Mathematical evidence for target vector type influence on MLP learning improvement

José Ricardo Gonçalves Manzan, Shigueo Nomura and Keiji Yamanaka

Abstract – ICAI - *This work proposes a mathematical proof for the use of orthogonal bipolar vectors (OBV) rather than conventional target vectors in artificial neural network MLP learning. A larger Euclidean distance provided by new target vectors is explored to improve the learning and generalization abilities of MLPs. The proposed proof compares the MLP performances by using different target vectors such as conventional binary and bipolar and orthogonal bipolar vectors. The evidence for performance improvement is shown by the study of updating process for the weights through the backpropagation algorithm. We have concluded that the use of orthogonal bipolar vectors as targets can provide a better keep of each pattern feature and reduce the interference of noises from a training pattern to the other one.*

Keywords: Mathematical proof, pattern recognition, multilayer perceptron, target vectors, orthogonal bipolar vectors

1 Introduction

Computational intelligence is a science field that has emerged as a set of powerful tools capable of solving problems that previously could not be solved. In this context, we have the Artificial Neural Networks (ANN) receiving important contributions from researchers since 80's. It is quite difficult to list all the ANN applications. Some applications are pattern recognition [1], sound signal processing [2], and biomedical signal processing [3].

Related works are presented in Section 2. Section 3 presents a motivation for the work. Hypothesis to be solved by this work is described in Section 4. The different types of target vectors are defined in Section 5. In Section 6, we can verify the mathematical evidence for affecting the MLP performance according to different target vector types. Some results are discussed in Section 7. Section 8 presents the conclusion of this work.

J. R. G. Manzan is with the Faculty of Electrical Engineering, Federal University of Uberlândia, MG 38400-902. Phone: +51 34 3319-6000, e-mail: josericardo@iftm.edu.br (corresponding author)
S. Nomura is with the Faculty of Electrical Engineering, Federal University of Uberlândia, MG 38400-902. Phone: +51 34 3239-4704, e-mail: shigueonomura@feelt.ufu.br
K. Yamanaka is with the Faculty of Electrical Engineering, Federal University of Uberlândia, MG 38400-902. Phone: +51 34 3239-4704, e-mail: keiji@ufu.br

2 Related works

Researches on pattern recognition mainly description and classification have been considered important in the computer field. Several techniques such as statistical approach, theoretical decision and syntactic approach have been adopted [11]. Currently, the ANN techniques have been widely used because of promising results. One of the advantages of using ANN is the ability for training in a supervised or unsupervised form.

It is known that traditional approaches on artificial intelligence use the sequential processing. On the other hand, ANN techniques use a learning mode with parallel and distributed processing. Their training methodology is based on biological neuron activity to learn through examples. Trial and error strategies contribute to the ability to differentiate patterns. ANN has a similar behavior when a large number of neurons send excitatory or inhibitory signals to other neurons composing the network.

Several researchers [4] [5] [6] [7] have focused on improving ANN performances. Some proposed strategies are regarded to input pattern improvement, ANN architecture optimization, learning algorithm enhancement and others.

Experimental results related to this work have been presented in [7] [8] [9] [10] showing the performance improvements.

3 Motivation

The biological cognition has abilities to recognize and distinguish patterns, even if they have a high degree of degradation an their features [12] [13] [14]. In case of ANN, an appropriate adjustment of parameters allows a learning with high degree of generalization. This is good for constructing a model with high flexibility to properly recognize very degraded patterns. However, if the training time is over then, the model becomes too rigid preventing the recognition of degraded patterns.

Several proposals in order to improve the ability to recognize degraded patterns have been carried out. In most cases they have focused on how to treat input vectors [15]. However, studies for the treatment of target vectors are still rare. This work shows effects of adopting orthogonal bipolar vectors as targets on improving the MLP performance to recognize degraded patterns. The previous works [7] [8] [9] [10] show satisfactory results in using orthogonal bipolar vectors as expectation values for MLP learning.

4 Hypothesis

In case of conventional bipolar vectors (CBV), the inner product between two of them is not null. On the other hand, orthogonal bipolar vectors (OBV) always have null inner product between them. Also, the similarity between two OBVs is lower than that corresponding similarity between two CBVs. Furthermore, the orthogonality between two OBVs leads to the largest Euclidean distance as well as possible. We believe that larger Euclidean distance and lower similarity of OBVs can affect on the MLP performance improvement to recognize degraded patterns.

However, we have realized that there is no investigation studying the influence of target vector type on the MLP learning. This paper proposes a new methodology for the learning. Our hypothesis is on the fact that a target vector type can significantly improve the ability of MLPs in recognized degraded patterns.

This paper presents a mathematical evidence for explaining the performance improvement of MLPs.

5 Representation of vectors

5.1 Orthogonal Bipolar Vector (OBV)

Equations (1) and (2) represent two possible target vectors, the equation (3) represents the inner product and equation (4) the Euclidean distance.

$$\vec{V} = \left(v_1, v_2, \ldots, v_n\right) \quad (1)$$

$$\vec{W} = \left(w_1, w_2, \ldots, w_n\right) \quad (2)$$

$$\vec{V} \cdot \vec{W}^T = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3 + \ldots + v_n \cdot w_n \quad (3)$$

$$d_{v,w} = \sqrt{\left(w_1 - v_1\right)^2 + \left(w_3 - v_3\right)^2 + \left(w_3 - v_3\right)^2 + \ldots + \left(w_n - v_n\right)^2} \quad (4)$$

Consider the case where $\vec{V}$ and $\vec{W}$ are orthogonal with size n. There will be n/2 components whose product is positive and n/2 components whose product is negative. Positive product components is correspond to the ones which the terms have the same signal. These terms do not affect on the result of the Euclidean distance given by equation (4). On the other hand, for the terms with opposite signals, the square of their difference is 4. The squares of differences contribute into the Euclidean distance resolution. Therefore, if we have larger number (n) of components then we have larger Euclidean distances. Equations (5) and (6) represent examples of OBVs. The inner product of those OBVs is given by equation (7). The OBVs can de generated by implementing the algorithm as described in [16].

$$\vec{V} \overset{def}{=} \left(1,1,1,1,-1,-1,-1,-1\right) \quad (5)$$

$$\vec{W} \overset{def}{=} \left(1,1,-1,-1,1,1,-1,-1\right) \quad (6)$$

$$\vec{V} \bullet \vec{W} = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot (-1) + 1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) = 0 \quad (7)$$

5.2 Conventional Bipolar Vector (CBV)

In case of conventional bipolar vector (CBV), one of its components values 1 at the position i corresponding to the pattern i represented by vector $\vec{V}$. All the other components value – 1 as represented by equation (8).

$$\vec{V} \overset{def}{=} \left(-1,-1,\ldots,1,\ldots,-1\right) \quad (8)$$

If $\vec{V}$ and $\vec{W}$ are conventional then the terms equation (4) are null except for two terms corresponding to the positive component of the vector $\vec{V}$ given by equation (8). So, the Euclidean distance for CBVs is smaller than the distance for OBVs.

5.3 Conventional Binary Vector (BV)

The binary vector (BV) is constituted by a unitary component at the position "i" to represent the i<u>th</u> pattern and other null components as given by equation (9).

$$\vec{V} \overset{def}{=} \left(0,0,\ldots,1,\ldots,0,0\right) \quad (9)$$

The BVs are orthogonal between them but their Euclidean distance is always equal to $\sqrt{2}$.

6 Improving the weights between the hidden layer and output layer

6.1 Updating the weights between the hidden layer and output layer

We have considered the backpropagation algorithm foundation [16] to develop the mathematical evidence of our proposal. A pattern of order q is propagated through the error backpropagation $\left(\delta_k\right)$ is given by equation (10).

$$\overset{q}{\delta}_k = \left(\overset{q}{t}_k - \overset{q}{y}_k\right) \cdot f`\left(\overset{q}{yin}_k\right) \tag{10}$$

Where:

- $\overset{q}{t}_k$ Represents the target vector corresponding to the q^th pattern that propagates through the network.
- $\overset{q}{y}_k$ Represents the network output for the q^th pattern propagating through the network.
- $f`\left(\overset{q}{yin}_k\right)$ denotes the differential value for the activation function of the net output considering the q^th pattern.

The vectorial form of equation (10) is given by equation (11).

$$\overset{q}{\delta}_k = \left(\begin{bmatrix} \overset{q}{t}_1 \\ t_2 \\ . \\ . \\ t_k \end{bmatrix} - \begin{bmatrix} \overset{q}{y}_1 \\ y_2 \\ . \\ . \\ y_k \end{bmatrix} \right) .* \left[y_1` \quad \overset{q}{y_2`} \quad \dots \quad y_k` \right]^T \tag{11}$$

Where:

- .* is the symbol for an unusual multiplication of two matrices with the same sizes. In this operation, each component of the first matrix, corresponding to the row i and the column j is multiplied with the corresponding component of the second matrix located at the row i and column j. The result from the operation is a matrix with the same size as the initial matrices.

In case of using BV as target vector we can verity that the k^th component of this vector is 0. So, the difference (t_i - y_i) from equation (11) is between - 1 and 0. On the other hand when the value of k^th of target vector element is 1, the difference (t_i - y_i) from equation (11) is between 0 and 1. These differences are always multiplied by the differential result for the activation function in y_i. The differential results will be positive since the activation function is assintotically non-decreasing. Therefore, δ_k will be negative for null components of BV and δ_k will be positive for the component of BV that is not null.

In case of using CBV as target vector we can note that the difference (t_i - y_i) from equation (11) will be from 0 to 2 for + 1 component of this vector. On the other hand, the difference (t_i - y_i) will be from – 2 to 0 for – 1 component of CBV. Therefore, δ_k will be negative for – 1 component of CBV and it will be positive for +1 components.

In case of using OBV as target vector, we can construct the first vector composed by only +1 components. So, δ_k will be positive for all the components of OBV. The second vector and others are composed by +1 and – 1 components in an equal number. Therefore, a OBV with n components will provide at least n/2 negative results and n/2 positive results of δ_k. δ_k is used for calculating Δw_{jk} in equation (12) and $\Delta w0_k$ in equation (13) as follows:

$$\Delta \overset{q}{w}_{jk} = \alpha \cdot \overset{q}{\delta}_k \cdot \overset{q}{z}_j \tag{12}$$

$$\Delta \overset{q}{w0}_k = \alpha \cdot \overset{q}{\delta}_k \tag{13}$$

In case of using CBV or BV as target vector, we can represent equation (12) as follows:

$$\Delta \overset{q}{w}_{jk} = \alpha \cdot \overset{q}{\delta}_k \cdot \overset{q}{z}_j = \left(\alpha \cdot \begin{bmatrix} \overset{q}{+} \\ - \\ . \\ . \\ - \end{bmatrix} \cdot \begin{bmatrix} + \\ + \\ . \\ . \\ - \\ + \\ - \end{bmatrix}^{q\,T} \right)^T = \alpha \cdot \begin{bmatrix} + & + & \dots & - & + & - \\ - & - & \dots & + & - & + \\ . & . & . & & . & \\ . & . & . & & . & \\ - & - & & + & - & + \\ - & - & \dots & + & - & + \end{bmatrix} \tag{14}$$

In case of using OBV as target vector, equation (12) can be represented as follows:

$$\Delta \overset{q}{w}_{jk} = \alpha \cdot \overset{q}{\delta}_k \cdot \overset{q}{z}_j = \left(\alpha \cdot \begin{bmatrix} \overset{q}{+} \\ + \\ . \\ . \\ - \\ - \end{bmatrix} \cdot \begin{bmatrix} + \\ + \\ . \\ . \\ - \\ + \\ - \end{bmatrix}^{q\,T} \right)^T = \alpha \cdot \begin{bmatrix} + & + & \dots & - & + & - \\ + & + & \dots & - & + & - \\ . & . & . & & . & \\ . & . & . & & . & \\ - & - & & + & - & + \\ - & - & \dots & + & - & + \end{bmatrix} \tag{15}$$

The influence of a target vector type on the term $\Delta w0_k$ can be analyzed as follows:

- In case of using BV or CBV as target vector we can note that the results for $\Delta w0_k$ will be negative for –1 component of the target vector and only one will be positive as given by equation (16);
- In case of using OBV as target vector, it is possible to get at least half number (n/2) of components from the target vector as positive results for $\Delta w0_k$ as given by equation (17).

$$\overset{q}{\Delta w0_k} = \alpha \cdot \overset{q}{\delta_k} = \alpha \cdot \begin{bmatrix} + & - & \overset{q}{\ldots} & - \end{bmatrix}^T \tag{16}$$

$$\overset{q}{\Delta w0_k} = \alpha \cdot \overset{q}{\delta_k} = \alpha \cdot \begin{bmatrix} + & \ldots & + & \overset{q}{-} & \ldots & - \end{bmatrix}^T \tag{17}$$

From equations (14) - (17), we can verify that the use of OBVs as target vectors can keep more pattern feature signal during its propagation. So, we can have more efficient mapping for pattern recognition learning.

6.2 Updating the weights between the input layer and hidden layer

A propagation of two consecutive training patterns will be considered: q order pattern and q + 1 order pattern. Equations (18), (19), and (20) are related to the q order pattern. Equation (21) is related to the q + 1 order pattern.

$$\overset{q}{\delta_k} = \left(\overset{q}{t_k} - \overset{q}{y_k}\right) \cdot f'\left(\overset{q}{yin_k}\right) \tag{18}$$

$$\overset{q}{\Delta w_{jk}} = \alpha \cdot \overset{q}{\delta_k} \cdot \overset{q}{z_j} \tag{19}$$

$$\overset{q+1}{w_{jk}} = \overset{q}{w_{jk}} + \overset{q}{\Delta w_{jk}} \tag{20}$$

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \left[\overset{q+1}{\delta_k} \cdot \overset{q+1}{w_{jk}} \right] \tag{21}$$

Replacing equations (18) and (20) with the terms δ_k and Δw_{jk} from equation (21), we can obtain the following equation (22):

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \left\{ \left[\left(\overset{q+1}{t_k} - \overset{q+1}{y_k} \right) \cdot f'\left(\overset{q+1}{yin_k} \right) \right] \cdot \left[\overset{q}{w_{jk}} + \overset{q}{\Delta w_{jk}} \right] \right\} \tag{22}$$

Also, replacing equation (19) with the term of equation (22), we can get the following equation (23):

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \left\{ \left[\overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) - \overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right] \cdot \left[\overset{q}{w_{jk}} + \alpha \cdot \overset{q}{\delta_k} \cdot \overset{q}{z_j} \right] \right\} \tag{23}$$

Furthermore, replacing equation (18) with the term of equation (23), we can obtain equation (24):

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \left\{ \begin{bmatrix} \overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \\ - \overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \end{bmatrix} \cdot \begin{bmatrix} \overset{q}{w_{jk}} + \\ \alpha \cdot \left(\overset{q}{t_k} - \overset{q}{y_k} \right) \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \end{bmatrix} \right\} \tag{24}$$

From equation (24), we can obtain equation (25).

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \left\{ \begin{bmatrix} \overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \\ - \overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \end{bmatrix} \cdot \begin{bmatrix} \overset{q}{w_{jk}} + \alpha \cdot \overset{q}{t_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \\ -\alpha \cdot \overset{q}{y_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \end{bmatrix} \right\} \tag{25}$$

Applying the distributive property to the matrix multiplication, we can obtain equation (26).

$$\overset{q+1}{\delta in_j} = \sum_{k=1}^{m} \begin{bmatrix} \left(\overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\overset{q}{w_{jk}} \right) \\ + \left(\overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\alpha \cdot \overset{q}{t_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \right) \\ - \left(\overset{q+1}{t_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\alpha \cdot \overset{q}{y_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \right) \\ - \left(\overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\overset{q}{w_{jk}} \right) \\ - \left(\overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\alpha \cdot \overset{q}{t_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \right) \\ + \left(\overset{q+1}{y_k} \cdot f'\left(\overset{q+1}{yin_k} \right) \right) \cdot \left(\alpha \cdot \overset{q}{y_k} \cdot f'\left(\overset{q}{yin_k} \right) \cdot \overset{q}{z_j} \right) \end{bmatrix} \tag{26}$$

Converting equation (26) into a vector representation, the expression is given by equation (27).

$$
\begin{aligned}
\overset{q+1}{\delta in_j} = {} & \overset{q+1}{[t_1 \quad t_2 \quad \ldots \quad t_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \overset{q}{\begin{bmatrix} w_{11} & w_{21} & \ldots & w_{j1} \\ w_{12} & w_{22} & \ldots & w_{j2} \\ . & . & . & . \\ . & . & . & . \\ w_{1k} & w_{2k} & \ldots & w_{kj} \end{bmatrix}} \\
& + \overset{q+1}{[t_1 \quad t_2 \quad \ldots \quad t_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \alpha \cdot \overset{q}{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_k \end{bmatrix}} .* \overset{q}{\begin{bmatrix} y_1` \\ y_2` \\ . \\ . \\ y_k` \end{bmatrix}} \cdot \overset{q}{\begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix}} \\
& - \overset{q+1}{[t_1 \quad t_2 \quad \ldots \quad t_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \alpha \cdot \overset{q}{\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_k \end{bmatrix}} .* \overset{q}{\begin{bmatrix} y_1` \\ y_2` \\ . \\ . \\ y_k` \end{bmatrix}} \cdot \overset{q}{\begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix}} \\
& - \overset{q+1}{[y_1 \quad y_2 \quad \ldots \quad y_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \overset{q}{\begin{bmatrix} w_{11} & w_{21} & \ldots & w_{j1} \\ w_{12} & w_{22} & \ldots & w_{j2} \\ . & . & . & . \\ . & . & . & . \\ w_{1k} & w_{2k} & \ldots & w_{kj} \end{bmatrix}} \\
& - \overset{q+1}{[y_1 \quad y_2 \quad \ldots \quad y_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \alpha \cdot \overset{q}{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_k \end{bmatrix}} .* \overset{q}{\begin{bmatrix} y_1` \\ y_2` \\ . \\ . \\ y_k` \end{bmatrix}} \cdot \overset{q}{\begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix}} \\
& + \overset{q+1}{[y_1 \quad y_2 \quad \ldots \quad y_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \alpha \cdot \overset{q}{\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_k \end{bmatrix}} .* \overset{q}{\begin{bmatrix} y_1` \\ y_2` \\ . \\ . \\ y_k` \end{bmatrix}} \cdot \overset{q}{\begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix}}
\end{aligned}
\tag{27}
$$

Extracting the second term from equation (27), we have equation (28).

$$
\overset{q+1}{[t_1 \quad t_2 \quad \ldots \quad t_k]} .* \overset{q+1}{[y_1` \quad y_2` \quad \ldots \quad y_k`]} \cdot \alpha \cdot \overset{q}{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_k \end{bmatrix}} .* \overset{q}{\begin{bmatrix} y_1` \\ y_2` \\ . \\ . \\ y_k` \end{bmatrix}} \cdot \overset{q}{\begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix}} \tag{28}
$$

In equation (28), we have a positive scalar α representing the learning rate and vectors represented by the following expressions:

$$
\overset{q+1}{t_k} = \begin{bmatrix} a_1 & a_2 & \ldots & a_k \end{bmatrix} \tag{29}
$$

$$
f`\left(\overset{q+1}{yin_k}\right) = \begin{bmatrix} m_1 & m_2 & \ldots & m_k \end{bmatrix} \tag{30}
$$

$$
\overset{q}{t_k} = \begin{bmatrix} b_1 & b_2 & \ldots & b_k \end{bmatrix}^T \tag{31}
$$

$$
f`\left(\overset{q}{yin_k}\right) = \begin{bmatrix} n_1 & n_2 & \ldots & n_k \end{bmatrix}^T \tag{32}
$$

$$
\overset{q}{z_j} = \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \tag{33}
$$

From equations (28) – (33), we can obtain equation (34) as follows:

$$
\left([a_1 \quad a_2 \quad \ldots \quad a_k] .* [m_1 \quad m_2 \quad \ldots \quad m_k] \right) \cdot \left(\alpha \cdot \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_k \end{bmatrix} .* \begin{bmatrix} n_1 \\ n_2 \\ . \\ . \\ n_k \end{bmatrix} \cdot \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \right) \tag{34}
$$

Solving only the especial results from multiplication element by element corresponding to ".*", we can get equations (35) and (36):

$$
\begin{bmatrix} a_1 m_1 & a_2 m_2 & \ldots & a_k m_k \end{bmatrix} \cdot \alpha \cdot \begin{bmatrix} b_1 n_1 \\ b_2 n_2 \\ . \\ . \\ b_k n_k \end{bmatrix} \cdot \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \tag{35}
$$

$$
\alpha \cdot \begin{bmatrix} a_1 m_1 & a_2 m_2 & \ldots & a_k m_k \end{bmatrix} \cdot \begin{bmatrix} b_1 n_1 \\ b_2 n_2 \\ . \\ . \\ b_k n_k \end{bmatrix} \cdot \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \tag{36}
$$

From equation (36), we can get equations (37) and (38).

$$
\alpha \cdot \left(a_1 m_1 b_1 n_1 + a_2 m_2 b_2 n_2 + \ldots + a_k m_k b_k n_k \right) \cdot \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \tag{37}
$$

$$
\alpha \cdot \left(a_1 b_1 m_1 n_1 + a_2 b_2 m_2 n_2 + \ldots + a_k b_k m_k n_k \right) \cdot \begin{bmatrix} z_1 & z_2 & \ldots & z_j \end{bmatrix} \tag{38}
$$

Extracting the scalar term from equation (38), we can obtain equation (39).

$$
\left(a_1 b_1 m_1 n_1 + a_2 b_2 m_2 n_2 + \ldots + a_k b_k m_k n_k \right) \tag{39}
$$

The products $m_i n_i$ for $1 \le i \le k$ are positive. These results are derived from the activation function that was supposed to be assintotically non-decreasing. The components a_i and b_i for $1 \le i \le k$, depend on the target vector. They can be 0, 1 or – 1 and cause different effects as follows:

- In case of CBV as target vector, the product $a_i b_i$ for $1 \le i \le k$ will be – 1 for two terms of equation (39);
- In case of BV as target vector, the product $a_i b_i$ for $1 \le i \le k$ will be null;
- In case of OBV as target vector, the result from equation (39) will be smaller than for the case of using CBV.

Considering the use of CBVs as target vectors with four or more components, the result from equation (39) will be larger than the case of using OBVs.

7 Discussion

As illustration, we have taken CBVs with 16 components and replace the variables of equations (29) - (32) with numerical values as follows:

$$f'\left(\overset{q+1}{yin_k}\right) = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \quad (40)$$

$$\left[f'\left(\overset{q}{yin_k}\right)\right]^T = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \quad (41)$$

$$\overset{q+1}{t_k} = [1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1] \quad (42)$$

$$\left[\overset{q}{t_k}\right]^T = [-1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1] \quad (43)$$

Then, equation (28) can be numerically expressed by equations (44) and (45).

$$= \left(\left(\underbrace{[1\ -1\ \dots\ -1\ -1]}_{1\times16} .* \underbrace{[1\ 1\ \dots\ 1\ 1]}_{1\times16}\right) \cdot \alpha \bullet \left(\underbrace{\begin{bmatrix}-1\\1\\-1\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ -1\end{bmatrix}}_{16\times1} .* \underbrace{\begin{bmatrix}1\\1\\1\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ 1\end{bmatrix}}_{16\times1}\right)\right) \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]}$$

$$= \left(\underbrace{[1\cdot1\ \ (-1)\cdot1\ \ (-1)\cdot1\ \dots\ (-1)\cdot1]}_{1\times16} \cdot \alpha \bullet \underbrace{\begin{bmatrix}(-1)\cdot1\\1\cdot1\\(-1)\cdot1\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ (-1)\cdot1\end{bmatrix}}_{16\times1}\right) \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]} \quad (44)$$

$$= [1\cdot1\cdot(-1)\cdot1 + (-1)\cdot1\cdot1\cdot1 + (-1)\cdot1\cdot(-1)\cdot1 + \dots + (-1)\cdot1\cdot(-1)\cdot1] \cdot \alpha \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]} \quad (45)$$
$$= 14 \cdot \alpha \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]}$$

Also, we have taken OBVs with 16 components and replaced the variables of equations (29) and (31) with numerical values as follows:

$$\overset{q+1}{t_k} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \quad (46)$$

$$\left[\overset{q}{t_k}\right]^T = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1] \quad (47)$$

Then, equation (28) can be numerically expressed by equations (48) and (49).

$$= \left(\left(\underbrace{[1\ 1\ \dots\ 1\ 1]}_{1\times16} .* \underbrace{[1\ 1\ \dots\ 1\ 1]}_{1\times16}\right) \cdot \alpha \bullet \left(\underbrace{\begin{bmatrix}1\\ \cdot\\ \cdot\\ 1\\ -1\\ \cdot\\ \cdot\\ -1\end{bmatrix}}_{16\times1} .* \underbrace{\begin{bmatrix}1\\ \cdot\\ \cdot\\ 1\\ 1\\ \cdot\\ \cdot\\ 1\end{bmatrix}}_{16\times1}\right)\right) \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]}$$

$$= \left(\underbrace{[1\cdot1\ \ 1\cdot1\ \dots\ 1\cdot1\ \ 1\cdot1]}_{1\times16} \cdot \alpha \bullet \underbrace{\begin{bmatrix}1\cdot1\\ \cdot\\ \cdot\\ 1\cdot1\\ (-1)\cdot1\\ \cdot\\ \cdot\\ (-1)\cdot1\end{bmatrix}}_{16\times1}\right) \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]} \quad (48)$$

$$= [1\cdot1\cdot1\cdot1 + \dots + 1\cdot1\cdot1\cdot1 + 1\cdot1\cdot(-1)\cdot1 + \dots + 1\cdot1\cdot(-1)\cdot1] \cdot \alpha \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]} \quad (49)$$
$$= 0 \cdot \alpha \cdot \overset{q}{[z_1\ z_2\ \dots\ z_j]}$$

We can verify that the result from equation (48) is null in case of using OBV and the result from equation (45) is always larger than that case. We can consider that the term of equation (39) has worked as an intensification factor for the term represented by equation (28).

Equation (28) is a term from equation (27) for updating weights during the training stage of MLPs. So, we can confirm the influence of target vector type on MLP learning. In case of using OBV as target vector, we can provide a reduced noise propagation contributing into an improved performance of MLPs on pattern recognition.

8 Conclusion

This work presented a mathematical proof to demonstrate the MLP performance improvement by adopting orthogonal bipolar vectors as targets. The mathematical results have shown the effects on noise reduction propagating from layer to layer due to the use of orthogonal bipolar vectors rather than the use of conventional target vectors for MLP learning. We also have verified that the use of orthogonal bipolar vectors provides a better separation of pattern features due to larger Euclidean distance between these vectors. We have concluded that the results can confirm the hypothesis of our work suggesting orthogonal bipolar vectors as expectation values for MLP learning in degraded pattern recognition.

9 Acknowledgments

This work was supported by a grant from PROPP-UFU through the project number 72/2010 and from CAPES through MINTER (UFU and IFTM) master program.

10 References

[1] B.C. Guingo, A. C. G. Thomé, R. J. Rodrigues, Automatic License Plate Recognition of Motor Vehicles using Artificial Neural Networks (in Portuguese), **XVII Congress of teaching and research in transport – ANPET**, Florianópolis (2004).

[2] N. C. L. Abreu, Extraction of speech signals in noisy environments by decomposition os statistically indepent basis functions (in Portuguese), Dissertation – **Federal University Maranhão** (2003).

[3] A. M. Santos, J. M. Seixas, B. B. Pereira, R. A. Medronho, Using Artificial Neural Networks and Logistic Regression in the Prediction of Hepatitis A (in Portuguese), **Brazilian Journal of Epidemiology** (2005); 8(2): 117-26.

[4] X. Wang, C. Chang, F. Du, Achieving a More Robust Neural Network Model for Control of a MR Damper by Signal Sensitivity Analysis, **Neural Computing & Applications**, 13(2002), 330-338.

[5] M. A. Costa, A. P. Braga, B. R. Menezes, Improving Neural Networks Generalization With New Constructive and Pruning Methods**, Journal Of Intelligent & Fuzzy Systems**, 13(2003), 75-83.

[6] C. M. Lee, S. S. Yang, C. L. Ho, Modified back-propagation algorithm applied to decision-feedback equalization, **Iee Proceedings – Vision, Image & Signal Processing**, 153(2006), 805-809.

[7] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, T. Shiose, Improving MLP learning via Orthogonal Bipolar Target Vectors, **Journal of Advanced Computacional Intelligence and Intelligent Informatics**, 9(2005), 580-589.

[8] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, T. Shiose, A New Approach to Improving Math Performance of Artificial Neural Networks (in Portuguese), **VIII Brazilian Symposium on Neural Networks**, (2004).

[9] J. R. G. Manzan, K. Yamanaka, S. Nomura. Improvement in Perfomance of MLP using New target vectors (in Portuguese), **X Brazilian Congress on Computational Intelligence** (2011).

[10] S. Nomura, J. R. G. Manzan, K. Yamanaka. An Experimentation with improved target vectors for MLP in classifying degraded patterns, **Journal of the Brazilian Neural Network Society, Vol. 8, Iss. 4, pp.240 – 252** (2010).

[11] A. Browne, Neural Network Analysis, Architetures, and Applications, **Institute of Physics Pub** (1997).

[12] T. Kohonen, Associative Memory: a System-Theoretical Approach, **Springer-Verlag** (1977).

[13] K. S. Fu, Syntatic Methods in Pattern Recognition, Academic Press (1974).

[14] L. N. Cooper, A Possible Organization of Animal Memory and Learning, Proc. of the Nobel Symposium on Collective Propertiers of Physical Systems, B. Lundquist and S. Lundquist (1973).

[15] R.O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, **Wiley** (1973).

[16] L. Fausset, Fundamentals os Neural Networks: Architecture, Algorithms, and Applications, **Prentice-Hall**, (1994).