Design of Rough Neurons: Rough Set Foundation and Petri Net Model

J.F. Peters[1], A. Skowron[2], Z. Suraj[3], L. Han[1], and S. Ramanna[1]

[1] Computer Engineering, Univ. of Manitoba, Winnipeg, MB R3T 5V6 Canada
[2]Institute of Mathematics, Warsaw Univ., Banacha 2, 02-097 Warsaw, Poland
[3]Institute of Mathematics, Pedagogical Univ., Rejtana 16A, 35-310 Rzeszów, Poland
{jfpeters,liting,ramanna}@ee.umanitoba.ca

Abstract. This paper introduces the design of rough neurons based on rough sets. Rough neurons instantiate approximate reasoning in assessing knowledge gleaned from input data. Each neuron constructs upper and lower approximations as an aid to classifying inputs. The particular form of rough neuron considered in this paper relies on what is known as a rough membership function in assessing the accuracy of a classification of input signals. The architecture of a rough neuron includes one or more input ports which filter inputs relative to selected bands of values and one or more output ports which produce measurements of the degree of overlap between an approximation set and a reference set of values in classifying neural stimuli. A class of Petri nets called rough Petri nets with guarded transitions is used to model a rough neuron. An application of rough neural computing is briefly considered in classifying the waveforms of power system faults. The contribution of this article is the presentation of a Petri net model which can be used to simulate and analyze rough neural computations.

1. Introduction

This paper considers the design of a rough neuron, which is based on rough set theory [1]-[3]. The study of rough neurons is part of a growing number of papers on neural networks based on rough sets. Rough-fuzzy multilayer perceptrons in knowledge encoding and classification were introduced in [4]. Rough-fuzzy neural networks have recently been also used in classifying the waveforms of power system faults [5]-[6]. Purely rough membership function neural networks were introduced in [7] in the context of rough sets and the recent introduction of rough membership functions [8]. There are two types of rough neurons: approximation neurons and rule-based decider neurons. An approximation neuron consists of a number of input ports governed by filters, a processing element which constructs a rough set, and one or more output ports which utilizes a rough membership function to compute the degree-of-accuracy of the approximate knowledge represented by the rough set derived by the neuron. The notion of an input port filter comes from signal processing. A filter is a device

Z.W. Raś and S. Ohsuga (Eds.): ISMIS 2000, LNAI 1932, pp. 283-291, 2000.

which transmits signals in a selected band of frequencies and rejects (or attenuates) signals in other bands [9]-[10]. Filters can be calibrated by adjusting the bandwidth of values, which can stimulate an approximation neuron. The contribution of this article is the presentation of a Petri net model of a rough neuron which can be used to simulate and analyze rough neural computations.

This paper is organized as follows. The basic concepts of rough sets, decision rules and rough membership functions underlying the design of rough neurons are presented in Section 2. The design of sample rough neurons is also presented in Section 2. A Petri net model of a rough neuron is given in Section 4.

2. Basic Concepts

A brief introduction to the basic concepts underlying the design of rough membership function neurons is given in this section.

2.1 Rough Sets

Rough set theory offers a systematic approach to set approximation [1]-[3], [8]. To begin, let S = (U, A) be an information system where U is a non-empty finite set of objects and A is a non-empty finite set of attributes where $a: U \rightarrow V_a$ for every a $\in$ A. For each B $\subseteq$ A, there is associated an equivalence relation $Ind_A(B)$ such that

$$\text{Ind}_A(B) = \{(x, x') \in U^2 \mid \forall a \in B.\ a(x) = a(x')\} \quad \textbf{(1)}$$

If (x, x') $\in$ $Ind_A(B)$, we say that objects x and x' are indiscernible from each other relative to attributes from B. The notation $[x]_B$ denotes equivalence classes of $Ind_A(B)$. For X $\subseteq$ U, the set X can be approximated only from information contained in B by constructing a B-lower and B-upper approximation denoted by $\underline{B}X$ and $\overline{B}X$ respectively, where $\underline{B}X = \{\, x \mid [x]_B \subseteq X \,\}$ and $\overline{B}X = \{\, x \mid [x]_B \cap X \neq \varnothing \,\}$. The objects of $\underline{B}X$ can be classified as members of X with certainty, while the objects of $\overline{B}X$ can only be classified as possible members of X. Let $BN_B(X) = \overline{B}X - \underline{B}X$. A set X is rough if $BN_B(X)$ is not empty.

2.2 Rough Membership Functions

A rough membership function (rmf) makes it possible to measure the degree that any specified object with given attribute values belongs to a given set X [8], [16]. 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 a rmf given in (2):

$$\mu_X^B : U \rightarrow [0,1] \text{ defined by } \mu_X^B(x) = \frac{\left|[x]_B \cap X\right|}{\left|[x]_B\right|} \tag{2}$$

2.3 Example Rough Membership Function

A sample rough member function computation is given in this section (see Fig. 1(a)).

$$\mu_F^B(u) = \frac{\left\|\overline{B}F \cap [u]_B\right\|}{\left\|[u]_B\right\|} = \frac{1}{4}$$

Fig. 1(a). Sample rmf value

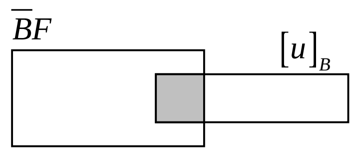

Fig. 1(b). Overlapping regions

Let B be a set of attributes of waveforms of power system faults (e.g., b1 = phase current, b2 = maximum phase current, and so on). Let F be a set of fault signal files. Further, let $\overline{B}F$ = {f3,f4,f7,f8} be an upper approximation, and let $[u]_B$ = {f4, f9, f10, f15} be an equivalence class containing files representing a known fault. For overlapping regions shown in Fig. 1(b), the degree of overlap between $\overline{B}F$ and $[u]_B$ can be computed as in Fig. 1(a).

2.4 Design of Rough Neurons

Neural networks are collections of massively parallel computation units called neurons. A neuron is a processing element in a neural network. Two types of rough neurons have been identified: approximation and decider neurons [7]. Let $\overline{B}X$ be an upper approximaton relative a set of attributes B and reference set X. An approximation neuron η computes $y = \mu_U^B\left(\overline{B}X, [u]_B\right)$. A decider rough neuron implements a collection of decision rules by (i) constructing a condition vector c_{exp} from its inputs which are rm function values (ii) discovering the rule $c_i \Rightarrow d_i$ with a condition vector c_i which most closely matches an input condition vector c_{exp}, and (iii) outputs $\min(e_i d_i)$ where $d_i \in \{0,1\}$ and $e_i = \|c_{exp} - c_i)\|/\|c_i\| \in [0,1]$. 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.

2.5 Sample Rough Neural Network

A high voltage direct current (dc) transmission system connected between ac source and ac power distribution system has two converters. In the case where the flow of power is from the ac side to the dc side as in Fig. 2, then a converter acts as a rectifier in changing ac to dc. The inverter in Fig. 2 converts dc power to ac power at desired output voltage and frequency. The Dorsey Station in the Manitoba Hydro system, for example, acts as an inverter in converting dc to ac, which is distributed throughout North America.

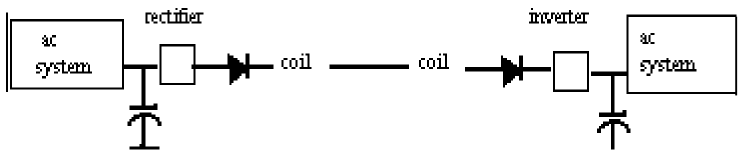

Fig. 2. dc Link Between ac Systems

A decision (d) to classify a waveform for a power transmission fault depends on an assessment of phase current (pc), current setting (cs), maximum phase current (max pc), ac voltage error (acve), pole line voltage (plvw) and phase current (pcw) waveforms. A sample commutation failure decision table is given next. In Table 1, d = 1 {0} indicates that the waveform for a fault represents {does not represent} a power system failure.

Table 1. Sample Power System Failure Decision Table

	acve	pc/cs	plvw	pcw	cs	max pc	d
file 1	0.059	0.069	0	0.0187	0	0	0
file 3	0.059	0.069	1	0.0187	0.1667	0.0856	1

Signal data needed to construct the condition granules in Table 1 come from files specified in column 1 of the table. Sample discretized rules derived from Table 1 using Rosetta [17] are given in (3) and (4).

plvw([*, 0.750)) AND cs([0.111, *)) AND max-pc([*, 0.043)) => d(no) **(3)**

plvw([0.750, *)) AND cs([0.111, *)) AND max-pc([0.043, *)) => d(yes) **(4)**

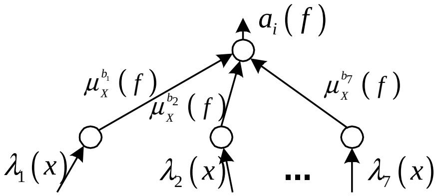

Fig. 3(a). Partial rough neural network

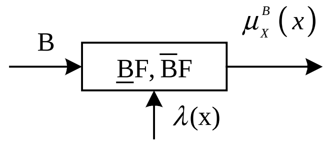

Fig. 3(b). Appoximation neuron

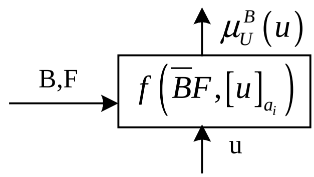

Fig. 3(c). Decider neuron

The basic structure of a rough neural network is given in Fig. 3(a)The decider neuron in Fig. 3(b) implements rules derived from Table 1. In the network in Fig. 3(a), the parameters to be tuned are represented by B, the set of relevant features. The goal of tuning is to improve the quality of concept (Fault) approximation.

2.6 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. 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.

3. Petri Net Model of a Rough Neuron

In what follows, it is assumed that the reader is familiar with classical Petri nets [19] and coloured Petri nets [20]. Rough Petri nets are derived from coloured and hierarchical Petri nets as well as from rough set theory [21]. A rough Petri net provides a basis for modeling, simulating and analyzing rough neurons, rough neural networks, and granular decision systems.

3.1 Rough Petri Nets

A rough Petri net (rPn) is a structure (Σ, P, T, A, N, C, G, E, I, W, $\Re$, ξ) where

- S is a finite set of non-empty data types called color sets.
- N is a 1-1 node function where N: A $\rightarrow$ (P x T) $\cup$ (T x P).

- C is a color function where C: $P \rightarrow \Sigma$.
- G is a guard function where G: $T \rightarrow [0, 1]$.
- E is an arc expression function where E: A $\rightarrow$ Set_of_Expressions where E(a) is an expression of type C(p(a)) and p(a) is the place component of N(a).
- I is an initialization function where I: P$\rightarrow$ Set_of_Closed_Expressions where I(p) is an expression of type C(p).
- W is a set of strengths-of-connections where ξ: A $\rightarrow$ W.
- $\Re = \{\rho_\xi \mid \rho$ constructs $\xi \in$ {rough set structure}}

Let U, S, A, d be a set of inputs, information system S, attributes of S, decision d, respectively. Examples of rough set structures constructed by ρ from information granules are the decision system S = (U, A $\cup$ {d}) and the set OPT(S) of all rules derived from reducts of a decisions system table for S. Borrowing from coloured Petri nets, a rough Petri net provides data typing (colour sets) and sets of values of a specified type for each place. The expression E(p, t) specifies the input associated with the arc from input place p to transition t, and the expression E(t, p') specifies a transformation (activity) performed by transition t on its inputs {E(p, t)} to produce an output for place p'.

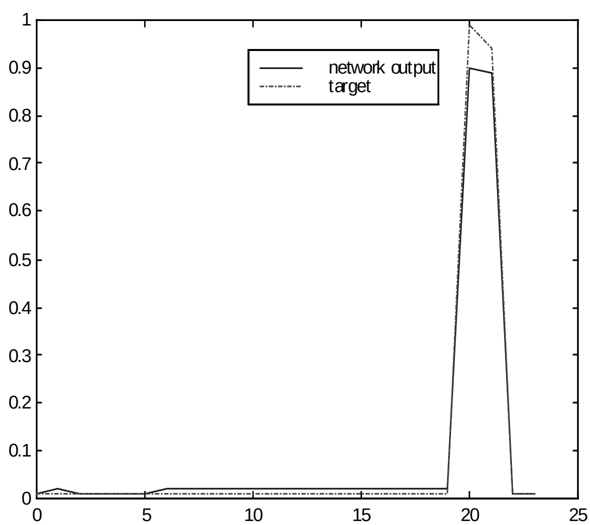

Fig. 4. Sample Verification

3.2 Guarded Transitions

In a rough Petri net, various families of guards can be defined which induce a level-of-enabling of transitions [21]. Consideration of level-of-enabling stems from guards named after Jan Lukasiewicz [22], who inaugurated the study of multivalued logic. Let U denote a universe of objects, and let $X \subseteq U$. Let $\lambda:U \rightarrow [0, 1]$.

Def. 1 *Lukasiewicz Guard.* A *Lukasiewicz guard* on transition t with input x is a higher order propositional function $P(\lambda(x))$ labeling the transition t with input x and output $\lambda(x)$. The guard $P(\lambda(x)) = \lambda(x) \in (0,1]$, where $0 < \lambda(x) \leq 1$ enables t.

With one exception, notice that $\lambda(x)$ can be used to model a filter on an input port of an approximation neuron, since there is interest in preventing input signals with zero strength from enabling an input transition. To complete the modeling of an input port filter, a restricted Lukasiewicz guard is needed.

Def. 2 *Restricted Lukasiewicz Guard.* A *restricted Lukasiewicz guard* on transition t with input x is a function $P(\lambda(x))$ labeling the transition t with input x and output $\lambda(x)$. The guard $P(\lambda(x)) = \lambda(x) \in (0,1]$, where $0 < \lambda(x) \leq 1$ enables t.

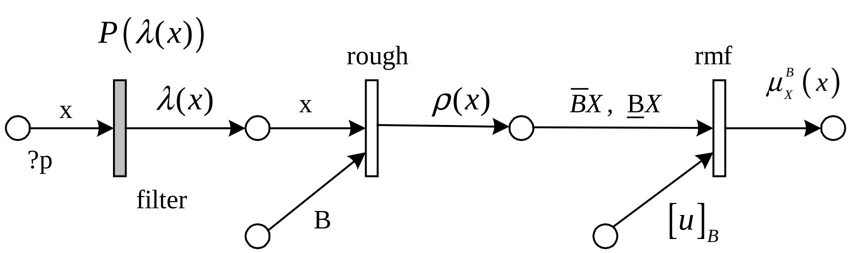

Fig. 5. Rough Neuron Petri Net Model

3.3 Petri Net Model of a Rough Neuron

Let η be an approximation neuron with a single input port p_i and single output port p_o. Let X be a set of inputs for η, B a set of attributes, and λ a filter on p_i of η. Let ρ be a procedure which constructs $\overline{B}X$, $\underline{B}X$ and let $\mu_X^B(x)$ compute the output of η (see Fig. 5). The notation ?p indicates a receptor place which is always input ready. The filter $\lambda(x)$ returns x in cases where $\lambda(x) > 0$, $\lambda(x) \in [a, b] \subseteq (0, 1]$. The transition labeled "rough" in Fig. 5 is enabled by the input of signal x and set of attributes B. When this transition fires, $\rho(x)$ constructs $\overline{B}X$, $\underline{B}X$. The availability of $\overline{B}X$, $\underline{B}X$ and equivalence class $[u]_B$ enables the transition labeled "rmf" in Fig. 5. Whenever the rmf transition fires, $\mu_X^B(x)$ computes the degree of overlap between $[u]_B$ and $\overline{B}X$. The advantage in constructing a Petri net model of a rough neuron is facilitates a number of tests such as reachability of each of the transitions in the model and the action of the guard modeling a filter on a rough neuron input port.

4. Concluding Remarks

The basic features in the design of a particular kind of rough neuron called an approximation neuron are presented in this paper. The introduction of rough neurons has been motivated by the search for improved means of identifying and classifying features in a feature space. The output of an approximation neuron is a rough membership function value, which indicates the degree of overlap between an approximation region and some other set of interest in a classification effort. A Petri net model of an approximation neuron has also been given. The guarded transitions in a rough Petri net make it possible to model a filter on an input port of a rough neuron. A sample application of these neurons in a power system fault classification system has been given. Future work will entail a study of a more complete classification and design of rough neurons.

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. M. Banerjee, S. Mitra, S.K. Pal, Rough fuzzy MLP: Knowledge encoding and classification, IEEE Trans. Neural Networks, vol. 9, 1998, 1203-1216.
5. 5 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.
6. L. Han, R. Menzies, J.F. Peters, L. Crowe, High voltage power fault-detection and analysis system: Design and implementation. Proc. CCECE99, 1253-1258.
7. J.F. Peters, A. Skowron, L. Han, S. Ramanna, Towards rough neural computing based on rough membership functions: Theory and Application, Rough Sets and Current Trends in Computing (RSCTC'2000) [submitted].
8. 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.
9. A. J. Rosa, Filters (Passive). In: R.C. Dorf (Ed.), The Engineering Handbook. Boca Raton, FL: CRC Press, Inc., 1996, 1155-1166.
10. P. Bowron, F.W. Stephenson, Active Filters for Communication and Instrumentation. London: McGraw-Hill Book Co., 1979.
11. W. Pedrycz, J.F. Peters, Learning in fuzzy Petri nets, in *Fuzziness in Petri Nets* edited by J. Cardoso and H. Scarpelli. Physica Verlag, a division of Springer Verlag, 1998.

12. 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.
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. 14 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.
15. 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.
16. 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.
17. Rosetta software system, http://www.idi.ntnu.no/~aleks/rosetta/
18. W. Pedrycz, Computational Intelligence: An Introduction, Boca Raton, CRC Press, 1998.
19. Petri, C.A., 1962, “Kommunikation mit Automaten”, Schriften des IIM Nr. 3, Institut für Instrumentelle Mathematik, Bonn, West Germany.
20. Jensen, K., 1992, “Coloured Petri Nets--Basic Concepts, Analysis Methods and Practical Use 1”, Berlin, Springer-Verlag.
21. J.F. Peters, A. Skowron, Z. Surai, S. Ramanna, Guarded transitions in rough Petri nets. In: Proc. of 7th European Congress on Intelligent Systems & Soft Computing (EUFIT’99), Sept. 1999, Aachen, Germany.
22. J. Lukasiewicz, O logice trojwartosciowej, Ruch Filozoficzny 5, 1920, 170-171. See “On three-valued logic”, English translation in L. Borkowski (Ed.), Jan Lukasiewicz: Selected Works, Amsterdam, North-Holland, 1970, pp. 87-88.