Determining Neural Network Architecture Using Data Mining Techniques

Mohamed Lafif Tej, Stefan Holban
Politehnica University of Timisoara, 300223 Timisoara, Romania
afiftej@gmail.com, stefan.holban@cs.upt.ro

***Abstract*— This paper presents techniques used to determine the optimal neural network architecture using pattern recognition and data mining. Clustering techniques highlight a number of common characteristics of input forms, which are classified into groups based on a given criterion. In the proposed method, the number of groups obtained using clustering techniques on the training data of a neural network represents the main factor for determining the optimal number of hidden layers for a multi-layer neural network. Use of this method allows the design of an optimal neural network to be unsupervised and will decrease its build time.**

Keywords—Artificial intelligence, clustering methods, data mining, multi-layer neural network, pattern recognition.

I. INTRODUCTION

Neural networks have complex structures because there is no available analytical method for determining the optimal number of hidden layers for a multi-layer neural network [1-4].

Among the methods currently used to determine the optimal neural network architecture is to choose multiple networks with different architectures. These networks are trained on a common dataset until they meet the performance requirements, and then the neural network architecture is chosen that has the minimum number of hidden layers and neurons, optionally taking into account the training time [5]. The disadvantages of the method are that the results depend largely on the structures originally selected and that it is time-consuming.

Usually, if the complexity of the problem increases, then the neural network [6] will become more sophisticated because the number of hidden layers and/or the number of nodes (neurons) in these layers will increase as well [7]. One advantage of using multiple hidden layers is improved predictive ability of the neural network. However, the training time increases (in some cases exponentially) owing to the higher complexity of the neural network.

This paper proposes an original method for determining the optimal neural network architecture and, optionally, number of hidden-layer neurons using clustering techniques that are part of the broader field of artificial intelligence known as pattern recognition.

II. CLUSTERING TECHNIQUE USED

In the method described in this paper, the clustering algorithm used is Kruskal's algorithm, which is based on a data analysis tree that minimizes distances. A minimum spanning tree is a spanning tree having a weight less than or equal to the weight of every other spanning tree [8-12]. After it is constructed, clusters are obtained by cutting all segments whose length is greater than a reference value chosen by the user, called the reference distance.

Fig. 1 illustrates the grouping technique based on Kruskal's algorithm. It is observed that the five groups of elements (a, b, c, d, e), (h, i, j, k, m, n), (f), (g), and (l) are obtained according to the chosen reference distance.

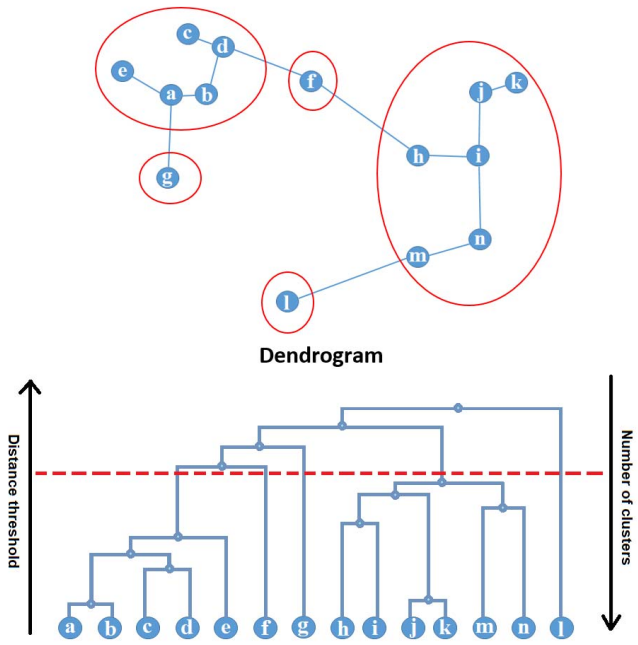


Fig. 1. Clustering technique that uses Kruskal's algorithm

978-1-5386-1495-2/18/$31.00 ©2018 IEEE

III. Datasets Used in Experiments to Validate Theoretical Results

In this paper, six datasets are presented and used in analyses to experimentally validate the theoretical results. The experimental data were deliberately chosen from a variety of fields of activity to ensure a more general analysis of the characteristics of the proposed method. The datasets used are as follows:

The Landsat satellite images dataset [13] was generated from Landsat Multispectral Scanner image data. Most of the data are in binary form and are distributed on magnetic tapes.

The dataset consists of the multispectral values of pixels in 3 × 3 neighborhoods in a satellite image and the classification associated with the central pixel in each neighborhood. The aim is to predict this classification given the multispectral values. In the sample dataset, the class of each pixel is coded as a number.

The Landsat satellite dataset is one of many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions, including multispectral and radar data and maps indicating topography and land use, is expected to assume vital importance with the onset of an era characterized by integrative approaches to remote sensing.

The Landsat satellite images dataset contains 4435 training set records with 36 attributes (4 spectral bands × 9 pixels in each neighborhood).

Another dataset pertains to the analysis of an ECG signal [14]. An ECG signal has three main components: the P wave, QRS complex, and T wave. The results of the analysis of these components together with other clinical data are used in the diagnosis of various cardiac disorders. Apart from the information obtained from the analysis of the components of an ECG signal for accurate diagnosis, there are other parameters that need to be calculated, including heart rate, duration of rate measurement, consistency of rate (constant or oscillating), the number of P waves in the cardiac cycle, and assessments of intervals and segments. With this information, a neural network can be trained to recognize most common cardiovascular diseases.

The last dataset is of a sonar signal [15]; the signal was sampled to obtain a vector form consisting of 60 features. Each pattern is a set of 60 numbers ranging from 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period. The integration aperture for higher frequencies occurs later in time, since these frequencies are transmitted later during the chirp.

The label associated with each record contains the letter "R" if the object is a rock or "M" if it is a mine (metal cylinder). The numbers in the labels are in increasing order of aspect angle, but they do not encode the angle directly. The dataset contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock.

IV. Determination of the Number of Hidden Layers Using Clustering Techniques

A. Optimization Problem

There are currently no valid general-purpose theories for choosing the number of hidden layers and the number of neurons in the design of a neural network. We are far from having a viable theory for determining the architecture of a neural network that depends on the complexity of the problem to be solved. Given these circumstances, the choice is made on the basis of designer experience and exhaustive simulations. Experiments performed are designed to address specific issues, and they provide results only for certain conditions.

Most designers find it more convenient to build multiple neural networks with simple and interconnected architectures instead of a single network that has a complex design. This requires decomposing complex problems into simpler ones if possible. This approach is time-consuming and can lead to reduced accuracy in determining the optimal architecture. In addition, poor choices result in longer training and testing times.

Here are the most widely used methods to determine the number of hidden neurons:

- Rules of Thumb: many researchers use numerous thumb rules such as the number of hidden neurons should be between the size of the input and output layers. The number of hidden neurons should be: (number of inputs + outputs) * (2/3). The number of hidden neurons should be less than twice the number of input layer neurons.
- Trial and Error: this approach does not yield good results except by accident, sometimes called exhaustive search.
- Exhaustive Search: This makes searching through all possible topologies and then select the one with the least generalization error. The disadvantage of this method is time-consuming.
- Growing Algorithms: this method makes searching through all possible topologies and then select the one with the least generalization error. Search in this method stops if the generalization error does not have remarkable change unlike exhaustive search.
- Pruning Algorithms: this method tries to train an oversized network, and then determines the relative importance of weights by analyzing them. This method prunes the weights with least importance and then repeats the task. The disadvantage of this method is that the analysis of weights is time-consuming.

B. Proposed method for optimization

The idea behind the proposed method for determining the optimal partition starts from the training items for a multi-layer neural network. It is assumed that the items in the training set can be grouped together using conventional methods of pattern recognition [16-18] according to a specific criterion [19-22]. In this way, a number of homogeneous classes will be obtained,

as well as a number of items that could not be associated with any group of elements. The number of classes obtained when the process is stable will be taken as the number of hidden layers of the multi-layer neural network [23-24].

Fig. 2 illustrates this method for determining the optimal neural network architecture.

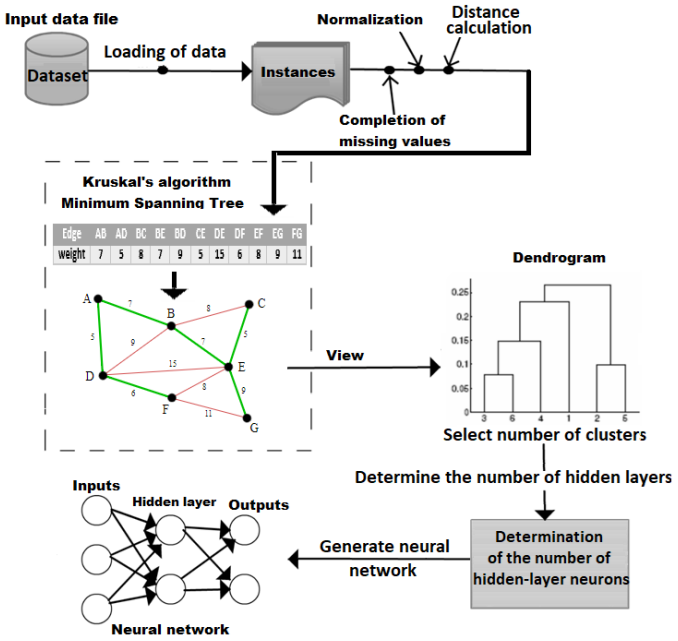


Fig. 2. Proposed method for determining the optimal neural network architecture

The method is divided into two phases:

- The first phase implements Kruskal's minimum spanning tree algorithm, thus obtaining a dendrogram that will help determine the number of hidden layers of the neural network.
- The second phase helps determine the number of neurons in the hidden layers.

In the first phase, clustering based on Kruskal's minimum spanning tree algorithm, a cluster analysis is performed on the training dataset to classify the information into groups. Clustering represents the process of grouping objects based only on the information describing the objects and the relations between them. The purpose of grouping is to form new groups that contain similar objects and at the same time are different from the objects in the other groups. A higher similarity within groups and a larger difference between groups indicates a better clustering process and thus a higher accuracy for the number of hidden layers of the neural network.

The second phase of the method offers the possibility of determining the number of neurons in the hidden layers based on various empirical formulas. The calculated number of neurons will be divided equally by the number of hidden layers.

C. Number of hidden layers using clustering technique

In the method proposed in this paper, the clustering technique described above is used to determine the number of hidden layers for a multi-layer neural network.

The number of hidden layers is equal to the number of groups obtained by the classification of items analyzed when the following two conditions are met:

- At least 90% of the items in the dataset used to train the neural network are grouped. It is believed that at this point, there are groups representative of the studied forms and the result can be extrapolated to the entire dataset.
- Increasing the reference distance does not change the number of groups. This indicates that the grouping has stabilized. Extreme cases are excluded, such as a very short distance, for which each element represents a group, or a relatively large distance, for which all elements are placed in a single group.

The number of clusters thereby obtained will be considered the optimal number of hidden layers for the neural network. One of the objectives of this study was to ascertain whether the number of clusters when these two conditions are met is equal to the optimal number of hidden layers

The following parameters are retained from this phase: minimum, maximum, and average distance, and the percentages of input formats that are below and above the average distance. These parameters are used in the following step, dimensioning the neural network.

V. Results Obtained Using Clustering Technique

A detailed analysis is presented here for the Landsat satellite dataset. Additionally, the results for the other five datasets are presented.

Table I shows the grouping results for the Landsat satellite dataset.

TABLE I. Data Obtained with Kruskal's Algorithm on Landsat Satellite Dataset

Reference Distance	Number of Clusters	Number of Items Grouped (%)
75	2	90
67	2	80
58	2	70
50	3	59
35	4	41
32	5	38
30	6	36

Fig. 3 shows the relationship between the reference distance and the number of groups obtained for the Landsat satellite dataset, including the percentage of items that were clustered. From the figure, it can be noted that an increase in the value of the reference distance will cause a decrease in the number of groups and an increase in the percentage of clustered elements obtained. For a larger value of reference distance, all items input will be clustered into one group.

Therefore, it can be concluded from Table I that an analysis of the Landsat satellite dataset should use a neural network having two hidden layers.

The importance of each parameter in determining the number of hidden neurons for a multi-layer neural network will be described in Section VI.

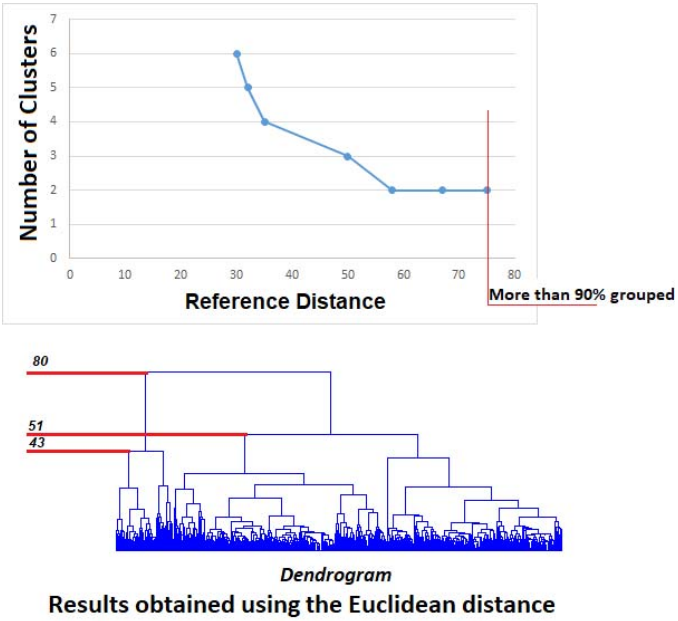


Fig. 3. Relationship between the reference distance and the number of groups obtained, and the corresponding dendrogram obtained from the analysis of the Landsat satellite dataset

In the same manner, analyses were conducted of the other five datasets, which are the P wave, QRS complex, T wave, ECG signal, and sonar signal datasets. Following this method, it was concluded that P-wave and sonar signals should use a neural network with two hidden layers, ECG and QRS should use a neural network with one hidden layer, T wave should use a neural network with three hidden layers, and to classify a Landsat satellite dataset, a neural network with two hidden layers is needed.

VI. Number of Neurons in Hidden Layers

Simplistically, the number of neurons in the input layer will be equal to the number of features that compose the training records [25], and for the output layer, the number of neurons must fit the correct representation of the neural network response [26].

Regarding the neurons in the hidden layers, however, there is no universally accepted method of calculating their numbers; these are determined experimentally or by using heuristic rules [27]. Therefore, simulations of the neural network are needed to enable changes in the number of neurons in each layer, changes in the number of layers, the activation function, and other adjustments. The number of neurons in each hidden layer must be large enough to generate configuration decision regions sufficiently complex to learn a given problem [28]. If the number of neurons is too high, the number of connections will be very high as well, and there is a risk that the weights of these connections may not be correctly calculated using the available training examples. The neural network may generate noise, or overlearning may result (in which the network tends to memorize the problem, leading to a weak generalization ability) [28]. If the number of neurons in the hidden layers is too small, the neural network cannot learn all the information present in the training dataset, and this decreases fault tolerance by reducing network redundancy [29].

In some papers, various empirical formulas for calculating the number of hidden layers and/or the number of hidden neurons are proposed. To determine the number of neurons in the hidden layers of the neural network, one of the following formulas may be used:

N_h is the number of neurons in the hidden layers.

N_i is the number of neurons in the input layer.

N_o is the number of neurons in the output layer.

N_{fh} is the number of neurons in the first hidden layers.

N_t is the number of training items.

$$(N_h) = (1/2)((N_i) + (N_o)) \quad (1)$$

$$(N_h) = \mathrm{SQRT}((1/2)((N_i) + (N_o))) \quad (2)$$

In other studies, this formula is proposed for calculating the number of hidden layers:

$$(N_h) = ((N_i) + (N_o))\times 2/3 \quad (3)$$

The optimal architecture of a neural network cannot be determined from the number of inputs and outputs alone because it depends critically on the training set size and the complexity of the functions that it needs to learn [30]. This empirical formula is used to determine the number of neurons in the first hidden layer:

$$(N_{fh}) = (N_t)/(10((N_i) + (N_o))) \quad (4)$$

If there are multiple (two or three) hidden layers, each usually has between 1/3 and 2/3 of the number of neurons in the previous hidden layer. In other studies, it is recommended that the number of neurons be the same in each hidden layer or that each layer have 1/2 of the number in the previous layer.

The number of neurons in a hidden layer can be calculated with this formula:

$$(N_h) = ((N_t) - (N_o))/(N_i) + (N_o) + 1 \quad (5)$$

NeuroShell ver. 2.0, a program for simulating and training neural networks, calculates the number of neurons in the hidden layers with this formula [31]:

$$(N_h) = 1/2((N_i) + (N_o)) + \mathrm{SQRT}(N_t) \quad (6)$$

In a case having multiple hidden layers, the number of neurons calculated with (6) will be divided equally by the number of hidden layers.

Determination of the number of neurons is performed using one of the formulas (1)–(6).

VII. Experimental Results

An analysis of the datasets used by a neural network was carried out until the optimal multi-layer neural network architecture can be defined. Each dataset was tested with different multi-layer neural network architectures using supervised learning methods, in particular back-propagation learning algorithms [32-35].

A. Experiment for optimal number of hidden layers

Simulation and training of neural networks were done using the simulation program NeuroShell ver. 2.0 (Ward Systems Group, Inc.) [31], which is considered one of the most complete and advanced simulators for neural networks.

For data analysis, three different types of neural network architecture were used, with one, two, and three hidden layers. Fig. 4 shows the structures of the three types of neural network architecture used.

The Landsat satellite dataset will be taken as the first example to demonstrate the differences between the error results for each neural network architecture, where the error is defined as the difference between actual and desired output values.

Fig. 5 presents a comparison of the errors obtained by different neural network architectures for the Landsat satellite dataset.

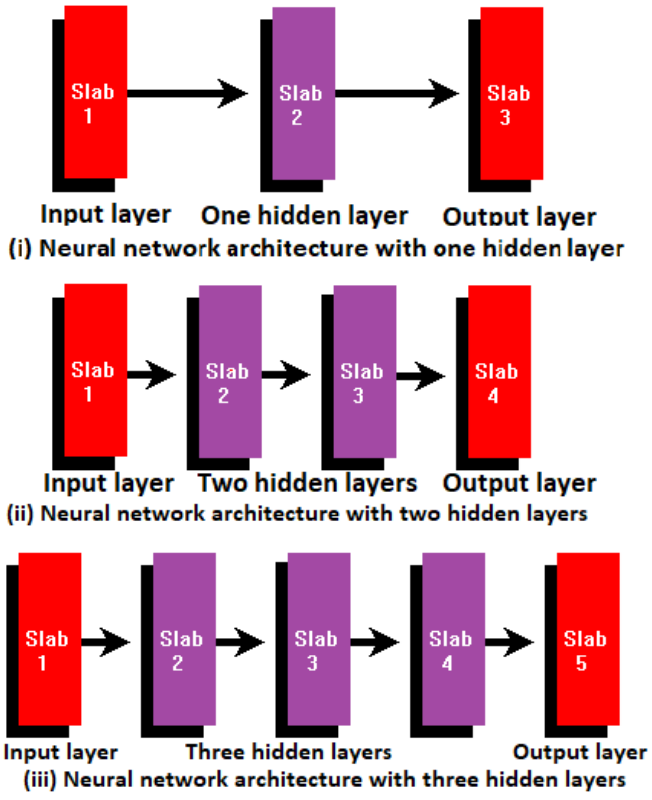
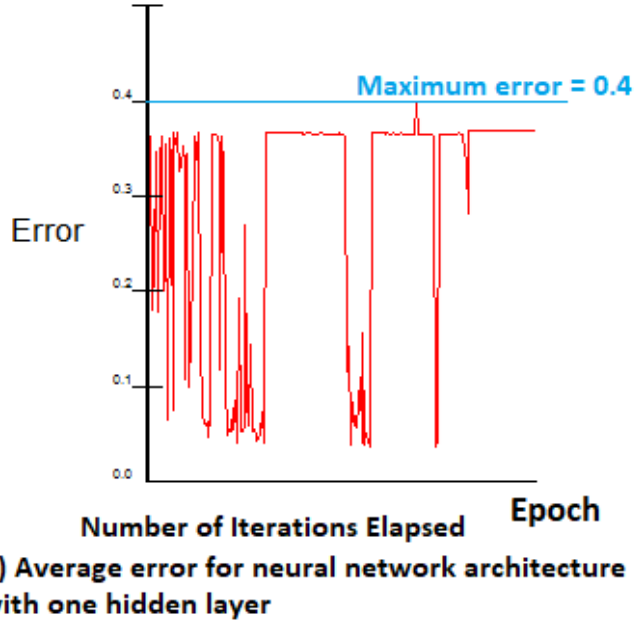
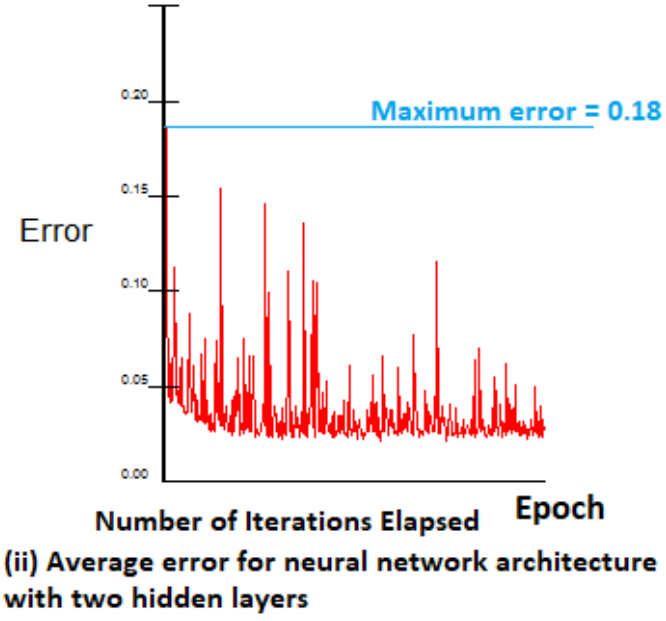


Fig. 4. The three types of neural network architecture used in the data analysis

Notice that the results of the neural network with one hidden layer has the largest number of errors greater than 0.35, and the largest error from the network with two hidden layers has a value less than 0.18. The network with three hidden layers has two errors greater than 0.30.

These experimental results comparing the errors obtained by different neural network architectures for the Landsat satellite dataset lead us to conclude that a network with two hidden layers has fewer errors than networks with one or three hidden layers, thus confirming the results presented previously that were obtained by the clustering program.





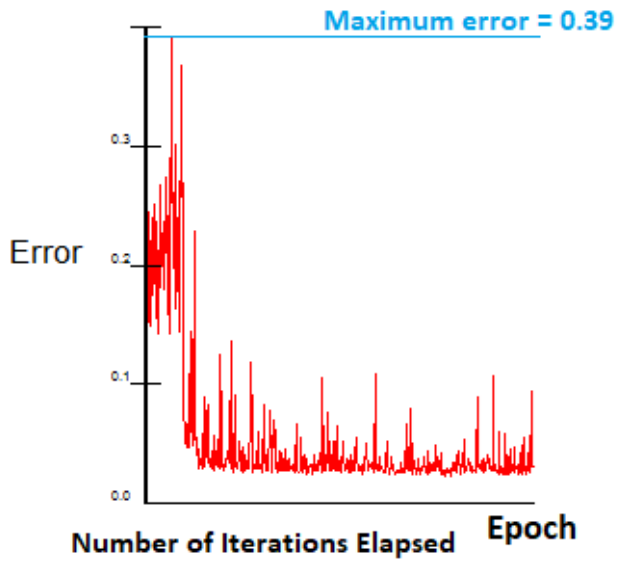
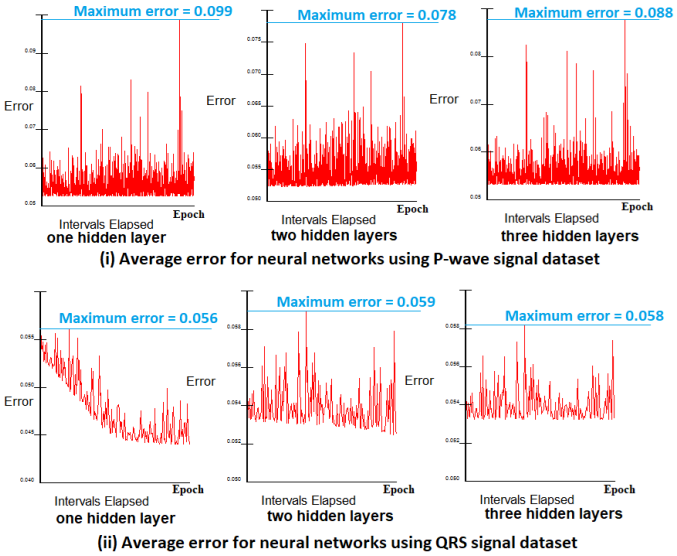


Fig. 5. Relative errors for the Landsat satellite dataset obtained by neural network architectures having different numbers of hidden layers

The same analysis was conducted for the other five datasets, with the results as shown in Fig. 6. From these results, it can be concluded that the optimal neural network architecture for diagnosis using the ECG signal dataset is a neural network with one hidden layers, one hidden layer for the QRS signal dataset, and three hidden layers for the T-wave signal dataset. Two hidden layers are optimal for the sonar signal dataset and for the P-wave signal dataset too. These results confirm those presented previously that were obtained by the clustering program.

Table II shows a summary of these results for the six datasets compared with those obtained by the clustering technique presented previously.



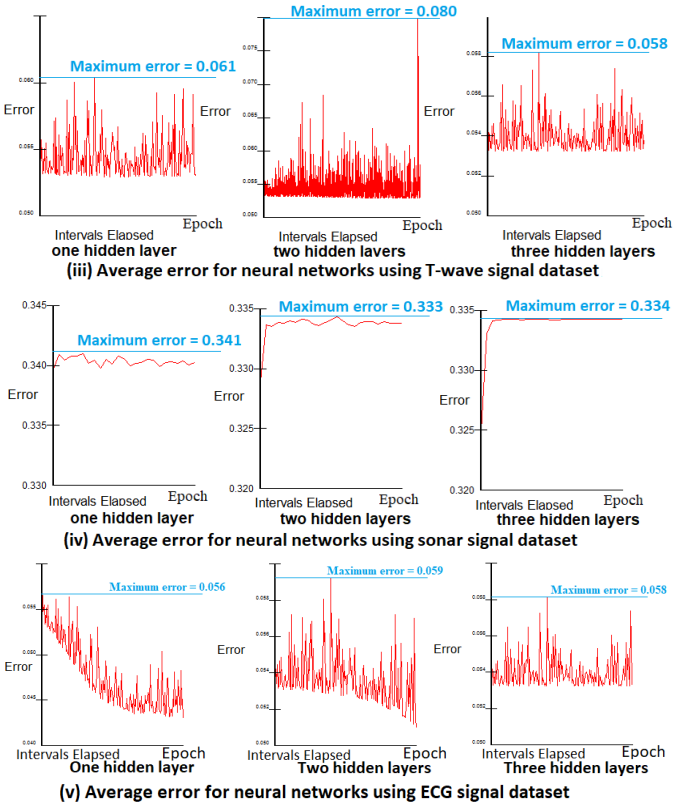


Fig. 6. Relative errors for the P-wave, QRS, T-wave, sonar, and ECG signal datasets obtained by neural network architectures having different numbers of hidden layers

TABLE II. COMPARISON OF NUMBER OF CLUSTERS OBTAINED BY CLUSTERING TECHNIQUE AND BEST NUMBER OF HIDDEN LAYERS ACCORDING TO COMPARISON OF DIFFERENT NEURAL NETWORK ARCHITECTURES

Dataset	Number of Clusters	Best Number of Hidden Layers According to Comparison of Different Neural Network Architectures
Landsat satellite	2	2
ECG	1	1
P-wave	2	2
QRS	1	1
T-wave	3	3
Sonar	2	2

Based on these results, it is confirmed that the optimal architecture of a multi-layer neural network for each of the six datasets analyzed can be determined using the clustering program.

B. Experiment for optimal number of hidden-layer neurons

The second analysis is based on experiments performed using Weka, a collection of machine learning algorithms for data mining tasks [36]. Weka can be used to determine the percentage of accuracy and error/epoch values for different numbers of neurons as calculated using formulas (1)–(6), defined previously.

For the Landsat satellite dataset, the number of clusters is two. Therefore, the number of hidden layers of the neural

network is taken to be equal to two. The number of hidden layers was set similarly for the other datasets. The number of neurons was calculated using each of formulas (1)–(6). For each dataset and each of the six formulas, we used Weka to calculate the accuracy and error/epoch values, with the results as given in Table III.

As observed from the table, the best accuracy for the Landsat satellite dataset was obtained with a number of neurons in each hidden layer equal to 13 or 65, each providing an accuracy of approximately 82%. The smallest error/epoch value, 0.02803, was obtained from the neural network with 13 neurons in each hidden layer. It is concluded that the best number of hidden neurons is 13, meaning that formula (6) is the best choice.

TABLE III. ACCURACY AND ERROR/EPOCH VALUES FOR NUMBERS OF NEURONS AS CALCULATED BASED ON FORMULAS (1)–(6) (SECTION VI)

Dataset	Formula	Number of Hidden Layers	Number of Neurons per Hidden Layer	Accuracy (%)	Error / Epoch
Landsat satellite	(1)	2	2	76.7535	0.04655
	(2)	2	1	50.3006	0.08779
	(3)	2	3	80.5611	0.03734
	(4)	2	2	76.7535	0.04655
	(5)	2	65	82.5651	0.02851
	(6)	2	13	82.3647	0.02803
ECG	(1)	1	11	57.9646	0.03162
	(2)	1	3	59.292	0.03469
	(3)	1	14	59.292	0.03140
	(4)	1	2	60.8407	0.03524
	(5)	1	89	59.292	0.03157
	(6)	1	32	59.292	0.03097
P-wave	(1)	2	4	53.7611	0.04217
	(2)	2	1	53.9823	0.04218
	(3)	2	6	53.9823	0.04216
	(4)	2	1	53.9823	0.04218
	(5)	2	117	53.9823	0.04222
	(6)	2	15	54.2035	0.04218
QRS	(1)	1	9	59.0708	0.03510
	(2)	1	3	58.8496	0.03573
	(3)	1	12	58.8496	0.03506
	(4)	1	2	59.5133	0.03610
	(5)	1	234	59.9558	0.03546
	(6)	1	30	58.8496	0.03513
T-wave	(1)	3	3	53.9823	0.04159
	(2)	3	1	54.2035	0.04249
	(3)	3	4	53.9823	0.04132
	(4)	3	2/3	-	-
	(5)	3	78	56.1947	0.03927
	(6)	3	10	56.6372	0.03928
Sonar	(1)	2	15	81.25	0.01459
	(2)	2	2	80.76	0.03562
	(3)	2	20	80.76	0.01360
	(4)	2	-	-	-
	(5)	2	3	81.73	0.01342
	(6)	2	22	81.25	0.01458

Formula (6) is proved to be the best choice of the six formulas since the NeuroShell program uses formula (6) as well. It can be concluded that the previous experimental tests carried out with the NeuroShell program confirm the performance of the simulation software used, thereby enhancing the accuracy of the previous results.

VIII. CONCLUSION

Neural networks represent an important tool in data classification. It is the only technique that allows generalizations based on a set of data to be analyzed. Regardless of the chosen neural network architecture and how learning is used, the number of hidden layers and the number of neurons in these layers are currently determined empirically by testing. Use of clustering techniques can help extract a number of common characteristics of input forms by which neural network input records are classified into groups. Using this method, a number close to the optimal number of hidden layers of a multi-layer neural network can be obtained.

With the use of clustering techniques, many factors affect the number of hidden layers of a multi-layer neural network. One such factor is the reference distance. The accuracy of this factor is important in determining the optimal number of hidden layers.

It is believed that the use of this method can save time in the building of an optimal neural network for solving any specific problem.

REFERENCES

[1] J.-Y. Wu, "An evolutionary multi-layer perceptron neural network for solving unconstrained global optimization problems," in Proc. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 2016, pp. 241-246.

[2] S. Li, Z. Qiu, "Research on calculating method of hidden layer nodes in BP network," in Proc. 2015 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, 2015, pp. 2912-2916.

[3] K. Gnana Sheela and S. N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks," Mathematical Problems in Engineering, vol. 2013, Article ID 425740, 11 pages, 2013.

[4] B. M. Wilamowski, "Neural network architectures and learning algorithms," IEEE Industrial Electronics Magazine, vol. 3, no. 4, pp. 56-63, Dec. 2009.

[5] S. Bunjongjit, A. Ngaopitakkul, C. Pothisarn, C. Jettanasen, "Improvement to reduce training time of back-propagation neural networks for discrimination between external short circuit and internal winding fault," in Proc. 2014 International Conference on Information Science, Electronics and Electrical Engineering (ISEEE), Sapporo, Japan, 2014, pp. 614-618.

[6] R. Lovassy, L. T. Kóczy, L. Gál, I. J. Rudas, "Fuzzy neural networks stability in terms of the number of hidden layers," in Proc. 2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2011, pp. 323-328.

[7] C.-Y. Lin, Y.-C. Chen, C.-Y. Wang, C.-Y. Huang, C.-T. Hsu, "Minimization of number of neurons in Voronoi diagram–based artificial neural networks," IEEE Transactions on Multi-Scale Computing Systems, vol. 2, no. 4, pp. 225-233, Oct.-Dec. 2016.

[8] C. Contreras-Bolton, G. Gatica, C. Barra and V. Parada, "A multi-operator genetic algorithm for the generalized minimum spanning tree problem," Expert Systems with Applications, vol. 50, pp. 1-8, 2016.

[9] A. Serra, D. Greco, R. Tagliaferri, "Impact of different metrics on multi-view clustering," in Proc. 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8.

[10] K. A. Patel, P. Thakral, "The best clustering algorithms in data mining," in Proc. 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 2016, pp. 2042-2046.

[11] C. Shah, A. Jivani, “Comparison of data mining clustering algorithms,” in Proc. 2013 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, 2013, Sr. No. 8.

[12] C. Zhong, M. Malinen, D. Miao and P. Fränti, “A fast minimum spanning tree algorithm based on K-means,” Information Sciences, vol. 295, pp. 1-17, 2015.

[13] A. Srinivasan, “Statlog (Landsat Satellite) Data Set,” 1993, in D. Dua, E. Karra Taniskidou, UCI Machine Learning Repository. Irvine, California: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29

[14] H. A. Guvenir, B. Acar, H. Muderrisoglu, “Arrhythmia Data Set,” 1998, in D. Dua, E. Karra Taniskidou, UCI Machine Learning Repository. Irvine, California: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml/datasets/Arrhythmia

[15] T. Sejnowski, “Connectionist Bench (Sonar, Mines vs. Rocks) Data Set,” 1988, in D. Dua, E. Karra Taniskidou, UCI Machine Learning Repository. Irvine, California: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Sonar%2C+Mines+vs.+Rocks%29

[16] Y. Tan, S. Chen, “Pattern recognition based on weighted fuzzy C-means clustering,” in Proc. 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, 2013, pp. 1061-1065.

[17] H. Pan, D. Liang, J. Tang, N. Wang, W. Li, “Shape recognition and retrieval based on edit distance and dynamic programming,” Tsinghua Science and Technology, vol. 14, no. 6, pp. 739-745, Dec. 2009.

[18] A. Fernández, Á. Gómez, F. Lecumberry, Á. Pardo and I. Ramírez, “Pattern Recognition in Latin America in the “Big Data” Era,” Pattern Recognition, vol. 48, no. 4, pp. 1185-1196, 2015.

[19] B. Boser, I. Guyon and V. Vapnik, “A training algorithm for optimal margin classifiers,” In Proceedings of the fifth annual workshop on Computational learning theory (COLT '92). ACM, New York, NY, USA, 144-152, 1992.

[20] R.-E. Precup, R.-C. David, E. M. Petriu, M.-B. Radac, S. Preitl, “Adaptive GSA-based optimal tuning of PI controlled servo systems with reduced process parametric sensitivity, robust stability and controller robustness,” IEEE Transactions on Cybernetics, vol. 44, no. 11, pp. 1997-2009, Nov. 2014.

[21] O. Castillo, H. Neyoy, J. Soria, P. Melin, F. Valdez, “A new approach for dynamic fuzzy logic parameter tuning in Ant Colony Optimization and its application in fuzzy control of a mobile robot,” Applied Soft Computing, vol. 28, pp. 150-159, Mar. 2015.

[22] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97, 2012.

[23] C. Guada, D. Gómez, J. T. Rodríguez, J. Yáñez, J. Montero, “Fuzzy image segmentation based on the hierarchical divide and link clustering algorithm,” in Proc. 2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Taipei, Taiwan, 2015, pp. 12-17.

[24] H. Li, S. Wang, R. Tang, “Research on the high robustness data classification and the mining algorithm based on hierarchical clustering and KNN,” in Proc. 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2016, Paper SP645.

[25] X. Chen, K. Chau and W. Wang, “A novel hybrid neural network based on continuity equation and fuzzy pattern-recognition for downstream daily river discharge forecasting,” Journal of Hydroinformatics, vol. 17, no. 5, pp. 733-744, 2015.

[26] R. M. Hristev, The ANN Book, Edition 1. GNU public license, p. 9, 1998.

[27] E. Micheli-Tzanakou, Supervised and Unsupervised Pattern Recognition. Boca Raton, FL: CRC Press, pp. 135-180, 1999.

[28] J. Yu, S. Wang, and L. Xi, “Evolving artificial neural networks using an improved PSO and DPSO,” Neurocomputing, vol. 71, no. 4–6, pp. 1054–1060, Jan. 2008.

[29] J. R. Rabuñal, J. Dorado, Artificial Neural Networks in Real-Life Applications. London: Idea Group Publishing, pp. 270-272, 2006.

[30] I. Kaastra and M. Boyd, “Designing a neural network for forecasting financial and economic time series,” Neurocomputing, vol. 10, no. 3, pp. 215–236, Apr. 1996.

[31] Ward Systems Group, Inc., “NeuroShell 2.” http://www.wardsystems.com/neuroshell2.asp

[32] T. Sasaki, K. Kinoshita, S. Kishida, Y. Hirata, S. Yamada, “Effect of number of input layer units on performance of neural network systems for detection of abnormal areas from X-ray images of chest,” in Proc. 2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS), Qingdao, China, 2011, pp. 374-379.

[33] A. S. Shafie, I. A. Mohtar, S. Masrom, N. Ahmad, “Backpropagation neural network with new improved error function and activation function for classification problem,” in Proc. 2012 IEEE Symposium on Humanities, Science and Engineering Research, Kuala Lumpur, Malaysia, 2012, pp. 1359-1364.

[34] K. Chen, S. Yang, C. Batur, “Effect of multi-hidden-layer structure on performance of BP neural network: Probe,” in Proc. 2012 Eighth International Conference on Natural Computation (ICNC), Chongqing, China, 2012, pp. 1-5.

[35] M. S. Al-Duais, A. R. Yaakub, N. Yusoff, “Dynamic training rate for backpropagation learning algorithm,” in Proc. 2013 IEEE 11th Malaysia International Conference on Communications (MICC), Kuala Lumpur, Malaysia, 2013, pp. 277-282.

[36] Machine Learning Group at the University of Waikato, “Waikato Environment for Knowledge Analysis (Weka).” https://www.cs.waikato.ac.nz/~ml/weka/