

# Improving Iris Recognition through New Target Vectors in MLP Artificial Neural Networks

José Ricardo Gonçalves Manzan, Shiguo Nomura, Keiji Yamanaka, Milena Bueno Pereira Carneiro, and Antônio C. Paschoarelli Veiga

Faculty of Electrical Engineering - Federal University of Uberlândia,  
Av. João Naves de Ávila, 2160, Bloco 3N - Campus Santa Mônica  
CEP: 38400-902 - Uberlândia - MG, Brasil

josericardo@iftm.edu.br, shigueonomura@feelt.ufu.br,  
{keiji,acpveiga}@ufu.br, milenabueno@yahoo.com.br  
<http://www.feelt.ufu.br>

**Abstract.** This paper compares the performance of multilayer perceptron (MLP) networks trained with conventional bipolar target vectors (CBVs) and orthogonal bipolar new target vectors (OBVs) for biometric pattern recognition. The experimental analysis consisted of using biometric patterns from CASIA Iris Image Database developed by Chinese Academy of Sciences - Institute of Automation. The experiments were performed in order to obtain the best recognition rates, leading to the comparison of results from both conventional and new target vectors. The experimental results have shown that MLPs trained with OBVs can better recognize the patterns of iris images than MLPs trained with CBVs.

**Keywords:** Biometric pattern, iris image, conventional bipolar vector, multilayer perceptron, orthogonal bipolar vector, pattern recognition, target vector.

## 1 Introduction

Among the various research fields in computing, computational intelligence has received enough attention on the possibilities of applications to genetic algorithms, fuzzy logic, and artificial neural networks (ANNs).

Research advances in ANNs have been realized from the 80's mainly due to the relevant contributions of expert researchers specialized in this field. However, it is evident that the researchers still are searching for further advances through several studies [1–5].

Several studies in pattern recognition techniques such as statistical approach [6] and connectionist approach [6] have been performed. The connectionist approach has involved ANN techniques providing relevant and promising results in pattern recognition tasks due to their generalization capabilities.

Among the ANN techniques, the MLP models have been widely applied to biometric pattern recognition problems. MLPs have been focused by researchers of this field around the world.

Some studies have been related to MLP topology [2] choice improvements. Also, studies for investigating improvements in the learning algorithm [3] are not rare. Furthermore, interesting results have been published due to a new methodology for choosing the initial synaptic weights [4] and use of signal sensitivity analysis [1].

Searching for mechanisms that can improve the performance of MLPs in biometric pattern recognition tasks, we could not find investigations focused on target vector studies.

Usually, we have the conventional use of bipolar vectors (CBVs) as targets for MLP training and one-per-class classification approach.

This paper proposes an unconventional use of orthogonal bipolar vector (OBV) as targets for training MLP models. We suppose that the use of new target vectors can influence on MLP performance improvement in recognizing biometric patterns. Mathematically, OBVs have advantages of reduced similarities between them.

Preliminary experimental results related to this proposal have been presented by previous works [7–10] and they have shown the effectiveness for the MLP performance improvement. This paper aims to show experimental results of this new approach applied to iris image recognition.

## 1.1 Motivation

It is known that biological neurons can recognize patterns with a high degree of degradation [11]. This ability to differentiate degraded patterns can also be obtained by MLPs trained with appropriate and adjusted conditions. An MLP model can recognize a degraded or biometric pattern even though it has not been presented during the training stage. Therefore, the biometric pattern recognition by MLPs is possible if the models provide a good generalization capability.

There are several proposals searching for the appropriate treatment of input vectors [12] to achieve the expected MLP performance improvement. However, there is no relevant investigation focusing on the treatment of target vectors for MLP learning. Regarding this investigation gap, we have decided to investigate the positive influence of OBVs used as targets on the MLP performance to recognize biometric patterns.

## 1.2 Mathematical Foundation

Analyzing conventional target vectors in terms of inner product between them, we can verify that the product increases according to the size of those vectors. Also, the inner product is high if the angle between two consecutive vectors is small. In other words, if the vectors are almost parallel then their degree of similarity is high.

On the other hand, if the angle between two consecutive vectors is the same as 90 degrees then the inner product between them is null. In this case, the vectors are orthogonal and the Euclidean distance between them is large compared to the conventional vectors.

In terms of similarity, we can realize that OBVs have null similarity, regardless of their sizes. Eq. 1 and eq. 2 represent two possible target vectors and eq. 3 represents the corresponding inner product between them. The Euclidean distance can be calculated by eq. 4.

$$\vec{V}_i = (v_1, v_2, \dots, v_n) \quad (1)$$

$$\vec{W}_i = (w_1, w_2, \dots, w_n) \quad (2)$$

$$\vec{V}_i \bullet \vec{W}_i = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3 + \dots + v_n \cdot w_n \quad (3)$$

$$d_{V,W} = \sqrt{(w_1 - v_1)^2 + (w_2 - v_2)^2 + (w_3 - v_3)^2 + \dots + (w_n - v_n)^2} \quad (4)$$

In terms of Euclidean distance, it is easy to verify that the distance between two CBVs is constant for any size of vectors. But, the distance between two OBVs increases when their sizes increase. Also, the Euclidean distance between OBVs is always larger than the distance between CBVs.

Section 2 presents the experimental procedure. The experimental results are presented in Section 3. In Sections 4 and 5, the discussion of results and conclusion are described.

## 2 Experimental Procedure

This section presents the experimental procedure for MLP training and iris image recognition using three model types. The first type of model is the MLP trained with CBVs, the second type is the model trained with OBVs, and the third one is the model trained with NOVs. The experiments aim to do the hypothesis confirmation for the performance improvement of MLPs trained with OBVs as targets to recognize iris images as biometric patterns.

### 2.1 Target Vectors for MLP Learning

A set of conventional bipolar vectors (CBVs) can be defined as a matrix described by eq. 5. Each row  $i$  of this matrix corresponds to the  $i$ -th CBV containing the component “1” for  $i = j$  and the component “-1” for others.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{for } i = j \\ -1 & \text{for } i \neq j \end{cases} \quad (5)$$

To generate a set of orthogonal bipolar vectors (OBVs), we have based on the theorem and its respective algorithm presented by Fausett [13]. According to

the theorem, the number  $n$  of components for a vector is calculated as  $n = 2^k m$ . The value of  $m$  is the number of components of seed vector  $V$  that starts the algorithm. The value of  $2^k$  is the number of generated orthogonal vectors. The operation  $[V, V]$  denotes the function for concatenation of vector  $V$  with itself producing a vector with double number of components. The steps of the algorithm can be described as follows:

1. Initializing the seed vector:  $V_m(1) = (1, 1, 1, \dots, 1)$  where  $m$  is odd;
2. Concatenating the seed vectors and generating mutually orthogonal vectors:  
 $V_{2m}(1) = [V_m(1), V_m(1)]$  and  $V_{2m}(2) = [V_m(1), -V_m(1)]$ ;
3. Concatenating consecutive orthogonal vectors:  
 $V_{4m}(1) = [V_{2m}(1), V_{2m}(1)]$ ,  $V_{4m}(2) = [V_{2m}(1), -V_{2m}(1)]$ ,  
 $V_{4m}(3) = [V_{2m}(2), V_{2m}(2)]$ ,  $V_{4m}(4) = [V_{2m}(2), -V_{2m}(2)]$ ;
4. Repeating step 3 until generating  $V_n(1), \dots, V_n(2^k)$  as OBVs.

A sample of generated OBVs with eight components can be as follows:

$$\begin{aligned} V_{8m}(1) &= (1, 1, 1, 1, 1, 1, 1, 1), V_{8m}(2) = (1, 1, 1, 1, -1, -1, -1, -1), \\ V_{8m}(3) &= (1, 1, -1, -1, 1, 1, -1, -1), V_{8m}(4) = (1, 1, -1, -1, -1, -1, 1, 1), \\ V_{8m}(5) &= (1, -1, 1, -1, 1, -1, 1, -1), V_{8m}(6) = (1, -1, 1, -1, -1, 1, -1, 1), \\ V_{8m}(7) &= (1, -1, -1, 1, 1, -1, -1, 1), V_{8m}(8) = (1, -1, -1, 1, -1, 1, 1, -1). \end{aligned}$$

In this experimental analysis, we also use the non-orthogonal bipolar vectors (NOVs) that have the same size as the OBVs, but they have non-null inner product. The difference of NOVs in relation to CBVs is only the size. The use of NOVs is a strategy for fair performance comparison with larger sizes of OBVs in relation to CBVs.

If the size of NOVs is large then the corresponding inner product between them is large. Since the inner product represents the degree of similarity between NOVs, the similarity increases while their sizes increase. We are supposing that the similarity between target vectors can cause low performance of MLPs.

## 2.2 MLP Topologies

Cross validation methods were applied to define the MLP topologies for the experiments. So, four MLP topologies (number of input neurons x number of hidden neurons x number of output neurons) were set as follows: 2400 x 200 x 71; 2400 x 200 x 128; 2400 x 800 x 71; and 2400 x 800 x 128.

The number of 71 units in the output layer was defined by the number of subjects corresponding to the iris images from CASIA repository. So, 71 CBVs with 71 components were generated to represent the target vectors for the usual pattern recognition experiments.

In case of using OBVs as target vectors, since OBVs have been generated with the sizes same as 2, 4, 8, 16, 32, 64, 128 and so on, we have chosen the number of 128 units in the output layer. Therefore, we have generated OBVs with 128 components for the experiments regarding the proposed approach. Based on

the size of OBV, we have generated NOV's with 128 components for the third type of MLP model.

### 2.3 Experimental Data

The training data consisted of human iris images from the Chinese Academy of Sciences - Institute of Automation database called CASIA [14]. The database contains iris images from 108 subjects and 71 of them consist of complete data with seven images. For this reason, we have adopted the data corresponding to 71 subjects. Randomly chosen four images composed the training set and other three ones composed the test set. According to CASIA repository, these images were taken by the use of infrared light to get the iris features with enough contrast for biometric pattern recognition.

The iris image processing takes place in a few steps. The first step is the location of the iris' region in the image that is performed using the circular Hough transformed. Then the iris' region, which has a ring shape, is normalized to be represented by a rectangular matrix. Finally, the extraction of the iris' features is done. In this paper, the iris' features were extracted convoluting the normalized image with the so-called log Gabor filter. Filtration gives rise to complex coefficients whose phases are quantized to one of the four quadrants of the complex plane. Each quadrant is referenced by two bits, and a binary template is created [15–17]. For each image there are 8640 pixels arranged in 18 concentric circles each containing 480 pixels.

In this work, we extracted only 5 concentric circles of the iris, eliminating the interference of the eyelids and reducing the training effort of the MLP. Thus, each training pattern corresponds to a set of  $5 \times 480 = 2400$  pixels. The white pixels were represented by value  $-1$  and the black pixels by  $1$ . So, the training vectors were constructed to represent single lines containing 2400 pixels that connect the points from the innermost circle to the outermost one of the iris.

### 2.4 MLP Training Stage

The experimental simulations were performed using the *traingdx* toolbox of Matlab software version R2008. The *traingdx* toolbox adopts the momentum and adaptive learning rate for MLP learning. For this reason, we can get a more rapid convergence assuring consistent results. As the toolbox uses randomly chosen initial synaptic weights, we have slightly different final weights for different simulations. So, we have performed three different simulations per each set of training parameters to assure a better representation of the experimental results shown in Tables 1 and 2.

The adopted initial learning rates were 0.1 and 0.3, since *traingdx* works with adaptive learning rates. A tolerance for error during the training epochs was adopted as stopping criterion.

The equations for calculating the mean squared error for using the stopping criterion are as follows:

- $E_p = \frac{1}{2} \sum_{j=1}^{N_s} (d_j - y_j)^2$  where  $E_p$  is the squared error of a pattern  $p$ ;  $N_s$  is the number of output neurons;  $d_j$  is the desired output for neuron  $j$ ; and  $y_j$  is the net output for neuron  $j$ ;
- $E_m = \frac{1}{N} \sum_{p=1}^N E_p$  where  $E_m$  is the mean squared error of all patterns for each epoch;  $N$  is the number of patterns;  $E_p$  is the squared error for pattern  $p$ ;

The training has been finalized when the stopping condition  $E_m < \varepsilon$  is satisfied. The value of  $\varepsilon$  is the initialized tolerance for error during the training process.

The simulations were performed on a computer's processor type INTEL(R) CORE i5-2410TM, 2.30 GHz and memory (RAM) of 4 GB.

### 3 Experimental Results

Tables 1 and 2 show the results from the MLP training and test using the Matlab toolbox. Tolerance in the tables means the maximum mean squared error (represented by  $\varepsilon$  in Section 2.4) to be achieved by the MLP training for satisfying the stopping condition. Each table provides the tolerance for error, the number of epochs for the training, the pattern recognition rate (performance), the simulation number, and the initial learning rate  $\alpha$ .

In Tables 1 and 2, we can verify that results from 3 simulations for a set of parameters are presented and these results present small standard deviation between them. The small difference between those results is due to the random initialization of synaptic weights for the MLP training with Matlab toolbox.

Table 1 compares the performance of three different types of target vectors with 200 hidden neurons and initial learning rate of 0.1 or 0.3. In Table 2, the results refer to the adoption of 800 hidden neurons. The bold and underlined value represents the best obtained recognition rate for a set of simulations.

Fig. 1 represents the best simulation results for each type of target vector setting up MLP with 200 hidden neurons and initial learning rate of 0.1. Fig. 2 corresponds to the results from simulations of MLPs with 200 hidden neurons and initial learning rate of 0.3. The best simulation results corresponding to experiments using MLPs with 800 hidden neurons and initial learning rate of 0.1 are presented in Fig. 3. Fig. 4 presents the best experimental results for each target vector type in MLP with 800 hidden neurons and initial learning rate of 0.3.

In another application [7], the recognition rate of characters from license plate degraded images with the use of OBVs as targets has increased around 5.4% comparing with the results using conventional target vectors. The work applied to the recognition of handwritten digits [9] has presented a variation on the recognition rate by 2% when OBVs are used as target vectors for MLP learning. The OBV experimental results for application to the digits extracted from license plate degraded images [10] have presented an increase of 8% on the MLP performance. All these results have strengthened the viability of using OBVs as target vectors for MLP in pattern recognition.

**Table 1.** Comparison of MLP performances with 200 hidden neurons for various target vectors and initial learning rates (represented by  $\alpha$ )

		Tolerance	1.00E-02	1.00E-03	1.00E-04	1.00E-05	1.00E-06	
$\alpha = 0.1$	OBV(128)	Simulation	1 Epochs	121	335	567	408	560
			1 Performance (%)	93.90	94.37	94.37	95.31	94.37
		Simulation	2 Epochs	125	203	227	707	370
			2 Performance (%)	93.90	92.96	<b>95.77</b>	93.90	94.84
		Simulation	3 Epochs	126	177	201	337	482
			3 Performance (%)	95.31	93.90	93.90	94.37	94.37
	NOV(128)	Simulation	1 Epochs	81	128	215	189	239
			1 Performance (%)	47.89	74.65	86.85	84.04	86.85
		Simulation	2 Epochs	82	153	194	187	249
			2 Performance (%)	46.48	78.40	84.04	82.16	<b>88.26</b>
		Simulation	3 Epochs	87	127	243	236	313
			3 Performance (%)	40.85	68.54	85.92	81.69	83.57
	CBV(71)	Simulation	1 Epochs	82	146	171	184	248
			1 Performance (%)	56.34	74.65	83.57	82.16	86.39
		Simulation	2 Epochs	82	124	173	190	235
			2 Performance (%)	55.67	70.89	84.98	81.69	<b>88.73</b>
		Simulation	3 Epochs	81	137	138	183	258
			3 Performance (%)	56.81	70.42	76.06	86.39	86.39
$\alpha = 0.3$	OBV(128)	Simulation	1 Epochs	105	188	598	563	359
			1 Performance (%)	<b>95.77</b>	94.37	93.90	93.90	94.37
		Simulation	2 Epochs	100	149	535	262	435
			2 Performance (%)	94.84	94.37	92.96	92.96	94.84
		Simulation	3 Epochs	102	202	450	470	819
			3 Performance (%)	94.37	94.37	95.31	93.43	93.84
	NOV(128)	Simulation	1 Epochs	74	148	144	213	219
			1 Performance (%)	46.48	79.34	73.71	86.85	<b>89.67</b>
		Simulation	2 Epochs	74	148	144	213	219
			2 Performance (%)	47.89	78.40	86.39	84.98	86.85
		Simulation	3 Epochs	59	125	159	343	234
			3 Performance (%)	46.01	75.59	79.34	88.26	89.20
	CBV(71)	Simulation	1 Epochs	76	122	115	174	213
			1 Performance (%)	65.26	78.40	79.81	84.51	88.26
		Simulation	2 Epochs	79	132	115	163	294
			2 Performance (%)	63.85	82.63	76.53	83.10	86.85
		Simulation	3 Epochs	63	159	131	163	262
			3 Performance (%)	59.62	74.65	81.69	86.39	<b>89.20</b>

**Table 2.** Comparison of MLP performances with 800 hidden neurons for various target vectors and initial learning rates (represented by  $\alpha$ )

		Tolerance	1.00E-02	1.00E-03	1.00E-04	1.00E-05	1.00E-06	
$\alpha = 0.1$	OBV(128)	Simulation	1 Epochs	95	136	179	225	272
			1 Performance (%)	94.37	<b>96.24</b>	95.31	95.77	95.31
		Simulation	2 Epochs	98	133	220	226	271
			2 Performance (%)	94.37	93.90	<b>96.24</b>	95.31	95.31
		Simulation	3 Epochs	99	141	203	225	271
			3 Performance (%)	93.43	95.31	95.31	95.31	95.77
	NOV(128)	Simulation	1 Epochs	95	142	185	224	307
			1 Performance (%)	44.60	78.40	80.75	86.39	87.32
		Simulation	2 Epochs	86	142	182	179	212
			2 Performance (%)	47.42	73.71	77.46	82.63	<b>88.26</b>
		Simulation	3 Epochs	91	139	234	273	230
			3 Performance (%)	53.05	75.12	83.57	86.85	84.98
	CBV(71)	Simulation	1 Epochs	141	153	189	241	202
			1 Performance (%)	66.20	84.04	83.10	90.14	<b>91.55</b>
		Simulation	2 Epochs	140	130	162	168	231
			2 Performance (%)	65.73	77.46	86.85	84.98	86.85
		Simulation	3 Epochs	140	116	159	159	208
			3 Performance (%)	65.73	76.53	84.04	84.51	85.45
$\alpha = 0.3$	OBV(128)	Simulation	1 Epochs	76	116	157	202	249
			1 Performance (%)	94.37	94.84	94.84	95.31	95.31
		Simulation	2 Epochs	77	111	157	235	265
			2 Performance (%)	94.37	94.84	94.84	96.24	95.77
		Simulation	3 Epochs	75	122	157	203	249
			3 Performance (%)	94.37	94.37	94.37	93.90	<b>96.71</b>
	NOV(128)	Simulation	1 Epochs	109	157	342	619	732
			1 Performance (%)	58.69	84.51	86.39	90.14	<b>91.55</b>
		Simulation	2 Epochs	96	157	390	424	310
			2 Performance (%)	58.69	84.98	90.14	90.61	87.80
		Simulation	3 Epochs	106	166	361	350	569
			3 Performance (%)	58.69	86.39	89.20	89.20	90.14
	CBV(71)	Simulation	1 Epochs	49	147	121	163	275
			1 Performance (%)	47.42	69.01	70.42	76.53	<b>85.45</b>
		Simulation	2 Epochs	45	89	100	170	261
			2 Performance (%)	42.72	68.08	78.40	77.47	84.51
		Simulation	3 Epochs	44	82	95	152	213
			3 Performance (%)	43.66	68.54	70.89	82.63	84.51



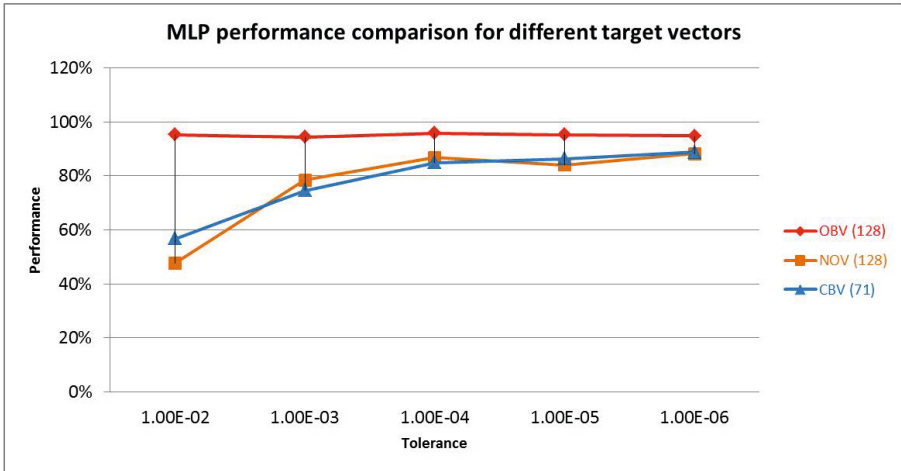


Fig. 1. 200 hidden neurons and initial learning rate of 0.1

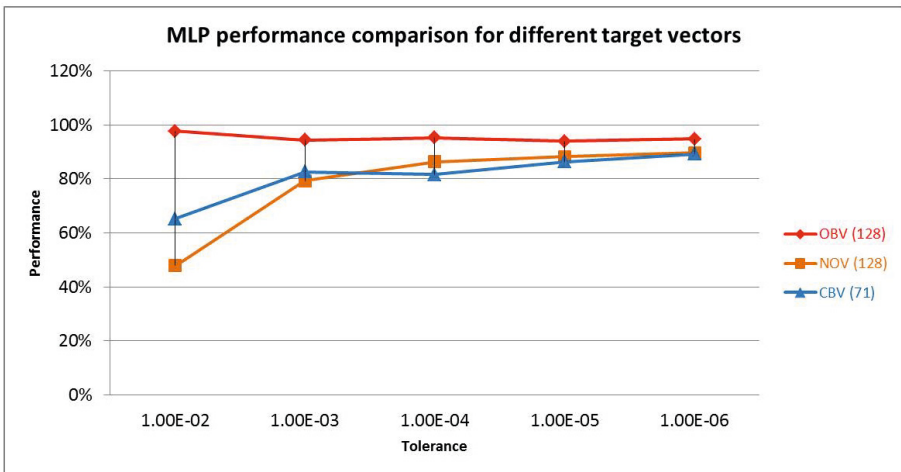


Fig. 2. 200 hidden neurons and initial learning rate of 0.3

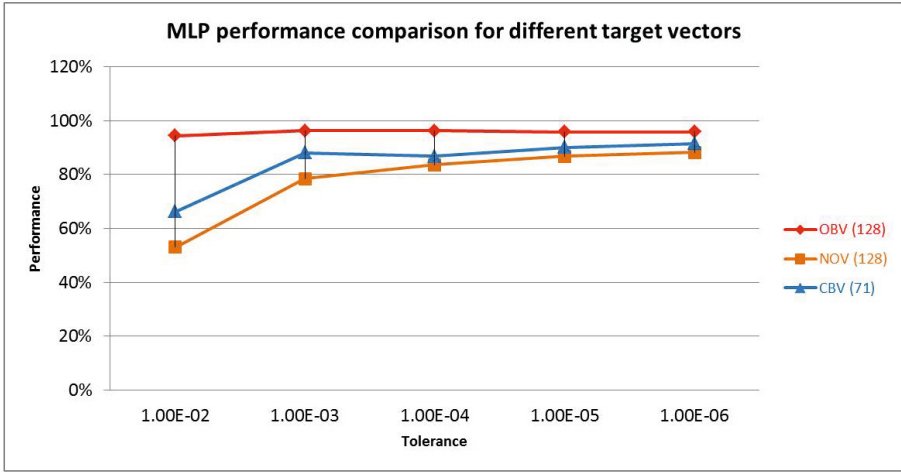


Fig. 3. 800 hidden neurons and initial learning rate of 0.1

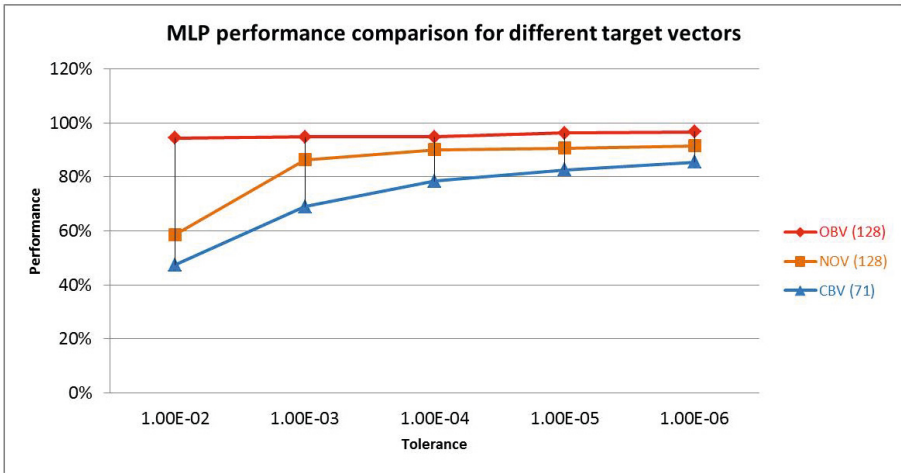


Fig. 4. 800 hidden neurons and initial learning rate of 0.3

## 4 Discussion

The results have shown that the use of OBVs as targets improves the performance on recognizing biometric patterns represented by iris images in all the cases.

We have verified that initial learning rates have caused some differences in MLP performance. Among others, the use of 800 neurons provided the best recognition rate which was 96.71% with initial learning rate of 0.3. Also, we have realized that recognition rates were higher than 93% using OBVs and rates from 47% to 66% using other vectors for MLP training in non-rigid tolerance (1.00E-02).

Furthermore, a large number of epochs were necessary to achieve the rates around 90% in case of using CBVs and NOVs as target vectors.

## 5 Conclusion

This paper proposed the unconventional use of OBVs as target vectors for MLP learning to improve the network performance on biometric pattern recognition. For this purpose, iris images from CASIA database were used to do MLP performance comparisons among CBVs, NOVs and OBVs as target vectors.

The experimental results showed high recognition rates using OBVs rather than other vectors. Also, the simulations showed that the use of OBVs provides smaller number of epochs for MLP learning compared to the use of other vectors. Consequently, the computational load can be reduced with the use of OBVs for MLP learning.

Another advantage of OBVs is related to the null inner product or null similarity between them while the inner product between conventional target vectors increases when the number of biometric patterns increases.

Therefore, we have concluded that the unconventional use of OBVs as target vectors for MLP learning and biometric pattern (represented by iris images) recognition is promising. The Euclidean distance increase and similarity reduction between vectors provided by the use of OBVs can support the obtained results.

**Acknowledgements.** We thank the PROPP in the Federal University of Uberlândia for supporting this work through the project number 72/2010. We also thank CAPES for supporting the Inter-institutional Master's program organized by the Federal University of Uberlândia and the Federal Institute of Triângulo Mineiro.

## References

1. Wang, X., Chang, C., Du, F.: Achieving a More Robust Neural Network Model for Control of a MR Damper by Signal Sensitivity Analysis. *Neu. Comp. & App.* 13, 330–338 (2002)
2. Costa, M.A., Braga, A.P., Menezes, B.R.: Improving Neural Networks Generalization With New Constructive and Pruning Methods. *J. Int. & Fuzzy Syst.* 13, 75–83 (2003)
3. Lee, C.M., Yang, S.S., Ho, C.L.: Modified Back-propagation Algorithm Applied to Decision-feedback Equalization. *IEE Proceedings - Vis., Ima. & Sig. Proc.* 153, 805–809 (2006)

4. Kim, D.: Improving Prediction Performance of Neural Networks in Pattern Classification. *Int. J. Comp. Math.* 82(4), 391–399 (2005)
5. Chen, L., Pung, H.K.: Convergence Analysis of Convex Incremental Neural Networks. *Annals of Mathematics and Artificial Intelligence* 52, 67–80 (2008)
6. Browne, A.: *Neural Network Analysis, Architectures, and Applications*. Institute of Physics Pub., Philadelphia (1997)
7. Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., Shiose, T.: Improving MLP Learning Via Orthogonal Bipolar Target Vectors. *J. Adv. Comp. Int. and Int. Inf.* 9, 580–589 (2005)
8. Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., Shiose, T.: A New Approach to Improving Math Performance of Artificial Neural Networks (in Portuguese). In: VIII Brazilian Symposium on Neural Networks, São Luís (2004)
9. Manzan, J.R.G., Yamanaka, K., Nomura, S.: Improvement in Performance of MLP Using New Target Vectors (in Portuguese). In: X Brazilian Congress on Computational Intelligence, Fortaleza (2011)
10. Nomura, S., Manzan, J.R.G., Yamanaka, K.: An Experimentation With Improved Target Vectors for MLP in Classifying Degraded Patterns. *Learning and Nonlinear Models* 8(4), 240–252 (2010)
11. Cooper, L.N.: A Possible Organization of Animal Memory and Learning. In: Lundquist, B., Lundquist, S. (eds.) *Nobel Symposium on Collective Properties of Physical Systems*, Sweden, pp. 62–84 (1973)
12. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. Wiley, New York (1973)
13. Fausset, L.: *Fundamentals of Neural Networks: Architecture, Algorithms, and Applications*. Prentice-Hall (1994)
14. Chinese Academy of Sciences - Institute of Automation, Database of 756 Greyscale Eye Images, <http://www.cbsr.ia.ac.cn/IrisDatabase.html>
15. Daugman, J.: High Confidence Visual Recognition of Person by a Test of Statistical Independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(11), 1148–1161 (1993)
16. Negin, M., Chmielewski Jr., T.A., Salganicoff, M., von Seelen, U.M., Venetainer, P.L., Zhang, G.G.: An Iris Biometric System for Public and Personal Use. *IEEE Computer Society* 33, 70–75 (2000)
17. Carneiro, M.B.P., Veiga, A.C.P.: Application of Genetic Algorithms to Improve the Reliability of An Iris Recognition System. In: *IEEE Workshop on Machine Learning for Signal Processing, Mystic*, pp. 159–164 (2005)