

Hybrid of Rough Neural Networks for Arabic/Farsi Handwriting Recognition

Elsayed Radwan

Computer Science Department, Faculty of Computer and Information Sciences,
Mansoura University, Egypt, P.O. Box: 35516
Assistant Prof., Institute of Scientific Research and Revival of Islamic Heritage,
Umm Al-Qura University, KSA.

Abstract—Handwritten character recognition is one of the focused areas of research in the field of Pattern Recognition. In this paper, a hybrid model of rough neural network has been developed for recognizing isolated Arabic/Farsi digital characters. It solves the neural network problems; proneness to overfitting, and the empirical nature of model development using rough sets and the dissimilarity analysis. Moreover the perturbation in the input data is violated using rough neuron. This paper describes an evolutionary rough neural network based technique to recognize Arabic/Farsi isolated handwritten digital characters. This method involves hierarchical feature extraction, data clustering and classification. In contrast with conventional neural network, a comparative study is appeared. Also, the details and limitations are discussed.

Keywords- *Rough Sets; Rough Neural Network; Arabic/Farsi Digit Recognition; Dissimilarity Analysis; and Classification.*

I. INTRODUCTION

There are various ways where computers directly take input from the human information system like Optical Character Recognition (OCR), speech recognition, symbolic (Icons, windows) interactive communication etc. Such systems are difficult to design and do not provide complete error free operation. General solution for speech or character recognition is very complex. Even fastest computers would need much larger computational time compared to human response time to perform the same job. To get working solution for the above class of problems, domain specific solutions should be much more efficient [9]. Also, combing different techniques is needed to overcome the shortcomings in each other. In hand-written character recognition problem, the domain is reduced to a small subset of characters of limited number of written characters using specific style. This subset is then further classified to smaller subsets, where each group represents a character [6, 9].

According to previous research [6, 15], the complexity of the handwritten character recognition is greatly increased by the noise problem. Moreover, it should be influenced by the almost infinite inconsistency of handwriting as the result of the writer and the nature of the writing. These characteristics make the progress of handwritten more complex and difficult than typewritten. Thus, a good and effective tool to deal with vagueness and uncertainty of information is needed. Rough Sets [10] is used in the pre-processing stage in this paper that give the ability for dealing with inconsistency and noise

reduction on the handwritten characters. This comes from the fact that Rough Sets can handle the missing feature value and inconsistency on the pattern [10, 12]. Moreover, rough sets are able to get the most essential part of the knowledge with minimum number of features based on its concept of reduct. The feature quality representations that are yielded from rough sets of a data dominate in typewritten character recognition, though it is well suited to all data analysis methods. Because of the nature of handwriting, the perturbation in the specific input data, pairwise representation, should be measured and treated, so a cluster based technique is needed. However the quality representation can be translated to this type of data into a pairwise representation using rough clustering method and problem dependent similarity measure. The main goal of this translation type is that a pairwise representation captures the structure that is captured by the quality data [1, 11]. By transferring quality data into a pairwise, the interpretation the significance of the individual feature is lost, so it is implicitly imbedded by the measure of dissimilarity result.

Since Real life data sets are relatively erroneous, there is an increasing need for effective tools that are able to deal with non-linear problems. Also, incorporating architectural changes result in improving accuracy of approximation. So far, nonlinear problems have been dealt with ANN (Artificial Neural Networks) [15]. ANN was used for pattern recognition for its advantages of, parallel processing and certain fault tolerance [15, 9]. But, this method also has its weaknesses, just as the learning time will be soaring and easy to sink into a local minimum point with the increase of the dimensions [4].

In order to optimize the ANN structure to improve the learning efficiency, a hybrid model based on Rough Sets and Artificial Rough Neural Network (RS-RNN) is proposed. RS-RNN has a drawback that input neurons with zero activation energy negatively affect processing time and space so using rough set reduction algorithm is essential for reducing these superfluous neurons. Moreover, the perturbation in the sample data set is defined as a lower approximation and upper approximation to introduce the idea of the rough neuron [5]. This method takes Rough Set as the modified disposal system of ANN to simplify the structure of it, and to reduce the attribute index and the sample numbers. So a practical method with theoretical support and methodological guarantee is provided efficiently to establish the Arabic/Farsi hand writing recognition system.

Although conventional neural network achieves a good ability to detect all possible interactions between predicted variables, and the availability of multiple training algorithms, it still suffer for different problems such as proneness to over fitting, and the empirical nature of model development. Combining Rough Sets solves the above mentioned problems of ANN and enhance its performance by discovering and removing input neurons with zero weights. In addition, rough neuron treats the problem of data perturbation. In contrast with ANN, a comparative study is mentioned. Database of handwritten Arabic/Farsi sample , IFHCDB (Isolated Farsi Handwritten Character Database) which is created at Amirkabir University of Technology (AUT) and isolated typewritten characters represented by 10×10 pixels [14], is used for more compromise results.

This paper is organized as follows; in section 2 a brief introduction of important fields (rough neural network, rough set) is discussed. Section 3 describes the fundamentals of our method where the method RS-RNN and its performance are given. In Section 4 examine the application and guide the user using it. Then we conclude with section 5 the purpose of that paper and its results.

II. PRELIMINARIES

This section briefs on the basic notions of rough sets that is used in this paper and the detailed definitions can be referred to some related papers [1, 3, 10, 12].

A. Rough Sets theory

Rough set theory is a new mathematical approach to imperfect knowledge [10, 12]. The principle notion of Rough Sets is that lowering the principle in data representation makes it possible to uncover patterns in the data, which may otherwise be obscured by too many details. At the basis of Rough Sets theory is the analysis of the limits of discernibility of subsets X of objects from the universe of discourse U .

Let U be a set of objects (universe of discourse), A be a set of attributes. A decision system is an attribute value table, in which objects of the universe and columns label rows by the attribute. It takes the form $DT = (U, A \cup \{d\})$, where $d \notin A$ is the decision attribute. The elements of A are called conditions attributes and an attribute $a \in A$ can be regarded as a function from the domain U to some value set $V_a, a:U \rightarrow V_a$.

With every subset of attribute $B \subseteq A$ one can easily associate an equivalence relation I_B on U : $I_B = \{(x, y) \in U : \text{for every } a \in B, a(x) = a(y)\}$. Then $I_B = \bigcap_{a \in B} I_a$. If $X \subseteq U$, the sets $\{x \in U : [x]_B \subseteq X\}$ and $\{x \in U : [x]_B \cap X \neq \emptyset\}$ where $[x]_B$ denotes the equivalence class of the object $x \in U$ relative to I_B , are called the B-lower and B-upper approximation of X in S and denoted by $\underline{B}X, \overline{B}X$. The aim of Rough Sets is to obtain irreducible but essential parts of the knowledge encoded by the given information system; these would constitute *reducts* of the

system. So one is, in effect, looking for maximal sets of attributes taken from the initial set (A , say) which induce the same partition on the domain as A . In other words, the essence of the information remains intact, and superfluous attributes are removed. Reducts have nicely characterized in [12] by discernibility matrices and discernibility functions.

A principle task is the method of rule generation is to compute relative to a particular kind of information system, the decision system. R-reducts and d-discernibility matrices are used for this purpose [13]. The methodology is described below.

Let $S = \langle U, A \rangle$ be a decision table, $A = C \cup D$, with C and $D = \{d_1, \dots, d_l\}$ its sets of condition and decision attributes respectively. Divide the decision table $S = \langle U, A \rangle$ into l tables $S_i = \langle U_i, A_i \rangle$, $i = 1, \dots, l$ corresponding to the l decision attributes d_1, \dots, d_l where $U = U_1 \cup \dots \cup U_l$ and $A_i = C \cup \{d_i\}$.

Let $\{x_{i1}, \dots, x_{ip}\}$ be the set of those objects of U_i that occur in $S_i, i = 1, \dots, l$. Now for each d_i -reduct $B = \{b_1, \dots, b_k\}$, a discernibility matrix is defined as follows

$$c_{ij} = \{a \in B : a(x_i) \neq a(x_j)\}, \quad i, j = 1, \dots, n \quad (1)$$

for each object $x_j \in x_{i1}, \dots, x_{ip}$ the discernibility function

$$f_{d_i}^{x_j} = \bigwedge \{\vee(c_{ij}) : 1 \leq i, j \leq n, j < i, c_{ij} \neq \emptyset\} \quad (2)$$

where $\vee(c_{ij})$ is disjunction of all members of c_{ij} . Then $f_{d_i}^{x_j}$ is brought to its conjunctive normal form (c.n.f). One of thus obtains as a dependency rule $r_i : P_i \rightarrow d_i$ where P_i is disjunctive normal form (d.n.f) of $f_{d_i}^{x_j}$, $j \in i_1, \dots, i_p$. The dependency factor γ_i for r_i is given by

$$\gamma_i = \frac{\text{card}(POS_i(d_i))}{\text{card}(U_i)}, \quad (3)$$

Where $POS_i(d_i) = \bigcup_{X \in Id_i} l_i(X)$ and $l_i(X)$ is the lower approximation of X with respect to I_i . The Coefficient $\bar{\gamma}_i = 1 - \gamma_i$ is called the inconsistency degree of DT[13].

B. Rough Neuron

Rough Neuron [5, 16] was developed with an aim of classifying a set of objects into three parts based on a given condition i.e. into the lower, the upper and the negative regions. Rough neural networks [5] consist of both conventional as well as the rough neurons in a fully connected fashion. The rough neuron consists of two individual neurons called the upper bound neuron and the lower bound neuron, which have a mode of sharing information as demonstrated in Figure 1. The lower bound neuron, deals only with the definite

or certain part of the input data and generates its output signal called as the lower boundary-signal. The second neuron called the upper boundary neuron processes only that part of the input data which lies in the upper boundary region evaluated based on the concepts of rough sets and generates the output called upper boundary Signal. This interpretation of upper and lower boundary regions is limited only to the learning or training stage of the neural network.

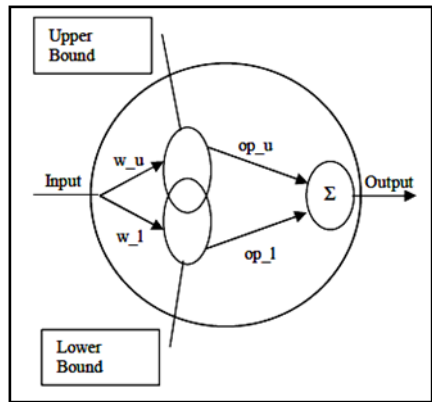


Fig. 1. Rough neuron structure

Tough the neurons (lower and upper) have a conventional sigmoid transfer; the actual output of the lower and upper neurons is given as the following functions

$$output_{\bar{r}} = \max(\text{transfer}(x_{\bar{r}}), \text{transfer}(x_r)) \quad (4)$$

$$output_{\underline{r}} = \min(\text{transfer}(x_{\bar{r}}), \text{transfer}(x_r)) \quad (5)$$

Where

$$x_s = input_i = \sum_{j:j \text{ connected with } i} w_{ij} \cdot output_j \quad (6)$$

Training in rough neural is similar to conventional neural network [16]. During training the network use inductive learning principle to learn from the training set. In supervised training the desired output from output neurons in the training set is known, the weight is modified using learning equation. Neural network use back propagation technique for training. Training using rough Back Propagation performs gradient descent in weight space on an error function.

III. HANDWRITING RECOGNITION BASED ON ROUGH NEURON

Arabic/Farsi handwriting recognition is widely accepted as the means of document authentication, authorization and personal verification in the modern societies of Middle East. For legality most documents like bank cheques, travel passport and academic certificate need to have authorized real time handwritten verification. Thus there is a need for automatic verification, although the difficulties faced in visual assessment of different types and different fonts [9].

In order to recognize handwritten characters, we need to digitize the position of the path of a drawing tool on a writing surface. A primary requirement of a handwriting recognition system is to allow handwriting anywhere on a writing surface and characters of any size.

The program requires the user to draw one or more copies of each character. As characters are drawn, bounding rectangles are calculated for each character. A hand-drawn character can be of any size because the bounding rectangle is used to normalize the image of the character to fit into a small two-dimensional grid that is used for input to a neural network [15]. As the result of normalization, missing values and noise should be appeared. Moreover inconsistency among different patterns for the same user should be discovered. Thus, a good and effective tool to deal with vagueness and uncertainty of information is needed to extract the local features from pattern. Also, an innovative classifier which treats the inconsistency and the perturbation among patterns should be used. Hence, Rough Sets [18] are used in noise reduction and discovering the most admissible local feature, core feature. Also, depending on rough sets methodology of dissimilarity analysis [17], the differences among patterns are localized, i.e. the architecture of the neural network for the whole patterns is designed. Based on the dis-similarity measure, the location where data are perturbed, to define the position of rough neuron, is defined. As the result, the optimal architecture of rough neural networks is detected. Wherever, the superfluous neuron is removed by data reduction of rough sets, the rough neurons in the input pattern are located by the dissimilarity analysis. Then, the outputs of rough neuron (r) are calculated from equations (4, 5). Finally, the conventional neurons in the input pattern are located in correspondence with the other local features without perturbation. The output of the conventional neuron is a function of the output combined from the rough neuron as equation (7)

$$output = \frac{output_{\bar{r}} - output_{\underline{r}}}{ave(output_{\bar{r}}, output_{\underline{r}})} \quad (7)$$

The above function uses the difference between outputs of the upper and lower neurons and normalizes it by the average of the outputs of upper and lower neurons. As the result, each input grid point (local feature) represents the value of an input neuron in a neural network.

In training stage, hand-drawn characters can be of any size, but they are down-sampled to a specific size of grid. The program must determine when the program user is done drawing an individual character. This determination is made by recording the current time in milliseconds when each mouse down movement is recorded. After no mouse down movements has occurred for 400 milliseconds, it is assumed that the training character is no longer being drawn.

A. Rough Sets in the Pre-Processing

Hand-drawn characters can be of any size, but they are down-sampled to a specific size of grid. In this stage, we concentrate on Arabic/Farsi digits that can be represented by eleven features, typewritten digits, as demonstrated in Figure 2.

Some of the attributes that represent Arabic/Farsi digits are superfluous. The superfluous attribute can be reduced using rough set reduction algorithm (using *discernibility Matrix*) and the result is then clustered and applied to rough neural network.

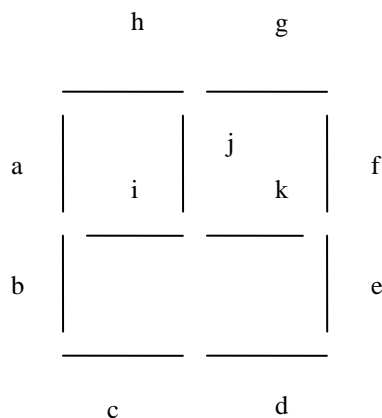


Fig. 2. Template For Arabic/Farsi Digits

Now we will use rough sets for data analysis. Rough Sets theory provides tools for expressing inexact dependencies within data. A Minimum Description Length Principle (MDL-Principle) gives us the reason of why we will use rough sets to reduce the input features of the data. It states, generally speaking, that rules of the most simplified construction, which preserve consistency with data, are likely to classify so far unseen objects with the lowest risk of error. Therefore, to enable classified more objects with high accuracy, it needs to neglect features being the source of redundant information, i.e. to use what is called reduct of attributes. A reduct is a subset of attributes such that it's enough to consider only the features that belong to this subset and still have the same amount of information. If the Decision table has n objects, so by $\Theta(DT)$ we denoted an $n \times n$ matrix (c_{ij}) , where c_{ij} is defined by equation (1), called the discernibility matrix of DT . The main idea of reduct algorithm is that if a set of attributes satisfies the consistency criterion (i.e. be sufficient to discern all the required objects), it must have a non-empty intersection with non-empty elements of the discernibility matrix. One can prove that $a \in CORE(C)$, if and only if there exist two objects, which have the same value for each attribute from C except a . This statement may be expressed by mean of matrix elements (c_{ij}) , as given in Figure 3.

The input data to the model will be quantized first, i.e. the features defined the problem should be identified and labeled. If the input data u_j are given, one has to divide the data into distinct sets and introduce new logical input variable s_k such that:

$$If (u_j \in X_j) THEN (s_k = label(u_j) = true), i \in I, j \in J \quad (8)$$

“Min” performs an operation that is analogical to checking for prime implicates of a Boolean function. The returned value is true if the argument R does not contain redundant attributes. Depending on the reduced set of attribute the robustness of the neural networks will be increase, according to the following theorem [2];

Theorem 1: Let $F(u_1, u_2, \dots, u_n)$ be an arbitrary linearly separable function π is the hyper plane separating

vertices. Then with decreasing the dimensionality from (n) to $(n-1)$, the distance of vertices from π in $(n-1)$ dimensions cannot decrease. \square

Thus;

Corollary 2:

Decreasing number of input neurons increase pattern robustness and reduce tolerance

Proof: it is proved by [2].

At present the reduction algorithm always focuses on reducing attributes and aims at obtaining the best attributes reduction. Potential knowledge contained in data is always targeted when we analyse the database. The complexity of the information system can be reduced by attributes reduction, although not all attribute values of each rule are necessary in the reduced information table, so the dis-similarity analysis among different objects is needed [10]

Input: decision table DT of signature images in binary form for particular person with set of rules describing rough neuron

Output: DT with minimal set of rules describing rough neuron such that the superfluous rules is considered to be conventional neuron (distinguish rough and crisp neurons)

- Compute indiscernibility matrix $\Theta(DT) = (\theta_{ij})$ using equation (1)
- Eliminate any empty or non-minimal elements of $\Theta(DT)$ and create a discernibility list, $k = (k_1, k_2, \dots, k_l)e$, where e is the number of any non empty element in $\Theta(DT)$
- Build families of sets R_0, R_1, \dots, R_e in the following way
 - Set $R_0 = \phi, i = 1$
 - while $i \leq e$ do
 - $S_i = \{R \in R_{i-1} : R \cap k_i \neq \phi\}$
 - $T_i = \bigcup_{k \in k_i} \bigcup_{R \in R_{i-1} : R \cap k_i \neq \phi} \{R \cup \{k\}\}$
 - $M_i = \{R \in T_i : Min(R, k, i) = true\}$
 - $R_i = S_i \cup M_i$
 - $i = i + 1$

Fig. 3. Reduct Computation

B. Rough Sets in Pattern Clustering

Since the nature of human writer differs from one another, the challenge is to discover pattern inherent among different patterns. Because changes in general lead to uncertainty, the appropriate approaches for uncertainty modelling in order to capture, model and predicate the perspective phenomena is considered in dynamic environment. As a consequence, the combination of dynamic data mining and soft computing is very promising. Rough clustering [3] offers strong tool to

detect such changing data structure and aggregate each pixel in its correct cluster.

The appropriate clusters are considered in accordance with the feature reduced by rough set, reduct set. When the new data h in each pattern i is considered to be located at cluster k if it is near to the existing cluster center v_k , which is defined by the following equation

$$d_{kh} \leq \frac{1}{2} \min(d(v_j, v_k)) \quad (9)$$

Where $d(v_j, v_k)$ is the distance between adjacent cluster centers v_j and v_k and d_{kh} is the distance between pixel data h and the class center v_k .

If the new data h in each pattern i does not fit well into the existing cluster where these pixels are far away from current cluster center v_k , new clusters should be formed. Whenever, a little of new pixels are discovered far away from existing clusters centers, they might form a noise that should be removed from our pattern. The number of noise pixels F that are far away from existing cluster center should at most less than the average number of pixels in the lower approximation of the existing clusters weighted by the multiplier f

$$F < f \frac{M^i}{K^i} = f \frac{\sum_{k=1}^{K^i} M_k^i}{K^i} \quad (10)$$

Where $M^i = \sum_{k=1}^{K^i} M_k^i$ is the total number of pixels in the lower approximations, the strict inequality of equation (9), in the pattern i and K^i is the total number of clusters at pattern i . This criterion requires the setting of multiplier f . The smaller f is, the smaller number of noise pixels and thus the greater the number of pixels needs to be established a new cluster.

C. Dissimilarity in Data Analysis

Without knowledge of the domain and specifically the data set description, finding an appropriate weighting to give reasonable result would be computationally expensive. Since rough sets is able to measure dissimilarity between records of boolean value and to compute the knowledge that represent each pattern, attributes of the same value for all records are disregarded and measure to be zero distance. On the other hand, attributes with different values are considered to be dissimilar. There are many dissimilarity / similarity measures that can be used for the comparison of objects studied in the space of measured variables [7, 11].

To fully understand the descriptor which is based on the centroid distance function, it is essential to foremost understand how one computes the centroid. In this section, we note that the formulas were found from [19]. The position of the centroid, the center of gravity in a pattern i , is fixed in relation to the shape in cluster k . The centroid can be calculated by taking the average of all the points that are defined inside a cluster. Under the assumption that the shape in a cluster k is simply connected, we can compute the centroid simply by using only the boundary points. Let N be the total number of points on the border of our cluster k . Here, $n \in [0, N - 1]$. The x and y coordinates of the centroid, denoted by g_x^k and g_y^k respectively, are given by:

$$\begin{cases} g_x^k = \frac{\sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)}{6A_k} \\ g_y^k = \frac{\sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)}{6A_k} \end{cases} \quad (11)$$

Here, the area of the shape, A , is given by the following equation:

$$A_k = \left(\frac{1}{2}\right) \left| \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i) \right| \quad (12)$$

The centroid distance function expresses the distances of the boundary points from the centroid (g_x^k, g_y^k) of a cluster center $v_k = (x_{v_k}, y_{v_k})$. It is given by the following formula

$$d_k^i = \sqrt{\bar{\gamma}_k [(x_{v_k} - g_x^k)^2 + (y_{v_k} - g_y^k)^2]} \quad (13)$$

For each pattern i , d_k^i represents the perturbation of the current data in cluster k that takes into account the cluster significance, where $\bar{\gamma}_k$ is the inconsistency coefficient of cluster k . Thus the level of significance, tolerance, for cluster k at pattern i is given by

$$\alpha_k^i = \frac{d_k^i}{\max_j(d(v_j, v_k))} \quad (14)$$

Where $d(v_j, v_k)$ is the distance between two adjacent cluster centers v_j and v_k . Hence the accepted variations captured in each pattern i forms an interval valued feature values;

$$\left[1 - \frac{1}{2} \alpha_k^i, 1 + \frac{1}{2} \alpha_k^i \right] \quad (15)$$

Hence, superfluous neuron, produced by the rough sets, is considered to be zero input and the other are considered to be rough neuron. If d_k^i is approximately zero then the corresponding neuron is a conventional neuron with input to be one, otherwise the lower and upper neurons inputs are dominated by the interval bounds of equation (15).

D. Adapting the Rough Neural Network

The proposed system uses a Back propagation (BP) RNNs for classification process. In which, the captured characters are trained by the rough neural network using back propagation technique. Back propagation, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task. [4, 8].

The structure of RNN consists of input layer, hidden layer, and output layer [5]. Neurons in the input layer are rough neurons. The number of nodes in the input layer differs according to the feature vector dimensionality. The number of neurons in the input layer is equal to the number of clusters corresponding to each digit pattern. Each input neuron is given its value by equation (15). The hidden layer consists of conventional neurons; the number of neurons in the hidden layer is approximately double the input layer size. The input of the hidden layer neuron is given by equation (7). The output layer consists of four crisp neuron that represent the binary representation corresponding to each Arabic/Farsi digit.

It is a supervised learning method, and is an implementation of the Delta rule. It requires a teacher that knows, or can calculate, the desired output for any given input. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections

that loop). The term is an abbreviation for "backwards propagation of errors". Back propagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable. A generalized sigmoid function equation (16) has been chosen in order to accommodate any non-linearity during the modeling process

$$F(x) = \frac{\alpha}{1 + e^{\beta x + \zeta}} \quad (16)$$

Where x the input is applied to the node and $\{\alpha, \beta, \zeta\}$ are the node parameters. Further, input to a succeeding layer (N) can be given as a linear combination of all outputs of neurons belonging to the preceding layer (M). The linear combination is based on the weighted connections between the respective neurons

$$input_{N_{pi}} = \sum_{i=1}^K w_{M_i N_p} \cdot (output_{M_i}) \quad (17)$$

For learning phase, the algorithm changes the weights till a specified number of epochs or a get a zero error free. Where the new weight is adjusted according to the formula

$$w_{M_i N_p}^{new} = w_{M_i N_p}^{old} + \eta \cdot err_{N_p} \cdot F'(input_{N_p}) \quad (18)$$

Where F' is the first order derivative of equation (16), η is the learning constant. The learning rate is an important parameter in the learning process. It always lies between 0 to 1. In a rough neural network the training goes in two tiers, i.e., two parallel processes run through the network during parameter approximation, one through the lower neuron and the other through the upper neuron [20]. The learning rates of individual neurons are different and are generally time varying and decreases with the number of iterations. The learning rate of the lower neuron is expected to be more than the upper neuron as it is having significant information, which helps in discerning the patterns. Due to the properties of sigmoid function, thus

$$w_{M_i N_p}^{new} = w_{M_i N_p}^{old} + \eta \cdot (d_{N_p} - y_{N_p}) \cdot input_{N_p} \cdot y_{N_p} \cdot (1 - y_{N_p}) \quad (19)$$

Where y_{N_p} is the actual output and d_{N_p} is the desired output of the neuron p .

The neural network space N is defined as a mapping that transforms a neural network $S_i \in N$ to a neural network $S_j \in N$. The main thing to be done is to find the connections between the way in which we make reduction and characteristics of the network that is constructed after reduction. The weights of the connections among input unit (rough or conventional neuron) are denoted by $w_i^h, i = 1, 2, \dots, n, h = 1, 2, \dots, H$ and the weights of the connections between the hidden units and the output units by $w_h^o, h = 1, 2, \dots, H, o = 1, 2, \dots, O$ where H is the number of hidden unit, n is the dimensionality of the input pattern and O is the number of output units.

Sine the output of the network is given by [4, 8, 20]

$$Y_0 = f\left(\sum_{h=1}^H w_h^o f\left(\sum_{i=1}^n x_i w_i^h\right)\right) \quad (20)$$

Where f is a sigmoid function. Since our goal is to find and eliminate as many unneeded network neurons as possible, it is important to identify the effect of violating connections of neurons to the output of the network.

Let Y_0 be considered as a function of single variable corresponding to the connection between the i^{th} input unit and the h^{th} hidden unit. The derivative of Y_0 with respect to the weights of the network is as follows

$$\frac{\partial Y_0}{\partial w_i^h} = Y_0 \times (1 - Y_0) \times w_h^o \times x_i \times \left(1 - f\left(\sum_{i=1}^n x_i w_i^h\right)\right) f\left(\sum_{i=1}^n x_i w_i^h\right) \quad (21)$$

$$\frac{\partial Y_0}{\partial w_h^o} = Y_0 \times (1 - Y_0) \times f\left(\sum_{i=1}^n x_i w_i^h\right) \quad (22)$$

By the Mean Value Theorem, let Y_0 be a function of single variable corresponding to the connection between the i^{th} input unit and the h^{th} hidden unit, thus;

$$Y_{0(w)} = Y_0(w_i^h) + \frac{\partial Y_0(w_i^h + \delta(w - w_i^h))}{\partial w_i^h} \times (w - w_i^h) \quad (23)$$

Where $0 < \delta < 1$. At w equals zero, thus

$$|Y_0(0) - Y_0(w_i^h)| \leq \left|w_i^h \times \frac{\partial Y_0(w_i^h)}{\partial w_i^h}\right| \quad (24)$$

Consider the activation function is sigmoid function that belong to the interval $[-0.5, 0.5]$, thus

$$\left|\frac{\partial Y_i(w)}{\partial w_i^h}\right| \leq \frac{w_h^o x_i}{8} \quad (25)$$

From equation (24) and (25) if follow that

$$|Y_0(0) - Y_0(w_i^h)| \leq \left|w_i^h w_h^o x_i\right| / 8 \quad (26)$$

Inequality (26) illustrates an upper bound on the changes in the output of the network when the weight w_i^h is eliminated. Since $x_i \in [0, 1]$, hence

$$|Y_0(0) - Y_0(w_i^h)| \leq \left|w_i^h w_h^o\right| / 8 \quad (27)$$

Similarly, by considering Y_0 is a function of a single variable v that corresponds to the connection between the h^{th} hidden unit and the o^{th} output unit, Hence

$$\left|\frac{\partial Y_i(v)}{\partial w_h^o}\right| \leq 1/8 \quad (28)$$

Thus, the changes in the output of the network after the weight w_h^o is eliminated is bounded by

$$|Y_i(0) - Y_i(w_h^o)| \leq \left|w_h^o\right| / 8 \quad (29)$$

Equations (27) and (29) show the maximum error that occurred in the network if a connection is eliminated from any layer in the model. Hence, removing the neuron and its connections that do not exist in the reduct set if the error, calculated by equation (27) and (29) are less than some threshold. When performing network learning is done in each iteration, a structure adaptation is consequently performed. The learning process finished whenever no significantly better results of classification with this network.

IV. TESTING HANDWRITING RECOGNITION

This section is aimed at providing an insight into results from the early section. It may also be regarded as space used to provide a reasonable justification for certain trends. This

system has to phases, learning phase and test phase. In learning phase two stages are applied; the first stage is applied on a typewritten data set, the other is applied on the handwritten data set. First, the Arabic/Farsi typewritten instance digits are clustered in accordance to eleven features as demonstrated early in Figure (2) and can be represented by data attribute value Table (1)

Table 1: Data Attribute Table For Arabic/Farsi Digits

	a	b	c	d	e	f	g	h	i	j	k
0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	1	1	0	0	0
3	1	1	0	0	0	1	0	0	1	1	1
4	1	1	1	0	0	0	0	1	1	0	0
5	1	1	1	1	1	1	1	1	0	0	0
6	0	0	0	0	1	1	1	1	0	0	0
7	1	1	1	1	1	1	0	0	0	0	0
8	1	1	0	0	1	1	1	1	0	0	0
9	1	0	0	0	1	1	1	1	1	0	1

Using rough set discernibility matrix, these attributes are reduced in which {e} is the most significant attribute associated with the reduct set to be {a,c,e,h,k}, where $\gamma_a = \gamma_c = \gamma_e = 0.4, \gamma_h = 0.6$ and $\gamma_k = 0.2$. By Dissimilarity analysis of rough set [10], differences among patterns and the optimal structure of NN are discovered, as illustrated in Figure 4. The number of nodes in input layer are chosen in accordance with the reduct set. The pattern of zero are differs than three pattern in one attribute is called a, while as zero pattern is characterized by only one attribute called k. Thus, the three pattern is characterized by the attributed {a,k}.

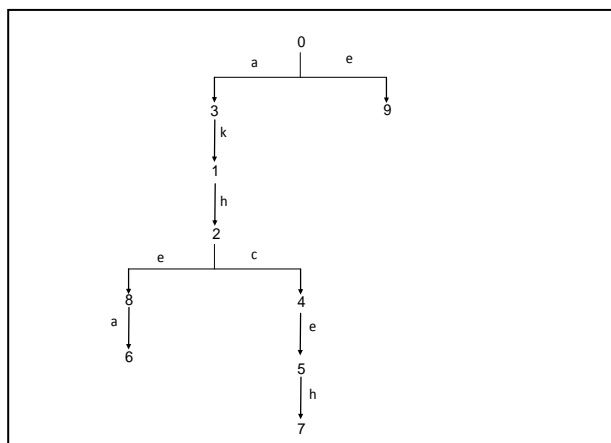


Fig. 4. The Dis-Similarity Diagram Among Different Arabic/Farsi Digits

Moreover, the center of gravity, calculated by equation (11), for typewritten data set represents v_k , center of the cluster. Using this reduction algorithm on a typewritten data reduces the number of learning epochs as a result of reducing number of input neurons as shown on Figure 5 which compare number of iterations before reduction and after it.

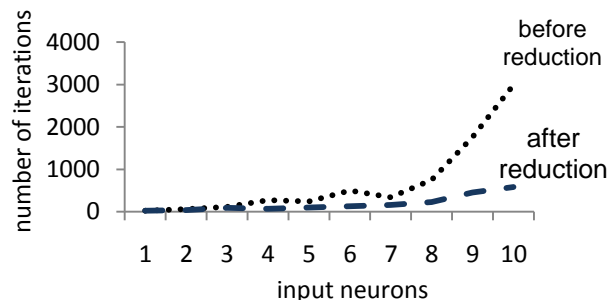


Fig. 5. Comparing Number Of Iterations Using Neural Network With Number Of Iterations After Applying Rough Sets

To evaluate the proposed hybrid model on isolated Arabic/Farsi handwritten characters, IFHCBD which is created at Amirkabir University of Technology has been used. The RS-RNN model is applied on Arabic/Farsi handwritten digits. IFHCDB contains a set of images for Arabic/Farsi digits that are divided into 88 training data and 30 test data. First, rough sets are used in clustering the handwritten data set, where the noise are discovered and removed by equation (10). Second, dissimilarity in data analysis is applied to compute the perturbation in the input data, by equation (15). Hence rough neurons are discovered and the optimal architecture is represented by rough sets' clusters. Finally, an implementation, as shown in Figure (6), for the Back propagation algorithm is performed where a structure adaptation is done by each learning epochs. The learning process is finished whenever the error function is at most 0.01.

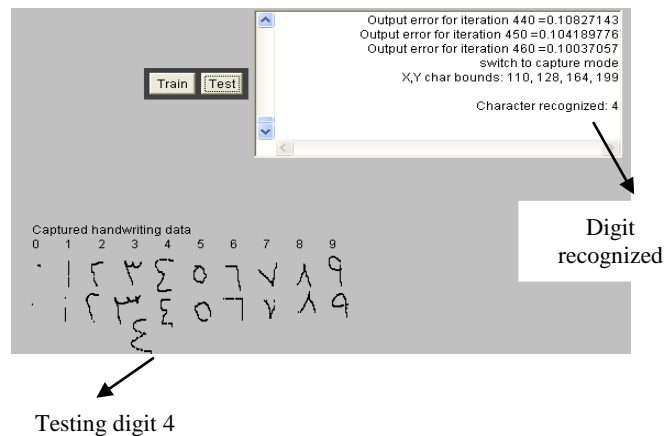


Fig. 6. The Implementation Of The RS-RNN

At the end of the study, we prove the ability of our System depending on the results for the recognition accuracy. Comparing errors for each digit pattern that are resulted from training by neural network against those are resulted from RS-RNN, as demonstrated in Figure (7).

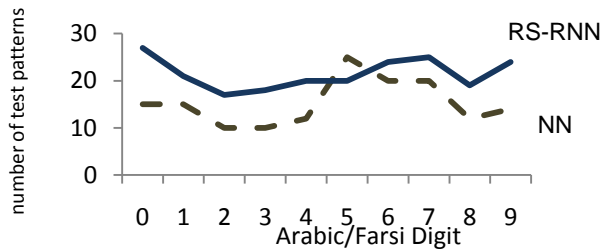


Fig. 7. Number Of Recognized Patterns By NN And RS-RNN

Moreover, a comparison among theories [15], such as genetic algorithms, simple object modeling, statistical method and rough sets with neural networks, and RS-RNN in terms of recognition accuracy is demonstrated in Figure(8). As mentioned RS-RNN approach is able to recognize the Arabic/Farsi handwritten digits more efficient than others.

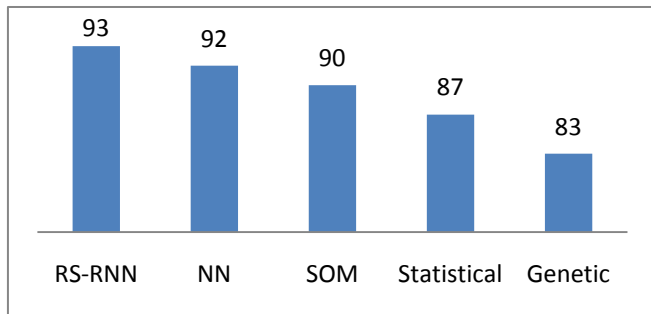


Fig. 8. the accuracy measure among different theories and RS-RNN

V. CONCLUSIONS

Handwriting number recognition, a system for recognizing isolated digits as standard characters, is a challenging problem because different users have their own handwriting style. Moreover, it is affected by noise that established during acquisition and normalization. The main goal to this paper is to recognizing isolated Arabic/Farsi digits exist in different forms.

This paper presents a hybrid model that starts with acquiring and normalizing an image containing Arabic/Farsi digits. The digitized image was treated by rough sets. Rough sets played an important role in reducing the feature attribute, reduct, and discovering dissimilarity among different patterns. Also, rough sets segment the user pattern into different clusters in accordance with the Arabic/Farsi digit pattern. Moreover, the noise has been eliminated and the dissimilarity between the user cluster and its corresponding feature was measured.

By this paper, the optimal architecture of the rough neural network was discovered. Finally an adaptation of the rough neuron was applied during the learning phase using the back propagation algorithm.

The results were tested on standard data and proved the efficiency of our method. This approach efficiently chooses a segmentation method to fit our demands. Our approach

successfully design and implement rough neural network which go without demands. After that RS-RNN are able to understand the Arabic/Farsi numbers that was manually written by the users.

REFERENCES

- [1] Duo Chen, Du-Wu Cui, Chao-Xue Wang, Zhu-Rong Wang, "A Rough Set- Based Hierarchical Clustering Algorithm for Categorical Data", International Journal of Information Technology, vol. 12, no. 3, pp. 149-159, 2006.
- [2] Elsayed Radwan, Omaira Nomir, and Eiichiro Tazaki, "A New Learning Method for Cellular Neural Networks Templates based on Hybrid of Rough Sets and Genetic Algorithms, " International Journal of Computer Science and Information Security, vol. 8, no. 3, pp. 155-164, August 2010.
- [3] Georg Peters, Richard Weber, Rene Nowatzke, "Dynamic Rough Clustering and its Applications", Applied Soft Computing, vol. 12, pp. 3193-3207, 2012.
- [4] Insung Jung, and Gi-Nam Wang, "Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network", International Journal of Electrical and Electronics Engineering, vol. 2, no. 6, pp. 365-369, 2008.
- [5] Lingras, P. J., " Rough Neural Networks", Proceedings of Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, Granda, Spain, pp. 1445-1450, 1996.
- [6] Luiz S. Oliveira and Robert Sabourin, "Support Vector Machines for Handwritten Numerical String Recognition", Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004).
- [7] M.M. Richter: Similarity, Studies in Computational Intelligence (SCI), 73, 25-90, Springer-Verlag Berlin Heidelberg 2008
- [8] Mark Watson, "Intelligent Java Applications for Internet and Intranets", Morgan Kaufmann Publisher, San Francisco, California, chapter 4,10 ,1997.
- [9] Nalini K. Ratha , Venu Govindaraju, "Advances in Biometrics -Sensors, Algorithms and Systems", Springer-Verlag London Limited, 2008.
- [10] Pawlak, Z., " Rough Sets- Theoretical Aspects of Reasoning about Data", Kluwer Academic, Dordrecht, 1991.
- [11] Piotr Zerzucha, and Beata Walczak, " Concept of Dis-Similarity in Data Analysis", Trends in Analytical Chemistry, vol. 38, pp. 116-128, 2012.
- [12] Polkowski, L., Skowron, A.(Eds), Rough Sets in Knowledge Discovery, Physica- Verlag, Heidelberg, vols: 1 and 2, 1998.
- [13] Roman, W. Swiniarski, Andrzej Skowron, " Rough Set Methods in Feature Selection and Recognition", Pattern Recognition Letters, vol. 24, pp. 833-849, 2003.
- [14] S. Mozaffari, K. Faez, M. Zairatban, and S. M. Golzan, " A comprehensive Isolated Farsi/Arabic Chracter DataBase for Hanwritten OCR Research" 10th International WorkShop on Frontiers in Handwriting Recognition (IWFHR), La Baule, France, pp. 385-389, 2006.
- [15] Saleh Ali K. Al-Omari, Putra Sumari, Sadik A. Al-Taweel and Anas J.A. Husain, " Digital Recognition using Neural Network", Journal of Computer Science, vol. 55, no. 6, pp. 427-434, 2009.
- [16] Sandeep Chandana, Rene V. Mayorga, " Rough Neuron based on Space Partitioning", Neurocomputing, vol. 74, pp. 2052-2061, 2011.
- [17] Sheela Ramanna, " Near Tolerance Rough Sets", Applied Mathematical Sciences, vol. 5, no. 38, 1895 – 1902, 2011.
- [18] Shinya imai, che-wei lin, junzo watada and gwo-hshiung tzeng, "Rough Sets Approach to Human Resource Development of Information Technology Corporations", IJSSST, vol. 9, no. 2, pp. 3142, 2008.
- [19] Yang Mingqiang, Kpalma Kidiyo and Ronsin Joseph, "A Survey of Shape Feature Extraction Techniques," Pattern Recognition Techniques, Technology and Applications, Peng-Yeng Yin (Ed.),pp. 43-90, 2008.
- [20] Yassar Hassan, Eiichiro Tazaki, Sina Egawa, Kazuho Suyama, " Rough Neural Classifier System", IEEE International Conference on Systems, Man and Cybernetics, 2002.



AUTHORS PROFILE

Elsayed Radwan, received his B.Sc. and M.Sc. from Mathematics Dept. Faculty of Science, Mansoura University, Egypt. He received his Ph.D. from Intelligent Lab., Graduate School of Engineering, Toin

University of Yokohama, Japan. He became a Lecturer at Computer Science Dept., Faculty of Computer and Information Sciences, Mansoura University, Egypt on 2007. His areas of interest are Soft Computing, Evolutionary Computing, Pattern Recognition, and Cellular Neural Networks. He is now working as an Assistant Professor of Computer Science in Institute of Scientific Research and Revival of Islamic Heritage.