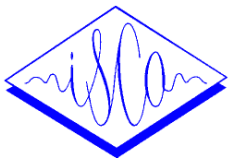

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

THE OPTIMAL GAIN SEQUENCE FOR FASTEST LEARNING IN CONNECTIONIST VECTOR QUANTISER DESIGN

Lizhong Wu Frank Fallside

Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, UK.
E-mail: LZW or FALLSIDE@uk.ac.cam.eng

ABSTRACT

Kohonen's self-organising algorithm has been widely used for the design of connectionist vector quantisers (CVQ). One of its features is that the weight update gain sequence $\eta^{(m)}$ is a decreasing function of the number of iterations, and if incorrectly chosen can lead to very long training times. Here we derive the time-optimal gain sequence and demonstrate its efficacy for a number of cases. It is demonstrated that the new method is time optimal and that its performance tends to that of VQ with the LBG algorithm. Finally the CVQ for linear predictive data with respect to the Itakura distance measure is applied to a multipulse linear predictive speech coder using data from the TIMIT database. Comparisons are made of waveforms and rate distortion functions.

1 INTRODUCTION

Kohonen's algorithm [6] is described by the following 3 steps:
(a) Initialize all weight vectors W_k, $k = 1, \ldots, L$ to small random values, where L is the number of neurons (the size of a codebook);
(b) For each input vector X_n, $n = 1, \ldots, N$, find W^* with a minimal distance to the X_n, and update the W^* and its neighbourhood denoted by N_c, with

$$\Delta W_k^{(m)} = -\eta^{(m)} \nabla_{W_k} D\left(X_n, W_k^{(m-1)}\right), k \in N_c; \quad (1)$$

(c) Repeat step (b) until the decrease of the quantising distortion is less than some given threshold.

The gain sequence $\eta^{(m)}$ is a slowly decreasing function of the number of iterations. Its proper choice is usually determined by trial and error adjustment. Linear, exponential and centrally adaptive forms have been proposed for this parameter [2]. Lucas and Kittler [5] made a comparison of these three forms. Wu and Ganesan [7] experimentally evaluated the efficiency of Kohonen's algorithm for the CVQ design and found that its rate-distortion curve saturated very quickly, and its rate-distortion performance was much worse than that of the VQ designed using the LBG algorithm [4].

In this paper, we derive an optimal form of the gain sequence which gives fastest learning in minimizing the distortion of the CVQ. The result is given for batch learning where the weights are updated only after each complete presentation of the input vectors and for continuous learning where the weights are updated after each input vector. (The same result has recently been given for batch learning [9]). We also study the gain sequence of the learning process with respect to the Itakura distortion measure.

Simulations are given to show the performances of CVQs with the proposed algorithms and to compare with VQs designed using the LBG algorithm. Both Gauss-Markov samples and speech from the TIMIT database are used in the simulations. The resulting CVQ for linear predictive data is applied to a multipulse linear predictive speech coder.

2 THE GAIN SEQUENCE FOR FASTEST LEARNING

Kohonen's self-organizing mapping is "topology-preserving". Neurons with similar weights are in a neighbourhood. So there is some correlation among neurons. In CVQ design, aiming at information compression, we expect that little redundancy exists between codewords. However, for a highly correlated source, although the redundancy between vectors can be reduced by extending the dimension of the vectors, memory requirments and computational complexity will increase exponentially with the dimension. An efficient way of removing the redundancy between vectors is to employ an adaptive vector quantiser [8]. Here we assume the W_k are independent of each other, and that there are no interconnections between output nodes. Therefore, our updating is carried out for the W^* only.

The $\eta^{(m)}$ is defined as a sequence of scale parameters in Kohonen's algorithm. In order to establish its accurate form, we instead use a $p \times p$ matrix, where p is the dimension of W. We also assume that the $\eta^{(m)}$ is associated with the neuron which is going to be updated with $\eta^{(m)}$, so $\eta^{(m)}$ is expressed as $\eta_k^{(m)}$ in the following.

2.1 Optimal Quantisation

For the Euclidean squared error measure,

$$d(X_n, W_k) = (X_n - W_k)^T (X_n - W_k) \quad (2)$$

Let the encoding function $e(\cdot)$ partition the input space $\mathcal{X}$ into L subspaces $P_k = \{X_n : e(X_n) = k\}$, so the distortion is

$$D = \sum_{k=1}^{L} D_k = \sum_{k=1}^{L} \sum_{X_n \in P_k} d(X_n, W_k) \quad (3)$$

Letting

$$\nabla_{W_k} D = 0 \quad (4)$$

leads to the optimal value of W_k as

$$W_k' = \arg \min_{W_k \in \mathcal{W}} \{D\} = \frac{1}{\| P_k \|} \sum_{X_n \in P_k} X_n \quad (5)$$

where $\| P_k \|$ is the cardinal number of the P_k and $\mathcal{W} = \{W_k, k = 1, \ldots, L\}$. Eqn(5) presents the generalized centroid of all input vectors encoded into the neuron k, and is the same as the formula for the reproduction alphabet of the VQ designed by the LBG algorithm [4]. So the VQ with the LBG design algorithm is optimal in the sense of eqn(4), which meets necessary but not sufficient conditions for optimality.

2.2 Batch Learning

In batch learning, the W_k are updated only after each complete presentation of input vectors. At the m^{th} updating

$$W_k^{(m)} = \left(I - 2\eta_k^{(m)} \| P_k^{(m)} \|\right) W_k^{(m-1)} + 2\eta_k^{(m)} \sum_{X_n \in P_k^{(m)}} X_n \tag{6}$$

The decrease of the distortion caused by the m^{th} updating is

$$\Delta D_k^{(m)} = \left[\nabla_{W_k} D_k^{(m-1)}\right]^T \left(\frac{1}{2}(\eta_k^{(m)} + [\eta_k^{(m)}]^T) - \| P_k^{(m)} \| [\eta_k^{(m)}]^T \eta_k^{(m)}\right) \left[\nabla_{W_k} D_k^{(m-1)}\right] \tag{7}$$

We find that the optimal $\eta_k^{(m)}$, $W_k^{(m)}$ and $\Delta D_k^{(m)}$ are

$$\eta'^{(m)}_k = \arg \max_{\eta_k^{(m)} \in \mathcal{R}_{pp}} \{\Delta D_k^{(m)}\} = \frac{I}{2 \| P_k^{(m)} \|} \tag{8}$$

$$W'^{(m)}_k = \frac{1}{\| P_k^{(m)} \|} \sum_{X_n \in P_k^{(m)}} X_n \tag{9}$$

$$\max\{\Delta D_k^{(m)}\} = \frac{1}{4 \| P_k^{(m)} \|} \| \nabla_{W_k} D_k^{(m-1)} \|^2 \tag{10}$$

where $\mathcal{R}_{pp}$ is a $p \times p$ dimensional real space. Eqn(9) is same as eqn(5), so it is also optimal in the sense of eqn(4). Therefore, if the gain sequence is defined as in eqn(8), because the decrease of the distortion is maximized, the learning process will converge fastest, and the rate-distortion performance of the resulting CVQ will be the same as that of the VQ using the LBG design algorithm.

2.3 Continuous Learning

Now

$$W_k^{(m)} = (I - 2\eta_k^{(m)}) W_k^{(m-1)} + 2\eta_k^{(m)} X_n \tag{11}$$

where $X_n \in P_k^{(m)}$. We renumber the $X_n \in P_k^{(m)}$ to X_{ki}, $i \in \{1, \ldots, \| P_k^{(m)} \|\}$. Since the weight is updated once for each input vector, $\| P_k^{(m)} \|$ is also equal to the updating times, i.e. $\| P_k^{(m)} \| = m$. Eqn(11) becomes

$$W_k^{(m)} = \prod_{i=1}^{m} (I - 2\eta_k^{(i)}) W_k^{(0)} + 2 \sum_{i=1}^{m} \prod_{j=i+1}^{m} (I - 2\eta_k^{(j)}) \eta_k^{(i)} X_{ki} \tag{12}$$

If we let

$$\eta_k^{(i)} = \frac{I}{2 \| P_k^{(i)} \|} \tag{13}$$

then

$$W_k^{(m)} = \frac{1}{m} \sum_{i=1}^{m} X_{ki} \tag{14}$$

Since

$$D_k^{(m)} = \sum_{i=1}^{m} (X_{ki} - W_k^{(m)})^T (X_{ki} - W_k^{(m)}) \tag{15}$$

hence, in the sense of the optimality defined by eqn(4), the optimal $W_k^{(m)}$ is

$$W'^{(m)}_k = \arg \min_{W_k \in \mathcal{W}} \{D_k^{(m)}\} = \frac{1}{m} \sum_{i=1}^{m} X_{ki} \tag{16}$$

By comparing eqn(14) and eqn(16), we obtain the optimal $\eta_k^{(i)}$ as

$$\eta'^{(i)}_k = \frac{I}{2 \| P_k^{(i)} \|} \tag{17}$$

and

$$D'^{(i)}_k = E_k^{(i)} - \| P_k^{(i)} \| \| W'^{(i)}_k \|^2 \tag{18}$$

where $E_k^{(i)}$ is the energy of the input vectors encoded into the neuron k.

2.4 THE CVQ FOR LINEAR PREDICTIVE DATA

In analysing the classification performance of a single layer connectionist model with linear predictive data, Fallside [3] suggested a form of CVQ structure for quantising linear predictive data. In this CVQ, a set of single layer networks are trained to span an appropriate region of the linear predictive coefficient space. The codevector of the current input data is then specified by the index of the network with the least output cost function. This method directly processes speech samples in the time domain. The cost function of the network of the type studied in [3] is of the form

$$d(Y_n, W_k) = (Y_n + Z_n W_k)^T (Y_n + Z_n W_k) \tag{19}$$

The definitions of Y_n, Z_n and W_n are same as in [3]. Since Y_n is linear predictive, $R_n \alpha_n = -Z_n^T Y_n$, where $R_n = Z_n^T Z_n$ and α_n is the linear predictive coefficient vector, so we obtain,

$$d(Y_n, W_k) = (W_k - \alpha_n)^T R (W_k - \alpha_n) \tag{20}$$

Therefore, the cost function of the network in the CVQ for linear predictive data is of the Itakura type.

As in the case of the Euclidean measure, we analyse the CVQ for linear predictive data in both batch learning and continuous learning modes. For batch learning, there is

$$W_k^{(m)} = [I - 2\eta_k^{(m)} \sum_{Y_n \in P_k^{(m)}} R_n] W_k^{(m-1)} + 2\eta_k^{(m)} \sum_{Y_n \in P_k^{(m)}} R_n \alpha_n \tag{21}$$

We obtain the optimal $\eta_k^{(m)}$ and $W_k^{(m)}$ as

$$\eta'^{(m)}_k = [2 \sum_{Y_n \in P_k^{(m)}} R_n]^{-1} \tag{22}$$

$$W'^{(m)}_k = [\sum_{Y_n \in P_k^{(m)}} R_n]^{-1} \sum_{Y_n \in P_k^{(m)}} R_n \alpha_n \tag{23}$$

From [4] we know that eqn(23) is exactly the centroid of the cell k of the VQ designed using the LBG algorithm with respect to the Itakura distortion measure.

For continuous learning, analogously to eqn(12), we obtain

$$W_k^{(m)} = \prod_{i=1}^{m}(I - 2\eta_k^{(i)}R_{ki})W_k^{(0)} + 2\sum_{i=1}^{m}\prod_{j=i+1}^{m}(I - 2\eta_k^{(j)}R_{ki})\eta_k^{(i)}R_{ki}\alpha_{ki} \quad (24)$$

and the optimal $\eta_k^{(i)}$ and $W_k^{(m)}$ are

$$\eta'^{(i)}_k = [2\sum_{l=1}^{\|P_k^{(i)}\|}R_{kl}]^{-1} \quad (25)$$

$$W'^{(m)}_k = [\sum_{l=1}^{m}R_{kl}]^{-1}\sum_{l=1}^{m}R_{kl}\alpha_{kl} \quad (26)$$

3 SIMULATIONS

A Gauss-Markov or first order Gauss autoregressive source is commonly used in the simulation of coding systems. This source is defined by the difference equation: $X_{n+1} = \alpha X_n + G_n$, where $\{G_n\}$ is zero mean, unit variance, i.i.d Gaussian samples and α is a correlation coefficient. Here, we let $\alpha = 0.9$. The theoretically achievable SNR given by Shannon's distortion rate function of this source with respect to the MSE measure and with 1 bps transmission rate is 13.2 dB [1]. In our experiments, 120000 samples were generated and divided into two equally sized groups, a training set and a test set. Each initial codeword of the VQ with the LBG algorithm, or initial weight in the CVQ is initialized with any one of input vectors. The design is completed when $(D^{(m)} - D^{(m-1)})/D^{(m)} \leq 0.001$. The transmission rates of all quantisers are the same and equal to 1 bit/sample.

Figure 1 compares the learning curve for a CVQ with batch learning, the learning curve for a CVQ with continuous learning, both with the proposed optimal gain sequences, and the learning curve for a CVQ designed using the original Kohonen self-organizing algorithm with its convention gain sequences. Since it was reported that there was no obvious difference among linear, exponential and centrally adaptive gain sequence forms [5], here we used a linear form in Kohonen's algorithm.

Table 1 compares VQs designed using the LBG algorithm, CVQs with batch learning and CVQs with continuous learning in their rate-distortion performances and the number of iterations taken to finish the design. From the above comparisons, we note that from the point of view of rate-distortion performance, the CVQ does not perform better than the VQ using the LBG design algorithm. Only when the gain sequence is defined as eqn(8) for batch learning and as eqn(17) for continuous learning, and the weights of the neurons are assumed to be independent of each other, will the rate-distortion performance of the CVQ approach that of the VQ using the LBG algorithm, and the number of iterations be not larger than that for the VQ with the LBG algorithm. Batch learning sweeps through all the inputs and accumulates $\nabla_W D$ before changing weights, so it is guaranteed to move in the direction of the steepest descent. In our results, the CVQ with batch learning is identical to the VQ with the LBG design algorithm and slightly better than the CVQ with continuous learning, but the latter converges more quickly and needs fewer memory locations. Table 1 also shows the generalization property of CVQs by giving their quantising performance for unseen data. The SNR performance for the test set was very near to that for the training set. However, the difference increased quickly with the dimension of input vectors.

p	*VQ-LBG*			*CVQ-BL*			*CVQ-CL*		
	Training		Test	Training		Test	Training		Test
	SNR	I	SNR	SNR	I	SNR	SNR	I	SNR
1	4.4	4	4.4	4.4	4	4.4	4.4	2	4.4
2	8.0	13	7.9	8.0	13	7.9	8.0	4	7.9
3	9.4	27	9.3	9.4	27	9.3	9.2	15	9.1
4	10.3	33	10.1	10.3	33	10.1	9.8	15	9.7
5	10.7	28	10.4	10.7	28	10.4	10.3	20	10.2
6	11.1	27	10.8	11.1	27	10.8	10.9	12	10.7
7	11.4	17	11.0	11.4	17	11.0	11.3	10	10.9
8	11.9	14	11.1	11.9	14	11.1	11.8	11	11.1

Table 1: Comparsions among VQs designed using the LBG algorithm (VQ-LBG), CVQs with batch learning (CVQ-BL) and CVQs with continuous learning (CVQ-CL), where 'p' is the dimension of input vectors, 'I' is the number of iterations and SNR is in *dB*.

To evaluate CVQs for linear predictive data, we applied them to a multipulse linear predictive speech coder (MPLPC) to quantise the linear predictive coefficients. All speech samples are from the TIMIT database and are pre-filtered to 8kHz. The training set consists of 16 different sentences from 16 speakers (8 females). The test set consists of another 16 different sentences from another 16 speakers (8 females). We set the order of the linear predictive filter to 12, the frame width to 128 samples, its step size to 100 samples, and the number of pulses per frame in the MPLPC to 12. A segmental SNR (SEGSNR) is used to average the frame performance measure over frames. Table 2 gives the rate-distortion performances of the MPLPC with CVQs for linear predictive coefficients. Figure 2 compares several segments of waveforms, in which the original speech is a part of the transition section of a syllable /sao/ in a female spoken sentence which is outside the training set.

4 CONCLUSIONS

Because of their adaptive and continuous learning abilities, connectionist models have been widely studied for VQs. We have demonstrated the optimality of CVQs and derived an optimal gain sequence for fastest distortion minimization in

Rate (bits/ frame)	Segmental SNR(dB) of MPLPC					
	VQ-LBG		CVQ-BL		CVQ-CL	
	Tra.	Test	Tra.	Test	Tra.	Test
1	5.8	6.0	5.6	5.8	5.6	5.9
2	6.4	6.6	6.3	6.5	6.3	6.6
3	6.7	6.9	6.7	7.0	6.6	6.9
4	7.1	7.2	7.1	7.3	6.9	7.0
5	7.4	7.5	7.2	7.5	7.3	7.6
6	7.7	7.7	7.7	7.8	7.7	7.9

Table 2: Comparison of the rate-distortion performances among the MPLPC respectively with the VQ-LBG, the CVQ-BL and the CVQ-CL for both the training and test speech set. The performances without linear quantising are $9.8dB$ for the training set and $10.5dB$ for the test set.

CVQ design. Its efficacy is demonstrated by a number of cases. Using optimal gain sequences one can obtain the same rate-distortion performance as that of the VQ designed using the LBG algorithm, and the number of iterations taken to design the CVQ will not be larger than that for the VQ with the LBG algorithm.

Acknowledgements

We thank our colleagues in the speech group of CUED, especially, Dr. M. Plumbley, H. Lucke and P. Steen for their valuable comments and productive discussions. One of authors, L. Wu, is supported by the Cambridge Overseas Trust.

REFERENCES

[1] T. Berger. *Rate distortion theory - a mathematical basis for data compression.* Prentice-Hall Inc. Englewood Cliffs, New Jersey, 1971.

[2] P. Brauer and P. Knagenhjelm. Infrastructure in Kohonen maps. In *Proc ICASSP*, pages 647–650, 1989.

[3] F. Fallside. On the analysis of multi-dimentional linear predictive/autoregressive data by a class of single layer connectionist models. In *Proc. IEE Conf. on ANN*, pages 176 – 180, 1989.

[4] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, COM-28(1):84–95, Jan. 1980.

[5] A. Lucas and J. Kittler. A comparative study of the Kohonen and multiedit neural net learning algorithms. In *Proc IEE Conf. on ANN*, pages 7–11, 1989.

[6] T.Kohonen. *Self-organization and associative memory.* Springer Verlag, New York, Third edition, 1988.

[7] F. Wu and K. Ganesan. Comparative study of algorithms for VQ design using conventional and neural-net based approaches. In *Proc ICASSP*, pages 751–754, 1989.

[8] L. Wu and F. Fallside. On the design of connectionist vector quantizers. to be published, Jul. 1990.

[9] E. Yair, K. Zeger, and A. Gersho. Conjugate gradient methods for designing vector quantizers. In *Proc. ICASSP*, pages 245–248, 1990.

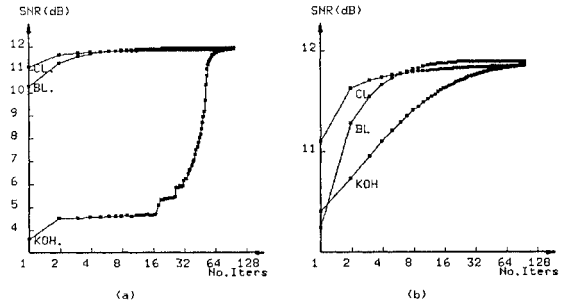

Figure 1: Learning curves for a CVQ with batch learning (BL), for a CVQ with continuous learning (CL), both with the optimal gain sequences and for a CVQ with the original Kohonen self-organizing algorithm (KOH) with its conventional gain sequences. In Figure 1-a, the learning process for the 'KOH' is divided into two phases as shown in [2]. Their gain sequences and the neighbourhoods are defined as below. In phase 1: $\eta(t) = C_1(1 - \frac{t}{T_1})$; $N_c(t) = (N_0 - 1)(1 - \frac{t}{T_1}) + 1$. In phase 2: $\eta(t) = C_2(1 - \frac{t-T_1}{T_2})$; $N_c(t) = 1$. where T_1, T_2 are the number of iterations in phase 1 and phase 2, and C_1 and C_2 are constants. Here $T_1 = 50$, $T_2 = 49$, $C_1 = 0.1$ and $C_2 = 0.01$. N_0 was the initial neighbourhood which covered half of the neurons at the begining. In Figure 1-b, the weight updating was restricted to a winning neuron. The learning process consisted of only the second phase, i.e. $T_1 = 0$, $T_2 = 99$, $C_1 = 0$ and $C_2 = 0.01$.

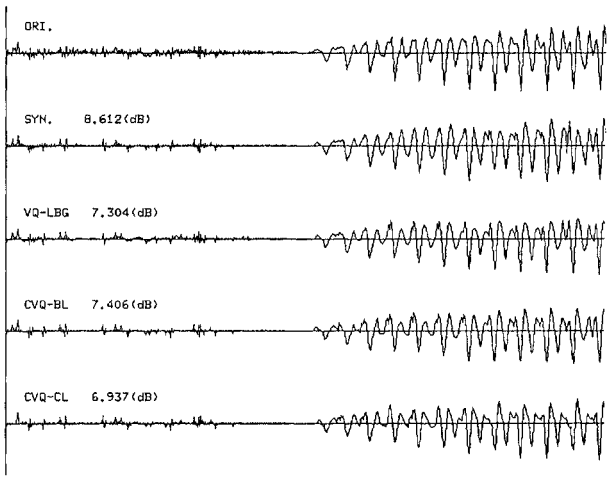

Figure 2: Comparison among the original speech (ORI.), the reconstructed speech waveforms respectively from the MPLPC without quantising (SYN.), the MPLPC with a VQ designed using the LBG algorithm (VQ-LBG), the MPLPC with a batch learning CVQ (CVQ-BL), and the MPLPC with a continuous learning CVQ (CVQ-CL). The transmission rates of all quantisers for linear predictive coefficients are the same and equal to 5 bits/frame. The segmental SNRs of the waveforms are also listed at the lefthand side.