

行政院國家科學委員會補助專題研究計畫成果報告

結合基因法則之類神經網路技術 在手寫辨識系統之應用

計畫類別：√ 個別型計畫 整合型計畫

計畫編號：NSC 89 - 2213 - E - 009 - 217 -

執行期間：89年 08月 01日至 90年 07月 31日

計畫主持人：陳永平 教授

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學電機與控制工程學系

中 華 民 國 90年 10月 30日

結合基因法則之類神經網路技術 在手寫辨識系統之應用

計畫編號：NSC 89-2213-E-009-217

執行期限：89年8月1日至90年7月31日

主持人：陳永平 教授 國立交通大學電機與控制工程學系

計畫參與人員：李克聰 國立交通大學電機與控制工程學系

趙慕霖 國立交通大學電機與控制工程學系

倪豐洲 國立交通大學電機與控制工程學系

中文摘要

本計劃在研究一種新的演算法則，它利用基因演算法來訓練類神經網路的初始值，且其染色體為浮點數形式，這樣一來可節省計算時間。它不僅可改善倒傳遞演算法容易掉入區域最小解的缺點而且能夠克服基因法則無法有效收斂至鄰近區域最小解的困難。更進一步的研究單一基因交配比全部基因交配更快收斂至最佳解。最後，由模擬結果證實了此種演算法有較佳的收斂特性並且收斂時間也明顯的縮短。

關鍵詞：基因演算法、類神經網路、倒傳遞演算法

Abstract

This study investigates a novel neural network training technique, which employs the genetic algorithm to finding the initial values of the neural network. It is represented by a chromosome containing parameters in floating-point, so that the convergence rate to the minima becomes faster. This hybrid algorithm can overcome not only the drawback of easily slumping into local minima of back-propagation but also the genetic algorithm's defect that can't efficiently converge to the minima of the neighborhood. Further, the thesis shows that a gene changing one by one is better than that changing totally at once. Finally, the results of computer simulations reveal that this algorithm has a better convergence property, the time of global searching is obviously decreased.

Keywords: Genetic Algorithm, Neural Network, Back-Propagation

1. Introduction

In the case of using a neural network trained with the back propagation algorithm [1] as a classifier, initial weight of the neural network may cause the problem of finding local minima in training with the gradient descent technique because gradient descent techniques proceed by locally searching the immediate neighborhood of a current solution. However, the genetic algorithm has excellent global sampling abilities because it ensures broad coverage over the entire search domain. This suggests that using the genetic algorithm to provide good "seeds" from which

the back propagation algorithm then continues to search will be effective [2]. Therefore, we combined a genetic algorithm with the back propagation algorithm to avoid the problem of finding local minima in training with the gradient descent technique [3].

The work presented in this study investigates a method of encoding network weights and biases onto a chromosome, and utilizing a genetic algorithm to determine the weights and biases of fixed architecture artificial neural networks. The goal of this work is to show that the hybrid algorithm of using new genetic algorithms to train neural network is capable. The two novel genetic algorithms adopt the real-coded population, which means the chromosome containing parameters in the floating-point expression. The two new methods, one is totally change a chromosome at once, the other is changing genes one by one, and we will compare these two methods.

The introduction is described in this section. Section 2 illustrates the two newly designed algorithms. Section 3 shows some simulation results and an application in detail. Finally, a conclusion is given in section 4.

2. A Novel Neural Network Training Technique

Some specialized concepts are employed in our hybrid algorithm, which include the expression of population, rank-based fitness, rank-based reproduction, real parametric crossover and mutation, age and lifetime[4], search-converge criterion[5], pocket algorithm.

2.1 Weight

Our hybrid algorithm uses GA as weight training to find out initial set of good weights of the NN and then use back-propagation to find tune the weight of the network. Depending on the clearly defined problem to be solved, the size of the population (Pop-Size) is set N chromosomes, then each chromosome has $[(i+1)*j+(j+1)*k]$ genes. Let the chromosomes of be, where $m \in N$:

$$P_m = \begin{bmatrix} w_{11}, w_{12}, \dots, w_{1i}, w_{21}, w_{22}, \dots, w_{2i}, \dots, w_{j1}, w_{j2}, \dots, w_{ji}, \\ w_{11}, w_{12}, \dots, w_{1j}, w_{21}, w_{22}, \dots, w_{1j}, \dots, w_{k1}, w_{k2}, \dots, w_{kj} \end{bmatrix}$$

2.2 Genetic Algorithm(new method I)

2.2.1 Parametric Crossover

During sexual reproduction, crossover occurs: in each parent, genes are exchanged between each pair of chromosomes to form a gamete, and then gametes from the two parents pair up to create a full set of the chromosomes. The crossover operator roughly mimics biological recombination between two single-chromosome organisms. The concept of the crossover operation was similar to slope in the mathematics. For example, assume the chromosomes \mathbf{p}_k and \mathbf{p}_m are respectively obtained from the ranking policy, where k and m are the rank- k and rank- m , $k < m$. Obviously, the \mathbf{p}_k is superior to the \mathbf{p}_m . Let the chromosomes of \mathbf{p}_k and \mathbf{p}_m be

$$\begin{aligned}\mathbf{p}_k &= [p_{k1} \ p_{k2} \ \cdots \ p_{k(n-1)} \ p_{kn}] \\ \mathbf{p}_m &= [p_{m1} \ p_{m2} \ \cdots \ p_{m(n-1)} \ p_{mn}]\end{aligned}$$

where p_{ki} and p_{mi} , $i=1,2,\dots,n$ are the parameters of the chromosomes.

Each couple of strings would produce their offspring according to the crossover probability and the low mutation probability. The crossover operation would happen, if a number for crossover, also random selected from a uniformly distributed $[0,1]$ domain, is less than the given crossover probability. They will generate one offspring

$$\mathbf{q} = [q_1 \ q_2 \ \cdots \ q_{n-1} \ q_n]$$

where

$$q_{ti} = p_{ki} + \hat{a}(p_{ki} - p_{mi})$$

with parameters $\hat{a} \in [0,2]$

If α is very large, the algorithm will take much time of the convergence. Otherwise it also take much time when α is very small. It is evident that q_{ti} is a good offspring, it trending towards the good direction.

Note that all the parameters of the chromosomes are float-point. The advantage of parametric crossover is that it takes the less time than the traditional binary genetic algorithm.

2.2.2 Parametric Mutation

Mutation should be used low mutation rate because it is a random search operates; otherwise, with high mutation rates, the algorithm will become little more than a random search. The mutation operation would happen, if a number for mutation, also random selected from a uniformly distributed $[0,1]$ domain, is less than the given mutation probability. They will generate one mutant offspring. For example:

$$q_{ti} = p_{ki} + \hat{a}(p_{ki} - p_{mi})$$

$$q_{ti} = p_{ki} + \tilde{a}(p_{ki} - p_{mi})$$

with parameters $\hat{b} \in [-2,2]$ and $\hat{g} \in [-8,8]$

2.3 Connection

The option uses the GA to evolve the interconnect structure of the NN, in which the architecture of the GA and NN are fixed. After deciding the initial values of the GA, it will use back-propagation to train the weight. The parameters of the every chromosome are using a back-propagation algorithm, which is to say if the population has N chromosomes, the algorithm will have N BP, showed in Fig 2.

2.4 Back-Propagation

In our back-propagation add the momentum term and search-converge criterion. They can increase the learning speed and avoid the system to be unstable. The search-converge criterion combined the advantages of the Least-Mean-Square Error learning method (LMS) and probability learning algorithm.

2.5 Genetic Algorithm(new method)

2.5.1 Parametric Crossover

Crossover occurs: in each parent, genes are exchanged between each pair of chromosomes to form a gamete, and then gametes from the two parents pair up to create a full set of diploid chromosomes. Crossover operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. The crossover operator roughly mimics biological recombination between two single-chromosome organisms. Assume that population \mathbf{p}_k and \mathbf{p}_m are of rank- k and rank- m , where $k < m$, obtained from the reproduction operator, respectively. For example, the strings \mathbf{p}_k and \mathbf{p}_m could be crossed over after the second locus in each to produce the two offspring \mathbf{q}_k and \mathbf{q}_m . Let the chromosomes of \mathbf{p}_k and \mathbf{p}_m be

$$\mathbf{p}_k = [p_{k1} \ p_{k2} \ \cdots \ p_{k(n-1)} \ p_{kn}]$$

$$\mathbf{p}_m = [p_{m1} \ p_{m2} \ \cdots \ p_{m(n-1)} \ p_{mn}]$$

where p_{ki} and p_{mi} , $i=1,2,\dots,n$ are the parameters of the chromosomes. They will generate two offspring

$$\mathbf{q}_k = [q_{k1} \ q_{k2} \ q_{m3} \ q_{m4} \ \cdots \ q_{m(n-1)} \ q_{mn}]$$

$$\mathbf{q}_m = [q_{m1} \ q_{m2} \ q_{k3} \ q_{k4} \ \cdots \ q_{k(n-1)} \ q_{kn}]$$

Note that all the parameters of the chromosomes are float-point. The advantage of parametric crossover is that it takes the less time than the traditional binary genetic algorithm.

2.5.2 Parametric Mutation

This operator randomly flips some of the digit

in a gene. The sum of the before and after digits are nine. For example, the gene $p_{k1}(0.1234)$ could be mutating after the second locus in each to produce the offspring gene $q_{k1}(0.1734)$.

3 Application

3.1 Problem Statement

The off-line recognition of totally unstrained handwritten numerals were used to verify the performance of the proposed algorithm. The algorithm using a simple multilayer cluster neural network trained with the back propagation algorithm and the genetic algorithm avoids the problem of finding local minima in training the multilayer cluster neural network with gradient descent technique, and improves the recognition rates. In order to maximize the performance of unstrained handwritten digital recognition, the following approaches could be considered. One is to design a feature extractor which does not miss important features while minimizing the number of meaningless pixels, and the other is to design a classifier which has good generalization power and minimum substitution error. Our algorithm focuses on the last part.

3.2 Implementation

This section is organized as follows. Part 1 describes the data extraction. In part 2, the feature extraction is given.

(1)Input: the system diagram is shown in Fig. 3. The numbers was written on a white sheet of paper and the CCD was used for the numeral image capture and digitization. And then the image scale normalization is necessary to avoid an unknown character image presenting size different from that of the reference image selected by the system for recognition use. An enclosed numeral image after being size normalized shows a 30*30 matrix format.

(2)Feature extraction (6): We reduce the number of features from the 900-black/white attributes as follows: (1) we use only rows and columns numbered 0,3,6,8,10,12,14,16,18,20,22,25,28. (2) Starting from the left and moving across the rectangle in each of the six selected rows, we count the number of changes from black to white and from white to black to obtain the first thirteen counts C_1, \dots, C_{13} . And (3) starting from the top and moving down the selected columns, we count the number of changes in each to obtain thirteen more counts C_{14}, \dots, C_{26} .

3.3 Simulation Result

The genetic algorithm has been used to create a population of initial weight vectors for the multilayer cluster neural network and then back propagation has been used to optimize each of these. Basic parameters of the hybrid algorithm are presented in table 1. The system is implemented in P-2 300 using C language. One thousand of handwritten numerals are

obtained from 10 individuals. These numerals are captured from the CCD. After character segmentation and normalization at preprocessing stage, each image has a size of 30*30 pixels. An experiment for the recognition 1000 numerals written in a completely random manner has been carried out. Our hybrid algorithm is applied to the tested samples and the overall system performance is over 99% correctly recognition patterns. The experimental results are listed in table 2.

4. Conclusion

The GA-NN can overcome not only the drawback of easily slumping into local minima of back-propagation but also the genetic algorithm's defect that can't efficiently converge to the minima of the neighborhood. This architecture was used in our hybrid algorithm. Simulation results indicate that the new hybrid algorithm (I) is more efficient than the other methods when finding the optimal solution. The simulation results also indicate that the change a gene is better than totally change chromosome at once during the crossover operation of the genetic algorithm. The application results indicate that GA is not suitable to use larger parameters. Summary, GA has been used effectively for optimizing the weights of small neural networks (less than 10 neurons). For larger networks, GA becomes inefficient primarily two reasons: (1) vastness of the search-space; (2) limited population size that can be used in the GA.

References

- [1] RUMELHART, D. E., G. E. HINTON, AND R. J. WILLIAMS, *Learning representations by error propagation*, In D. E. Rumelhart, J. L. Mc Clelland, et al., eds., *Parallel Distributed Processing*, VOL.1, chap.8. Cambridge, MA: MIT Press, 1986a.
- [2] R. K. Belew, J. McInerney, and N. N. Schraudolph, *Evolving Networks: Using the Genetic Algorithm with Connectionist Learning*, Artificial Life II, CG. Laughton, C. Taylor, J. D. Farmer, and S. Rasmussen, Eds. Addison-Wesley, pp.511-547, 1991.
- [3] Seong-Whan Lee, "Off-line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 18. No.6, pp.648-652, 1996.
- [4] L. C. Jain and R. K. Jain, "An Evolutionary Approach to Training Feed-Forward and Recurrent Neural Networks," *IEEE, Second International Conference on Knowledge-Based Intelligent Electronic System*, pp. 21-23 April, 1998, Adelaide, Australia.
- [5] Holland, J. H., *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
- [6] Carl G. Looney, *Pattern Recognition Using Neural Networks-Theory and Algorithms for Engineers and Scientists*, 1997

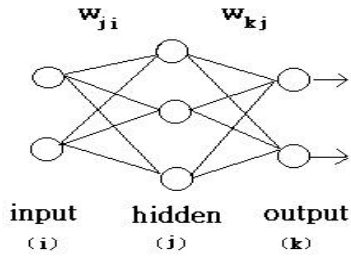


Fig1. Multilayer feedforward network with one hidden layer

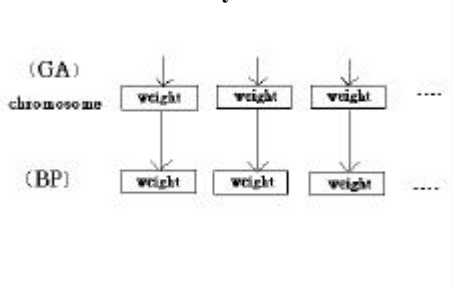


Fig2. The connection of the algorithm

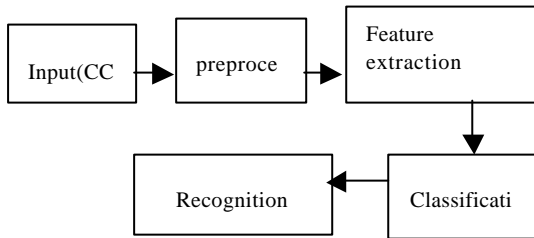


Fig3. The block diagram of our system

Numeral	Recognition rate	Error rate
0	99%	1%
1	100%	0%
2	98%	2%
3	98%	2%
4	100%	0%
5	98%	2%
6	99%	1%
7	98%	2%
8	100%	0%
9	100%	0%

Table 2. Recognition Results

	Other	BP	GA
New algorithm (I)	M_a = 5	T_t = 10000 T_p = 1000 I_n = 26 H_n = 30 O_n = 4 M = 0.03 L_r1 = 5 L_r2 = 10	P_s = 20 M_r1 = 2 M_r2 = 8 M_r1 = 0.9 M_r2 = 0.01

Table1. Basic parameters of the hybrid algorithm