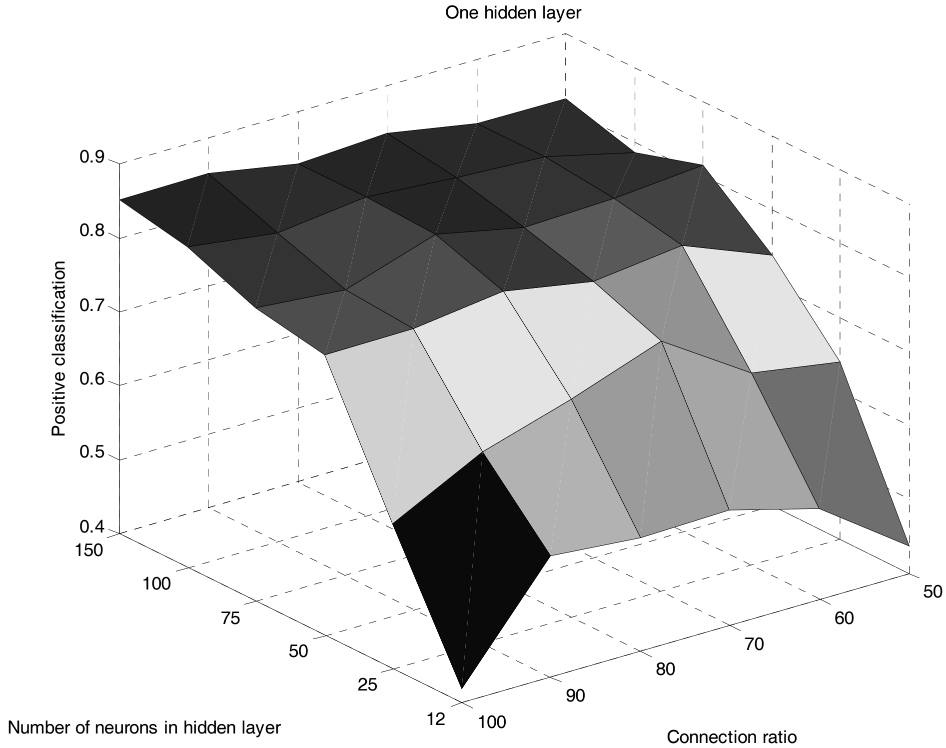

Fig. 3. One hidden layer network analysis
Rys. 3. Wyniki analizy dla jednej ukrytej warstwy

Table 1

Positive classification(one hidden layer)

Number of neurons in hidden layer	Connection ratio						Average	Std. dev.
	100%	90%	80%	70%	60%	50%		
12	0.420	0.564	0.554	0.556	0.523	0.436	0.509	0.059
25	0.597	0.659	0.695	0.739	0.661	0.641	0.665	0.044
50	0.780	0.780	0.795	0.774	0.787	0.738	0.776	0.018
75	0.798	0.787	0.826	0.800	0.805	0.815	0.805	0.012
100	0.834	0.818	0.831	0.823	0.815	0.785	0.818	0.016
150	0.851	0.852	0.831	0.836	0.815	0.813	0.833	0.015
Average	0.713	0.743	0.755	0.755	0.735	0.705		
Std. dev.	0.155	0.100	0.102	0.094	0.109	0.134		

Regarding the connection ratio between the neurons, it may be concluded that ratio between 50% and 100% has no significant impact on the performance, assuming those networks that presented low performance because of the low number of neurons in the hidden layer were abandoned. The Table 1 also contains an average probability of the correct recognition using one fixed parameter while the other changes and its standard deviation.

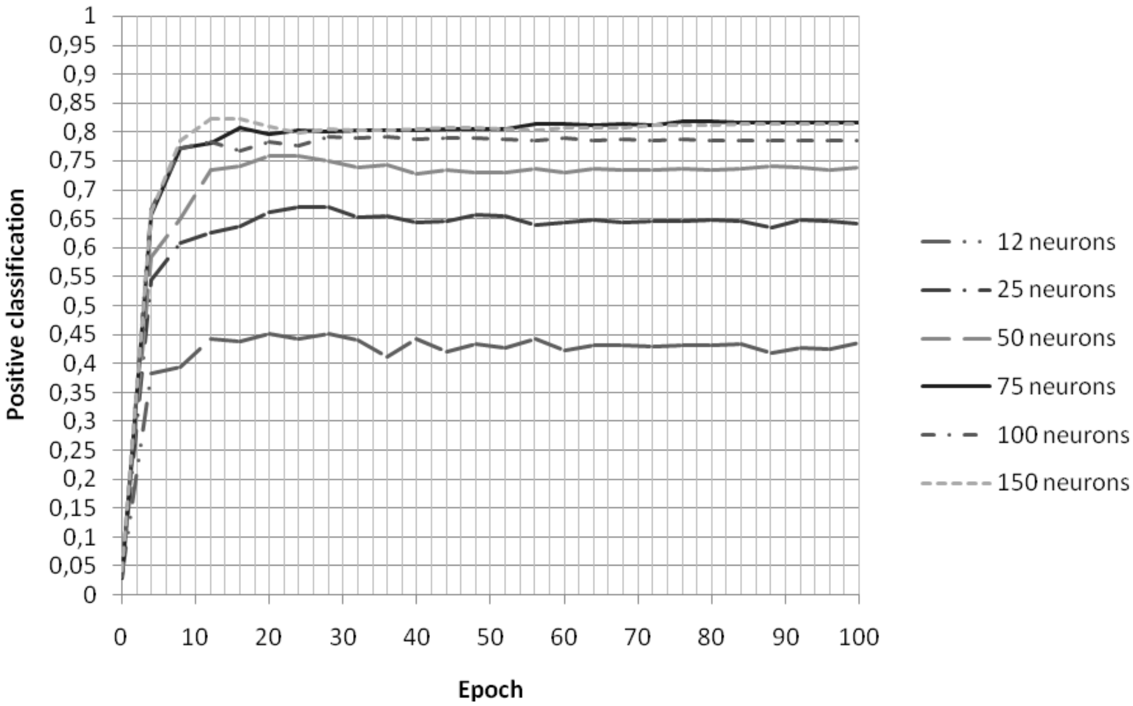

Fig. 4. Training process of the one hidden layer network with connection ratio 50%
Rys. 4. Proces uczenia sieci z jedną warstwą ukrytą, wypełnieniem 50%

Analyzing the results it is observable that the number of neurons within the hidden layer has greater impact on the overall network performance than its connection ratio (Fig. 3). This may be explained by the fact that many weights connection during the training stage receive a 0 value, thus they become inactive and then do not differ from the non-exiting once since the beginning of the training of the network. When bigger amount of neurons is present within the hidden layer, the better and more stable the network performs. The training process for the sample network regarding the number of neurons in hidden layer and 50% connection ratio is presented on the Fig. 4.

4.2. Optimal Number of Neurons in Two Hidden Layers

During this experiment, a network with two hidden layers were tested. The neuron activation function was fixed sigmoidal (eq. (1)) with β parameter equal 2.0.The connection ratio between the neighbor layers was fixed 100% (full connection network). The detailed results of the positive classification of the test data are presented in Table 2. It is worth to mention that comparing to the classification results of the single hidden layer (Table 1, Fig. 3), the double hidden layer networks perform worse in case of simple OCR (Fig. 5) that stays in contract with observation presented in Chester [3].

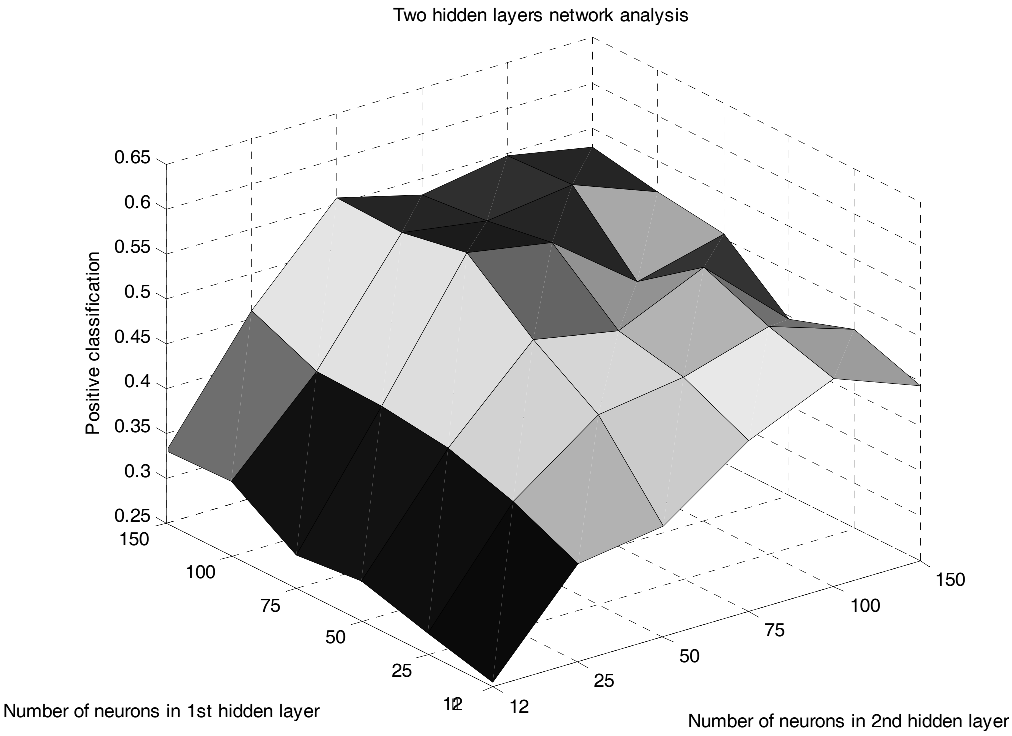

Fig. 5. Two hidden layers network analysis
Rys. 5. Wyniki analizy, zależność między liczbą neuronów w warstwach

Table 2

Positive classification (two hidden layers)

Number of neurons in 1st hidden layer	Number of neurons in 2nd hidden layer						Average	Std. dev.
	12	25	50	75	100	150		
12	0.256	0.359	0.374	0.441	0.482	0.446	0.393	0.074
25	0.274	0.392	0.461	0.474	0.503	0.472	0.429	0.077
50	0.295	0.415	0.508	0.490	0.533	0.446	0.448	0.079
75	0.287	0.426	0.569	0.551	0.479	0.505	0.470	0.094
100	0.333	0.428	0.554	0.539	0.551	0.515	0.487	0.081
150	0.331	0.459	0.556	0.531	0.546	0.528	0.492	0.078
Average	0.256	0.413	0.504	0.50	0.516	0.485		
Std. dev.	0.028	0.031	0.068	0.039	0.029	0.033		

The following experiment analyzed the impact of the network connection ratio vs. number of neurons in the second hidden layer, whereas first hidden layer contains a fixed number of neurons equal 150. The other experiment parameters remain unchanged comparing to the previous one.

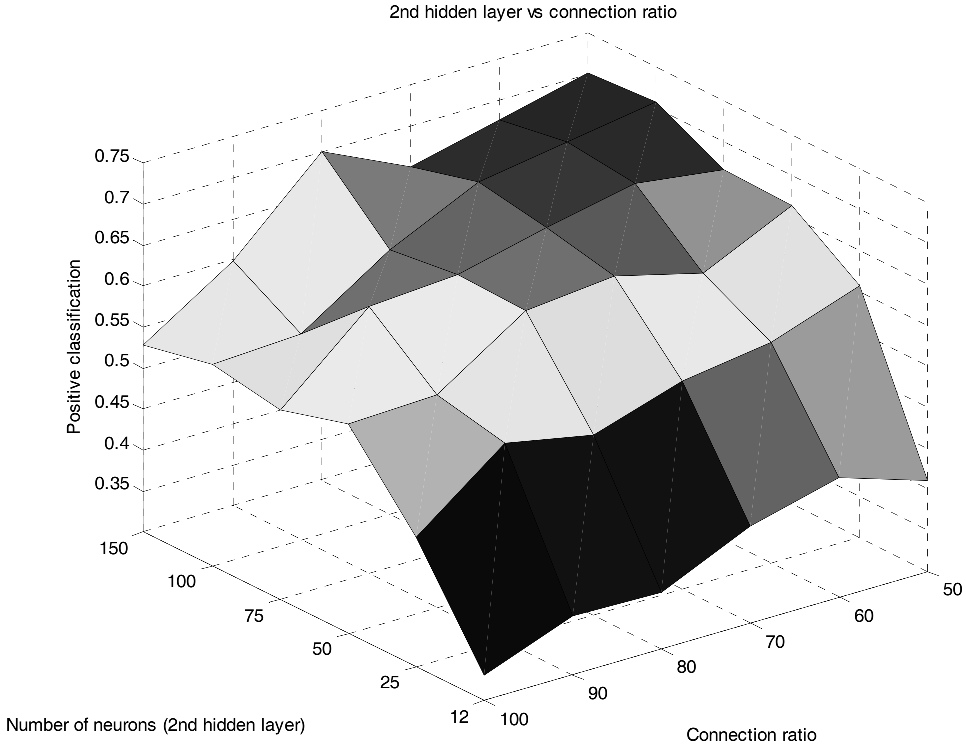

Fig. 6. Two hidden layers, connection ratio impact analysis
Rys. 6. Wpływ liczby połączeń na jakość klasyfikacji, dwie warstwy ukryte

Interestingly, in case of two hidden layers, the network connection ratio has an observable impact on the network performance, possibly as a result of multimodality of the objective function during training of the network when using the classical backpropagation method. The results are presented in the Table 3.

Table 3

Positive classification (2^{nd} fixed layer vs connection ratio)

Number of neurons in hidden layer	Connection ratio						Average	Std. dev.
	100%	90%	80%	70%	60%	50%		
12	0.331	0.372	0.369	0.418	0.446	0.410	0.391	0.038
25	0.459	0.541	0.520	0.554	0.569	0.608	0.542	0.046
50	0.556	0.559	0.631	0.641	0.613	0.664	0.611	0.040
75	0.531	0.626	0.633	0.659	0.682	0.667	0.633	0.050
100	0.546	0.551	0.623	0.674	0.690	0.708	0.632	0.064
150	0.528	0.600	0.700	0.651	0.677	0.702	0.643	0.062
Average	0.492	0.542	0.579	0.600	0.613	0.627		
Std. dev.	0.086	0.089	0.118	0.098	0.094	0.112		

Networks with two layers, higher number of neurons and lower number of neurons to neuron connections (connection ratio) present significantly better performance (Fig. 6). The sample network training process of the two layered networks with a first layer constant number of neurons equal 150 is presented on Fig. 7.

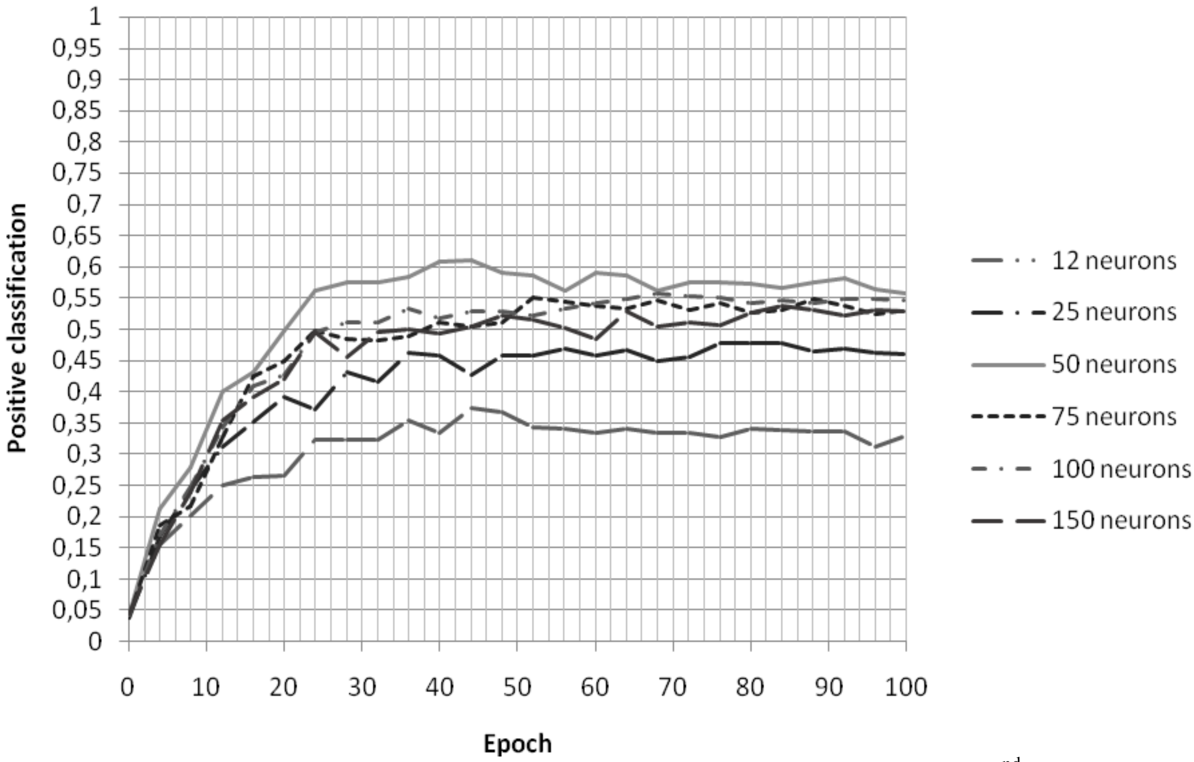

Fig. 7. Training process of the network while a variable number of neurons in 2nd hidden layer
Rys. 7. Proces uczenia sieci w funkcji liczby neuronów w drugiej warstwie ukrytej

4.3. Activation Function Optimization

The choice of the appropriate activation function is an important issue, having strong impact on the correctness and accurateness of the ANN [9]. For the activation function examination the MFN with a single hidden layer composed of 150 neurons and 90% connection ratio was chosen as one of the best performing in previous research (chapter 4.1). The output of the MFN is plain, non-encoded, as presented in previous experiments and described in chapter 2.The investigated functions are presented on eq. (1) to eq. (5).Each analyzed function contains one parameter, only Gaussian function contains two. The results of the Sine, Hyperbolic Tangent and Bipolar activation function are presented in Table 4. The results of the Sigmoidal function are juxtaposed in Table 5 and Gaussian in Table 6.

Table 4

Positive classification (Sine, Hyperbolic Tangent and Bipolar)

Activation function	β parameter					
	0.01	0.02	0.04	0.08	0.15	0.3
Sine	0.751	0.792	0.777	0.744	0.759	0.692
Hyperbolic Tangent	0.821	0.810	0.810	0.772	0.756	0.726
Bipolar	0.315	0.792	0.846	0.849	0.818	0.669

Table 5

Positive classification (Sigmoidal)

Activation function	β parameter					
	0.5	0.75	1.0	1.5	2.0	3.0
Sigmoidal	0.851	0.859	0.851	0.851	0.833	0.818

Table 6

Positive classification (Gaussian)

Activation function	σ parameter					
	4.0	4.5	5.0	5.5	6.0	7.0
Gauss μ=2	0.823	0.857	0.839	0.836	0.815	0.828
Gauss μ=3	0.043	0.038	0.092	0.836	0.851	0.831
Gauss μ=4	0.043	0.036	0.038	0.038	0.038	0.038

The observed results show that common approach using Sigmoidal and Hyperbolic Tangent functions is a good choice, providing reasonable results in a wide range of arguments while using Gaussian and Bipolar function requires a careful approach to its parameters as may lead to poor performance when inappropriate set of parameters is chosen. The training of the network using a Bipolar activation function with respect to the β parameter is presented on the Fig. 8 while using Sigmoidal on the Fig. 9.

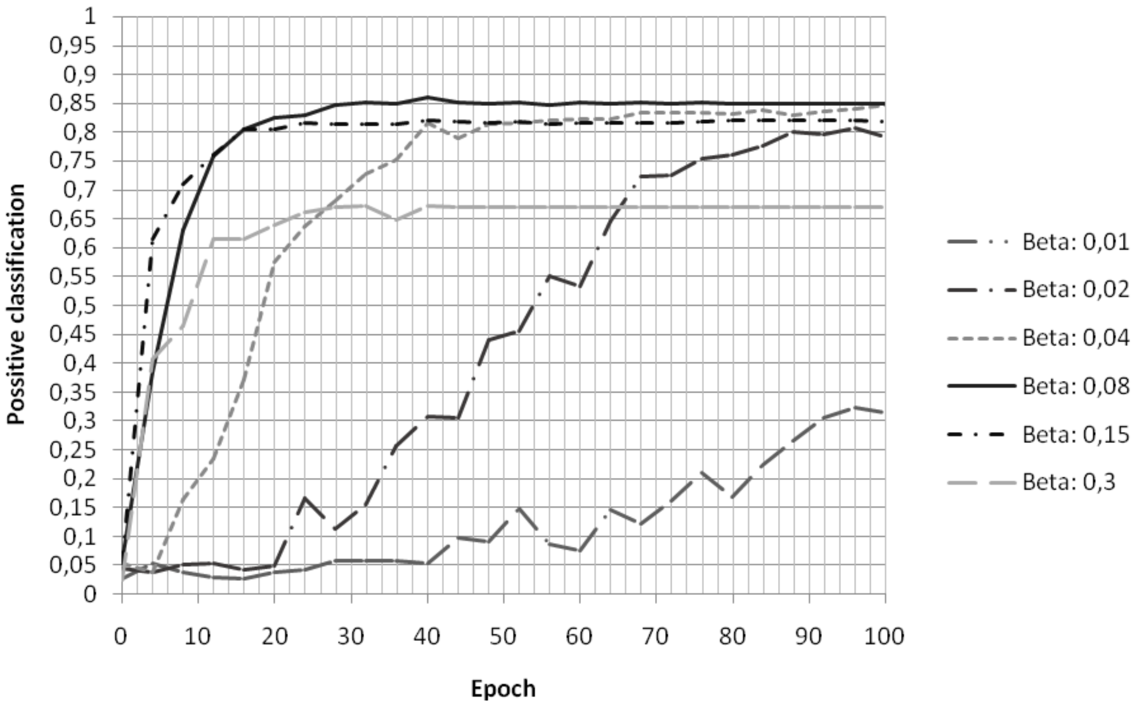

Fig. 8. Training process of the network with bipolar activation function, connection ratio 90%
Rys. 8. Proces uczenia sieci z bipolarną funkcją aktywacji i wypełnieniem 90%

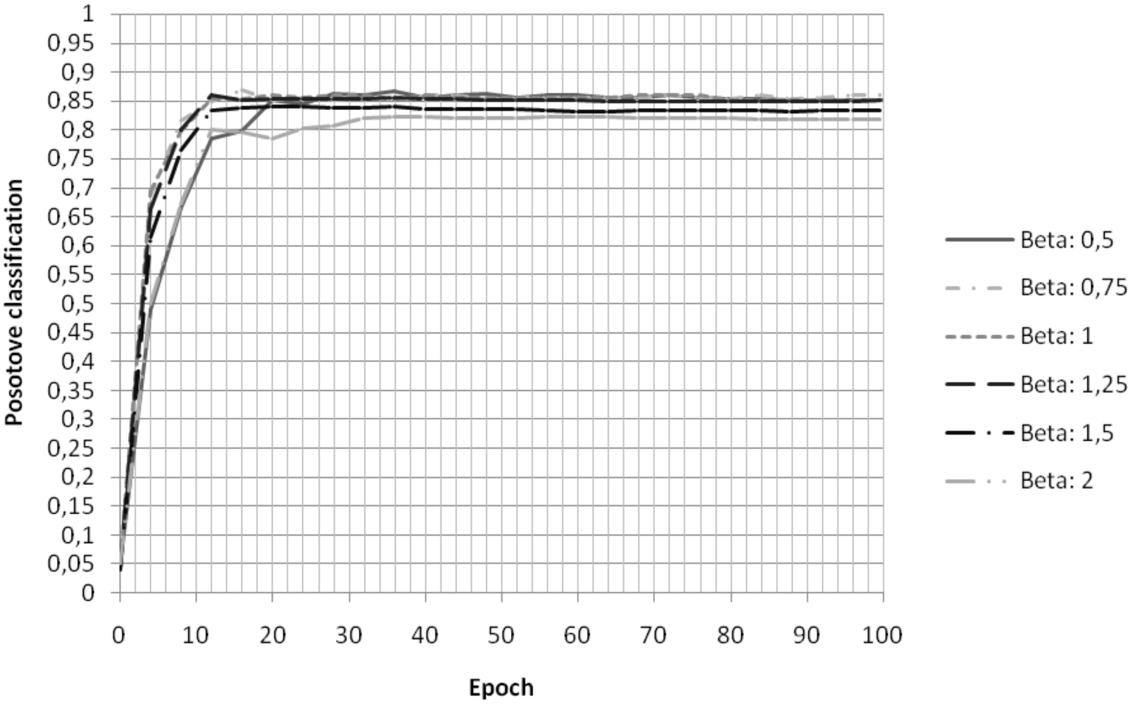

Fig. 9. Training process of the network with sigmoidal activation function, connection ratio 90%
Rys. 9. Proces uczenia sieci z sigmoidalną funkcją aktywacji i wypełnieniem 90%

4.4. Output data format

While it is obvious that in case of the classification of the symbols the number of outputs of the MFNis usually equal at least the number of symbols to be identified, the capacity of the output code may arise the question if deliberate coding of the output data. Limiting the number of outputs has a positive impact on the network training performance. The experiments were performed one hidden layer network, fixed connection ratio equal 90%, Sigmoidal activation function (β=1) and variable numbers of neurons in the hidden layer {75, 100, 150, 200}. Those parameters were chosen by the authors a'priori as providing reasonable results, according to the experimental results presented in chapters 4.1, 4.2 and 4.3. The output configurations investigated are presented in chapter 2.

Concluding the results it is observable that networks with plain (1 of N) coding performedbetter than those using data packing. More complex coding of the output data is used, the worse the network classification is observed when fixed structure.

The number of neurons in the hidden layer, as chosen to ensure appropriate network capacity, here between 75 and 200, has almost no impact on the resulting performance. The detailed results are presented in Table 7 and Fig. 10.

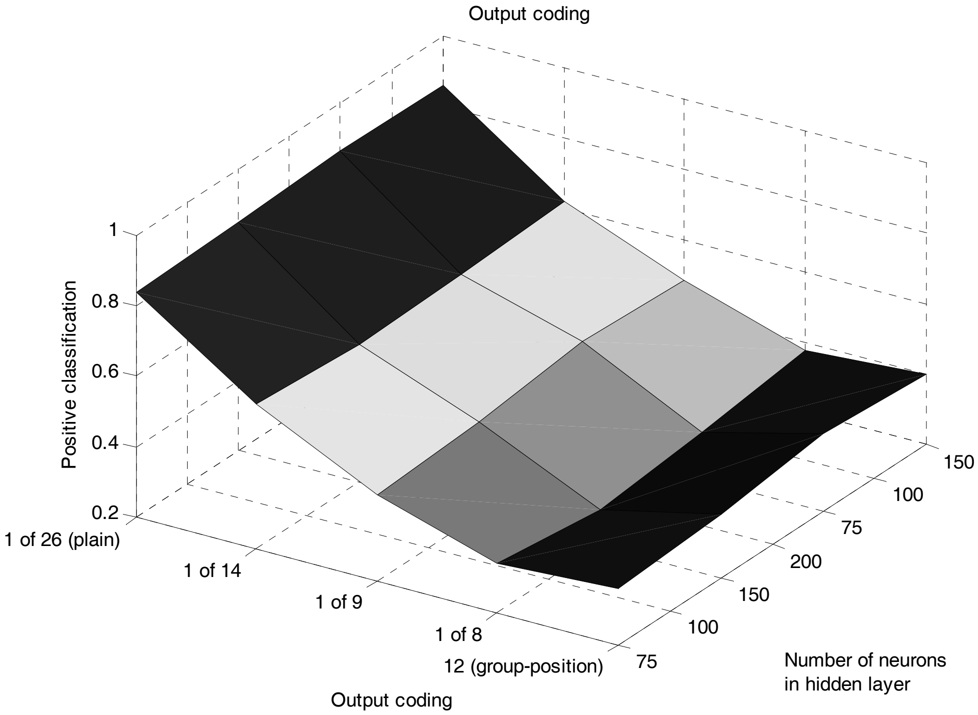

Fig. 10. Analysis of the impact of the output coding on classification quality
Rys. 10. Analiza wpływu kodowania wyjść na jakość klasyfikacji

Table 7

Positive classification (output coding)

Number of neurons in hidden layer	Output coding				
	1 of 26 (plain)	1 of 14	1 of 9	1 of 8	12 (group-position)
75	0.836	0.613	0.446	0.341	0.359
100	0.849	0.590	0.462	0.305	0.379
150	0.862	0.603	0.500	0.333	0.418
200	0.857	0.620	0.485	0.377	0.400

5. Conclusion

Presented analysis can be treated as a general approach when working with applied MFNs in basic structure. General observations conclude as in case of the simple pattern recognition (i.e. OCR) the MFN with just one hidden layer is a good choice. The output coding should be kept as plain as possible. The initial network connection ratio among neighbor neurons does not have to be 100%, assuming sufficient number of neurons in the hidden layer. Decreasing the number of the initial connections by few percent usually has no meaningful impact on the overall system quality, but indeed increases the training performance and computational complexity. Introducing second layer may lead to increased overlearning of the network, finally lowering the overall quality of the solution. Selecting the correct activation function and its parameters is state of the art and has a strong impact on the resulting quality and performance of the system. The inappropriate choice of the network parameters may lead to the low performance on letter recognition.

BIBLIOGRAPHY

1. Ahmmed S., Abdullah-Al-Mamun K., Islam M.: A novel algorithm for designing three layered artificial neural networks. Int. J. Soft Computing, Vol. 2, No. 3, 2007, p. 450÷458.
2. Armstrong A.: Google's Deep Learning - Speech Recognition. [@:] http://www.i-programmer.info/news/105-artificial-intelligence/4638-googles-deep-learning-speech-recognition.html, I Programmer, 2012.
3. Chester, D. L.: Why Two Hidden Layers are Better than One. IJCNN-90-WASH-DC, Lawrence Erlbaum, Vol. 1, 1990, p. 265÷268.
4. Choudhary A., Rishi R.: Improving the Character Recognition Efficiency of Feed Forward Bp Neural Network, Int. J. of Computer Science & Information Technology, Vol. 3, No. 1, 2011.
5. Emmerson M. D., Damper R. I.: Determining and improving the fault tolerance of multi layer perceptrons in a pattern-recognition application., IEEE Trans. Neural Netw., Vol. 4, 1993, p. 788÷793.
6. Hammerstrom D.: Neural networks at work. IEEE Spectrum, Vol. 30, No. 6, 1993.
7. Huang S.C., Huang Y.F.: Bounds on the number of hidden neurons in multilayer perceptrons. IEEE Trans. Neural. Netw., Vol. 2, 1991, p. 47÷55.
8. Henrique M., Lima L., Seborg E.: Model structure determination in neural network models, Chem. Eng. Sci., Vol. 55, 2000, p. 5457÷5469.
9. Kamruzzaman J., Aziz S. M.: A Note on Activation Function in Multilayer Feedforward Network, IEEE Xplore, 2002.
10. Koerich A. L., Sabourin R., Suen C. Y.: Large vocabulary off-line handwriting recognition: A survey. Pattern Analysis and Applications, Vol. 6, No. 2, 2003, p. 97÷121.
11. Kriesel D.: A brief introduction to neural networks, [@:] http://www.dkriesel.com/en/science/neural_networks, 2007.
12. Kröse B.,Van der Smagt P.: An introduction to neural networks. University of Amsterdam, 1993.
13. Reitermanova Z.: Feedforward Neural Networks – Architecture Optimization and Knowledge Extraction. Proc. WDS’08, Vol. 1, 2008, p. 159÷164.

14. Rojas R.: Neural Networks – A Systematic Introduction. Springer-Verlag, Berlin, New-York 1996.
15. Rutkowski L.: Metody i techniki sztucznej inteligencji: inteligencja obliczeniowa. Wydawnictwo Naukowe PWN, Warszawa 2005.
16. Steinherz T., Rivlin E., Intrator N.: Off-line cursive script word recognition – A survey. International Journal of Document Analysis and Recognition, Vol. 2, 1999, p. 90÷110.
17. Tadeusiewicz R.: Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#. Polska Akademia Umiejętności, 2007.
18. Zięba M.: Multistage Neural Networks: Multistage Neural Networks for Solving Pattern Recognition Problems. LAP LAMBERT Academic Publishing, 2010.

Omówienie

W publikacji autorzy zaprezentowali wpływ wielu parametrów jednokierunkowej, wielowarstwowej, sztucznej sieci neuronowej na jakość rozpoznawania wzorców, na przykładzie rozpoznawania dużych liter alfabetu łacińskiego. Analizie zostały poddane następujące elementy (szczegóły przedstawiono w rozdziale 2): struktura sieci neuronowej z jedną i dwiema warstwami ukrytymi, liczba neuronów w warstwach ukrytych, wybór funkcji aktywacji neuronu oraz parametrów tych funkcji, sposób kodowania wartości wyjściowych. Eksperymenty przeprowadzono w zamkniętym środowisku z ustalonymi pozostałymi parametrami sieci tak, aby zapewnić porównywalność wyników (parametry uczenia sieci przedstawiono w rozdziale 3, a zestaw danych uczących w rozdziale 3.1). Analiza wpływu liczby neuronów na jakość klasyfikacji, w przypadku jednej warstwy ukrytej w funkcji wypełnienia połączeń między sąsiadującymi warstwami, została przedstawiona w rozdziale 4.1, analiza wpływu liczby neuronów w każdej z warstw ukrytych oraz wpływu wypełnienia połączeń na jakość klasyfikacji, w funkcji liczby neuronów w drugiej warstwie ukrytej zostały przedstawione w rozdziale 4.2, analiza wpływu zastosowanej funkcji aktywacji neuronu oraz wpływu parametrów tej funkcji na jakość klasyfikacji sieci przedstawiono w rozdziale 4.3. W rozdziale 4.4 poruszono zagadnienie wpływu kodowania danych wyjściowych na jakość działania sieci neuronowej o strukturze wielowarstwowej z jedną warstwą ukrytą. W rozdziale 5 przedstawiono konkluzję, z której wynika, iż w zagadnieniu rozpoznawania tekstu (OCR) oraz wzorców wystarczająca wydaje się sieć z jedną warstwą ukrytą. Ponadto, nieznaczne zmniejszenie liczby początkowych połączeń między neuronami z sąsiednich warstw ma znikomy wpływ na działanie sieci, a jest pożądane w celu zmniejszenia złożoności obliczeniowej w trakcie uczenia sieci. Wskazano też, że istotnym elementem jest dobranie zarówno odpowiedniej funkcji aktywacji, jak i parametrów tej funkcji. Wyniki przeprowadzonych badań zostały przedstawione w formie omówienia, tabelarycznej oraz wykresów w poszczególnych rozdziałach. Dla wybranych przykładów zaprezentowano w formie graficznej postępy w uczeniu sieci.

Addresses

Piotr CZEKALSKI: Politechnika Śląska. Instytut Informatyki. ul. Akademicka 16
44-100 Gliwice, Polska. piotr.czekalski@polsl.pl.
Karol ŁYP: Politechnika Śląska. Instytut Informatyki. ul. Akademicka 16, 44-100 Gliwice
Polska. karol.lyp@gmail.com