

BICONN: A Binary Competitive Neural Network

J. Muñoz-Pérez, M.A. García-Bernal, I. Ladrón de Guevara-López and J.A. Gómez-Ruiz

Dpto. Lenguajes y Ciencias de la Computación
E.T.S. Ingeniería Informática. Universidad de Málaga
Campus de Teatinos s/n
29071-Málaga, Spain.
e-mail: munozp@lcc.uma.es

Abstract. In this paper a competitive neural network with binary synaptic weights is proposed. The aim of this network is to cluster or categorize binary input data. The neural network uses a learning mechanism based on activity levels that generates new binary synaptic weights that become *medianoids* of the clusters or categorizes that are being formed by process units of the network, since the *medianoid* is the better representation of a cluster for binary data when the Hamming distance is used. The proposed model has been applied to codebook generation in vector quantization (VQ) for binary fingerprint image compression. The binary neural network find a set of representative vectors (codebook) for a given training set minimizing the average distortion.

1 Introduction

The unsupervised competitive learning is a mechanism that allows the detection of regularities in the patterns inputs. It was introduced by Grossberg ([1] and [2]) and von der Malsburg ([3]) and developed by Amari *et al* ([4] and [5]), Bienstock *et al* ([6]) and Rumelhart *et al* ([7]). The aim of unsupervised competitive learning is to cluster or categorize the input data. Similar inputs should be classified as being in the same category, and so should fire the same output unit.

A competitive neuronal network consists of a single layer of M process units (neurons) fully connected to a same input $\mathbf{x} \in \mathbb{R}^N$ and producing an outputs $y_i \in \{0,1\}$, $i=1,2,\dots,M$. We say that process unit i is active if $y_i=1$. For each input (stimulus) there is only one active unit. This active unit is called the *winner* and is determined as the unit with largest *activation potential*. The activation potential of the process unit i is the inner products $\mathbf{w}_i^T \mathbf{x}$, where \mathbf{w}_i is the synaptic weight vector of process unit i and $\|\mathbf{w}_i\|=1$. Thus the winner is that process unit with the weight vector closest to the input vector \mathbf{x} (in Euclidean distance sense). That is, the best match of the input vector \mathbf{x} with the synaptic weight vectors. The synaptic weight vector of the winning process unit, r , is updated according to the standard competitive learning rule,

$$\Delta \mathbf{w}_r = \eta (\mathbf{x} - \mathbf{w}_r). \quad (1)$$

which moves \mathbf{w}_r directly toward \mathbf{x} . In this way, the synaptic weight vectors will become the *centroids* of the M clusters or categories that are formed by the networks. However, since the learning parameter $\eta \in (0,1)$ the new weight vector \mathbf{w}_r could be no binary. We look for a new learning rule where the weight vectors became *medianoids* of cluster of input data. Note that the *medianoid* of a cluster of binary inputs is its best representation.

A well known binary competitive network is the Hamming network that is a maximum likelihood classifier that can be used to determine which of a group of prototype vectors is more similar to the input vector (stimulus). The prototype vectors determine the synaptic weights. The measure of similarity between the input vector and stored prototypes vectors is given by N minus the Hamming distance between these vectors. Another learning model that also allows the formation of clusters is the method ART1 (Adaptative Resonance Theory) that was developed in [8] (a simplified version has been shown in [9]). This algorithm adjusts the winning vector \mathbf{w}_r by deleting any bits in it that are not also in \mathbf{x} . However, while \mathbf{w}_r preserves its binary nature, the new prototype \mathbf{w}_r can only have fewer and fewer 1s as training progresses.

In this paper a new model with binary inputs, output and synaptic weights is proposed based on a learning rule that preserves the binary nature of the synaptic weights and they become the *medianoids* of clusters of input data. Moreover, the binary synaptic weight vectors are not normalized as in the simple competitive learning and so we have 2^N possible binary vectors instead of N . This algorithm uses only binary operations and so it is very computationally efficient. The rest of this paper is organized as follows. In the section 2 we develop the model Biconn (Binary Competitive Neural Network). We present an application to compression of fingerprint images by codebook generation in section 3. Conclusions are presented in the section 4.

2 A Binary Competitive Neural Networks

Consider the Hamming space \mathbf{H}^N ,

$$\mathbf{H}^N = \{\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathcal{R}^N: x_i \in \{0, 1\}, i=1, 2, \dots, N\}$$

The Hamming distance between the binary vectors $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_N)^T \in \mathcal{R}^N$, is given by expression $d_H(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T(\mathbf{1} - \mathbf{y}) + (\mathbf{1} - \mathbf{x})^T \mathbf{y}$. Let $C = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ be a set of p vectors of \mathbf{H}^N . We define the *medianoid* of C as the vector $\mathbf{m} = (m_1, m_2, \dots, m_N)$ of \mathbf{H}^N given by expression

$$m_i = \begin{cases} 1 & \text{if } u_i \geq v_i \\ 0 & \text{if } u_i < v_i \end{cases}$$

where u_i is the number of i -th components of vectors of C with value one and v_i is the number of i -th components of vectors of C with value zero. It is easy to prove that the *medianoid* de C is the best representation of C by a vector of \mathbf{H}^N when the Hamming

distance is used. That is, a *medianoid* of C is the vector \mathbf{w} that minimizes the distortion function given by the expression

$$\sum_{j=1}^p d_H(\mathbf{x}_j, \mathbf{w}). \quad (2)$$

Next we propose a binary competitive neural networks to cluster or categorize input data where synaptic weight vectors become the *medianoids* of clusters.

The binary neural network consist of a single layer of M output units, $\{O_1, O_2, \dots, O_M\}$, each receiving the same input $\mathbf{x}=(x_1, x_2, \dots, x_N)'$ and producing an outputs $\{y_1, y_2, \dots, y_M\}$. They are fully connected to a set of inputs (x_1, x_2, \dots, x_N) via connections w_{ij} , $i=1, 2, \dots, M$, $j=1, 2, \dots, N$, that are called synaptic weights. We consider that inputs, outputs and synaptic weights are binary. Only one of the output units, called the winner, can be activated at a time. The active unit is determined as the unit with the largest net input, where the net input of unit i is given by the expression

$$h_i = \sum_{j=1}^N w_{ij} x_j - \frac{1}{2} \sum_{j=1}^N w_{ij} = \sum_{j=1}^N \left(x_j - \frac{1}{2} \right) w_{ij} \quad (3)$$

and its output is

$$y_i(k) = \begin{cases} 1 & \text{if } h_i = \max_r \{h_r\} \\ 0 & \text{otherwise} \end{cases}, \quad i=1, 2, \dots, M$$

Thus, the winner is that unit with weight vector closest (in Hamming distance sense) to the input vector. It is established in the following proposition:

Proposition 1

$$h_i > h_r \Leftrightarrow d_H(\mathbf{x}, \mathbf{w}_i) < d_H(\mathbf{x}, \mathbf{w}_r) \quad (4)$$

where $\mathbf{w}_i=(w_{i1}, w_{i2}, \dots, w_{iN})'$ is the synaptic vector of the unit i .

Proof.

We have

$$\begin{aligned} d_H(\mathbf{x}, \mathbf{w}_i) &= (d_E(\mathbf{x}, \mathbf{w}_i))^2 = (\mathbf{x} - \mathbf{w}_i)^T (\mathbf{x} - \mathbf{w}_i) \\ &= \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{w}_i + \mathbf{w}_i^T \mathbf{w}_i \\ &= \mathbf{x}^T \mathbf{x} - 2h_i \end{aligned}$$

since $\mathbf{w}_i^T \mathbf{w}_i = 1$. Thus, the result is followed.

Next we present a learning process to determine the synaptic weights. First, we consider a set $C = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ of p inputs. We want to find M synaptic vectors to minimize the distortion function, that is,

$$E(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M) = \sum_{i=1}^M \sum_{j=1}^p d_H(\mathbf{x}_j, \mathbf{w}_i) \delta_i(\mathbf{x}_j, \mathbf{w}_1, \dots, \mathbf{w}_M) \quad (5)$$

where

$$\delta_i(\mathbf{x}_j, \mathbf{w}_1, \dots, \mathbf{w}_M) = \begin{cases} 1 & \text{if } d_H(\mathbf{x}_j, \mathbf{w}_i) = \min_{1 \leq r \leq M} \{d_H(\mathbf{x}_j, \mathbf{w}_r)\} \\ 0 & \text{otherwise} \end{cases}$$

In this way, synaptic vectors are the best representation of inputs data by M binary vectors. That is, the synaptic vectors must be medianoids of clusters of the input data.

The rule of simple competitive learning for continuous inputs leads to a update of the neural network so that its synaptic vectors are moved toward input vectors, that is, the new synaptic vector is a linear combination between the old synaptic vector and the input vector. However, the lineal combination of two binary vectors could be a no binary vector and so the new synaptic vector could be a no binary vector. It is necessary a new learning process such that the new synaptic vector becomes a binary vector at the same time that it also comes near to input vector. Thus, we propose a new learning rule based on *activity levels* to attain this goal. In each stage, we updates the components of the winner synaptic vector according to a learning rate $\eta_{ij}(k)$ that depends on the *activity level*. The *activity level* of the j th component of the synaptic weight \mathbf{w}_i after presentation of the k th training pattern, $\mathbf{x}(k) = (x_1(k), x_2(k), \dots, x_N(k))^T$, is given by

$$a_{ij}(k) = \sum_{s=1}^k \left(x_j(s) - \frac{1}{2} \right) y_i(s) = \sum_{s \in C_i(k)} \left(x_j(s) - \frac{1}{2} \right) \quad (5)$$

where $C_i(k) = \{s \in Z_+ : y_i(s) = 1, s \leq k\}$. It give us a balance between zeros and ones of the j th components of presentation patters with $y_i(s) = 1, s \leq k$. That is, if $a_{ij}(k) > 0$ then the median of the set $\{x_j(s), s \in C_i(k)\}$ is one while if $a_{ij}(k) < 0$ then the median is zero. Note that

$$a_{ij}(k) = a_{ij}(k-1) + \left(x_j(k) - \frac{1}{2} \right) y_i(k). \quad (6)$$

The new learning rule is given by expression

$$\Delta w_{ij}(k+1) = w_{ij}(k+1) - w_{ij}(k) \quad (7)$$

$$= \eta_{ij}(k) \left(x_j(k) - w_{ij}(k) \right) \quad (8)$$

where

$$\eta_{ij}(k) = \begin{cases} 0 & \text{if } \text{sgn}(a_{ij}(k)) = \text{sgn}(a_{ij}(k-1)) \\ 1 & \text{otherwise} \end{cases}$$

That is, the weight w_{ij} is changed only when the unit i is winner and the sign of the *activation level* is modified by the new training pattern. Note that the expression (8) is similar to simple competitive rule but the learning parameter is now different. In this way, the synaptic weight vectors will become *medianoids* of inputs patters.

It is easy to prove that this learning rule guarantees that the new synaptic vector will be more similar to input vector than the old synaptic weight. It is established in the next proposition.

Proposition 2

$$d_H(\mathbf{w}_i(k+1), \mathbf{x}(k)) \leq d_H(\mathbf{w}_i(k), \mathbf{x}(k)), \quad i=1,2,\dots,M. \quad (9)$$

Proof:

It is obvious since

$$w_{ij}(k) = \begin{cases} x_j(k) & \text{if } \eta_{ij}(k) = 1 \\ w_{ij}(k) & \text{if } \eta_{ij}(k) = 0 \end{cases}$$

Thus

$$\begin{aligned} d_H(\mathbf{w}_i(k+1), \mathbf{x}(k)) &= \sum_{j: \eta_{ij}(k)=0} |w_{ij}(k) - x_j(k)| \\ &\leq \sum_{j=1}^N |w_{ij}(k) - x_j(k)| \\ &= d_H(\mathbf{w}_i(k), \mathbf{x}(k)). \end{aligned}$$

Next we study the binary learning parameter η_{ij} . Let $\{\mathbf{x}(r), r=1,2,\dots,p\}$ be drawn independently according to a finite mixture distribution. Then $\eta_{ij}(k)$ is a random variable that depends on $\{x_j(s), s \in C_i(k)\}$. If $P(X_j(r)=1) = p, r \in C_i(k)$, then

$$\begin{aligned} P(\eta_{ij}(k)=1) &= P(a_{ij}(k) \neq a_{ij}(k-1)) \\ &= P\left(\sum_{r \in C_i} X_j(r) \geq \frac{|C_i|}{2}, \sum_{r \in C_i} X_j(r) + X_j(k) < \frac{|C_i|+1}{2}\right) \\ &\quad + P\left(\sum_{r \in C_i} X_j(r) < \frac{|C_i|}{2}, \sum_{r \in C_i} X_j(r) + X_j(k) \geq \frac{|C_i|+1}{2}\right) \\ &= \begin{cases} \binom{k}{k/2} p^{k/2} (1-p)^{1+k/2} & \text{if } k \text{ is even} \\ \binom{k}{(k-1)/2} p^{(k+1)/2} (1-p)^{(k+1)/2} & \text{if } k \text{ is odd} \end{cases} \quad (10) \end{aligned}$$

This probability tends to zero as k tend to infinite. So the learning parameter $\eta_{ij}(k)$ entails a convergence to zero. Like competitive neural networks, this network evolves until a local minimum is reached.

In this way we have the following algorithm:

Step 1: Initialization.

Generate M binary synaptic vectors $\mathbf{w}_1, \dots, \mathbf{w}_M$.

Step 2: k th iteration. Synaptic *potentials*.

Given a input vector, $(x_1(k), x_2(k), \dots, x_N(k))$ determine the synaptic potentials h_i , $i=1, 2, \dots, M$, by expression

$$h_i = \sum_{j=1}^N (x_j(k) - 0.5) w_{ij}$$

Step 3: *Winner unit.*

The unit r is activated (winner), $y_r=1$, if $h_r \geq h_j$, $j=1, 2, \dots, M$.

Step 4: *Update activity levels*

$$a_{ij}(k) = a_{ij}(k-1) + (x_j(k) - \frac{1}{2}) y_i(k)$$

Step 4: *Update synaptic weight*

$$\Delta w_{ij}(k+1) = \eta_{ij}(k) (x_j(k) - w_{ij}(k))$$

where

$$\eta_{ij}(k) = \begin{cases} 0 & \text{if } \text{sgn}(a_{ij}(k)) = \text{sgn}(a_{ij}(k-1)) \\ 1 & \text{otherwise} \end{cases}$$

Paso 6: *Stop rule.*

If the learning parameters are equal to zero during an epoch of training, stop. In other case, go to step 2.

On the other hand, in an update in batch mode we only have to assign to each synaptic vector, \mathbf{w}_i , the *medianoid* vector of the set of inputs that activate unit i .

3. Experimental results: Codebook generation in vector quantization

The aim of this network is to cluster or categorize the input data. It can be used for data encoding and compression through vector quantization, where each input vector is replaced by the code of the winner output unit. In this application, the digital images to be encoded are decomposed into small blocks, say 3 by 3 pixels, called *vectors*. The resulting vectors are represented by the “nearest” of a reduced set of prototype vectors, called *codewords*. The set of codewords used to represent an image, or a portion of an image, is called *codebook*. The codebook is given by the synaptic weights of the network.



Fig. 1. (a) Fingerprint image. (b) Compressed fingerprint image.

To illustrate the performance of the proposed algorithm, we consider a fingerprint image of size 255×255 (see figure 1(a)). It is partitioned in 7225 blocks (windows) of size 3×3 . The problem is to find 4 prototype blocks (windows) of size 3×3 such that when each block is represented by one of the prototype blocks then we obtain the best possible representation. That is, we want to minimize the distortion function and so the prototype blocks should be medianoids.

In our experiments, we use the *peak signal to noise ratio* (PSNR) to evaluate the quality of a compressed image. For an original image $F = \{f_{ij}, i=1,2,\dots,m, j=1,2,\dots,n\}$ and its corresponding compressed image $F' = \{f'_{ij}, i=1,2,\dots,m, j=1,2,\dots,n\}$, the PSNR is defined as follows:

$$\text{PSNR} = 10 \log_{10} \frac{1}{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (f_{ij} - f'_{ij})^2}$$

where 1 is the peak grey level of the image, f_{ij} and f'_{ij} are the pixel grey levels from the original and compressed images, and $m \times n$ is the number of pixels in the image. In general, the higher PSNR value of an image implies the better image quality.

The compressed image is constituted by prototype blocks, that is, medianoids, and when each block is replaced by its prototype block then we would obtain the most look like image to original image. In this way, we need only 2 bits to represent 4 prototype blocks while each block of the original image is represented by 9 bits, so the compression rate was about 9 to 2, that is, *2/9 bits per pixels* (bpp). This problem is solved by a binary competitive network with 4 units. It finds the prototypes blocks

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

They correspond to 3311, 2704, 629 and 581 windows, respectively. The mean distortion error of a block is 0.8851 and the PSNR achieved was 10.07 dB. The figure 2(b) shows the compressed image. The compressed image retains features as such ridge lines, ridge bifurcation, arch, deltas, etc. Note that the new compressed image is formed by only 4 different blocks and so it can be compressed better than the original

one by any compression techniques. Moreover, like other algorithms, the algorithm finds several solutions (local minimum); it depends by the initial set of synaptic weights. So, it also finds other prototype blocks such as

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

that correspond to 843, 839, 2840 and 2703 windows, respectively. The figure 2(c) shows the compressed image. The mean distortion error of a block is 0.8861 and the PSNR achieved was 10.06 dB. This experiment has a lot of local minimum but the network always find solutions with mean distortion error less than 1. Although the experimental outcomes do not distinguish the proposed algorithm from the standard competitive learning algorithm and other algorithms such as the well-known generalized Lloyd algorithm (GLA), also referred as *k-means* algorithm due to McQueen (see [10]), however, it always uses binary synaptic weights and binary operations, and evolves from binary values to binary values, whereas other algorithms use real values. Hence, the proposed algorithm is more computationally efficient.

4. Conclusions

A binary competitive neural network has been proposed where synaptic weight vectors are binary vectors. The new synaptic weight vector is obtained by a learning mechanism that guarantees that it will be closer to input vector and at the same time that it will be also binary. First, we have shown that a necessary condition for a optimum solution to the problem to minimize the distortion function is that binary vectors have to be *medianoids*. In the proposed model each synaptic weight vector evolves to the *medianoid* vector of cluster that is being formed by process units of the network. Moreover, this model is more computationally efficient than the simple competitive model with continuous weights. Finally, the model has been applied to image compression and though it reaches a local minimum this is global or a good solution.

References

1. Grossberg, S. Neural expectation: Cerebellar and retinal analogues of cells fired by unlearnable and learnable pattern classes, *Kybernetik*, **10** (1972) 49-57.
2. Grossberg, S. Adaptive pattern classification and universal recording: I. Parallel development and coding of neural detectors, *Biolog. Cybernetics*, **23** (1976) 121-134.
3. von der Malsburg, C. Self-organization of orientation sensitive cells in the striate cortex, *Kybernetik*, **14** (1973) 85-100.
4. Amari, S. and Takeuchi, M. Mathematical theory on formation of category detecting nerve cells, *Biological Cybernetics*, **29** (1978) 127-136.

5. Amari, S.-I. Field theory of self-organizing neural nets, *IEEE Transactions on Systems, Man and Cybernetics, SMC-13* (1983) 741-748.
6. Bienenstock, E., Cooper, E. & Munro, P. Theory for the development of neural selectivity: Orientation specificity and binocular interaction in visual cortex, *Journal of Neuroscience*, **2** (1982) 32-48.
7. Rumelhart, D. and Zipser, D. (1985). Feature discovery by competitive learning, *Cognitive Science*, **9** (1985) 75-112.
8. Carpenter, G. A. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing* **37** (1987) 54-115.
9. Hassoun M.H. *Fundamentals of Artificial Neural Networks*. The MIT Press, Cambridge, 1995.
10. Gersho A. and Gray R.M. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.