APPLYING NEURAL NETWORKS FOR SUPERVISED LEARNING OF MEDICAL DATA

Ourida Ben Boubaker

Department of Computer Science, College of Science and Arts, Tabarjal, Jouf University, Kingdom of Saudi Arabia

ABSTRACT

Constructing a classification model based on some given patterns is a form of learning from the environment perception. This modelling aims to discover new knowledge embedded in the input observations. Learning behaviour of the neural network model enhances the classification properties. This paper considers artificial neural networks for learning two different medical data sets in term of number of instances. The experiment results confirm that the back-propagation supervised learning algorithm has proved its efficiency for such non-linear classification issues.

KEYWORDS

Supervised learning, Artificial Neural Networks, Artificial intelligence, Learning, Classification

1. INTRODUCTION

Introducing cognitive reasoning into a conventional computer can solve problems by mapping patterns by recognition [1], classification and forecasting [2]. Artificial Neural Networks provide these functions. They are essentially a mathematical modelling based on some function. They are associated with particular learning algorithms or rules to emulate human actions. Artificial Neural Networks have proved their interest in dealing with medical recognition data [3].

The overall organization of the paper is as follows. The section 2. Outlines the various learning methods. The section 3. Deals with some of the algorithms presented in the literature for supervised learning and more specifically Neural Networks. In section 4. We discuss the end results of this algorithm on two different data sets. And finally the last section concludes with a final discussion and proposes some research lines for further studies.

2. LEARNING METHODS

Learning methods can be classified as supervised, unsupervised and reinforcement models. Unsupervised learning model groups observation based on pattern information deduced from algorithm heuristics. Reinforcement learning learns through trial and error interactions with its environment (reward/penalty assignment). Whereas, supervised learning modelling assumes the availability of a teacher or supervisor who classifies the training examples into a given number of classes using the information on training instances membership.

2.1. Unsupervised Learning

Some models learn using unsupervised learning algorithms in order to identify hidden patterns in input data. The property unsupervised refers to the ability to learn and classify data instances without producing an error signal used to calculate the correctness level of the classification of a new element. The lack of direction for the learning algorithms in unsupervised learning can sometime be advantageous, since it lets the algorithm look after patterns that have not been previously considered.

2.2. Reinforcement Learning

In reinforcement learning the network is associated to a logical or a real value after executing a number of actions referring to whether the results have been rewarding or inversely. Algorithms specifies how an artificial agent can learn to select actions in order to maximize the total expected reward. Intuitively, this mechanism is shown to be more reliable than unsupervised learning since it is based on specific criteria for problem-solving.

2.3. Supervised Learning

Based on a training set holding instances already labelled, the supervised learning matches a new potential item to its more likely correct class. Such techniques can be split into two processing groups (fig 1.): First the K-nearest neighbours algorithms [4], KNN for short, which consist on grouping data inputs into sets based on similarity measures that forward will be used to store new cases. Second, Artificial Neural Networks [5], shortly ANN. As well as for our study, for this class of algorithms, the training set is composed of input instances along with their correct assignments representing the precise activation values of the corresponding output neurons. The error measure, which shows the difference between the network output and the output from the training samples, is used to guide the learning process. Different algorithms for solving supervised learning problems have been introduced throughout the years. Some are dedicated to continuous variables: K-means [6], Decision Trees [7], Random Forest [8]. Others for categorical ones: Logistic regression [9], Naive Bayes [10], Support Vector Machine [11], K-Nearest Neighbors [12]... The following article [13] presents a deeper study of these algorithms.

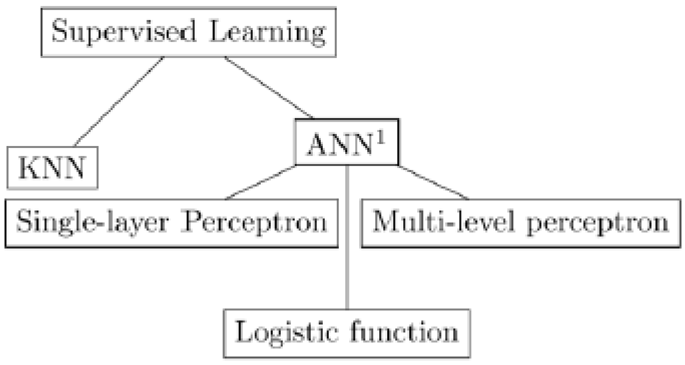


Figure 1. Supervised learning methods

3. Algorithms for Unsupervised Learning

3.1. K-Nearest Neighbours

KNN is a simple algorithm that stores and learns from all available cases in order to classify new ones based on a similarity measure which is a distance function. Most known distance measure functions are the Euclidean distance (1), the Manhattan distance (2) and the Minkowski distance (3). Let us consider the element *X* characterized by the attributes $x_1, x_2 \dots x_n$ and Y characterized by $y_1, y_2 \dots y_n$.

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

$$d(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \qquad (2)$$

$$d(X,Y) = \left(\sqrt{\sum_{i=1}^{n}(|x_i - y_i|)^q}\ \right)^{\frac{1}{q}} \qquad (3)$$

3.2. Artificial Neural Network

The artificial neural networks got their name from biological counterparts about the activity and modelling of the human brain mechanism. Actually, the brain consists of approximately 1011 of highly connected elements. Each composed by approximately 104. These structures are called neurons and are composed of three principal components illustrated by the Figure 2. namely, the dendrites, the cell body and the axon. Short branched extensions of the nerve cell referred to as dendrites, transmit the received electrical signals from synapses to other body cells. The incoming signals are thus summed by the cell body. The axon represents the long thread-like part of a nerve cell along which electrical signals are conducted from the cell body to other cells. A synapse corresponds to the junction between two cells, more explicitly, between the axon of a first one and the dendrite of the another. The neural network function is determined by a complex chemical process induced by the positioning of neurons and the power of the individual synapses. The Figure 2. illustrates a simplified schematic representation of two biological neurons. One of the neural networks categories is the Self-Organizing neural network which represents a variant that uses unsupervised learning methods [14].

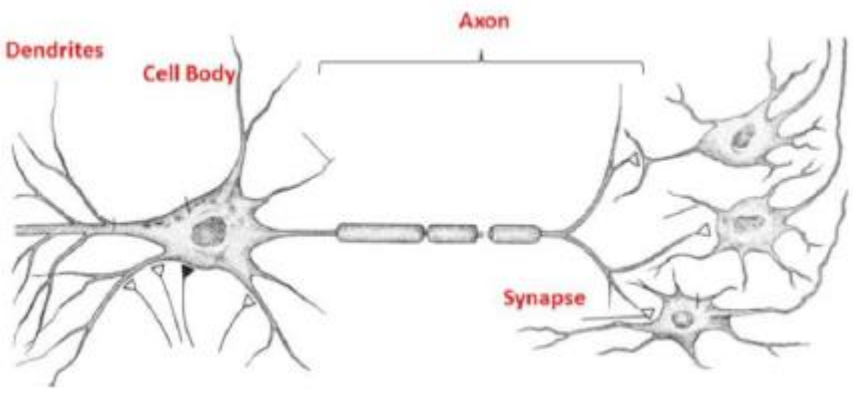


Figure 2. Essential components of a neuron

3.2.1. Graphical Modelling

A neural network is a sorted triple (N, V, w) where the tuple N, V refers to the set of neurons and their respective associated values. A network connection weight between neurons *i* and *j* is shortened to w_{ij}. As shown in Figure 3. an artificial neural network is composed of three layers: First, the input layer where each neuron *i* represents an instance or an attribute. Second, the hidden or intermediate layer composed of a set of neurons completely connected to output neurons which represent the output layer. Neurons in a layer are not connected with each other, but completely connected to neurons from the next layer.

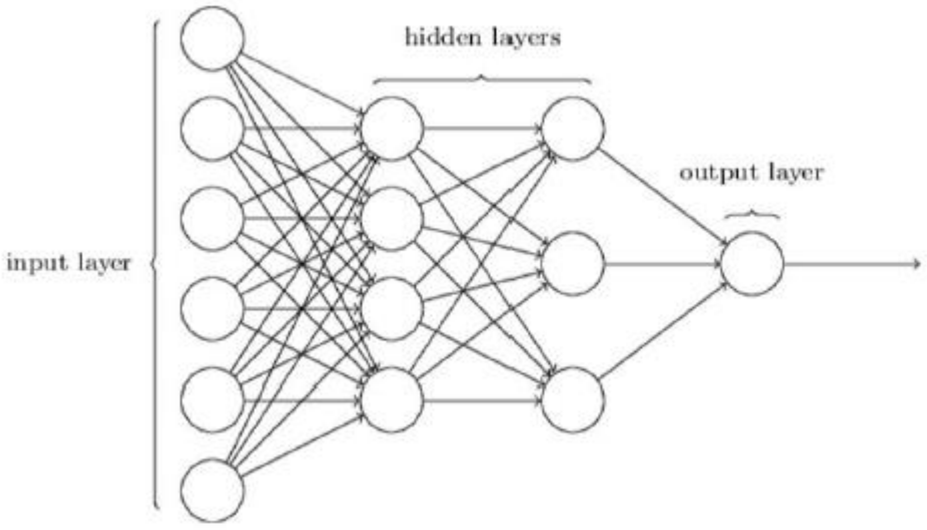


Figure 3. Artificial neurone network Structure

3.2.2. Principle

Looking at a neuron j, we will usually find a lot of neurons which transfer their output to j. Each neuron j computes three equations: The propagation function, the activation function and the output function [15]. In a first step, the propagation function of a neuron j collect outputs of the relevant neurons connected to j. In a second step, taking into consideration the corresponding weight values w_{ij}, the function calculates the input of the next neuron, which is further propagated using the activation function. Thus, the network activation input a_k of the neuron *k* is the result of the propagation function that can be calculated using the equation below:

$$a_k = Logistic(\sum_{j=1}^{N} w_{jk} a_j) \qquad (4)$$

Logistic is a function that expresses the non-linearity distribution of a data set *x*. It represents the activation or transfer function. It is a mapping from a value scale [0,1] to the hyperbolic tangent going from -1 to 1. Both functions are differentiable. It is formulated as follows:

$$Logistic(x) = \frac{1}{1-exp^{-x}} \qquad (5)$$

The threshold value is expressed by means of the logistic function. Each neuron j has its unique threshold value which indicates the maximum variation value of the activation function. It is a linear function that takes the value 1 if the input z is different of 0 and 0 otherwise. The received values transferred to neurons connected to a neuron j are calculated using on the output function of j based on its activation state a_j. Often this function is the activation a_j that directly refers to the output of the neuron j, unless explicitly specified.

3.2.3. Learning Strategy

The learning strategy permit to the network to produce the desired output for a given input. The problem can be transformed into an optimization based learning process using the back propagation learning. Based on the relationship between the actual output of a neuron and the correct output for a given training example, the algorithm considers internal representations in order to deduce a mapping for any arbitrary input. Iteratively, it considers a simple neural network with two input units, one output unit and no hidden units, where each neuron uses a linear that is the weighted sum of its input. Details of this process are given by the Algorithm 1.

```
function BACK-PROP-LEARNING(examples, network) returns a neural network
  inputs: examples, a set of examples, each with input vector x and output vector y
          network, a multilayer network with $L$ layers, weights $w_{i,j}$, activation function $g$
  local variables: $\Delta$, a vector of errors, indexed by network node

  for each weight $w_{i,j}$ in network do
      $w_{i,j} \leftarrow$ a small random number
  repeat
     for each example (x, y) in examples do
        /* Propagate the inputs forward to compute the outputs */
        for each node $i$ in the input layer do
            $a_i \leftarrow x_i$
        for $\ell = 2$ to $L$ do
            for each node $j$ in layer $\ell$ do
                $in_j \leftarrow \sum_i w_{i,j} a_i$
                $a_j \leftarrow g(in_j)$
        /* Propagate deltas backward from output layer to input layer */
        for each node $j$ in the output layer do
            $\Delta[j] \leftarrow y_j - a_j \quad (= -\partial Loss/\partial in_j)$
        for $\ell = L - 1$ to 1 do
            for each node $i$ in layer $\ell$ do
                $\Delta[i] \leftarrow g(in_i)(1 - g(in_i)) \sum_j w_{i,j} \Delta[j]$
        /* Update every weight in network using deltas */
        for each weight $w_{i,j}$ in network do
            $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
  until some stopping criterion is satisfied
  return network
```

Algorithm 1. Back Propagation algorithm

Data inputs consider separable and non-separable instances. Single Layer Perceptron and the logistic function classify data into two categories, Whereas, Multi-layer Perceptron aims to resolve the non-linearity separation of data sets. Supervised learning paradigm of an ANN is efficient and finds solutions to several linear and nonlinear problems.

Perceptron

Single neuron Perceptron [16] can classify input data into two clusters. It models the problem by use of a linear function and a threshold. The algorithm is presented in the following:

```
For each vector($x_{t};y_t$) $\epsilon$ D

    a. Compute
       $h_w(x_j)$=Threshold(w.$x_j$)
    b. If $y_t \neq h_w(x_t)$
       $w_i \leftarrow w_i + \alpha(y_j - h_w(x_j))x_{j,i}$ $\forall_i$

Repeat 1. until a stop criteria is
reached (e.g. stepmax, err=0)
```

Algorithm 2. Perceptron algorithm

Logistic Function

The Threshold function devises data inputs into two categories. It assumes that they are separable by mean of a linear function. However, sometimes inputs can't be divided into two groups. As a solution, the logistic function, presented in subsection 3.2.2, by the equation (5) deals with non-linear separable input vectors. Note that the threshold function is not differentiable at the threshold and for the rest the derivate to 0. Due to this fact, back-propagation learning is impossible.

Multi-layer Perceptron

Multi-layer Perceptron found as a solution to represent non-linearly separable functions. The learning problem can be formulated as an optimization problem, where for each training set, we are aiming to minimize a distance function *Loss* $(y_j,h_w(x_j))$ between the actual output of the output neuron y_j and the predicted one $h_w(x_j)$., formally denoted by the equation (6). For a correct prediction it equals 0, otherwise it is assigned to the difference between $w.x_j$ and the threshold for which the prediction is true.

$$Loss(y_j, h_w(x_j)) = -(y_j, h_w(x_j))w.x_j \qquad (6)$$

4. Experimental Study

4.1. Technical Environment

For our study, a 4GB memory computer has been used, equipped with an Intel(R) Pentium(R) CPU B960, delivered from Dell company computer. The language and environment R has been used for the statistical computations.

4.2. Data Set Description

Data set 1. Pen-Based Recognition of Handwritten Digits is a database referring to 250 diagnoses samples. 68% of observations are used for training, cross- validation, while the remaining 32% are used for independent testing. The data base holds 16 integer variables laying from 0 to 100. The last variable refers to the class code which goes from 0 to 9. The total number of instances is 10992. They are divided into two sets: a training set used for learning and a testing set. The data set do not hold uncertain values.

As a second data base ***data set 2. Optical Recognition of Handwritten Digits*** data base holds 64 attributes, all are integers in the range [0,16]. The last attribute represents the class code that lays from 0 to 9. The total number of instances equals 10992. They are divided into a training set of 3823 instances, used for learning, and a testing set of 1797 instances. The data set do not hold missing attributes.

4.3 EXPERIMENTATION

4.3.1 Experimental Processing

The range value of all attributes goes from 0 to 100. It first needs to be normalized in order to be treated.

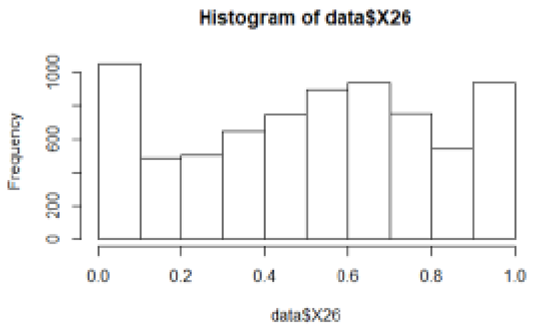
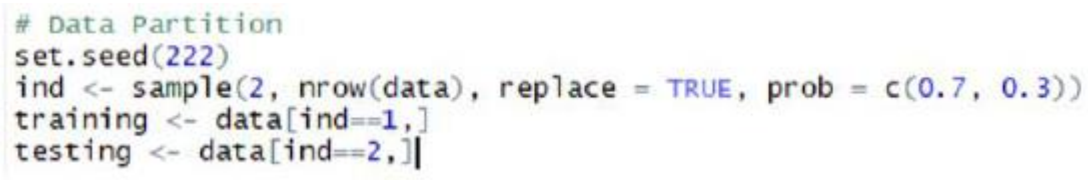


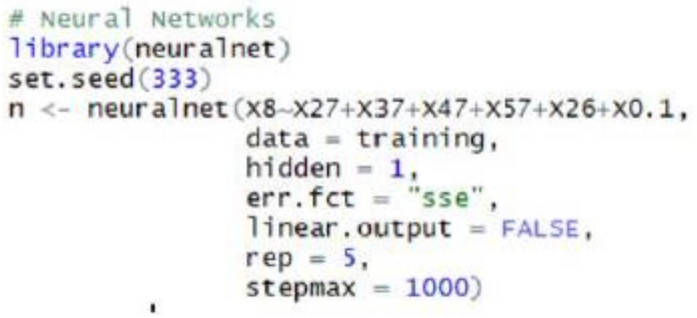
Figure 3. Histogram of normalized attributes X26

In a second step, the data is split into two parts; the first one representing 70% of the data set is used for training, the rest for testing.



```
# Data Partition
set.seed(222)
ind <- sample(2, nrow(data), replace = TRUE, prob = c(0.7, 0.3))
training <- data[ind==1,]
testing <- data[ind==2,]
```

After downloading neural net package, we specify the training set data and compute the neural network while varying the number of hidden layers, the number of nodes for each hidden layer and the number of repetition. The error function can either be cross entropy for binary output, the sum of squared errors otherwise.



```
# Neural Networks
library(neuralnet)
set.seed(333)
n <- neuralnet(X8~X27+X37+X47+X57+X26+X0.1,
               data = training,
               hidden = 1,
               err.fct = "sse",
               linear.output = FALSE,
               rep = 5,
               stepmax = 1000)
```

4.3.2 Experimental Results

Data set 1. Pen-Based Recognition of Handwritten Digits: The Table1. shows the results of neural network's computation. The minimum induced error is shown to be assigned to 5 neurons for one hidden layer. Further computations have been studied and no one has shown better results. The number of layer and the number of neurons for each layer have been varied. We notice that the error function continues to decrease, until at some point it begins to increase. That point represents the best average of error above which the neural network is over-fitting learning. Resulting network is illustrated in the figure 3.

n^o neurons	1	2	3	4	5	6
n^o steps	70	51	38	35	24	22
n^o repetitions	9	1	1	4	7	2
exec. time(s)	0.16	0.18	0.18	0.21	0.15	0.15
err	.5083	.50778	.50701	.50657	**.50577**	.50644

Table 1. Detailed parameter values for the best reduced error rate of Data set 1.

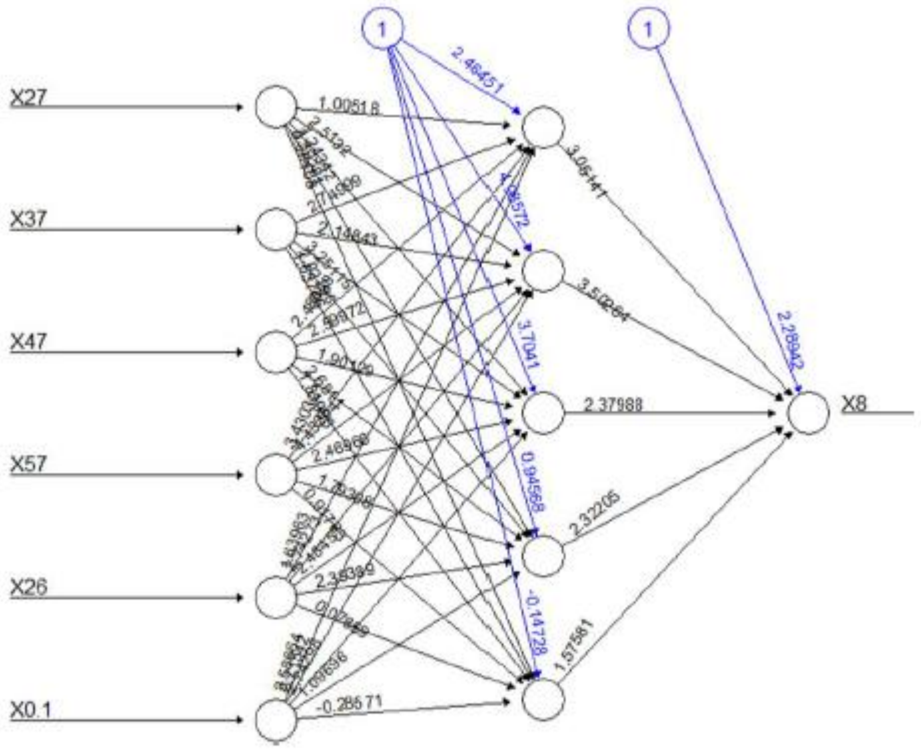


Figure 3. Resulting Neural Network for the *Pen-Based Recognition of Handwritten Digits data set*

Data set 2. Optical Recognition of Handwritten Digits: As for the second data set, the neural network starts over fitting learning for a number of neuron by a single hidden layer equal to 3. Further computations have also been studied and no one has shown better results. Computational details and results are given by the Table 2. and the Figure 4.

n^o neurons	1	2	3	4
n^o steps	84	51	37	37
n^o repetitions	5	2	3	8
exec. time(s)	0.16	0.12	0.09	0.14
err	.50825	.50431	**.50399**	.50448

Table 2. Detailed parameter values for the best reduced error rate of Data set 1.

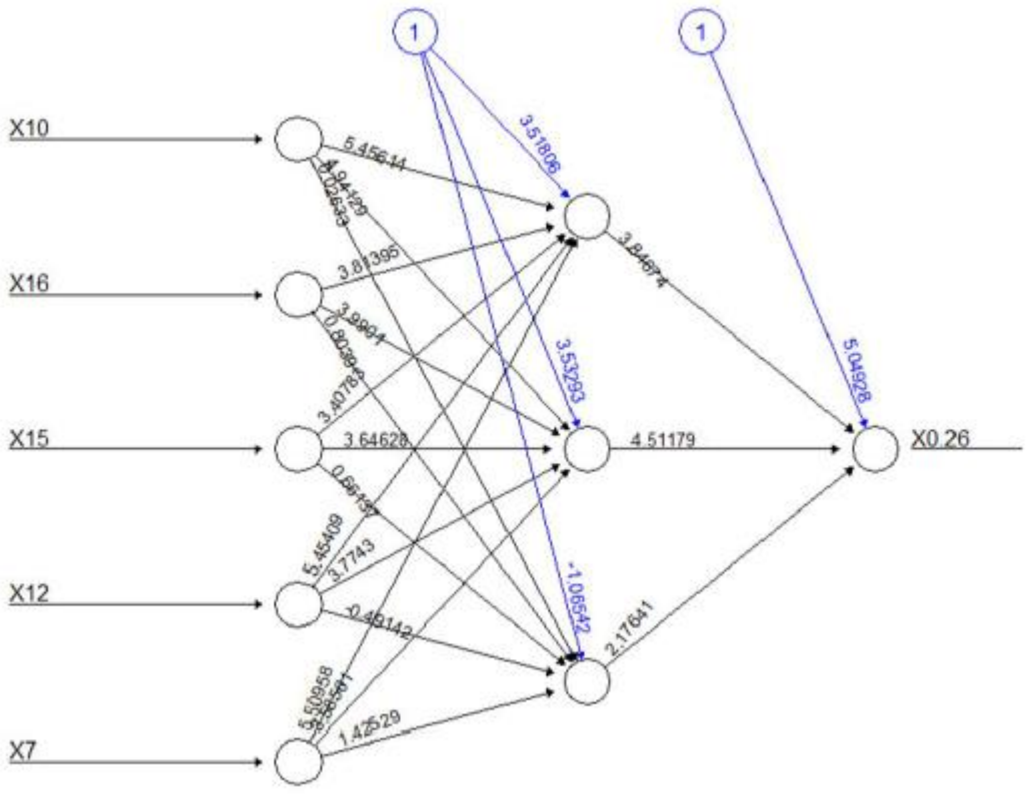


Figure 4. Resulting Neural Network for the Optical Recognition of Handwritten Digits data set

5. Concluding Remarks

The experiments have shown that the execution time of an ANN increases while increasing the number of neurons and layers. However, the number of steps in order for the algorithm to converge decreases while increasing the number of neurons in the hidden layer. For further studies, we can consider other measures of error function. The algorithm for back propagation learning can be adapted to handle uncertain data.

References

[1] S. Nandini, S. Md, S. Goutam. (2016). "Lung sound classification using cepstral-based statistical features". Computers in Biology and Medicine. Vol. 75, No. 1, pp 118–129.

[2] E. Díaz, V. Brotons, R. Tomás. (2018). "Use of artificial neural networks to predict 3-D elastic settlement of foundations on soils with inclined bedrock". Soils and Foundations. Vol. 58, No. 6, pp 1414–1422.

[3] K. Usha Rani. (1992). "Analysis of heart diseases dataset using neural network approach". IJDKP. Vol. 1, No. 5, pp 175–185.

[4] N.S. Altman. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. Vol. 46, No. 3, pp 175–185.

[5] Mc., Warren, W. Pitts. (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. Vol. 5, No. 4, pp 115–133.

[6] Forgy, E. (1965) Cluster analysis of multivariate data: efficiency vs. interpretability of classification, Biometrics, Vol.21.

[7] J. Ross Quinlan. (1990). "Decision trees and decision-making". IEEE Trans. Systems, Man, and Cybernetics Vol. 20, No. 2, pp 339-346.

[8] L. Breiman. (2001). "Random Forests". Machine Learning. Vol. 45, No 1, pp 5-32.

[9] D. R. Cox. (1958). "The Regression Analysis of Binary Sequences". Journal of the Royal Statistical Society. Series B (Methodological), Vol. 20, No. 2, pp. 215-242.

[10] Pearl, J. (1986). " Probabilistic reasoning using graphs". IPMU. pp 200-202.

[11] C. Cortes, V. Vapnik. (1995). "Support-Vector Networks". Machine Learning. Vol. 20, No 3, pp 273-297.

[12] T. M. Cover, P. E. Hart. (1967). "Nearest neighbor pattern classification". IEEE Trans. Information Theory. Vol. 13, No 1, pp 21-27.

[13] S. M. Al-Tabbakh, H. M. Mohamed, H. El-Zahed. (2018). "Machine Learning Techniques for Analysis of Egyptian Flight Delays". IJDKP. Vol.8, No.3

[14] K. Fukushima. (1979). "Self-Organization of a Neural Network which Gives Position-Invariant Response". IJCAI. pp 291-293.

[15] J.S. Hesthaven, S. Ubbiali. (2018). "Non-intrusive reduced order modeling of nonlinear problems using neural networks". Journal of Computational Physics. Vol. 363, pp. 55-78.

[16] F. Rosenblatt. (1957), "The Perceptron--a perceiving and recognizing automaton". Report 85-460-1.

AUTHOR

Ourida Ben Boubaker Saidi Ph. D in Computer Science from the Paul Sabatier University of Toulouse, France. Ex-head of the Department of Computer Science in the ISG of Tunis, University of Tunis, a member of SOIE research Laboratory, and actually Coordinator of the Department of Computer Science, College of Science and Arts, Tabarjal, Jouf University, Kingdom of Saudi Arabia. Her research interests are Artificial Intelligence, Information Systems Security and Data mining.