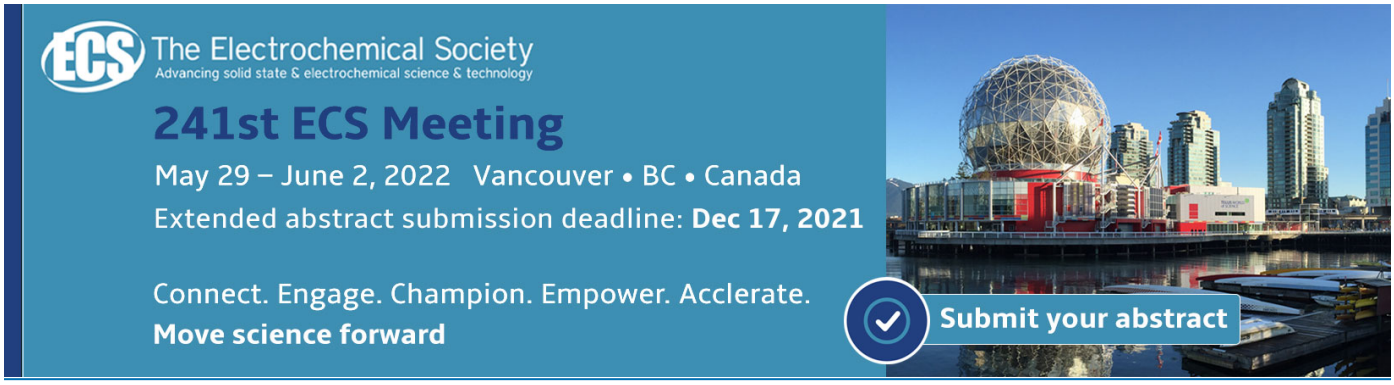
Deep learning architecture for the recursive patterns recognition model



IOP Conf. Series: Journal of Physics: Conf. Series **1126** (2018) 012035 doi:10.1088/1742-6596/1126/1/012035

Deep learning architecture for the recursive patterns recognition model

E Puerto[1], **J Aguilar**[2], **J Reyes**[3] **and D Sarkar**[4]
[1] Grupo de Investigación GIDIS, Universidad Francisco de Paula Santander, San José de Cúcuta, Colombia
[2] Grupo de Investigación CEMISID, Universidad de Los Andes, Mérida, Venezuela
[3] Laboratorio de Prototipos, Universidad UNET, San Cristóbal, Venezuela
[4] Department of Computer Science, University of Miami, Miami, USA

E-mail: eduardpuerto@gmail.com, aguilar@ula.ve

Abstract: In this work, we propose a deep learning approach for the recursive pattern recognition model, called AR2P (for its acronym in Spanish: "Algoritmo Recursivo de Reconocimiento de Patrones"), by extending its supervised learning capability towards a semi-supervised learning scheme. The deep learning architecture is composed of three phases: the first one, called discovery phase, discovers the atomic descriptors. The second one, called aggregation phase, creates a feature hierarchy (merge of descriptors) from atomic descriptors. Finally, the classification phase carries out the classification of the inputs based on the feature hierarchy. The last phase uses a supervised learning approach, while the first two follow an unsupervised learning approach. In this paper is tested the performance of the proposed model, using a dataset from the UCI Machine Learning Repository.

1. Introduction

In the context of machine learning, "Deep Learning" (DL) refers to the automatic knowledge acquisition, for the classification of patterns [1]. The great advantage of DL is that it does not require a definition "by hand" of the characteristics that identify the patterns, but that these characteristics are automatically generated from raw data [2]. Currently, there are a large number of artificial neural networks for pattern classification based on Deep Learning, such as: Convolutional Neural Network (CNN) [3], Deep Boltzmann Machine (DBM) [4], short and long memory (LSTM) [5], Deep Belief Networks (DBN) [6], among others. These models use combinations of supervised and unsupervised learning, at different levels.

On the other hand, AR2P is a pattern recognition algorithm based on the systematic functioning of the human brain [7-9]. Different aspects characterize AR2P, first its modularity and recursively (i.e., its hierarchical modular network topology). Also, the neurophysiological processes subjacent in AR2P are related to pattern recognition and learning capabilities [10]. Both processes are based on the principle of hierarchical organization of the brain, described in the "Pattern Recognition Theory of Mind" (PRTM) model [11,12]. In addition, AR2P can recognize both static and dynamic patterns [13]. Our formalism works based on thresholds and time series and is dynamically embedded in a recursive hierarchical process. Too, AR2P has two learning mechanisms. The first one, called New_learning, occurs when the input pattern is not recognized, therefore a new pattern recognition module is created to recognize it. The second one, called Reinforcement-Learning, occurs when the input pattern is

recognized, in order to update the modules with the information of the input pattern recognized, such as the weight of importance of a signal.

The previous learning approach of AR2P is supervised. Thus, the problem addressed in this paper is to extend this supervised learning approach to a semi-supervised learning. It is important to tackle this problem (the semi-supervised learning capacity of AR2P) because with the previous learning scheme implies that the patterns must be presented to AR2P. That approach constrains the ability of AR2P for building features (atomic patterns) by itself. Theoretical results suggest that to learn complex patterns, one may need deep architectures [14,15]. Therefore, it is natural to use the deep learning paradigm to design the semi-supervised learning approach for AR2P.

Deep Learning in AR2P allows learning in an automatic way the descriptors/features that will make up the sets of signals associated with the pattern recognition modules, from which the recognition/classification will be made. Specifically, our Deep Learning architecture for AR2P automatically discovers the descriptors/patterns using classic machine learning techniques, such as: K-means, X-means [16,17], and DBSCAN [18]. Our Deep Learning architecture for AR2P is composed of three phases: The first phase, called discovery phase, receives as input a data set from which is discovered a set of atomic descriptors/features. The second phase, called aggregation phase, creates a feature hierarchy from the atomic descriptors. The last phase, called the classification phase, carries out the classification based on the feature hierarchy of the previous phase.

The main difference of our approach compared to classic Deep Learning approaches is the training procedure. The training procedures in the classic approaches are more complex (feedforward, backpropagation, optimization of cost functions, etc.), while in our approach, the training process is easier, without error spreading, and in a single execution.

2. Methods

The methodology of this work starts from a reductionist process (top-down), characterized by the application of an unsupervised mechanism, through an integration process (bottom-up), characterized also by the application of an unsupervised mechanism; until a process of classification and/or recognition, characterized by the application of a supervised mechanism. Hence, the architecture follows a semi-supervised learning approach (see Figure 1).

The architecture is composed of three phases: the first one, named discovery phase, discovers/detects descriptors/features from a dataset. The second one, named aggregation phase, creates a feature hierarchy (merge of descriptors/features) since the descriptors of the lower level. Finally, the classification phase, carries out the classification using the feature hierarchy of the aggregation layer.

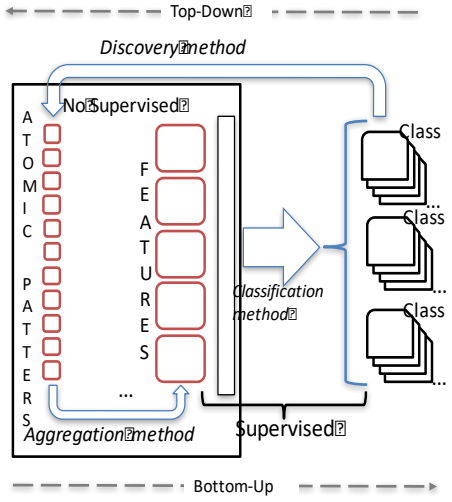


Figure 1. General Architecture of DL-AR2P.

The general algorithm of our Deep Learning Architecture is shown in the Figure 2.

```
Input: Data set
Output: Classification/Recognition
  1. Discover atomic descriptors // fine-grain
  2. Repeat
  2.1. Aggregation of descriptors until all descriptors of the
       previous layer are combined.
  3. Do Classification // complex descriptors
  4. End
```

Figure 2. Deep learning algorithm for Ar2p.

The architecture receives as input a set of data (*D*). Then, from this set *D*, a process of detection of atomic descriptors is initiated (an atomic descriptor is a small piece of information that makes up a class). The atomic descriptors are represented by the small red squares on the left side of the Figure 1, which conform the set of atomic descriptors *S*. Then, from this set *S*, an aggregation process begins, until all the mergeable descriptors of the previous layer are combined. These features that have been merged are represented by the large red squares on the right side of the Figure 1, and formally define the set of parameters from which a supervised classification process is carried out.

The general algorithm works as follows (see Figure 1 and 2). The first step detects atomic descriptors (discovery method of the Figure 1 and line 1 of the Figure 2). Layer 1 is created from these descriptors. Then, it starts an agglomerative hierarchical clustering that combines these small sub-clusters/atomic descriptors, until all descriptors of the previous layer are combined (aggregation method of the Figure 3 and the lines 2 and 2.1 of the Figure 2). These descriptors/features define the layer 2. Subsequently, a classification starts (classification method of the Figure 1 and line 3 of the Figura 2), using the descriptors of the layer 2. Once the features most representative of the classes of the original data/image are identified, these become the input signals for the AR2P recognition modules of each class. These AR2P recognition modules compose the layer 3. Next, each phase is described in more detail.

2.1. Phase discovery

In this work are used two strategies of clustering for the discovery phase of descriptors, a variant of K-means, known as sequential incremental K-means [16], which determine the value of K (named $M_{k_{prev}}$, that denotes the previous value of K); and X-means [17], used in our learning architecture to discover/detect more descriptors from the seed descriptors found $M_{k_{prev}}$. X-means consists of two cyclical phases, as described below. Let's suppose that the data input is D:

- Phase 1. Initial determination of K: it is the execution of the sequential incremental K-means until its convergence.
- Phase 2. Structural improvement: it determines if new centroids should appear within the current model (M_i).

The emergence of new centroids is carried out by dividing into 2 some clusters that have been classified as optimizable, according to the Schwarz criterion (BIC score).

For our implementation, the initial set of clusters is determined through the K-means model previously described. X-means takes this initial value ($M_{K_{prev}}$) and calculates the BIC score for each cluster through the Equation (1).

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}\log R \quad (1)$$

Where, $\hat{\boldsymbol{l}}_{\boldsymbol{j}}(\boldsymbol{D})$ represents the logarithm of the measure of similarity that exists between the data assigned to all the clusters/patterns; $\boldsymbol{p_j}$ is equal to the sum of the probabilities (k-1) of the classes; and R represents the number of samples present in *D* (*R*=|*D*|).

2.2. Phase aggregation

For this phase, our architecture uses as strategy of aggregation based on the DBSCAN algorithm [18]. This method finds the ε neighbors of every point, and identifies the points with more neighbors (core points, which are the initial clusters (C_i)). Next, it assigns each non-core point to a nearby cluster (Ci), if the cluster is one of its ε neighbors, otherwise the non-core point is a noise. The Eps-neighborhood of a point p (denoted by $N_{Eps}(p)$) is defined by the Equation (2).

$$N_{Eps}(p) = \{q \in D | \, dist(p,q) \leq Eps\} \tag{2}$$

Where $dist(p,q)$ is a distance function between p and q (e.g. Manhattan or Euclidian distance) that determines the shape of a neighbourhood, and Eps is a parameter of neighborhood (defined manually or automatically).

On the other hand, it is fixed the minimum number of points (*MinPts*) in the Eps-neighborhood of a core point. In a naive density approach is required that for the q core point, the set of p points is inside of the Eps-neighborhood of q and $N_{Eps}(q)$ contains at least *MinPts* points ($|N_{Eps}(q)| \geq MinPts$). A point p is density-reachable from a core point q if $p \in N_{Eps}(q)$. So, an aggregation of points for *D*, satisfies the following conditions:

- Maximality: $\forall\, p,q$: if $p \in D$ and q is a core point of *D*, then $p \in N_{Eps}(q)$.
- Connectivity: $\forall\, p,q$: if $p \in D$ and q is a core point of *D*, then p is density-reachable from q.

The noise is a set of points in the database *D* that do not belong to any cluster (i.e. noise=$\{p \in D | \, \forall i: p \notin Ci\}$

2.3. Phase classification

This phase is based on the classic AR2P model. AR2P exploits the idea of recursivity in the recognition process, and unbundling/integration of patterns to recognize. A pattern is composed of three parts: The first part is the lowest-level patterns that comprise it. The second part is its identifier or ("name"). In the third and last part are the patterns of higher level, of which in turn it is part. Each pattern recognized at a certain level triggers the next level. That is, when the pattern has been recognized, then it generates a signal sent to the higher levels. A pattern recognition module for our model is formally defined as a 3-tuple, as presented in Equation (3).

$$E = < E, U, S_0 > \tag{3}$$

Where: E is an array composed of 2-tupla E=<S,C>; S=<Signal,State> is an array that represents the set of signals that conform to the pattern recognized by Γ and their respective states. The state variable is "True" when the signal is present and "False" otherwise. The number of signals that conform to the pattern is specific to each module Γ; C is an array that encodes information about the pattern, defined by the 3-tupla C=<F,V,W>, where, F represents the descriptors of Γ, V is the domain vector for each D, W is the weight (importance) of each D for the recognition of the pattern.

The process of construction of the modules can be done manually (created by the knowledge engineer) or through learning algorithms [10].

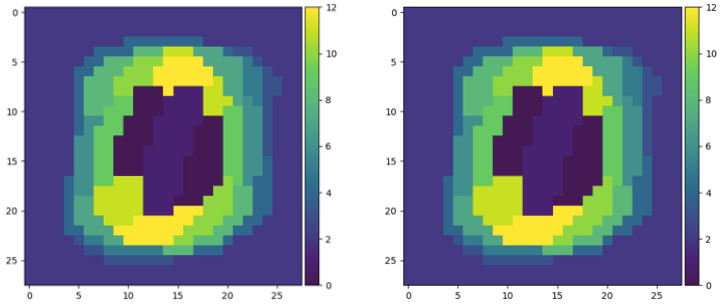
So, the classification phase is carried out in two stages. A first stage consists of extracting the underlying patterns of the descriptors present in layer 2. Subsequently, from these patterns, the data structures of AR2P that make up the recognition modules of the interest patterns are instantiated (see Table 1). Once the pattern recognition modules are established and/or hierarchically instantiated, either manually or using learning algorithms, it is possible to apply the AR2P pattern recognition recursive algorithm to perform the classification and/or recognition of a pattern.

3. Results and discussion

We have used one database available on the UCI Machine Learning Repository (https://goo.gl/KupZFz). The database used is *MNIST handwritten digits*. This database contains examples of normalized and pre-processed handwritten digits with dimension 28x28 pixels.

3.1. Description of the DL-AR2P behaviour using the handwritten digits database

In the discovery phase, with the K-means method are discovered 21 descriptors. Then, such reduced number of clusters are taken as basis to discover more granular descriptors, based on the X-means approach. Such descriptors will define the set of atomic descriptors of the first layer of DL-AR2P, which will represent groups of characteristics or pixels with similar patterns of the training samples. In our case, this new value is equal to 12 descriptors (see Figure 3).



(a) (b)

Figure 3. (a) results of the decomposition phase (12 descriptors) and (b) results of the aggregation phase (10 descriptors) for the MNIST handwritten digits database

After the discovery phase, we proceed to check if the previously generated descriptors can be grouped using the DBSCAN algorithm, the results of this algorithm can be interpreted as: all the clusters formed by the algorithm represent grouped descriptors due to their shared similarities, while those descriptors that DBSCAN determines as noise are descriptors that cannot be linked to any other. These clusters are used to extract the atomic patterns that will be used to build each one of the recognition modules that AR2P will use in the classification phase. For the MNIST handwritten digits database, it determines 10 descriptors (clusters) (see Figure 4).

Now, the extraction of the atomic patterns for AR2P depends on the application domain. In the case of MNIST database, a similarity measure is used that compares the values of the pixels of the original images with respect to each descriptor (cluster) given by DBSCAN. If this value is bigger than a threshold, then it is a candidate atomic pattern of the original images. The similarity value between a descriptor a and an image i, for any descriptor/cluster, is given by the Equation (4).

$$Sim(a,i) = \frac{\vartheta(a,i)}{N} \tag{4}$$

Where ϑ represents the number of pixels whose values are identical in both samples, and N is the total number of pixels in the descriptor to be compared. $Sim(a,i)$ takes values in the interval [0,1].

Based on the previous measure, when the similarity is greater than 0.8, then a is a candidate to be an atomic pattern of the image i. The different as define the set of candidate atomic patterns of the image i (Qp_i) (see Equation (5)).

$$Qp_i = \{a \in layer\ 3\ of\ DL\_AR2P | Sim(a,i) \geq 0.8\} \tag{5}$$

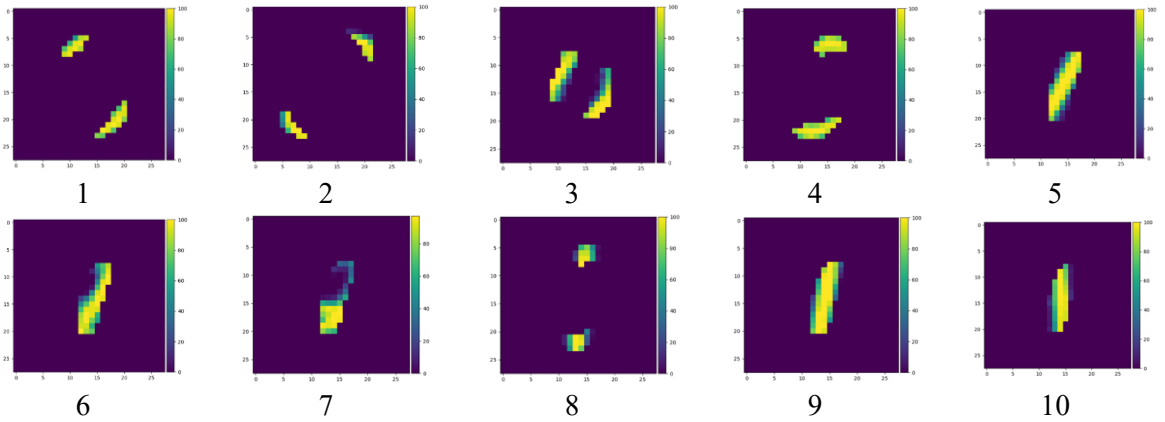


Figure 4. Part of patterns extracted from the layer 2 for the MNIST handwritten digits dataset.

Now, for each member of the set Qp_i is calculated the frequency of its appearance within the set of training images. The atomic patterns of the image i (P_i) are obtained if their frequencies of appearance in the training images for the image i are greater than the average of the frequency of all the candidates, which can be defined by the Equation (6).

$$P_i = \{a \nvdash a \in Qp_i | \text{appearance}(a) \geq average\_\text{appearance}(i)\} \quad (6)$$

The patterns discovered (Figure 4 shows an example of the atomic pattern extracted from the layer 2 for the MNIST handwritten digits dataset) become the set of atomic patterns of AR2P. Based on these patterns, the set of modules of the upper layer X_2 is constructed in a supervised manner, which will recognize the handwritten digits.

Two examples of definition of modules of the upper layer X_2 of AR2P are shown, to recognize the digits '0' and '1', the other modules are defined similarly. The first image of the Figure 6 annotated with a 1, will be represented in the modules as "pattern1", the second image of the Figure 6 annotated with a 2, will be represented in the modules as "pattern2", and so forth.

3.1.1. Creation of the recognition module for "0". For the creation of the recognition module for "0", a subset of atomic patterns that constitute or characterize it, must be selected from the set of patterns established in layer 2 (see Figure 4). They define the general structure of the recognition module of "0" (see Table 1).

The values of the parameters (W) weights and thresholds (U) are defined in a supervised manner, considering the importance of the characteristics/atomic patterns.

Table 1. General recognition model for Γ="0" [2].

E				
S		M		
Signal	State	Descriptor (F)	Domain (V)	Weigh(W)
1	False	Pattern1	<values>	0.9
2	False	Pattern2	< values >	0.9
3	False	Pattern3	< values >	0.9
4	False	Pattern4	< values >	0.9
U:< ΔU1=0.9, ΔU2=0.8>				

3.1.2. Creation of the recognition module for "1". The creation of the pattern recognition module for "1" is defined in a similar way. It is based on a subset of atomic patterns that constitute or characterize it, selected from the set of patterns established in layer 3 (see Table 2).

Table 2. General recognition model for Γ="1" [2]

E				
S		M		
Signal	State	Descriptor (F)	Domain (V)	Weight(W)
1	False	Pattern5	<values>	0.9
2	False	Pattern7	< values >	0.9
3	False	Pattern7	< values >	0.6
4	False	Pattern8	< values >	0.7
5	False	Pattern9	< values >	0.9
6	False	Pattern10	< values >	0.9
U:< ΔU1=0.8, ΔU2=0.7>				

Unlike the previous module, this module contains non-key signals. Particularly, the signals 3 and 4. The signals that are non-key is because they can be in other modules.

3.2. Results analysis

Several simulations were performed for the recognition of 10 images of the digits 0,1,2,…9 written by hand. The Figure 5 shows four of these entries.

Figure 5. Four handwritten digits of MNIST.

According to the results of the Table 3, the precision equal to 1 determines that DL-AR2P recognizes the patterns without making other unexpected recognitions. This precision value is because AR2P learns very specific and unique situations. Recall equal to 1 indicates that DL-AR2P can discover all the digit patterns.

Table 3. Capability of recognition of digits using DL-AR2P.

Metrics/Digit Patterns	Precision	Recall
"1"… "9"	1	1

4. Conclusions

In this work, a deep learning architecture (DL-AR2P) has been proposed, in order to extend the learning capabilities of AR2P. The architecture uses unsupervised learning techniques in the first two layers, then they are coupled with our pattern recognition model to make the classification. The first phase discovers the greatest possible number of atomic descriptors, the second phase, called aggregation, seeks to merge the descriptors more similar through a process of clustering based on density, and the third phase is the supervised classification process of AR2P.

With DL-AR2P, the atomic descriptors or patterns of a domain are automatically extracted, solving the problem of manual definition of the atomic patterns in AR2P. The preliminary classification results are very encouraging, they indicate that the performance of the model with the recognition modules defined in the aggregation phase are very accurate.

As future work, we are going to test our approach with other dataset from the UCI Machine Learning Repository, and we are going to introduce classic feature extraction techniques on the first layers of DL-AR2P, to try the minimization of the execution time of the unsupervised techniques, which is very important in real-time recognition problems.

References

[1] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521(7553)** 436
[2] Schmidhuber J 2015 Deep learning in neural networks: An overview *Neural networks* **61** 85-117
[3] LeCun Y and Bengio Y 1995 Convolutional networks for images, speech, and time series *The Handbook of Brain Theory and Neural Networks* ed Michael A. Arbib (Massachusetts: MIT Press Cambridge) pp 255-258
[4] Salakhutdinov R and Hinton G 2009 Deep boltzmann machines *Proc. Intl Conference on Artificial Intelligence and Statistics* ed David van Dyk and Max Welling (Florida USA: Journal of Machine Learning Research) pp 448–455
[5] Hua Y, Guo J and Hua Z. 2015 Deep belief networks and deep learning *Proceedings of International Conference on Intelligent Computing and Internet of Things* (Harbin, China: IEEE) pp 1–4
[6] Senior A Sak H and Shafran I 2015 Context dependent phone models for LSTM RNN acoustic modelling *Proceedings International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Brisbane, QLD, Australia: IEEE) pp 4585-4589
[7] Puerto E and Aguilar J 2017 Un algoritmo recursivo de reconocimiento de patrones *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia* **40** 95-104
[8] Puerto E Aguilar J and Chavez D 2017 A New Recursive Patterns Matching Model Inspired in Systematic Theory of Human Mind *International Journal of Advancements in Computing Technology (IJACT)* **9(1)** 28-39
[9] Puerto E and Aguilar J 2016 Formal description of a pattern for a recursive process of recognition *Proceedings Latin American Conference on Computational Intelligence* (Cartagena, Colombia: IEEE) pp 1-2
[10] Puerto E and Aguilar J 2016 Learning algorithm for the recursive pattern recognition model *Appl. Artif. Intell* **30** 662–678
[11] Puerto E, Aguilar J and Perez B 2014 Análisis de la teoría de la Mmnte humana basada en el reconocimiento de patrones *Proc. Congreso Internacional en Innovación y Apropiación de las Tecnologías de la Información y las Comunicaciones* (Bucaramanga, Colombia: UNAB) cap. 21
[12] Kurzweil R 2013 How to make mind: Can Nonbiological Brains Have Real Minds of Their Own? *The Futurist* **47(2)** 14–17
[13] Puerto E, Aguilar J and Chávez D 2018 A recursive patterns matching model for the dynamic pattern recognition problem *Appl. Artif. Intell.* **32** 419-432
[14] Aguilar J 2001 Learning algorithm and retrieval process for the multiple classes random neural network model *Neural Processing Letters* **13(1)** 81-91
[15] Aguilar J 1998 Definition of an energy function for the random neural to solve optimization problems *Neural Networks* **11(4)** 731-737
[16] Pham D, Dimov S and Nguyen C 2005 Selection of K in K-means clustering *Proc. Inst. Mech. Eng*. **219(1)** 103-119
[17] Pelleg D and Moore A 2000 X-means: Extending K-means with efficient estimation of the number of clusters *Proceedings of the Seventeenth International Conference on Machine learning (ICML)* (Stanford, USA: Stanford University) pp 727-734
[18] Ester M, Kriegel H, Sander J, and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* ed Evangelos Simoudis, Jiawei Han and Usama Fayyad (Portland, USA: AAAI American Association for Artificial Intelligence) pp 226-231