Design of Fuzzy System by NNs and Realization of Adaptability

Hideyuki TAKAGI
Computer Science Division
University of California at Berkeley
(Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.)

1. Proposal of NN-driven Fuzzy Reasoning (1988)

The issue of designing and tuning fuzzy membership functions by neural networks (NNs) was started by NN-driven Fuzzy Reasoning in 1988. NN-driven Fuzzy Reasoning involves a NN embedded in the fuzzy system which generates membership values. In conventional fuzzy system design, the membership function are hand-crafted by trial and error for each input variable. In contrast, NN-driven Fuzzy Reasoning considers several variables simultaneously and can design a multidimensional, nonlinear membership function for the entire subspace.

2. Knowledge/Skill acquisition by NN-driven Fuzzy Reasoning. (1989)

Consider the problem of balancing a pole starting with an initial swing from the hanging-down position. NN-driven Fuzzy Reasoning can process the raw data generated by a human adept at this task and can learn to infer the rules necessary for executing this task. This method has shown its ability to acquire knowledge and skill which is difficult to convey using language but is easily demonstrated.

3. Simplified design method for membership functions (1990).

Two issues affected by NN-driven Fuzzy Reasoning emerged in 1990. One was the design of structured NNs (Neural networks designed on Approximate Reasoning Architecture). The other concerned shortening the design time of membership functions so that the techniques could be used in a practical setting.

This simplified method works with one-dimensional, triangular membership functions instead of the fully general, multidimensional, nonlinear shapes, but this restriction helps speed up the design phase significantly. Currently this method is used for the design of several consumer products involving fuzzy logic (FL) and NNs by Matsushita Electric group.

4. Application of NN and FL in consumer products (1991).

Following the application of such technology in an air-conditioner in 1990, several consumer products using FL and NNs have appeared on the market in

1991. Till autumn 1991, fourteen such products had appeared on the Japanese market. In the context of consumer products, NNs have been put to use in the following five ways : (l) development tools, (2) independently of the fuzzy system, (3) as a correcting mechanism, (4) in cascade combination with FL, and (5) for learning user preferences. Equipment designed with the method mentioned in Sec. 3 falls in category (1).

5. Realization of Adaptability : Current Issues

Achieving adaptability is an important concern when fusing NNs and FL. It is too inflexible to pre-program things that depend on the user's preferences or environment. What is needed is some way to learn the usage patterns and adjust the rules using the adaptive capability of NNs. Category (5) in the previous section is intended to follow this direction.

Realization of "equipment of which handling easiness is improved as it is used more" corresponds to incremental learning in NNs. Suppose we wish to modify the equipment based on data provide by the user's actions and environment. In this case, the additional learning should have the following properties: (a) do not use all of the past training data, (b) the changes should have local effect only, in some sense, (c) training data which is more recent and supersedes older data should be recognized as such and the older information forgotten, (d) if the changes lead to violation of strict safety constraints, such data is potentially harmful and should be ignored.

6. Realization of Adaptability : Proposed Algorithm to Extract Boundary of Datasets.

Partitioning the input space is essential for determining the rulebase, such as in a fuzzy controller. Adaptive rule modification corresponds to modifying the partitioned subsets of the space. If the new data is on the boundary of the distribution of the training set, then the problem can be solved so that the four requirements in Sec. 5 above are obeyed.

This algorithm for extracting boundary data uses n-dimensional ellipses of which all axes but the major axis are equal. These shapes are used to eliminate data which lies inside the boundary, leaving the boundary points of the training dataset.

If new data is introduced on top of a boundary as shown in Fig. 1 (a), the algorithm will modify the old boundary and incorporate the new data as shown in Fig. 1 (b). This is a modification of the rule partitioning.