Towards Rough Neural Computing Based on Rough Membership Functions: Theory and Application

J.F. Peters[1], A. Skowron[2], L.Han[1], S.Ramanna[1]

[1] Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 2N2 Canada
{jfpeters,liting,ramanna}@ee.umanitoba.ca
[2]Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Abstract. This paper introduces a neural network architecture based on rough sets and rough membership functions. The neurons of such networks instantiate approximate reasoning in assessing knowledge gleaned from input data. Each neuron constructs upper and lower approximations as an aid to classifying inputs. Rough neuron output has various forms. In this paper, rough neuron output results from the application of a rough membership function. A brief introduction to the basic concepts underlying rough membership neural networks is given. An application of rough neural computing is briefly considered in classifying the waveforms of power system faults. Experimental results with rough neural classification of waveforms are also given.

1 Introduction

A form of rough neural computing based on based on rough sets, rough membership functions, and decision rules is introduced in this paper. Rough sets were introduced by Pawlak [1], and elaborated in [2]-[3]. Rough membership functions were introduced by Pawlak and Skowron [4]. Studies of neural networks in context of rough sets are extensive [5]-[12]. This paper considers the design and application of neural networks with two types of rough neurons: approximation neurons and decider neurons. The term *rough neuron* was introduced in 1996 [5]. In its original form, a rough neuron was defined relative to upper and lower bounds and inputs were assessed relative to boundary values. More recent work considers rough neural networks (rNNs) with neurons, which construct rough sets and output the degree of accuracy of an approximation [10]-[11], which is based on an earlier study [9]. The study of rough neurons is part of a growing number of papers on neural networks based on rough sets. Rough-fuzzy multilayer perceptrons (MLPs) in knowledge encoding and classification were introduced in [12]. Rough-fuzzy neural networks have recently been also used in classifying the waveforms of power system faults [10]. Purely rough membership function neural networks (rmfNNs) were introduced in [11] in the context of rough sets and the recent introduction of rough membership functions [4]. This paper considers the design of rough neural networks based on rough membership functions, and hence this form of network is called a rough membership neural network (rmNN). Preliminary computations in a rmNN are carried out with a layer of approximation neurons, which construct rough sets and where the output of each approximation neuron is computed with a rough membership function. The values produced by a layer of approximation neurons are used to construct a condition vector. Each new condition vector provides a stimulus for a decider neuron in the output layer of a rmNN. A decider neuron enforces rules derived from decision tables based on rough set theory. A decision table reflects our knowledge of the world at a given time. This knowledge is represented by condition vectors and corresponding decisions. Information granules in the form of rules are extracted from decision tables using rough set methods. Discovery of decider neuron rules stems from an application of the rule derivation method given in [13]-[14]. This characterization of a decider neuron is based on the identification of information granules based on decision rules [15]. Each time a decider neuron is stimulated by a new condition vector constructed by the approximation neuron layer, it searches for the closest fit between each new condition vector and existing condition vectors extracted from a decision table. Decider neurons are akin to what are known as logic neurons described in [16].

2 Rough Membership Functions

A brief introduction to the basic concepts underlying the construction of rough membership neural networks is given in this section. A rough membership function (rm function) makes it possible to measure the degree that any specified object with given attribute values belongs to a given set X [4], [21]. A rm function μ_X^B is defined relative to a set of attributes $B \subseteq A$ in information system S = (U, A) and a given set of objects X. The equivalence class $[x]_B$ induces a partition of the universe. Let $B \subseteq A$, and let X be a set of observations of interest. The degree of overlap between X and $[x]_B$ containing x can be quantified with the rough membership function:

$$\mu_X^B : U \to [0,1] \text{ defined by } \mu_X^B(x) = \frac{\left|[x]_B \cap X\right|}{\left|[x]_B\right|}$$

3 Design of Rough Neural Networks

Neural networks are collections of massively parallel computation units called neurons. A neuron is a processing element in a neural network.

3.1 Design of Rough Neurons

Typically, a neuron y maps its weighted inputs from $\mathbf{R}^n$ to [0, 1] [16]. Let T be a decision table (X, A, {d}) used to construct $\underline{B}X$, $\overline{B}X$, and let $X \subseteq Y$. A selection of different types of neurons is given in Table 1: common neurons, approximation neurons and decider neurons.

Table 1. Selection of Different Types of Neurons

Common Neuron	**Upper Approximation neuron**
$y = f\left(\sum_{i=1}^{n} w_i x_i + \vartheta\right)$, where input x_i has connection (weight) w_i, which denotes a modifiable neural connection, and bias ϑ [16]	$y_x = f(\overline{B}X, \underline{B}X, X)$
	Lower approximation Neuron
	$y = f(\underline{B}X, X)$
	Decider Neuron
	$y_{rule} = \min(e_i, d_i([\mu_X^{A_1}(x) \ldots \mu_X^{A_n}(x)]))$, with condition granule $[\mu_X^{A_1}(x) \ldots \mu_X^{A_n}(x)]$

Let B, F, $[f]_B$ denote set of attributes, set of neuron inputs (stimuli), and equivalence class containing measurements derived from known objects, respectively. The basic computation steps performed by an approximation neuron are reflected in the flow graph in Fig. 1.

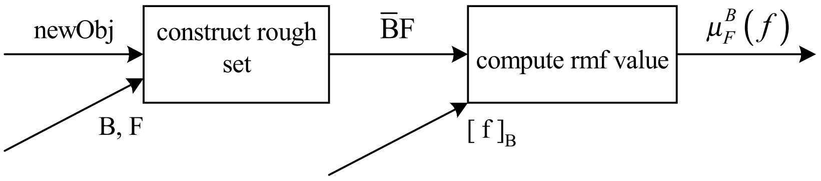

Fig. 1. Flow Graph for Basic Approximation Neuron Computation

An approximation neuron measures the degree of overlap of a set $[f]_B$ and $\overline{B}F$ representing certain as well as uncertain classifications of input signals. A flow graph showing the basic computations performed by a decider neuron is given in Fig. 2.

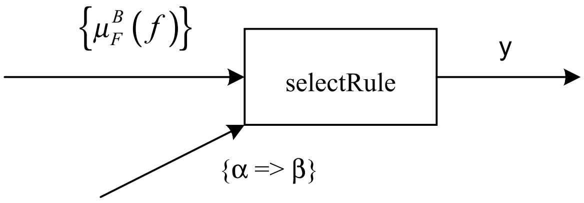

Fig. 2. Flow Graph for Decider Neuron

A decider neuron implements a selectRule algorithm.

```
Algorithm selectRule {
 input set {α=>β}                        //set of decision rules
 input vector [$c_{exp1}$, $c_{exp2}$, ..., $c_{expn}$];  //condition vector input { $\mu_F^B(f)$ }
 int chosenRule; //index used to identify decision rule
 float[ ] sum; //stores sum of differences | $c_{exp\,j} - c_{ij}$ |
 float bestMatch; //used to store value of best match
 int vectorSize = 2, i = 1, j = 1;
bestMatch = $\sum_{j=1}^{n} \left| c_{\exp j} - c_{1j} \right|$; chosenRule = 1;  //for vector $\alpha_1$
 while (vectorSize <= n) {
   sum[vectorSize] = $\sum_{j=1}^{n} \left| c_{\exp j} - c_{ij} \right|$;
   if (sum[vectorSize] < bestMatch) { chosenRule = vectorSize }
   vectorSize++; i++;                     // use i to select i^th condition vector
   j = 1;
  }//while
  return chosenRule;
} // end Algorithm selectRule
```

In Fig. 2, the set rmf = { $\mu_F^B(f)$ } consists of approximation neuron measurements in response to the stimulus provided a new object requiring classification. The elements of the set rmf are used by a decider neuron to construct an experimental condition vector α_{exp}. A second input to a decider neuron is the set R = {α=>β}. The elements of the set R are rules which have been derived from a decision table using rough set theory. After a decision rules has been selected, a decider neuron outputs $\min(e_i, d_i)$ where $d \in \{0,1\}$, and relative error $e_i = | c_{exp} - c_i |/c_i \in [0,1]$. In cases where d = 0, then $y_{rule} = \min(e_i, d_i) = 0$, and the classification is unsuccessful. If d = 1, then $y_{rule} = \min(e_i, d_i) = e_i$ indicates the relative error in a successful classification.

3.2 Rough Neural Network Example

By way of illustration, a rough neural network is constructed with two layers: input layer consisting of upper approximation neurons, and output layer with a single decider neuron (see Fig. 3). Using a sample of 61 fault files, a partial decision table has been constructed (see Table 2). Let v, i denote voltage, current, respectively. To complete the design of a decider neuron, rules are extracted from decision Table 2.

Table 2. Sample Power System Commutation Fault Decision Table

ac v error (a1)	phase i / i order (a2)	pole line v (a3)	6 pulse (a4)	phase i type (a5)	phase i ord (a6)	max phase i (a7)	**d**
0.0588	0	0.5	0	0	0	0	0
0.0588	0.06977	0.5	1	0.1875	0.05405	0.08571	1
0.0588	0.06977	0.5	1	0.1875	0.05405	0.08571	1

A sample of the rules derived from Table 2 using Rosetta [22] are as follows.

a1(0.058824) AND a5(0.187500) AND a7(0.000000) => d(0.00)
a1(0.058824) AND a5(0.187500) AND a7(0.085714) => d(1.00)

Rules like those given above are incorporated in a decider neuron repository (storage of rules associated with a decider neuron). In the experiments described in this section, each approximation neuron is defined relative to a single attribute such as AC disturbance (see a1 in Fig. 3). The decider neuron in Fig. 3 implements the selectRule algorithm to produce its output. The design of a particular decider neuron hinges on derivation of rules from a decision table reflecting our current knowledge of the world.

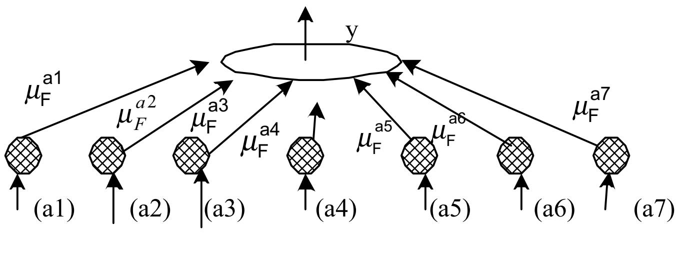

Fig. 3. Sample rough neural network

In the constructed networks, the weights are not primitive but they are functions of some other parameters like set of features (attributers). The relationships between the weight values and these other parameters are expressed in the paper by the rough membership function. This function allows to measure a degree in which B-indiscernibility classes are included in a given set (in the considered example in the upper approximation of one of the decision class). Hence, the process of tuning weights in the network should be connected with tuning of parameters on which these weights depend. In particular, in the considered example this can be related to searching for relevant feature set B of attributes.

3.3 Sample Verification

A comparison between the output from a rough neural network used to classify power system faults relative to 24 fault files and known classification of the sample fault data is given in Fig. 4.

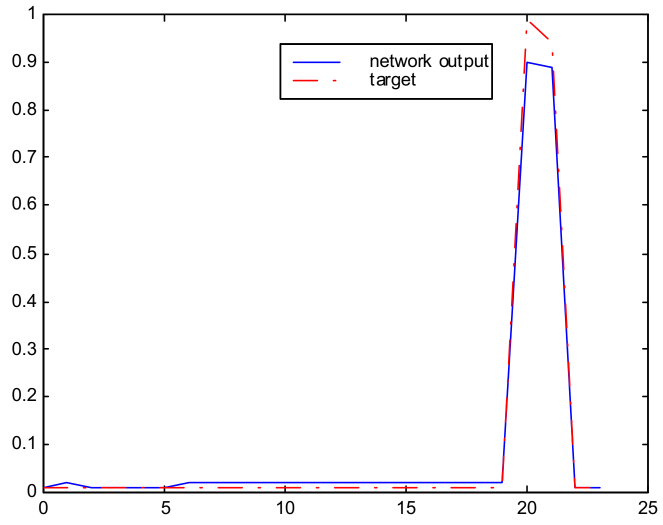

Fig. 4. Comparison of Rough Neural Network Output and Target Values

In all of the cases considered in Fig. 4, there is a close match between the target faults and the faults identified the neural network. Further, it should be observed that a total of eleven neural networks were used (one for each type of fault file) to generate the data used in Fig. 4, and carry out a complete classification of all fault files.

4 Concluding Remarks

Two basic types of rough neurons have been identified: approximation neurons and rule-based neurons. The output of an approximation neuron is a rm function value, which indicates the degree of overlap between an approximation region and some other set of interest in a classification effort. The output of rule-based neuron is a classification decision, which represents an assessment of the closeness of experimental data to a known feature of a feature space. A sample application of these neurons in a power system fault classification system has been given. We consider the problem of learning schemes of information granule construction. These schemes transform input granules into output ones. It is necessary to tune parameters in these schemes to obtain the output granules of satisfactory quality from input granules. One of the method of tuning these parameters can be based on finding function embedding these schemes into classical neural networks. Next known learning methods for neural networks can be applied. The weight values in such networks will reflect the inclusion degrees between granules. Hence the process of changing weights in neural networks should correspond to tuning degrees of granule inclusion. The paper presents an example of such situation.

References

1. Z. Pawlak, Rough sets, Int. J. of Computer and Information Sciences, Vol. 11, 1982, 341-356.
2. Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Boston, MA, Kluwer Academic Publishers.
3. Z. Pawlak. Reasoning about data--A rough set persepective. Lecture Notes in Artificial Intelligence 1424, L. Polkowski and A. Skowron (Eds.). Berlin, Springer-Verlag, 1998, 25-34.
4. Z. Pawlak, A. Skowron, Rough membership functions. In: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, NY, John Wiley & Sons, 1994, 251-271.
5. P.J. Lingras, Rough neural networks. In: Proc. of the 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'96), Granada, Spain, 1996, 1445-1450.
6. H.S. Nguyen, M. Szczuka, D. Slezak, Neural networks design: Rough set approach to real-valued data. In: Proc. of PKDD'97, Trondheim, Norway. Lecture Notes in Artificial Intelligence 1263, Berlin, Springer-Verlag, 1997, 359-366.
7. P. Wojdyllo, Wavelets, rough sets and artificial neural networks in EEG analysis. In: Proc. of RSCTC'99, Lecture Notes in Artificial Intelligence 1424, Berlin, Springer-Verlag, 1998, 444-449.
8. M.S. Szczuka, Refining classifiers with neural networks, International Journal of Intelligent Systems [to appear].
9. L. Han, R. Menzies, J.F. Peters, L. Crowe, High voltage power fault-detection and analysis system: Design and implementation. Proc. CCECE99, 1253-1258.
10. L. Han, J.F. Peters, S. Ramanna, R. Zhai, Classifying faults in high voltage power systems: A rough-fuzzy neural computational approach. In: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence 1711. Berlin: Springer, 1999, 47-54.
11. J.F. Peters, A. Skowron, Z. Surai, L. Han, S. Ramanna, Design of rough neurons: Rough set foundation and Petri net model, International Symposium on Methodologies for Intelligent Systems (ISMIS'2000) [submitted].
12. M. Banerjee, S. Mitra, S.K. Pal, Rough fuzzy MLP: Knowledge encoding and classification, IEEE Trans. Neural Networks, vol. 9, 1998, 1203-1216.
13. A. Skowron, C. Rauszer. The discernability matrices and functions in information systems. In: Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Slowinski, R. (Ed.), Dordrecht, Kluwer Academic Publishers, 1992, 331-362.
14. A. Skowron and J. Stepaniuk. Decision rules based on discernability matrices and decision matrices. In: Proc. Third Int. Workshop on Rough Sets and Soft Computing, San Jose, California, 10-12 November 1994.
15. A. Skowron, J. Stepaniuk, Information granules: Towards foundations of granular computing, Internation Journal of Intelligent Systems [to appear].
16. W. Pedrycz, Computational Intelligence: An Introduction, Boca Raton, CRC Press, 1998.

17. H.S. Nguyen. Discretization of real-valued attributes: Boolean reasoning approach. Doctoral Thesis, Faculty of Mathematics, Computer Science and Mechanics, Warsaw University, 1997.
18. H.S. Nguyen. Rule induction from continuous data: New discretization concepts. Proc. of the Third Joint Conf. on Information Sciences, Raleigh, N.C., 1-5 March 1997.
19. A. Skowron, J. Stepaniuk, Information granules in distributed environment. In: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence 1711. Berlin: Springer, 1999, 357-365.
20. A. Skowron, J. Stepaniuk. Constructive information granules. In: Proc. of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Berlin, Germany, 24-29 August 1997. Artificial Intelligence and Computer Science 4, 1997, 625-630.
21. J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, Rough sets: A tutorial. In: S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization: A New Trend in Decision-Making. Singapore: Springer-Verlag, 1999, 3-98.
22. Rosetta software system, http://www.idi.ntnu.no/~aleks/rosetta/