Gauss-Morlet-Sigmoid Chaotic Neural Networks

Yao-qun Xu and Ming Sun

Institute of System Engineering, Harbin University of Commerce, 150028, Harbin, China
Xuyq@hrbcu.edu.cn; Snogisun@tom.com

Abstract. Chaotic neural networks have been proved to be powerful tools for escaping from local minima. In this paper, we first retrospect Chen's chaotic neural network and then propose a novel Gauss-Morlet-Sigmoid chaotic neural network model. Second, we make an analysis of the largest Lyapunov exponents of the neural units of Chen's and the Gauss-Morlet-Sigmoid model. Third, 10-city traveling salesman problem (TSP) is given to make a comparison between them. Finally we conclude that the novel chaotic neural network model is more effective.

1 Introduction

Many combinatorial optimization problems arising from science and technology are often difficult to solve entirely. Hopfield and Tank first applied the continuous-time, continuous-output Hopfield neural network (HNN) to solve TSP [1], thereby initiating a new approach to optimization problems [2, 3]. The Hopfield neural network, one of the well-known models of this type, converges to a stable equilibrium point due to its gradient decent dynamics; however, it causes sever local-minimum problems whenever it is applied to optimization problems.

Chaotic neural networks have been proved to be powerful tools for escaping from local minima. M-SCNN has a more powerful performance than Chen's in solving combinatorial optimization problems, especially in searching global minima of continuous nonlinear function and traveling salesman problems [4]. Now we do research on the activation function in detail.

In this paper, we mainly make research on the effect of Gauss function on the performance of the network. We first review the Chen's chaotic neural network. Second, we propose a novel chaotic neural network model. Third, the time evolution figures of their largest Lyapunov exponents of the neural units are given. At last, we apply them to 10-city traveling salesman problem (TSP) in order to make a comparison. Finally we conclude that the novel chaotic neural network model we proposed is more valid. Because the wavelet function is a kind of basic function, for any function $f(x) \in L_2(R)$ and any wavelet Ψ, the known formula can be described as follows:

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{j,k} \Psi_{j,k}(x) \tag{1}$$

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCS 4113, pp. 115 – 125, 2006.

2 Chaotic Neural Network Models

In this section, two chaotic neural network models are given. And the first is proposed by Chen, the second proposed by ourselves.

2.1 Chaotic Simulated Annealing with Decaying Self-coupling

Chen and Aihara's transiently chaotic neural network [5] is described as follows:

$$x_i(t) = f(y_i(t)) = \frac{1}{1+e^{-y_i(t)/\varepsilon}} \tag{2}$$

$$y_i(t+1) = ky_i(t) + \alpha\left[\sum_j W_{ij}x_j + I_i\right] - z_i(t)(x_i(t) - I_0) \tag{3}$$

$$z_i(t+1) = (1-\beta)z_i(t) \tag{4}$$

where $x_i(t)$ is output of neuron i; $y_i(t)$ denotes internal state of neuron i; W_{ij} describes connection weight from neuron j to neuron i, $W_{ij} = W_{ji}$; I_i is input bias of neuron i, a a positive scaling parameter for neural inputs, k damping factor of nerve membrane, $0 \le k \le 1$, $z_i(t)$ self-feedback connection weight (refractory strength) ≥ 0, β damping factor of $z_i(t)$, $0 < \beta < 1$, I_0 a positive parameter, ε steepness parameter of the output function, $\varepsilon > 0$.

2.2 Gauss-Morlet-Sigmoid Chaotic Neural Network Model (G-M-SCNN)

Gauss-Sigmoid chaotic neural network is a novel model proposed by ourselves, described as follows:

$$y_i(t+1) = ky_i(t) + \alpha\left[\sum_j W_{ij}x_j + I_i\right] - z_i(t)(x_i(t) - I_0) \tag{5}$$

$$z_i(t+1) = (1-\beta_1)z_i(t) \tag{6}$$

$$x_i(t) = f(y_i(t)) \tag{7}$$

$$f(y_i(t)) = BG((1+r_i(t))y_i(t)) + S(\mu_0(1+\eta_i(t))y_i(t)) + M(\mu_1(1+\eta_i(t))y_i(t)) \tag{8}$$

$$\eta_i(t+1) = \frac{\eta_i(t)}{\ln(e+\lambda(1-\eta_i(t)))} \tag{9}$$

$$r_i(t+1) = \beta_2 r_i(t) \tag{10}$$

$$G(u) = e^{-h(u-a)^2} \tag{11}$$

$$S(u) = \frac{1}{1+e^{-u}} \tag{12}$$

$$M(u) = e^{-u^2/2}cos(5u) \tag{13}$$

where $x_i(t)$, $y_i(t)$, W_{ij}, α, k, I_i, $z_i(t)$, I_0 are the same with the above. And λ is a positive parameter which controls the speed of this annealing process; $r_i(0)$ is an important parameter of activation function which should be varied with kinds of special optimization problems, $0< \beta_1 \leqslant 1$, $0< \beta_2 <1$. A, μ_0, μ_1 are positive parameters of Morlet wavelet function and Sigmoid function. h, a, b, c are important parameters of *Gauss* function.

3 Research on Lyapunov Exponent of Neural Units

In this section, we mainly make research on the effect of the parameter β_2 of Gauss function on the largest Lyapunov exponents. We make an analysis of the time evolution figures of the neural units ($\alpha = 0$) of Chen's and G-M-SCNN in the same annealing speed of $\beta = \beta_1$ =0.008.

3.1 Chen's Chaotic Neural Unit

The parameters are set as follows:

k =0.6, I_0=0.1, ε =1/250 ,z (0)=0.1, y(0)=0.283.

The time evolution figure of the largest Lyapunov exponent is shown as Fig.1:

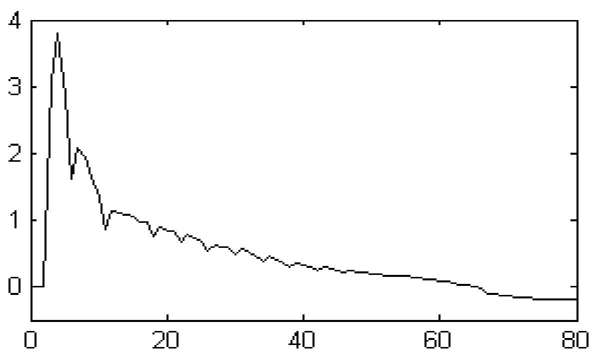

Fig. 1. Lyapunov exponent time evolution figure

3.2 Gauss-Morlet-Sigmoid Chaotic Neural Unit

We make an analysis of the time evolution figures of the neural unit G-M-SCNN with the change of β_2.

(1) The parameters are set as follows:

β_2 =0.1, k =0.092, I_0 =0.8, $y(0)$ =0.283, $z(0)$ =0.8, λ= 0.008, β_1 =0.008, μ_0 =0.8, μ_1 =20, B =10, $r(0)$ =200.5, $\eta(0)$ =0.8, h =0. 2, a =−2. 1, b =5. 0, c =5. 0.

The time evolution figure of the largest Lyapunov exponent is shown as Fig.2.

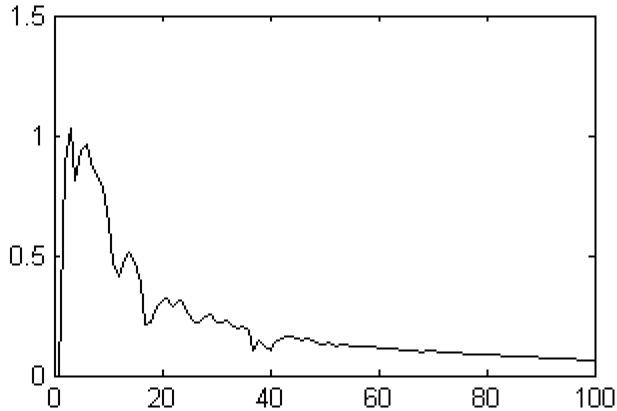

Fig. 2. Lyapunov exponent time evolution figure

(2) The parameters are set as follows:

β_2 =0.5, k =0.092, I_0 =0.8, $y(0)$ =0.283, $z(0)$ =0.8,λ= 0.008, β_1 =0.008, μ_0 =0.8, μ_1 =20, B =10, $r(0)$ =200.5, $\eta(0)$ =0.8, h =0. 2, a =−2. 1, b =5. 0, c =5. 0.

The time evolution figure of the largest Lyapunov exponent is shown as Fig.3.

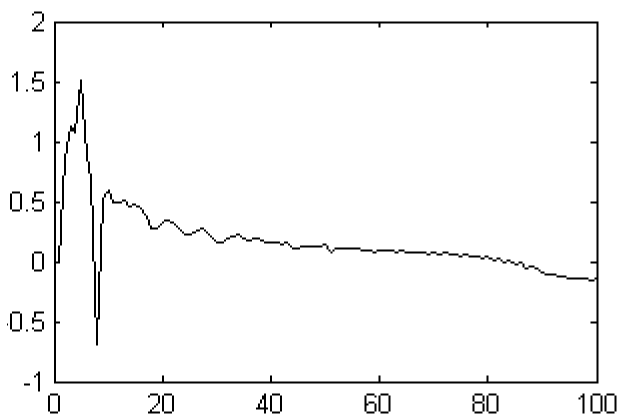

Fig. 3. Lyapunov exponent time evolution figure

(3) The parameters are set as follows:

β_2 =0.9, k =0.092, I_0 =0.8, $y(0)$ =0.283, $z(0)$ =0.8,λ= 0.008, β_1 =0.008, μ_0 =0.8, μ_1 =20, B =10, $r(0)$ =200.5, $\eta(0)$ =0.8, h =0. 2, a =−2. 1, b =5. 0, c =5. 0.

The time evolution figure of the largest Lyapunov exponent is shown as Fig.4.

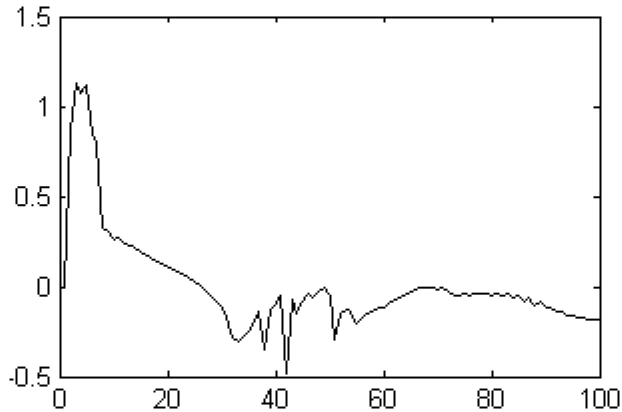

Fig. 4. Lyapunov exponent time evolution figure

Seen from the above analysis, we can conclude that the change of β_2 does have a profound effect on the time evolution of Lyapunov exponent. So, in this paper we will only make analysis of the performance of the network with different β_2 in solving traveling salesman problem (TSP)

4 Application to Traveling Salesman Problem

A solution of TSP with N cities is represented by N×N-permutation matrix, where each entry corresponds to output of a neuron in a network with N×N lattice structure. Assume v_{xi} to be the neuron output which represents city x in visiting order i. A computational energy function which is to minimize the total tour length while simultaneously satisfying all constrains takes the follow form [7]:

$$E=\frac{A}{2}\left(\sum_{x=1}^{N}\left(\sum_{i=1}^{N}v_{xi}-1\right)^2+\sum_{i=1}^{N}\left(\sum_{x=1}^{N}v_{xi}-1\right)^2\right)+\frac{D}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N}d_{xy}v_{xi}v_{y,i+1} \tag{14}$$

where $v_{i0}=v_{in}$ and $v_{i,n+1}=v_{i1}$. A and D are the coupling parameters corresponding to the constraint function and the cost function of the tour length, respectively. d_{xy} is the distance between city x and city y.

The coordinates of 10-city is as follows:

(0.4, 0.4439),(0.2439, 0.1463),(0.1707, 0.2293),(0.2293, 0.716),(0.5171, 0.9414), (0.8732, 0.6536),(0.6878, 0.5219),(0.8488, 0.3609),(0.6683, 0.2536), (0.6195, 0.2634). The shortest distance of the 10-city is 2.6776.

Here are the results of the test about Chen's and G-M-SCNN.

The coupling parameters corresponding to the constraint function and the cost function of the tour length we adopt are set as follows: $A=2.5$, $D=1$.

(1) The parameters of Chen's are set as follows :

$\alpha=0.2$, $k=1$, $I_0=0.5$, $\varepsilon=1/20$, $z(0)=[0.5,0.5]$.

200 different initial conditions are generated randomly in the region [0, 1] when $\beta=0.008$, as are shown in table1. (VN= valid number; GN= global number; VP= valid percent; GP=global percent.)

Table 1. Simulation result of Chen's chaotic neural network

Model	VN	GN	VP	GP
	188	188	94%	94%
	185	185	92.5%	92.5%
	183	183	91.5%	91.5%
	184	184	92%	92%
	181	180	90.5%	90%
Chen's	175	175	87.5%	87.5%
	180	179	90%	89.5%
	189	189	94.5%	94.5%
	187	186	93.5%	93%
	178	178	89%	89%
average	183	182.7	91.5%	91.35%

In order to gain insight into the effect of β_2 on the proposed model, the tests are shown as follows:

(2) The parameters of G-M-SCNN are set as follows:

$k=1, I_0=0.5,\ z(0)=0.1, \lambda=\ 0.008, \beta_1=0.008, \mu_0=0.8, \mu_1=20,\ B=10,\ r(0)=200.5$, $\eta(0)=0.8, h=0.\ 2, a=-2.\ 1, b=5.\ 0, c=5.\ 0$.

Let $\beta_2=0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9$, we can get Table2~10.

Table 2. Simulation result of G-M-SCNN($\beta_2=0.1$)

Model	VN	GN	VP	GP
	187	178	93.5%	89%
	193	188	96.5%	94%
	194	187	97%	93.5%
G-M-SCNN	188	183	94%	91.5%
($B=10$,	190	176	95%	88%
$\beta_2=0.\ 1$)	193	183	96.5%	91.5%
	190	184	95%	92%
	192	181	96%	90.5%
	188	183	94%	91.5%
	189	182	94.5%	91%
Average	190.4	182.5	95.2%	91.25%

Table 3. Simulation result of G-M-SCNN(β_2 =0.2)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 2)	187	180	93.5%	90%
	187	175	93.5%	87.5%
	191	184	95.5%	92%
	187	180	93.5%	90%
	193	188	96.5%	94%
	194	187	97%	93.5%
	188	183	94%	91.5%
	190	176	95%	88%
	192	182	96%	91%
	190	184	95%	92%
average	189.9	181.9	94.95%	90.95%

Table 4. Simulation Result of G-M-SCNN(β_2 =0.3)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 3)	190	183	95%	91.5%
	192	181	96%	90.5%
	187	182	93.5%	91%
	189	183	94.5%	91.5%
	189	175	94.5%	87.5%
	190	183	95%	91.5%
	187	180	93.5%	90%
	187	178	93.5%	89%
	188	175	94%	87.5%
	190	181	95%	90.5%
average	188.9	180.1	94.45%	90.05%

Table 5. Simulation result of G-M-SCNN(β_2 =0.4)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 4)	193	183	96.5%	91.5%
	188	179	94%	89.5%
	182	171	91%	85.5%
	191	187	95.5%	93.5%
	191	180	95.5%	90%
	194	186	97%	93%
	193	188	96.5%	94%
	194	187	97%	93.5%
	187	181	93.5%	90.5%
	188	178	94%	89%
average	190.1	182	95.05%	91%

Table 6. Simulation result of G-M-SCNN(β_2 =0.5)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 5)	193	183	96.5%	91.5%
	191	179	90.5%	89.5%
	193	183	96.5%	91.5%
	193	183	96.5%	91.5%
	193	181	96.5%	90.5%
	187	182	93.5%	91%
	189	183	94.5%	91.5%
	188	175	94%	87.5%
	189	183	94.5%	91.5%
	188	181	94%	90.5%
average	190.4	181.5	95.2%	90.75%

Table 7. Simulation result of G-M-SCNN(β_2 =0.6)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 6)	188	180	94%	90%
	185	171	92.5%	85.5%
	189	180	94.5%	90%
	193	183	96.5%	91.5%
	181	171	90.5%	85.5%
	188	177	94%	88.5%
	188	182	94%	91%
	190	183	95%	91.5%
	193	186	96.5%	93%
	187	179	93.5%	89.5%
average	188.2	178.2	94.1%	89.1%

Table 8. Simulation result of G-M-SCNN(β_2 =0.7)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 7)	190	182	95%	91%
	181	173	90.5%	86.5%
	193	181	96.5%	90.5%
	189	180	94.5%	90%
	190	185	95%	92.5%
	194	187	97%	93.5%
	191	189	95.5%	94.5%
	191	183	95.5%	91.5%
	189	185	94.5%	92.5%
	184	170	92%	85%
average	189.2	181.5	94.6%	90.75%

Table 9. Simulation result of G-M-SCNN(β_2 =0.8)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0. 8)	193	185	96.5%	92.5%
	192	180	96%	90%
	192	185	96%	92.5%
	190	180	95%	90%
	185	175	92.5%	87.5%
	193	183	96.5%	91.5%
	191	181	95.5%	90.5%
	192	178	96%	89%
	193	180	96.5%	90%
	183	174	91.5%	87%
average	190.4	180.1	95.2%	90.05%

Table 10. Simulation result of G-M-SCNN(β_2 =0.9)

Model	VN	GN	VP	GP
G-M-SCNN (B =10, β_2 =0.9)	188	180	94%	90%
	192	182	96%	91%
	189	180	94.5%	90%
	194	183	97%	91.5%
	187	175	93.5%	87.5%
	186	173	93%	86.5%
	192	177	96%	88.5%
	190	179	95%	89.5%
	190	181	95%	90.5%
	190	182	95%	91%
average	189.8	179.2	94.9%	89.6%

An examination of Table2~10 yields the following Table 11.

Table 11. Simulation result of G-M-SCNN

Model		AVP	AGP
Chen's		91.5%	91.35%
G-M-SCNN	β_2 =0.1	95.2%	91.25%
	β_2 =0.2	94.95%	90.95%
	β_2 =0.3	94.45%	90.05%
	β_2 =0.4	95.05%	91%
	β_2 =0.5	95.2%	90.75%
	β_2 =0.6	94.1%	89.1%
	β_2 =0.7	94.6%	90.75%
	β_2 =0.8	95.2%	90.05%
	β_2 =0.9	94.9%	89.6%

(AVP=average valid path percent, AGP=average global path percent)

Seen from the Table11, the proposed model can solve the TSP with a higher valid minimum-path percent and a little lower valid-path percent under all the values of β_2. This means that the proposed model is more valid in only solving TSP than Chen's.

The time evolution figures of the energy function of G-M-SCNN and Chen's in solving TSP are respectively given in Fig.5 and Fig.6.

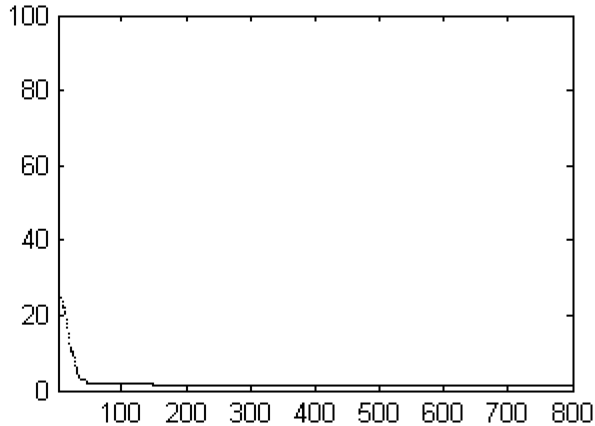

Fig. 5. Energy time evolution figure of G-M-SCNN

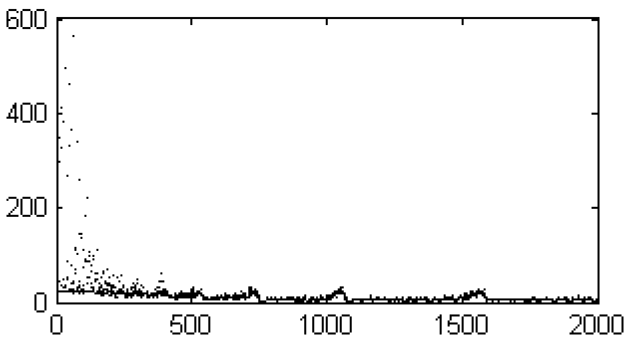

Fig. 6. Energy time evolution figure of Chen's

By comparison, it is concluded that M-SCNN is superior to Chen's model. From the Fig.5, Fig.6, one can see that the velocity of convergence of G-M-SCNN is faster than that of Chen's in solving TSP. Because, the state of Fig.5 become stable in 60 iterations while that of Fig.6 become stable in 1500 iterations.

The superiority of G-M-SCNN contributes to several factors: First, because of the higher nonlinear nature of Gauss function, the activation function of G-M-SCNN has a further performance in solving combinatorial optimization problems than Chen's.

Second, it is easier to produce chaotic phenomenon [8] in that the activation function is non-monotonic. Third, the activation function of G-M-SCNN is varied with time. Third, the wavelet function is a kind of basic function.

5 Conclusions

We have introduced two models of chaotic neural networks. To verify the effectiveness of it, we have made comparison with Chen's model in optimization problems. By comparison, one can conclude that G-M-SCNN is superior to Chen's in searching global minima of continuous nonlinear function.

Different from Chen's model, the activation function of G-M-SCNN is composed by Gauss, Morlet wavelet and Sigmoid. So, besides it has the quality of sigmoid activation, the activation function of G-M-SCNN has a higher nonlinear nature than Sigmoid, which is easier to produce chaotic phenomenon [7] because of its non-monotonic. Due to these factors, G-M-SCNN is superior to Chen's in solving TSP.

References

1. Hopfield, J.J., Tank, D.W.: Neural Computation of Decision in Optimization Problems. Biol. Cybern.52 (1985)141-152
2. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In: Proc. Natl. Acad. Sci., 79 (1982) 2554-2558
3. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In: Proc. Natl. Acad. Sci., 81 (1984) 3088-3092
4. Xu, Y.Q., Ming, S., Duan, G.R.: Wavelet Chaotic Neural Networks and Their Application to Optimization Problems. Lecture Notes in Computer Science, 3971 Springer Berlin (2006) 379-384
5. Wang, L.P., Tian, F.Y.: Noisy Chaotic Neural Networks for Solving Combinatorial Optimization Problems. International Joint Conference on Neural Networks. Italy: IEEE, (2000)37-40
6. Sun, S.Y., Zhen, J.L.: A Kind of Improved Algorithm and Theory Testify of Solving TSP in Hopfield Neural Network. Journal of electron, 1 (1995)73-78
7. Potapove, A., Kali, M.: Robust Chaos in Neural Networks. Physics Letters A, 277 (2000) 310-322