Implementation of Neural Gas training in analog VLSI

Fabio Ancona, Stefano Rovetta, and Rodolfo Zunino

Department of Biophysical and Electronic Engineering
University of Genova
Via all'Opera Pia 11a, 16145 Genova (ITALY)
Fax: +39 10 353 2175 – E-mail Ancona,Rovetta,Zunino@dibe.unige.it

Abstract

The design and implementation of a vector quantization neural network is presented. The training algorithm is Neural Gas. The implementation is fully parallel and mainly analog (only control function and long-term memory are digital). A sequential implementation of the required sorting function allows to compute the Neural Gas updating step.

1 Introduction

Vector-quantization (VQ) neural networks are useful for many neural processing and generic signal-processing applications. For instance, image compression/processing is frequently approached through VQ. A neural approach allows concepts and algorithms developed in the fields of neural modeling and physics to be exploited in other research areas. This is the approach of the work presented here.

Martinetz *et al.* proposed the Neural Gas (NG) neural network in [1]. They adopted the standard vector-quantization scheme for recall, without modifications; however, they proposed an interesting training algorithm. The NG training procedure is a stochastic gradient descent on a cost function defined by the distortion criterion adopted, which is assumed to be Euclidean distance. To avoid local minima, the algorithm adopts a strategy similar to that of Kohonen's Self Organizing Maps (SOM) [2]: optimization begins with generalized updates (each neuron is somewhat moved at each training step) and end with very specific updates (only the appropriate neuron is moved). Accordingly, local minima in the error function emerge slowly during training, so that they can be avoided. However, SOMs propagate the updating step from the winner to its neighbors exploiting topology relations inherent to the network. NG instead uses a rank-based distribution, that allows to overcome some drawbacks of the fixed topology.

Specifically, the training step for NG is summarized as follows. x_l is the l-th input vector, w_i the i-th reference vector, ϵ a constant (learning coefficient). NG first sorts all reference vectors with respect to their distance from x_l, then defines a function $k_i(x_l)$ that maps the i-th reference vector into its position in the ordered list (its rank). Finally, the rank k_i of each reference vector is used to compute an appropriate step size through the function $h(k_i)$. The updating step is therefore simply

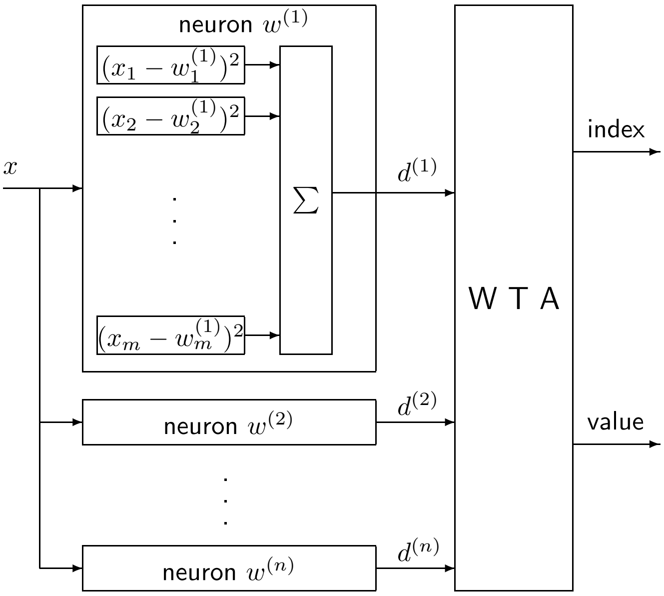
$$\Delta w_i^{\mathrm{NG}} = \epsilon \cdot h(k_i(x_l)) \cdot (x_l - w_i). \tag{1}$$

The result of this strategy is that very good minimum points can be found by the NG algorithm, as compared to SOM or other techniques such as k-means, and they are obtained in a low number of steps.

---

[*] This work was supported by MURST 40% fundings and by contributions to research training of graduate students from the University of Genova.

Figure 1: Functional diagram of the encoder.



Implementation of any VQ optimization algorithm can be costly in terms of time, since they all require computing all distances from the input vector. In addition, NG training is more computationally expensive due to the need for sorting. Therefore, for real-time VQ (both recall *and* training) a hardware implementation is advisable.

This paper presents such a realization. In the relevant literature, the main electronic implementations of VQ are in digital technology and tailored for signal-processing applications only (mainly image compression) [3][4]. Although it adopts the same reference application, the implementation of the proposed project is mainly analog, with some digital control subsystems. Moreover, the majority of systems are time-multiplexed, whereas the proposed project is fully parallel, with an $O(1)$ time complexity (operation is independent of both vector size and number of reference vectors). Some VQ implementations with similar features can be found [5][6]; however, they often make compromises such as reduced number of neurons, non-standard vector size, or external implementation of some function. This project features full support for recall mode and implements the functions required for training.

2 The analog VLSI realization

The basic encoder functions for VQ [7] are presented in Figure 1. It is possible to notice that the circuit features both the standard output of a VQ encoder (index of the winning reference vector), and an analog output, that is, the distance of the winning reference vector from the input vector. This is a key feature to implement NG learning, and also for connecting multiple chips in a modular fashion to build networks with the desired number of neurons.

All blocks shown in Figure 1 are implemented in the circuit, and most of them are in analog form. Each neuron stores its reference vector in a medium-term memory, a MOS capacitor. The long-term memory is kept in an external RAM, from which the analog memory is refreshed through an analog bus and D/A converters.

The analog memory is used to compute the distance from the input vector:

$$\sum_{j=1}^{D} (x_{l,j} - w_{i,j})^2, \tag{2}$$

where x_l and w_i are represented as voltage values. The square of difference is implemented with

Figure 2: Circuit for the square of a difference.

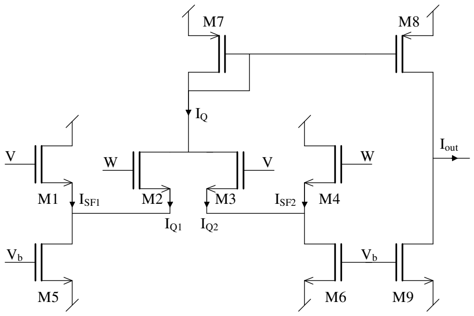
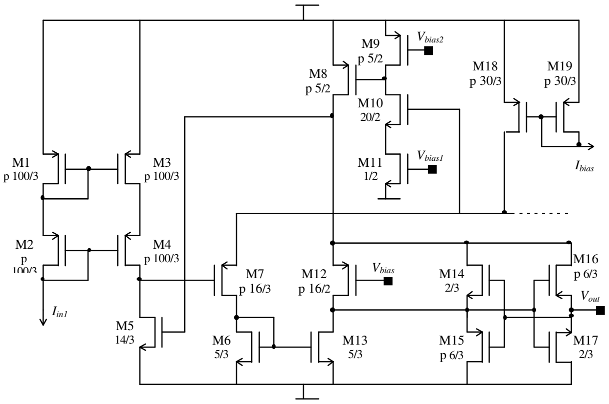


Figure 3: Competition and selection circuit.



the block represented in Figure 2. Its output is a current, proportional to the desired value. To obtain the sum, it is sufficient to feed the current outputs of all blocks into a single node.

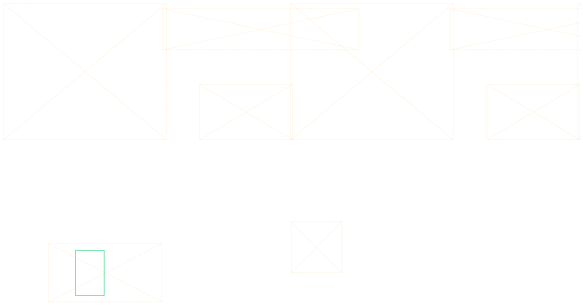
The subsequent block is the competition/selection subsystem, also termed “winner-take-all”. The original structure has been proposed by Lazzaro *et al.* [8]. In this system, the standard WTA circuit has been modified [9] (see Figure 3) to implement both the analog output (distance of the winner from the input) and the “minimum” instead of “maximum” selection function.

3 Implementation of the “Neural Gas” updating

The encoding system described in the above section supports the VQ encoding function. An external circuitry is used to implement the training step. The NG adaptation step Δw_i^{NG} is computed as a sum of decreasing terms. However, some of these terms give only a little contribution to it [10], and, given the stochastic nature of on-line training, can be neglected without effect (since they act essentially as an additive noise of small intensity). Therefore only a limited number of terms of the sum in Equation 1 is necessary. Experimental verifications have demonstrated that in the cases of interest only 7-8 terms are required.

The training procedure is implemented in mixed analog/digital technology, since the weights are stored in a digital RAM, but the distances $||x_l - w_i||^2$ are analog quantities. The sorted list of distances is computed sequentially, through the circuit shown in Figure 5. The circuit, presented in [11] and [12], is based on two iterated steps: searching the maximum, performed by the standard encoding circuit, and inhibiting the winner for subsequent operations. A simulated experiment is shown in Figure 6. At each training step, the first n distances are output in by

Figure 4: Layout of two competition cells.



the sort circuit, and their value is used to compute the value of Δw_i^{NG} according to Equation 1. The result is then converted into digital form (8 bits) and used to update the RAM long-term memory.

4 Remarks

The VQ chip has been designed and carefully simulated with HSPICE level 13 parameters for the ECPD10 1μm CMOS ES-2 technology. The experimental verifications have been performed on the extracted netlist, including all parasitics as modeled by the layout editor. The chip is currently being realized as a prototype for testing. The sorting function has been simulated, and a chip for its implementation and for the step of updating parameters is under study.

The authors wish to thank the graduate students A. Novaro, G. Oddone, and G. Uneddu and acknowledge their contribution to the development of this work.

References

[1] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten, "'Neural gas' network for vector quantization and its application to time–series prediction", *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[2] Teuvo Kohonen, *Self Organization and Associative Memories*, Springer, 3rd edition, 1989.

[3] Wai-Chi Fang, Chi-Yung Chang, Bing J. Sheu, Oscal T.-C. Chen, and John C. Curlander, "VLSI systolic binary tree–searched vector quantizer for image compression", *IEEE Transactions on VLSI Systems*, vol. 2, no. 1, pp. 33–44, March 1994.

[4] Heonchul Park and Viktor K. Prasanna, "Modular VLSI architectures for real–time full–search–based vector quantization", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 4, pp. 309–317, August 1993.

[5] Wai-Chi Fang, Bing J. Sheu, Oscal T.-C. Chen, and Joongho Choi, "A VLSI neural processor for image data compression using self–organization networks", *IEEE Transactions on Neural Networks*, vol. 3, no. 3, pp. 506–518, May 1992.

[6] Kevin Tsang and Belle W. Y. Wei, "A VLSI architecture for a real–time code book generator and encoder of a vector quantizer", *IEEE Transactions on VLSI Systems*, vol. 2, no. 3, pp. 360–364, September 1994.

[7] Fabio Ancona, Giorgio Oddone, Stefano Rovetta, Gianni Uneddu, and Rodolfo Zunino, "A vector quantization ciruit for trainable neural networks", in *Proceedings of the 1996 International Conference on Electronics, Circuits and Systems, Rhodos, Greece*, October 1996, pp. 1131–1134.

[8] J. Lazzaro, R. Ryckebush, M. A. Mahowald, and C. Mead, "Winner-take-all networks of $O(n)$ complexity", in *Advances in Neural Information Processing Systems II*, San Mateo, 1989, pp. 703–711, Morgan Kaufmann.

[9] Fabio Ancona, Giorgio Oddone, Stefano Rovetta, Gianni Uneddu, and Rodolfo Zunino, "Enhanced WTA network with linear output and stable transimpedance", *Alta Frequenza*, vol. 8, no. 5, pp. 71–73, September 1996.

[10] Fabio Ancona, Sandro Ridella, Stefano Rovetta, and Rodolfo Zunino, "On the role of sorting in "neural gas" for training vector quantizers", in *Proceedings of the 1997 International Conference on Neural Networks, Houston, USA*, June 1997, to be published.

[11] Giorgio Oddone, Stefano Rovetta, Giovanni Uneddu, and Rodolfo Zunino, "Mixed analog-digital circuit for linear-time programmable sorting", in *Proceedings of the 1997 International Conference on Circuits and Systems, Hong Kong, China*, June 1997, to be published.

[12] Fabio Ancona, Giorgio Oddone, Stefano Rovetta, Gianni Uneddu, and Rodolfo Zunino, "VLSI architectures for programmable sorting of analog quantities with multiple-chip support", in *Proceedings of the Seventh Great Lakes Symposium on VLSI, Urbana, Illinois, USA*, March 1997.