Chapter 10
Rough Set Approaches to Unsupervised Neural Network Based Pattern Classifier

Ashwin Kothari and Avinash Keskar

Abstract Unsupervised neural network based pattern classification is a widely popular choice for many real time applications. Such applications always face challenges of processing data with lot of consistency, inconsistency, ambiguity or incompleteness. Hence to deal with such challenges a strong approximation tool is always needed. Rough set is one such tool and various approaches based on Rough set, if are applied to pure neural (unsupervised) pattern classifier can yield desired results like faster convergence, feature space reduction and improved classification accuracy. The application of such approaches at respective level of implementation of neural network based pattern classifier for two case studies are discussed here. Whereas more emphasis is given on the preprocessing level based approach used for feature space reduction.

Keywords Discernibility · Feature extraction · Pattern classification · Reducts · Rough neuron · Rough sets · Unsupervised neural network

10.1 Introduction

Rough sets theory exploits the inconsistency and hidden patterns present in the data. Rough sets have been proposed for a variety of applications. In particular, the rough set theory approach seems to be important for artificial intelligence and cognitive sciences, especially for machine learning, knowledge discovery, data mining, expert systems, approximate reasoning and pattern recognition. Artificial Neural Networks in the most general form aim to develop systems that functions similar to the human brain. The nature of connections and the data exchange in the

A. Kothari (✉)
Senior Lecturer, Department of Electronics & Computer Science, VNIT, Nagpur, India
e-mail: agkothari72@rediffmail.com

A. Keskar
Professor and Dean R&D, Department of Electronics & Computer Science, VNIT, Nagpur, India

S.-I. Ao et al. (eds.), *Advances in Machine Learning and Data Analysis*,
Lecture Notes in Electrical Engineering 48, DOI 10.1007/978-90-481-3177-8_10,
© Springer Science+Business Media B.V. 2010

network depend on the type of application. Rough sets and Neural Networks can be combined, as they would be effective in cases of real world data, which are ambiguous, imprecise, incomplete and error prone.

Here two rough-neuro based hybrid approaches are proposed for classification by unsupervised ANN. As the rough set theory can be used for reducing the input feature space for the neural network doing the classification, first approach is suggested at preprocessing level. The second approach respectively explores the application of rough set theory for architectural modifications in the neural network. The first case study of printed character recognition has been undertaken to establish that a Rough-Neuro Hybrid approach has reduced dimensionality and hence also has resulted in lesser computations as compared to the pure neural approach. The data set used consists of characters A–Z, in 18 different fonts. Whereas second case study is focused on application of above discussed approaches for hand off prediction for cellular communications. The results for pattern classification using a Pure Neural approach have been used for benchmarking for both cases. Hence nature of information system (*IS*) and futures used in both cases along with the steps of image preprocessing and Feature extraction are discussed in initial sections. As the values obtained are continuous, the discretization steps used are explained in the following section. The feature space reduction and rough hybrid approach are discussed in the subsequent sections. The last section presents the results and conclusions.

10.1.1 Basics of Rough Set Theory

In this section some of the terminologies related to the basics of rough set theory and frequently used in the subsequent sections are explained [1].

10.1.1.1 Information System (IS)

The basic vehicle for data representation in the rough set framework is an information system. An information system is in this context a single flat table, either physically or logically in form of a view across several underlying tables. We can thus define an information system I in terms of a pair (*U, A*), where *U* is a non-empty finite set of objects and *A* is a non-empty finite set of attributes. The input feature labels are termed as condition attributes whereas class label is termed as decision attribute. The Table 10.1 shown below is the sample of *IS* used for second case study. The seven columns or input features namely BST2, BST3, BST4 (base station id), RSS (received signal strength), TMSTMP (time stamp), SPEED (speed of user motion) and TIME (time) are the condition attributes while the output class column CLASS is the decision attribute.

Table 10.1 Sample information system (IS)

BST2	BST3	BST4	RSS	TMSTMP	SPEED	TIME	CLASS
5	6	5	80	20	50	10	1
1	1	6	75	36	28	10	3
4	3	2	72	26	23	10	5
4	4	4	72	18	49	10	4
5	5	3	81	34	32	10	2

10.1.1.2 Reducts

The reduced sets of attributes are called as Reducts. Reducts derived out of input vectors using rough set postulations ease the process of making predictions and decision making which in turn gives improved classification with reduced dimensionality of feature space. Whereas theoretically reducts are defined as follows,

A subset B of set A is a reduct if and only if,

- $B^* = A^*$.
- B with this property i.e. $(B - \{a\})^* \neq A^*$ for all $a \in B$, * is the partition on U because of indiscernibility.

Once reducts are known, rules can be easily generated for classification.

10.1.1.3 Core

Cores are the prime attributes or Indispensable condition attributes. Core is the set of all those attributes, which are essential for classification between two classes, and there is no alternative for those attributes. If core is not included in the reducts then efficiency dramatically decreases. For example in the above shown sample *IS*, BST2, BST3 and BST4 are some of the core attributes.

10.1.1.4 Discernibility Matrix

It is an information system I defines a matrix MA called a discernibility matrix. Each entry MA (x, y) which is subset of A consists of the set of attributes that can be used to discern between objects x, y which are elements of U.

10.2 Image Processing and Feature Extraction for the First Case Study

The original data set is subjected to a number of preliminary processing steps to make it usable by the feature extraction algorithm. Pre-processing aims at producing data that is easy for the pattern recognition system to operate accurately. The main

objectives of pre-processing [2] are: binarization, noise reduction, skeletonization, boundary extraction, stroke width compensation [3], truncation of redundant portion of image and resizing to a specific size. Image binarization consists of conversion of a gray scale image into a binary image. Noise reduction is performed using morphological operations like dilation, erosion etc. Skeletonization of the image gives an approximate single-pixel skeleton, which helps in the further stages of feature extraction and classification. The outermost boundary of the image is extracted to further obtain the boundary related attributes such as chain codes and number of loops. Stroke width compensation is performed to repair the character strokes, to fill small holes and to reduce uneven nature of the characters. The white portion surrounding the image can create noise in the feature extraction process and also increases the size of image unnecessarily. Truncation is performed to remove this white portion. In the end, the image is resized to a pre-defined size: 64×64 pixel in this case. All such results are indicated in Fig. 10.1 below. In feature extraction stage, each character is represented as a feature vector, which becomes its identity. The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements. The feature extraction process used consists of two types of features: statistical and structural [3–5]. The major statistical features used are: zoning, crossings, pixel density, Euler number, compactness, mean and variance. In zoning, the 64×64 character image is divided into 16×16 pixel parts and pixel density of each part is calculated individually. This helps in obtaining local characteristics rather than global characteristics and is an important attribute for pattern recognition. Crossings count the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image. In Fig. 10.2 there are six vertical crossings (white to black and black to white) and four horizontal crossings in both the upper and lower part. Pixel Density is calculated over the whole 64×64 image. Euler number of an image is a scalar whose value is the total number of objects in the image minus the total number of holes in those objects. Euler number is also calculated for each image. Structural features are based on topological and geometrical properties of the character, such as aspect ratio, loops, strokes and their directions etc. The boundary of the image is obtained and chain code is calculated for it. Then the number of ones, twos till number of eights is calculated. The number of loops present in a character is also obtained.

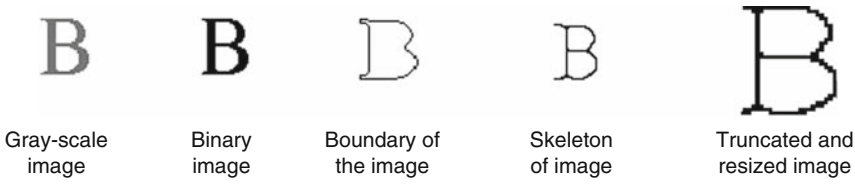

Fig. 10.1 Various stages of image processing

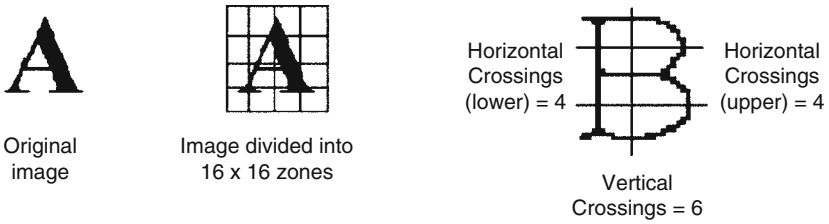

Fig. 10.2 Image segmentation and feature extraction

10.3 Discretization

The solution to deal with continuous valued attributes is to partition numeric variables into a number of intervals and treat each such interval as a category. This process of partitioning continuous variables is usually termed as discretization. Data can be reduced and simplified through discretization. In case of continuous valued attributes, large number of objects are generated with a very few objects mapping into each of these classes. The worst case would be when each value creates equivalence with only one object mapping into it. The discretization in rough set theory has particular characteristic, which should not weaken the indiscernibility ability. A variety of discretization methods have been developed.

10.3.1 Algorithm

Consider an information system $IS = (U,\ A,\ V,\ f)$ where U: is the universal sets containing all objects i.e. $U = \{x_1,\ x_2,\ x_3, \ldots..,\ x_n\}$, 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 contains the sets of values each condition attribute can take; f is a function between the element in U and its value, the value of object x_i to the attribute a is $a(x_i)$. The process of discretization is to find the sets of cut points for each attribute and hence discretize each of them. For an attribute a, V_a the set containing the values the attribute can take. Cuts are nothing but partitions of the set V_a i.e. $V_a = [c\ d]$, where $[c\ d]$ is the interval of the continuous valued attribute. Then a partition: $c < p_1 < p_2 \ldots \ldots < p_m < d$, where the set $\{p_1, p_2, \ldots..p_m\}$ forms the set of cut points which divides the interval $[c\ d]$ into m intervals without intersection: $[c\ p_1)$, $[p_1\ p_2)$, $[p_2\ p_3) \ldots..(p_m d]$ and the continuous values of attribute a turn out to be $m + 1$ discrete values: $V_1,\ V_2,\ V_3,\ \ldots\ \ldots,\ V_{m+1}$ by the following equations [6]:

$$\{\mathrm{V}_1, \quad \text{if a(x)} \leq \mathrm{p}_1\} \tag{10.1a}$$

$$\mathrm{V}(x) = \{\mathrm{V}_i, \quad \text{if } \mathrm{p}_{i-1} < \mathrm{a(x)} \leq \mathrm{p_i}, \mathrm{i} = 1, 2, 3. \ldots, \mathrm{m}\} \tag{10.1b}$$

$$\{\mathrm{V}_{\mathrm{m}+1}, \quad \text{if a(x)} > \mathrm{p_m}\} \tag{10.1c}$$

10.3.2 Steps to Obtain Cuts

To search the cuts points we make use of the discernibility matrix. The discernibility matrix of a given decision table is defined as:

$(M_{\{d\}}(i,j))_{nxn}$ Where

$$M_{\{d\}}(i,j) = \{\{a_k | a_k(x_i) \neq a_k(x_j), \text{ for all } a_k \text{ in } C\} \text{ when } d(x_i) \neq d(x_j)\}$$
$$\{0, d(x_i) = d(x_j)\} \tag{10.2}$$

10.3.3 Steps for Algorithm

1. Construct the discernibility matrix for the given table
2. For $i = 1$ *to* t
3. For all $1 \leq j \leq k \leq n$, if the attribute a_i is a member of the discernibility matrix entry, construct the cuts interval $[a_i(x_j)\ a_i(x_k)]$
4. For every set of intersecting intervals, construct the cut point as:
   $(p_j)_i = \{\max(\text{lower bounds}) + \min(\text{upper bounds})\}/2$. (3)
5. Discretize attribute a_i according to the cuts obtained
6. Next i
7. End

The Figs. 10.8 and 10.9 show an undiscretized table and the discretized table using the above-discussed algorithm. It is observed that the continuous interval valued attributes are discretized without affecting their ability to discern. It can be further seen that this helps in attribute reduction also.

10.4 Reduction of Attributes

Use of rough sets theory in the preprocessing stage results into dimensionality reduction and optimized classification with removal of redundant attributes. Also, neural network is the most generalized tool for pattern recognition and has capability of working in noisy conditions also. Here a new Rough-Neuro Hybrid Approach in the pre-processing stage of pattern recognition is used. In this process, a set of equivalence classes, which are indiscernible using the set of given attributes, are identified. Only those attributes are kept which preserve the indiscernibility relation and the redundant ones are removed, as they do not affect the classification. A reduction is thus resulting in a reduced set of attributes, which classifies the data set with the same efficiency as that of the original attribute set.

10.4.1 Steps for Finding Reduced Set of Attributes

Consider an information system $IS = (U,\ A,\ V,\ f)$ where U: is the universal sets containing all objects i.e. $U = \{x_1,\ x_2,\ x_3, \ldots ..,\ x_n\}$, 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 contains the sets of values each condition attribute can take; f is a function between the element in U and its value, the value of object x_i to the attribute a is $a(x_i)$. The total number of condition attributes is m i.e. $|C| = m$. The number of decision classes is t i.e. $|V_d| = t$. Discernibilty matrix $(t \times t)$, whose entries contain the relative significance of each attribute. In this method the relative significance of each attribute in discerning between two compared classes x and y is as given by:

$$\mathrm{P(x)_{x,y}|a_i} \tag{10.3}$$

where $P(x)_{x,y}|a_i$ is the probability of an object belonging to class x given the only information as attribute $a_i (i = 1, 2, 3, \ldots, m)$ when discerning the objects of x from y. If the value turns out to be 1, then the particular attribute is the most significant and if the value turns out to be 0, then the particular attribute is the least significant. The probability described is subjective and there can be other viewpoint. The rough sets deal with uncertainty of data sets or information granules.

10.4.2 Algorithm for Finding Reducts

The algorithm used is as follows:

1. Obtain the information system of which the feature space is to be reduced.
2. The discrenibility matrix (t × t) is to be constructed and entry for each attribute is given by the above method.
3. The relative sum for each attribute is obtained i.e. the contribution of each attribute over the table is summed up.
4. The most significant contributors based on the relative sum are selected according to the requirement or set threshold.

10.5 Rough Neuro Based Hybrid Approach and the Experimentation Done

As discussed in the introduction Rough set can deal with consistent, inconsistent and even incomplete type of data. Hence the methodology proposed must be tested for all the above-mentioned types of data. Hence to examine the outcome for consistent type of data case study one is considered in which the approaches are tested for printed character recognition. While for testing outcome for inconsistent or incomplete data case study 2 is considered in which approaches are used for hand off prediction. In the coming era of cellular communications the cell size will reduce

and in situations like mobile user using mass rapid transit systems, the overall performance of the system can be improved if the network can predict the next cell to which user is going to move. Because dynamic cellular traffic conditions and numerous path profiles of moving mobile users data generated will be highly inconsistent in nature.

For both the cases the results achieved by the hybrid approach are compared with the pure neural approach. The type of ANN used in both the case studies for pure and rough-neuro hybrid approach is unsupervised with two layers. When pure neural approach is used for the first case study, the input layer consisting of 31 nodes for 31 input features, fed for classification is used. Whereas the output layer contains 26 nodes for 26 output classes in which the input samples are to be classified.

Whereas for second case study the input layer consists of seven nodes for seven input features and the output layer contains six nodes representing six possible destinations for each current cell of the user location for predicting the handoffs. The data table used for training is formulated with the steps explained earlier. Many such tables with variations in data patterns were used for training and testing in 80:20 proportion. The training algorithm used is (of competitive learning type) Winner take all [7–10]. Rough set approach used for preprocessing or attributes reduction and the downsized set of attributes can be fed to the neural classifier as shown below in Fig. 10.5. Reducts derived out of input vectors using rough set postulations ease the process of making predictions and decision making which in turn gives improved classification with reduced dimensionality of feature space. For estimation of reducts discernibility matrix is first calculated, weighted contributions obtained and the significant contributors are selected as per the fixed threshold. Thus rough set mainly exploits discernibility and hidden inconsistency in the data. The results shown here are for the same table used earlier for the first approach in Fig. 10.3 and Fig. 10.4.

Attribute→	1	2	3	4	5	6	7	8	9	10
1	0.056274	4	2	2	2	1	0	69	69	0
2	0.1073	4	3	2	2	2	106	136	88	110
3	0.066528	–1	2	2	2	0	70	103	64	109
4	0.104919	–1	3	3	3	1	201	122	77	0
5	0.120117	1	4	2	2	0	247	112	112	112
6	0.111206	5	3	3	1	0	110	242	44	202
7	0.10791	4	4	2	2	0	19	172	141	87
8	0.136475	8	1	2	2	0	160	132	132	164
9	0.25	8	1	2	2	0	256	256	256	256
10	0.057068	2	1	1	1	0	0	0	0	176
11	0.1297	6	1	2	2	0	160	144	58	29
12	0.097595	5	2	1	2	0	88	232	16	0
13	0.137634	5	1	4	4	0	117	193	32	256
14	0.114502	4	1	3	4	0	236	34	0	160
15	0.09198	4	2	2	3	1	48	152	141	126
16	0.101807	3	3	3	2	1	229	112	120	137
17	0.092346	4	3	2	4	1	63	134	132	91
18	0.108459	5	2	2	3	1	200	128	133	100
19	0.08783	2	3	1	2	0	103	98	102	101
20	0.062134	0	2	1	1	0	112	157	157	112
21	0.077271	0	1	2	2	0	144	0	0	144

Fig. 10.3 Undiscretized data for case study-1

Attribute→	1	2	3	4	5	6	7	8	9	10
1	1	15	3	3	3	3	1	21	29	1
2	70	15	5	3	3	5	30	55	34	42
3	11	5	3	3	3	1	10	34	26	41
4	67	5	5	5	5	3	73	44	32	1
5	81	9	7	3	3	1	85	37	42	45
6	76	17	5	5	1	1	33	81	18	81
7	72	15	7	3	3	1	5	69	66	24
8	94	22	1	3	3	1	58	51	58	71
9	99	22	1	3	3	1	86	82	86	87
10	2	11	1	1	1	1	1	1	1	77
11	91	19	1	3	3	1	58	60	23	5
12	55	17	3	1	3	1	20	79	10	1
13	95	17	1	7	7	1	40	72	15	87
14	79	15	1	5	7	1	81	10	1	69
15	46	15	3	3	5	3	7	62	66	51
16	63	13	5	5	3	3	80	37	50	56
17	48	15	5	3	7	3	9	53	58	28
18	73	17	3	3	5	3	72	48	60	35
19	38	11	5	1	3	1	27	30	39	36
20	7	7	3	1	1	1	36	63	72	45
21	21	7	1	3	3	1	50	1	1	61

Fig. 10.4 Discretized data for case study-1

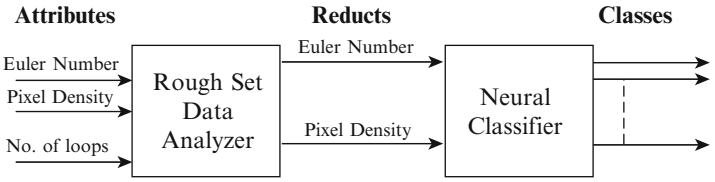

Fig. 10.5 Use of rough set at preprocessing level as first approach

The second approach suggests modification to the architecture of the two layer neural network. In pure neural approach each input layer node is connected with each node of output layer. After computation of reducts we easily know the most significant contributors (input features) to the classification process [11, 12]. Hence depending on the set of reducts obtained the links related to the insignificant features can be removed which further results in lesser computational overheads.

10.6 Results

For both the case studies the training to testing proportion was 80:20 in percentage. The *IS* for case study one has condition attributes of two types structural and statistical which respectively brings consistency and inconsistency to the data. In the second case study the approaches were studied for data of 1,000 user path profiles with different combinations of timestamps and received signal strengths. Hence *IS* for second case study is highly inconsistent.

10.6.1 Results for Case Study-1

The data set generated is given to a pure neural network as described earlier with learning rate $\alpha = 0.4$ and the network converges in almost 1,000 iterations. The attribute set was then reduced by the proposed method and fed into the neural network with the set learning rate. The graphs shown in Fig. 10.6 is achieved for reduction of the attributes space using the above discussed method. The horizontal axis shows the number of resulted reductions and the vertical axis shows the corresponding classification accuracies. From this graph, we conclude that the attribute reductions result in the optimum classification accuracy. Thus the total 31 atributes are reduced to 18 as maximum reduction, 21 as moderate reduction and 26 as minimum reduction. It is found that the minimum classification accuracy of 92.5% is much comparable with the accuracy achieved for pure neural approach using all 31 attributes. The number of reducts to be used hence always is trade off between feature space diamensionality to handled and classification accuracy. This is because too much reduction results in loss of information also. It is observed that at the point of dimentionality reduction certain structural attributes which contribte to the discernability of the system are also neglected (for example numberof loops in the character). Forceful addition of such feature to the reducts yeilds a better efficency. This is shown in Table 10.2. Note that the structural features are added to the reduced features which are 21 in number.

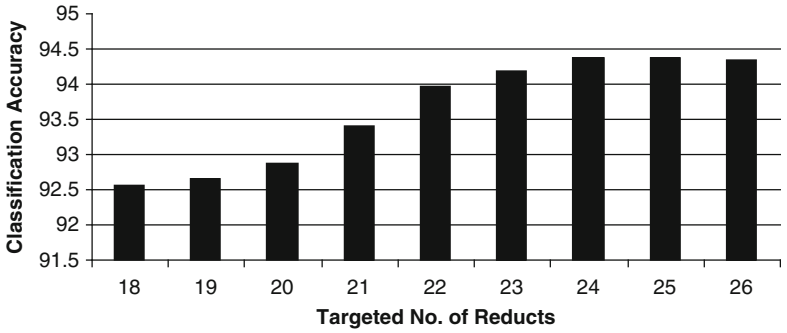

Fig. 10.6 Different classification accuracy values for respective number of reducts

Table 10.2 Classification accuracy for different approaches for case study-1

The kind of data given to the network	Entire attribute space to pure neural network	Reduced attributes from the reducts algorithm (21 No.)	Reduced atrributes (21 No.) and loops
Classification accuracy	92.37%	93.76%	94.39%

10.6.2 Results for Case Study-2

For case study two as explained earlier data of 1,000 user path profiles are used. This results in *IS* with seven condition attributes. BST2, BST3, BST4 (base station ids), RSS (received signal strength), TMSTMP (time stamp), SPEED (speed of user motion) and TIME (time). The preprocessing approach results in two sets of reducts with three and four features respectively. Hence for verification of the second approach (rough) neural network with missing links was used. The respective values for classification accuracy are shown in Table 10.3.

Also from the error curves drawn in Figs.10.7–10.9 respectively for various methods, it is seen that pure neural network with reducts and neural network designed with rough set philosophy by missing links, converge much faster than the conventional unsupervised neural network.

Table 10.3 Classification accuracy for different approaches for case study-2

The kind of data given to the network	Entire attribute space to pure neural network	Reduced attributes from the reducts algorithm (3 No.)	Reducts attribute with four missing links in rough neural network
Classification accuracy	98.97%	99.06%	98.23%

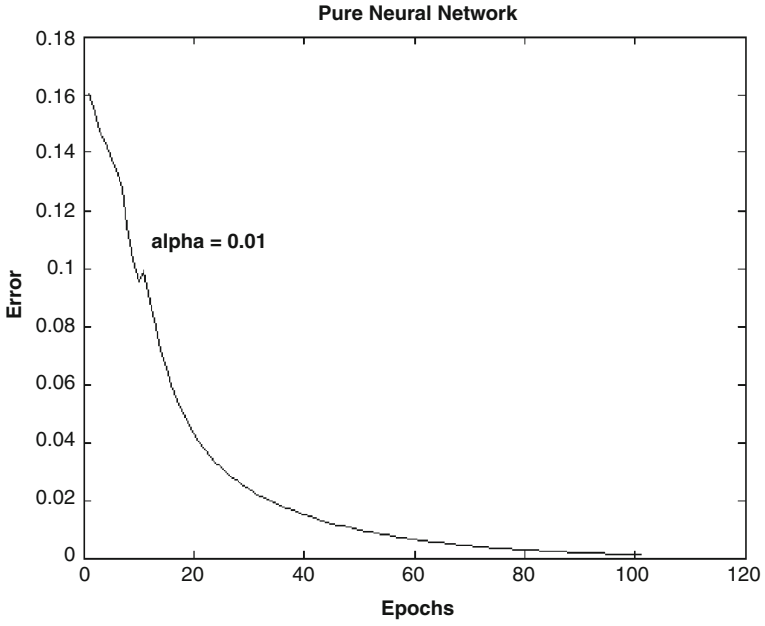

Fig. 10.7 Error versus epochs for pure neural approach for case study-2

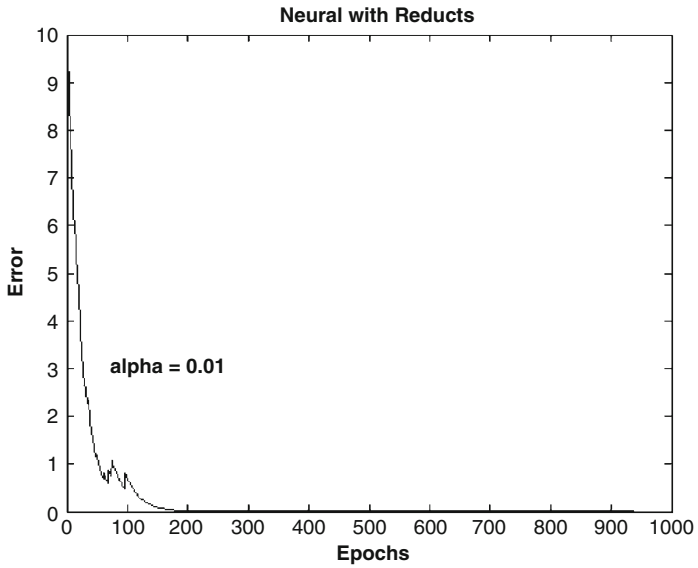

Fig. 10.8 Error versus epochs for neural network fed with reducts for case study-2

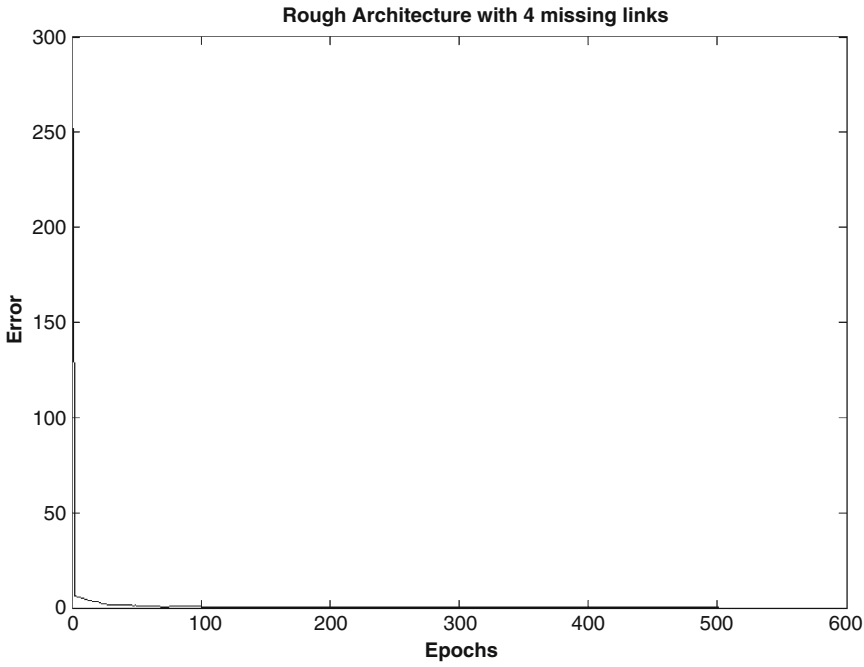

Fig. 10.9 Error versus epochs for neural network with four missing links

10.7 Conclusion

The data set generated contains both the consistent and inconsistent values. According to the rough sets theory consistent attribute values contribute less towards classification but on observation in case study-1, the important structural features such as the loops and crossings when included improves the efficiency of classification. Thus using rough set theory and analysing the importance of features from classification point of view efficient reduction algorithms can be developed. Also it has been observed from the results of case study-2 that for reduct based training or network design we get improved classification accuracy with faster convergence. In future such algorithm can be tested for rough neuron based networks where input data is split in to two parts for specially designed rough neuron.

Acknowledgment The support of Dr. A.P. Gokhale and the team of students consisting of Mr. Bharthan Balaji, Mr. Pradeep Dhananjay, Ms. Y.T. Vasavdatta and Ms. Deepti pant is highly acknowledged for carrying out the experimentation and acquiring of data in the lab.

References

1. Zdzislaw Pawlak: "ROUGH SETS – Theoretical Aspects of Reasoning About Data", 1992, Kluwer, Dordrecht, pages 1–43.
2. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddiins: "Digital Image Processing Using MATLAB", First Impression, 2006, Pearson Education, NJ, USA, pages 348–497.
3. Jianming Hu, Donggang Yu, Hong Yan: "Algorithm for stroke width compensation of handwritten characters", Electronics Letters Online No. 19961501.
4. Hongsheng Su, Qunzhan Li: "Fuzzy Neural Classifier for Fault Diagnosis of Transformer Based on Rough Sets Theory": IEEE, CS, 2223 to 2227.
5. Giorgos 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).
6. Jiang-Hong Man: "An Improved Fuzzy Discretization Way for Decision Tables with Continues Attributes", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19–22 August 2007.
7. S.N. Sivanandam, S. Sumathi, S.N. Deepa: "Introduction to Neural Networks Using Matlab 6.0" first edition, 2006, Tata MCGraw Hill, OH, USA, pages 531–536.
8. J.S.R. Jang, C. T Sun, E. Mizutani: "Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", First edition reprint 2005, Pearson education, NJ, USA, pages 327–331.
9. Xian-Ming Huang, Ji-Kai Yi, Yan-Hong Zhang: "A method of constructing fuzzy neural network based on rough set theory", International Conference on Machine Learning and Cybernetics, 2003, Publication Date: 2–5 Nov. 2003, Volume: 3, pages 1723–1728.
10. Ashwin G. Kothari: "Data Mining Tool for Semiconductor Manufacturing Using Rough Neuro Hybrid approach", Proceedings of International Conference on Computer Aided engineering-CAE-2007, IIT Chennai, 13–15 December 2007.
11. C. Sandeep, R. Mayoraga: "Rough Set Based Neural Network Architecture", International Joint Conference on Neural Networks, Vancouver, BC, Canada, 2006.
12. Pawan Lingras: "Rough Neural Network," Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty, Granada, pages 1445–1450, 1996.