A NEW METHOD OF IMAGE COMPRESSION BASED ON QUANTUM NEURAL NETWORK

LI Huifang
School of Electronic Information
Northwestern Polytechnical University
Xian 710072, China
Lhuifang@nwpu.edu.cn

LI Mo
School of Electronic Information
Northwestern Polytechnical University
Xian 710129, China
LM365308056@163.com

***Abstract*—In this paper we combine with quantum neural networks and image compression using Quantum Gates as the basic unit of quantum computing neuron model, and establish a three layer Quantum Back Propagation Network model, then the model is used for realizing image compression and reconstruction. Since the initial weights of neural networks were slow convergence, we use Genetic Algorithm (GA) to optimize the neural network weights, and present a mechanism called clamping to improve the genetic algorithm. Finally, we combined the Genetic Algorithm with quantum neural networks to finish image compression. Through an experiment we can see the superiority of the improved algorithm.**

Keywords-Genetic Algorithm; Quantum Neural Networks; Mutational Clamping; Image Compression

I. INTRODUCTION

With the rapid development of the information society, multimedia data requires considerable storage capacity and transmission band width. Especially, the continuing growth of data intensive multimedia-based web applications not only have sustained the need for more efficient ways to encode signals and images but also have made compression of such image focused on storage and communication. Images compression has attracted a lot of research recently. Many methods have been proposed for image compression and most of them focus increasing compression rate while protecting the image information.

In this paper, a new scheme of image data compression based on Genetic Algorithm (GA) optimization and quantum neural network is discussed. Quantum Neural Networks (QNN), which combines the characteristics of Artificial Neural Networks (ANN) with quantum theory, is a new technical theory. It takes advantages of Neural Networks and quantum computing, and has high theoretic value and using potential on account for increasing the system processing ability and the learning self-adapt ability.

Firstly, the conceptions of quantum computing and neural computing are discussed in this paper. A QNN model which can input quantum and output quantum are created. Secondly, A scheme of combining QNN with a modified genetic algorithm is given to set the learning rates and neural network topologies. Since the training weights of neural networks were slow convergence, we present a method through fixed larger or smaller locus to improve the performance of simple genetic algorithm. When any set of genes gets fixed in the population, the representation of the problem space can be thought to have changed. And the new representation may contain one or more sets of genes which may not have had a detectable fitness signal in the old representation. Each time a small set of genes gets fixed, the average fitness of the population will increase by an amount that may be tiny. As the fixation of small sets of genes continues, however, these amounts will begin to add up. Thirdly, we applied GA to Quantum neural network optimization of the initial connection weights and image compression. Finally, the experiments results show that QNN has better learning velocity and image compression ability than Back Propagation (BP) neural network.

II. QUANTUM NEURAL NETWORK MODEL

There are many approaches to the development of QNN models, proposed by NORIAKI [1], RiGui[2] et al. These models focus on different aspects of quantum computations and neural processing. In quantum computing, As the smallest unit of information, a quantum bit or qubit is a quantum system whose states lie in a two dimensional Hilbert space. Like bits in classical computers, qubits labeled and express one bit of information: corresponds to the bit 0 of classical computers, and to the bit 1. Qubit state expresses a coherent superposition of states:

$$|\psi\rangle=\alpha|0\rangle+\beta|1\rangle \quad (|\alpha|^2+|\beta|^2=1) \tag{1}$$

Where α and β specify the probability of the corresponding states.

Quantum gate which includes the characteristics of quantum computing is the basis for physical implementation of quantum computing. The universal logic sets are included in the quantum logic. Similar with classic bit, the basic gate can form arbitrary quantum gates and complete quantum state of some of the logic of transformation. An element based on the 1 bit phase-shift gate and 2 bit controlled-Not gate in quantum dynamics are taken as a activation function in Neural Networks. To facilitate application, the following form of complex function is given to express the quantum state:

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

978-0-7695-4132-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ISME.2010.242

$i=\sqrt{-1}$ is imaginary unit, θ is quantum phase, the probability amplitude of $|0\rangle$ is the real part of complex function, the probability amplitude of $|1\rangle$ is the imaginary part of complex function. Then a quantum state can be described as:

$$|\psi\rangle=\cos\theta|0\rangle+\sin\theta|1\rangle \tag{3}$$

The 1 bit phase shift gate and 2 bit controlled not gate can be described as follow:

1) phase-shift gate :

$$f(\theta+\theta')=f(\theta)f(\theta') \tag{4}$$

It changes the phase of quantum state, so that quantum state phases rotation an angle.

2) Controlled-not gate:

$$f\left(\frac{\pi}{2}\gamma-\theta\right)=\begin{cases}\cos\theta+i\sin\theta\,(\gamma=1)\\ \cos\theta-i\sin\theta\,(\gamma=0)\\ else\end{cases} \tag{5}$$

γ is a control parameter, when $\gamma=1$ quantum state overturns; when $\gamma=0$, although the phase of the probability amplitude of $|1\rangle$ overturns, the probability of it was observed remains unchanged, so we do not think quantum state overturns. Different quantum states corresponding to different γ.

Quantum neuron model based on 1 bit phase-shift gate and 2 bit controlled-Not gate are showed as follow.

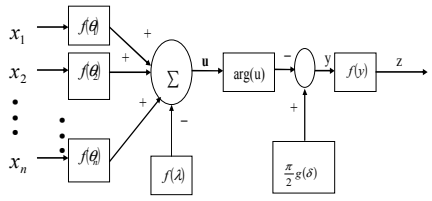

Figure 1. Quantum neuron model

In the Fig 1, $x_i\,(i=1,2,...n)$ is the i^{th} input to the quantum states of the neurons, $\theta_i\,(i=1,2,...n)$ is the phase transfer coefficient of weight, λ is the threshold coefficient, δ is the phase control factor, z is output state, arg(u) is the phase of u ($\arg(u)=actag\left(\mathrm{Im}(u)/\mathrm{Re}(u)\right)$), function f is defined as the formula (1), g(x) is sigmoid function.

Quantum Neural Network discussed in this paper is a combination of the qubit neuron and multi-layered feed-forward neural networks. All neurons in the input layer convert the input value [0, 1] into the quantum state phase [0,π/2]. If an input sample is given by a vector $(x_1,x_2,...,x_L)\,(x_i\in[0,1])$, the output of the i^{th} neuron in the input layer is $f(\pi x_i/2)$.

Back Propagation (BP) algorithm is one of algorithm widely used in the neural network (NN) training. But BP algorithm has some unavoidable defects such as slow speed in training (especially for large training samples), liable to get into local minimum, and so on. Although some improved BP algorithm is given, these algorithms can hardly avoid the BP algorithm getting into local minimum. Considering of the ability of Genetic Algorithm (GA)'s global searching[3,4,5], we use improved GA to train quantum neural network architecture to avoid it getting into local minimum and obtain effective result.

III. THE IMPROVED GENETIC ALGORITHM

Unlike the BP rule, a GA can avoid local minimum traps while performing a global search, but a drawback of the GA is slower when handling large computations. In the simple genetic algorithm (SGA), with the increase of the number of generations, optimizing rate gradually becomes flattening curve, and finally tends to extreme. For this phenomenon we assume that an individual locus's mutational probability is μ, after generations n, the probability of a gene keep constantly is $(1-\mu)^n$. This shows that with the increase of evolution, the mutation probability of a gene is increase. For all of the population each individual is change along with the fitness function. We can also think that the sensitivity of the fitness function continues to decrease. This reduction in sensitivity is the reason that optimization curve continued to flatten, it is called as mutational drag.

In according to above property, we present a new method to improve the mutational drag, known as improved GA (IGA), and fixed larger or smaller locus to improve the performance of simple genetic algorithm. When any set of genes is fixed in the population, the representation of the solution space can be thought to have changed. And the new representation may contain one or more sets of genes which may not have had a detectable fitness signal in the old representation. In driving such genes to fixation, the SGA raised the average fitness of the population by a small amount. As the fixation of small sets of genes continues, however, these amounts will begin to add up. Suppose in generation m we had fixed the locus $(t_1,t_2...,t_n)$.

Then the population of generation m is

$$\left(x_1^k,x_2^k...x_{t_1}^k...x_{t_2}^k...x_{t_n}^k...x_n^k\right),\ (k=1,2,3...P) \tag{6}$$

The population of generation m+u, $u\in(1,G-m)$ is

$$\left(y_1^k,y_2^k...x_{t_1}^k...x_{t_2}^k...x_{t_n}^k...y_n^k\right),\quad (k=1,2,3...P) \tag{7}$$

These depend on the following parameters. Flag is $b_j \in (0, 0.5)$, uFlag is $ub_j \in (b_j, 0.5)$ and Flag period is b_{jt} , if in generation n: $t_i \le b_j$ or $t_i \ge 1 - b_j$, then that locus is flagged, once flagged a locus remains flagged as long as the $t_i \le b_j$ or $t_i \ge 1 - b_j$, If a flagged locus in some generation t has remained constantly flagged for the last b_{jt} generations then the locus is considered to have passed our fixation test, and is not mutated in last generations. We call this mechanism mutational clamping. We expect that in the absence of mutation, a locus that has passed our fixation test will quickly go to strict fixation for the remainder of the run. The training procedure of neural network weight based on the improved GA is following:

(0) Generation $\leftarrow 0$.

(1) Initialize population: code scheme is selected. The code string is composed by five parts: weight value, threshold value, the state of node, learning rate and momentum factor.

(2) Evaluate fitness of the initial population.

(3) Generation $\leftarrow$ Generation+1.

(4) Tournament selection operation.

(6) Mutation operation.

(7) Evaluate fitness of new population: the weights are optimized based on the objective function defined as:

$$\text{Minimize : } \text{MS}E = \frac{1}{2P} \sum_{i_S=1}^{P} \left(y_{d_i} - \hat{y}_{d_i} \right)^2 \tag{8}$$

Where P is the number of training samples, MSE is the square sum of error between the target y_{d_i} and output $\hat{y}_{d_i}$

(8) If generation is maximum, then output best weights.

(9) Repeat (3)-(8).

We take a three-layer neural network as the model. The parameters are as follows: size of population is 300, gene length is 1612, max generations is 1000, uniform crossover, mutation probability is 0.003, b_j =0.3， ub_j =0.4， b_{jt} =100 。The results are as Table 1：

TABLE I. THE AVERAGE FITNESS AND MAXIMUN FITNESS OF SGA AND IGA

Generations	Factors		
	Algorithms	*Average Fitness*	*Maximum Fitness*
100	SGA	0.672	0.715
	IGA	0.671	0.716
500	SGA	0.761	0.780
	IGA	0.885	0.909
1000	SGA	0.782	0.801
	IGA	1.343	1.374

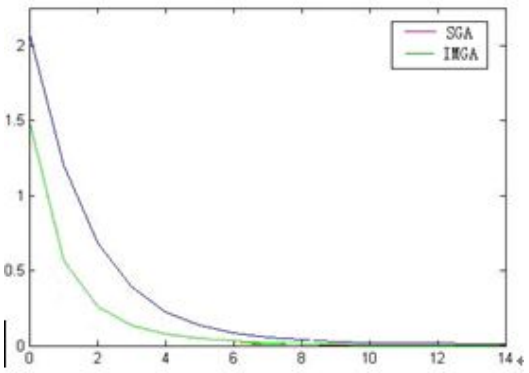

Figure 2. The Comparison of Convergence curve.

From Table 1, we can know that in driving genes to fixation the SGA continue to raise the average fitness of the population. If we continue to increase the maximum number of generations, optimization can get better results. From Fig 2, we also can see the IGA faster convergence than SGA

IV. IMAGE COMPRESSION

The original image ($X * Y$ pixels) is split into non-overlapping small images ($X_m * Y_m$ pixels), and each small image pixels values are normalized in [0， 1], the normalization of the source image pixel values as input and output of the quantum neural network. The input layer and output layer neurons are N= $X_m * Y_m$ 。The pixel value of input layer is converted from [0, 1] to [0, π / 2] used as the phase value of the quantum state. Image compression is completed by the quantum state from the input layer to hidden layer. We use IGA method based on mean square error to adjust the weight, until the image reconstruction error less than target error. Then the probability of hidden layer neuron output value $|\bullet\rangle$ is the results of image compression, and the probability of output layer neuron output value $|\bullet\rangle$ is the results of image compression reconstruction. Image reconstruction is the inverse process of image compression, the purpose is to restore the output image pixels $X * Y$ and multiplied by 255.

In this paper we use a 256×256 pixels image as the original image, using the discussed method, and construct neural network model for image compression. In order to achieve a similar compression ratio, we use network structure with 64-8-64. Target error is 0.0001. The maximum numbers of iterations are 1000. And we use IGA optimized connection weights as the result of compression.

Table 2 shows the corresponding reconstructed image compression performance. Comparison signal to noise ratio (SNR) and peak signal to noise ratio (PSNR) of reconstructed image, QNN is better than BP, ands the difference between the original image and the Reconstructed image almost is impossible to be watched (as figure 3).

TABLE II. THE PERFORMANCE OF IMAGE COMPRESSION AND RECONSTRUCTION

Algorithms	Factors		
	SNR(db)	*PSNR(db)*	*Epochs*
QNN	42.7528	45.1300	586
BP	42.7414	45.0867	1084

(a) Original image (b) QNN Reconstructed image

(c) BP Reconstructed image

Figure 3. Original image and reconstructed image of QNN and BP

From our research we can know that quantum neural network has good ability of image compression, and with the increase in compression ratio, not only reduces the size of the network, speed up the learning convergence speed of network, but also the reconstructed images do not exist in the visual distortion. This because QNN network in the process of image compression training, the information is stored not only in the network weights and threshold values, but also in the phase shift parameter, less loss during the compression of data. In other hand, QNN on the real and imaginary parts of the training at the same time, this not only greatly improves the computational efficiency, but also closer to the original value.

V. CONCLUSION

In this paper we construct a QNN model used in image compression. The quantum neuron structure of the quantum controlled NOT gate can play the role of micro-correction of the output of BP neurons, and there has phase shift parameters which change between 0-1 in the network, so the Quantum neurons output will continue to change, thus out of local minima. Moreover, we present a mechanism called mutational clamping that dramatically improved the performance of SGA. Meanwhile, we change space by fixed smaller or larger gene locus. Each time drive such genes to fixation the SGA raised the average fitness of the population by a small amount. With the increase of evolution, new locus is fixed. If we continue to increase the maximum number of generations, optimization can still get better results. In combination with quantum neural networks can be seen that the algorithm is better than simple genetic algorithm in largely data processing.

ACKNOWLEDGMENT

I wish to thank my teacher, Huifang LI for teaching me the essence of quantum mechanics, and for his guidance through the whole year. Without his assistance help me this paper will not take on its shape. And last but not least, I would like to thank all my friends, especially those give me constant encouragements.

REFERENCES

[1] NORIAKI KOUDA1, An Examination of Qubit Neural Network in Controlling an Inverted Pendulum Neural Processing Letters,2005, No.2, pp277–290.

[2] RiGui Zhou,HongYuan Zheng, SelfOrga- nizing Quantum Neural Network, 2006 International Joint Conference on Neural Networks 2006, July, pp16-21.

[3] G Harik,F G Lobo.The Compact Genetic Algorithm. IEEE Transon evolutionary computation 1999, 12, pp287-297.

[4] Francesco Cupertino,Ernesto Mininno. Elitist, Compact GenticAlgorithms for Induction Motor Self-tuning Control, IEEE Congress on Evolutionary Comptution, Vancouver, Canada,2006, pp3057-3064.

[5] Muhlenbein H. Parallel Genetic Algorithms in Combinatorial Optimization, Computer Science and Operation Research New Developments, NewYork: Pergamon Press, 1992. 441-453.

[6] Keki M.Burjorjee, On the Workings of Genetic Algorithms The Genoclique Fixing Hypothesis.arXiv:0905.2473v1 [cs.NE] 15 May 2009.