Supervised Learning of Fuzzy Logic Systems

M. Mohammadian
University of Canberra, Australia

INTRODUCTION

Conventionally modelling and simulation of complex nonlinear systems has been to construct a mathematical model and examine the system's evolution or its control. This kind of approach can fail for many of the very large non-linear and complex systems being currently studied. With the invention of new advanced high-speed computers and the application of artificial intelligence paradigms new techniques have become available. Particularly neural networks and fuzzy logic for nonlinear modelling and genetic algorithms [Goldberg, D. (1989)] and evolutionary algorithms for optimisation methods have created new opportunities to solve complex systems [Bai, Y., Zhuang H. and Wang, D. (2006)].

This paper considers issues in design of multi-layer and hierarchical fuzzy logic systems. It proposes a decomposition technique for complex systems into hierarchical and multi-layered fuzzy logic sub-systems. The learning of fuzzy rules and internal parameters in a supervised manner is performed using genetic algorithms. The decomposition of complex nonlinear systems into hierarchical and multi-layered fuzzy logic sub-systems reduces greatly the number of fuzzy rules to be defined and improves the learning speed for such systems. In this paper a method for combining subsystems to create a hierarchical and multilayer fuzzy logic system is also described. Application areas considered are - the prediction of interest rate, unemployment rate predication and electricity usage prediction.

Genetic Algorithms can be used as a tool for design and generation of fuzzy rules for a fuzzy logic system. This automatic design and generation of fuzzy rules, via genetic algorithms, can be categorised into two learning techniques namely, supervised and unsupervised. In supervised learning there are two distinct phases to the operation. In the first phase each individual is assessed based on the input signal that is propagated through the system producing output respond. The actual respond produced is then compared with a desired response, generating error signals that are then used as the fitness for the individual in the population of genetic algorithms. Supervised learning has successfully applied to solve some difficult problems. In this paper design and development of a genetic algorithm based supervised learning for fuzzy models with application to several problems is considered. A hybrid integrated architecture incorporating fuzzy logic and genetic algorithm can generate fuzzy rules that can be used in a fuzzy logic system for modelling, control and prediction.

Fuzzy logic systems typically have a knowledge base consisting of a set of rules of the form

If (x_1 is $A_1^{'}$and x_2 is $A_2^{'}$ and … and x_n is $A_n^{'}$)
Then (z_1 is $B_1^{'l}$ else z_2 is $B_2^{'l}$ else … else z_m is $B_m^{'l}$)

where $A_k^{'l}$;k = 1, …, n are normalised fuzzy sets for n input variables x_k , k = 1 ;..., n, and where $B_k^{'l}$, k ; k = 1, …, m are normalised fuzzy sets for m output variables z_k , k = 1, …,m. The heart of the fuzzy logic system is the inference engine that applies principles of intelligent human reasoning to interpret the rules to output an action from inputs. There are many types of inference engines in the literature, including the popular Mamdani inference engine, [Bai, Y., Zhuang H. and Wang, D. (2006)].

Given a fuzzy rule base with M rules and n antecedent variables, a fuzzy controller as given in Equation 1 uses a singleton fuzzifier, Mamdani product inference engine and centre average defuzzifier to determine output variables, has the general form for a single output variable, say z_1

$$z_1 = \frac{\sum_{l=1}^{M} y_k^l \left(\prod_{i=1}^{n} \mu \, A_l^i \, (x_i) \right)}{\sum_{l=1}^{M} y_k^l \left(\prod_{i=1}^{n} \mu \, A_l^i \, (x_i) \right)} \tag{1}$$

where y_k^l are centres of the output sets B_k^l and membership function μ defines for each fuzzy set A_i^l the value of x_i in the fuzzy set, namely, $\mu\, A_1^i(x_i))$. Common shapes of the membership function are typically, triangular, trapezoidal and Gaussian. A first step in the construction of a fuzzy logic system is to determine which variables are fundamentally important. It is known that the total number of rules in a system is an exponential function of the number of system variables [Raju G. V. S. and Zhou, J. (1993), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)]. In order to design a fuzzy system with the required accuracy, the number of rules increases exponentially with the number of input variables and their associated fuzzy sets to the fuzzy system. A way to avoid the explosion of fuzzy rule bases in fuzzy logic systems is to consider Hierarchical Fuzzy Logic systems [Raju G. V. S. and Zhou, J. (1993)]. Hierarchical fuzzy logic systems have the property that the number of rules needed to construct the fuzzy system increases only linearly with the number of variables in the system.

The idea of hierarchical fuzzy logic systems is to put the input variables into a collection of low-dimensional fuzzy logic systems, instead of creating a single high dimensional rule base for a fuzzy logic system. Each low-dimensional fuzzy logic system constitutes a level in the hierarchical fuzzy logic system. Assume that there are n input variables $x_1,\ldots,x_n$ then the hierarchical fuzzy logic system is constructed as follows [Raju G. V. S. and Zhou, J. (1993)]

- The first level fuzzy rule base for fuzzy system with n_1 input variables $x_1,\ldots,x_n$ which is constructed from the rules

  If x_1 is A_1^t and ... and x_n is A_{n1}^t, Then y_1 is B_1^l

  where $2 \le n_1 \le n$, and $l = 1,2,\ldots, M_1$.
- The i'th level ($i > 1$) fuzzy rule base for a fuzzy system with $n_i + 1$ ($n_i \ge 1$) input variables, which is constructed from the rules

  If $x_{N_{i+1}}$ is $A_{N_{i+1}}$ and ... and $A_{N_{i+n_i}}^l$ and y_{i-1} is Then y_i is B_i^l

  where

  $$N_i = \sum_{j=1}^{i-1} N_j \,,$$

  and $l = 1,2, \ldots , M_i$
- The construction of fuzzy rule bases for fuzzy systems continues until $i=l$ such that

  $$N_i = \sum_{j=1}^{i-1} N_j \;= n \,,$$

  that is, until all the input variables are used in one of the levels.

The first level has n_1 input variables $x_1,\ldots,x_n$ with one output variable y_1, which is then sent to the second level as input. In the second level another n_2 variables $x_{n_{i+1}},\ldots,x_{n_1+n_2}$ and the variable y_1 are combined to produce the output variable y_2, which is then sent to the third level. This procedure continues until all the variables $x_1,\ldots,x_n$ are used [Raju G. V. S. and Zhou, J. (1993), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998), Magdalena, L. (1998), Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001)]. The number of rules in a hierarchical fuzzy logic system is a linear function of the number of input variable and their associate fuzzy sets [Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)]. Other ways to reduce the fuzzy rules of a fuzzy logic system are

1. Fusing variables before input into the inference engine, thereby reducing the number of rules in the knowledge base,
2. Grouping the rules into prioritised levels to design hierarchical or multi-layered structures,
3. Reducing the size of the inference engine directly using notions of passive decomposition of fuzzy relations,
4. Decomposing the system into a finite number of reduced-order subsystems, eliminating the need for a large-sized inference engine.
5. Reducing the number of fuzzy sets of each input variable, thereby reducing the number of rules in the knowledge base of fuzzy logic system.

Using hierarchical fuzzy logic systems the typically the most influential parameters are chosen as the system variables in the first level, the next most important parameters are chosen as the system variables in the second level, and so on, [Raju G. V. S. and Zhou, J. (1993)]. In this hierarchy, the first level gives an approximate output which is then modified by the second level rule set, this procedure can be repeated