Applying WEKA towards Machine Learning With Genetic Algorithm and Back-propagation Neural Networks

Zeeshan Ahmed1,* and Saman Zeeshan2

1Department of Neurobiology and Genetics, Biocenter, University of Wuerzburg, Germany
2Department of Bioinformatics, Biocenter, University of Wuerzburg, Germany

Abstract

Machine learning aims of facilitating complex system data analysis, optimization, classification and prediction with the use of different mathematical and statistical algorithms. In this research, we are interested in establishing the process of estimating best optimal input parameters to train networks. Using WEKA, this paper implements a classifier with Back-propagation Neural Networks and Genetic Algorithm towards efficient data classification and optimization. The implemented classifier is capable of reading and analyzing a number of populations in giving datasets, and based on the identified population it estimates kinds of species in a population, hidden layers, momentum, accuracy, correct and incorrect instances.

Keywords: Back Propagation Neural Network; Genetic Algorithm; Machine learning; WEKA

Introduction

Machine learning [1] is a branch of Artificial Intelligence, facilitating probabilistic system development for complex data analysis, optimization, classification and prediction. Different learning methods have been introduced e.g. supervised learning, unsupervised learning, semi supervised learning, reinforcement learning, transduction learning and learning to learn etc.

Several statistical algorithms (e.g. Genetic Algorithm [2], Bayesian statistics [3], Case-based reasoning [4], Decision trees [5], Inductive logic programming [6], Gaussian process regression [7], Group method of data handling [8], k-NN [9], SVMs [10], Ripper [11], C4.5 [12] and Rule-based classifier [13] etc.) have been proposed for the learning behavior implementation. The criterion for choosing a mathematical algorithm is based on the ability to deal with the weighting of networks, chromosome encoding and terminals.

Different machine learning approaches have been proposed towards the implementation of adaptive machine learning systems and data classification e.g. Fast Perceptron Decision Tree Learning [14], Massive Online Analysis (MOA) [15], 3D Face Recognition Using Multi view Key point Matching [16], Evolving Data Streams [17], Classifier Chain [18], Multi-label Classification [19], Multiple-Instance Learning [20], Adaptive Regression [21], nearest neighbor search [22], Bayesian network classification [23,24], Naive Bayes text classification [25], ML for Information Retrieval [26], Probabilistic unification grammars [27], Instance Weighting [28], KEA [29] and Meta Data for ML [30] etc. Apart from the fact of existence of these referred valuable approaches, we have decided to implement our own software application during this research and development, consisting of different methodology.

In this research, we are interested in finding the most suitable algorithm to establish the process of estimating best optimal input parameters and on the basis the selected parameters, train network to best fit with the use of suitable learning techniques. We discuss a script implementing the Genetic Algorithm for data optimization and back propagation neural network algorithm for the learning behavior. The objective is to analysis different datasets based on the number of attributes, classes, instances and relationships.

Keywords: Back Propagation Neural Network; Genetic Algorithm; Machine learning; WEKA

Following the agenda (Section 1), this short paper is organized in the upcoming sections: data classifier and its methodology explain in section 2, validation is performed in section 3 and observed results are concluded in section 4.

Optimal Data Classifier

The implemented classifier is proficient in reading and analyzing a number of populations in giving datasets. The classifier is developed using Java programming language and WEKA library [31,32]. The WEKA library provides built-in classes and functions to implement and utilize different mathematical and statistical algorithms (e.g. genetic algorithm and back propagation algorithm etc.) for data classification.

Based on the number of identified population, it estimates following results: kinds of species in a population (if there are more than 1), correctly classified instances, incorrectly classified instances, hidden layers, momentum and accuracy (optimized, weighted results).

The classifier is capable of processing standard Attribute Relation File Format (ARFF) dataset files, which describes the list of instances sharing a set of attributes, especially used to develop for machine learning projects. The classifier’s workflow starts with the analysis of inputted data and extraction of attributes, classes, instances and relationships. In the next step classifier extracts the information about number of hidden layers, learning rate and momentum to identify correctly and incorrectly classified instances. At the final step, classify the data using Back Propagated Neural Network for Multilayer Perception and optimize results using Genetic Algorithm.

During data classification using the genetic algorithm; first chromosomes are estimated, then learning rate and momentum is set
to perform cross over using a pair of the best results (Figure 1). The next
the mutation of two offspring is performed on the basis of obtained
accuracies of two previously estimated offspring. The offspring with
lower values are replaced with two new offspring. In the last steps, after
cross validation, the individual and commutative weights of instances
are calculated. The obtained results are validated and final output is
presented to the user in the end. The measurement and prediction
procedure can be repeated until the satisfactory results are achieved.

Validation
We have validated the classifier using two different data sets: Zoo
database (http://www.hakank.org/weka/zoo.arff) and Labor database

<table>
<thead>
<tr>
<th>Zoo Database</th>
<th>Labor Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1617 Mammals, 539 Birds, 0 Reptile, 637 Fish, 0 Amphibian, Insects and 49 Invertebrate from the whole population of 3332 species in dataset.</td>
<td>49 Good and 441 Bad of all 490 Population.</td>
</tr>
<tr>
<td>68 instances are correctly classified and rest 33 are incorrectly classified from all 101 instances</td>
<td>10 instances are correctly classified and rest 47 are incorrectly classified from all 57 instances</td>
</tr>
<tr>
<td>No Hidden layer</td>
<td>No Hidden layer</td>
</tr>
<tr>
<td>0.3 Learning rate</td>
<td>0.1 Learning rate</td>
</tr>
<tr>
<td>0.1 Momentum</td>
<td>0.5 Momentum</td>
</tr>
<tr>
<td>0.67326732673267326733 Accuracy</td>
<td>0.17543859649122806 Accuracy</td>
</tr>
</tbody>
</table>

Table 1: Results of Data classification.

(http://www.hakank.org/weka/labor.arff). Zoo database contains 101 Instances with of 18 Attributes; 2 numeric attributes (animal and legs) and 16 Boolean attributes (hair, feather, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic, cat size and type). Whereas the Labor database comprises of 57 instances including of 16 Attributes; 13 numeric attributes (duration, wage increase first year, wage increase second year, wage increase third year, cost of living adjustments, working hours, pension, standby pay, shift differential, statutory holidays, vacations, contribution dental plan and contribute to health plan) and 3 Boolean attributes (bereavement assistance, long term disability assistance and education alliance) (Table 1).

Both datasets are analyzed using implemented classifier, using WEKA explorer (Figure 2A and 2C). The observed results are (Figure 2B and 2D) are presented in Table 1. We have validated the classifier in three ways: (1) by increasing the learning rate and placing the momentum constant, (2) by increasing both learning rate and momentum and (3) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10–value) for learning rate and 3 bit decimal values for momentum.

Conclusions

We have observed during the validation process that by keeping the default weight of instance, the results become stable but by increasing the weight of instance the size of results increases. The findings lead to the outcome that mutation can affect the accuracy by increasing and decreasing it. Moreover, we have also observed that classifier produces results in minimum possible time with value 1, and if we will increase the value of classifier it will take more time.

Acknowledgement

We would like to thank Deutsche Forschungsgemeinschaft (DFG SFB 1047/Z) for funding and University of Wuerzburg Germany for the support. We thank anonymous reviewers for helpful comments on the manuscript and the Land of Bavaria Germany.

References


J Data Mining Genomics Proteomics
ISSN: 2153-0602 JDMGP, an open access journal

Volume 5 • Issue 2 • 1000157


