# Determining Optimal Multi-layer Perceptron Structure Using Linear Regression

Mohamed Lafif Tej(✉) and Stefan Holban

Faculty of Automation and Computers, Politehnica University of Timisoara,
Timisoara, Romania
afiftej@gmail.com, stefan.holban@cs.upt.ro

**Abstract.** This paper presents a novel method to determine the optimal Multi-layer Perceptron structure using Linear Regression. Starting from clustering the dataset used to train a neural network it is possible to define Multiple Linear Regression models to determine the architecture of a neural network. This method work unsupervised unlike other methods and more flexible with different datasets types. The proposed method adapt to the complexity of training datasets to provide the best results regardless of the size and type of dataset. Clustering algorithm used to impose a specific analysis of data used to train the network such us determining the distance measure, normalization and clustering technique suitable with the type of training dataset used.

**Keywords:** Multi-layer Perceptron · Linear regression · Clustering methods · Pattern recognition · Artificial neural network

## 1 Introduction

Determining the structure of Multi-layer Perceptron is a critical issue in the design of a Neural Network [1]. Until now, there is no general equation to define the structure of Multi-layer Perceptron, which can deal with different kind of problems to be resolved by the neural network. Each problem needs a particular structure that responds to his requirements. Methods currently used do not rely on the complexity of the problem must be solved by the Multi-layer Perceptron. Most currently used methods are very limited, time-consuming and supervised [2] such us Growing and Pruning Algorithms, Exhaustive Search, Evolutionary Algorithms and so on. In this paper, a novel method to determine the optimal Multi-layer Perceptron structure using Linear Regression will be introduced. The idea is to group the dataset used to train the Multi-layer Perceptron using conventional methods of pattern recognition [3, 4] according to specific criteria until we get a set of useful parameters, which will be used in the design of Multi-layer Perceptron structure. The results obtained from clustering the dataset used to train the network are used as independent variables to define a linear regression models [5] used to determine the Multi-layer Perceptron structure. The equation defined by the linear regression used to minimize the distance between a fitted line and all the data points. The regression model aims to achieve maximum accuracy in determining the number of hidden layers and the number of neurons in these layers.

W. Abramowicz and R. Corchuelo (Eds.): BIS 2019, LNBIP 353, pp. 232–246, 2019.
https://doi.org/10.1007/978-3-030-20485-3_18

## 2 Related Work

The design of the structure of a neural network is an extremely active area of research and does not yet have any definitive guiding theoretical principles. The currently used methods are very limited and time-consuming such as Growing and Pruning algorithms [6], exhaustive search, and evolutionary algorithms [7]. Here are some widely spread methods for determining the number of hidden neurons.

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 [8]. These rules provide a starting point but do not achieve the best architecture only after a number of tests based on trial and error. Trial and Error approach does not yield good results except by accident, sometimes called exhaustive search [9]. Exhaustive Search approach makes searching through all possible topologies and then select the one with the least generalization error. The disadvantage of this method is time-consuming.

The Growing neural network algorithm was initially proposed by Vinod et al. [10]. Growing Algorithms 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 method tries to train an oversized network, and then determines the relative importance of weights by analyzing them. This method prunes the weights with the least importance and then repeats the task. The disadvantage of this method is that the analysis of weights is time-consuming.

In this paper, we proposed a method to determine the structure of a Multi-layer Perceptron based on the complexity level of the considered problem making it more flexible with different datasets types than classical methods. In addition, this method makes the design of Multi-layer Perceptron unsupervised.

## 3 Multiple Linear Regression Method Used

Following a set of criteria in the analysis of clusters obtained through hierarchical clustering of the dataset used to train the neural network, which results a number of parameters can be useful to define a linear regression model to determine the structure of Multi-layer Perceptron [11]. Parameters obtained from clustering will be evaluated using statistical hypothesis testing [12] to be able to identify whether it exists dependencies between these parameters and the number of hidden layers and the number of hidden neurons. The parameters selected through this evaluation used as independent variables of the regression models [13].

Figure 1 presents a framework of the regression model. The model shows how to use results obtained from clustering the training dataset to determine the regression model used to generate the optimal number of hidden layers and the number of neurons in these layers.

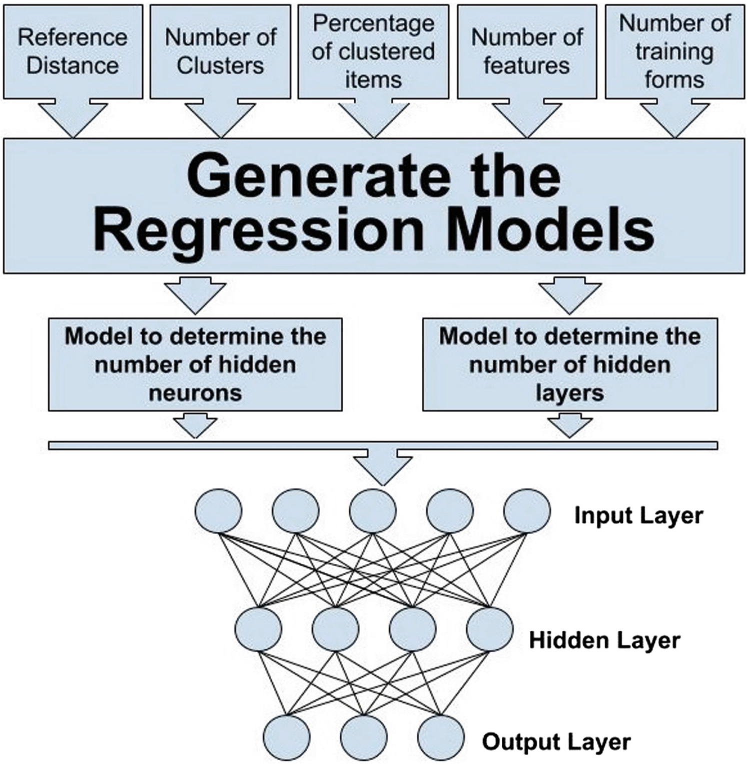

Reference Distance
Number of Clusters
Percentage of clustered items
Number of features
Number of training forms
Generate the Regression Models
Model to determine the number of hidden neurons
Model to determine the number of hidden layers
Input Layer
Hidden Layer
Output Layer

**Fig. 1.** The framework of the regression model

### 3.1 Regression Analysis

Regression analysis is a statistical technique to predict a quantitative relationship between a dependent variable and a set of independent variables [14]. The defined regression equation depends on the assumption concerning the relationship between the dependent variable and the independent variables [15]. The linear regression equation seeking to minimize the errors to fit the data points to a straight regression line representing the equation. Using information obtained by observations or measurements, the equation is defined. The indicator of multiple determination coefficient $R^2$ is required to determine the relationship between the Independent variable and the dependent variables. $R^2$ expresses the variation of the dependent variable affected by the variation of independent variables. The indicator of multiple determination coefficient is essential special if supported by other statistical indicators [16].

The mathematically Multiple Linear Regression model having this form:

$$f = X \rightarrow Y$$

$$f(X) = w_0 + \sum\nolimits_{j=1}^{n} w_j x_j \tag{1}$$

The regression models consist of unknown parameter w, the dependent variable Y and the independent variable X.

### 3.2 Statistical Hypothesis Testing

The statistical hypothesis testing [17] are used to examine parameters obtained through hierarchical clustering of training dataset to select a number of parameters to determine the regression model. The hypothesis testing used to prove that the regression model is significant. Depending on the null hypothesis $H_0$, which assume no significant relationship between the independent variables X and the dependent variable Y.

> *$H_0$: There is no relationship between the clustering results and the structure of Multi-layer Perceptron*
>
> *Ha: There is a relationship between the clustering results and the structure of Multi-layer Perceptron*

The probability coefficients of independent variables (P-value) have a value of less than 0.05 based on parameters proposed to be independent variables of the regression model.

F-Test analysis [18] used for the analysis of variance will be taken as an evidence to prove that the structure of Multi-layer Perceptron depends on the selected factors.

### 3.3 The Independent Factors Selected

Based on statistical hypothesis testing and F-Test a set of factors are proposed to determine the regression equation in addition to that it has been proven that there is a link between all factors in models. Moreover, relatively small positive and negative correlations exist [19]. The selected factors prove the effectiveness and efficiency of the proposed model through the Multiple Coefficient of Determination [20] and the Multiple Correlation Coefficient [21] where they obtain results close to 1.

The proposed factors obtained by clustering the training dataset will be used as independent variables to determine the regression models:

- The number of obtained cluster
- The percentage of grouped items
- The reference distance
- The number of training forms
- The number of features in the input

Moreover, the quality measure of the network structure was considered as an independent factor. The quality measure factor takes into account the configuration and interconnection layers [22].

The proposed regression models consist of two models the first model used to determine the number of hidden layers and the second model used to determine the number of hidden neurons.

**Regression Model to Determine the Number of Hidden Layers.** A set of factors prove the ability to influence on the dependent variable y, which represent the number of hidden layers of a Multi-layer Perceptron. Using the statistical hypothesis testing mentioned previously and experimental results it is turned out that the dependent variable y depending on changes of the following independent variables.

- $X_1$: The number of obtained cluster multiplied by the reference distance
- $X_2$: The reference distance
- $X_3$: The percentage of grouped items
- $X_4$: The quality measure of the network structure

The independent variables $X_1$, $X_2$, and $X_3$ obtained through clustering of the training dataset. In addition to that, a quality measure of the network structure $X_4$ is taken as independent factors. The quality measure factor depends on the reference distance and the structure of Multi-layer Perceptron.

A Multiple Linear Regression model representing the equation to determine the number of hidden layers of the Multi-layer Perceptron have the following mathematical form:

$$y = a_0 + \sum\nolimits_{j=1}^{4} a_j x_j \tag{2}$$

The dependent variable y represents the number of hidden layers and $a_0$, $a_1$, $a_2$, $a_3$, and $a_4$ present the constants used to predict the dependent variable y. $a_0$ is the intercept parameter and $a_1$, $a_2$, $a_3$, and $a_4$ are the slope parameters.

Based on the percentage of contribution of each independent variable in the regression model and the absolute values of partial correlation coefficients let us concluded that dependent variable y is influenced by several factors. Among these factors is Reference Distance, which has an important influence on the number of hidden layers.

**Regression Model to Determine the Number of Hidden Neurons.** The regression model used to calculate the number of hidden neurons will be determined using a set of factors selected in accordance with the above considerations from the results obtained through clustering of the training dataset. The number of hidden neurons depending on changes in the following independent variables.

- $X_1$: The number of features in the input
- $X_2$: The number of obtained cluster
- $X_3$: The reference distance
- $X_4$: The quality measure of the network structure

The independent variable $X_1$ represents the number of features of the training dataset. $X_2$ and $X_3$ present the obtained number of cluster and the reference distance respectively obtained using clustering of the training dataset.

A Multiple Linear Regression model representing the equation to determine the number of hidden neurons of the Multi-layer Perceptron have the following mathematical form:

$$y = a_0 + \sum\nolimits_{j=1}^{4} a_j x_j \tag{3}$$

The dependent variable y represents the number of hidden neurons. The obtained number of hidden neurons will be evenly distributed to the hidden layers if the number of hidden layers exceeds one layer. Therefore, each hidden layer contains a number of neurons equal to others.

The independent factors influence on the number of hidden neurons with varying levels. The factor that has the highest influence being the Reference Distance.

The number of hidden neurons obtained using regression method will be divided equally by the number of hidden layers.

### 3.4 Clustering of the Training Dataset

The proposed regression method depends mainly on the results obtained from clustering of the training dataset. The most convenient clustering algorithm for the proposed method is Agglomerative Hierarchical Clustering algorithm [23]. Each cluster obtained through Agglomerative Hierarchical Clustering seeks to ensure the highest similarity of objects within the cluster and at the same time the highest dissimilarity between clusters [24]. Clusters obtained using Agglomerative Hierarchical Clustering can contain several sub-clusters then there will be a hierarchical clustering. The hierarchical clustering is a set of nested clusters that build a cluster tree (Dendrogram) to represent objects. The root of the tree represents the cluster, which group all other clusters and objects. In some cases, the leaves of the tree represent clusters of one objects. The Agglomerative Hierarchical Clustering algorithm [25–29] consider each object as a single cluster and then try to join the closest clusters until obtaining only one single cluster. The optimal number of clusters is determined by making a cut of all segment with a length greater than a predefined value [30]. This reference value (Reference Distance) is chosen according to specific criteria.

The value of Reference Distance, which is appropriate to obtain the optimal number of clusters, must attain a set of criteria. Implementation of the following criteria can make the number of obtained clusters useful for the proposed regression method to determine the structure of Multi-layer Perceptron.

The first criterion requires grouping at least ninety percent of the items of the training dataset. The ninety percent of items grouped considered sufficient where the result could cover the entire training dataset.

The second criterion requires that the number of clusters should be taken as few as possible in order to minimize the size of the network with reason that the increase of the number of clusters causes an increase in the number of hidden layers and the number of hidden neurons using the proposed method. With a few numbers of hidden layers and neurons the complexity of Multi-layer Perceptron reduce [31].

The third criterion requires a Reference Distance value in which any increase on it does not affect the number of obtained clusters [32–34]. While taking into account the condition, which should be avoided such as the very short value of Reference Distance for which each leaf of the tree, represents a cluster of one object or a relatively large value of Reference Distance for which grouping all objects in one cluster.

The fourth criterion requires the right selection of distance metrics (such as Manhattan and Euclidean) and linkage methods (single, complete, and average linkage) appropriate to the clustering algorithm and the type of training dataset [35]. A good choice of distance metrics increases the accuracy of the proposed method.

By following these criteria, the results of clustering of training dataset can generate a set of parameters useful to construct the regression models used for determining the optimal structure of a Multi-layer Perceptron.

# 4 Experimental Results

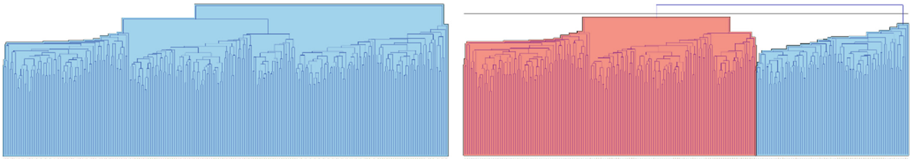
A number of experimental tests will be conducted to prove the effectiveness of the proposed method. The training dataset will be trained using different Multi-layer Perceptron structure then compared to results of the proposed method.

In this paper, the Waveform Database Generator Version 1 Dataset used to prove the validity of the proposed method. Waveform dataset consists of 21 attributes and 5000 instances. Dataset classes are generated from a combination of two of three "base" waves.

## 4.1 Experimental Results Obtained from Clustering of the Training Dataset

Agglomerative Hierarchical clustering is used to cluster the Waveform dataset. The number of clusters varies based on the value of Reference Distance. According to the criteria described above the perfect Reference Distance value is 8.

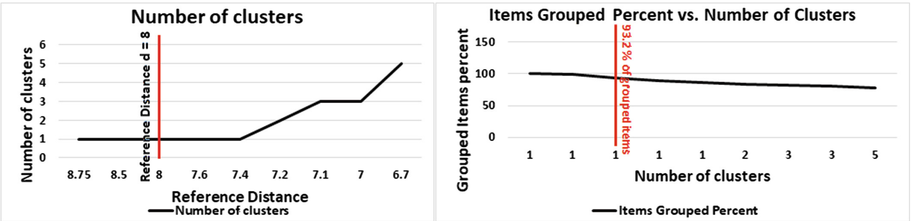
Figures 2 and 3 below presents the number of clusters and the percentage of objects clustered and the corresponding values of Reference Distance.



(i) one cluster obtained for more than 90% of items grouped

(ii) two clusters obtained for less than 90% of items grouped

**Fig. 2.** Clusters obtained based on the percentage of items grouped



(i) Clusters number obtained vs. the corresponding Reference Distance of Waveform dataset

(ii) Items Grouped Percent vs. number of clusters of Waveform dataset

**Fig. 3.** Clusters number obtained vs. the corresponding reference distance of waveform dataset and items grouped percent vs. number of clusters of waveform dataset

The appropriate parameters selected in this case study is Normalization type "standard", clusters distance "Average Link" and "Manhattan" distance.

Figure 2 presents the obtained Dendrogram and the corresponding number of clusters obtained according to the reference distance values. Based on criteria listed above the optimal number of clusters is one cluster.

According to the criteria described above, we conclude that the ideal number of clusters is one cluster with 93.2% of items clustered for a value of Reference Distance equal to 8. The results obtained can be useful for determining the regression models used to construct the optimal structure of Multi-layer Perceptron for training the Waveform dataset.

### 4.2 Calculating the Number of Hidden Layers Using Regression Model

Based on the clustering of Waveform dataset the independent factors used to determine the Eq. (2) for calculating the number of hidden layers has the following values: $X_1 = 1 \times 8$, $X_2 = 8$, $X_3 = 93.2$, $X_4 = 98.38$.

According to the criteria described above, the selected value of Reference Distance is 8 for a percentage of grouped items more than 90% and the corresponding number of clusters is one, therefore, the value of $X_1$ will be $8 \times 1$ equal to 8. $X_2$ represents the value of Reference Distance therefore, $X_2 = 8$. $X_3$ represents the percentage of grouped items therefore $X_3 = 93.2$ as we see in Fig. 3. For $X_4$ which represents the quality measure of the network structure, it was determined by creating a structure based on the number of hidden layers equal to the obtained number of clusters corresponding to the selected Reference Distance and for the number of neurons is determined using the Formula (9). Based on that the quality measure of the network structure $X_4 = 98.38$ corresponding to the selected Reference Distance.

Using the above values in Eq. (2) will result in $y = 1$.

$Y = 1$, this concludes that the optimal number of hidden layers using the proposed method is equal to one layer.

The values of the Multiple Determination Coefficient $R^2$ obtained is close to 1. $R^2$ expresses the level of variation of the number of hidden layers affected by the variation of selected independent variable $X_1$, $X_2$, $X_3$, and $X_4$. It proves the validation of the proposed model and the successful choice of independent factors.

### 4.3 Calculating the Number of Hidden Neurons Using Regression Model

The implementation of Eq. (3) to calculate the number of neurons in the hidden layers use the following values of independent factors obtained from the clustering of Waveform dataset.

Based on the clustering of Waveform dataset the independent factor used to determine the Eq. (3) for calculating the number of hidden layers has the following values.

$X_1 = 21$, $X_2 = 1$, $X_3 = 8$, $X_4 = 98.38$

Using the above values in Eq. (3) will result in $y = 74$.

$Y = 74$, this concludes that the optimal number of hidden neurons using the proposed method is equal to 74 hidden neurons.

$R^2$ obtained is close to 1. $R^2$ expresses the level of variation of the number of hidden neurons affected by the variation of selected independent variable $X_1$, $X_2$, $X_3$ and $X_4$. It proves the validation of the proposed model and the successful choice of independent factors.

### 4.4 Comparison of the Proposed Method with Classical Methods

To validate the results obtained using the proposed method a comparison with widely spread methods are conducted. The proposed regression method will be compared with the classical methods so that we can prove the validity of the proposed method. The following classical formulas will be used in this comparison:

*In – number of input neurons*
*Out – number of output neurons*
*Hidden – number of hidden neurons*
*Training – number of training forms*

$$Hidden = 1/2(In + Out) \tag{4}$$

$$Hidden = SQRT\,(1/2\,(In + Out)) \tag{5}$$

$$Hidden = (In + Out) * 2/3 \tag{6}$$

$$Hidden = Training\,/10\,(In + Out) \tag{7}$$

$$Hidden = (Training - Out)\,/\,In + Out + 1 \tag{8}$$

$$Hidden = 1/2\,(In + Out) + SQRT\,(Training) \tag{9}$$

Formula (10): The number of hidden neurons should be between the size of the input layer and the size of the output layer. Formula (11): The number of hidden neurons should be less than twice the size of the input layer. A set of datasets used such as Waveform Database Generator dataset, Image Segmentation dataset, Glass identification dataset, Landsat dataset, Sonar dataset, ECG dataset, QRS dataset, P-wave dataset and T-wave datasets. Table 1 Presents specifications of datasets used.

**Table 1.** Specifications of neural networks

| Dataset | Sonar | ECG | P-wave | QRS | T-wave | Landsat | Glass | Segmentation | Waveform |
|---|---|---|---|---|---|---|---|---|---|
| Input neurons | 60 | 6 | 2 | 2 | 2 | 4 | 9 | 19 | 21 |
| Output neurons | 2 | 16 | 16 | 16 | 16 | 7 | 7 | 7 | 3 |
| Training items | 208 | 452 | 452 | 452 | 452 | 6435 | 214 | 2310 | 5000 |

Table 2 presents the number of hidden neurons using the classical method:

**Table 2.** Number of hidden neurons using the classical method

| Dataset | Formula (4) | Formula (5) | Formula (6) | Formula (7) | Formula (8) | Formula (9) | Formula (10) | Formula (11) |
|---|---|---|---|---|---|---|---|---|
| Sonar | 31 | 5.75 | 41 | 0 | 6 | 45 | 2 < x < 60 | x < 120 |
| ECG | 11 | 3.32 | 14 | 2 | 89 | 32.26 | 6 < x < 16 | x < 12 |
| P-wave | 9 | 3 | 12 | 2 | 235 | 30.26 | 2 < x < 16 | x < 4 |
| QRS | 9 | 3 | 12 | 2 | 235 | 30.26 | 2 < x < 16 | x < 4 |
| T-wave | 9 | 3 | 12 | 2 | 235 | 30.26 | 2 < x < 16 | x < 4 |
| Landsat | 5 | 2.24 | 7 | 58 | 1615 | 85.22 | 4 < x < 7 | x < 8 |
| Glass | 8 | 2.83 | 10 | 1 | 31 | 22.63 | 9 < x < 7 | x < 18 |
| Segmentation | 13 | 3.61 | 17 | 8 | 129 | 62 | 19 < x < 7 | x < 38 |
| Waveform | 12 | 3.46 | 16 | 20 | 241 | 82.71 | 21 < x < 3 | x < 42 |

Table 3 presents the number of hidden neurons using the proposed method:

**Table 3.** Number hidden neurons using the proposed method

| Dataset | Neurons | Layers |
|---|---|---|
| Sonar | 35 | 2 |
| ECG | 20 | 1 |
| P-wave | 10 | 2 |
| QRS | 21 | 1 |
| T-wave | 20 | 3 |
| Landsat | 24 | 2 |
| Glass | 207 | 2 |
| Segmentation | 75 | 1 |
| Waveform | 74 | 1 |

**Comparison Based on the Training Time Using Classical Methods vs. the Proposed Method.** A comparison of the training time using classical methods vs. the proposed method results that the training time of the proposed method does not have the best training time for all datasets but for ECG, P-wave and QRS perform well. For example, the training time of ECG is 0.75 s using the proposed method and the best classical method record is equal to 1.47 s. The failure of the proposed method with some datasets to obtain the best training time is because of the number of neurons selected. The training time depends mainly on the number of neurons in the network and the size of dataset regardless of the used formula. Since the number of neurons is selected based on the complexity of the problem, therefore, the training time is affected by the complexity of the problem using the proposed method.

**Comparison Based on the Percentage of Accuracy Using Classical Methods vs. the Proposed Method.** Table 4 below shows a comparison of the results in terms of percentage of classification accuracy [36] using classical methods vs. the proposed method.

**Table 4.** Comparison of the percentage of accuracy using classical methods vs. the proposed method

| Dataset | Formula (4) | Formula (5) | Formula (6) | Formula (7) | Formula (8) | Formula (9) | Formula (10) | Formula (11) | proposed method |
|---|---|---|---|---|---|---|---|---|---|
| Sonar | 81.25 | 80.76 | 80.76 | 74.5192 | 81.73 | 81.25 | 81.7308 | 81.25 | 82.2115 |
| ECG | 57.9646 | 59.292 | 59.292 | 60.8407 | 59.292 | 59.292 | 57.9646 | 59.5133 | 60.8407 |
| P-wave | 53.7611 | 53.9823 | 53.9823 | 53.9823 | 53.9823 | 54.2035 | 53.9823 | 53.9823 | 54.2035 |
| QRS | 59.0708 | 58.8496 | 58.8496 | 59.5133 | 59.9558 | 58.8496 | 59.5133 | 59.5133 | 60.177 |
| T-wave | 53.9823 | 54.2035 | 53.9823 | 54.2035 | 56.1947 | 56.6372 | 56.6372 | 54.2035 | 56.6372 |
| Landsat | 76.7535 | 50.3006 | 80.5611 | 76.7535 | 82.5651 | 82.3647 | 84.8703 | 85.3213 | 85.6595 |
| Glass | 85.0467 | 72.8972 | 82.7103 | 61.6822 | 97.6636 | 94.3925 | 85.0467 | 82.7103 | 98.5981 |
| Segmentation | 97.5325 | 95.2381 | 97.9221 | 95.8442 | 97.6623 | 98.8312 | 95.7576 | 98.8312 | 99.1342 |
| Waveform | 95.74 | 89.28 | 97.32 | 97.86 | 98.22 | 98.4 | 96.82 | 97.86 | 98.6 |

As observed from Table 4 and Fig. 4, the proposed method has the best percentage of accuracy for most datasets. The classical methods sometimes get good results but it depends on the database. For example, formula (7) has a good percentage of accuracy for ECG dataset and the lowest percentage accuracy for sonar and Landsat datasets. Formula (8) obtained the highest percentage of accuracy compared to other classical methods for Glass dataset while getting the lowest percentage for ECG dataset. Formula (9) perform well with datasets Glass, Segmentation and waveform but for other dataset have a medium percentage of accuracy.

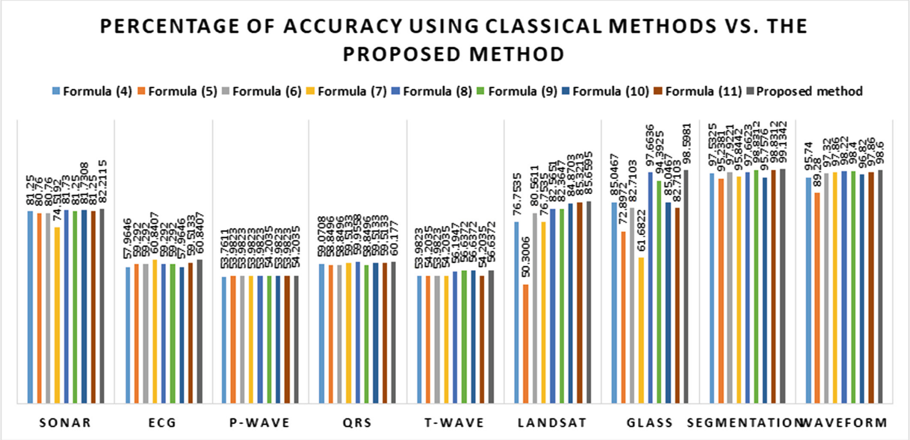


**Fig. 4.** Comparison of the percentage of accuracy using classical methods vs. the proposed method

**Comparison Based on the Error/Epoch Using Classical Methods vs. the Proposed Method.** Table 5 presents a comparison of the error/epoch [37] using classical methods vs. the proposed method:

**Table 5.** Comparison of the error/epoch using classical methods vs. the proposed method

| Dataset | Formula (4) | Formula (5) | Formula (6) | Formula (7) | Formula (8) | Formula (9) | Formula (10) | Formula (11) | proposed method |
|---|---|---|---|---|---|---|---|---|---|
| Sonar | 0.014593 | 0.035628 | 0.013602 | 0.046503 | 0.013424 | 0.014582 | 0.00485 | 0.01448 | 0.00487 |
| ECG | 0.031626 | 0.034692 | 0.031409 | 0.035248 | 0.031574 | 0.030976 | 0.03186 | 0.03314 | 0.03507 |
| P-wave | 0.042172 | 0.042182 | 0.042169 | 0.042182 | 0.042229 | 0.042180 | 0.04216 | 0.04218 | 0.04209 |
| QRS | 0.035101 | 0.035733 | 0.035060 | 0.036108 | 0.035463 | 0.035133 | 0.03524 | 0.03610 | 0.03548 |
| T-wave | 0.041593 | 0.042493 | 0.041322 | 0.042407 | 0.039277 | 0.039284 | 0.03913 | 0.04240 | 0.03913 |
| Landsat | 0.046556 | 0.087793 | 0.037348 | 0.046556 | 0.028518 | 0.028037 | 0.02892 | 0.03111 | 0.02671 |
| Glass | 0.036890 | 0.061292 | 0.036047 | 0.084371 | 0.003342 | 0.010134 | 0.03689 | 0.03604 | 0.00267 |
| Segmenta- | 0.005022 | 0.010025 | 0.004892 | 0.008103 | 0.003333 | 0.002705 | 0.00941 | 0.00365 | 0.00239 |
| Waveform | 0.026669 | 0.052205 | 0.017591 | 0.014369 | 0.011738 | 0.010616 | 0.02026 | 0.01436 | 0.00934 |

As observed in Table 5 and Fig. 5, the proposed method has the lowest values of error/epoch for most datasets. The classical methods sometimes get good results with formulas (8) (9) (10) these results somewhat acceptable compared to other classical methods. Formulas (5) and (7) have the highest values of error/epoch.

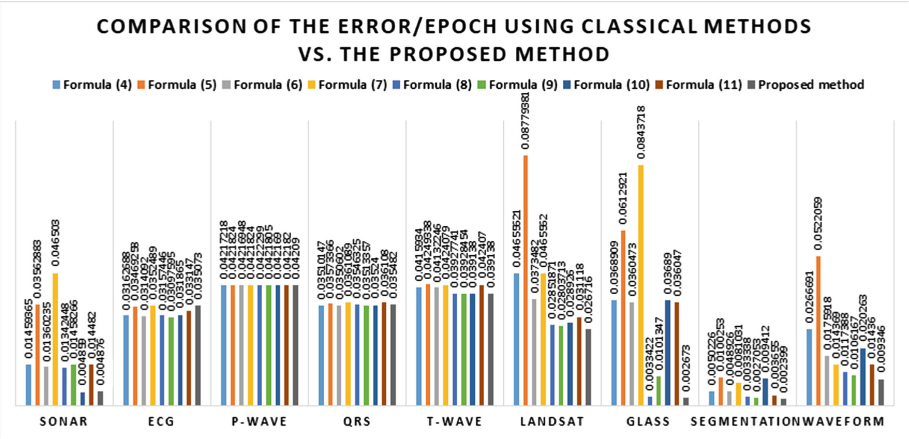


**Fig. 5.** Comparison of the error/epoch using classical methods vs. the proposed method

**Comparison of Classical Methods vs. the Proposed Method Conclusion.** The proposed method get the best percentage accuracy for most datasets, unlike the classical methods. The classical formulas (4) (5) (6) (10) (11) perform well with small datasets which have a few training items. Whereas Formulas (7) (8) (9) perform well with large datasets because they take into consideration the number of training items. Formula (9) is better than (7) and (8) which mean the SQRT of training items have a positive effect on the results. Formulas (4) (5) (6) depend mainly on the number of input and output neurons making it effective for a small datasets while do not perform well with large datasets which have complex problems to solve. The training time depends mainly on the number of neurons in the network and the size of dataset regardless the used formula. The results of error/epoch obtained is almost similar to the result of the percentage of accuracy. Comparison of the proposed method with classical methods leads us to deduce that the proposed method performs well for the different type of datasets, which mean that the proposed method is more flexible with different datasets types than classical methods. The proposed method adapt to the complexity of datasets to provide the best results regardless of the size of the dataset. In some cases, the dataset is chosen with a size more than required, which leads to bad results using classical methods but this problem is avoided by using the proposed method since it focuses on the complexity of the problem to be solved regardless the size of the dataset.

## 5 Conclusion

It is noticeable that Pattern Recognition plays a significant role in the determination of the optimal structure of Multi-layer Perceptron using the proposed method. The proposed method makes the design of Multi-layer Perceptron unsupervised and helps to dispense with the need for designer experience and the waste of time using trial and error methods. By clustering the training dataset, we can collect a set of parameters useful to determine the structure of Multi-layer Perceptron as independent variables used to determine the regression models of the proposed method. The independent variable Reference Distance has the highest influence on the results compared to other variables. Comparison of the proposed method with classical methods leads us to deduce that the proposed method performs well for the different type of datasets, which mean that is more flexible with different datasets types than classical methods. The proposed method adapt to the complexity of datasets to provide the best results regardless of the size of the dataset.

## References

1. Xie, Y., Fan, X., Chen, J.: Affinity propagation-based probability neural network structure optimization. In: Tenth International Conference on Computational Intelligence and Security (CIS), pp. 85–89. IEEE, November 2014. https://doi.org/10.1109/cis.2014.156
2. Thomas, A.J., Petridis, M., Walters, S.D., Gheytassi, S.M., Morgan, R.E.: On predicting the optimal number of hidden nodes. In: International Conference on Computational Science and Computational Intelligence (CSCI), pp. 565–570. IEEE, December 2015. https://doi.org/10.1109/csci.2015.33

3. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006). ISBN 978-1-4939-3843-8
4. Pan, H., Liang, D., Tang, J., Wang, N., Li, W.: Shape recognition and retrieval based on edit distance and dynamic programming. Tsinghua Sci. Technol. **14**(6), 739–745 (2009). https://doi.org/10.1016/S1007-0214(09)70144-0
5. Amiri, S.S., Mottahedi, M., Asadi, S.: Using multiple regression analysis to develop energy consumption indicators for commercial buildings in the US. Energy Build. **109**, 209–216 (2015). https://doi.org/10.1016/j.enbuild.2015.09.073
6. Dora, S., Sundaram, S., Sundararajan, N.: A two stage learning algorithm for a growing-pruning spiking neural network for pattern classification problems. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE, July 2015. https://doi.org/10.1109/ijcnn.2015.7280592
7. Sheela, K.G., Deepa, S.N.: Review on methods to fix number of hidden neurons in neural networks. Math. Prob. Eng. (2013). http://dx.doi.org/10.1155/2013/425740
8. Berry, M.J., Linoff, G.: Data Mining Techniques: For Marketing, Sales, and Customer Support. Wiley, New York (1997). ISBN 0471179809
9. Esfe, M.H., et al.: Thermal conductivity of Cu/TiO2–water/EG hybrid nanofluid: experimental data and modeling using artificial neural network and correlation. Int. Commun. Heat Mass Transfer **66**, 100–104 (2015). https://doi.org/10.1016/j.icheatmasstransfer.2015.05.014
10. Vinod, V.V., Ghose, S.: Growing nonuniform feedforward networks for continuous mappings. Neurocomputing **10**(1), 55–69 (1996). https://doi.org/10.1016/0925-2312(95)00024-0
11. Faraway, J.J.: Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models, vol. 124. CRC Press, Boca Raton (2016)
12. Dangeti, P.: Statistics for Machine Learning. Packt Publishing Ltd, Birmingham (2017)
13. Brown, S.H.: Multiple linear regression analysis: a matrix approach with MATLAB. Alabama J. Math. **34**, 1–3 (2009)
14. Austin, P.C., Steyerberg, E.W.: The number of subjects per variable required in linear regression analyses. J. Clin. Epidemiol. **68**(6), 627–636 (2015). https://doi.org/10.1016/j.jclinepi.2014.12.014
15. Sasaki, T., Kinoshita, K., Kishida, S., Hirata, Y., Yamada, S.: 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: IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS), pp. 374–379. IEEE, September 2011. https://doi.org/10.1109/iccis.2011.6070358
16. Naseem, I., Togneri, R., Bennamoun, M.: Linear regression for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **32**(11), 2106–2112 (2010). https://doi.org/10.1109/TPAMI.2010.128
17. Pozo, F., Vidal, Y.: Wind turbine fault detection through principal component analysis and statistical hypothesis testing. Energies **9**(1), 3 (2015). https://doi.org/10.3390/en9010003
18. Cohen, P., West, S.G., Aiken, L.S.: Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. Psychology Press, New York (2014). ISBN 9781135468255
19. Wang, W., Morrison, T.A., Geller, J.A., Yoon, R.S., Macaulay, W.: Predicting short-term outcome of primary total hip arthroplasty: a prospective multivariate regression analysis of 12 independent factors. J. Arthroplasty **25**(6), 858–864 (2010). https://doi.org/10.1016/j.arth.2009.06.011

20. Ghaedi, M., Reza Rahimi, M., Ghaedi, A.M., Tyagi, I., Agarwal, S., Gupta, V.K.: Application of least squares support vector regression and linear multiple regression for modeling removal of methyl orange onto tin oxide nanoparticles loaded on activated carbon and activated carbon prepared from Pistacia atlantica wood. J. Colloid Interface Sci. **461**, 425–434 (2016). https://doi.org/10.1016/j.jcis.2015.09.024
21. Chatterjee, S., Hadi, A.S.: Regression Analysis by Example. Wiley, New York (2015)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Parallelizing neural networks during training. U. S. Patent 9,811,775, Google Inc. (2017)
23. Bouguettaya, A., Yu, Q., Liu, X., Zhou, X., Song, A.: Efficient agglomerative hierarchical clustering. Expert Syst. Appl. **42**(5), 2785–2797 (2015). https://doi.org/10.1016/j.eswa.2014.09.054
24. Ng, M.K., Li, M.J., Huang, J.Z., He, Z.: On the impact of dissimilarity measure in k-modes clustering algorithm. IEEE Trans. Pattern Anal. Mach. Intell. (3), 503–507 (2007). http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.53
25. Karypis, G., Han, E.H., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. Computer **32**(8), 68–75 (1999). https://doi.org/10.1109/2.781637
26. Murtagh, F., Contreras, P.: Algorithms for hierarchical clustering: an overview, II. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **7**(6), e1219 (2017). https://doi.org/10.1002/widm.1219
27. Dalbouh, H.A., Norwawi, N.M.: Improvement on agglomerative hierarchical clustering algorithm based on tree data structure with bidirectional approach. In: Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pp. 25–30. IEEE, February 2012. https://doi.org/10.1109/isms.2012.13
28. Aggarwal, C.C., Reddy, C.K. (eds.): Data Clustering: Algorithms and Applications. CRC Press, Boca Raton (2013). ISBN 1466558210, 9781466558212
29. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 773–780 (1989). https://doi.org/10.1109/34.192473
30. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for R. Bioinformatics **24**(5), 719–720 (2007). https://doi.org/10.1093/bioinformatics/btm563
31. Zhao, Z., Xu, S., Kang, B.H., Kabir, M.M.J., Liu, Y., Wasinger, R.: Investigation and improvement of multi-layer perceptron neural networks for credit scoring. Expert Syst. Appl. **42**(7), 3508–3516 (2015). https://doi.org/10.1016/j.eswa.2014.12.006
32. Raghuvanshi, A.S., Tiwari, S., Tripathi, R., Kishor, N.: Optimal number of clusters in wireless sensor networks: an FCM approach. In: International Conference on Computer and Communication Technology (ICCCT), pp. 817–823. IEEE, September 2010. https://doi.org/10.1109/iccct.2010.5640391
33. Wang, L.C., Wang, C.W., Liu, C.M.: Optimal number of clusters in dense wireless sensor networks: a cross-layer approach. IEEE Trans. Veh. Technol. **58**(2), 966–976 (2009). https://doi.org/10.1109/TVT.2008.928637
34. Liu, X., Croft, W.B.: Experiments on retrieval of optimal clusters. Technical report IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts (2006)
35. Kumar, V., Chhabra, J.K., Kumar, D.: Performance evaluation of distance metrics in the clustering algorithms. INFOCOMP **13**(1), 38–52 (2014)
36. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE, September 2015. https://doi.org/10.1109/mlsp.2015.7324337
37. Lillicrap, T.P., Cownden, D., Tweed, D.B., Akerman, C.J.: Random synaptic feedback weights support error backpropagation for deep learning. Nature Commun. **7**, 13276 (2016). https://doi.org/10.1038/ncomms13276