An Evaluation of Classification Accuracy in a Multilayer Perceptron

Hiroaki Yui
The University of Aizu
Database Systems Laboratory
Ikki-machi, Aizu-Wakamatsu, Fukushima
965-8560
hiroakiyui@gmail.com

Subhash Bhalla
The University of Aizu
Database Systems Laboratory
Ikki-machi, Aizu-Wakamatsu, Fukushima
965-8560
bhalla@u-aizu.ac.jp

ABSTRACT
In data mining, there are a number of algorithms for classification, association rule, clustering and regression. In this article, we classify the data sets into each instance by a Multilayer Perceptron (MLP), which is one of artificial neural networks. The algorithm precisely classifies the data sets as an instance. We evaluate the classification accuracies by changing various hidden layers in the MLP. Also we compare the classification accuracies by the MLP and the other classifiers.

Categories and Subject Descriptors
H.2.8 [Database Applications]: Data mining

General Terms
Machine Learning

Keywords
Data mining, Neural network, Perceptron, Neuron, Sigmoid function, C4.5, Machine learning, Support vector machines

1. INTRODUCTION
An artificial neural networks (ANNs) [1] are applied for a number of pattern recognitions and predictions such as image recognition, handwriting recognition, speech recognition, weather prediction, financial stock prediction and so on. The ANNs are statistical learning models, based on a model of biological neural structures connected with neurons.

The ANNs have a long history in computer science over 50 years. The first computational model for neural network, based on mathematics and neuroscience, was created by Warren McCulloch and Walter Pitts in 1943 [2]. Rosenblatt developed an algorithm applied for the perceptron in 1959 [3]. The first perceptron has only two layers in a neural network.

Nowadays, the ANNs present two kind of layer systems such as a single-layer perceptron and a multilayer perceptron (MLP) [4]. The single-layer perceptron is a simple neural network, which consists of a single layer of output nodes. The MLP is organized as a set of hidden layers of artificial neurons. These hidden layers have many neurons, which send and receive messages from multiple sources for computation of neural network. Recently, the ANNs improve the pattern recognition and classification, adding to artificial algorithms, developed by University of Toronto in 2009. We call new type of neural network a deep learning [5].

In this article, we discuss only about the MLP. We evaluate the instances by changing various hidden layers in the MLP. Also we compare the classification accuracies by the MLP and j48, Naive Bayes, AdaBoost and Support vector machines (SVM) [6, 7].

1. J4.8 is a decision tree algorithm and an implementation of the C4.5 algorithm in the weka data mining tool.
2. Naive Bayes classifiers are based on the Bayes’ Theorem.
3. AdaBoost is binary classification in a machine learning meta-algorithm
4. Support vector machines are a supervised algorithm and one of classification models.

2. MULTILAYER PERCEPTRON
The perceptron in the MLP is organized by many layers between inputs and outputs. Each layer involves a number of neurons. These neurons connect each other. We called these connections synapses.

2.1 Perceptron
Figure 1 describes the MLP, including 13 neurons, 3 hidden layers, 5 inputs and 5 outputs. From the left layer in this figure, we call these layers sensory, association and response.

2.2 Neuron
The neuron receives n messages \( a_1, a_2, \ldots, a_n \) with the weights \( w_1, w_2, \ldots, w_n \) on synapses in a unit of neuron (Figure 2). These messages are sent into one output \( S \) computing the sigmoid function.
Table 1: Comparison of the number of hidden layers and neurons in the MLP. The first column corresponds to the number of neurons in sensory, association and response.

<table>
<thead>
<tr>
<th>Number of neurons</th>
<th>Correctly Classified Instances (%)</th>
<th>Mean absolute error</th>
<th>Root mean squared error</th>
<th>Root relative squared error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3, 3, 0)</td>
<td>75.3906</td>
<td>0.3041</td>
<td>0.1492</td>
<td>87.9540</td>
</tr>
<tr>
<td>(5, 5, 0)</td>
<td>75.1302</td>
<td>0.2983</td>
<td>0.4281</td>
<td>88.0694</td>
</tr>
<tr>
<td>(3, 3, 3)</td>
<td>75.7813</td>
<td>0.3092</td>
<td>0.4162</td>
<td>87.3101</td>
</tr>
<tr>
<td>(3, 5, 5)</td>
<td>76.0417</td>
<td>0.3129</td>
<td>0.4192</td>
<td>87.9519</td>
</tr>
<tr>
<td>(5, 5, 5)</td>
<td>73.8281</td>
<td>0.3125</td>
<td>0.4302</td>
<td>90.2485</td>
</tr>
<tr>
<td>(5, 5, 10)</td>
<td>75.0000</td>
<td>0.3051</td>
<td>0.4220</td>
<td>88.5258</td>
</tr>
<tr>
<td>(10, 10, 10)</td>
<td>74.4792</td>
<td>0.3036</td>
<td>0.4355</td>
<td>91.3773</td>
</tr>
<tr>
<td>(20, 20, 20)</td>
<td>74.4792</td>
<td>0.2978</td>
<td>0.4274</td>
<td>89.6779</td>
</tr>
</tbody>
</table>

Figure 1: Description of the Multilayer perceptron. The MLP includes 13 neurons, 3 hidden layers and 5 inputs and 5 outputs.

Figure 2: Description of the Nuron. The neuron computes these multiple messages with a sigmoid function. The sigmoid function is based on biological neuron model. The function is described as below:

\[ \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \]  

where \( x \) is the input in the neural network. Figure 3 shows the sigmoid curve as below:

Figure 3: Sigmoid function.

3. EXPERIMENTS AND DISCUSSION

3.1 Environment and Equipment

In this section, we use a data mining tool, which is called Weka 3.6.12 [8, 9], including classification, clustering and prediction and association rule generation. We run Weka for evaluating the data sets on a machine with a 2.66GHz Intel Core 2 Duo processor and 8G-byte memory running under MacOS 10.10.4. For the experiments, we use a Pima Indians Diabetes data set, provided by National Institute of Diabetes and Digestive and Kidney Diseases [10]. The data set is organized by numerical values. The number of instances, attributes and classes are 768, 8 and 2, respectively.

3.2 Experiments

We evaluate the data set by changing the number of hidden layers and neurons. In addition, we compare the MLP with other classifiers such as J4.8, Naive Bayes, AdaBoost and SVM. The experiments are used in weather data set provided by Weka. In the experiment, we use C4.5, AdaBoost.M1 and Sequential Minimal Optimization (SMO) instead of J4.8, AdaBoost and SVM, respectively. The Weka implements a variant of C4.5 called J4.8 [11]. AdaBoost.M1 gives higher accuracy [12]. In Weka, SMO indicates with SVM. Also we describe statistical terms below:

- **Correctly Classified Instances** is the actual correct classification.
Table 2: Comparison of MLP and other classifiers.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Correctly Classified Instances (%)</th>
<th>Mean absolute error</th>
<th>Root mean squared error</th>
<th>Root relative squared error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP (3, 3, 3)</td>
<td>76.1719</td>
<td>0.4464</td>
<td>0.3078</td>
<td>87.5931</td>
</tr>
<tr>
<td>J4.8</td>
<td>73.8281</td>
<td>0.3158</td>
<td>0.4463</td>
<td>93.6293</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>76.3021</td>
<td>0.2841</td>
<td>0.4168</td>
<td>87.4349</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>74.3490</td>
<td>0.3127</td>
<td>0.4178</td>
<td>87.6631</td>
</tr>
<tr>
<td>SVM</td>
<td>77.3438</td>
<td>0.2266</td>
<td>0.4760</td>
<td>99.8620</td>
</tr>
</tbody>
</table>

- **Mean absolute error (MAE)** measures the average magnitude of the errors in a set of forecasts. The MAE formula is described below:

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n} \sum_{i=1}^{n} |e_i|,
\]

where \(e_i = |f_i - y_i|\).

- **Root mean squared error (RMSE)** is the standard deviation of the model prediction error. The RMSE formula is described below:

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f_i - y_i)^2},
\]

where the predicted values on the test instances are \(f_1, \ldots, f_n\); the actual values are \(y_1, y_2, \ldots, y_n\).

3.3 Results and Discussion

In the MLP, Table 1 shows the classified accuracy instances. The classified instances are not proportion to the number of layers and neurons. We have the best classified accuracy instance when the number of neurons in sensory, association and response are 3, 3 and 3, respectively. The result is 76.1716%. Even though the number of layers are the same in the perceptron, the results are different.

Table 2 shows that comparison of the MLP and other classifiers. We choose the best MLP accuracy instance in Table 1, comparing with other classifiers. From the results, we know that the MLP is not the best result from other classifiers’ experiments.

4. CONCLUSIONS

In future, we would like to evaluate more data sets in the multilayer perceptron and deep learning algorithm.

5. REFERENCES