2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE).
Mexico City, Mexico November 9-11, 2022

Metaheuristic Method for Dimensionality Reduction Tasks

A.Anzueto-Ríos
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
alvaro.anzueto@cinvestav.mx

F. Gómez-Castañeda
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
fgomez@cinvestav.mx

L.M. Flores-Nava
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
lmflores@cinvestav.mx

J.A. Moreno-Cadenas
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
jmoreno@cinvestav.mx

2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) | 978-1-6654-5508-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCE56709.2022.9975991

***Abstract*—This work demonstrates an alternative training approach, which can be fairly compared with the learning procedures for autoencoders within dimensionality reduction schemes. Therefore, this work's metaheuristic artificial bee colony optimization method trains an encoder-type neural network, which can then compute orthogonal features straight away. Experimental sessions of this method lead us to show competitive results based on public domain datasets.**

***Index Terms*—Autoencoder, Dimensionality Reduction, Neural Network, Metaheuristics, Artificial Bee Colony algorithm**

I. Introduction

Nowadays, machine learning techniques are being adopted in the way of intelligent systems. This fact is possible due to artificial neural networks technology, where the autoencoder architecture, among others, has a special role [1]. This network, in its fundamental manner, is a one-hidden layer system. The connections between this hidden layer with the input and output layers define the encoder and decoder sections, respectively. The encoder reduces the dimension of the input vectors. Additionally, the autoencoders minimize the reconstruction error in the training phase via evaluating either the Euclidian distance or the Frobenius norm, i.e., their suitable objective functions. The first objective function refers to updating connections; meanwhile, instantaneous samples occur, and the second one attends to batch processes. Moreover, adding an orthogonalization term to the objective function can improve the performance. In this work, we show that a neural network, which works as the encoder section of one autoencoder, reduces the dimension of input vectors and draws useful orthogonal information. Therefore, the training phase uses the metaheuristic artificial bee colony algorithm, replacing efficiently any other method.

A set of experimental sessions based on Matlab code for handwritten numbers and cropland images demonstrates the comparative capabilities of our dimensionality reduction method.

II. Dimensionality Reduction with Autoencoder

The primary operational function of one autoencoder is reducing the dimension of input vectors [2]. The fundamental features of the reduced vectors in this task can be observed with linear algebra analysis of the encoder's connection matrices and the decoder's [3]. Likewise, analysis with an information-theoretic basis further improves the performance of autoencoders [4], [5]. Fig. 1 depicts the architecture of one autoencoder, where the matrices W_{EN} and W_{DE} contain the connections of the encoder and decoder sections. The hidden layer is also called Bottleneck. The input and output vectors are represented by q_{IN} and q_{OUT}, respectively. The vector r_H carries the information at the hidden layer. The dimension of W_{EN} and W_{DE} is $m \times n$ and $n \times m$, respectively. Both q_{IN} and q_{OUT} have dimension n; and, the dimension of r_H is m. Moreover, the neurons in the hidden and output layers can work with sigmoidal or linear transfer functions. In the training phase, either loss function (1) or (2) should be minimized; where (1) and (2) refer to instantaneous and batch processes, respectively.

$$L_{F1} = \|q_{OUT} - q_{IN}\|^2 \tag{1}$$

$$L_{F2} = \|Q_{OUT} - Q_{IN}\|_F^2 \tag{2}$$

Where: $\|\cdot\|^2$ and $\|\cdot\|_F^2$ mean the Euclidian and Frobenius norms, respectively. Q_{OUT} and Q_{IN} are output and input matrices, respectively whose dimension is $n \times s$; where s is the number of samples contained in a lot of a batch process. Generally speaking, the autoencoder is an efficient vehicle for remarking particular features of the input data. This functionality is observed via adding one or more new loss terms to the primary loss function. In the context of classification, clustering or recognition problems [6]–[8], the orthogonal features provide a notable capability in the process of separation of classes, which in combination of shallow or deep stages complete the detection of patterns or objects. Therefore, the loss function that includes orthogonality in the autoencoder is given by (3).

$$L_{F3} = \|Q_{OUT} - Q_{IN}\|_F^2 + \lambda \left\|R^T R - I\right\|_F^2 \quad (3)$$

Where: R^T is a matrix whose entries belong to the set of hidden vectors created in a batch process, R^T is the transpose of the matrix R, and I is the identity matrix. R has the dimension $s \times m$. A lambda factor is a weighting number of the orthogonalization term.

III. Dimensionality Reduction with Metaheuristics

In the following lines, we introduce a metaheuristic method to realize the dimensionality reduction task. The encoder section of the autoencoder is used alone for this method, as shown in Fig. 2, where the neurons of the hidden layer become the output neurons and have linear transfer functions. The meaning and dimensions of W_{EN}, q_{IN} and r_H remain. The encoder section should minimize the loss function L_{F4} in the training phase in batch mode, approaching the orthogonality in the set $\{r_{H1}, r_{H2}, \dots, r_{HS}\}$. L_{F4} is presented in (4), whose matrices have the same description as in (3).

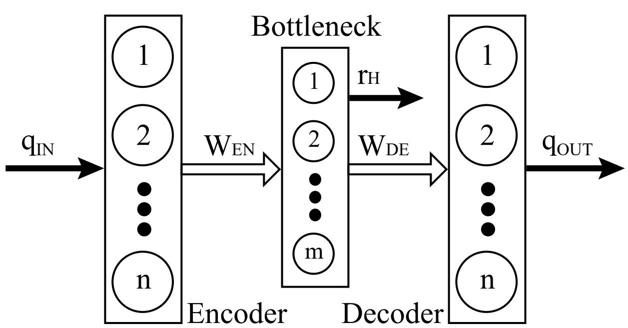

Fig. 1. Architecture of the Autoencoder.

$$L_{F4} = \left\|R^T R - I\right\|_F^2 \quad (4)$$

The minimization of (4) in this work uses the metaheuristic artificial bee colony algorithm (ABC algorithm), which is an efficient optimization method for solving a wide range of complex engineering problems [9]. This technique is a pioneer nature-inspired algorithm in front of many new ones [10]. The ABC algorithm searches for possible solutions using three kinds of bees namely: employed, onlookers, and scouts. The pseudocode of the activities of the bees is available elsewhere, e.g., in [11].

In Algorithm 1 the instructions named Employed Bees' labor, Onlooker Bees' labor, and Scout Bees' labor follow the Algorithms 2–4, respectively, described in [11]. An employed bee randomly selects one food source x_k from the current

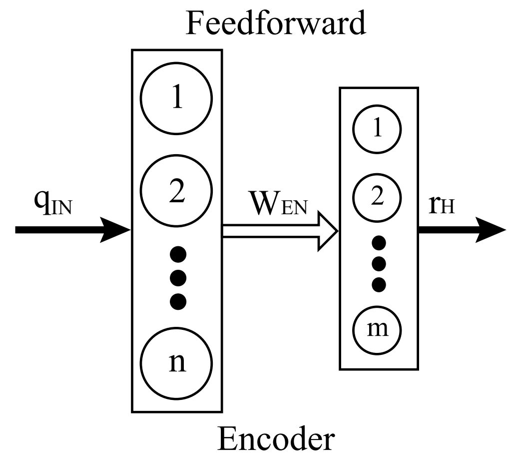

Fig. 2. Architecture of the Encoder Used in This work.

population and chooses a random dimension index j. The new food source is obtained by applying (5).

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \quad (5)$$

where i is the solution currently being exploited, k is a randomly chosen neighbor solution, and ϕ_{ij} is randomly chosen from $[-1, 1]$ drawn from the uniform distribution. An employed bee unloads the nectar and then gives information to onlookers about the quality and the location of her source. High-quality solutions have a high chance to be selected but the solutions with low quality can also be selected by the onlookers [11]. The probability of each solution (p_i) can be calculated proportionally to its fitness value using (6).

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (6)$$

Algorithm 1 ABC Pseudocode.

```
Data: set control parameter values;
SN: Number of Food Sources,
MCN: Maximum Counter Number,
limit : For deciding either a source is exhausted
Initialize and evaluate the food source locations
counter = 1
while counter < MCN do
    Employed Bees' labor (see Algorithm 2)
    Onlooker Bees' labor (see Algorithm 3)
    Memorize the Best Solution
    Scout Bees' labor (see Algorithm 4)
    counter + +
end while
```

IV. Experimental Sessions

This section deals with training the encoder network in Fig. 2 with the ABC algorithm. The underlying dimensionality reduction task is demonstrated via experimental sessions using two benchmarking data sets, which come from [12] and [13].

Algorithm 2 Employed Bees' Labor Pseudocode (Adapted from [11]).

Data: food source population;
Begin
  for $foodsources < x_i$ **do**
    new solution $x' \leftarrow produced\ by\ Equation(5)$
    $f(x') \leftarrow evaluated\ new\ solution$
    if $f(x') < f(x_i)$ **then**
      $x_i \leftarrow x'$
      $exploit(x_i) \leftarrow 0$
    else
      $exploit(x_i) \leftarrow exploit(x_i) + 1$
    end if
  end for
End

Algorithm 3 Onlooker Bees' Labor Pseudocode (Adapted from [11]).

Data: food source population; Probability of each solution
Begin
  for $foodsourcesx_i$ **do**
    $p_i \leftarrow assign\ probability\ by\ Equation(6)$
  end for
  $i \leftarrow 0$
  $t \leftarrow 0$
  while $t < SN$ **do**
    $r \leftarrow rand(0, 1)$
    if $r < p(i)$ **then**
      $t \leftarrow t + 1$
      $x' \leftarrow a\ new\ solution\ by\ Equation(5)$
      $f(x') \leftarrow evaluate\ new\ solution$
      if $f(x') < f(x_i)$ **then**
        $x_i \leftarrow x'$
        $exploit(x_i) \leftarrow 0$
      else
        $exploit(x_i) \leftarrow exploit(x_i) + 1$
      end if
    end if
    $i \leftarrow (i + 1) \mod (SN - 1)$
  end while
End

Algorithm 4 Scout Bees' Labor Pseudocode (Adapted from [11]).

Data: food source population; Exploitation Counters
Begin
  $si = i : exploit(i) = \max(exploit)$
  if $exploit(si) > limit$ **then**
    $x_{si} \leftarrow random\ solution\ by\ Equation(5)$
    $exploit(si) \leftarrow 0$
  end if
End

The first one is a collection of handwritten numbers extracted from the MNIST database, 5 groups have been generated for analysis.

The second is a collection of data that comes from images of cropland with radar sensors; seven crop type classes exist for this data set as follows: 1—Corn; 2—Peas; 3—Canola; 4—Soybeans; 5—Oats; 6—Wheat; and 7—Broad-leaf. Table I summarizes the parameters of the two database.

TABLE I
DATABASE

	Classes	Feature	Dimension
Handwritten Numbers	3 (0, 1, 2, numbers)	784	900 × 784
	3 (3, 4, 5, numbers)	784	900 × 784
	2 (6, 9, numbers)	784	600 × 784
	2 (7, 8, numbers)	784	600 × 784
	10 (all numbers)	784	3000 × 784
Cropland	7	145	5834 × 148

In order to develop the dimensionality reduction process a neural architecture with three final neurons and orthogonal learning has been built, such as the one presented in Fig. 2. The ABC algorithm has been applied as the search method for the parameters of the synaptic weight matrix, W_{EN}. The initial parameters of the ABC algorithm are listed in Table II. The optimization function applied to the search process of the bees is presented in (4). Table III shows the performance of the ABC algorithm.

TABLE II
PARAMETERS OF ABC ALGORITHM

ABC algorithm	Parameters
Number of Bees	100
Number of Food Source	50
Number of Trial Limits	50
Iterations	5000
Lower Limit	-1
Upper Limit	1
$lambdafactor$	1
Output Neuron	3

Figs. 3−7 contain the dimensional 3D graphs of the data set formed with the MNIST database. The axes are related to each of the neurons in the output layer and are labeled as Dimension 1, 2, and 3, respectively. Each sample is analyzed and assigned a color for the class to which it belongs. Its position inside the graph corresponds to the numerical value obtained from the three neurons considered in the output layer of the proposed architecture.

Both in Fig. 3 and Fig. 4, the results of the separability of three classes are presented; it is possible to notice that three defined groupings are generated, demonstrating the excellent performance of the proposal of considering learning a data output of orthogonal type. Similar behavior is presented in the distributions of Fig. 5 and Fig. 6; the separability between the classes is notable. Fig. 7 shows the total allocation of the ten types when processing the complete MNIST base.

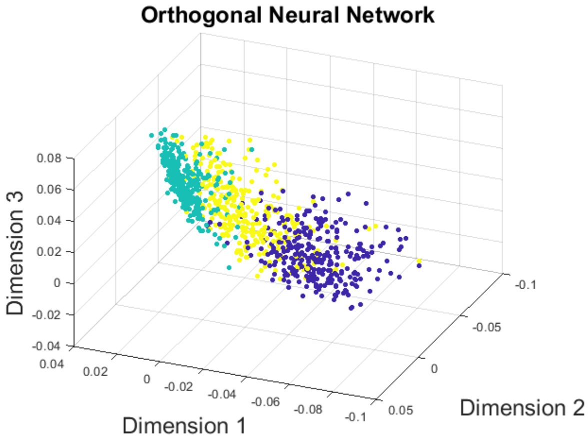

Fig. 3. Group 0, 1, 2.

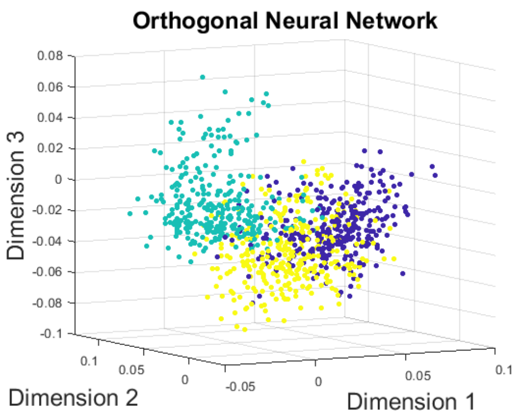

Fig. 4. Group 3, 4, 5.

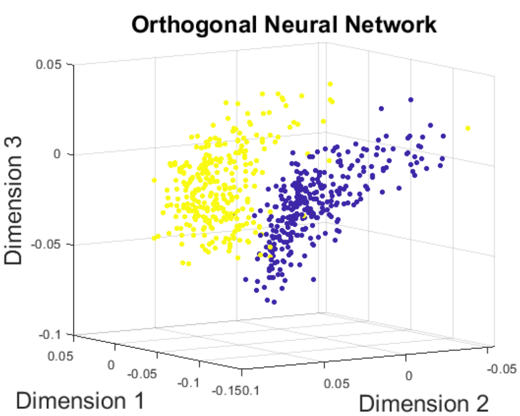

Fig. 5. Group 6, 9.

TABLE III
PERFORMANCE OF THE ABC ALGORITHM

ABC algorithm ERROR	Time 5000 Iterations
0.0025	9.168 min.
0.0118	9.171 min.
0.0147	6.873 min.
0.03422	6.833 min.
0.0957	15.569 min.
0.0037	16.594 min.

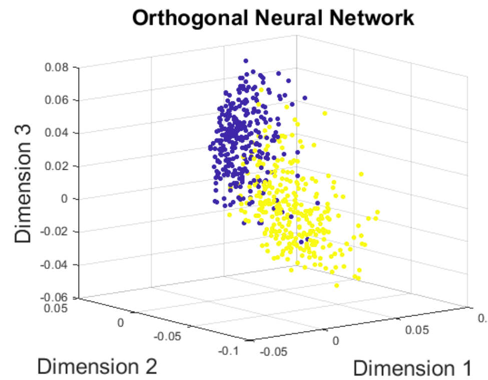

Fig. 6. Group 7, 8.

As a final test, the database about cropland is analyzed. For this database, seven groups with 148 traits are considered, and the dimensionality reduction process is carried out by applying the learning process of the proposed neural network. The graph in Fig. 8 shows the result of the class separability when considering only three output features. With this, it is possible to demonstrate the excellent performance of the dimensionality reduction task in the data.

From these experimental results, we glimpse this method as a rapid manner to perform dimensionality reduction tasks, belonging to a process of classifying data. The whole classification model might also be metaheuristic, which is part of our ongoing research framework.

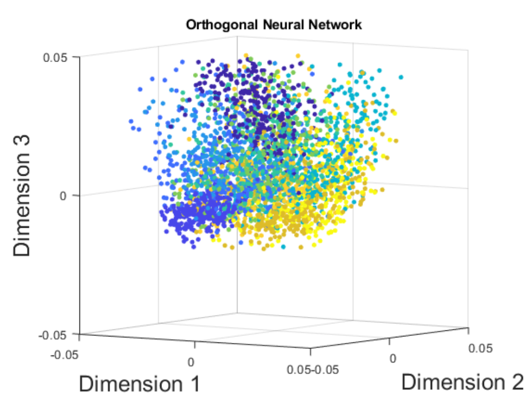

Fig. 7. All Numbers

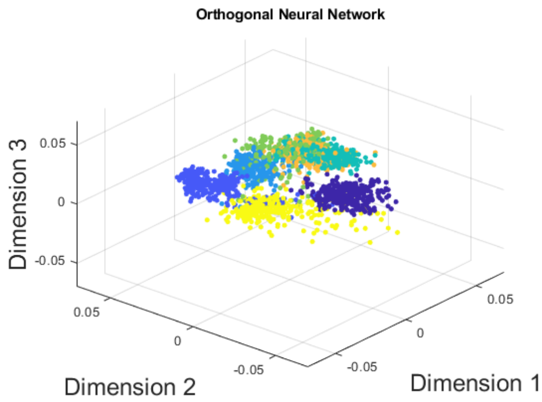

Fig. 8. Cropland Database

V. Conclusion

At last, the main points to remark that come from the experimental results are: 1) the usage of the ABC algorithm, which replaces the optimization methods in machine learning, and 2) the minimal architecture of the encoder type network, which is attractive for hardware implementation, 3) the linear transfer function of the neurons, that reduces, even more, their realization complexity, 4) the underlying potential of metaheuristics for training, 5) the usage of the original orthogonalization term in the original autoencoder model as the objective function for the ABC algorithm.

References

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[3] C. C. Aggarwal *et al.*, "Neural networks and deep learning," *Springer*, vol. 10, pp. 978–3, 2018.

[4] E. Santana, M. Emigh, and J. C. Principe, "Information theoretic-learning auto-encoder," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3296–3301.

[5] S. Lee and J. Jo, "Information flows of diverse autoencoders," *Entropy*, vol. 23, no. 7, p. 862, 2021.

[6] V. Ayma, V. Ayma, and J. Gutierrez, "Dimensionality reduction via an orthogonal autoencoder approach for hyperspectral image classification," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 357–362, 2020.

[7] W. Wang, D. Yang, F. Chen, Y. Pang, S. Huang, and Y. Ge, "Clustering with orthogonal autoencoder," *IEEE Access*, vol. 7, pp. 62 421–62 432, 2019.

[8] W. Shi, Y. Gong, D. Cheng, X. Tao, and N. Zheng, "Entropy and orthogonality based deep discriminative feature learning for object recognition," *Pattern Recognition*, vol. 81, pp. 71–80, 2018.

[9] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," in *International fuzzy systems association world congress*. Springer, 2007, pp. 789–798.

[10] V. C. SS and A. HS, "Nature inspired meta heuristic algorithms for optimization problems," *Computing*, vol. 104, no. 2, pp. 251–269, 2022.

[11] F. Bennis and R. K. Bhattacharjya, *Nature-Inspired Methods for Metaheuristics Optimization: Algorithms and Applications in Science and Engineering*. Springer, 2020, vol. 16.

[12] Y. LeCun, "Handwritten numbers - mnist data set," available online: http://yann.lecun.com/exdb/mnist/ (accessed on 18 May 2022).

[13] U. of California, "Fused optical-radar data set," available online: https://archive.ics.uci.edu/ml/datasets/Crop+mapping+using+fused+optical-radar+data+set (accessed on 26 May 2022).