

Competitive stochastic neural networks for Vector Quantization of images

M. Graña, A. D'Anjou, A. I. Gonzalez, F.X. Albizuri, M. Cottrell*
Dept. CCIA Univ. Pais Vasco/EHU¹, Apto 649, 20080 San Sebastián, España
e-mail: ccprrom@si.ehu.es
* SAMOS Univ. Paris I

Abstract A stochastic approximation to the nearest neighbour (NN) classification rule is proposed. This approximation is called Local Stochastic Competition (LSC). Some convergence properties of LSC are discussed, and experimental results are presented. The approach shows a great potential for speeding up the codification process, with an affordable loss of codification quality.

keywords: Vector Quantization, Stochastic Neural Networks, Competitive Neural Networks, Radial Basis Function Networks

0 INTRODUCTION

Vector Quantization (VQ) is a technique that can be used to map analogue waveforms or discrete vector sources into a sequence of digital data for storage or transmission over a channel [8, 5, 11]. A vector quantizer is a mapping of input vectors into a finite collection of predetermined codevectors. The set of codevectors is called the *codebook*. In this paper we are not concerned with the search for good codebooks. Our concern is the acceleration of the codification process. We have found in the literature [17, 15, 3] some attempts to speedup the computation of the NN decision rule. These approaches used the fact that quite frequently the decision that a codevector is not the nearest neighbour of an input vector can be taken without fulfilling the computation of the Euclidean distance. They didn't involve any loss of accuracy and didn't propose any kind of computational distributed scheme. Our approach is a significant departure from that, because it is a stochastic approximation that involves loss of classification accuracy, though it has a potential for greater speedups given fully distributed implementations.

Following work started in [6,7], we propose here a Local Stochastic Competition (LSC) decision rule for VQ. The LSC rule is intended as a distributed stochastic approximation to the Nearest Neighbour (NN) rule usually applied in VQ to perform the mapping of the input vector into the codebook. The approach is related to Radial Basis Function (RBF) neural network architectures [9, 2, 14, 19, 18]. Radial Basis Function networks have been applied to function approximation or interpolation. The function domain is decomposed into a set of overlapping regions, each characterised by a kernel function whose parameters usually are the centroid and width of the region. The most usual kernel functions are Gaussian. From the Bayesian classification point of view [4], the use of Gaussian kernels can be interpreted as the approximation of the input distribution by a mixture of Gaussian distributions, each characterised by its mean and variance parameter. Each Gaussian distribution models the probability that a given input belongs to a class. Our approach assumes this kind of probabilistic framework. We assume that each codevector represents a separate class, being the mean of the Gaussian distribution. We assume that variance parameters can be estimated, either by the codebook design algorithm, or from the codebook itself. In the experiments reported here, the later has been assumed. Local Stochastic Competition, then, consists in the parallel sampling of the "a posteriori" probabilities of the codevector classes, taken as

¹This work is being supported by a research grant from the Dpto de Economía of the Excm. Diputación de Guipuzcoa

independent one-class problems. Note that, in the same framework, NN is the optimal Bayesian decision rule when the class variances are identical.

In section 1 we introduce Vector Quantization. In section 2 we describe the Local Stochastic Competition. In section 3 we give the results of the experiments upon a test image and some conclusions.

1 VECTOR QUANTIZATION

A grey level image is a matrix of pixels, each pixel taking values in the range $\{0..255\}$. A row-wise vector decomposition of dimension d of the image is a succession of vectors $\mathbf{X}=\{\mathbf{x}_1,\dots,\mathbf{x}_N\}$, each \mathbf{x}_n composed of d row-adjacent pixels. Column-wise and matrix-wise decompositions can be defined as well. A vector quantization of the image is a map from each \mathbf{x}_n into a natural number, that transforms the vector decomposition into a sequence of codes $\{c_1,\dots,c_N\}$. A Vector Quantization (VQ) is usually defined through a set of reference vectors $\mathbf{Y}=\{\mathbf{y}_1,\dots,\mathbf{y}_M\}$ called the codebook. M is the size of the codebook, and the number of different codes. The compression rate obtained through VQ, for grey level images (8 bits per pixel), is $8d/\log_2 M$. The vector quantization map is usually defined by the nearest neighbour rule:

$$C_{NN}(\mathbf{x}, \mathbf{Y}) = i \text{ s.t. } \|\mathbf{x} - \mathbf{y}_i\| = \min_{j=1..M} \{\|\mathbf{x} - \mathbf{y}_j\|\}$$

The decodification, that allows to recover the codified image, is, thus, defined as:

$$D(i, \mathbf{Y}) = \mathbf{y}_i$$

The problem of Vector Quantization design is the search for codebooks that minimise the distortion introduced by replacing the original \mathbf{x}_n vectors by their class representative \mathbf{y}_i . This task is intimately related to clustering problems, and, so, the first approach found in the literature [11] was the application of the well-known Isodata algorithm. Recent approaches include the application of neural network ideas, such as the Simple Competitive Learning (SCL) [1], the Self Organising Map (SOM) [10, 12, 13] or the Soft-Competition [16, 20]. The codebooks used in our experiments were produced using a threshold algorithm to determine the initial codevectors for the application of a Simple Competitive Learning.

The threshold algorithm starts by assigning the first sample vector as the first codevector: $\mathbf{y}_1=\mathbf{x}_1$. It then iterates through the vector decomposition of the image until it finds a vector \mathbf{x}_k such that its distance to each of the already found codevectors $\{\mathbf{y}_1,\dots,\mathbf{y}_c \ c<M\}$ is greater than a threshold value. This is a new codevector $\mathbf{y}_{c+1}=\mathbf{x}_k$. When the whole sample has been examined without achieving the completion of the codebook, the threshold is halved and the search restarted. In our experiments the threshold value is given by the formula: $d*\theta^2$.

The Simple Competitive Learning (SCL) algorithm is the simplest adaptive algorithm to compute the centroids of the clusters of a sample, and the most efficient, provided good initial conditions. It can be formally expressed as follows:

$$\Delta \mathbf{y}_i(n) = \begin{cases} \alpha_i(n)(\mathbf{x}_n - \mathbf{y}_i(n)) & i = C_{NN}(\mathbf{x}_k, \mathbf{Y}(n)) \\ 0 & j \neq i \end{cases}$$

where $\alpha_i(k)$ is the gain parameter that decreases to zero as the adaptation proceeds. In our case, we decrease this parameter making $\alpha_i=\alpha_i*0.9$ whenever $\Delta \mathbf{y}_i>0$.

2 LOCAL STOCHASTIC COMPETITION

As said in the introduction, our starting assumption is that each codevector represents a class of input vectors whose distribution is a Gaussian centred at the codevector. Each codevector samples independently a Bernoulli random variable of parameters $(p_n(i), 1 - p_n(i))$:

$$P[\mathbf{x}_n \in R_i] = p_n(i) = e^{-\|\mathbf{x}_n - \mathbf{y}_i\|^2 / \sigma_i^2} \quad i \in \{1..M\}$$

That is, each $p_n(i)$ is interpreted as the probability that \mathbf{x}_n belongs to the class represented by \mathbf{y}_i , taken as an independent one-class classification problem for each codevector. An algorithmic description of the sequential simulation of the LSC classification of an input vector \mathbf{x}_n , as done for our experiments, follows:

step 0- $k=0$

step 1- Built up the probability vector $\mathbf{p}=(p_n(i,k) \ i=1..N)$ computed as follows:

$$p_n(i,k) = e^{-\|\mathbf{x}_n - \mathbf{y}_i\|^2 / t_i(k)} \quad \text{with} \quad t_i(k) = f(k)\sigma_i^2$$

step 2- Sample the probabilities in \mathbf{p} : Built up the set

$$S_k = \{\mathbf{y}_i \in \mathbf{Y} \mid p_n(i,k) \geq u_i\}$$

where (u_1, \dots, u_N) are random numbers uniformly distributed in $[0,1)$

step 3- If $|S_k| = 0$ increase k by 1 and go to step 1

step 4- If $|S_k| > 0$ perform a random selection with equal probabilities in the set S_k . If codevector \mathbf{y}_i is chosen the codification is:

$$C_{LSC}(\mathbf{x}_n, \mathbf{Y}) = i$$

This process could be easily implemented in parallel, granting a separate processor for each codevector in the codebook. An algorithm of the process associated with each codevector could read as follows:

wait for input \mathbf{x}_n or reintent

In **case** of new input $k=1$

In **case** of reintent $k=k+1$

Compute $p_n(i,k) = e^{-\|\mathbf{x}_n - \mathbf{y}_i\|^2 / t_i(k)}$ with $t_i(k) = f(k)\sigma_i^2$

Generate a random number u . If $p_n(i) > u$ **signal** 1, if not **signal** 0.

Where reintents are asked for by a process that receives the codevectors signals and detects that no one of them has accepted the input vector as belonging to its class. The expected speedup of the codification process comes from the substitution of the sequential search through the codebook by the parallel test of the separate one-class probabilities. To have the guarantee that the process will converge, in the sense of giving any response, the function $f(k)$ must be monotonically increasing with k . The faster the increase, the shorter the response. Mathematically, LSC generates, for a given \mathbf{x}_n , a random sequence of sets $\{S_1, \dots, S_K\}$ with $|S_k| = 0$ for $k < K$, and $|S_K| > 0$. The stopping condition for this process is, then, to find a non-empty set. It is easy to verify that the probability of finding a non-empty set increases as the process goes on. The probability of finding an empty set at trial k is

$$P[|S_k| = 0 \mid \mathbf{x}_n, \mathbf{Y}] = \prod_{i=1}^N (1 - p_n(i,k)) = \prod_{i=1}^N \left(1 - e^{-\|\mathbf{x}_n - \mathbf{y}_i\|^2 / t_i(k)} \right)$$

Given that $f(k)$ is increasing: $\lim_{k \rightarrow \infty} e^{-\|\mathbf{x}_n - \mathbf{y}_i\|^2 / t_i(k)} = 0$, and so: $\lim_{k \rightarrow \infty} P[|S_k| = 0 \mid \mathbf{x}_n, \mathbf{Y}] = 0$, therefore:

$$\lim_{k \rightarrow \infty} P[|S_k| > 0 | \mathbf{x}_n, \mathbf{Y}] = 1$$

The increasing nature of $f(k)$ is of great relevance, both theoretical and practical. In our works we have chosen the exponential 2^{k-1} , because of the emphasis we put in speeding up the classification process. The fast increase of the variance term has the side effect of increasing the probability of bad classifications [7].

The last topic that remains to be discussed, before the presentation of our experimental results, is the estimation of the variance parameters. In the present work we have estimated these variance parameters from the codebook itself as follows. Let D_i denote the minimum distance from codevector \mathbf{y}_i to any other

codevector: $D_i = \min \left\{ \|\mathbf{y}_i - \mathbf{y}_j\|^2 \mid j = 1..M, j \neq i \right\}$ and compute the mean minimum distance between

codevectors $\bar{D} = \frac{1}{M} \sum_{i=1}^M D_i$. The estimate of the variance associated with codevector \mathbf{y}_i are, then, computed

as follows:

$$\hat{\sigma}_i^2 = \begin{cases} D_i & \text{if } D_i \leq \bar{D} \\ \frac{2d}{\bar{D}} & \text{if } D_i > \bar{D} \end{cases}$$

3 RESULTS AND CONCLUSIONS

We have performed a set of experiments of codification/decodification on the image in figure 1, applying the threshold algorithm to obtain the initial codebooks, and SCL to improve over them. SCL was applied only once to the vector decomposition of the image. The experiment parameters were the dimension of the vector decomposition d , and the threshold parameter θ . The quality measures computed are the distortion (δ) and signal-to-noise ratio (SNR). The expected speedup (s) of LSC over NN is computed as the number of codevectors divided by the mean number of trials that LSC performs. Tables 1 and 2 show the numerical results. Table 1 shows the results of NN and LSC codification with 256 codevectors of varying dimension (8, 16, 32, 64) obtained by application of SCL to the result of the threshold algorithm with threshold parameters $\theta=8$ and 32. Increasing d gives greater compression ratio. The variation of θ was intended to give different initial conditions for SCL.

θ	d	NN		LSC		
		SNR	δ	SNR	δ	s
8	8	27.0	466	23.4	1012	70
8	16	24.3	1328	20.5	3532	69
8	32	22.6	1878	19.1	5851	68
8	64	22.3	1655	19.3	8063	72
32	8	26.4	373	23.3	1217	79
32	16	23.4	822	19.0	4281	74
32	32	22.2	1196	18.0	8251	77
32	64	24.9	1057	20.6	4530	78

Table 1. M=256, results for NN and LSC coding for varying d and θ .

Table 2 shows the results of increasing number of codevectors. The codebooks are also obtained by application of the threshold and SCL algorithms. Figure 2 shows the decodification of the test image after NN codification using a codebook with 1024 codevectors of dimension 8. Figure 3 shows the decodification of the LSC codification with the same codebook. This precise codebook was chosen because it is the one that has the greater expected speedup.

M	NN		LSC		
	SNR	δ	SNR	δ	s
128	25.4	791	22.0	1547	34
256	27.0	466	23.4	1028	71
512	28.3	208	24.6	602	144
1024	30.8	50	26.6	234	304

Table 2. $d=8, q=8$. results for NN and LSC for increasing number of codevectors.

Table 1. Results for varying threshold parameter and vector decomposition dimension

From the data in both tables, it can be perceived an almost constant degradation of the LSC codification relative to NN. The observation of the images in figure 2 and 3 shows that this degradation can be acceptable, for applications without severe quality requirements. On the hand, table 2 shows how the expected speedup increases with the number of codevectors, which makes LSC a suitable alternative for applications with large codebooks.

REFERENCES

- [1] Ahalt S.C., A.K. Krishnamurthy, P. Chen, D.E. Melton (1990) "Competitive Learning Algorithms for Vector Quantization" *Neural Networks* 3 pp.277-290
- [2] Bishop C (1991) "Improving the generalization properties of radial basis function neural networks" *Neural Computation* 3 pp.579-588
- [3] Conway J.H., Sloane N.J.A. (1982) "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes" *IEEE trans. Inf. Theory* 28 pp.227-232
- [4] Duda R.O., P.E. Hart (1973) "Pattern classification and scene analysis" Wiley
- [5] Gersho A. (1982) "On the structure of vector quantizers" *IEEE Trans. Inf. Th.* 28(2) pp.157-166
- [6] Graña M., A. D'Anjou, F.X. Albizuri, F.J. Torrealdea, M.C. Hernandez (1993) "Local Stochastic Competition and Vector Quantification" in J. Mira, J. Cabestany, A. Prieto (eds) *New trends in Neural Computation*, LNCS 686, Springer Verlag pp.216-222
- [7] Graña M., A. D'Anjou, A.I. Gonzalez, F.X. Albizuri, M. Cottrell (1994) "Local Stochastic Competition for Vector Quantization of images" *WCNN'94 San Diego CA*
- [8] Gray R.M. (1984) "Vector Quantization" *IEEE ASSP* 1pp.4-29
- [9] Hartman EJ, JD Keeler, J. M Kowalski (1990) "Layered neural networks with gaussian hidden units as universal approximations" *Neural Computation* 2 pp.210-215
- [10] Kohonen T. (1984) (1988 2nd ed.) "Self-Organization and associative memory" Springer Verlag
- [11] Linde Y., A. Buzo, R.M. Gray (1980) "An algorithm for vector quantizer design" *IEEE TRans. Comm.* 28 pp.84-95

- [12] Nasrabadi NM, Y. Feng (1988) "Vector Quantization of images based upon the Kohonen Self-Organizing feature maps" IEEE Int. Conf. on Neural Net. San Diego pp.1101-1108
- [13] Naylor J., K.P. Li (1988) "Analysis of neural network algorithm for vector quantization of speech parameters" Proc. First Ann. INNS Meet. Pergamon Press p.310-315
- [14] Park J, JW Sandberg (1991) "Universal approximation using radial basis function networks" Neural Computation 3 pp.246-257
- [15] Sethi, I.K. (1981) "A Fast Algorithm for Recognizing Nearest Neighbors" IEEE trans. Syst. Man Cyb. 11 pp.245-248
- [16] Tomasini, L. (1993) "Apprentissage d'une representation statistique et topologique d'un environment" PhD Thesis, Ecole Nationale Superieure de l'Aeronautique et de l'Espace.
- [17] Venkateswarln N.B. , Rajn P.S.V.S.K.(1993)"Fast Isodata Clustering Algorithms" Pattern Recognition 25 (1) pp.335-342
- [18] Verleysen M., K. Hlavackova (1994) "An optimized RBF network for approximation of functions" in ESSN'94 pp.175-180
- [19] Xu L, A Krzyzak, A Yuille (1994) "On radial basis function nets and kernel regression: statistical consistency, convergence rates and receptive field size" Neural Networks 7 pp.609-628
- [20] Yair E., K. Zeger, A. Gersho (1992) "Competitive Learning and Soft Competition for Vector Quantization" IEEE Trans. Sign. Proc. 40(2) pp.294-308



Figure 1. Original image



Figure 2. Image decoded from the codification obtained with NN. Compression rate 6,4.



Figure 3 Image decoded from the codification obtained with LSC. Compression rate 6,4.