

Using Genetic Algorithm Trained Perceptrons with Adaptive Structure for the Detection of Premature Ventricular Contraction

J Zhou, L Li

Northern Illinois University, DeKalb, IL, USA

Abstract

Neural networks, typically multilayer perceptrons, have been widely used for the detection of heart arrhythmia based on electrocardiogram recordings. However, most state-of-the-art neural networks have a static preset structure, which tends to be over-massive and contributes to the problem of overfitting. In this study, we use genetic algorithm to adaptively decide the optimal model structure of a multilayer perceptron. The number of hidden layers and the number of nodes in each layer are dynamically obtained together with the synaptic connecting weights. Searching and cleanup stages are used in training to decide the optimal number of neurons in each layer based on crossover and mutation operators as well as cloning. Cooperative populations are chosen during evaluation. It works layer-by-layer until no more layer is required. The algorithm is applied to the MIT-BIH arrhythmia database to distinguish premature ventricular beats from normal beats. The experiment is conducted on all available records with at least 50 premature ventricular beats: records 106, 119, 200, 201, 203, 208, 210, 213, 215, 217, 219. The training starts from 500 initial neurons. As a result of adaptive structure searching, 10 out of 11 records needs only one layer except record 106 which needs 2 layers. The number of nodes on each layer is at most 3. The average recognition rate of all records is 96.96%. Results show that top performances can be obtained by multilayer perceptrons with simpler model structure than commonly reported in literature.

1. Introduction

Neural networks, typically multilayer perceptrons, have been widely used to detect heart arrhythmia based on electrocardiogram recordings [6][8]. However, most perceptrons have static preset model structure. The number of hidden layers and the number of neurons of each layer are decided in an ad hoc way. Such structures are not optimal and can be unnecessarily massive. Researchers have pointed out that an overly complex structure contributes to the problem of overfitting and

decreases the generalization capability of the model [2][10]. To overcome this problem, regularized neural network was suggested that is equivalent to a multiplayer perceptron with one hidden layer [2]. In this paper, we use an alternative approach to overcome the structure selection problem, which is to use genetic algorithm to adaptively obtain the optimal model structure of the multilayer perceptron. We then apply the method to the ECG arrhythmia analysis.

Section 2 details the method. Section 3 lists the experimental results. Section 4 is discussions and the conclusion.

2. Methods

Genetic algorithm has been used as an optimization algorithm to find out global solutions of nonlinear problems in various fields [1][3]. It emulates the natural evolution to find out organisms (solutions) that survive and prosper in the environment. In [1], a constructive algorithm is used to train the multiplayer perceptron. It is capable of adaptively obtain the model structure during training. The number of hidden layers and the number of nodes in each layer are dynamically obtained together with the synaptic connecting weights. Searching and cleanup stages are used in training to decide the optimal number of neurons in each layer based on crossover and mutation operators as well as cloning. Cooperative populations are chosen during evaluation. It works layer-by-layer until no more layer is required. The method is adopted in this paper as the classifier to detect premature ventricular contraction beats.

In the method, every weight vector for any single neuron is encoded as a 16-bit string, representing a real number of range [-1.0, 1.0]. This is the chromosome of each individual. Typically, the initial values of the chromosomes are set as binary numbers randomly.

A genetic algorithm consists of these important basic steps based on the combination of which a training process can be developed:

1. Evaluation

This stage uses training data to evaluate the

effectiveness of the weight vector, the result of which will be used for selection. The definition of the evaluation function is up to the algorithm designer. Here we adopt the cooperative population in [9]. The evaluation of the fitness for a given weight vector is defined as

$$F = \sum_{i=1}^N \delta_i, \quad (1)$$

where N is the size of training set.

$$\delta_i = \begin{cases} 0 & \text{if } x_i \text{ is classified incorrectly.} \\ f(n_i) & \text{otherwise.} \end{cases} \quad (2)$$

n_i is the total number of weight vectors which classify training observation x_i correctly. The $f(x)$ is a monotonically decreasing function defined as $1/x^2$ in our experiment.

2. Selection

In order to select the breeding population, we need to define a probability of selecting a particular weight vector based on the ranking of its fitness value r_i . The higher the fitness, the higher the rank. The *rank probability* for selected WV_i is calculated as following:

$$P(WV_i) = \frac{r_i}{\sum_{j=1}^K j} = \frac{2r_i}{K(K+1)} \quad (3)$$

where K is the total number of weight vectors for the current population.

3. Reproduction

Crossover, mutation and cloning operators are commonly used in the genetic algorithm for reproduction. These operators controls the changes when one generation reproduces its decedents. A pair of parent weight vectors generate new pairs with crossed values using crossover operator; mutation operator on the other hand, does not need a pair of weight vectors, and the mutation happens within the bit pattern of one individual weight vector; cloning copies one individual to its next generation. In our experiment, single point crossover and double point crossover are experimented. The former chooses one random point to start crossover and the latter uses two points.

The above three steps can be combined and repeated during the search for the surviving individuals (weight vectors), and the clean-up of the population. In a clean-up process, only cloning is used in the reproduction phase. As the result of clean-up, the number of different types of individuals (weight vectors) will be the number

of neurons for that layer. For every hidden layer, the algorithm is repeated layer by layer until only one neuron is sufficient, in which case the training stops.

Figure 1 illustrates the process of adaptive learning using genetic algorithm.

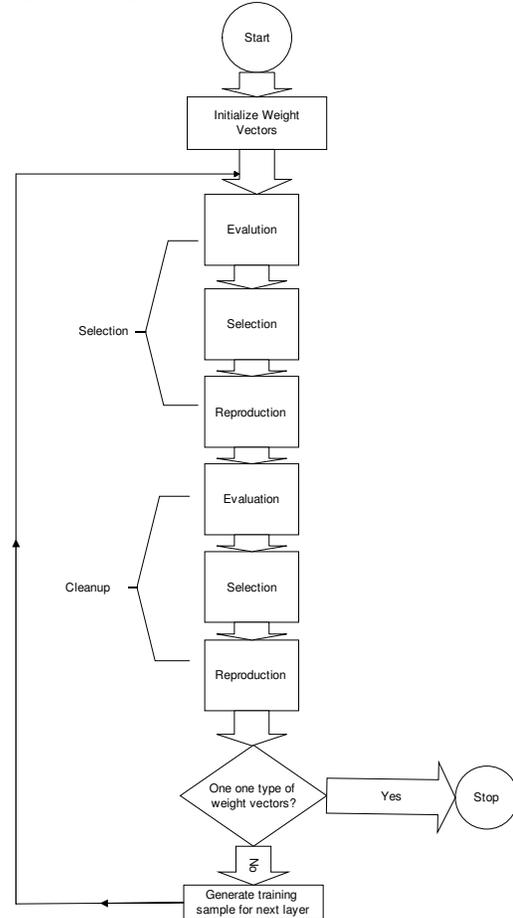


Figure 1 The learning process.

3. Experiments and results

The algorithm is applied to the MIT-BIH arrhythmia database [4]. The trained neural network is used to distinguish premature ventricular beats from normal beats. The experiment is conducted on all available records with at least 50 premature ventricular beats: records 106, 119, 200, 201, 203, 208, 210, 213, 215, 217, 219. The inputs to the neural network are the first three primary components based on the shapes of the QRS complex of the beats [7].

The algorithm is implemented in Java language. Table 1 lists the parameters that we used to get the results. Table 2 lists the results of the algorithm on all records. From our experiments on the MIT-BIH data sets, the

algorithm performs very steadily despite of the differences in parameters. Two types of crossover operators also generated comparable results. The reported results are based on single point crossover. The convergence time ranges from a few minutes to 40 minutes on a SunBlade 2000 workstation.

Table 1 Parameters of the experiments.

Type	Value
Number of initial neurons	500
Number of generations for search	300
Number of generations for cleanup	25
Probability for pair crossover	0.5
Probability for individual mutation	0.2
Probability of bit mutation	0.1

Table 2 Testing results.

Record	Performance on Training Set (Recog. Rate in %)	Performance on Testing Set (Recog. Rate in %)	Model Structure (layer of neurons other than the input layer)
106	88.75	92.08	2
119	100.00	100.00	1
200	95.42	96.02	1
201	100.00	100.00	1
203	92.34	97.30	1
208	99.33	99.59	1
210	98.00	100.00	1
213	100.00	100.00	1
215	82.73	80.00	1
217	99.60	98.65	1
219	99.00	100.00	1
Average	95.92	96.96	-

4. Discussion and conclusions

As a result of adaptive structure searching, 10 out of 11 records needs only one layer (no hidden layer) except record 106 which needs 2 layers (one hidden layer). The number of nodes on each layer is at most 3. The average recognition rate of all records is 96.96%. Results show that top performances can be obtained by multilayer perceptrons with simpler model structure than commonly reported in literature. For example, our method achieves high recognition rate of 99.59% on record 208, which was also tested by [6] and [8] and reported recognition rates for PVC of 99.59% and 96.48% respectively. However, in [6], the architecture uses a layer of two to ten

discriminators each of which in turn has several RAM-nodes. The authors of [8] use a hierarchical topology consisting of fuzzy neural networks, neural networks and fuzzy systems. Comparatively, the adaptive construction using genetic algorithm results in a simple linear structure for record 208. Some further observations of the approach and the experiment are as following:

1. The genetic algorithm trained MLP turns out to be a linear structure most of time, i.e. it contains only an input layer with arbitrary number of input neurons (depending on the training data set) and an output layer with the output neuron containing only 3 weights in most cases. This extremely simple linear and flexible structure implies obvious advantage over the fixed structure of nonlinear MLP, for example, it facilitate the use of the neural network by providing efficiency and rule clarity, which most conventional neural networks lack.
2. The simple structure is the result of the complex learning process. Every neuron and its synaptic weights come from iterative process of evaluation, selection and reproduction. The approach suggests that the optimal structure suitable for the data set is constructed, and the recognizing performance should be reasonably well. Because the structure will not be unnecessarily complex, overfitting can be avoided. Our experiments on MIT-BIH database show the average testing performance is better than the training performance, indicates that the scheme has good generalization capability instead of only fitting the training data. Being a global optimization problem, genetic algorithm also avoids local minima. As a result of above advantages, we show that even though the resulting structure of MLP is simple, the performance is good.
3. The inputs to the neural network are the primary components based on the shapes of the QRS waves of the beats. The simplicity of the model indicates that after primary component analysis, the feature space becomes relatively well separable.
4. This approach gives the designer the greatest flexibility in constructing learning rules suitable for the particular characteristics of a particular data set. This breaks out of the restrictions of conventional neural network with fixed structures. The fact that the resulting structure is able to absent from hidden layers not by mandatory choice but by genetic evolution possibly opens a door to a research direction that naturally links the linear and nonlinear modelling.

5. The vital evaluation criterion is the fitness value in the training process. The designer of the network can incorporate the error rate into the process. We can also use *parental linkage* in the process, i.e. instead of only evaluate current generation, the parent generation are included for competition, so that the good weight vectors can be kept. More experiments are ongoing to evaluate the effects of these techniques.
6. The resource consumption during training is one consideration of the approach. Tremendous amount of work is done internally during training, accordingly reasonably large amount of resources are needed. But with the rapid pace of computing resources, this may not be a major concern. The researcher can scale the amount of resources needed by adjusting the parameters based on the size and complexity of the training data set. In theory, it's suspected that the larger values the resource-related parameters are given, the better potentials the training result will be. In reality, trial-and-errors are needed to find out the set of parameters for a particular data set. From our experiments on the MIT-BIH data sets, the algorithm performs very steadily despite of the differences in parameters. It showed that the resources needed on training the regular complexity data set are moderate. Meanwhile, our experiments also show that the converging process is reasonably quick in most cases on the MIT-BIH data sets. In addition, the resource concern is only related to the designer of the network. It puts no pressure on the user of the network. A simpler resulting network structure means less resource consumption during testing phase and minimum resources are needed for the user with the linear structure.
7. One limitation of current algorithm is that it only identifies binary problems since the last layer has only one neuron. But there are different ways to get around this limitation by extending the algorithm to incorporate multi-class situations. One way is to let the final output be consisted of multiple neurons. The other way is to use the binary learner as a base learner that can be combined to achieve a multi-class classification. For example, many research have been conducted to extend binary support vector machine to a multi-class classifier [5].

In conclusion, the paper applied multilayer perceptrons with adaptive structure to detect the premature ventricular beats. The genetic algorithm based training is capable of

automatically obtaining the optimal structure of the model. The experiments on MIT-BIH database show that simple structure can achieve competitive results.

Acknowledgements

The work is partially supported by Northern Illinois University Graduate School Summer Research and Artistry Grant.

References

- [1] Andersen H, Tsoi A. A constructive algorithm for the training of a multilayer perceptron based on the genetic algorithm. *Complex Systems*, 1993;7:249-268.
- [2] Gironi F, Jones M, Poggio T. Regularization theory and neural networks architecture. *Neural Computation* 1995;7:219-269.
- [3] Goldberg DE, Richardson J. Genetic algorithms with sharing for multi-modal function optimization. *Genetic Algorithm and their Applications*. In: Grefenstette, editor. *Proceedings of the Second International Conference on Genetic Algorithms*. Morgan Kaufman, 1988:41-49
- [4] Goldberger AL, Amaral L, Glass L, Hausdorff JM, Ivanov P, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation(Electronic)* 2000;101:215-220.
- [5] Hsu CW, Lin CJ. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks* 2002;13:415-425.
- [6] Manintveld WA. Automated ECG analysis using weightless neural networks. Master's thesis, Dept. of Applied Mathematics, Delft University of Technology, 1996.
- [7] Moody GB and Mark RG. QRS morphology representation and noise estimation using the Karhunen-Loeve transform. *Computers in Cardiology* 1989;16:269-272.
- [8] Ramirez-Rodriguez CA, Hernandez-silveira MA. Multi-thread implementation of a fuzzy neural network for automatic ECG arrhythmia detection. *Computers in Cardiology* 2001;28:297-300.
- [9] Smith RE, Forrest S, Perelson AS. Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation* 1993;1(2):127-149.
- [10] Vapnik V. *Statistical Learning Theory*. John Wiley, New York, 1998.

Address for correspondence

Jie Zhou and Liqun Li
 Department of Computer Science
 Northern Illinois University
 DeKalb IL USA 60115