Comparative Analysis of Methods for Determining Number of Hidden Neurons in Artificial Neural Network

Tijana Vujičić, Tripo Matijević, Jelena Ljucović, Adis Balota
Faculty of Information Technology
University Mediterranean
Vaka Đurovića bb, 81 000 Podgorica, Montenegro
{tijana.vujicic, tripo.matijevic, jelena.ljucovic, adis.balota}@unimediteran.net

Zoran Ševarac
Faculty of Organizational Sciences
University of Belgrade
Jove Ilića 154, 11000 Belgrade, Serbia
sevarac.zoran@fon.bg.ac.rs

Abstract*. Neurons in an artificial neural network are grouped in three layers: input, output and hidden layer. Determination of an optimal number of neurons in hidden layer is one of the major difficulties in the process of creating artificial neural network topology. The main goal of this paper is to explore and compare existing methods for determining number of hidden neurons. The research is conducted on two separate datasets with different number of input values and different number of training pairs.*

Keywords. artificial neural networks, hidden neurons, methods, test error, comparison

1 Introduction

Artificial intelligence is scientific field which has a goal to create machines that will be able to think like humans. Human brain consists of billions of interconnected neurons, a structure that is known as biological neural network. One of primary ideas of artificial intelligence is creation of artificial neural networks. In IT, term "artificial neural networks" is commonly used without prefix "artificial".

Neural networks are using mathematical models and human brain structure in order to develop strategy for data processing. They are able to acquire, preserve and use experiential knowledge. Neural network (NN) is able to learn from examples, which are composed of input values and expected outputs for those inputs. Once it establishes calculation rules (connection between inputs and outputs) it is capable to determine output for any input.

NN is composed of highly interconnected processing elements (neurons) that are grouped in the following layers: (Kriesel, 2011) (Heaton, 2005)

- input layer – layer of neurons that receive input from the user program
- output layer – layer of neurons that send data to the user program
- hidden layers – between the input layer and output layer there can be zero or more hidden layers

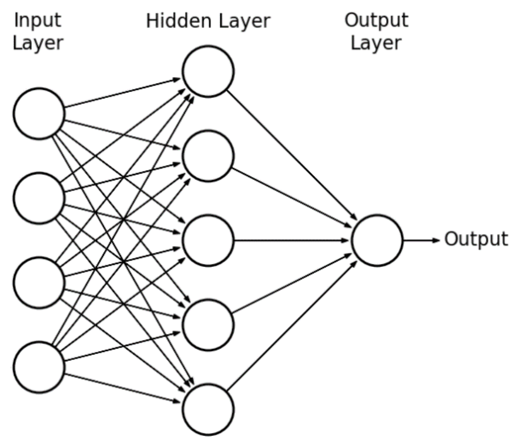

Figure 1. Neural network topology

For the majority of practical problems there is no need for more than one hidden layer. Three-layer neural networks have been applied to many problems in different areas.

It is very important to choose proper topology for neural network. Number of neurons in input layer is the same as number of input values, and number of neurons in output layer is the same as number of output values. Determination of an optimal number of neurons in hidden layer is one of the major difficulties facing researchers in this field.

If number of neurons in the hidden layer is too small, the network may not be powerful enough to meet the desired requirements. On other hand, large number of hidden neurons can cause very long training and recalling time.

This paper presents existing methods for determining number of hidden neurons, comparison of those methods, results that were achieved and conclusions that were derived.

In chapter Methods for determining number of hidden neurons authors give detail description of existing methods. The description of used datasets and methods, as well as comparison of the results, are presented in chapter Comparative analysis. Key parameter in the comparison is the number of iterations in training process and the mean square error in testing process. In chapter Conclusion authors give their view of the comparative analysis results and plans for future work.

2 Methods for determining number of hidden neurons

Over time a large number of methodologies to determine the number of hidden neurons have been proposed. Many books and articles that have been written on topic of neural networks are offering “rules of thumb” for choosing the correct number of neurons to use in the hidden layer (Heaton, 2005). Basic rules of thumb are:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be: (number of inputs + outputs) * (2/3)
- The number of hidden neurons should be less than twice the input layer size.

Researchers in this field were not satisfied with these rules and they proposed their own methods that are presented in Table 1. In order to clarify equations, symbols for number of certain neuron types in equations, in this paper, are standardized as:

- *Ni* – number of input neurons
- *No* – number of output neurons
- *Nh* – number of hidden neurons
- *Nt* – number of training pairs

Table 1. Methods for determining number of hidden neurons

Li, Chow and Yu method	
Description	For given input data set with *Ni* elements and its desired corresponding output data set, neural network with *Nh* hidden units can realize arbitrary function defined on the input data set (Li at al., 1995).
Equation	$Nh = \frac{\sqrt{1+8Ni}-1}{2}$
Tamura and Tateishi method	
Description	Three-layer neural network with *Ni*-1 hidden units can give any *Ni* input-target relations with a negligibly small error (Tamura & Tateishi, 1997).
Equation	$Nh = Ni - 1$
Xu and Chen	
Description	If the target function f is known, the best number of hidden layer nodes is: $Nh = \frac{1}{2} \cdot Cf \frac{Nt}{Ni \log Nt}$ The maximum of n has been proved to be *Nt*/*d*. In most practical cases the target function f is unknown, so they have found that when *Nt*/*d* is less than or close to 30, the optimal *Nh* most frequently occurs on its maximum, however, when *Nt*/*d* is greater than 30, the optimal *Nh* is close to the value of: (Xu & Chen, 2008) $Nh = \frac{1}{2} \cdot \frac{Nt}{d \log Nt}$
Equation	$Nh = \frac{1}{2} \cdot \frac{Nt}{Ni \log Nt}, \quad \frac{Nt}{Ni} > 30$ $Nh = \frac{Nt}{Ni}, \quad \frac{Nt}{Ni} \leq 30$
Shibata and Ikeda method	
Description	Authors tried to provide guideline to roughly estimate the number of hidden neurons and introduced equation that takes into account the number of input and output neurons, even though further adjustment is necessary for different problems or conditions (Shibata & Ikeda, 2009).
Equation	$Nh = \sqrt{Ni\,No}$
Hunter, Yu, Pukish III, Kolbusz and Wilamowski	

Description	Generalized solution for all Parity-N cases with N parity number is: $N = 2^n - 1$, where n is the total number of neurons in the network (Hunter at al., 2012).
Equation	$Nh = \log_2(Ni + 1) - No$
Sheela and Deepa	
Description	Researchers have tested 101 various criteria in order to propose a new method to determine the number of hidden neurons. The results showed that their proposed model improves the accuracy and minimizes the error (Sheela & Deepa, 2013).
Equation	$Nh = \frac{(4Ni^2 + 3)}{Ni^2 - 8}$

3 Comparative analysis

Comparison of described methods was conducted on two separate datasets. Data were obtained from monitoring station for lightning research on the mountain Lovćen in Montenegro that was set up within the LAMS project (LAMS, 2015). Datasets were chosen to be different from each other, considering size of input data vector and total number of records.

First dataset has 6 input values, 1 output value and 1728 records. Second dataset has 34 input values, 1 output value and 366 records. Both datasets were divided into training and testing datasets: 70% of records were used for training and 30% were used for testing. Basic data about both datasets are presented in Table 2.

Table 2. Basic data about used datasets

Dataset	DS1	DS2
Input values	6	34
Output values	1	1
Total records	1728	366
Records for training	1228	256
Records for testing	500	110

Fields in those datasets were of different data types: integers, decimal numbers and text. All these values were normalized before neural network training. Main goal of training is to minimize the mean squared error (MSE) on the training set. Maximal error is determined before training, and every network that has training error less than or equal to that maximum is considered as successfully trained. After training process neural network is tested with testing dataset and testing mean squared error is calculated and used for comparison.

Application for creating, training and testing neural networks with different topologies was implemented in Java programming language using Neuroph Java neural network framework (Sevarac, 2008). During neural network training, maximal error was set to 0.003, and maximal number of iterations was set to 50000. Separate neural network was created, trained and tested for each of described methods. Network was considered as successfully trained if error value below maximal error was achieved before reaching maximal number of iterations. Results are shown in Table 3 and Table 4. Successfully trained networks are shown first, and results are sorted according to test error, from smallest to largest.

Table 3. Results for the first dataset

DS1				
Method	*Nh*	Train error	Iterations	Test error
Sheela and Deepa	5	0.003	2198	0.0843
Li et al.	3	0.003	21335	0.0962
Tamura and Tateishi	5	0.003	2076	0.1102
Xu and Chen	5	0.003	2849	0.1195
Rule of Thumb	5	0.003	1972	0.1459
Shibata and Ikeda	2	0.005	50000	0.0366
Hunter et al.	2	0.005	50000	0.0502

Table 4. Results for the second dataset

DS2				
Method	*Nh*	Train error	Iterations	Test error
Sheela and Deepa	4	0.003	32	0.0087
Hunter et al.	4	0.003	40	0.0105
Shibata and Ikeda	6	0.003	37	0.0105
Li et al.	8	0.003	49	0.0122
Rule of thumb	23	0.003	106	0.0126

Xu and Chen	7	0.003	36	0.0144
Tamura and Tateishi	33	0.05	50000	0.1617

From results that are shown in Tables 3 and 4 it is easy to conclude that Sheela and Deepa method gave neural network that was trained in small number of iterations and had the smallest test error. Also Li et al. method and general Rule of Thumb gave good results for both datasets. Tamura and Tateishi method gave good results for first dataset, but neural network for second dataset was not successfully trained. On the other hand, Hunter et al. and Shibata and Ikeda method gave really good results for dataset with larger number of input values, but for dataset with 6 input values both methods gave neural networks that have reached the maximal number of iterations during training process. It can also be noticed that rule of thumb, that is usually suggested in literature for neural networks, occupies the fifth place in both cases, not being the best, but not being the worst either.

Network topologies that gave the best results are presented on fig. 2 and fig. 3. Neuroph studio tool was used for visualization of NN topologies (Sevarac, 2008).

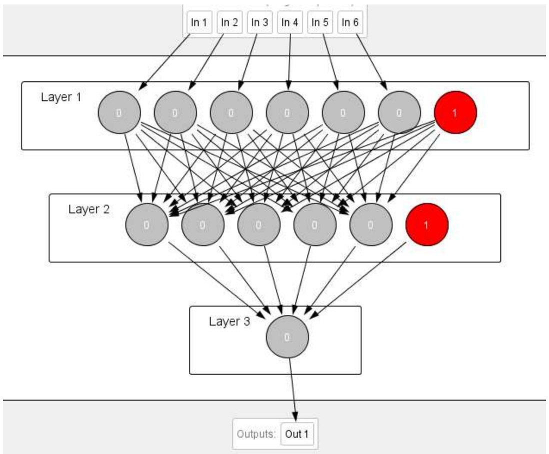

Figure 2. Network topology that gave best results for first dataset

4 Conclusion

In this paper authors gave review of existing methods for determining number of hidden neurons in artificial neural networks and comparative analysis of those methods. Comparison was conducted on two separate datasets with different number of input values and different number of training pairs.

Based on obtained results, it can be concluded that each method has different results for different datasets. Methods that gave good results for neural networks with smaller number of input neurons did not show comparably good results for networks with larger number of input neurons, and vice versa. Number of hidden neurons that are calculated using these methods are very good starting points that should be considered during the creation of network topology.

The best way to choose final topology of neural network for specific problem is to train and test network with different number of hidden neurons, each calculated by one of the most used methods. When different network configurations are coded, trained and tested, it is easy to choose one that gives the best results.

For future work it is planned to use results and application described in this paper as basis for intelligent system that will be able to predict parameters of lightning on the mountain Lovćen for the desired time period. The core of this system will be artificial neural network that will be trained on data obtained from LAMS monitoring station for the year 2015 and data on weather conditions for that period, that will be provided by Institute of Hydrometeorology and Seismology of Montenegro.

Acknowledgments

This paper has been supported by LAMS (Lightning Activity Monitoring System) project. Data that were used in this paper were obtained from monitoring station for lightning research on the mountain Lovćen in Montenegro.

.

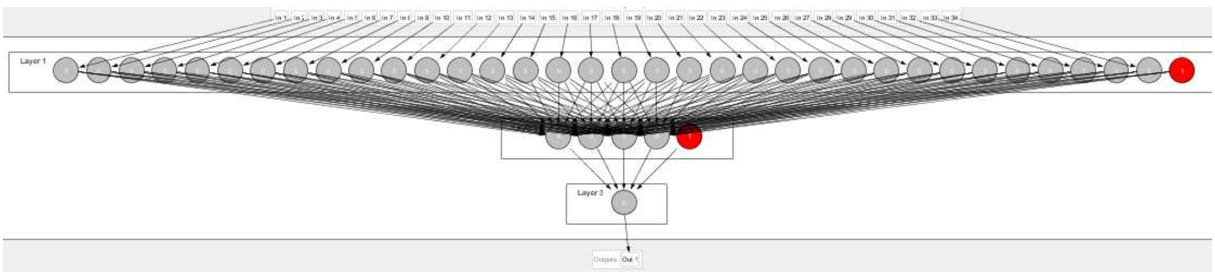

Figure 3. Network topology that gave the best results for second dataset

References

Heaton, J. (2005). *Introduction to Neural Networks with Java*, Heaton Research Inc

Hunter, D., Yu, H. , Pukish III, M. S., Kolbusz, J., Wilamowski, B. M. (2012). Selection of proper neural network sizes and architectures: a comparative study, *Proceedings of the IEEE Transactions on Industrial Informatics* (pp. 228 – 240)

Kriesel, D. (2011). A Brief Introduction to Neural Networks, Retrieved from http://www.dkriesel.com/en/science/neural_networks, 2011

LAMS project, http://www.lams-project.me/

Li, J. Y., Chow, T. W. S., Yu, Y. L. (1995). Estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network, *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1229 - 1233)

Sevarac Zoran, Neuroph Java neural network framework (2008). Retrieved from http://neuroph.sourceforge.net/

Sheela, K., Deepa, S. N. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks, Mathematical Problems in Engineering

Shibata K., Ikeda, Y. (2009). Effect of number of hidden neurons on learning in large-scale layered neural networks," *Proceedings of the ICROS-SICE International Joint Conference* (pp. 5008 – 5013)

Tamura, S., Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network: four layers versus three, *Proceedings of the IEEE Transactions on Neural Networks* (pp. 251 - 255)

Xu, S., Chen, L. (2008). A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining, *Proceedings of the 5th International Conference on Information Technology and Applications* (pp. 683 - 686)