Training Algorithms for Supervised Machine Learning: Comparative Study

Dr. Rafiqul Zaman Khan¹, Md. Haider Khalaf Jabbar²

¹Associate Professor,
Department of Computer Science,
Aligarh Muslim University, Aligarh-202002 (U.P), INDIA
rzk32@yahoo.co.in, rzkhan.cs@mail.amu.ac.in

²Research Scholar,
University of Misan, Misan, Iraq and
Department of Computer Science
Aligarh Muslim University, Aligarh-202002 (U.P), INDIA
haider.allamy@gmail.com, he25_allamy@yahoo.com

ABSTRACT
Supervised machine learning is an important task for learning artificial neural networks; therefore a demand for selected supervised learning algorithms such as back propagation algorithm, decision tree learning algorithm and perceptron algorithm has been arise in order to perform the learning stage of the artificial neural networks. In this paper, a comparative study has been presented for the aforementioned algorithms to evaluate their performance within a range of specific parameters such as speed of learning, overfitting avoidance, and their accuracy. Besides these parameters we have included their benefits and limitations to unveil their hidden features and provide more details regarding their performance. We have found the decision tree algorithm is the best as compared with other algorithms that can solve the complex problems with a remarkable speed.

Keywords
artificial neural networks, supervised learning, back propagation, Perceptron, Decision Tree learning algorithm.

Academic Discipline And Sub-Disciplines
Computer science, Expert System and Artificial Intelligence

SUBJECT CLASSIFICATION
Artificial Intelligent, Machine Learning

TYPE (METHOD/APPRAOCH)
Comparative Study
2. LEARNING

The ability to learn new knowledge is a basically trait of intelligence. For learning network, we need to know two things: first we must have a model of environment in which neural network will be operated that means we must know the learning paradigm and second, we must know how the weight of neural network will update and by which rule the weight will be updated [14]. There are three ways to learn neural network (learning paradigm); supervised learning, unsupervised learning and hybrid learning. Each paradigm is working for different tasks based on one of the rules: Error Correction, Boltzmann, Hebbian and Competitive learning [1] [14]. The details are explained and showing in the Table 1.

Table 1. Rule of learning and Learning Algorithms for machine learning

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Rule of learning</th>
<th>Learning Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>Error-correction</td>
<td>Perceptron learning algorithm, back-propagation &amp; Adaline, madeleine.</td>
</tr>
<tr>
<td></td>
<td>Boltzmann</td>
<td>Boltzmann learning algorithm</td>
</tr>
<tr>
<td></td>
<td>Hebbian</td>
<td>Linear discriminant analysis</td>
</tr>
<tr>
<td></td>
<td>Competitive</td>
<td>Learning vector quantization</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>Error-correction</td>
<td>Sammon’s projection</td>
</tr>
<tr>
<td></td>
<td>Hebbian</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td></td>
<td>Competitive</td>
<td>Associative memory learning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vector quantization &amp; Kohonen’s SOM</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Error-correction &amp; competitive</td>
<td>RBF learning algorithm</td>
</tr>
</tbody>
</table>

2.1 Supervised Learning

Supervised learning is a machine learning technique that uses to train the artificial neural networks by the value of input and output by finding the relationship between the input (same time call independent variable) and the output (same time call dependent variable), that means making map between input and output for the artificial system that want to learn. Supervised can predict the value of output for system given new input [1] [12] [10] [16]. Supervised learning is a method for training artificial neural networks when the input and output is known, we use supervised learning when we have a set of training data, this training data is a collection of few input data that connected with few output data, this training data it will use by learning algorithms like back propagation and perceptron, which we have discussed about them in section 3 and 4 [3] [6] [7] [12]. The process of training will be done when the neural network will use the input data to produce its own output data and then compares this with the desired output data or the target output data. If there is no difference between an actual output data and the target output data, that means the learning for this data is completed and no need to continue the training for this data, otherwise, the connection weights will be changed by the algorithm of training to reduce the difference between the input and target output data (see fig 1).

![Fig 1: The process of supervised learning](image)
3. BACK-PROPAGATION

The perceptron that uses in today’s artificial neural networks was conceived by Rosenblatt in 1950 and he proposed it in 1962 [3] [15]. The back-propagation or (back error-propagation) is the most commonly adopted Multi Layers Perceptron (MLPs) training algorithm, it was first presented by Werbos in 1974 and then it was independently re-invented in 1986 by Rumelhart et al. Neural networks are a mathematical model inspired by biological neural networks that shows the set of connected input/output units in which each connection has a weight associated with it, this algorithm is one of the neural used supervised learning method. The learning of neural network will be done by adjusting the weight of neural network [13] [15] [8]. Back propagation uses the target values to calculate the mean square error of the artificial neural networks and level of individual in a population [3] [4] [6].

![Fig 2: Architecture of back propagation (MLP)](image)

The back propagation algorithm is the most used algorithms for learning in neural network especially for Multi Layers Perceptron (MLPs) [8].

The algorithm is summarized as follow [18] [19]:

1. Initialize all weights for the neural network and bias.
2. Select the input data to the neural network and apply to input layer of network.
3. Activation of the units of the output layer through the network.
4. Calculate the error for the neuron. We mean by errors are the deference between the desired output and neural output.
   \[ \text{Error}_b = \text{Output}_b (1 - \text{Output}_b)(\text{target}_b - \text{Output}_b) \quad (1) \]
   Calculate what is the output should have been and a scaling factor, how much lower or higher. The output must be adjusted to match the desired output.
5. Reduce the error by changing the weight let \( W'_{AB} \) is the new (training) and \( W_{AB} \) is initial weight.
   \[ W'_{AB} = W_{AB} + (\text{ERROR}_b \cdot \text{Output}_b) \quad (2) \]
   Step 6: Finding the error for the hidden layer. Also change the weights for the hidden layers.
   \[ \text{Error}_h = \text{Output}_h (1 - \text{Output}_h)(\text{Error}_b W_{AB} - \text{Error}_c W_{AC}) \quad (3) \]
6. Repeat the steps from step 2 to step 6 to train a network of any number of layers.

4. PERCEPTRON ALGORITHM

Perceptron algorithm has proposed in 1958 by Rosenblatt and its one of the neural networks algorithms that are based on error correction rules. Error correction is using the error signal (d-y) to modify the connection weight to gradually reduce the error. Perceptron consists of a single neuron with adjustable weight, \( w_{ij} = 1,2,..,n \), and threshold \( U \), as shown in (Figure 3) given an input vector \( x= (x_1, x_2,..,x_n) \) the net input to neuron is:

\[ V = \sum_{i=0}^{n} W_i x_i - U \quad (4) \]
The output is +1 if \( V > 0 \) and 0 otherwise [14]. The benefit of perceptron algorithm is its low computational requirement and its ability to guarantee convergence to a solution for linearly separable problem [11].

\[ n \]
\[
\sum_{i=1}^{n} \quad \text{h} \quad \text{y}
\]

Fig 3: Architecture of Perceptron

Perceptron learning algorithm is summarized as follow [14]:

1. Initialize the weight and threshold to small random number.
2. Present a pattern vector \((x_1, x_2, \ldots, x_n)^T\) and evaluate the output of the neuron.
3. Update the weight according to:

\[
w_j(t + 1) = w_j(t) + \eta (d - uX_j).
\]

Where \( d \) is the desired output, \( t \) is the iteration number, and \( \eta \) (0.0 < \( \eta \) < 1.0) is the gain (step size).

5. DECISION TREE

The algorithm of decision tree is one of the supervised learning algorithms used like technique machine that build a decision tree from a set of class labeled training sample during the machine learning process. The main purpose of decision tree is the structural information contained in data [9]. Decision tree proposed by Quinlan and there are various Top-down decision tree induces such as ID3 (Quinlan 1986), C4.5 (Quinlan 1993), CART (Brevnan et al. 1984). Decision tree are considered to be one of the most popular approaches for representing classification [1] [15].

The decision tree algorithm is summarized as follow [5] [9]:

If all instance at the current node belong to the same category then create a leaf node if the corresponding class

\[
\text{else}
\]

\[
\text{Find the feature A that maximized the goodness measure}
\]

\[
\text{Make A the decision feature for the current node}
\]

\[
\text{for each possible value v of A}
\]

\[
\text{Add a new branch below node testing for A = v}
\]

\[
\text{Instances}_v := \text{subset of instance with A = v}
\]

\[
\text{if Instances}_v \text{ is empty then add a leaf with label the most common value of Target_feature in Instance;}
\]

\[
\text{else}
\]

\[
\text{below the new branch add subtree}
\]

\[
\text{DT(Instances}_v, \text{ Target_feature, Features – \{A\})}
\]

\]
6. COMPARATIVE STUDY

Supervised learning algorithms are used in machine learning for training artificial neural networks. Our comparison subdivided into two sections; first section comparison based on various parameters derived from some theoretical studies showing in table 2 |2| [5] |[9] |[15] |[21] |[22] |[23]. And second section comparison based on benefits and limitations showing in table 3 |14| [15] |[17].

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Back-propagation</td>
</tr>
<tr>
<td>Accuracy in general</td>
<td>Very good</td>
</tr>
<tr>
<td>Speed of learning</td>
<td>Average</td>
</tr>
<tr>
<td>Speed of classification</td>
<td>Excellent</td>
</tr>
<tr>
<td>Tolerance to missing values</td>
<td>Average</td>
</tr>
<tr>
<td>Tolerance to irrelevant attributes</td>
<td>Average</td>
</tr>
<tr>
<td>Tolerance to redundant attributes</td>
<td>Good</td>
</tr>
<tr>
<td>Tolerance to highly interdependent attributes</td>
<td>Very good</td>
</tr>
<tr>
<td>Dealing with discrete/binary/continuous attributes</td>
<td>Not discrete</td>
</tr>
<tr>
<td>Tolerance to noise</td>
<td>Good</td>
</tr>
<tr>
<td>Dealing with danger of overfitting</td>
<td>Average</td>
</tr>
<tr>
<td>Attempts for incremental learning</td>
<td>Good</td>
</tr>
<tr>
<td>Explanation ability/ transparency of knowledge/classification</td>
<td>Average</td>
</tr>
<tr>
<td>Support Multi classification</td>
<td>Naturally extended</td>
</tr>
</tbody>
</table>

Table 3. Comparative study by benefit and limitation between different supervised algorithms

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td></td>
</tr>
<tr>
<td>1. Very simple and fast.</td>
<td>1. It has long training time.</td>
</tr>
<tr>
<td>2. Can handle high dimensional data.</td>
<td>2. Lack of available memory, when dealing with large databases.</td>
</tr>
<tr>
<td>3. Representation is easy to understand.</td>
<td>3. The resulting regions after partitioning are all hyper rectangles.</td>
</tr>
<tr>
<td>4. Have good accuracy.</td>
<td>4. Most algorithms cannot perform well with problems that require diagonal partitioning.</td>
</tr>
<tr>
<td>Back-propagation</td>
<td></td>
</tr>
<tr>
<td>1. Able to tolerate noisy data.</td>
<td>1. Long learning time therefore more suitable for application where this is feasible.</td>
</tr>
<tr>
<td>2. They can be used when we have the little knowledge of the relationship between attributes and classes.</td>
<td>2. Poor interpretability as knowledge is represented.</td>
</tr>
<tr>
<td>3. Well suited for continuous valued inputs and outputs.</td>
<td>3. Require number of parameters that are to be determined empirically.</td>
</tr>
<tr>
<td>4. Successful on several real world applications like handwritten character recognition, pathology and laboratory medicine.</td>
<td></td>
</tr>
<tr>
<td>Perceptron</td>
<td></td>
</tr>
<tr>
<td>1. Noisy or incomplete inputs will be classified because of their similarity with pure and complete inputs.</td>
<td>1. Computationally expensive learning process.</td>
</tr>
<tr>
<td>2. Relearning after damage can be relatively quick.</td>
<td>2. No guaranteed solution.</td>
</tr>
</tbody>
</table>
7. CONCLUSION

In this paper we performed a comparative study on various algorithms of supervised learning and we have found the decision tree algorithm is the best as compared with other algorithms that can solve the complex problems with a remarkable speed. As for the back propagation algorithm is the most used algorithm for learning in neural networks especially for multi layers perceptron’s (MLPs) and it is very good algorithm for solving simple problems and able to tolerate noisy data. While perceptron algorithm can deal with noisy or incomplete inputs because of their similarity with pure and complete inputs. The benefit of perceptron algorithm is its low computational requirement and its ability to guarantee convergence to a solution for linearly separable problem and Relearning after damage can be relatively quick.

8. REFERENCE


[17] Riedmiller, M., Machine Learning: Multilayer Perceptrons, available online : https://www.google.co.in/search?newwindow=1&q=ml.informatik.uni-freiburg.de%2F_media%2Fteaching%2Fs10%2F05_mlps.printer.pdf%E2%80%8E&oq=ml.informatik.uni-freiburg.de%2F_media%2Fteaching%2Fs10%2F05_mlps.printer.pdf%E2%80%8E&gs_l=serp.12...77260.77260.0.78848.1.1.0.0.0.0.301.301.3.1.1.0.0.0...0.0...1c.17.serp.t9Wz3Rv-_.

[18] Pfeifer, R., Damain, and Fuchslin, R., Neural Networks, available online : https://www.google.co.in/search?newwindow=1&q=R.+Pfeifer,+DO.+Domain+and+R.+F%C3%BCchslin+Neural+Networks&spell=1&sa=X&ei=86HjUduzK4WFrAft1YGdW&ved=0CCsQBSgA&biw=1066&bih=523.


Biography Authors:

Dr. Rafiqul Zaman Khan: Dr. Rafiqul Zaman Khan, is presently working as a Associate Professor in the Department of Computer Science at Aligarh Muslim University, Aligarh, India. He received his B.Sc Degree from M.J.P Rohilkhand University, Bareilly, M.Sc and M.C.A from A.M.U. and Ph.D (Computer Science) from Jamia Hamdard University. He has 18 years of Teaching Experience of various reputed International and National Universities viz King Fahad University of Petroleum & Minerals (KFUPM), K.S.A, Itithad University, U.A.E, Pune University, Jamia Hamdard University and AMU, Aligarh. He worked as a Head of the Department of Computer Science at Poona College, University of Pune. He also worked as a Chairman of the Department of Computer Science, AMU, Aligarh. His Research Interest includes Parallel & Distributed Computing, Gesture Recognition, Expert Systems and Artificial Intelligence. Presently 04 students are doing PhD under his supervision. He has published about 35 research papers in International Journals/Conferences. Names of some Journals of repute in which recently his articles have been published are International Journal of Computer Applications (ISSN: 0975-8887), U.S.A, Journal of Computer and Information Science (ISSN: 1913-8989), Canada, International Journal of Human Computer Interaction (ISSN: 2180-1347), Malaysia, and Malaysian Journal of Computer Science(ISSN: 0127-9084), Malaysia. He is the Member of Advisory Board of International Journal of Emerging Technology and Advanced Engineering (IJETAE), Editorial Board of International Journal of Advances in Engineering & Technology (IJAET), International Journal of Computer Science Engineering and Technology (IJCSET), International Journal in Foundations of Computer Science & technology (IJFCST) and Journal of Information Technology, and Organizations (JITO).

Haider Khalaf Jabbar Allamy: Haider allamy, is a research scholar in the Department of Computer Science; Aligarh Muslim University, Aligarh, India. He joined to his Ph.D course in 28-09-2012. His research interest includes Artificial Intelligence and Expert system. He did B.Sc Degree in computer science from University of Basrah, College of Science, Iraq, M.Sc in computer science from Jamia Hamdard University, Delhi, India. He is envoy by the University of Maysan/Iraq for complete his Ph.D course in Aligarh Muslim University, Aligarh, India.