Real-Time Neuro-Fuzzy Systems for Adaptive Control of Musical Processes

Michael A. LEE and David WESSEL
Center for New Music and Audio Technologies (CNMAT)
University of California, Berkeley
Berkeley, CA 94710
e-mail: lee@cnmat.berkeley.edu & wessel@cnmat.berkeley.edu
telephone: 1 510 643 9990

Abstract

We have added a real-time interactive fuzzy reasoning system and a neural network simulator to the MAX real-time music programming language. This environment allows us to quickly prototype and experiment with Neuro-Fuzzy systems in the context of real-time musical instrument control. By fusing connectionist models with fuzzy systems, we gain the full advantages of both technologies. In this paper we describe our MAX Fuzzy System implementation and techniques for combining them with neural networks and examples that illustrate these techniques.

1 Introduction and Motivation

A well designed user interface provides novices quick access to the underlying technology of a system and at the same time allows experts to develop virtuosity. One approach to satisfying these goals is to empower the interface with adaptive capabilities. By allowing the interface to adapt according to user specified gestures, novices can quickly develop a satisfactory rapport with the system that allows them to use it.

Incorporating an adaptive element into the interface also allows the interface to change according to the growing demands placed on it as the user feels the need to be more articulate. Therefore, the interface should be capable of adding newly developed user specified gestures to its vocabulary. One such technique that has been proposed for realizing the adaptive element in self adapting interfaces is the use of connectionist models. A disadvantage of pure connectionist models is that it is difficult to embed *a priori* knowledge into the network. Fuzzy systems on the other hand, are rule based systems that can easily be structured to include prior knowledge, but can be less efficient than connectionist models when faced with multi-dimensional inputs. By fusing connectionist models with fuzzy systems, we gain the full advantages of both technologies.

In the next section, we will describe fuzzy systems and our MAX implementation of them. Following, we will present techniques for combining them with neural networks and then describe examples illustrating these techniques.

2 Fuzzy Systems

2.1 Fuzzy Systems

Computers and electronic devices are designed for manipulating precise or crisp values. Humans, on the other hand, are not high precision computing entities and usually reason using linguistic values which quantify measurements. Fuzzy systems were invented by Zadeh (Zadeh 1965) to address precisely this

problem. Fuzzy systems, based on fuzzy logic, provide a computational framework for manipulating and reasoning about imprecise expressions of knowledge. For example, an everyday rule might be expressed as:

IF fast tempo THEN short note durations

In this example, fast is a linguistic value quantifying tempo and duration quantifies note length. Although the meaning of fast tempo and short note durations is subject to some interpretation, humans would have no problem understanding the concept expressed in this rule. The question is then 'how is a computer expected to deal with this statement?' Bivalent logic systems might assign a threshold value to fast tempo and classify all tempo values to be either fast or not fast and nothing in-between. In fuzzy systems, fuzzy sets are used to describe these linguistic values and can admit varying degrees of membership. For example, 110 bpm may be a fast tempo to degree 0.6 and not fast to degree 0.3 (see Figure 1).

Figure 1: Fuzzy sets admit set membership degrees continuously on the range [0,1] while membership values of crisp sets are bivalent.

Fuzzy systems usually have five major components (Chen 1989) (see Figure 2):

1. **Fuzzy Set Database:** definition of fuzzy sets used to represent linguistic values used in the fuzzy rules.
2. **Fuzzy Rule Base:** a collection of Fuzzy IF-THEN rules.

3. Fuzzy Inference Engine: reasoning mechanism to do computations given a fuzzy rule base and input values.

4. Fuzzification Interface: to translate raw sensor values into linguistic values.

5. Defuzzification Interface: (optional) to translate fuzzy set output values into a crisp value for use by the environment.

The rule base of a fuzzy system consists of fuzzy IF-THEN rules. Fuzzy IF-THEN rules provide an easy means to express and capture human rule-of-thumb type knowledge, because they are expressed using linguistic terms. The fuzzy rules together with the fuzzy sets serve to partition the input and output spaces into fuzzy regions.

Figure 2: Fuzzy System Diagram showing major components and data types.

There are two common types of rules used in fuzzy systems that differ in their consequent parts. The first type is the conventional fuzzy rule which uses conventional fuzzy sets in the consequent part. The following are examples of conventional fuzzy rules:

IF tempo is fast and loudness is high
THEN medium note duration.

IF tempo is slow and loudness is high
THEN long note duration.

The second type of fuzzy rule is the Takagi-Sugeno-Kang or TSK rule (Takagi & Sugeno 1985). The TSK rule is used widely in control applications. In this type of rule, the consequent part is a function, usually linear, of the inputs. For example:

IF tempo is fast and loudness is high
THEN duration = $w_f \cdot \text{tempo} + w_l \cdot \text{loudness} + c$.

IF tempo is slow and loudness is high
THEN duration = $w_s \cdot \text{tempo} + w_l \cdot \text{loudness} + c$.

The advantage of the conventional fuzzy rule over the TSK is that it may be easier for humans to state rules of this form or to gain intuition from rules of this form. The TSK rules, on the other hand, have the advantage that they are easier to compute.

The fuzzy inference procedure can be described in four steps (defuzzification is sometimes omitted

depending on whether or not the application requires a single valued answer i.e. control tasks).

1. Fuzzification: input variables converted from raw sensor values into linguistic terms. The degree to which an input variable belongs to each of the linguistic terms defined on its space is computed.

2. Rule Inference: all rules in the rule base are 'fired' to obtain a rule firing strength. This value is computed by combining the membership values of each of the antecedent linguistic variables using combination and disjunction operators. The combination operator Normally the operator

3. Composition: after obtaining rule strengths for all rules in the rule-base, the consequent parts for each firing rule are computed producing either a membership value for an output fuzzy set in the case of a conventional rule or a single value in the case of a TSK rule.

4. Defuzzification: crisp output values are computed for each output variable by aggregating all qualified consequents. A more detailed account of defuzzification will be given in the next section.

Fuzzification as mentioned in the previous section is the process whereby raw sensor values are translated into linguistic values. When a crisp value is fuzzified, it is represented by a set of pairs which specify fuzzy set and the degree to which it belongs that set.

Defuzzification computes a crisp single value from the aggregation of output values for a given output variable. Several methods have been proposed, such as center of gravity, height method, or winning rule centroid. The height method weights the defuzzified centroids of all qualified consequent fuzzy sets according to the firing strength of the rule that qualified it:

$$y = \frac{\sum c_i \cdot w_i}{\sum w_i}$$

where c_i is the centroid of the consequent fuzzy set and w_i is the firing strength of the qualifying rule in the case of a conventional fuzzy rule. In the case of a TSK rule, the c_i is replaced by the output value specified by the consequent part computation.

2.2 Example Fuzzy Inference Process

Figure 3 demonstrates the fuzzy inference process. In this example, two rules with the same output variable are being fired. The MIN function is being used as the conjunction operator and centroid defuzzification is used to compute a crisp output. The input variables are tempo and loudness, and the output variable is note duration. In this example, tempo has fuzzy sets SLOW and FAST, loudness has fuzzy sets low (LO) and high (HI), and duration has fuzzy sets long (LO) and short (SH).

First, the input is fuzzified by determining to what degree the input is in the various fuzzy sets for each input dimension. Next the MIN operator is applied to the membership values of each input fuzzy sets for each rule to determine each rule's firing strength. This MIN value is then used to truncate the associated output fuzzy set. The centroid of the output fuzzy sets is then computed to give a crisp value for use by the environment.

Figure 3: Fuzzy Inference Process for the two fuzzy rules:

IF tempo is SLOW and loudness is LO
THEN duration is LONG
IF tempo is FAST and loudness is HI
THEN duration is SHORT

2.4 Fuzzy Systems in MAX

Our fuzzy system implementation in MAX consists of a collection of objects; fuzzy variables, fuzzy sets, and conventional or TSK fuzzy rules. Examples of MAX patcher syntax to represent the various components are given in Figure 4. As each fuzzy variable, fuzzy set, and fuzzy rule are added to the system, they are automatically wired into the fuzzy knowledge-base.

In this example, there are two input variables, tempo and loudness, and one output, variable duration. Each of the input and output variables have two

triangular fuzzy sets to describe their ranges. With the help of the LCD object, the fuzzy sets can be displayed as they are adjusted (see Figure 5).

Figure 5: LCD object displaying membership function shapes for loudness

3 Neuro-Fuzzy Systems in MAX

An adaptive interface is an interface whose mapping can be altered either by the user, by itself, or both. Connectionist models have been proposed as the adaptive elements. (Lee, Freed *et al.* 1991; Lee, Freed *et al.* 1992; Lee and Wessel 1992). A disadvantage of a pure connectionist model is that it is difficult to embed *a priori* knowledge into the network. Fuzzy systems on the other hand, are rule based systems that can easily be structured to include prior knowledge. Although fuzzy systems can also adapt using a variety of learning techniques (Jang 1992), they can be less efficient than connectionist models when faced with multi-dimensional inputs. Using the MAXNet object developed in (Lee, Freed *et al.* 1991) and the current fuzzy system, we can gain the full advantages of both technologies by fusing connectionist models within MAX.

Figure 6: Serial and Parallel combinations of fuzzy systems and neural networks

Among the structural methods used for fusing neural networks and fuzzy systems are to serialize or

Figure 4: MAX patcher syntax for fuzzy systems

parallelize them. By combining fuzzy and neural systems in these manners, one can take advantage of the ability to embed *a priori* knowledge and constraints into the system. This can have the effect of faster learning times and better system performance.

In fact, work in fuzzy systems and neural networks is converging. A radial basis function neural network is remarkably like a fuzzy system. Back propagation and other optimization methods can be applied to such neuro-fuzzy systems.

4 Conclusions

We have shown how to use fuzzy systems within the context the MAX real-time programming environment and how to combine fuzzy systems with neural networks. Using these hybrid architectures as adaptive elements can lead us closer to the reality of instruments that adapt to the specific and often quite personal needs of the musician [Wessel, 1991]. The use of fuzzy rule systems allows the incorporation prior knowledge but maintains the potential for adaptation.

Acknowledgments

The authors would like to thank Hideyuki Takagi and the CNMAT staff for their assistance. This research supported in part by EPRI agreement RP8010-34.

References

- Chen, Y.-Y. (1989). *The Global Analysis of Fuzzy Dynamical Systems*. University of California, Berkeley.
- Jang, J. (1992). "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation." *IEEE Transactions on Systems Man, and Cybernetics* 3(5): 714-723.
- Lee, M., Freed, A. & Wessel, D. (1992) Real-Time Neural Network Processing of Gestural and Acoustic Signals, *Proceeding of the 17th International Computer Music Conference 1991*, International Computer Music Association, 1991.
- Lee, M., Freed, A. & Wessel, D. (1992) Neural Networks for Simultaneous Classification and Parameter Estimation in Musical Instrument Control, *Proceedings SPIE Conf. on Robotics and Intelligent Systems*, May 1992.
- Lee, M., Garnett, G. & Wessel, D. An Adaptive Conductor Follower, *Proceedings of the 18th International Computer Music Conference 1992*, International Computer Music Association, 1992.
- Lee, M. A. & Wessel, D. (1992) Connectionist Models for Real-Time Control of Synthesis and Compositional Algorithms, *Proceedings of the 18th International Computer Music Conference 1992*, International Computer Music Association.
- Lee, M. A. & Takagi H. (1993) "Integrating design stages of fuzzy systems using genetic algorithms," *IEEE 2nd Int. Conf. on Fuzzy Systems*, San Francisco, CA, 1993, pp 612-617.
- Lee, M. A. & Takagi H. (1993) "Embedding *a priori* knowledge into an integrated fuzzy system design method based on genetic algorithms," *5th IFSA Congress*, Seoul, Korea, 1993.
- Takagi, T. & Sugeno, M. (1985). "Fuzzy Identification of Systems and Its Applications to Modelling and Control." *IEEE Transactions on Systems, Man, and Cybernetics* 5(3): 116-132.
- Wessel, D. (1991) Instruments That Learn, Refined Controllers, and Source Model Loudspeakers, *Computer Music Journal*, Vol 15, No. 4, Winter 1991, 82-86 MIT Press.
- Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control* 8: 338-353.