2007 IEEE Workshop on Nonlinear Circuit Networks
NCN'07, Tokushima, Japan, Dec 7--8, 2007

Neural Gas Containing Two Kinds of Update Functions

Keiko Kanda, Haruna Matsushita and Yoshifumi Nishio
Department of Electrical and Electronic Engineering,
Tokushima University
Email: kanda@ee.tokushima-u.ac.jp, haruna@ee.tokushima-u.ac.jp, nishio@ee.tokushima-u.ac.jp

***Abstract*—In this study, we propose a new Neural Gas algorithm which contains two kinds of update functions (called TKF-NG). One converges rapidly, and the other converges slowly. In this method, all neurons learn according to winner's character at each learning. We confirm that the proposed Neural Gas can obtain more effective learning results than the conventional Neural Gas.**

I. Introduction

Recently, the Neural Gas network (NG) [1] has attracted attention as an image-recognition technology. In the learning, NG uses one type of neurons. However, in the real world, an elephant, which has a slow heart rate, and a mouse, which has a rapid heart rate, have different senses of time.

In this study, we propose a new Neural Gas algorithm; Neural Gas containing two kinds of update functions (called TKF-NG). One converges rapidly, and the other converges slowly. In this method, all neurons learn according to winner's character at each learning. We confirm that TKF-NG can obtain more effective learning results than the conventional NG.

II. Neural Gas Containing Two Kinds of Update Functions (TKF-NG)

We explain the learning algorithm of the proposed method, which has two kinds of update functions whose features are different. The network consists of a set A containing M neurons; $A = \{c_1, c_2, \cdots, c_M\}$. The neurons of the set A are classified into a set A_1 and a set A_2 at random. A_1 contains m_1 neurons, and A_2 contains m_2 neurons, namely $M = m_1 + m_2$. The neurons which belong to A_1 converges rapidly and the neurons which belong to A_2 converges slowly. To be specific, the convergence rate of the learning rate is different between A_1 and A_2. Each neuron c_i $(i = 1, 2, \cdots, M)$ has a d-dimensional weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$. The initial values of the weight vectors are given at random. The range of the elements of d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ are assumed to be from 0 to 1.

(TKF-NG1) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(TKF-NG2) Euclid distances d_{ji} between the input vector $\boldsymbol{x}_j$ and all the weight vectors $\boldsymbol{w}_i$ are calculated;

$$d_{ji} = \|\boldsymbol{x}_j - \boldsymbol{w}_i\|. \tag{1}$$

(TKF-NG3) Each neuron c_i is assigned a rank $k_i = 0, \cdots, (M-1)$ depending on their d_{ji}. k_i denotes the rank of neuron c_i. The rank of 0 indicates the neuron closest to $\boldsymbol{x}_j$ and the rank of $(M-1)$ indicates the neuron farthest from $\boldsymbol{x}_j$. The closest neuron to $\boldsymbol{x}_j$, namely $k_i = 0$, is defined as a winner.
(TKF-NG4) The weight vectors of the neurons are updated according to

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + E_F(t) h_\lambda(t,k)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \tag{2}$$

where learning rate $E_F(t)$ is decided by the winner's character;

$$E_F(t) = \begin{cases} E_0 \left(\dfrac{E_f}{E_0}\right)^{\frac{t}{T}}, & \text{winner} \in A_1, \\ (p + E_0 - E_f)\left(\dfrac{p}{p + E_0 - E_f}\right)^{\frac{t}{T}} & \\ \quad + E_f - p, & \text{winner} \in A_2, \end{cases} \tag{3}$$

where t is the time step and T is the total number of training steps. E_0 and E_f are the initial and the final values of $E(t)$, respectively. The fixed parameter p controls the convergence rate of the learning rate $E_F(t)$. In this way, all neurons learn according to the winner's character. It is provided $p \geq E_f$. As a result of Eq. (3), the learning rate $E_F(t)$ of A_1 converges rapidly. On the other hand, the learning rate $E_F(t)$ of A_2 decreases with time linearly and converges slowly. In other words, if the winner belongs to A_1, all neurons learn based on the character of A_1, otherwise, all neurons learn based on the character of A_2. $h_\lambda(t,k)$ is the neighborhood ranking function described as

$$h_\lambda(t,k) = \exp\left(-\frac{k_i}{\lambda(t)}\right). \tag{4}$$

$\lambda(t)$ controls the width of the neighborhood function;

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_0}{\lambda_f}\right)^{\frac{t}{T}}, \tag{5}$$

where λ_0 and λ_f are the initial and the final values of $\lambda(t)$, respectively.
(TKF-NG5) If $t < T$, return to (TKF-NG1).

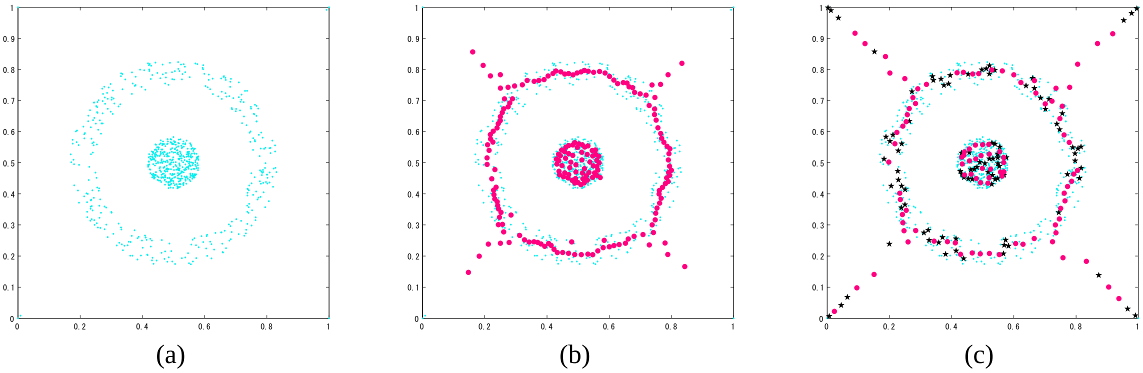

Fig. 1. Experimental result of two algorithms for Target data. (a) Input data. (b) Conventional NG. (c) TKF-NG. Points and stars correspond to neurons belonging to A_1 and A_2, respectively.

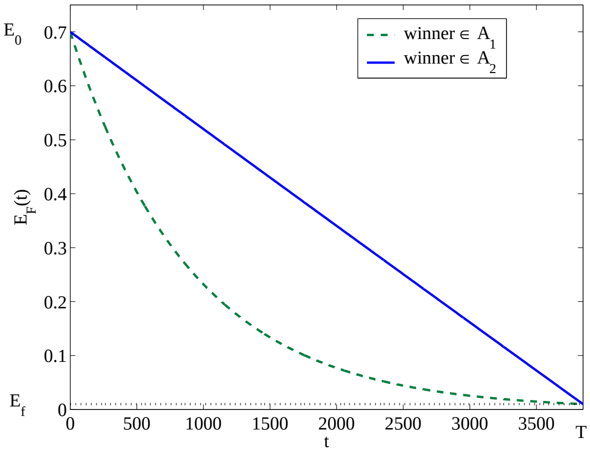

Fig. 2. Learning rate $E_F(t)$ according to Eq. 3.

III. Experimental Results

We carry out learning simulation for the 2-dimensional input data, Target data set [2] shown in Fig. 1(a).

Both the conventional NG and TKF-NG have 200 neurons. In the TKF-NG, the numbers of neurons included in A_1 and A_2 are equal, namely $m_1 = m_2 = 100$. The parameters of the learning are chosen as follows; $E_0 = 0.7$, $E_f = 0.01$, $\lambda_0 = 70$, $\lambda_f = 0.2$, $p = 50$. We use the same E_0, E_f, λ_0, and λ_f to the conventional NG and TKF-NG for the comparison.

The learning results of the conventional NG and TKF-NG are shown in Figs. 1(b) and (c), respectively. In Fig. 1(c), we can see that TKF-NG has two kinds of neurons. In these figures, we can see that some neurons of TKF-NG are attracted to the corner input data more strongly than the conventional one. Thus, TKF-NG can obtain more effective result reflecting the features of input data than the conventional NG.

Meantime, Fig. 2 shows the learning rate $E_F(t)$ according to Eq. (3) with above parameters when winner belongs to A_1 (shown as points in Fig. 1(c)) and belongs to A_2 (shown as stars in Fig. 1(c)). From this figure, we can see that $E_F(t)$ of A_1 (shown as a dashed line) converges rapidly, and $E_F(t)$ of A_2 (shown as a solid line) decreases with time linearly and converges slowly. If the neuron belonging to A_1 becomes winner at time t, all the neuron update with the learning rate $E_F(t)$ according to the dashed line in Fig. 2. Conversely, if the neuron belonging to A_2 becomes winner at time t, all the neuron update with the learning rate $E_F(t)$ according to the solid line in Fig. 2. In other words, TKF-NG learns the input data according to two kinds of features of neuron's update functions.

Furthermore, we calculate quantization error Q_e [3] to evaluate the learning performance of TKF-NG in comparison with the conventional NG. This measures the average distance between each input vector and its winner.

TABLE I
Quantization Error Q_e of two algorithms for Target data

	Conventional NG	TKF-NG
Q_e	0.0162	0.0128

In Table I, the quantization error of TKF-NG is smaller than the conventional NG. In this case, the improvement rate from the conventional NG to TKF-NG is 21.0%.

IV. Conclusions

In this study, we have proposed a new Neural Gas algorithm; Neural Gas containing two kinds of update functions (TKF-NG). We have applied TKF-NG to Target data set and confirmed that TKF-NG can obtain more effective results than the conventional NG in the quantization error. From these results, we can say that NG containing neurons with plural features of the update functions can learn the input data more accurately than NG containing neurons with only one feature.

Acknowledgment

This work was partly supported by Yamaha Music Foundation.

References

[1] T. M. Martinetz and K. J. Schulten, "A "neural-gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors," *Artificial Neural Networks*, pp. 397–402, 1991.

[2] A. Ultsch, "Clustering with SOM: U*C", *Proc. Workshop on Self-Organizing Maps*, pp. 75–82, 2005.

[3] T. Kohonen, *Self-organizing Maps*, 2nd ed., Berlin, Springer, 1995.