

Lemma 2 requires to impose the constraint  $v_i < v_{\max}$  and, therefore, the convergence theorem holds for weights connecting the hidden to the output units that are constrained in  $[v_{\min}, v_{\max}]$ . Notice that, because of the use of threshold LMS cost functions, in no way is the constraint  $v_i < v_{\max}$  restrictive.

The constraints  $v_i \in [v_{\min}, v_{\max}]$  can be implemented by fixing the value of  $v_i$  to  $v_{\min}$  or  $v_{\max}$  if the updating rule causes a "left" or "right" violation of this constraint, respectively.

*Remark 3—Threshold LMS Functions:* The use of threshold LMS functions turns out to be useful for proving the convergence of on-line backpropagation. Basically, it allows us to deal with weight solutions where  $v_i \in [v_{\min}, v_{\max}]$ . In the case of symmetrical squashing functions, to guarantee an optimal solution with null cost, the absolute value  $d(d \doteq |d^+| = |d^-|)$  of the targets must follow the condition  $d < f((nd + 1)v_{\max})$ .

*Remark 4—On the Discovered Weight Solution:* Notice that no each hidden unit's weight vector needs to be a solution since, because of the used stopping criterion, the algorithm terminates as long as  $|y_o(W(k), V(k), k_{\text{mod } Q})| < \epsilon$  [21]. Hence, the learning behavior in the hidden units is different since it depends on the value of weights  $V$  of the last layer.

#### IV. CONCLUSIONS

In this paper, we have proven the companion of Rosenblatt's PC theorem for feedforward networks stating that on-line backpropagation is guaranteed to converge to an optimal solution in the case of linearly separable patterns.

To some extent, this result is an attempt to address some of the theoretical questions on optimal convergence raised by Minsky and Theoret in their intriguing epilogue on connectionist models [1].

A somewhat surprising result is that the optimal convergence is guaranteed with no upper bound on the learning rate.

This suggests that on-line algorithms should not be considered necessarily as an approximation of batch mode schemes, and that, at least in the case assumed in this paper, the optimal convergence of on-line algorithms can also be given a clear theoretical foundation that need not rely on the shape of the cost function.

#### ACKNOWLEDGMENT

This paper is dedicated to the memory of Prof. E. R. Caianiello, who stimulated one of the authors to find the connections from Rosenblatt's PC theorem and pattern mode backpropagation during an invited lecture given at IIASS (International Institute for Advanced Scientific Studies, Salerno, Italy) in March 1992. The authors also thank Dr. M. Bianchini for her very useful comments and suggestions on an earlier draft of this paper and the anonymous reviewers of the paper for their constructive criticisms which improved significantly the quality of the presentation.

#### REFERENCES

- [1] M. Minsky and S. Papert, *Perceptrons—Expanded Edition*. Cambridge, MA: MIT Press, 1988.
- [2] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53–58, 1989.
- [3] E. D. Sontag and H. J. Sussman, "Backpropagation separates when perceptrons do," in *Proc. Int. Joint Conf. Neural Networks*, vol. 1, Washington D.C., June 1989, pp. 639–642.

- [4] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-14, pp. 76–86, Jan. 1992.
- [5] X. H. Yu, "Can backpropagation error surface not have local minima?" *IEEE Trans. Neural Networks*, vol. 3, pp. 1019–1020, Nov. 1992.
- [6] M. Bianchini, P. Frasconi, and M. Gori, "Learning without local minima in radial basis function networks," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 749–756, May 1995.
- [7] M. Bianchini, M. Gori, and M. Maggini, "On the problem of local minima in recurrent neural networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 167–177, Mar. 1994.
- [8] D. Rumelhart, J. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA: MIT Press, 1986.
- [9] F. Rosenblatt, "On the optimal convergence of reinforcement procedures in simple perceptrons," Cornell Aeronautical Lab., Buffalo, NY, Tech. Rep. VG-1196-G-4, Feb. 1960.
- [10] S. Gallant, "Optimal linear discriminants," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, Oct. 28–31, pp. 849–852, 1986.
- [11] S. D. Wang and C. H. Hsu, "Terminal attractor learning algorithms for backpropagation neural networks," in *Proc. Int. Joint Conf. Neural Networks*, Singapore, pp. 183–189, Nov. 1991.
- [12] A. G. Parlos, B. Fernandes, A. F. Atiya, J. Muthusami, and W. K. Tsai, "An accelerated learning algorithm for multilayer perceptron networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 493–497, May 1994.
- [13] Y. le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a backpropagation network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. San Mateo, CA: Morgan Kaufmann, 1990, pp. 396–404.
- [14] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.
- [15] F. Jordan and G. Clement, "Using the symmetries of multilayered networks to reduce the weight space," in *Proc. Int. Joint Conf. Neural Networks*, vol. 2, Seattle, WA, July 8–12, 1991, pp. 391–396.
- [16] A. M. Chen, H. Lu, and R. Hecht-Nielsen, "On the geometry of feedforward neural network error surfaces," *Neural Computa.*, vol. 5, no. 6, pp. 910–927, 1993.

#### Comments on "On a Novel Unsupervised Competitive Learning Algorithm for Scalar Quantization"

Lachlan L. H. Andrew

**Abstract**—This note proposes an efficient alternative to a recently proposed neural network for designing scalar quantizers. It also points out that the performance measure used is of limited applicability.

#### I. INTRODUCTION

A recent letter [1] presented a novel neural-network learning rule, BAR, (boundary adaptation rule) which was shown to converge to a scalar quantizer with equiprobable outputs. Such quantizers will be called maximum entropy quantizers (MEQ's). It is interesting that such a simple rule can produce these quantizers. Its practical usefulness is limited, however, by two factors. First, there are more efficient algorithms which yield better results, and second MEQ's are unsuitable for many quantization tasks, as discussed below.

Manuscript received February 2, 1995; revised April 28, 1995. This work was supported by a scholarship from the Australian Telecommunications and Electronics Research Board (ATERB).

The author is with the Department of Electrical and Electronic Engineering at the University of Melbourne, Parkville 3052, Victoria, Australia.

Publisher Item Identifier S 1045-9227(96)00174-9.

## II. DESIGN ALGORITHM FOR SCALAR MEQ'S

A  $k$ -level MEQ for a training set of  $N$  samples is any quantizer whose  $i$ th threshold,  $\theta_i$ , satisfies  $x(\lfloor iN/k \rfloor) \leq \theta_i \leq x(\lfloor iN/k \rfloor + 1)$ , where  $x(n)$  denotes the  $n$ th smallest sample in the training set, and  $\lfloor \cdot \rfloor$  denotes the integer part. The required  $x(n)$  can be obtained with about  $(2 + 2 \ln 2)N \log_2 k$  comparisons and no multiplications using an algorithm employing the quantile selection procedure based on Quicksort [2, ch. 9]. When  $k$  is a power of two, the algorithm is

- 1) Locate the median, and hence partition the set into halves and determine  $\theta_{k/2}$ .
- 2) Repeat for each half of the training set until all  $\theta_i$  are known.

Each of the  $\log_2 k$  levels of recursion requires  $(2 + 2 \ln 2)N + o(N)$  comparisons on average, yielding the stated result.

Once the above algorithm terminates, in a small finite time, it yields an exact MEQ for the training data. In comparison, although the BAR algorithm converges to an exact MEQ for the input distribution in infinite time, at any finite time it will not generally yield the best MEQ for the data seen so far. Let  $N$  be the minimum number of samples required to specify  $\theta_i$  to a given accuracy with a given confidence. If the BAR uses a binary search to determine  $\text{Act}_{D_i}(x)$ , then each step requires  $O(\log k)$  time, so both algorithms require  $O(N \log k)$  time. The BAR, however, requires many more than  $N$  samples for this accuracy. In addition, since the  $\theta_i$  above are simply order statistics of the data set, the size of the set required for a given accuracy can be calculated, which is less straightforward for the BAR.

As with the BAR, this algorithm is fundamentally limited to the scalar case by selecting thresholds rather than reconstruction levels. Since this algorithm designs a static quantizer, it is only suitable for stationary inputs, which is also a condition assumed in the convergence proof of the BAR. While better algorithms no doubt exist, this produces superior results faster than the BAR.

## III. APPLICABILITY OF MEQ'S

Quantization aims to find the "best" discrete representation of a continuous signal. For pattern recognition, "best" is that which retains most semantic information. For lossy coding, "best" is that which minimizes the reconstruction distortion. Although intuitively appealing, MEQ's do not arise as the "best" quantizers for a particular problem. (In fact, for coding it is desirable to minimize the entropy for a given distortion to facilitate entropy coding.) Vector MEQ's have been proposed [3] to avoid vector under-use in minimum distortion quantizers. (Enforcing equidistortion directly is better [4].) In the vector case, this method of minimizing distortion is justified by the theorem that in the limit of high dimension the minimum distortion coder is also an MEQ since the distribution becomes uniform within its support [5]. Thus designing a scalar quantizer on this principle is questionable. Indeed, a multilevel MEQ with nearest-neighbor coding for a highly peaked distribution can degenerate to producing a single reconstruction level for all inputs, which is clearly undesirable. (For example, a three-level MEQ for

$$p(x) = \left(\frac{1}{6\alpha} - \frac{1}{3}\right) \text{rect}\left(\frac{x}{2\alpha}\right) + \left(\frac{1}{3}\right) \text{rect}\left(\frac{x}{2 + 2\alpha}\right)$$

as  $\alpha \rightarrow 0$ , where  $\text{rect}(x) = 1$  if  $|x| < 1/2$ , 0 otherwise, maps any input to zero.)

Also, although in high dimensions a minimum distortion quantizer is an MEQ, the converse, used in [3], need not hold. Consider a quantizer which is an MEQ of the first component and always maps

the other components to their mean values. This will be an MEQ for the vector source, but is generally far from optimal.

*Author's Reply*—Marc M. Van Hulle

The boundary adaptation rule (BAR) is an unsupervised competitive learning rule for scalar quantization which maximizes information-theoretic entropy and, thus, which yields an equiprobable quantization of univariate probability distributions [1], [6]. In other words, BAR is an algorithm for generating maximum entropy quantizers (MEQ's) without using *a priori* knowledge of the probability distribution.

In essence, BAR is a numerical technique for integrating the following system of ordinary differential equations:

$$\frac{d\theta_j}{dt} = \eta(p(D_{j+1}) - p(D_j)), \quad j = 1, \dots, k-1 \quad (1)$$

with  $\eta$  the learning rate and  $D_j$  and  $D_{j+1}$  two consecutive and nonoverlapping quantization intervals separated by the boundary point  $\theta_j$ . Hence, (1) represents a learning rule of a  $k$ -point scalar quantizer in which the boundary points are updated, rather than the interval's midpoints (centroids). A faster way to integrate this equation is by using the fast BAR rule (FBAR) [6] so that the original BAR rule is of theoretical interest only. Recently, the BAR concept has been generalized toward  $r$ th power law distortion minimization [7], for the high resolution case ( $k$  large).

## IV. BAR AS A DESIGN ALGORITHM FOR SCALAR MEQ'S

It is evident that, given a training set of  $N$  samples, the fastest way to arrive at a maximum entropy quantizer (MEQ) is to apply an algorithm employing quantile selection based on a sorting algorithm such as Quicksort. Procedures like this are in fact batch algorithms. The major drawback of batch algorithms is that the design of the quantizer only begins after the entire training set is available. Hence, notwithstanding that the Quicksort-based algorithm is very fast, it requires the memorization of all training samples. In addition, such a procedure is unable to accommodate "on-line" changes in the probability distribution. Finally, it cannot be extended to the multidimensional case.

The boundary adaptation rule (BAR) is a neurally inspired learning algorithm that operates in a completely different way. In BAR, the quantizer is built "on the fly;" after the presentation of each input sample, the boundary points are updated with small increments. In this way, BAR is able to accommodate "on-line" changes in the input probability distribution and to avoid the memory problem, but at the expense of a reduced speed of convergence.

This trading of memory for speed in the Quicksort-based algorithm brings to mind another issue. In case the samples are drawn from a continuous distribution, we will face the formidable problem of memorizing a massive number of training samples to achieve a given accuracy. Hence, quantile selection will have to be replaced by histogram collection. In that case, however, the *a priori* choice of the location of the histogram's quantization regions poses a chicken-and-egg problem.

Manuscript received May 12, 1995.

The author is with the Laboratorium voor Neuro- en Psychofysiologie at the K. U. Leuven, Campus Gasthuisberg, B-3000 Leuven, Belgium.

Publisher Item Identifier S 1045-9227(96)00173-7.

The most widely used design algorithm for scalar quantizers, the Lloyd I algorithm [8], is a batch algorithm in which weights are adjusted incrementally. The Lloyd I algorithm has been generalized also for designing mean absolute error (MAE) quantizers [9]. Since minimum MAE quantization approximates maximum entropy quantization, we have compared the speed of BAR, or better of fast BAR (FBAR), with generalized Lloyd I [6]. We have found that FBAR is at least an order of magnitude faster for a any reasonable size of  $k$ . Hence, batch algorithms performing incremental weight adjustments are inherently slower than those performing sorting, but the former are not necessarily faster than neurally-inspired learning algorithms.

In conclusion, a comparison between a Quicksort-based algorithm and an algorithm performing incremental weight updates is not entirely fair if it is based on speed considerations only; the two algorithms have different memory/speed trade-off's and serve different purposes. This should be taken into account in any comparison.

#### V. CHOICE OF ENTROPY MAXIMIZATION AS A DESIGN PRINCIPLE

The design of any quantizer is the subject of a compromise, i.e., the solution of an optimization problem with a given objective function. For example, optimal quantizers maximize entropy or minimize a distortion metric defined as a function of the difference between the actual and the quantized input. Many distortion metrics have been proposed in literature [10], [11] but the most commonly used one is the mean squared error (MSE) metric due to its mathematical simplicity. By maximizing entropy, one attempts to achieve an optimal resource usage: the quantization intervals are used equally frequently (equiprobable quantization). On the other hand, by minimizing a given distortion metric, one attempts to achieve an optimal signal representation.

There has been a lot of confusion in the neural network literature about equiprobable quantization. Several neural network researchers have started from the standard unsupervised competitive learning rule (UCL), which minimizes the MSE distortion [12], and introduced various mechanisms inspired by the "conscience" mechanism [13], [14] as a way to avoid that neurons would become underused. In essence, these mechanisms are aimed at achieving an equiprobable quantization. MSE minimization, however, is not the same as entropy maximization [6], [15] so that these modified UCL rules cannot yield optimal quantizers.

In conclusion, it is not the optimization principle chosen for designing scalar quantizers which is questionable, but the choice of an algorithm which fails to yield an optimal quantizer.

#### REFERENCES

- [1] M. M. Van Hulle and D. Martinez, "On a novel unsupervised competitive learning algorithm for scalar quantization," *IEEE Trans. Neural Networks*, vol. 5, pp. 498–501, May 1994.
- [2] R. Sedgewick, *Algorithms in C*. Redwood City, CA: Addison-Wesley, 1990.
- [3] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, pp. 277–290, 1990.
- [4] N. Ueda and R. Nakano, "A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers," *Neural Networks*, vol. 7, no. 8, pp. 1211–1227, 1994.
- [5] C. E. Shannon, "Coding theorem for a discrete source with a fidelity criterion," in *Information and Decision Processes*, R. E. Machol, Ed. New York: McGraw-Hill, 1960, pp. 93–126.
- [6] M. M. Van Hulle and D. Martinez, "On an unsupervised learning rule for scalar quantization following the maximum entropy principle," *Neural Computa.*, vol. 5, pp. 939–953, 1993.
- [7] D. Martinez and M. M. Van Hulle, "Generalized boundary adaptation rule for minimizing  $r$ th power law distortion in the high resolution case," *Neural Networks*, vol. 8, no. 6, pp. 891–900, 1995.
- [8] S. P. Lloyd, "Least-squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 127–135, 1982.
- [9] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Dordrecht, Netherlands: Kluwer, 1991.
- [10] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.
- [11] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.
- [12] H. Ritter, "Asymptotic level density for a class of vector quantization processes," *IEEE Trans. Neural Networks*, vol. 2, no. 1, pp. 173–175, 1991.
- [13] S. Grossberg, "Adaptive pattern classification and universal recoding," *Biol. Cybern.*, vol. 23, pp. 121–134, 1976.
- [14] D. DeSieno, "Adding a conscience to competitive learning," in *Proc. 1988 Int. Conf. Neural Networks*, vol. I, San Diego, pp. 117–124.
- [15] N. Ueda and R. Nakano, "A new learning approach based on equidistortion principle for optimal vector quantizer design," in *Proc. IEEE-NNSP Wkshp.*, Linthicum Heights, MD, 1993, pp. 362–371.

#### Corrections to "On the Local Minima Free Condition of the Backpropagation Learning"

X.-H. Yu and G.-A. Chen

We regret that in [1], the author's corrections were not included in the final version. The proof of Lemma 3 should read as follows.

*Proof:* Let  $u_k = [u_k(1), \dots, u_k(N_h)]$ ,  $1 \leq k \leq P$ , be the rows of  $\bar{X}W_1^o$ . We simply assume that there is at least one column (say, the  $i$ th column for notational convenience) of  $\bar{X}W_1^o$  in which identically valued elements do not exist, since for the case without identical inputs, a column of  $\bar{X}W_1^o$  having identical elements only occurs on a thin (Lebesgue measure zero) set<sup>3</sup> in the weight space formed by  $W_1$ . Otherwise we can simply add a disturbance  $\delta W_1^o$  to  $W_1^o$  such that this assumption is met. To characterize the singularity of the matrix  $\bar{H}$ , we modify the input-to-hidden weight matrix as  $W_1(\lambda) = [W_1^o(1), \dots, \lambda W_1^o(i), \dots, W_1^o(N_h)]$ . Noting that it is assumed that for  $W_1^o$  the matrix  $\bar{H}^o$  is singular, we have

$$\det \bar{H}(\lambda) = 0, \quad \text{for } \lambda = 1 \quad (9)$$

Manuscript received October 15, 1995.

The authors are with the National Communications Research Laboratory, Department of Radio Engineering, Southeast University, Nanjing 210018, P. R. China.

Publisher Item Identifier S 1045-9227(96)01471-3.

<sup>3</sup>In fact, for noncoincident inputs, if  $u_k(i) = u_j(i)$  holds for  $k \neq j$ ,  $W_1$  should take values on a hyperplane given by  $(X_k - X_j)W_1(i) = 0$ , where  $W_1(i)$  is the vector formed by all weights from input to the  $i$ th sigmoidal hidden neuron.