Article ID: 1000 - 8152(2001)06 - 0887 - 04

Improving the Convergence of the Genetic Neural Network in the Crystallizing of Sugar Using Q-Learning *

LUO Fei, MAO Zongyuan and MO Hongqiang
(Department of Automatic Control Engineering, South China University of Technology · Guangzhou, 510641, P.R.China)
LU Ziguang
(College of Electric Engineering, Guangxi University · Nanning, 530004, P.R.China)

Abstract: The crystallizing speed of cane sugar is learned and predicted by the model of feedforward neural network using genetic algorithms. To counter the problem in the model which needs a lot of calculations but has slow speed of convergence, we use Q-learning with reinforcement to decide on the variation probability of genetic algorithms and to increase the convergence speed of learning. The results of the simulation show the effectiveness of the method.

Key words: Q-learning; genetic neural network; convergence; cane sugar
Document code: A

Q 学习对制糖结晶遗传神经网络收敛性的改进

罗 飞 毛宗源 莫鸿强 卢子广
(华南理工大学自动控制工程系·广州,510641) (广西大学电气工程学院·南宁,530004)

摘要: 采用多层前馈遗传神经网络模型对甘蔗制糖结晶速度进行学习和预测,并针对该模型存在的计算量大,收敛慢的问题,采用具有强化作用的 Q 学习确定遗传算法的变异概率,以提高学习的收敛速度,仿真结果表明了该方法的有效性.

关键词: Q 学习; 遗传神经网络; 收敛性; 甘蔗制糖

1 Introduction

It is difficult to determine the boiling duration in the crystallizing process of cane sugar, due to the failure of the concentration-sensor after long-time usage. So far no effective solution to this problem has been found[1]. Multi-layer feed-forward genetic neural networks (GNN's), which have powerful function to identify nonlinear-systems[2-5], can be employed to model the crystallizing process of cane sugar and to estimate the boiling duration. However, it takes a long time for a GNN to converge when the number of its weights is large. We need effective measures to improve its convergence. This object can be achieved by properly modifying the mutation probability of the GNN.

Q-learning, which was proposed by Watkins[6] to solve Information Non-perfect Markovian Decision problems, is an effective reinforcement learning algorithm. In this paper, we use Q-learning algorithm to adaptively tune the mutation probability of the GNN. The simulation results show that this reinforcement learning algorithm improves the convergence of the GNN in modeling the crystallizing process of cane sugar.

2 Reinforcement learning of mutation probability by Q-learning algorithm

2.1 Q-learning algorithm

Suppose that the state set $Z = \{z_t \mid t \geq 0\}$, and the motion set $A = \{a_t \mid t \geq 0\}$. The value of Q depends on the state z_t and the motion a. When motion a is selected for state z_t at moment t, the corresponding value of Q is denoted as $Q_t(z, a)$. The Q-learning algorithm which Watkins[6] proposed consists of the following steps:

1) Set $t = 0$; randomly initialize $Q_t(z, a)$.

2) Observe the present state z_t.

* Foundation item: supported by the National Natural Science Foundation of China(69864001).
Received date: 2000 - 02 - 23; Revised date: 2001 - 02 - 14.

3) Denote the probability of a certain motion $a \in A$ being selected for a certain state z as $P(a \mid z_t)$. There holds

$$P(a \mid z_t) = \frac{\exp[Q_t(z,a)/T]}{\sum_{b \in A} \exp[Q_t(z,b)/T]}, \quad (1)$$

where T is the temperature factor, with which Q can be tuned.

4) Select a motion a_t according to the state z_t and the corresponding probability $P(a \mid z_t)$; figure out enforcement signal r_t (also named as reward function), and then renew the value of Q according to formula (2):

$$Q_{t+1}(z,a) = \begin{cases} (1-\alpha_t)Q_t(z,a) + \alpha_t[r_t + \\ \beta \max_b Q_t(z_t,b)], \ (z,a) = (z_t,a_t), \\ Q_t(z_t,a_t), \ (z,a) \neq (z_t,a_t), \end{cases} \quad (2)$$

where α_t and β are respectively learning factor and discount factor, which satisfy $0 < \alpha_t, \beta < 1$.

5) Let $t = t + 1$, and then go back to (2).

Watkins[6] has proved that if a certain condition is satisfied for the learning factor α_t, say $0 < \alpha_t < 1$, and an infinitely iteration of step 2) can be employed to any (z, a), then $Q_t(z,a)$ will converge with probability 1 to $Q(z,a)$ when $t \to \infty$.

2.2 Deciding on mutation probability by Q-learning

Suppose the state set to be $Z = \{$success, failure, uncertainty$\}$, and the motion set to be $A = \{$mutation, reserve$\}$. The value of the state is selected from the set Z according to the value of the index function $J(t)$. When $J(t) > J(t-1)$, z_t = success; $J(t) < J(t-1)$, z_t = failure; $J(t) = J(t-1)$, z_t = unvarying. The value of the reward function r_t is figured out according to formula (3):

$$r_t = \begin{cases} 2, \text{ success } (J(t) > J(t-1)), \\ 1, \text{ unvarying } (J(t) = J(t-1)), \\ 0, \text{ failure } (J(t) < J(t-1)). \end{cases} \quad (3)$$

If $P(a \mid z_t)$ in formula (1) is employed as the mutation probability of the GNN, and the value of Q in formula (1) is figured out according to formula (2), then r_t will enforce the learning by accelerating the convergence of GNN evolution. Since $Q_t(z,a)$ converges with probability 1 to $Q(z,a)$ when $t \to \infty$, there holds that $J(t) - J(t-1) \to 0$. The Q-learning algorithm therefore improves the convergence performance of the GNN.

3 Multi-layer feed-forward genetic neural network model of the crystallizing process of cane sugar

It is very important to restrict the crystallizing rate to a certain range because the rate determines the crystallizing duration and the product quality. As was showed by Sillin[7], the crystallizing rate K satisfies

$$K = \frac{k_1}{d}\left[\Delta c + 0.5\frac{k_1}{dk_2} - \sqrt{\frac{k_1}{dk_2}(\Delta c + 0.25\frac{k_1}{dk_2})}\right], \quad (4)$$

where k_1, k_2, and d are respectively the diffusion coefficient, a constant and the thickness of boundary membrane. Δc is represented as $\Delta c = c - c_0$, where c is the concentration of oversaturation, and c_0 is the density of saturation.

In fact, the nonlinear relationships between K and the temperature of sugar juicy, K and the purity of sugar juicy, and K and the density of sugar juicy are much more complex than that showed in formula (4). There are over 300 groups of crystallizing data that were obtained in real canesugar process in [7], and the relationships are implicitly expressed within these data. The temperature of sugar juicy ranges between 40℃ and 80℃, the density between 70% and 86%, and the purity 60% and 90%. According to these data, we established a multi-layer feed-forward neural network as shown in Fig. 1. The network is composed of three layers. It consists of three input nodes, four hidden-layer nodes and one output node. The inputs to this network are respectively the temperature x_1, the density x_2 and the purity x_3; the output is crystallizing rate K. Each node uses sigmoid function as its stimulation function $\sigma(x)$ which satisfies $\lim_{x \to -\infty} \sigma(x) = 0$ and $\lim_{x \to +\infty} \sigma(x) = 1$.

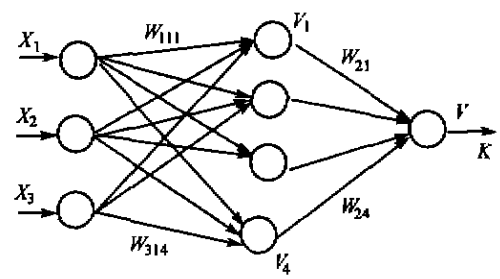

Fig. 1 Feed-forward neural network model of the crystallizing process of cane sugar

We employ a real-coded genetic algorithm[8] to the weight optimization of this network. In a multi-layer feed-forward neural network, each of the hidden-layer nodes can be viewed as a special recognizer of input patterns. In order to prevent these recognizers from being frequently destroyed in crossover, the weights of each of these recognizers should abut against each other on the chromosome[3]. Therefore, the weights are assigned on the chromosome as follows:

w_{11}	w_{21}	w_{31}	v_1	w_1	$\cdots$	w_{1i}	w_{2i}	w_{3i}	v_i	w_i	$\cdots$	w_4

where $w_{ji}(i = 1,\cdots,4; j = 1,2,3)$ is the weight from the input node j to the middle node i; v_i is the increment which is added to the middle node i; w_i is the weight from the middle node i to the output node. The weights of the input and the output are respectively put in the two ends of the line to form a full line of chromosome.

In the genetic algorithm used in Fig. 1, the crossover operation is completed by a scheme called convex combination, in which the individuals of the filial generation are composed of the fine ones of the father generation and the sum total is 1[9]. Supposing that the individuals of the father generation in the crossover operation of the t generation are X_1^t and X_2^t, and that the crossover position is in front of the locus to w_{ji}, and that the individuals of the filial generation are X_1^{t+1} and X_2^{t+1}, we have

$$(w_{ij})_1^{t+1} = a(w_{ij})_1^t + (1 - a)(w_{ij})_2^t, \quad (5)$$

$$(w_{ij})_2^{t+1} = (1 - a)(w_{ij})_1^t + a(w_{ij})_2^t, \quad (6)$$

where $(w_{ij})_l^t(l = 1,2)$ is the value of the locus to w_{ji} in the chromosome of the individuals X_1^t of the father generation, and $a \in [0, 1]$($a = 0.5$ in this paper).

In this GNN, the weights are restricted to the range $(-10.000, 10.000)$, its population size is 100 and fitness function is e^{-J}, where $J = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(K_{ti} - K_i)^2}$ is the training error, K_{ti}, K_i are respectively the ith measured value and the ith output value of crystallizing rate, and n is the sample number. Since $J > 0, e^{-J} \in (0,1]$. The convex combination is employed as its crossover operator, and the crossover possibility is 0.4. The best solution in the population is maintained before selection, while the rest are processed by a roulette wheel with slots sized according to their fitness values.

$P(a \mid z_t)$ in formula (1) is employed as the mutation probability, which accelerates the evolution of the GNN. We can therefore get the estimated crystallizing rate from the output of the GNN more quickly. We adjust the temperature and density of the sugar juicy according to this estimated rate, and therefore realize an automatic control of the sugar process without the concentration-sensor.

4 Simulation results

Randomly select 176 groups of data from those listed in [7] to train the GNN. A simple GA of mutation probability 0.1 and a GA with Q-learning are separately employed to optimize the GNN. The calculation results are showed in Table 1. Then estimate the crystallizing rate with the trained GNN. The performances of their estimation are respectively showed in Fig. 2 and Fig. 3. As shown in Fig. 2 and Fig. 3, both of the GNN's trained respectively by the simple GA and the Q-learning GA fit the process well.

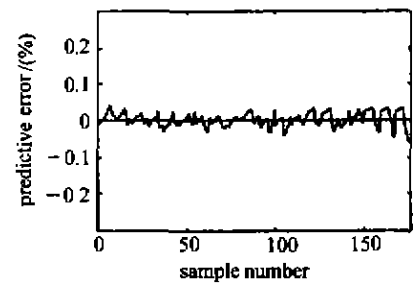

Fig. 2 Predictive error curve before using Q-learning

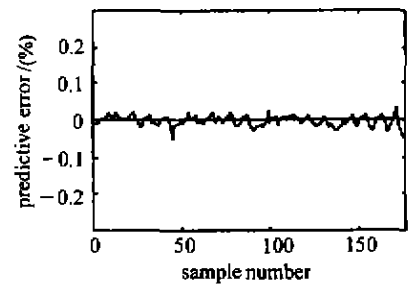

Fig. 3 Predictive error curve before using Q-learning

However, as shown in Table 1, when the GNN is optimized by the simple GA, the trained error usually rests at around 0.025 ~ 0.03, and seldom goes further than 0.01. That is to say, the simple GA is most likely to slump into the local extrema. On the other hand, it takes less iteration steps for the Q-learning GA to achieve the same train error with that of the simple GA. Moreover, the Q-learning GA can reach a smaller error. Therefore, the Q-learning GA performs better than the simple one in the GNN optimization.

Table 1 The convergency comparison of simple GA and GA with Q-learning

Trian error	Iteration stepp	
	Simple GA	Q-learning GA
0 125	350	125
0.085	700	525
0.050	4,000	2,000
0.025	7,000	3,500
0.015	10,000	4,000
0.010	—	6,500

5 Conclusion

In view of the model nonlinearity in the crystallizing process of cane sugar, a genetic neural network model is established in this paper. This model is optimized by the Q-learning GNN, the mutation probability of which is adaptively adjusted by Q-learning algorithm according to the fitness value. The simulation results show that this enforcement learning algorithm accelerates the evolution process of the GNN, so that the calculation time is reduced and the convergence performance is improved.

References

[1] Tanabe K and Kubota H. Control of sugar plant with computers by measuring the density of sugar [J]. Measurement and Instrument, 1993,36(6):44 - 49

[2] Moutana D and Davis L. Training feedforward neural networks using genetic algorihtms [A]. Proceedings of Eleventh International Joint Conference on Artificial Intelligence [C], San Mateo, CA: Morgan Kaufmann, 1989,762 - 767

[3] Yao X. A review of evolutionary artificial neural networks [J]. International Journal of Intelligent Systems, 1993,8(4):539 - 567

[4] Whitley D and Hanso T. Optimizing neural networks using faster, more accurate genetic search [A]. Schaffer J D, ed. Proceedings of the Third International Conference on Genetic Algorithms and Their Applications [C], San Mateo, CA:Morgan Kaufmann,1989,391 - 396

[5] Marshall S J and Harrison R F. Optimization and training of feedforward neural networks by genetic algorithms [A]. The Second IEE International Conference on Artificial Neural Networks [C]. Bournemouth, UK, 1991,39 - 43

[6] Watkins C J C H and Dayan P. Technical note, Q-learning [J]. Machine Learning, 1992,4(8):279 - 292

[7] Chen Shugong. Theory of Modern Sugar-Refining Technics [M]. Beijing: Light Industry Publisher, 1985 (in Chinese)

[8] Chen Guoliang. Genetic Algorithms and Their Applications [M]. Beijing: People's Post and Telecommunication Publisher, 1996(in Chinese)

[9] Chen M S and Liao F H. Neural networks training using genetic algorithms [A]. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics [C], San Diego, CA,1998,3:2436 - 2441

本文作者简介

罗 飞 见本刊 2001 年第 2 期第 248 页.

毛宗源 见本刊 2001 年第 2 期第 165 页.

莫鸿强 见本刊 2001 年第 2 期第 284 页.

卢子广 1963 年生.1985 年和 1988 年于北京钢铁学院获学士和硕士学位.现为广西大学副教授,从事电气传动,鲁棒控制研究.