

Rough winner-take-all for hardware oriented vector quantization algorithm

Hakaru Tamukoh^{1a)}, Keiichi Horio², Takeshi Yamakawa²,
and Masatoshi Sekine¹

¹ Institute of Engineering, Tokyo University of Agriculture and Technology, 2–24–16
Nakamachi, Koganei-shi, Tokyo, 184–8588, Japan

² Graduate School of Life Science and System Engineering, Kyushu Institute of Tech-
nology, 2–4 Hibikino, Wakamatsu-ku, Kitakyushu, 808–0196, Japan

a) tamukoh@cc.tuat.ac.jp

Abstract: In this paper, we propose a hardware oriented vector quantization algorithm employing rough-winner-take-all neural network. The proposed algorithm is almost same as K-means clustering which is the simplest vector quantization. The only different point is that the proposed method employs rough-winner-take-all as the substitute of ordinary winner-take-all. In a rough-winner-take-all strategy, the winner is roughly selected in the early learning stage and is strictly assigned in the later stage. The simulation results show that the quantization performance of the proposed method is nearly equal to Neural Gas which is an excellent vector quantization. Besides, the proposed method can be realized as an extra mode of existing K-means or Self-Organizing Map hardware by changing its winner-take-all controlling.

Keywords: vector quantization, rough-winner-take-all, WTA, k-means, neural gas, self-organizing map, digital hardware

Classification: Science and engineering for electronics

References

- [1] K.-L. Du, “Clustering: A neural network approach,” *Neural Networks*, vol. 23, pp. 89–107, 2010.
- [2] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, ““Neural-Gas” network for vector quantization and its application to time-series prediction,” *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 558–569, 1993.
- [3] T. Kohonen, *Self-Organizing Maps 3rd ed.*, Berlin, Springer, 2001.
- [4] H. Tamukoh, K. Horio, and T. Yamakawa, “Fast learning algorithms for self-organizing map employing rough comparison WTA and its digital hardware implementation,” *IEICE Trans. Electron.*, vol. E87-C, no. 11, pp. 1787–1794, 2004.
- [5] M. Porrman, U. Witkowski, and U. Rückert, “A massively parallel architecture for self-organizing feature maps,” *IEEE Trans. Neural Netw.*, vol. 5, pp. 1110–1121, 2003.
- [6] H. Hikawa, “FPGA implementation of self organizing map with digital phase locked loops,” *Neural Networks*, vol. 18, no. 5-6, pp. 514–522, 2005.

1 Introduction

Vector quantization (VQ) is one of the most important techniques for data compression. To realize an efficient VQ, a learning approach is effective at deciding the reference vector's distribution [1]. Therefore, neural network based VQ methods (e.g. Neural Gas (NG) [2], Self-Organizing Maps (SOM) [3], K-means clustering [1]) have been proposed and applied to wide range of applications, such as speech and image processing and so on.

Learning algorithms of K-means clustering and SOM are based on winner-take-all (WTA). K-means clustering is the simplest VQ, but the learning result has many local minima and its quantization performance is quite low. SOM achieves better quantization than K-means clustering. However, the quantization performance of SOM is less than NG which involves a fixed two dimensional topology reference vectors on a competitive layer [3].

NG is a topology-free approach and the learning result has no local minima [2]. Thus, it is an excellent VQ method. However, algorithmic complexity is high because NG uses a ranking function which involves a global sorting. The sorting hardware requires much calculation time and hardware area. Therefore, a hardware implementation of NG becomes high in complexity.

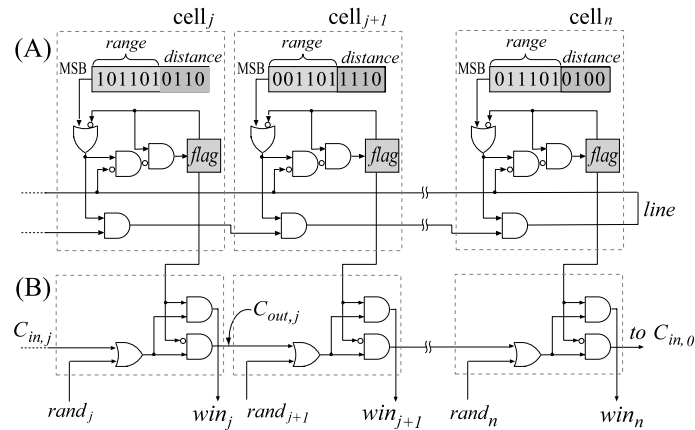
In this paper, we propose a new vector quantization method for an efficient digital hardware implementation employing rough-winner-take-all neural network. Quantization performance of the proposed method is superior to both of K-means clustering and SOM, and is nearly equal to NG. Besides, the proposed method can be realized as an extra mode of SOM hardware by changing its winner-take-all controlling.

2 Rough-winner-take-all

In this section, we describe rough-winner-take-all (RWTA) [4] which is the most important technique of the proposed VQ.

Figure 1 (A) and (B) show a digital hardware architecture of RWTA. Each cell consists of a shift register called *distance* which stores the distance between an input vector and a reference vector, and also consists of one bit memory called *flag* which represents the cell whether to be the winner or not. At first, each *distance* register loads the distance, and each *flag* is set to '1'. Comparison starts from the most significant bits (MSBs) of the *distance* registers. MSBs are connected to an AND-circuit-chain called *line*. If the *line* is '0' then all cells whose MSBs are '1' reset their flags to '0'. It means that these cells are no longer winner candidates. After resetting the *flag*, each *distance* are shifted left by one bit, whereby the next bit enters the MSB position. After all bits of the *distance* are handled, only one unit which has minimum number should make its flag '1', thus it is a *strict* WTA.

On the other hand, RWTA handles only the *range* bits. The *range* represents the number of bits for the comparison on the digital hardware implementation of RWTA. The rest of the unhandled bits becomes quantization error, thus it is a *rough* comparison. Figure 1 (C) shows an algorithm



```

repeat{
    if ( own MSB==1 AND line==0 ) then
        flag := 0;
        shift distance one bit left
    } until range(t) bit in distance
(C)

```

Fig. 1. Digital hardware architecture of RWTA; (A) searches winner candidates, (B) selects only one winner from the winner candidates and (C) is the algorithm flow.

flow of RWTA, and it is also defined by a following equation.

$$c = \arg \min_j (distance_j \gg (D_a - range(t))), \quad (1)$$

where, ‘ \gg ’ means a logical right shift operation and D_a represents a number of the *distance* register bits.

If several winner candidates (i.e. *flag* = ‘1’ cell) are selected after the comparison, only one of them is assigned to the final winner at random by using Figure 1(B). The *rand* signal is generated by Linear Feedback Shift Resister which randomly selects the only one *cell_k* and assign *rand_k* = ‘1’, the other unassigned ones to ‘0’. Then, the only one *win_k* becomes ‘1’. RWTA selects the final winner from winner candidates stochastically as the *range* bit reduces. This RWTA’s feature has a positive influence on quantization performance of the proposed VQ.

3 Rough-winner-take-all neural network

In this section, we propose a new VQ algorithm employing rough-winner-take-all neural network. The basic algorithm of the proposed method is almost same as K-means clustering which is the simplest VQ. The only different point is that the proposed method employs RWTA as the substitute of WTA.

To perform the learning, reference vectors are initialized randomly at first. Then, the learning changes reference vector’s position roughly by a large

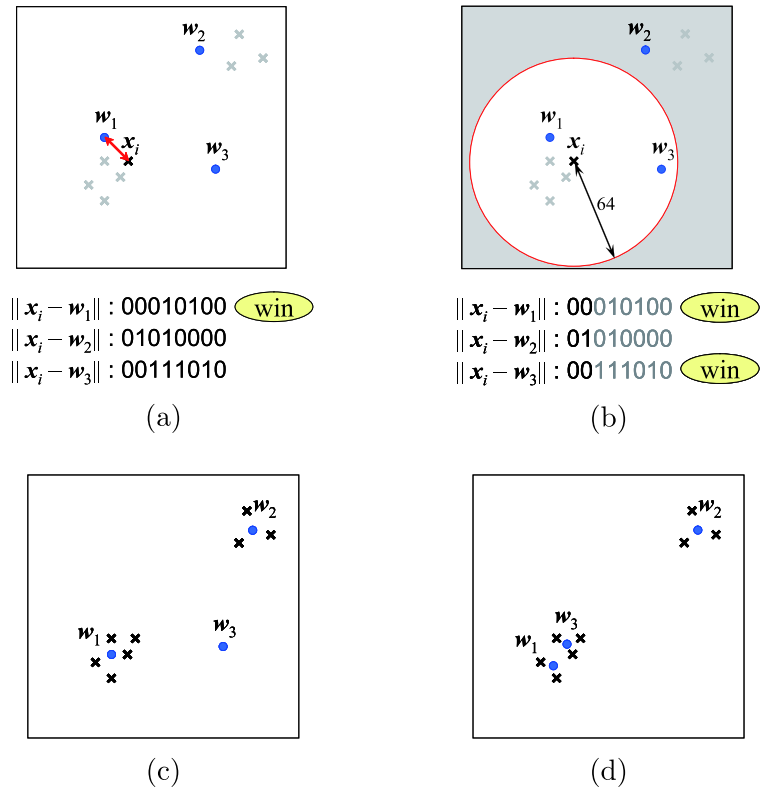


Fig. 2. Sketch of winner definition in the early learning stage; (a) K-means clustering, (b) proposed method, $range(t) = 2$, and learning results in the latter stage; (c) K-means clustering with local minimum, (d) proposed method.

learning rate in the early stage. In other words, a strict winner definition is not required in the early stage, but is required in the latter stage to realize an accurate vector quantization. In that sense, it suits well when the proposed method performs *rough* comparison in the early stage and performs *strict* comparison in latter stage. This RWTA strategy of the proposed method yields two advantages over K-means clustering.

The first advantage is an improvement of the vector quantization performance by a vanishing the local minima. Figure 2 (a) and (b) show a sketch of winner definition in the early learning stage. w_1 , w_2 and w_3 represent reference vectors and x-points represent input vectors. In figure 2 (a), K-means clustering performs *strict* comparison WTA and selects w_1 as the winner. In this initialization case, w_3 will be never chosen as the winner. Therefore, w_3 will become local minimum which largely reduces vector quantization performance (figure 2 (c)). On the other hand, in figure 2 (b), the proposed method performs *rough* comparison WTA with $range(t) = 2$, where eventually the final winner is stochastically selected from the winner candidates w_1 and w_3 . Therefore, the w_3 obtains an opportunity to learn the input vectors, and the proposed method vanishes the local minimum (figure 2 (d)).

The second advantage is a reduction in the calculation cost for WTA. The digital hardware architecture of WTA takes 1 clock cycle per 1 bit compari-

son. For example, in the case of Figure 2 (a), K-means clustering takes 8 clock cycles for *strict* comparison. On the other hand, the proposed method shown in figure 2 (b) takes only 2 clock cycles. Therefore, the proposed method drastically reduces the calculation cost for WTA by employing RWTA.

In this paper, we propose an online and a batch learning algorithm of rough-winner-take-all neural network.

3.1 Online learning version

The processing procedure of the online learning is as follows:

0: Initializing all reference vectors \mathbf{w}_j ($j = 1, \dots, J$) using random values.

1: Calculating the distance d_j between the input vector \mathbf{x} and the reference vector \mathbf{w}_j . The input vector \mathbf{x} is randomly selected from the set of learning data \mathbf{x}_i ($i = 1, \dots, I$). The distance d_j between \mathbf{x} and \mathbf{w}_j is represented by:

$$d_j = \|\mathbf{x} - \mathbf{w}_j\|. \quad (2)$$

2: Selecting a winner using RWTA. RWTA decides the winner based on d_j and $range(t)$, and assigns a winner flag by the following equation.

$$h_j = \begin{cases} 1, & \text{if } \mathbf{w}_j \text{ is winner for } \mathbf{x} \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

The parameter $range(t)$ should be monotonically increased according to a learning step t . For example, it is defined by:

$$range(t) = \left\lceil D_a \cdot \frac{t}{T} \right\rceil, \quad (4)$$

where, T represents the total number of learning steps.

3: Updating the reference vectors using the following equation.

$$\mathbf{w}_j^{new} = \mathbf{w}_j^{old} + \alpha(t) \cdot h_j \cdot (\mathbf{x} - \mathbf{w}_j^{old}), \quad (5)$$

where, \mathbf{w}_j^{new} and \mathbf{w}_j^{old} are the reference vector after and before updating, respectively. $\alpha(t)$ is the learning coefficient.

4: The procedures 1 to 3 are repeated until that all data are selected. In the iteration, the learning rate $\alpha(t)$ decreases with learning steps. For example, it is defined by:

$$\alpha(t) = \alpha_i \left(\frac{\alpha_f}{\alpha_i} \right)^{\frac{t}{T}}, \quad (6)$$

where α_i and α_f are the initial and final values of the learning coefficient, respectively.

The proposed method updates only the winner in the step 3 while SOM updates the winner and its neighborhoods. This topology-free approach and the stochastic winner selecting achieves a high-performance quantization.

3.2 Batch learning version

The processing procedure of the batch learning is as follows:

- 0: Initializing all reference vectors \mathbf{w}_j ($j = 1, \dots, J$) using random values.
- 1: Calculating the distances $d_{j,i}$ between \mathbf{w}_j and the applied input vectors \mathbf{x}_i by equation (2).
- 2: Selecting a winner using RWTa and assigns a winner flag by equation (3).
- 3: Updating all reference vectors using the following equation after applying all input vectors.

$$\mathbf{w}_j^{new} = \frac{\sum_{i=1}^I h_{j,i} \mathbf{x}_i}{\sum_{i=1}^I h_{j,i}} \quad (7)$$

- 4: Repeating from step 1 to step 3 until T .

4 Simulation results and discussion

To verify the vector quantization performance, we achieved following simulations. The 100 reference vectors represent a data distribution in \mathbb{R}^2 that consists of 25 separated square shaped clusters. The calculation accuracy of reference vector is set to 16 bits. In each simulation, well optimized parameters were used, and the total number of learning steps and input vectors are set to $T = 100$ and 10,000, respectively. Figure 3 shows comparison results among four methods. The results show that the proposed method had no local minima, and the quantization performance was nearly equal to NG which is the excellent VQ.

In SOM hardware, two types of WTA circuit are often used for its implementation. First one is a bit-serial word-parallel architecture [4, 5] which was shown in Figure 1. In this architecture, RWTa can be realized by the ordinary WTA circuit by changing its controlling. The second is a binary comparator tree architecture [6]. By applying a mask bit to *distance* register, it can be used as RWTa. For instance, “11000000” AND mask is same as $range(t) = 2$ operation. The bit masking takes few hardware resources and it is easy to implement. Therefore, existing SOM hardware [4, 5, 6] can be used as a high-performance VQ by replacing WTA with RWTa.

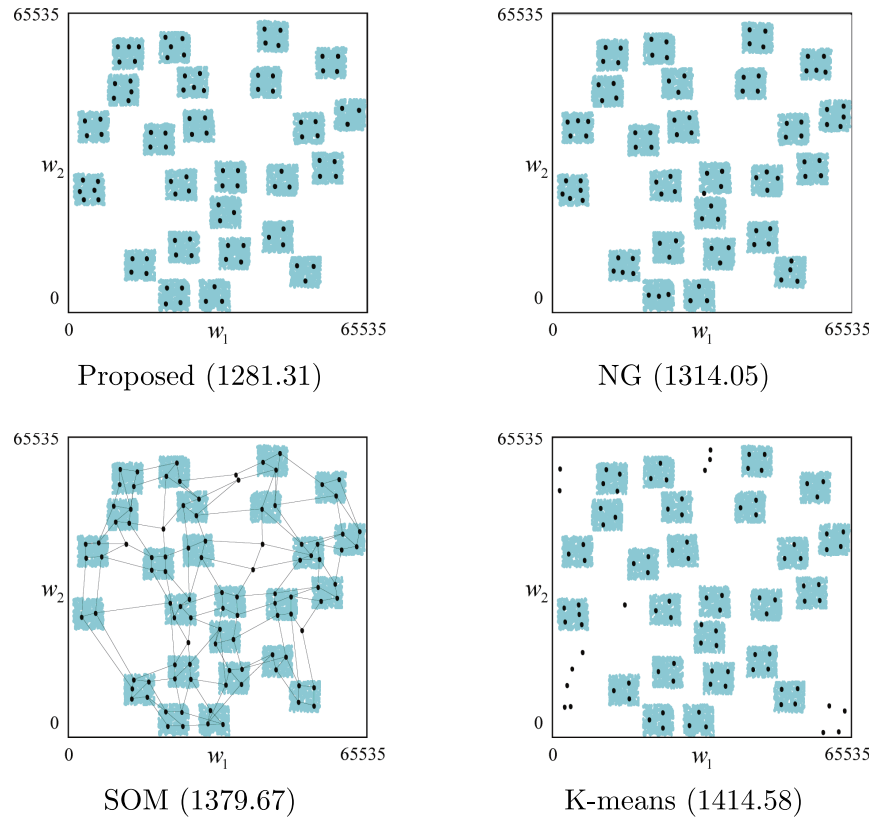
5 Conclusion

In this paper, we propose the new vector quantization algorithm for efficient digital hardware implementation. Simulation results show that the vector quantization performance of the proposed method is superior to K-means clustering and SOM, and is nearly equal to NG. Moreover, we show that the proposed method can be realized as the extra mode of SOM hardware by changing its WTA controlling.

In future work, we will implement K-means clustering, SOM and proposed method into a reconfigurable platform, and evaluate with actual applications.

	Online learning	Batch learning
Proposed	1281.60 (13.87)	1316.34 (18.87)
NG	1315.40 (32.83)	1298.15 (15.05)
SOM	1380.92 (23.96)	1373.04 (24.43)
K-means	1400.85 (38.71)	1530.44 (52.31)

(a)



(b)

Fig. 3. Evaluation of quantization performances among four methods. (a) Average value of the mean square error and the standard deviation in brackets over 100 trials changing data distributions. (b) Example of quantization results and mean square errors. The light blue and black dots represents data clusters and reference vectors, respectively. The lines between reference vectors represents neighborhood in SOM. The proposed and NG had no local minima and also achieved good vector quantization.

Acknowledgments

This work was supported by Grant-in-Aid for Young Scientists (B) 20700207.