Rule extraction from MLP using clustering genetic algorithm

Xuanyu Lin

Research School of Computer Science
The Australian National University
E-mail: u6686380@anu.edu.au

Abstract

Multilayer preceptor (MLP) achieve high accuracy for classification, and it has been proven efficient for artificial intelligence systems. However, its disadvantage is that acquiring knowledge embedded in the MLP is challenging. In this paper, a method is shown to extract rule from MLP on classification problem. There are two main steps of the method. First, to find the clusters of hidden unit activation values, a clustering genetic algorithm is utilized. Second, these clusters are transformed in to rules in respect of the inputs. To evaluate the algorithm, a famous dataset, iris, which represents a real-world data is used.

1.Introduction

It has been proved that neural network is efficient for handling data mining problems in different area. Although high performance can be achieved by multilayer preceptor, acquiring knowledge from multilayer preceptor is still considered incomprehensible.[1]

In assignment 1, a rule extraction method using causal index and characteristic pattern has been applied to extract rules from a simple full-connected 3 layer hidden neural network with 7 input neurons, 3 hidden neurons and an output neuron.[2] In this paper, we extend the area from basic neural network to deep learning area, specifically the MLP network.

1.1 Iris Data

Since MLP has shown its high accuracy on classification, the goal of this paper is to extract rule from an MLP trained for classification problem. Therefore, dataset iris which is chosen because it is considered as the benchmark for data mining application.[3]

Iris Data has 600 rows. Each instance represents as a data of a flower. Iris Data has five attributes, sepal length, sepal width, petal length, petal width and species. There are 3 classes of species, each represent a specie of flower. The former four attributes are considered as input attributes to predict the last attribute, species.

In our paper, to simplify the dataset for further investigation, the original input attributes are not utilized. [3] 'Sepal area' and 'petal area' are used instead. The attribute 'Sepal area' is achieved by multiplying 'sepal width' and 'sepal length'. Similarly, the attribute 'petal area' is achieved by multiplying 'petal width' and 'petal length'.

1.2 Clustering genetic algorithm

To extract rules rom MLP, clustering genetic algorithm (CGA) is employed and examined. The goal of CGA is to cluster the activation unit value and generate rules from these clusters.

Generally, there are 7 main steps of CGA [3]:
1. Initialize a population of random genotypes.
2. Evaluate each genotype according to its silhouette
3. Applying a linear normalization
4. Select genotypes by proportional selection
5. Apply cross over and mutation
6. Replace the old genotypes by the ones formed by step 5
7. If convergence is attended, stop; else, go to step 2

2.Method

Fu et al. [4] describe a method to generate rules with evolutionary algorithm. The thought of CGA is similar to Fu. To obtained a solution to divide N data into K clusters is a NP-complete problem.[5] It is widely believed that genetic algorithm is efficient to handle NP-complete problem.[5] Hence, CGA is adopted and evaluated in this paper.

2.1 Encoding

In CGA, each genotype is in the form of an integer vector. If the dataset has N instance, the length of genotype is N+1, where the i^{th} position of the genotype represent the cluster which the i^{th} instance belongs to. The last number of the genotype represents the number of clusters this genotype has.

Consider a dataset with only 10 instances. The genotype is in the form of

G1: 22113333444

It means that the {1,2} instances belong to cluster 2. The {3,4} instances belong to cluster 1.
The {5,6,7,8} instances belong to cluster 3. The {8,9} instances belong to cluster 4. Hence, there are four clusters in this genotype.

2.2 Set up initial population

The first procedure of creating initial population is the selection of representative instances. The first instance is which located most centrally in the dataset. To obtain another instance, the steps below are executed

First, a non-selected instance i is considered to be the potential representative instance.

Second, a non-selected instance j is considered. The distance $\mathbf{D_j}$ of instance j with the most similar previously selected instance is calculated. Besides, the distance of instance j and instance i **d(j,i)** is calculated as well. Then, the difference between $\mathbf{D_j}$ and **d(j,i)** is achieved.

Third, if the difference is positive, it means that j has positive influence of selecting i. $\mathbf{C_{j,I} = max\{Dj - d(j,i)\}}$ is calculated.

Forth, the total gain $\sum_j Cji$ is achieved.

Fifth, instance i which maximizes the total gain is calculated.

This algorithm stopes when the total gain is negative since selecting instance i when the total gain is negative is not appropriate

After the representative instances are selected, the initial population is created Non-selected instances are clustered into different classes according to their distance with different representative instances.

Assuming that there are n representative instances are found, the number of population is n-1. The first genotype contains 2 clusters. The second genotype contains 3 clusters. Therefore, the r-1th genotype contains r clusters. Therefore, all classes of instances are contained in the initial population.

2.3 The Cross-Over Operator

The cross-over operator is described below.

1st, G1 and G2 are selected as parents.

1st

G1: 11223344555

G2: 22115544335

G1: 112<u>23</u>34<u>4</u>555

G2: 22115544335

2nd, randomly chose c = {k_1,k_2,k_3, …, k_n} genes from G1. These genes are directly copied into the same position of G2. The unchanged part of G2 is set to zero.

2nd

G1:112<u>23</u>34<u>4</u>555

G2:221<u>23</u>44<u>4</u>535

G1:11223344555
G2:00023004505

3rd, the zeros are soon replaced by the non-zero nearest gene. Then, the offspring G3 is produced.

3rd
G1:11223344555
G2:22223344555

Another offspring G4 is produced in the same way, but copying genes from G2 to G1

2.4 Mutation Operator

There are two mutation operators in CGA. The first is to eliminate genes randomly in a genotype. The eliminated parts are filled with nearest non-zero genes. The second operator is to divide a genotype into two new genotypes randomly. For example, a selected genotype is divided. The first new cluster is formed by instances closer to the original centroid. The second cluster is formed by those instances closer to the farthest instance from the centroid.

2.5 Objective Function

Silhouette [6] is the basic concept of objective function. For example, instance i belongs to cluster A. The average distance between i and all other instances in cluster A is represented as a(i). Similarly, the average distance between i and all other instances in cluster B is represented as b(i). Then, we can have.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

The higher s(i), the higher chance that instance i is assigned to cluster B.

2.6 Selection and setting

In each generation, genotypes are selected according to roulette wheel strategy. [7] The key feature of roulette wheel strategy is that negative function values can not be admitted. The highest fitness genotype is always copied into the succeeding generation.

3. Result and discussion

The architecture of MLP contains only one hidden layer with full connection between

each layer. The number of input neurons is the same as the number of attributes. The number of hidden neurons is 5. The number of output units is 3, which is equal to the number of species. Backward propagation algorithm is adapted to train MLP. The learning rate is set to 0.05. Initial weights are in the range of [-0.3, +0.3] randomly.

3.1Comprison

Edurado et al. [4] applies CGA to all instances and create an initial population contains 72 genotypes. The right solution is obtained below.

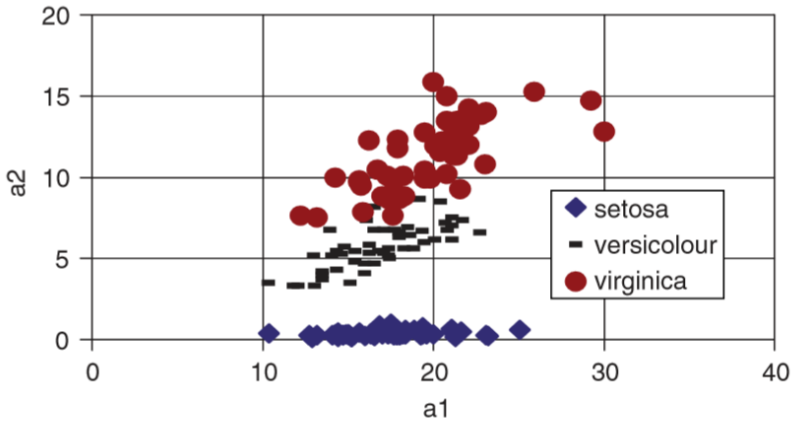

Fig. 3. Right clustering provided by the known classes.

The 98% instances are classified accurately after 15000 epochs.[4]
The hidden unit activation value for each training instance are calculated according to [4]:

$$a_1 = 0.22A_s - 2.16A_p + 0.84,$$

$$a_2 = 3.12A_s - 11.91A_p + 41.99,$$

The A_s is the sepal area. The A_p is the petal area.
Fig4 shows the clusters of activation values

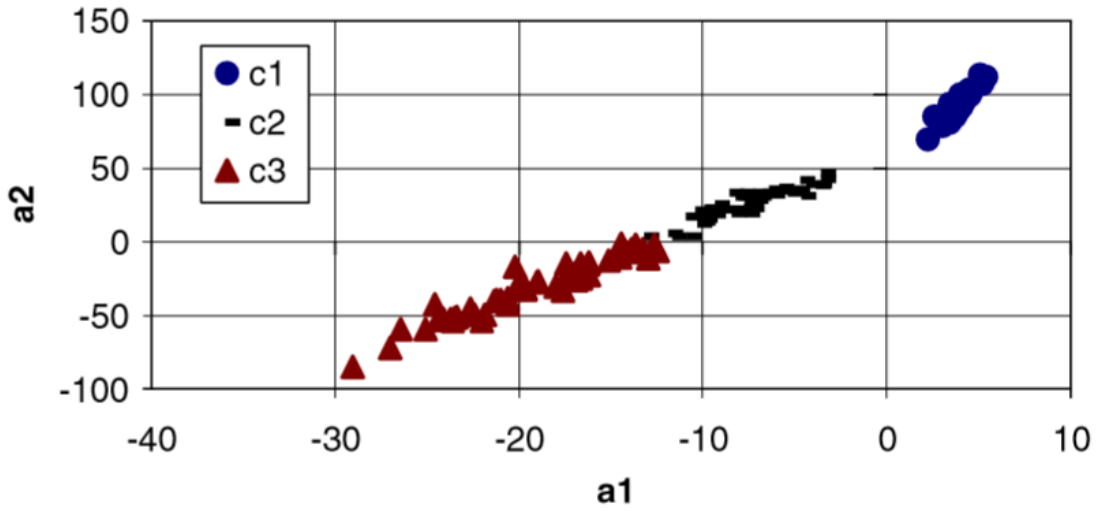

Fig. 4. Clusters of activation values.

The corresponding clusters are transformed into rules.[4]

If ($2.27 \leqslant a_1 \leqslant 5.43$ and $69.64 \leqslant a_2 \leqslant 113.09$) then C1—Setosa.
If ($-13.09 \leqslant a_1 \leqslant -3.46$ and $3.47 \leqslant a_2 \leqslant 46.54$) then C2—Versicolour.
If ($-29.04 \leqslant a_1 \leqslant -12.46$ and $-84.56 \leqslant a_2 \leqslant -1.14$) then C3—Virginica.

96% of the instances are classified correctly by rules and the knowledge of the MLP is achieved efficiently.[4]

Conclusion:

Multilayer preceptor (MLP) achieve high accuracy for classification, and it has been proven efficient for artificial intelligence systems. However, its disadvantage is that acquiring knowledge embedded in the MLP is challenging. In this paper, CGA is shown to extract rule from MLP on classification problem.

With genetic algorithm, CGA shows efficiency of setting up initial population. The number of genotypes is determined by total gain of the dataset. Therefore, the complexity of rule set is reasonable. With all the reason above we can consider that CGA is a efficient algorithm to cluster data set. The accuracy of the algorithm is high. CGA shows high performance of extracting rules from MLP. Excellent approximations of functions in MLP are produced by rules

Since the rule extraction above is under classification problem. The future work focus on rule extraction from regression network with CGA.

Reference

[1]W. Duch, R. Adamczak, K. Grabczwski **A new methodology of extraction** optimization and application of crisp and fuzzy logical rules, IEEE Trans. Neural Networks, 11 (2) (2000), pp. 1-31

[4]R.H Eduardo, N. F. F Ebecken Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach

[5]Y. Park, M. Song, A genetic algorithm for clustering problems, in: Proceedings of the Genetic Programming Conference, University of Wisconsin, July, 1998.

[6] L. Kaufman, P. J. Rousseeuw, Finding Groups in Data—An Introduction to Cluster Analysis, Wiley Series in Probability and Mathematical Statistics, 1990.