

Paper:

# Rough Set Approach for Overall Performance Improvement of an Unsupervised ANN-Based Pattern Classifier

Ashwin Kothari and Avinash Keskar

Department of Electronics & C.S. Engineering  
Visvesvaraya National Institute of Technology Nagpur, India, 440010  
E-mail: agkothari72@rediffmail.com, ashwinkothari@ece.vnit.ac.in  
[Received December 11, 2008; accepted March 10, 2009]

Most conventional approaches to pattern classification using unsupervised ANN use clusterification with the entire feature set. The redundancy (dependence) of some features in such cases makes feature space dimensionality too complex to handle. Early convergence is another factor desired for the training phase in networks trying different neural architectures or learning algorithms. As approaches evolve and are applied, the hybridization of neural concepts with other tools has yielded useful results. A rough set is one such approximation tool that works well when in environments heavy with inconsistency and ambiguity in data or involving missing data. Approaches using rough sets may be used at the preprocessing, learning and neuron architectural levels. Preprocessing and architectural approaches are discussed here using Rough sets to improve overall performance of pattern classifiers used in character recognitions.

**Keywords:** unsupervised ANN, pattern classification, rough sets, reducts, rough neuron

## 1. Introduction

Rough sets have been proposed for numerous applications, especially rough set theory for artificial intelligence and cognitive sciences, e.g. machine learning, knowledge discovery, data mining, expert systems, approximate reasoning and pattern recognition. Artificial Neural Networks (ANNs) in their most general form target the development of systems functioning similarly to the human brain. Connections and data exchange in the network are application dependent. Rough sets and neural networks are combined because they yield hidden patterns in ambiguous data and provide tools for data and pattern analysis. Rough sets are useful in dealing with inconsistent data, having the following advantages [1, 2, 3]:

- Providing efficient algorithms for discovering hidden patterns in data.
- Identifying relationships not otherwise found using statistical methods.

- Enabling both qualitative and quantitative data to be used.
- Finding minimal sets of data that can be used for classifications i.e., data reduction.
- Evaluating the significance of data.
- Generating sets of decision rules from data.

Rough set theory is applied in the following three stages in neural-network based pattern classifiers:

**Preprocessing stage:** For any pattern classifier making choice of optimum feature sets containing all significant and independent features only is very critical. Hence before actual neural-based classification, it is essential to eliminate insignificant and dependent attributes. Rough set concepts can be used in the preprocessing stage for reducing number of attributes and thereby the complexity of hardware implementation. Rough set concepts are also used in selecting appropriate, relevant attributes alone during feature extraction. Rough-set based methods can also be used without adversely affecting neural classifier classification accuracy [4].

**Learning algorithm stage:** Competitive learning is widely used when neural networks are trained unsupervised. To speed up convergence and increase classifier classification accuracy different rules are used to update weights. These training rules give better results when are combined with different widely used soft computing tools. Rules based on rough set theory are similarly defined on the set consisting of winner and adjacent neurons.

**Architectural stage:** In dealing with real-life data, rough-set-based improvement in the first two stages definitely enhances pattern classifier performance. Classification accuracy—can further be improved using architectural changes based on the principle of rough sets. This involves modifying regular neurons in to rough neurons and using neural rough—neuron topology.

The sections that follow detail the use of rough set theory concepts in neural network architecture, together with the preprocessing stage approach.



Table 1. IS.

Pixel Density	Euler Number	Verical Crossings	Horizontal Crossings	Number of Loops	zone 1	zone 2	Chain Code No. of 1s	Chain Code No. of 2s	Compactness	Perimeter
0.0919	5	2	2	1	48	152	21	29	0.5517	220.7523
0.0869	5	2	2	1	82	127	18	27	0.574	218.9949
0.08489	5	2	2	1	83	107	19	30	0.5591	222.5097
0.1295	4	2	3	1	91	119	17	31	0.5359	219.4802
0.0726	5	2	2	1	71	66	20	25	0.5718	217.1371
0.1691	7	3	3	1	246	172	6	55	0.47	260.2843
0.1151	4	4	2	1	68	188	20	30	0.5124	225.4386
0.0668	-2	2	2	0	79	48	17	31	0.5351	228.3675
0.0824	5	3	2	1	94	84	15	35	0.5235	227.6812

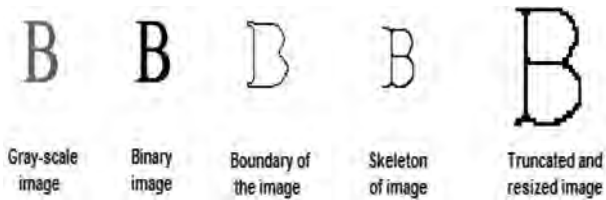


Fig. 1. Image pre-processing stages.

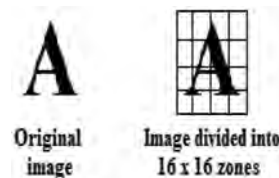


Fig. 2. Zoning.

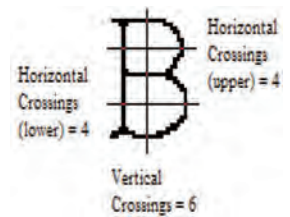


Fig. 3. Crossings.

## 2. Image Preprocessing

Implementing the pre-processing dataset approach using rough sets and reducing the size of feature space, requires the tabular formatted dataset known as an information system (IS) [4]. Numerous image preprocessing and feature extraction steps are taken on input data set of various character images of different fonts to form the required IS. Pre-processing mainly targets: binarization, noise reduction, skeletonization, boundary extraction, stroke compensation [4, 5, 6, 7] truncation of redundant image portion and resizing to a specific size. Image binarization converts a gray scale image to a binary image. Noise is reduced using morphological operations such as dilation, and erosion. Skeletonizing the image yields an approximate single-pixel skeleton that aids in subsequent feature extraction and classification stages. The image's perimeter is extracted to further obtain the boundary-related attributes such as chain codes and numbers of loops. Stroke-width compensation repairs character strokes, fills small holes and reduces character unevenness. Because white areas surrounding an image can create noise in feature extraction and increases image size unnecessarily, truncation removes this white portion. Eventually resizing the image to a pre-defined size:  $64 \times 64$  pixels in this case. Fig. 1 shows preprocessing for a sample input image.

## 3. Feature Extraction

Feature extraction as used here involves two features types—statistical and structural [8, 9, 10]. Major statistical features used are: zoning, crossings, pixel density, Euler numbers and compactness. Structural features are based on topological and geometrical character features, such as the aspect ratio, loops and strokes and their direc-

tions. The image boundary is obtained and its chain code calculated involves counting of number of ones, twos till number of eights. Chain code is employed as a useful discernibility attribute between two classes. The number of loops and image perimeter are also obtained. Figs. 2 and 3 show zoning and crossings used in feature extraction.

Having thus extracted feature table for each input image a table of features extracted for each input image, each of the columns or attributes is called a condition attribute and the entire table becomes the required the IS. Table 1 shows 32 condition attribute extracted and the IS. In all 32 such condition attributes extracted and the respective IS is shown in Table 1. Using rough sets reduces the numbers of these attributes.

## 4. Reduct Based Preprocessing

Pure competitive neural-based classification uses the entire set of input features [11, 12]. Many attributes of this set are redundant or insignificant however, and contribute little to classification. Many hybrid approaches have been attempted to unify neural processing with other soft computing tools to speed up convergence with an optimum number of training attributes having the desired accuracy. The neuro-fuzzy hybridized approach widely used and result oriented but is limited by lengthy membership function calculation and an inability to strongly approximate for any type of data i.e. consistent, inconsistent or incomplete [13, 14]. Recent approaches combining neural processing and rough sets are used at different levels, as stated, discussed earlier. The hybrid rough neuro approach we propose adds a neural network with a rough set-based input data preprocessor, To reduce redundant calculation and implementation overhead. The neural network is also made to provide the desired accuracy with an optimum number of features as shown in Fig. 4. Subse-

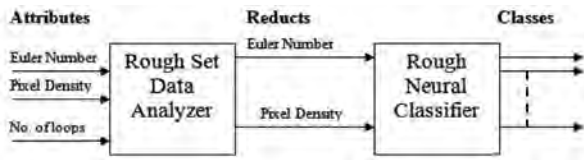


Fig. 4. Hybrid rough neuro approach.

quently implementing a rough neural classifier combines the second and third approaches discussed earlier. Our thus unifies all three approaches.

The basic principle of rough-set-based learning is finding redundancies and dependencies between given features of data to be classified. Rough set theory efficiently handles uncertainty (upper and lower data point approximations) and granular computing using the discernible objects in the information of the systems [13, 14]. Condition attribute play a major role in IS processing with the theory’s application directly suggesting the use of reducts and cores to obtain reduced feature sets. The following arguments are constructed when such reductions are applied to an unsupervised artificial neural network using the Kohonen learning rule:

- If the cardinality of the reduced space is less than the number of decision classes, then the reduction may not directly aid in making classification efficient. Efficiency obtained with original feature space may reduce significantly.
- Evaluating cores and hence the useful reduct set is not always necessary when applied to the case of neural networks.

The following four types of reduction proposed using the theory of rough sets are useful in easily identifying and arranging significant discriminative contributors useful in partitioning the input signal in Rough Neuron Architecture discussed later. The key to reducing the number of attributes lies in the discernibility matrix and probabilities in rough set theory.

Consider IS = (U, A, V, f) where U is the universal sets containing all objects i.e.,  $U = \{x_1, x_2, x_3, \dots, x_n\}$ , n is the total number of objects;  $A = CU\{d\}$  where C denotes the set of condition attribute and d is the decision attribute; V denotes sets of values taken by each condition attribute; and f is a function between the element in U and its value, and the value of object xi to attribute a is a(x<sub>i</sub>). The total number of condition attributes is m i.e.  $|C| = m$ . The number of decision classes is t i.e.  $|Vd| = t$ . Discernibility matrix (t × t), whose entries contain the relative significance of each attribute is explained as follows [15, 16].

**4.1. Method 1**

Method1 for relative attribute significance in discerning between two compared classes x and y is as follows:

$$P(x)x,y|a_i. \dots \dots \dots (1)$$

Table 2. Relative contribution of different condition attributes.

Class	1	2	3
1	–	Euler(0.24), PixDen(0), AvgPixDen(0.79)	Euler(0.33), PixDen(0), AvgPixDen(0.56)
2	Euler(0.66), PixDen(0), AvgPixDen(0.49)	–	Euler(0.33), PixDen(0), AvgPixDen(0.66)
3	Euler(0.66), PixDen(0), AvgPixDen(0.5)	Euler(0.24), PixDen(0), AvgPixDen(0.39)	–

Where  $P(x)x,y|a_i$  is the probability of an object belonging to class x given the only information as attribute  $a_i$  ( $i = 1, 2, 3, \dots, m$ ) when objects of x are discerned from y. If the value is 1, the particular attribute is most significant and if the value is 0, then the particular attribute is the least significant.

**4.2. Method 2**

In method 2, relative attribute significance in discerning between two compared classes x and y is as follows:

$$P(a_i(x))x,y|a_i. \dots \dots \dots (2)$$

Where  $P(a_i(x))x,y|a_i$  is the probability of attribute value  $a_i(x)$  which discerns class x from y, given the only information as attribute  $a_i$  ( $i = 1, 2, 3, \dots, m$ ).

**4.3. Methods 3 and 4**

In methods 3 and 4, the relative significance of each attribute in discerning two compared classes x and y is as follows:

$$P(x)x,y|a_i + P(a_i(x))x,y|a_i \dots \dots \dots (3)$$

$$P(x)x,y|a_i * P(a_i(x))x,y|a_i. \dots \dots \dots (4)$$

Where terms  $P(x)x,y|a_i$  and  $P(a_i(x))x,y|a_i$  are explained above. The algorithm used is as follows,

- Step 1. Obtain the IS whose feature space must be reduced.
- Step 2. Discernibility matrix (t×t) is constructed and the entry for each attribute is given by one of methods 1 to 4.
- Step 3. The relative sum for each attribute is obtained i.e., the contribution of each attribute over the table is summed.
- Step 4. The most significant contributors based on the relative sum are selected based on the requirement or set threshold.

This results in a set of reducts [15, 16] to be used for training the neural network. With this we also get the relative contribution of each condition attribute. Using these values we easily determine which attributes are most significant for classifying the two classes under consideration as clarified in **Table 2**. Row and column headings

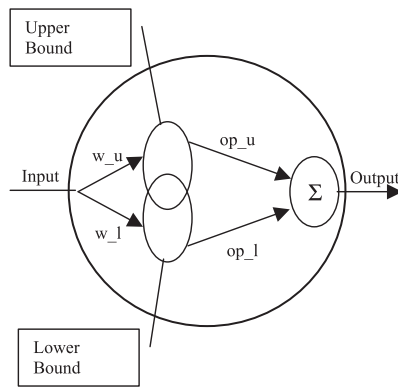


Fig. 5. Rough Neuron Structure.

are classes. So **Table 2** defines data for a partial case of a three-class classification problem for which limited data is used from the original IS. Each entry related to a row and column crossing defines the contribution of different attributes for discerning classes. A close look at this pattern recognition data shows that conditional attribute PixDen (zone1) is redundant., so only Euler Number and Avg PixDen are needed to make a decision. These reduced attributes are called reducts.

### 5. Rough Neuron Based Uann

In most conventional neural networks data given to the network is precise and expected to be consistent throughout the process an assumption rarely borne out in real life situations, so data is pre and post processed to achieve this. Given that most data inherently contains randomness or impreciseness that cannot always be processed renders the neural network unable to deal with such data. Given this situation, the rough neuron considers random or erroneous parts of data and processes it to help classify the pattern. Data is divided into significant, more contributory parts and redundant, less contributory parts using the rough concepts described earlier. The rough neuron consists of a combination of two individual neurons, lower-boundary and upper-boundary [17, 18] in which the lower boundary neuron, deals only with definite or certain input data and generates its output as a lower boundary-signal. The upper boundary neuron processes only that input data in the upper boundary region evaluated based on rough sets concepts to generate output called an upper-boundary signal. This upper and lower boundary region interpretation is limited only by neural network’s learning or training stage, as detailed in **Fig. 5**.

Lower and upper-boundary neurons have a generalized sigmoid transfer function. Rough neuron output is calculated as follows:

$$F(x) = \alpha / (1 + e^{\beta \cdot x + \zeta}) \dots \dots \dots (5)$$

Both upper and lower neuron input is :

$$(\text{input}) = \sum (X - \text{weight})^2 \dots \dots \dots (6)$$

The output of the lower and upper neurons is as follows:

$$\begin{aligned} (\text{output})_{\text{lower}} &= \min(F(\text{input}_{\text{lower}}, F(\text{input}^{\text{upper}})) \\ (\text{output})^{\text{upper}} &= \max(F(\text{input}_{\text{lower}}, F(\text{input}^{\text{upper}})). \end{aligned} \dots (7)$$

Making combined rough neuron output:

$$\text{Output} = (\text{output})_{\text{lower}} + (\text{output})^{\text{upper}} \dots \dots (8)$$

### 6. Training

In the unsupervised mode used here the neural network operates on the competitive learning principle, where every neuron competes with every other neuron to win given input. Once the winner is determined only that neuron’s weight is updated. To make the system more adaptive, adjacent are also be updated appropriately, making neurons unique triggers for objects of the same class. For a given test input set, output for all rough neurons is calculated using Eqs. (5) to (8). For each input, the neuron whose weight closely matches input is declared the winner, i.e., the neuron with the lowest output value. Once the winner is declared the weights of that neuron is updated based on Eq. (9).

$$W_{\text{new}} = W_{\text{old}} + \alpha(t) \times h(t) \times (X - W_{\text{old}}) \dots \dots (9)$$

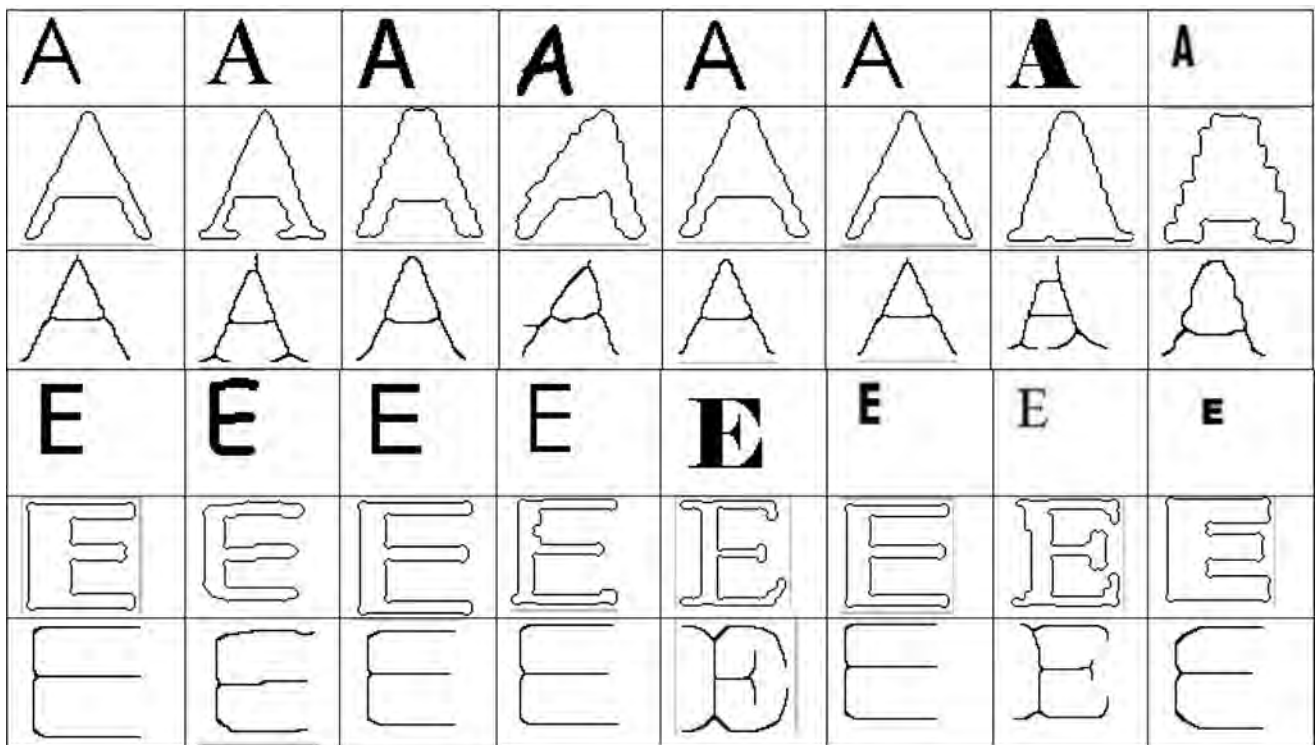
Where  $\alpha(t)$  is the time varying learning constant and  $h(t)$  is the adjacent function .The learning rate, an important parameter in the learning process lies in 0 to 1. Larger values of  $\alpha$  may speed up convergence but also may cause oscillations in output. Smaller  $\alpha$  may lead to the network requiring an unusually long time to converge despite less error at convergence. The learning rate generally is application-dependent and is typically chosen through experiments. Here, however it is chosen based on the principle of adaptive learning. In the rough neural network used here training occurs at two levels, meaning that the two parallel processes run through the network in parameter approximation, one through the lower neuron and the other through the upper neuron [18]. Learning rates of individual neurons differ, are generally time-varying and decreases with the number of iterations. The rough neural network thus uses a dynamic or adaptive learning constant. The learning rate of the lower neuron is expected exceed that of the upper neuron because it has significant information, which helps in discerning patterns. A network’s point of convergence is that at which no further significant change occurs in neuron’s weight. At convergence we expect input to be classified into distinct classes by input value. At this point the network is declared to have been trained.

### 7. Results and Conclusions

The character recognition problem is used to verify our proposed approaches. A data set is formed by images of English letters in 18 different fonts. **Fig. 6** shows the training data subset. **Fig. 7 (a)** and **(b)** show sample images

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
A	B	C	D	E	F	G	H	I	J	K	L	M
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
A	B	C	D	E	F	G	H	I	J	K	L	M
Q	R	S	T	U	V	W	X	Y	Z	Q	R	S
T	U	V	W	X	Y	Z	Q	R	S	T	U	V
W	X	Y	Z	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>
<b>Z</b>	Q	R	S	T	U	V	W	X	Y	Z	o	o
o	<b>o</b>	O	P	<b>P</b>	P	<b>P</b>	P	N	<b>N</b>	N	<b>N</b>	N

Fig. 6. Subset of data used for training and forming the IS.



(a)

Pixel Density	Euler Number	Verical Crossings	Horizontal Crossings	Number of Loops	zone 1	zone 2	Chain Code No. of 1s	Chain Code No. of 2s	Compactness	Perimeter
0.0562	4	2	2	1	0	69	30	42	0.3624	254.9361
0.0592	6	2	2	1	0	46	36	39	0.3486	297.6224
0.0922	5	2	2	1	0	143	26	49	0.3797	261.1371
0.0986	5	2	2	1	0	22	37	37	0.3579	263.7645

(b)

Fig. 7. (a) Sample Images. (b) Sample of IS after boundary and skeleton extraction.

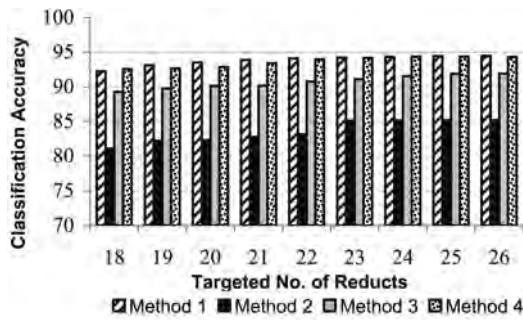


Fig. 8. Number of Reducts vs Classification accuracy for all four methods.

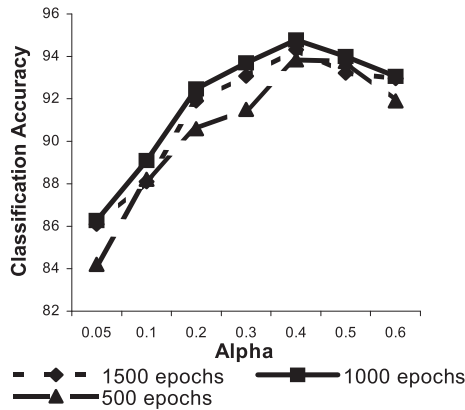


Fig. 9. Epochs vs Classification Accuracy for various  $\alpha$  values.

with boundary and skeleton extraction. This data forms the required IS. So the work here is considered for verifying the application of the rough neuro hybrid approach for consistent and inconsistent types of data. Verification for the application of such an approach to missing data is projected work.

The result for reduct based preprocessing is summarized as follows:

- Algorithms proposed for obtaining reductions are advantageous, because they do not use the step, evaluating cores and reducts of the set.
- Such reductions provide more flexibility in choosing useful information for neural network based classification.
- The Classification accuracy of the pure neural network depends on the number of input features fed to it. Due to reduct calculation, 32 input features are reduced to a targeted number of reducts. As shown by the graph in Fig. 8 methods 1, 3 and 4 works reasonably better than method 2. Excessive attribute reduction for all methods may result in loss of information, in turn reducing classification accuracy. When the number of attributes is below 18, accuracy falls dramatically. Even if we consider the case of 25 input features, the resulting reduction is 21% for a total number of 32 attributes. Reduction is maximum and 44%, if the attributes numbers 18.

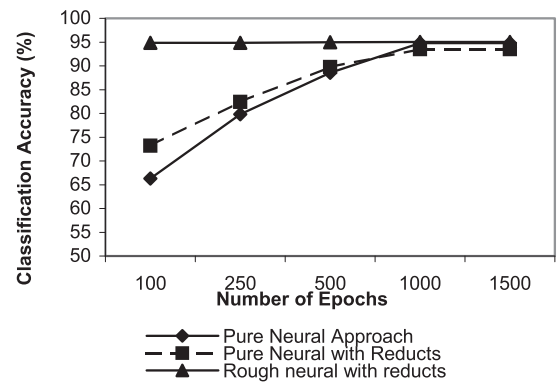


Fig. 10. Number of Epochs vs Classification accuracy for three methods for  $\alpha = 0.4$ .

Table 3. Results.

Network Type	Number of iterations	Learning rate ( $\alpha$ )	Classification accuracy (%)
Pure Neural	1000	0.4	94.78
Pure Neural with reducts	1000	0.4	93.54
Rough Neural	100	0.4, 0.3 (Lower and Upper neurons resp.)	94.84

The graph in Fig. 9 shows variation in classification accuracy vs values of epochs. Efficiency is maximized at learning co-efficient ( $\alpha$ ), 0.4 and epochs numbering 1000. Efficiency rises from 500 to 1000 epochs, but, the efficiency for 1000 epochs is as good as or better than that for 1500 epochs.

Figure 10 shows gradually improving classification accuracy with the increasing number of epochs for all the three methods for a common learning constant value of 0.4. Convergence occurs early for the third method and classification accuracy remains the same without significant improvement. The network based on the rough neuro hybrid approach gives almost compatible classification accuracy with reduced attribute numbers and convergence time compared to the other two methods.

For rough neural network testing results are benchmarked with results of conventional pure neural networks. In both cases training to test the set percentage ratio was 80:20. The selected number of reducts for both cases was 20. Table 3 summarizes results. The value for  $\alpha$  is 0.4 based on the graph in Fig. 9.

Table 3 shows that classification accuracy of the neural network increased with fewer iterations proving that the introduction of rough neurons significantly increases the network performance.

**References:**

- [1] Z. Pawlak, "ROUGH SETS- Theoretical Aspects of Reasoning about data," Kluwer Academic Publishers, pp. 1-43, 1992.
- [2] J. W. Grzymala-Busse, "Introduction to Rough Set Theory and Applications".
- [3] Y. Hassan, E. Tazaki, S. Egawa, and K. Suyama, "Rough Neural Classifier," IEEE SMC, WAID5, 2002.
- [4] R. W. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition," Pattern Recognition Letters 24, pp. 833-849, Elsevier Science, 2003.
- [5] R. C. Gonzalez, R. E. Woods, and S. L. Eddiins, "Digital Image Processing using MATLAB," First Impression, PEARSON Education, pp. 348-497, 2006.
- [6] G. Vamvakas , "Optical Character Recognition for Handwritten Characters," National Center for Scientific Research "Demokritos" Athens - Greece, Institute of Informatics and Telecommunications and Computational Intelligence Laboratory (CIL), 2008.
- [7] R. Seethalakshmi, T. R. Sreeranjani, T. Balachandar, A. Singh, M. Singh, R. Ratan, and S. Kumar, "Optical Character Recognition for printed Tamil text using Unicode," Journal of Zhejiang University SCIENCE, 2005.
- [8] N. A. Shaikh, Dr. Zubair, and A. Shaikh, "A Generalized Thinning Algorithm for Cursive and Non-Cursive Language Scripts," 9th Int. Multitopic Conf., IEEE INMIC, 2005.
- [9] J. Hu, D. Yu, and H. Yan, "Algorithm for stroke width compensation of handwritten characters," Electronics Letters Online No19961501, 1996.
- [10] C. Jou, T.-Y. Hsiao, and H.-C. Lee, "Handwritten Numeral Recognition based on Reduced features extraction and Fuzzy Membership function," workshop on Artificial Intelligence, 2006.
- [11] S. N. Sivanandam, S. Sumathi and S. N. Deepa, "Introduction to Neural Networks using Matlab 6.0," first edition, Tata MCGraw Hill, pp. 531-536, 2006.
- [12] S. Haykins, "Neural networks," 2e, Pearson Education Publication, 1999.
- [13] H. Su and Q. Li, "Fuzzy Neural Classifier for Fault Diagnosis of Transformer Based on Rough Sets Theory," Electrical Machines and Systems, ICEMS 2005. Proc. of the Eighth Int. Conf. on pp. 2223-2227, 2005.
- [14] G. Xin, Y. Xiao, and H. You, "Discretization of continuous interval-valued attributes in rough set theory and its application," Proc. of the Sixth Int. Conf. on Machine Learning and Cybernetics, Hong Kong, 19-22 August, 2007.
- [15] A. Kusiak, "Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing," IEEE transactions on electronics packaging manufacturing, Vol.24, No.1, pp. 44-50, January 2001.
- [16] A. G. Kothari, "Data Mining Tool for Semiconductor Manufacturing Using Rough Neuro Hybrid approach," Proc. of Int. conf. on Computer Aided engineering- CAE-2007, IIT Chennai, 13-15 December, 2007.
- [17] Sandeep and Mayoraga, "Rough set based Neural Network Architecture," Int. Joint conf. on neural networks. 2006 Vancouver, BC, Canada, 2006.
- [18] P. Lingras, "Rough Neural Network." In Proc. of the 6th Int. Conf: on Information Processing and Management of Uncertainty, Granada, pp. 1445-1450, 1996.



**Name:**  
Ashwin Kothari

**Affiliation:**  
Sr. Lecturer & Research Scholar, Department of Electronics and Computer Science, Visvesvaraya National Institute of Technology, Nagpur (VNIT-Nagpur)

**Address:**  
Department of Electronics & CS, VNIT Campus, South Ambazari Road, Nagpur, (M.S.), India-440010

**Brief Biographical History:**  
1994 Received B.E.-Electronics, Nagpur University  
2005 Received M.Tech. Electronics, Nagpur University

**Main Works:**

- Intelligent Energy Monitoring using Unsupervised Neural Network based Pattern Recognition.
- FPGA Implementation of An Unsupervised Neural Network Based Pattern Classifier.

**Membership in Academic Societies:**

- Inter national Rough Set Society (IRSS)
- International Association of engineers (IAENG)
- Indian Society for Technical Education (ISTE)



**Name:**  
Avinash Keskar

**Affiliation:**  
Dean R&D, Professor - Department of Electronics and Computer Science, Visvesvaraya National Institute of Technology, Nagpur (VNIT-Nagpur)

**Address:**  
Department of Electronics & CS, VNIT Campus, South Ambazari Road, Nagpur, (M.S.), India-440010

**Brief Biographical History:**  
1979 Received B.E.-Electrical, Nagpur University  
1983 Received M.E. Electrical, IISC Bangalore  
1997 Received Ph.D., Nagpur University

**Main Works:**

- Intelligent Transportation System - Efficient Control and Management based on Soft Computing Tools.
- Real Time Embedded Application In Image Processing.
- Computer Simulation of Neuro Evolutionary Technique for Performance of Single Phase Power Electronic Circuits.
- Intelligent Face Recognition Techniques: A Comparative Study.
- FPGA implementation of Fuzzy Controllers.

**Membership in Academic Societies:**

- Senior Member, Institution of Electrical and Electronics Engineers (IEEE)
- Fellow, The Institution of Electronics & Telecommunication Engineering. (IETE)
- Fellow, Institution of Engineers
- Member, Indian Society for Technical Education (ISTE)