A Research on the Application of Quantum Neural Network Optimization

Su Heng-yang
Experimental Training Management Center, Guangdong Industry Technical College,
Guangzhou 510300, China
email: shy_lovey@163.com

Abstract

The characters of fixed working hours table including the large amount of input and uncertainty of number of input parameters, make the encoding method has great influence on design method of traditional BP network combining genetic algorithm. This paper analyzes the features of searching ability of the traditional BP and genetic algorithm, based on which combining the quantum calculation and neutral model to compose quantum neurons. Then the quantum neurons are expanded to quantum neutral network to replace the traditional neutral network. Comparing to the drawbacks of the traditional genetic algorithm, the paper adopts a variation clamping mechanism. The mechanism gradually narrows the genetic operation space by fixing not sensitive single gene locus in the populations, so that the gene loci do not meet the requirements are more likely to participate in crossover and mutation to accelerate the speed up the genetic algorithm optimization and prevent it from falling into local extreme value. Finally, based on fixed table of mechanical standard working hours, compared to a variety of commonly used methods, the improved algorithm has better performance.

Keywords*: quantum neural network, improved genetic algorithm, fixed working hours table*

1. Introduction

Fixed working hours management is an important foundation for enterprise management, enterprise project management, and it is also an important basis for economic accounting, production schedule control, cost control and product pricing. The quality of fixed working hours formulating not only directly affect labor hours, equipment utilization, production cycles and labor remuneration of employees, but also promote the increase in labor productivity [1, 2]. Common used methods for calculating fixed working hours are table method, empirical estimates method, analogy method, the learning curve method, etc. In addition, artificial intelligence has also been introduced in order to obtain better results. Currently, a little work has been done on calculation of fixed working hours for neutral network in our country, such as Shujuan in Xi'an University of Technology [3], Zhong Hongcai in Shanghai Jiaotong University [4-7], Zhu Lixin in Northwestern Polytechnical University, Liu Shuhong in Jiangnan University, ZhuQiaoqiao in Southeast University and etc. The results they achieved prove neural network is feasible in calculation of fixed working hours. They use BP neural network which has shortcomings of easy convergence and easy to fall into local minima to train fixed working hours. Even if the neural network structure is reasonable, we cannot guarantee to quickly reach the optimal value. The biological evolution based genetic algorithm can obtain a larger probability global optimal solution, and has strong robustness, adaptability and high parallelism characteristics which is widely used in optimization problems. Genetic algorithms (GA) has parallel global search ability and BP neural network (BP) has local search ability, so the combination of these two algorithms can mutually compensate their deficiencies [8, 9].

In general, task time table has large quantity of data and the data input parameters are not fixed. These characteristics make BP design have different structures, so while applying GA to optimize BP weights and threshold value, the length change range of encoding is wide. And it is difficult to encode the initial group. Population initialization's operation efficiency is low. So how to choose the appropriate method for GABP optimization, quickly and accurately train the fixed working hours table and establish a model meets the requirements, are three problems need to be solved when training hours fixed standards. This paper uses a fixed working hours'

Received August 6, 2013; Revised November 1, 2013; Accepted November 22, 2013

table as the training sample and optimizes separately, and inducts and analyzes the optimizing effects in order to improve the training effect of genetic neural network training fixed working hours' quantities table [10-12].

2. BP Neural Network Model of Quantum

Quantum Neural Networks is a neural network model combining quantum computation based on the traditional BP artificial neural network [13-15]. The neural network model can be classified to quantum neural network model based on quantum gates. Quantum BP neural network model conveys based on changes in the angle of quantum states information and achieve quantum phase operation via shifting and rotating general quantum gate which is different from the traditional Neural networks. Quantum gates are the basis for quantum computing, and quantum BP neural network mainly uses universal quantum gates instead of the activation functions to operate on input vectors. Re-formulation of quantum bits are listed, for a quantum state $|\psi>$, we can get:

$$|\psi>=\alpha|0>+\beta|1> \tag{1}$$

α,β satisfy the normalization condition:

$$|\alpha|^2+|\beta|^2=1$$

The above equation of quantum states is expressed as the plural form:

$$f(\theta)=e^{i\theta}=\cos\theta+i\sin\theta \tag{2}$$

Comparing the two equations, we can obtain that |0> corresponds to the cosine, |1> corresponds to the sine which is also means the imaginary part and θ indicates the phase angle.

According to the above complex representation of quantum states, one bits phase gate and two bits controlled NOT gate can be expressed as follows:

One bit phase gate:

$$f(\theta_1+\theta_2)=f(\theta_1)f(\theta_2)=e^{i(\theta_1+\theta_2)} \tag{3}$$

Two bits controlled NOT gate:

$$f\left(\frac{\pi}{2}\gamma-\theta\right)=\begin{cases}\sin\theta+i\cos\theta & (\gamma=1)\\ \cos\theta-i\sin\theta & (\gamma=0)\\ else\end{cases} \tag{4}$$

From the formulas above, we can infer that the function of one bit phase shift gate is to phase shift the quantum states. As a control parameter of two bits controlled NOT gate, when $\gamma=1$, the quantum states rotate at the same time when $\gamma=0$, the phase changes but the probability amplitude observed doesn't change, so we define there is no change. When γ is other value, the quantum states can be freely changed. According to the description above, the quantum neuron model composed by one bit phase shift gate and two bits controlled NOT gate can be expressed as follows:

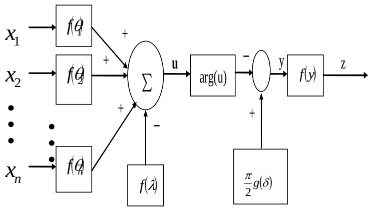

Figure 1. Quantum Neuron Model

$X = (x_1, x_2, \cdots, x_n)$ represents the angle input corresponding to the actual problem, $\theta = (\theta_1, \theta_2, \cdots, \theta_n)$ indicates the phase shift which also called weight, λ is called offset or threshold, δ mean the control factor for angle, arg(u) represents the phase of u $\arg(u) = actag(\mathrm{Im}(u) / \mathrm{Re}(u))$, and z is the output of quantum neuron. $g(x)$ is sigmoid function.

Suppose that I_i is the i-th input of quantum neuron, then the quantum neuron above can be expressed by the following formula:

$$\begin{aligned} u &= \sum_{i=1}^{n} f(\theta_i) f(I_i) - f(\lambda) \\ y &= \frac{\pi}{2} g(\delta) - \arg(u) \\ O &= f(y) \end{aligned} \tag{5}$$

The quantum neuron model is mainly adjusted by $\theta_i, \lambda, \delta$. The process of quantum neuron contains three steps. First, the input quantum state is phase shifted. Then the offset angle is added to correct the phase shift result. Finally, the correct result is processed by controlled NOT gate to get the quantum neuron output.

3. Improved Genetic Algorithm

The improved genetic algorithm find the approximate optimal solution through the good genes bit OS. This is equivalent to decompose large individual into a large number of small individual, collect good gene fragments appears, and then combine them into a new individuals with high fitness. When a certain period gene meet the optimization accuracy, in order to avoid the damage caused by crossover and mutation in next selection and narrow the scope of optimization, we fix it to achieve the goals. Finally, we apply it to a simple addition sum optimization function, the resulting effect is far better than that of simple genetic algorithm.

Assuming the size of a population is P, individual length is L, uniform crossover, mutation rate is M, and the maximum algebra is G. x_i^k represents the i th gene loci in the k th individual, $k \in [1, P]$ and $i \in [1, L]$. Evaluation function is f^k. In a simple genetic algorithm, with the increase in the number of genetic, optimization curve gradually level off, and ultimately achieve extreme value then stay unchanged. Mutation probability of single gene locus of the individual is M and the probability that the gene bit remains unchanged in n year is $(1-M)^n$. The two situations indicate that with the increase in the number of evolutionary gene, locus mutation probability is also increasing. For the entire population, each individual change along the direction of fitting evaluation function (ie individuals with great fitness are retained, individuals with little fitness are eliminated). We can also consider that the sensitivity individual to fitness function is decreasing, which is also the reason constantly flats the optimization curve called mutation relaxation.

This paper presents an improved method, called variant clamped which improve the performance of genetic optimization algorithm by fixing a larger or smaller loci. Each time fixing a gene can change genetic algorithm optimization space, in which likely includes the signal information cannot observed in the old space. Each space optimization can make certain of improvement to the average fitness of the population. As there are always new gene locus is fixed, fitness of the population is also rising. This mechanism is adjusted by the following parameters: mark parameter $bj \in (0, 0.5)$, unmark parameter $unbj \in (bj, 0.5)$ and mark time bjt. Then, the following equation can be gotten.

$$t_i = \sum_{k=1}^{p} x_i^k \Big/ p \quad , \quad i \in [1, L] \ , \quad k \in [1, P] \tag{6}$$

If in the n generation $t_i \le bj$ or $t_i \ge 1-bj$, all the i th segment gene in the individual are marked. In the following evolution if $unbj \le t_i \le 1-unbj$, all the individual exit the marking process, or else go on. If the marked gene segment is the bjt th generation, it will be fixed and no longer participate in the genetic algorithm optimization operation, which is called variation clamping.

The initial mark vector is $(z_1, z_2 ... z_n) = (0, 0 ... 0)$. If in the h-1th generation $unbj \le t_i \le 1-unbj$, and the hth generation $t_i \le bj$ or $t_i \ge 1-bj$, then we get z_i+1. In the next bjt generation, if always meet the condition $t_i \le bj$ or $t_i \ge 1-bj$, all the ith segment gene of the individual in the h+bjt th generation will be fixed. Otherwise, is no one in the next bjt generations meet $unbj \le t_i \le 1-unbj$, z_i=0 that is back to the origin

Assume that the $(t_1, t_2 ..., t_n)$ th loci in the mth generation meets the fixed requirement, then:

Population of the m-th generation is:

$$\left(x_1^k , x_2^k \ldots x_{t_1}^k \ldots x_{t_2}^k \ldots x_{t_n}^k \ldots x_n^k \right) \ , \quad (k = 1, 2, 3 \ldots P) \tag{7}$$

Populations of the m+u-th ($u \in (1, G-m)$) generation is:

$$\left(y_1^k , y_2^k \ldots x_{t_1}^k \ldots x_{t_2}^k \ldots x_{t_n}^k \ldots y_n^k \right) \ , \quad (k = 1, 2, 3 ... P) \tag{8}$$

In the case of no variation, the gene loci have been fixed will be strictly in the vicinity of greater or smaller value.

The operating steps of the improved genetic algorithm are as follows:

1) Representation of the problem: code the required problem and define the objective function (fitness function).

2) Initial population parameters, including overall size, crossover probability and mutation probability, and randomly generate an initial population.

3) Repeat the following operation to chromosome in the population until termination condition is met.

a. Calculating population fitness value for each gene locus.

b. If the fitness value of a gene locus satisfies the set threshold value (i.e. the above-identified optimum value), mark it and make it out of the genetic manipulation, thereby the gene locus is retained.

c. Retention optimal value, the resulting optimal gene segments are combined into a new individual, and replace the individual with the worst fitness value in generation.

d. To the gene loci meet the fixed requirements, all genes in that position of the individuals in the population are replaced by the fixed gene, and they are no longer in crossover and mutation operations.

f. According to size and value of fitness, the crossover probability and mutation probability to select, crossover and mutate chromosomes to produce a new generation.

A simple example is selected to illustrate that the algorithm is superior more relatively than the simple genetic algorithm. For a binary sum function:

$$y = x_1 + x_2 + ... + x_n \quad x_i \in \{0,1\}, n = 10000 \tag{9}$$

The selected parameters are as follows: the individual length is L = 10000, the population size is P = 500, the maximum evolution algebra MaxG = 600, chromosomes are encoded as a binary code, uniform crossover, and mutation probability is 0.003.

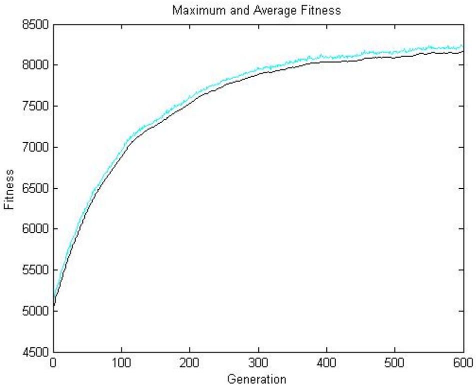

Figure 2. SGA Optimization Results

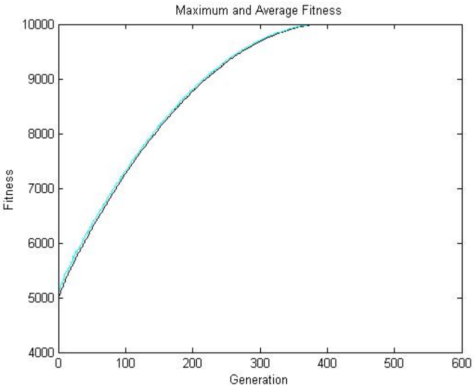

Figure 3. Optimization Results of Variation Clamping Genetic Algorithm

Figure 2 is a optimization curve of simple genetic algorithm. As can be seen from the Figure 2 that the convergence curve is gradually smooth with the increasing of the evolution algebra. So it is difficult to achieve the maximum. Figure 3 shows the improved genetic algorithm optimization curve which maintain a rapid convergence speed and can eventually converge to the maximum. Through the above simulation experiments we can obtain that the variation clamping genetic algorithm can effectively improve the rate of genetic algorithm optimization and prevent falling into local extremum. Later in this chapter, we apply clamp genetic variation neural network algorithm to the optimization of the initial weight threshold, also have good results.

4. Simulation Experiment and Analysis

Take three tables of production quota as the training sample, and test the above three coding methods' effect. Table 1 listed parts of the data in the three tables, in which T is standard man-hour quota (output), the rest parameters are input parameters. Each symbol's meaning is: D - outside diameter, L - length, S – across flats distance, B - width, T – depth

Table 1. Standard Task Time List

	S/mm	16	16	20	20	24	24	30
End mill square comprehensive time schedule	T/min	3.5	3.9	3.8	4.3	5.2	6.5	85
	D/mm	22.5	22.5	26.5	26.5	33.9	33.9	42.4
	L/mm	10	17	13	22	28	47	60
Cutter groove comprehensive time schedule	D/mm	63	63	63	63	80	80	80
	t/mm	3	6	10	7	4	6	8
	B/mm	6	6	10	10	8	12	16
	T/min	3	3.2	3.7	3.9	4	5.1	6.5
	L/mm	20	26	34	45	59	78	102

From task time table, it can be seen that the inputs are respectively 3 and 4, the output is 1. Determining the number of hidden layer nodes according to Kolmogorov theorem are 7 and 9, so establish BP model whose structure is 3-7-1 and 4-9-1. GA is adopted to optimize BP initial weights and threshold values. The optimization results are assigned to BP, and again carry on the training. Program flow chart is shown in Figure 1.

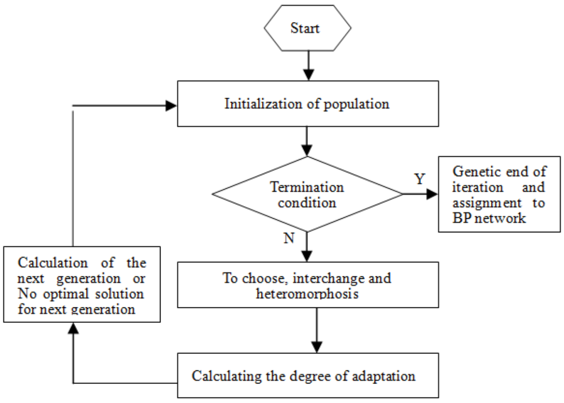
Start
Initialization of population
Termination condition
Y
Genetic end of iteration and assignment to BP network
N
To choose, interchange and heteromorphosis
Calculation of the next generation or No optimal solution for next generation
Calculating the degree of adaptation

Figure 4. Program Flow Chart

According to the structure of BP, determining the total weights and threshold value number, encode each parameter eight binary bits. The training's BP structure is 3-7-1 and 4-9-1, the numbers of parameters are 36 and 55, after encoding, there are 36 * 8 = 288 and 55 * 8 = 440 binary digits. Genetic algorithm (ga) adopts a standard genetic algorithm.

The chromosome genes numbers of variation clamping coding are 36 and 55. Based on real coded genetic algorithm, variation operation is the key. In this paper, the following scheme is adopted to implement variation: x is gene position, high and low are the upper limit and lower limit of x, rand is a small random number produced between 0 and 1. Due to the low possibility of the same random numbers produced by rand, x can be taken as variation value to use.

$$x = rand \times (high - low) + low \tag{10}$$

Because there has not direct conversion algorithm between real and gray code, in the gray code coding, first transform real number into 01 code, then transform 01 code into gray code and the reverse sequence is conducted in decoding. Other GA operations are the same with 01 code.

The above three coding genetic algorithms all adopt Formula (22) as fitness function, in which E is mean square error got in neural network pre-training of each chromosome.

$$fitness = 1/(E+1) \tag{11}$$

The above three coding modes are used to optimize BP parameters, in which the parameter selection of GA is as follows: population size 80, iterative upper limit 2000, variation probability is 0.01, crossover probability is 0.65. The time and the largest fitness value by taking the three methods after optimization are as shown in Table 2

Parameter values after optimization is assigned to BP training. Set the number of training 5000 times, Table 3lists the training accuracy, training error and maximum, minimum relative generalization error. It can be seen that training results' relative error of three coding method meets the requirement of 5%.

Table 2. Genetic Algorithm Training Correlation Table

		01coding	Gray code	Variation clamping coding
End mill square comprehensive time schedule	program time-consuming (s)	359.391	1.7256E+03	318.015
	Maximum fitness value	9.9854E-01	8.6617E-01	0.99966
Cutter groove comprehensive time schedule	Program time-consuming (s)	809.94	3.5624E+03	718.00
	Maximum fitness value	9.9865E-01	9.9492E-01	9.9875E-01

Table 3. Neural Network Training Contrast Table

		Training accuracy	Training steps	The maximum relative error	The minimum relative error
End mill square comprehensive time schedule	01coding	1.00E-04	5	4.73E-02	6.9633E-04
	Gray code	1.00E-04	5	3.63E-02	3.6905E-04
	Variation clamping coding	1.00E-04	5	3.61E-02	1.20E-03
Cutter groove comprehensive time schedule	01coding	1.00E-05	5000	4.95E-02	6.7488E-06
	Gray code	1.00E-04	87	4.96E-02	1.0548E-04
	Variation clamping coding	1.00E-05	58	4.97E-02	1.2711E-04

From the above table data, it can be seen that these three coding methods' final optimization results are close to each other. The Variation clamping method has the shortest encoding length and the least coding time. Each bit of variation clamping coding represents a parameter which does not need decode. The result is the optimization solution. While each parameter coding length of 01 code and gray code is determined according to the accuracy of parameters, and 01 coding adds a process from binary to real decoding, gray code should first decode to binary, and decode from binary to real.

In addition, it can be calculated from fitness value, after the optimization of genetic algorithm, neural network parameters got can make the training sample up to 10-4 orders of magnitudes. If put the parameters optimized into neural network, it may soon reach the specified convergence precision.

5. Conclusion

This paper discusses different coding methods' efficiency in training formula quota table of genetic neural network, and uses an example to illustrate the advantages and disadvantages of different coding methods: 01 code is easy to realize and crossover and mutation operation are simple; Variation clamping coding length is short with less time consuming, genetic operator operation is a little complicated; Gray code overcomes the shortcomings of Hamming in 01 coding, but consumes a lot of time. Although after genetic algorithm optimization, the neural network rapidly converges and increases greatly the probability of optimal solution, if neural network structure is not reasonable or parameter settings are improper, neural network fall into local minimum or cannot reach the specified generalization error. So besides coding problem, the structure design and parameter setting of the neural network are also important problems need to be solved in the future.

References

[1] Takeshi Hashimoto, Abdullah S Alaraimi, Chenggao Han ICI. Mitigation Schemes for Uncoded OFDM over Channels with Doppler Spreads and Frequency Offsets – Part II: Asymptotic Analysis. *Journal of Communications*. 2012; 7(9): 686-700.

[2] Xiao Hong, Cao Maojun, Li Panchi. Quantum-inspired Neural Network with Application to Image Restoration. *AISS*. 2013; 5(10): 1198-1207.

[3] Manju Mathew, A Benjamin Premkumar, Chiew-Tong Lau. Multi-user Interference and Cross Correlation Effects of Spline Multiwavelet based Cognitive Radio Network. *Journal of Communications*. 2012; 7(9): 701-711.

[4] Qi Ren, Jianhui Wu, Weijun Guan, Yun Li, Lihua Cui. Research on Happiness based on Analysis of Variance. *IJACT*. 2013; 5(5): 232-239.

[5] Gilbert Micallef, Louai Saker, Salah E Elayoubi. Hans-Otto Sceck Realistic Energy Saving Potential of Sleep Mode for Existing and Future Mobile Networks. *Journal of Communications*. 2012; 7(10): 740-748.

[6] Peng Jiang, Bo Xu, Huali Cai, Yuexiang Yang. The Aircraft Life Cycle Cost Estimation based on Generalized Virtual Economy. *JCIT*. 2013; 8(5): 128-137.

[7] Masahiro Sasabe, Hirotaka Nakano. Perfect Cell Partitioning Scheme for Micro-Cellular Networks. *Journal of Communications*. 2012; 7(10): 749-757.

[8] Han Xiaodong. Fault Positioning Technology for Power Grid Based on Wavelet Neural Network. *Bulletin of Science and Technology*. 2013; 29(6): 59-61.

[9] Cho See Jung. Working-time Reductions and Changes in Employment: The Case of Korea. *JCIT*. 2013; 8(12): 429-435.

[10] Se-Hee Jung, Yunsuk Cha. Wellness in the Workplace: Health Promotion Programs. *JCIT*. 2013; 8(13): 440-446.

[11] Zhifeng Zhang, Su Wang, Xingang Miao. Realization of Robot-assisted Laser Scanning Measurement System based on Assembly Platform. *JDCTA*. 2013; 7(1): 153-158.

[12] Liangming Li, Ai-hua Yang, Hui-Tang. The effects of GLUT4 and IL-6 on the development of insulin resistance inhibited by exercise. *JDCTA*. 2012; 6(21): 382-389.

[13] Hongjuan Zhang, Long Quan. The Kinetic Characteristics of the Clamping Unit in Injection Molding Machine Driven by New Pump Control System. *AISS*. 2012; 4(21): 468-475.

[14] Nagaraj Mudukpla Shadaksharappa. Optimum Generation Scheduling for Thermal Power Plants using Artificial Neural Network. *International Journal of Electrical and Computer Engineering*. 2011; 1(2): 134-139.

[15] YU Li-yang, WANG Neng, ZHANG Wei. Neural-network Based Aggregation Framework for Wireless Sensor Networks. *Computer Science*. 2008; 35(12): 43-37.