NEUROCOMPUTING

ELSEVIER Neurocomputing 36 (2001) 29–44

www.elsevier.com/locate/neucom

Fuzzy-rough and rough-fuzzy serial combinations in neurocomputing

Pawan Lingras*

Saint Mary's University, Halifax, Nova Scotia, Canada B3H 3C3

Received 12 May 1999; accepted 29 August 2000

Abstract

Rough and neo-fuzzy neurons are two different ways of introducing semantic structures in a neural network. Both have been shown to be useful in practical applications. A rough neuron consists of an upper and a lower neuron. Rough neurons can be used to effectively represent an interval or a set of values. The rough neural networks provide more flexible architectures than exclusive interval-based neural networks. Neofuzzy neurons are used to elaborate on a value by partitioning the crisp value into fuzzy segments for processing by a neural network. Previous work has shown that fuzzy values can be used to describe the difference in output of a rough neuron. This paper provides a more comprehensive introduction to serial combinations of rough and neofuzzy neurons in neurocomputing. The neofuzzy neurons are used to augment output of a rough neuron. On the other hand, rough neurons are shown to be useful for extending the expressive powers of a neofuzzy neuron. The first type of serial combination is termed a rough-fuzzy subnet. The latter serial combination is called a fuzzy-rough subnet. This paper describes the architectures of fuzzy-rough and rough-fuzzy subnets. A discussion on potential applications of the subnets is also provided along with examples. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy computing; Interval calculus; Rough computing; Neural networks

1. Introduction

Fuzzy set theory has received considerable attention since its inception [2,16]. A wide variety of tools have been developed based on the notion of fuzzy membership

* Tel.: + 902-420-5798; fax: + 902-420-5141.
E-mail address: pawan.lingras@stmarys.ca (P. Lingras).

PII: S0925-2312(00)00334-9

functions. Rough set theory [13–15] is another important component of soft computing, which has proved itself useful in the development of intelligent information systems. Researchers have also shown that it is possible to combine rough and fuzzy sets to enhance the reasoning power of an intelligent system. Recently, Lingras [7–10] proposed the concept of rough neurons, which are used for processing interval-valued variables. The term rough neuron is used to distinguish it from interval-based neural networks proposed by Ishibuchi and Tanaka [6]. Ishibuchi and Tanaka substituted the interval algebra for conventional algebra, resulting in tightly coupled interval values. Rough neurons provide loose coupling of upper and lower bounds of the intervals that can be combined with different types of neurons to form a rough-neural network. Just as the combinations of rough sets and fuzzy sets provide improved reasoning schemes, combinations of rough neurons and fuzzy membership functions can increase the modelling power of a neural network. Previous experiments have illustrated the use of fuzzy membership functions to elaborate the information contained in a rough neuron [9].

A rough neuron can be used to model interval-valued variables. Some examples of interval-valued variables are: daily high and low temperatures; range of rain fall; high and low traffic volumes; high and low values of stocks, commodities, or stock market indices. Each rough neuron stores the upper and lower bounds of the input and output values. Depending upon the nature of the application, two rough neurons can be connected to each other using either two or four connections. A rough neuron can also be connected to a conventional neuron using two connections. Many of the mathematical operations on rough neurons are borrowed from the interval calculus.

Lingras [10] used an analogy with the heap sorting algorithm and object oriented programming to stress the importance of rough neuro-computing. The argument was that the concept of a binary tree was necessary to visualize the heap sort. It would have been difficult to derive the heap sort by limiting visualization exclusively to array operations. It was similarly argued [10] that any object oriented program can also be written in a procedural programming language. However, object-oriented technology makes it easier to design, implement and analyze object-oriented programs. In the same vein, it should be noted that any computation done using rough neurons can also be rewritten in the form of conventional neurons. However, rough neurons provide a better semantic interpretation of results in terms of upper and lower bounds. Moreover, some of the numeric computations cannot be conceptualized without explicitly discussing the upper and lower bound framework [5,6,10].

Neofuzzy neurons transform crisp input to fuzzy membership functions which are then forwarded to conventional neurons for further processing. A neofuzzy neuron partitions a crisp input into fuzzy segments. Addition of such fuzzy semantics has been shown to improve the performance of a conventional neural network [12].

Previous work by Lingras [7–10] showed how intervals could be effectively used to create rough neural networks. In one of the experiments [9], rough neural networks were further modified using the notion of neofuzzy neurons. This combination of rough and neofuzzy neurons provide the motivation for the present study, which explores generic serial combinations of rough and neofuzzy neurons.

The information provided by upper and lower bounds of the output from a rough neuron can be augmented using fuzzy membership functions. This paper formalizes such a serial combination of rough neurons followed by neofuzzy neurons as rough-fuzzy subnets. The paper describes two semantic structures involving rough-fuzzy subnets. In the first semantic structure, the upper and lower bounds of the output from a rough neuron are separately partitioned into fuzzy segments. These fuzzy segment partitions will be similar to those used for conventional crisp values. The upper and lower bounds of a variable also indicate the range of fluctuation. However, such fluctuation may need to be put in a context that can be useful in decision making. In the second semantic structure based on rough-fuzzy subnets, the difference between the upper and lower bounds of the output from a rough neuron is partitioned into fuzzy segments. This partition into fuzzy segments can provide additional information about the range of fluctuation in the value of a variable.

The notion of fuzzy membership can also be extended using rough neurons. The partition of a crisp value into fuzzy segments is based on a membership function. Each fuzzy segment is assigned a single membership value. In some cases, such a partition is based on the perception of a single person. Different persons may prefer different membership functions. Rough neurons can be used to represent the range of membership values that may be associated with a fuzzy segment. This type of serial combination of neofuzzy neurons followed by rough neurons is formally described in the paper as fuzzy-rough subnets.

This paper describes the architectures of the rough-fuzzy and fuzzy-rough subnets. Examples are used to illustrate the usefulness of both types of subnets. Results of an experiment are also provided to demonstrate practical applications of the proposed subnets.

The approach proposed here is significantly different from that proposed by Ishibuchi [5], who extended the interval algebra to triangular fuzzy numbers. The resulting fuzzified neural networks are similar to Ishibuchi and Tanaka's [6] interval-based neural networks. Ishibuchi's fuzzy neural networks do not combine interval and fuzzy membership functions, instead they provide monolithic neural networks based exclusively on triangular fuzzy numbers. The approach described in this paper provides a flexible way to combine conventional, rough, and neofuzzy neurons.

Section 2 describes rough and neofuzzy neurons. Section 3 formalizes fuzzy-rough and rough-fuzzy subnets with the help of examples. An experiment using rough-fuzzy subnets is summarized in Section 4. Conclusions appear in Section 5.

2. Neural networks with rough and neofuzzy extensions

In its most general form, an artificial neural network is a collection of interconnected neurons. The nature of the connections and data exchange through the connections depend upon the application. Researchers have introduced different semantic interpretations that assist in the design of an artificial neural network [4]. This section describes two complementary types of neurons, namely, rough neurons

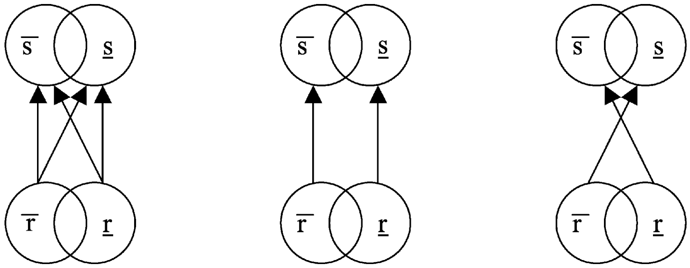


Fig. 1. Connections between rough neurons.

[7] and neo-fuzzy neurons [12]. A combination of conventional, rough and neofuzzy neurons can be used to introduce semantic structures in a neural network.

2.1. Rough neurons

An interval-value x of a variable is specified using lower $\underline{x}$ and upper $\bar{x}$ bounds, as $[\underline{x}, \bar{x}]$. A conventional precise value can be represented as an interval by specifying both the upper and lower bounds to be equal to the value of the variable. The interval algebra provides an elaborate set of operations on intervals. Some of these operations are useful in neurocomputing [5,6]. Intervals can be added as

$$x + y = [\underline{x} + \underline{y},\ \bar{x} + \bar{y}], \tag{1}$$

where x and y are intervals given by $[\underline{x}, \bar{x}]$ and $[\underline{y}, \bar{y}]$, respectively. An interval x can be multiplied by a number c as

$$c \times x = \begin{cases} [c \times \underline{x},\ c \times \bar{x}] & \text{if } c \geq 0, \\ [c \times \bar{x}, c \times \underline{x}] & \text{if } c < 0. \end{cases} \tag{2}$$

Rough neurons can be used to process intervals in a neural network.

A rough neuron r can be viewed as a pair of neurons, one for the upper bound called *upper neuron* ($\bar{r}$) and the other for the lower bound called *lower neuron* ($\underline{r}$). A rough neuron is connected to another rough neuron through two or four connections. Fig. 1 depicts three types of connections between rough neurons. The overlap between the upper and lower neurons indicates that upper and lower neurons exchange information. The upper and lower neurons are loosely coupled, in the sense that they can be separately connected to other neurons. This approach is different from Ishibuchi and Tanaka's [6] tightly coupled upper and lower bounds into a single interval value. Ishibuchi and Tanaka's neural networks are exclusively interval based. The rough neural networks, on the other hand, make it possible to use other types of neurons within the same network. The connections in rough neural networks are more flexible than exclusive interval-valued neural networks.

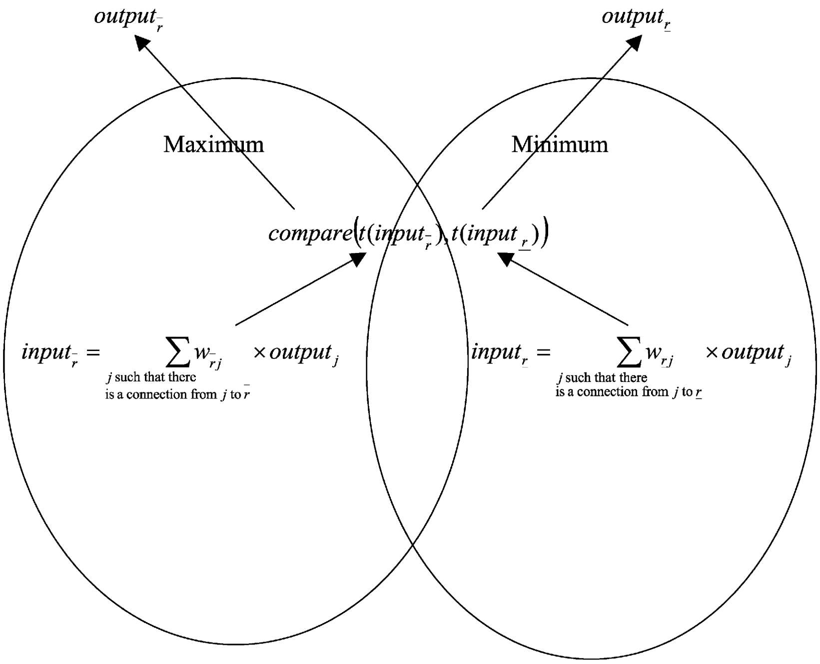


Fig. 2. Architecture of a rough neuron.

The two rough neurons in Fig. 1(a) are *fully connected*. A rough neuron r is said to be *fully connected* to another rough neuron s, if $\overline{r}$ and $\underline{r}$ are connected to both $\overline{s}$ and $\underline{s}$. If a rough neuron r is fully connected to s, then there are four connections from r to s. In Fig. 1(b) and Fig. 1(c), there are only two connections from r to s. If the rough neuron r *excites* the activity of s (i.e. an increase in the output of r will result in an increase in the output of s), then r will be connected to s as shown in Fig. 1(b). On the other hand, if r *inhibits* the activity of s (i.e. an increase in the output of r leads to a decrease in the output of s), then r will be connected to s as shown in Fig. 1(c).

The input and output of rough neurons depend upon the application. Fig. 2 summarizes the input and output of a rough neuron. The only restriction on the output of the rough neuron is that the output of the upper neuron should be consistently larger than the output of the lower neuron. The input of a conventional, lower, or upper neuron is calculated using the weighted sum as

$$input_i = \sum_{\substack{\text{j such that there is a} \\ \text{connection from j to i}}} w_{ji} \times output_j, \tag{3}$$

where i and j are either conventional neurons or upper/lower neurons of a rough neuron. The outputs of a rough neuron r are calculated using a transfer or activation function, t, as

$$output_{\overline{r}} = \max(t(input_{\overline{r}}), t(input_{\underline{r}})), \tag{4}$$

$$output_{\underline{r}} = \min(t(input_{\overline{r}}), t(input_{\underline{r}})). \tag{5}$$

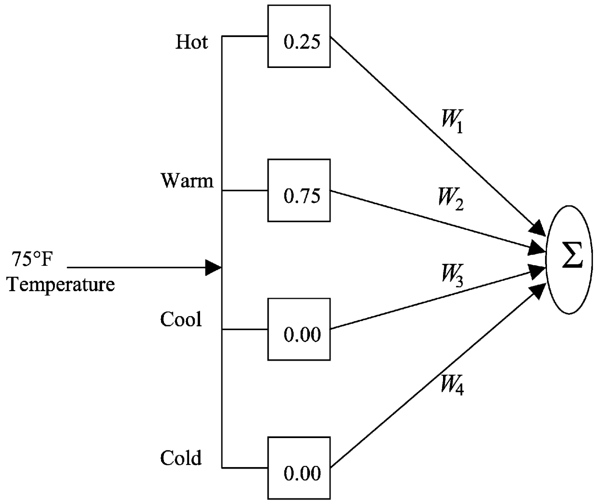


Fig. 3. A neofuzzy neuron.

Note that t can be any one of the transfer or activation functions used for conventional neurons such as the logistic function

$$t(x) = \frac{1}{1 + e^{-c \times x}}. \tag{6}$$

If two rough neurons are partially connected, then the excitatory or inhibitory nature of the connection can be determined dynamically by polling the connection weights. The network designer can make initial assumptions about the excitatory or inhibitory nature of the connections. If a partial connection from a rough neuron r to another rough neuron s is assumed to be excitatory and weights of both the connections are negative, then the connection from r to s is changed from excitatory to inhibitory. On the other hand, if r is assumed to have an inhibitory partial connection to s and the weights of both the connections are positive, then the connection from r to s is changed from inhibitory to excitatory.

2.2. Neofuzzy neurons

There have been many attempts to incorporate fuzzy membership functions in neurocomputing [1,5,12]. This study uses the neofuzzy neuron [12]. The structure of a neofuzzy neuron is depicted in Fig. 3 using an example.

A crisp temperature of 75°F is partitioned into four fuzzy segments

$(Cold, 0)$, $(Cool, 0)$, $(Warm, 0.75)$, $(Hot, 0.25)$

using membership functions given by Fig. 4. The membership functions are then used to calculate the defuzzified weighted sum for our example input of 75°F as

$$output = w_1 \times 0.25 + w_2 \times 0.75 + w_3 \times 0.0 + w_4 \times 0.0. \tag{7}$$

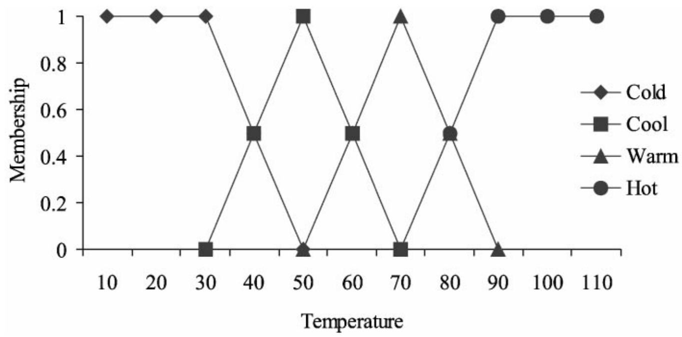


Fig. 4. Fuzzy membership functions for daily temperature.

A neofuzzy neuron can be added to a conventional neural network. The weights $w_1, \ldots, w_4$ are modified using a conventional learning rule such as the generalized delta rule. The fuzzy segments in neofuzzy neurons are based on semantic interpretations of the input and can generally be used only in the input layer. Neofuzzy neurons can be easily implemented by preprocessing the input pattern to add fuzzy segments as additional input values. The modified input pattern can then be used with a conventional neural network.

3. Serial combinations of rough and neofuzzy neurons

This section provides formal definitions of rough-fuzzy and fuzzy-rough subnets. The usage of these subnets are illustrated with examples.

3.1. Rough-fuzzy subnets

Many temporal variables are associated with a set of values. For example, daily temperature is associated with a set of temperatures recorded at regular intervals throughout the day. An interval, consisting of upper and lower bounds, can represent such a variable. Intervals convey the bounds as well as the fluctuation in the values of these variables. Such an indicator of fluctuation may be important in decision making. Lingras [9] described how fluctuation in daily traffic volumes in a week could be represented using a rough and a neofuzzy neuron for short-term prediction of traffic.

Such a combination of interval and fuzzy values can convey additional information about a variable. Let us consider a variable such as the daily temperature represented by an interval. It is possible to represent the upper and lower bounds of the temperature using fuzzy segments based on membership function given by Fig. 4. For example, a daily temperature given by [35°F, 75°F] can be represented using fuzzy segments as

$$[(Cold, 0.75), (Cool, 0.25); (Warm, 0.75), (Hot, 0.25)],$$

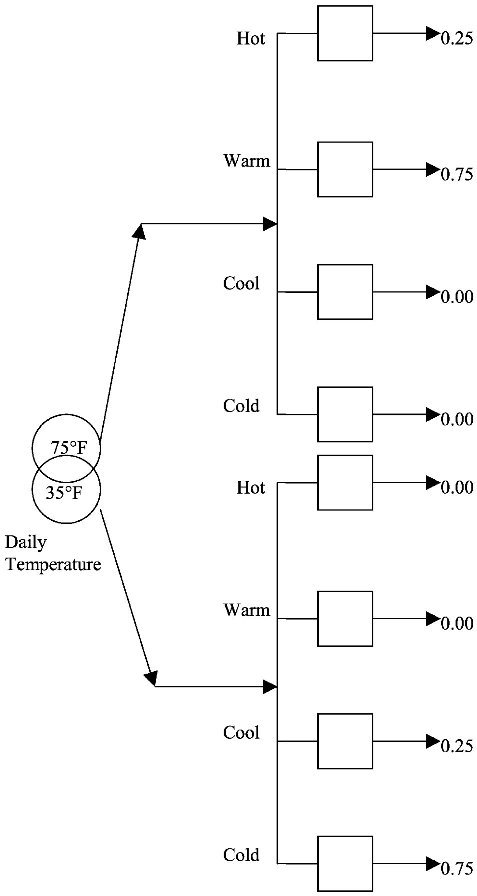


Fig. 5. Rough-fuzzy subnet to partition crisp values of upper and lower bounds.

where the semicolon is used to separate the fuzzy segments for the upper bound from the fuzzy segments for the lower bound.

It was possible to incorporate the fuzzy segments of a precise value of a temperature using a neofuzzy neuron shown in Fig. 3. However, daily temperature is represented by an interval. If we were to represent the daily temperature as fuzzy segments in neural network computations, we should combine rough neurons and neofuzzy neurons as shown in Fig. 5. The upper neuron of the daily temperature is connected to a neofuzzy neuron that partitions the high temperature into fuzzy segments. Similarly, the lower neuron of the daily temperature is connected to an identical neofuzzy neuron that partitions the low temperature into fuzzy segments.

We can also measure the fluctuation in the temperature using the fuzzy segments. Fig. 6 shows membership functions that can be used to describe the temperature fluctuations using fuzzy segments such as high, moderate or low fluctuation. The

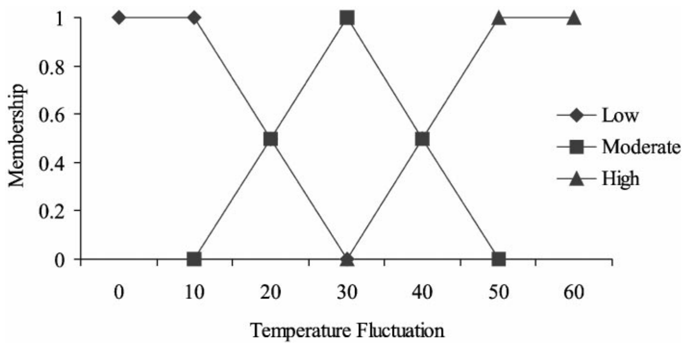


Fig. 6 . Membership functions for fluctuation in daily temperatures.

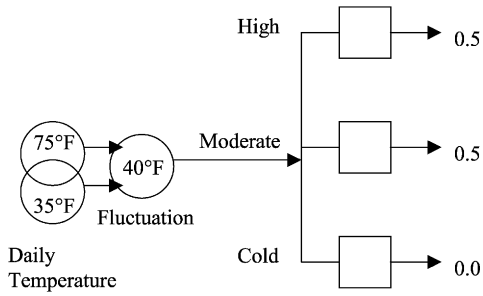


Fig. 7. Rough-fuzzy subnet to partition fluctuation between upper and lower bounds.

fluctuation in the daily temperature of $[35°F, 75°F]$ can be described as (*Moderate*, 0.5), (*High*, 0.5). In order to include such fluctuations in neurocomputing, the upper and lower neurons of daily temperature are connected to a neofuzzy neuron through a conventional neuron as shown in Fig. 7. The conventional neuron simply calculates the difference between the daily high and low temperatures and forwards the value to a neofuzzy neuron. The neofuzzy neuron partitions the crisp value of the fluctuation into fuzzy segments according to the fuzzy membership functions given by Fig. 6.

Examples used in this section demonstrate two different ways in which neofuzzy neurons can refine the output from a rough neuron for further processing. In the first case, the outputs from upper and lower neurons were individually processed by two neofuzzy neurons. Both of these neofuzzy neurons used the same fuzzy membership functions for partitioning the values of high and low temperatures. In the second case, the outputs from the upper and lower neurons were fed into a conventional neuron to calculate the crisp value of fluctuation. The crisp output from the conventional neuron was sent to a neofuzzy neuron for partitioning. In this latter case, a conventional neuron acted as an intermediary between the rough and neofuzzy neurons. The next section illustrates how rough neurons can enhance the output of neofuzzy neurons.

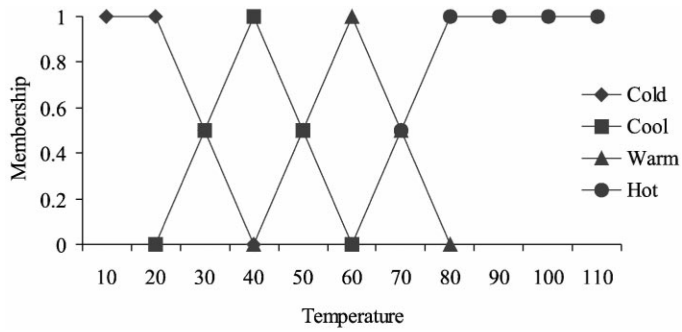


Fig. 8. Membership functions for a person extremely fond of cool weather.

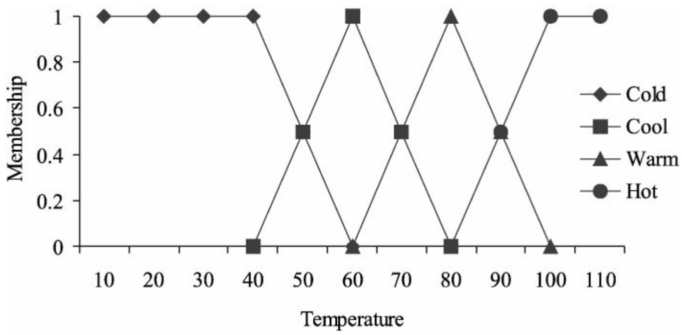


Fig. 9. Membership functions for a person extremely fond of warm weather.

3.2. Fuzzy-rough subnets

The fuzzy membership functions such as the ones shown in Fig. 4 are subjective. For the same temperature, different people will assign different membership values to the fuzzy segments Cold, Cool, Warm, and Hot. If one were to create a generic partition scheme for a range of people, it would be better to represent the membership values as intervals. Fig. 8 shows the perception of a person who is *extremely* fond of cooler weather, while Fig. 9 shows the perception of a person who is *extremely* fond of warmer weather. These persons may be hypothetical. The membership functions are created using the upper and lower bounds of the membership values supplied by different people for each temperature. It is assumed that everybody's preference is between these two extremes. The temperature of $65°F$, in our example, may be represented as

$$(Cold, [0,0]),\ (Cool, [0,0.75]),\ (Warm, [0.25,0.75]),\ (Hot, [0,0.25]).$$

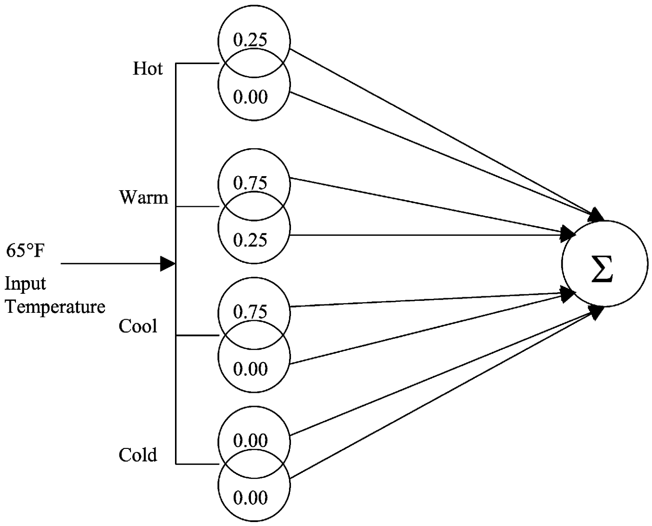


Fig. 10. Fuzzy-rough subnet.

Note that each fuzzy segment is associated with an interval of membership functions. The membership value provided by any person is expected to fall within this range.

A fuzzy-rough subnet uses rough neurons to represent each fuzzy segment as shown in Fig. 10. If the input temperature were $65^\circ F$, the hot segment will produce an output interval of [0,0.25]. Similarly, Warm, Cool, and Cold segments will output intervals [0.25,0.75], [0,0.75], and [0,0], respectively.

4. An application of rough-fuzzy subnets

The modeling of traffic volume time series is important for the prediction of traffic volumes in the immediate future. Such predictions will have applications in Intelligent Transportation Systems (ITS). The main objective of the experiment in this section is to demonstrate the semantic structures that can be introduced by rough and neofuzzy neurons.

Traffic volume data used in the study consisted of five year traffic volumes collected at a permanent traffic counter (PTC) site on an urban highway section north of Calgary, Alberta, Canada. The PTC site collects data for every hour in a given year. Modeling of hourly traffic volumes can be useful for the prediction of future demand in intelligent vehicle highway systems. However, such a modeling is computationally expensive. Recently, Lingras and Osborne [11] used daily traffic volumes for studying the characteristics of traffic flow time series. This study uses the same time series for the illustration of rough and neofuzzy computing. The objective of the experiment is to predict traffic volume on a given day. Hence, the daily traffic volume V_n for a given day represents the output or independent variable for all the models. The daily

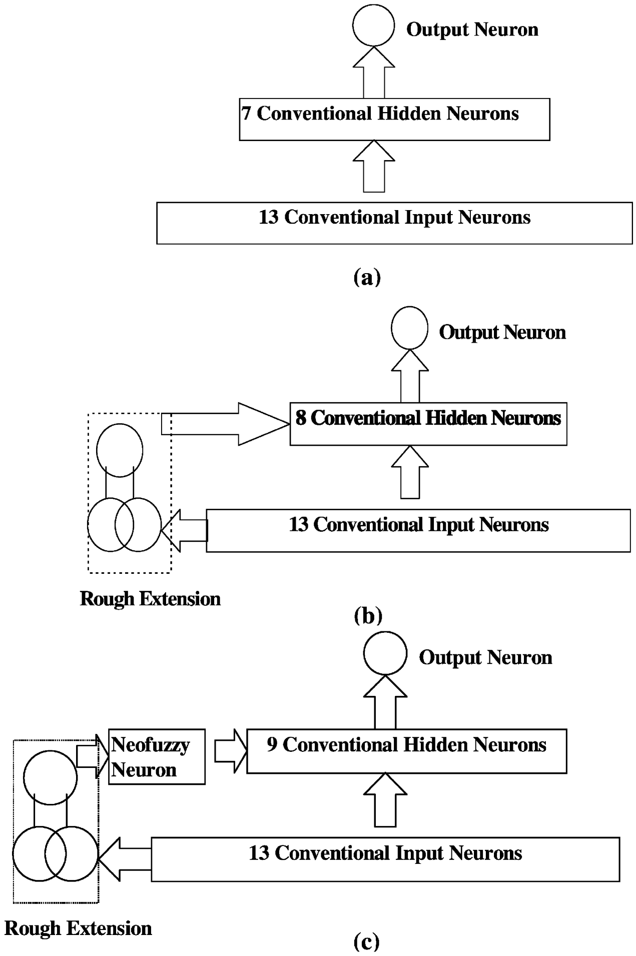


Fig. 11. Three models used for time series analysis.

volume is predicted using the historical values of daily volumes from previous 13 days. That is, the previous 13 daily volumes $V_{n-1}, V_{n-2}, \ldots, V_{n-13}$ are used as the input or independent variable. Mathematically, the models can be written as

$$V_n = f(V_{n-1}, V_{n-2}, \ldots, V_{n-13}). \tag{8}$$

The conventional neural network model, shown in Fig. 11(a), has 13 input neurons, seven hidden layer neurons and one output neuron. Neurons in the input layer are fully connected to neurons in the hidden layer. Neurons in the hidden layer are fully connected to the neuron in the output layer. The conventional model is modified by the introduction of a rough neuron. The weekly variation in traffic volumes is affected by seasons as well as special events such as holidays. Such effects should be explicitly

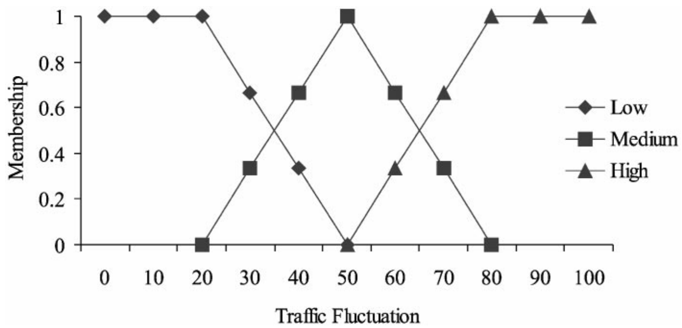


Fig. 12. Membership functions for fluctuation in traffic volumes.

taken into account during the modeling process. This study used an additional rough neuron in an intermediate layer between the input and hidden layer as shown in Fig. 11(b). The rough neuron r accepts all the inputs from the input layer. The outputs from the upper and lower neurons are calculated as the maximum and minimum values of the inputs. The upper neuron $\bar{r}$ sends out the maximum value from the inputs. The minimum value from the inputs is the output of the lower neuron $\underline{r}$. The outputs from the upper and lower neurons are sent to a conventional neuron c. The output of the conventional neuron is a function of the output from the rough neuron:

$$output_c = \frac{output_{\bar{r}} - output_{\underline{r}}}{average(output_{\bar{r}}, output_{\underline{r}})}. \tag{9}$$

The above function uses the difference between the outputs of the upper and lower neurons and normalizes it by the average of the outputs. Such a function may provide a reasonable indication of the fluctuation between the daily volumes in the previous two weeks. In addition to the output from the input layer neurons, $output_c$ is sent to the hidden layer. In the third model the crisp value $output_c$ is sent as input to the neofuzzy extension as shown in Fig. 11(c). The crisp value is partitioned into three fuzzy segments called *high fluctuation*, *medium fluctuation* and *low fluctuation* using the membership functions shown in Fig. 12. The output in the form of membership values from the fuzzy segments is sent to the hidden layer along with the output from the input layer neurons. Fig. 13 shows the reduction of errors during the training process for all three networks. The error decreases with progressive additions of rough and fuzzy components.

It should perhaps be emphasized that this experiment demonstrates how rough and neofuzzy components can be used to add semantic structures in a neural network. The enhanced model described here is not meant to be the final solution to the modeling of daily traffic volume time series. A more complete description of the experiment can be found in [9].

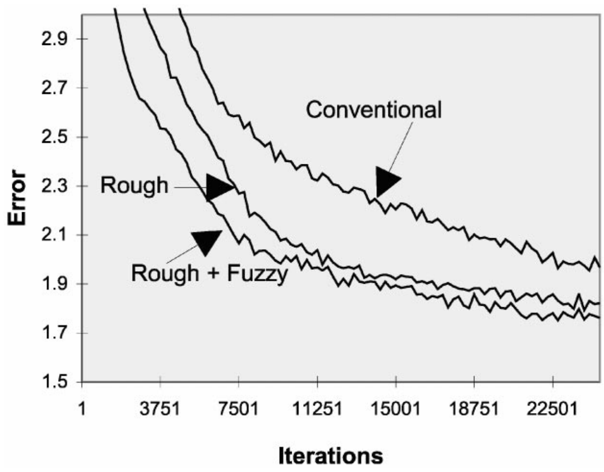


Fig. 13. Reductions in errors for conventional, rough, and rough-fuzzy models.

5. Conclusions

This paper described generic combinations of rough and neofuzzy neurons. Inputs and outputs of a rough neuron are specified using lower and upper bounds. Variables such as daily temperature are associated with a set of values instead of a single value. The upper and lower bounds of the set can represent values of such variables.

A crisp value of a variable may be based on an artificial scale, such as the Fahrenheit scale used for temperatures. Such crisp values can be partitioned into more meaningful fuzzy segments such as — Cold, Cool, Warm and Hot — using fuzzy membership functions.

This paper illustrated how fuzzy segments can be used to represent information contained in an interval using rough-fuzzy subnets. The paper described two different semantic structures that can be obtained using rough-fuzzy subnets.

Fuzzy membership functions are largely subjective. Different people may specify different membership functions for the same values. For example, the extent to which one person thinks a certain temperature is warm may be different from the perception of another person. In such cases, rough neurons are shown to be useful for fuzzy partitions, where membership is a range of values instead of a single value. This rough extension of fuzzy segments is called fuzzy-rough subnets.

The paper described architectural details of rough-fuzzy and fuzzy-rough subnets along with examples. Financial and weather time series analyses are two major areas of applications for rough and neofuzzy neurocomputing. The range and fluctuations in the values are as important as the closing prices of stocks and stock market indices. The range of values can be accommodated using rough neurons while subjective interpretation of the values and fluctuations can be incorporated using neofuzzy neurons. Similar comments can also be made about the weather data. Variables such as daily temperature cannot be represented using a single value. The normal practice is to represent daily temperatures by an interval using upper and lower bounds, i.e. daily highs and lows. A fuzzy measure of the difference between upper and lower bounds of the temperature can also play an important role in the prediction of the next day's temperature.

The summary of an experiment reported in the previous section showed how a conventional neural network model can be enhanced with the use of a rough neuron. The model was further extended using a rough-fuzzy subnet. A rough neuron was used to explicitly consider fluctuations in values of a variable in the time series analysis. A neofuzzy neuron was used to classify the fluctuations into different fuzzy segments.

The interval calculus provides a rich set of operations that can be beneficial to the further development of rough computing. The concept of upper and lower bounds features in some statistical analysis procedures, especially in financial applications. A comprehensive framework for rough and neofuzzy neurocomputing will enable systematic development of tools and techniques that can be used in a wide range of applications.

Rough neurons are used for numeric processing. Han et al. [3] showed how rough sets can be used in neurocomputing. The objects were sent as input to the system. The resulting cardinality of rough sets was forwarded to a neural network. Since the cardinalities of a rough set will form an interval corresponding to the upper and lower bounds, rough neurons can be useful in handling the input effectively. The relationship between rough neurons and rough sets will be discussed further in a future publication.

Acknowledgements

The author would like to thank the Natural Sciences and Engineering Research Council of Canada for the financial support.

References

[1] G. Bortolan, An architecture of fuzzy neural networks for linguistic processing, Fuzzy Sets and Systems 100 (1998) 197–215.

[2] A.F. 1. da Rocha, R.R. Yager, Neural nets and fuzzy logic, in: A. Kandel, G. Langholz (Eds.), Hybrid Architectures for Intelligent Systems, CRC Press, Ann Arbor, 1992, pp. 3–28.

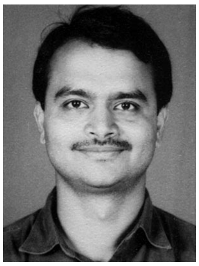
[3] L. Han, J. Peters, S. Ramanna, R. Zhai, Classifying faults in high voltage power systems: a rough-fuzzy neural computational approach, in: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Direction in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence 1711, Springer, Tokyo, 1999, pp. 47–54.

[4] R. Hecht-Nielsen, Neurocomputing, Addison-Wesley Publishing, Don Mills, Ont, 1990.

[5] H. Ishibuschi, Development of fuzzy neural networks, in: W. Pedrycz (Ed.), Fuzzy modelling Paradigms and Practice, Kluwer Academic Publishers, Dordrecht, 1996, pp. 185–202.

[6] H. Ishibuchi, H. Tanaka, An extension of the BP-algorithm to interval input vectors — learning from numerical data and expert's knowledge, in: Proceedings of IEEE International Joint Conference on Neural Networks, 1991, pp. 1588–1593.

[7] P. Lingras, Rough Neural Networks. in: Proceedings of Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 1445–1450.
[8] P. Lingras, Unsupervised Learning Using Rough Kohonen Neural Network Classifiers, in: Proceedings of Symposium on Modelling, Analysis and Simulation, CESA'96 IMACS Multiconference, Lille, France, 1996, pp. 753–757.
[9] P. Lingras, Comparison of neofuzzy and rough neural networks, Inform. Sci.: Int. J. 110 (1998) 207–215.
[10] P. Lingras, Applications of Rough Patterns, in: L. Polkowski, A. Skowron (Eds.), Rough Sets in Data Mining and Knowledge Discovery 2, Series Soft Computing, Physica Verlag, 1998, pp. 369–384.
[11] P. Lingras, P. Osborne, Linearity in daily traffic volume time series, in: Proceedings of Annual Conference of Canadian Society of Civil Engineers, Sherbrooke, Quebec, 1995, pp. 341–350.
[12] W. Nunes, F. Gomide, R. Loureiro, Recurrent learning of neofuzzy neural models, in: Proceedings of Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 1025–1030.
[13] Z. Pawlak, Rough sets, Int. J. Inform. Comput. Sci. 11 (1982) 145–172.
[14] Z. Pawlak, Rough classification, Int. J. Man-Mach. Studies 20 (1984) 469–483.
[15] Z. Pawlak, S.K.M. Wong, W. Ziarko, Rough sets: probabilistic versus deterministic approach, Int. J. Man-Mach. Studies 29 (1988) 81–95.
[16] L. Zadeh, Fuzzy sets as a basis for theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.



Pawan Lingras' academic background includes Computer Science and Civil/Transportation Engineering. His research interests include theoretical and experimental work on Rough and Fuzzy Computing, Neural Networks, Genetic Algorithms, Time Series Analysis, Data Mining, and Intelligent Transportation Systems.