Representing Complex Fuzzy Membership Functions in a Connectionist Network

by

Steve G. Romaniuk

Department of Information Systems and Computer Science

National University of Singapore

10 Kent Ridge Crescent

Singapore, 0511

Abstract

The problem of deriving membership functions as a means for describing linguistic variables (for some control process) and the choice of fuzzy inference operators and connectives is at the heart of developing fuzzy control systems. Over the years connectionist systems have obtained prominence as a means to solve complicated learning tasks. More recently a surge in interest for applying neural systems to fuzzy control problems has occurred. In this paper, it will be shown how complicated fuzzy membership functions can be composed of simpler π-shaped functions. The importance lies in the fact, that the decomposition process can be implemented through use of a connectionist representation. Furthermore, it allows augmenting an existing hybrid symbolic/connectionist learning system - SC-net - to incorporate fuzzy variables described by more complicated membership functions than was earlier possible. Finally, it provides the necessary tools for constructing connectionist fuzzy controllers, which by means of machine learning can be trained to adapt to an ever changing environment.

1 Introduction

1.1 Representing Fuzzy Sets

Due to the inherent ability of connectionist networks to display learning capabilities and their exploitation of parallelism at the neural level, they have been viewed as attractive alternatives to symbolic approaches. Applying connectionist methods to successfully solve fuzzy control problems has recently been emerging [2] [11]. One of the main problems one faces, is to effectively represent fuzzy sets into a connectionist network. Several methods have been proposed [2] [3] [1]. The difference of the method described herein is the membership functions are directly encoded into the network structure, rather than being learned by sampling some representative set of points. This guarantees a quick and accurate implementation of user defined membership functions. If required, learning techniques described in [8] can be applied to improve on the user specified membership functions based on a representative set of training examples. The proposed scheme for representing complex fuzzy membership functions guarantees - regardless of the complexity of the underlying fuzzy set -, only a fixed number of layers is required. This allows the resulting network to fully utilize the inherent parallelism of the encoded fuzzy sets.

1.2 Overview of SC-net

SC-net is a prototypical knowledge acquisition tool designed for the construction of expert systems and is based on a hybrid symbolic/ connectionist architecture. Fuzzy logic is used to deal with uncertainty. Learning is similar to instance based schemes, in that cells are recruited whenever new, yet unknown, examples are encountered during the learning process. Due to its nature, it is incremental and only requires a single pass through the training data. It allows representing symbolic constructs such as comparators and fuzzy quantifiers [8]. Rule inclusion as well as extraction (from learned data) is possible, making it a worthwhile candidate for not only knowledge acquisition but also knowledge refinement [8] [9]. SC-net has also been extended to a production system [12] and successfully applied to various real world problem solving tasks [6] [7]. Additionally, in [10] it was shown how SC-net can be used for acquiring knowledge in a domain that contains uncertainty (creditworthiness), and in [11] how the system can act as a fuzzy controller for a steam engine (with or without applying expert knowledge).

Next, we provide a brief description of the cell activation formulae used to update the various cells employed by SC-net. Every cell C_i with activation CA_i (except for input cells) computes its new cell activation according to the formula given in Figure 1. The output O_i is a non-linear function with response in [0,1]. If cell C_i and cell C_j are connected then the weight of the connecting link is given as $CW_{i,j}$, otherwise $CW_{i,j} = 0$.

For a given output cell, a response of 0 indicates no presence, 0.5 indicates unknown and 1 indicates true.

CA_i – cell activation for cell C_i.
O_i – output for cell C_i in [0,1].
$O_{i_{positive}}$ and $O_{i_{negative}}$ are the positive and negative collector cells for C_i respectively.
$CW_{i,j}$ – weight for connection between cell C_i and C_j, $CW_{i,j} \in \mathcal{R}$.
CB_i – cell bias for cell C_i, $CB_i \in \mathcal{R}$.

$$CA_i = \begin{cases} min_{j=0,..,i-1,i+1,..n}(O_j * CW_{i,j}) * |CB_i| & C_i\ is\ a\ min\ cell \\ max_{j=0,..,i-1,i+1,..n}(O_j * CW_{i,j}) * |CB_i| & C_i\ is\ a\ max\ cell \\ \left|\Sigma_{j=0;j\neq i}^{n} O_j * CW_{i,j}\right| * CB_i & C_i\ is\ a\ ;ltc \\ 1 - (O_j * CW_{i,j}) & C_i\ is\ a\ negate\ cell \\ O_{i_{positive}} + O_{i_{negative}} - 1/2 & C_i\ is\ either\ an\ intermediate \\ & or\ final\ output\ cell. \end{cases}$$

$$O_i = max(0, min(1, CA_i))$$

Figure 1: Cell activation formula

2 Representing Simple Fuzzy Variables

SC-net supports fuzzy variables, which allow either the user or the system itself to divide the numerical range of a variable into its fuzzy equivalent. In general fuzzy variables are described by a set of membership functions, where each function is associated with a linguistic hedge (variable) such as *high*, *small*, etc [13]. These membership functions correlate a given numerical value with a degree of membership indicating the strength (membership) of the numerical value being a member of the predefined fuzzy sets. A linguistic hedge in SC-net is defined by 4 quantities:

$$< LinguisticHedge >: B_L \ldots B_U(P_L, P_U) \quad (1)$$

Whenever the value of a given fuzzy variable lies between B_L and B_U $< LinguisticHedge >$ takes on a membership value of 1. If the value falls outside the P_U and P_L range a membership value of 0 is assigned. In every other case a graded membership response is returned, which in turn is described by a linear function (arms of pi-shaped membership function). Figure 2 depicts the general network structure used by SC-net to represent a fuzzy variable (labeled attribute) and a linguistic hedge (labeled attribute[value]). Cells labeled with the numerical value of -1 return the minimum of the incoming activations, whereas cells with a 0 label return

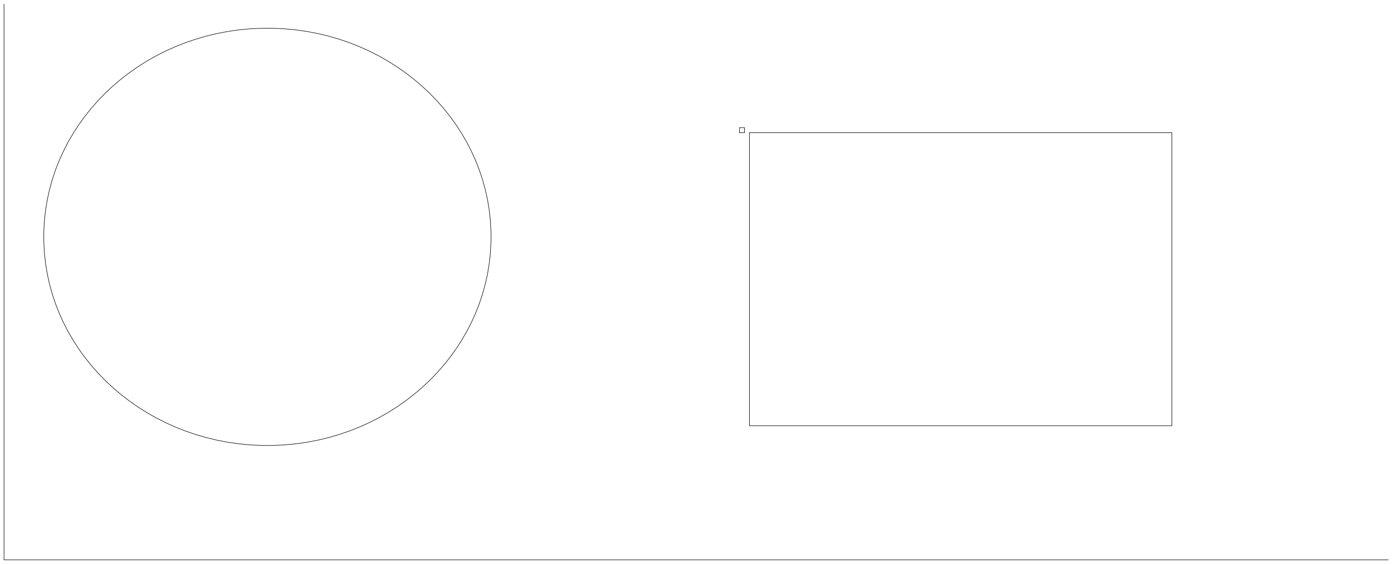

Figure 2: SC-net Network Representing Fuzzy Attribute

the strong negation $(1 - Activation)$ of the incoming activation.

Weights are calculated as follows:

$$\begin{array}{l} Weight - 1 = \frac{1}{P_U} \\ Weight - 2 = 1 \\ Weight - 3 = \frac{P_U}{P_U - B_U} \\ Weight - 4 = \frac{1}{1 - P_L} \\ Weight - 5 = \frac{1 - P_L}{B_L - P_L} \end{array} \tag{2}$$

Finally, SC-net allows the arms of fuzzy membership functions to be dynamically adapted through use of the Dynamic Plateau Modification Algorithm (DPM for short) [8]. Initially P_L and P_U are set to the smallest and the largest variable range value, respectively. By presenting encoded instances of examples learned by RCA and represented into a network structure (the SC-net network) the membership arms of the pi-shaped functions are modified. The central idea of the algorithm is to place constraints on the degree of generalization provided by each of the arms. If the degree of membership calculated by either arm is too high (for a given encoded example), it is lowered by appropriately moving either the P_L value closer to B_L or P_U closer to B_U. The amount of adjustment is determined by comparing the actual and the expected output response of a cell. For further detail on fuzzy variables, activation functions used by SC-net, and

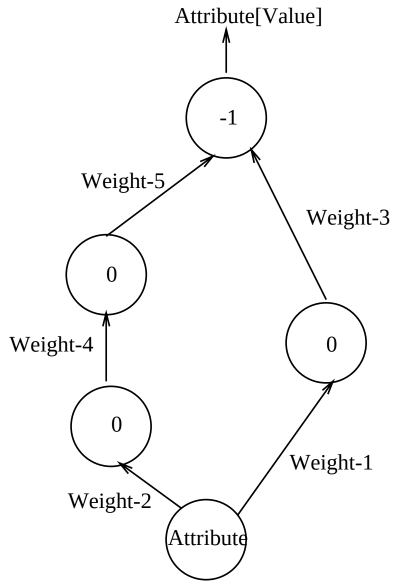

Figure 3: SC-net Network Representing Fuzzy Attribute

the Dynamic Plateau Modification algorithm refer to [8, 9].

3 Composition of Complex Membership Functions

In this section we will show how complicated fuzzy sets can be constructed from simple π-shaped membership functions as described in the previous section. A small demonstrative example will close out this section. We start of with a definition of a monotonic fuzzy set.

Definition 1 *A monotonic fuzzy set* $F \in \mathcal{F}$ *has the property that its membership values initially monotonically increase (decrease) to some maximum (minimum) point before they monotonically decrease (increase). Given* $F = \{v_1/M(v_1), v_2/M(v_2), \ldots, v_N/M(v_N)\}$*, where the* v_i $i = 1, \ldots, N$ *represent the domain values and the* $M(v_i)$ *correspond to the associated membership values of the* v_i*. We obtain*
$\forall F \in \mathcal{F}\ \exists k\ \forall i, j v_1 \leq v_i, v_j < v_k \wedge v_i \leq v_j \Rightarrow M(v_i) \leq M(v_j) \wedge \forall i, j\ v_k < v_i, v_j \leq v_N \wedge v_i \geq v_j \Rightarrow M(v_i) \geq M(v_j)$

Theorem 1 *Any monotonic fuzzy set* $F \in \mathcal{F}$ *can be decomposed into a disjunction of simple* π*-shaped membership functions.*

Proof Theorem 1 *We prove Theorem 1 by construction, which provides us with the necessary algorithm for performing the transformation from* *any monotonic fuzzy set to the desired disjunction of π-shaped membership functions. Let L_i be a π-shaped fuzzy membership function. We can define L_i as a quadruple $L_i(B_L, B_U, P_L, P_U)$. Here, the B_L and the B_U represent the lower and upper bound values of L_i, and P_L and P_U the lower and upper plateau values, respectively. Assume, indexes i, j, and l are given such that the following condition holds,*

$$\begin{array}{ll} M(v_i) \leq M(v_j) \leq M(v_l), v_i < v_j < v_l & \textit{and} \\ \frac{v_j - v_i}{M(v_j) - M(v_i)} \neq \frac{v_l - v_i}{M(v_l) - M(v_i)} & \end{array} \tag{3}$$

Whenever the above condition holds the membership curve spanned between the domain points v_i, v_j, and v_l is non-linear. Let v'_i and v'_j be symmetrical points around v_k. That is,

$$\begin{array}{ll} v_k - v_i = v'_i - v_k & \textit{and} \\ v_k - v_j = v'_j - v_k & \end{array} \tag{4}$$

2 conditions may result from this.

$$\begin{array}{l} \frac{v_j - v_i}{M(v_j) - M(v_i)} = \frac{v'_j - v'_i}{M(v'_j) - M(v'_i)} \\ \frac{v_j - v_i}{M(v_j) - M(v_i)} \neq \frac{v'_j - v'_i}{M(v'_j) - M(v'_i)} \end{array} \tag{5}$$

We will turn our attention to the first case. It clearly indicates a symmetry between the left and right arm piece spanned by the four quantities v_i, v_j, and their symmetrical counterpoints v'_i and v'_j. We can construct a π-shaped membership function L_t for this case by setting $B^t_L = v_j$ and $B^t_U = v'_j$. The corresponding lower and upper plateau values can be derived as,

$$\begin{array}{l} \frac{v_j - v_i}{M(v_j) - M(v_i)} = \frac{v_i - P^t_L}{M(v_i) - 0} \Rightarrow \\ P^t_L = v_i - \frac{(v_j - v_i) M(v_i)}{M(v_j) - M(v_i)} \\ \frac{v'_j - v'_i}{M(v'_j) - M(v'_i)} = \frac{v'_i - P^t_U}{M(v'_i) - 0} \Rightarrow \\ P^t_U = v'_i - \frac{(v'_j - v'_i) M(v'_i)}{M(v'_j) - M(v'_i)} \end{array} \tag{6}$$

P^t_L is determined such that $M(P^t_L) = 0$ (same holds for P^t_U). Since the second case is not symmetrical, we calculate P^t_L as in the first case, whereas P^t_U is determined by

$$\begin{array}{l} \frac{v_j - v_i}{M(v_j) - M(v_i)} = \frac{v'_i - P^t_U}{M(v'_i) - 0} \Rightarrow \\ P^t_U = v'_i - \frac{(v_j - v_i) M(v_i)}{M(v_j) - M(v_i)} \end{array} \tag{7}$$

Recall, the initial condition

$$\frac{v_j - v_i}{M(v_j) - M(v_i)} \neq \frac{v'_j - v'_i}{M(v'_j) - M(v'_i)} \tag{8}$$

For the case were the two slopes are equal, the domain point v_i *can be removed from the triplet of values* (v_i, v_j, v_l) *under consideration. Instead, a new value* v_s *(if available) may be selected and the previously described process repeated. The new value triplet is given as* (v_j, v_l, v_s) *with property* $M(v_j) \leq M(v_l) \leq M(v_s), v_j < v_l < v_s$. *The process of selecting a set of triplets and constructing a* π*-shaped function from it continues until all points* $v_i, 1 \leq i \leq n$ *have been associated with the appropriate membership values* $M(v_i)$. *Under worst case conditions* $(n-1)$ π*-shaped functions need to be constructed for any given monotonic fuzzy set. This number arises when every value triplet* (v_i, v_j, v_l) *contains a non-linearity.*
q.e.d.

We conclude this section by applying the results of Theorem 1 to an actual example. Assume we are given the following monotonic fuzzy set. $F = \{1/0.0, 2/0.0, 3/0.2, 4/0.6, 5/0.8, 6/1.0, 7/0.6, 8/0.2, 9/0.0, 10/0.0\}$. The corresponding graph is shown in Figure 3.

Applying the algorithm for decomposition, we obtain 4 π-shaped membership functions $(L_0 \cdots L_3)$ which model the graph displayed earlier in Figure 3. The 4 membership functions are depicted in Figure 4.

In Theorem 1 we showed how a monotonic fuzzy set can be decomposed into a conjunction of simple π-shaped functions. In Figure 5 we show how the conjunction is realized in terms of a connectionist network. The first layer of cells (labeled L_i) corresponds to the output cells for the sub-networks which implement the individual π-shaped functions. The membership weight returned by the L_i networks is scaled by appropriately multiplying the output value. The box labeled *in* represents a network structure that determines if a value (output of L_i) is within the range $[\alpha_{i-1}, \alpha_i]$ or not. The third layer corresponds to a set of min-cells (return minimum of incoming activations) which will either output the right membership value for one of the L_i's, or zero. Finally, a max-cell (returns maximum) is used to form the fuzzy disjunction of the L_i. In Figure 6 the

Figure 4: Graph for fuzzy membership set F

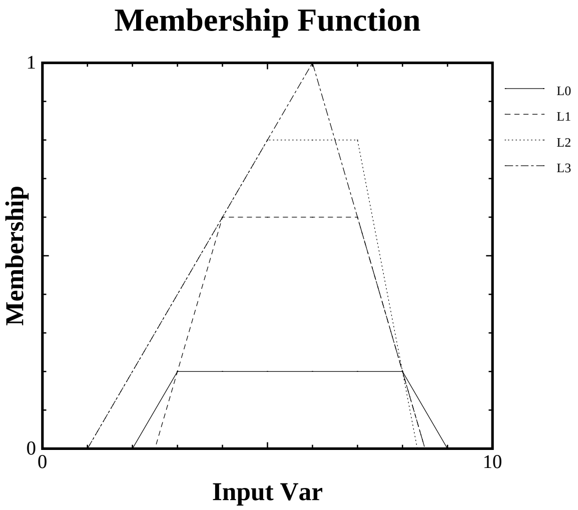

Figure 5: Decomposition of F into L_0, L_1, L_2, and L_3

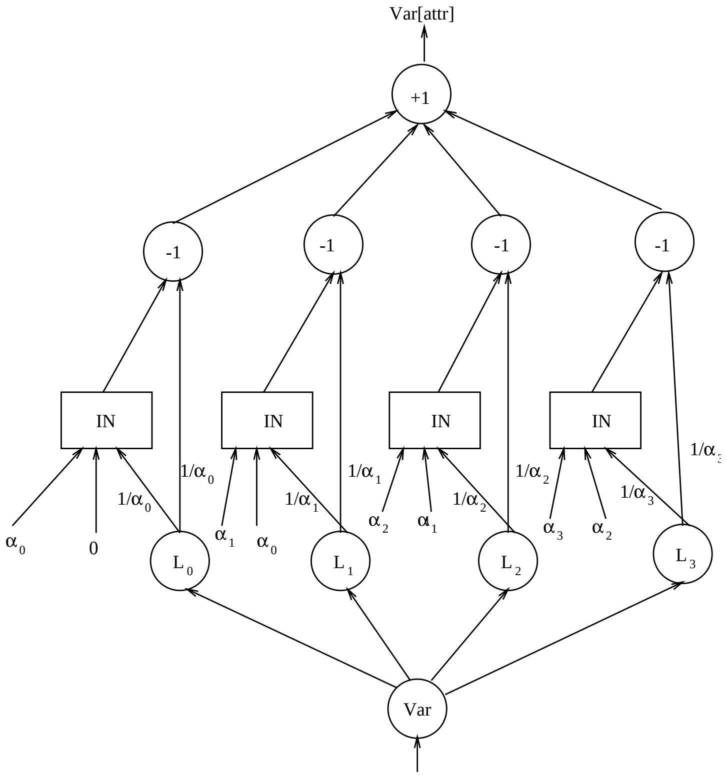

Figure 6: Complete network for modeling fuzzy set F

sub-network that realizes the *in* network is presented. The cell labeled LTC returns an activation of zero if $x \in [L, U]$. In every other case the value is greater than zero. Multiplying the activation by infinity and negating the product results in the correct decision. We mentioned earlier that the arms of π-shaped membership functions can be automatically adjusted by the DPM algorithm. The same can be accomplished using more complicated membership functions. Instead of adapting a single membership function the individual membership functions of the disjunction are modified.

4 Summary

Deriving the necessary methods for converting complex monotonic fuzzy sets into a connectionist representation by means of decomposition into simpler functions was investigated, and applied to a simple but representative example. As a result expert defined fuzzy sets and variables implementing these sets can directly be constructed and incorporated into hybrid symbolic/connectionist architectures, which guarantee exact per-

Figure 7: Representing operator *in*

formance as defined by the fuzzy sets. Fixing the number of layers, regardless of the complexity of the underlying fuzzy sets, allows the final networks to exploit any inherent degree of parallelism present in the fuzzy sets. Finally, employing machine learning techniques allows the encoded membership functions to be dynamically modified based pre-dominantly on the training set.

References

[1] Cios, K.J., Shin, I.H., Goddenday, L.S. (1991) "Generation of Fuzzy sets and Expert Rules from Scintigraphic Data to Diagnose Coronary Artery Stenosis", University of Toledo and Medical College of Ohio, Toledo, Ohio 43606.

[2] Keller, James M., Yager, Ronald R., Tahani, Hussein (1989) "Neural Network Implementation of Fuzzy Logic." Dept. CSE, University of Missouri Columbia, 1989.

[3] Keller, James M., Tahani "Backpropagation neural Networks for Fuzzy Logic." Electrical & computer Engineering Department, University of Missouri-Columbia, Columbia, MO 65211.

[4] Kibler, D. and Aha, D.W. (1990), Learning representative Exemplars of Concepts: An Initial Case Study, in Readings in Machine Learning (ed. Shavlik and Dietterich), Morgan Kaufman, Los Gatos, Ca.

[5] Kiszka, J.B., Kochanska, M.E., Sliwinska, D. S. (1985) The Influence of some Fuzzy Implication Operators on the Accuracy of a Fuzzy Model - Part II, Fuzzy Sets and Systems 15, pp. 223-240.

[6] Lee, H., Romaniuk, S.G., Hall, L.O. (1991), A Study of Machine Learning Approaches for some Classification Domains, FLAIRS-91, Coco Beach, Florida, April.

[7] Perez, R.A., Hall, L.O., Romaniuk, S.G., Lilkendey, J,T. (1992), Inductive Learning For Expert Systems In Manufacturing, HICCS-25, Hawaii International Conference on Systems Sciences, Koloa, Hawaii, January.

[8] Romaniuk, S.G. (1991) Extracting Knowledge from a Hybrid Symbolic, Connectionist Network, PhD Dissertation, University of South Florida.

[9] Romaniuk, S.G., Hall, L.O. (1992) SC-net: A Hybrid Connectionist, Symbolic Network, To appear: Journal of Information Sciences.

[10] Romaniuk, S.G. (1992) "Fuzzy Rule Extraction for Determining Creditworthiness of Credit Applicants", Submitted Ninth IEEE Conference on Artificial Intelligence for Applications.

[11] Romaniuk, S.G. (1992) "Learning Fuzzy Control with Hybrid Symbolic, Connectionist Networks." Submitted Second IEEE International Conference on Fuzzy Systems.

[12] Sanou, K., Romaniuk, S.G., Hall, L.O. (1992) "A Hybrid/Symbolic Connectionist Production System." 4th International Conference on Tools with Artificial Intelligence.

[13] Zimmermann, H. J. (1991) Fuzzy Set Theory - and Its Applications, Second, Revised Edition" Kluwer Academic Publishers