Tree Fuzzy Inference Systems

Introduction to Fuzzy Intelligence Computation

Prof. D.Sc. BARSEKH-ONJI Aboud

Faculty of Engineering
Universidad Anáhuac México Sur

October 30, 2025

Agenda

Principle Idea

As the number of **inputs to a fuzzy system increases**, the number of **rules increases exponentially**. This large rule base reduces the **computational efficiency** of the fuzzy system. It also makes the operation of the fuzzy system harder to understand, and it makes the **tuning** of rule and membership function parameters more dicult. Because many applications have a limited amounts of training data, a large rule base reduces the generalizability of tuned fuzzy systems.

Agenda

Hierarchical Fuzzy Systems

To overcome this issue, you can implement a fuzzy inference system (FIS) as a tree of smaller interconnected FIS objects rather than as a single monolithic FIS object. These FIS trees are also known as **hierarchical fuzzy systems** because the fuzzy systems are arranged in hierarchical tree structures. In a tree structure, the outputs of the low-level fuzzy systems are used as inputs to the high-level fuzzy systems. A FIS tree is more computationally ecient and easier to understand than a single FIS with the same number of inputs.

We can create a FIS tree using:

- A `fistree` object.
- The Fuzzy Logic Designer App.

Agenda

Types of Hierarchical Structures

There are several FIS tree structures that you can use for your application. The following gure shows commonly used FIS tree structures: an **incremental**, **aggregated**, or **cascaded** structure.
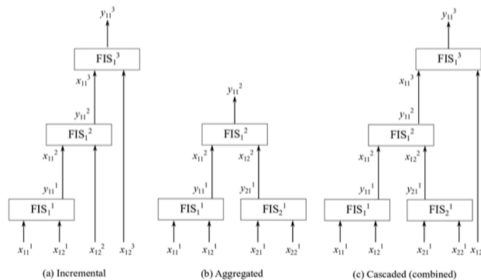

Figure 1: Types of Hierarchical FIS Tree Structures

Incremental Structure

Nature of Input Variables

In an incremental structure, input values are incorporated in multiple stages to refine the output values in several levels. The following figure shows a monolithic ($n = 1$) FIS with four inputs ($j = 1, 2, 3, 4$) and three MFs.
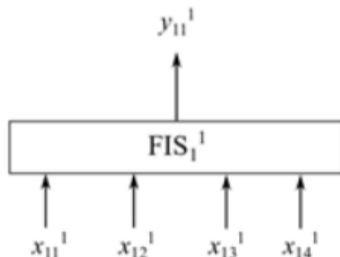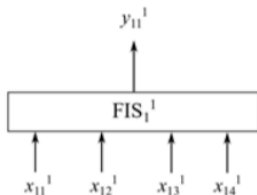

Figure 2: Single FIS example

Numbers of Rules

Figure 3: Single FIS example

The total of rules number can be calculated using:

$$N_{rules} = N_{fis}.N_{MFs}^{N_{input}} \tag{1}$$

In this case: $N_{rules} = 1 \times 3^4 = 81$, Hence, the total number of rules in an incremental FIS tree is linear with the number of input pairs.

Incremental Structure

Input Selection in a FIS Tree

Input selection at different levels in an incremental FIS tree uses **input rankings based on their contributions to the final output values**. The input values that contribute the most are generally used at the lowest level, while the least influential ones are used at the highest level. In other words, low-rank input values are dependent on high-rank input values.

Incremental Structure

In an incremental FIS tree, each input value usually contributes to the inference process to a certain extent, without being significantly correlated with the other inputs. For example, a fuzzy system forecasts the possibility of buying an automobile using four inputs: color, number of doors, horse power, and autopilot. The inputs are four distinct automobile features, which can independently influence a buyer's decision. Hence, the inputs can be ranked using the existing data to construct a FIS tree, as shown in the following figure.
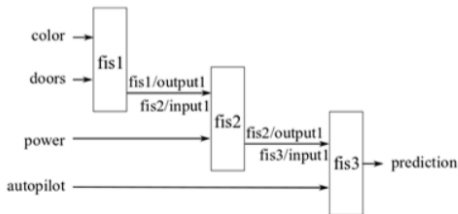

Figure 4: FIS tree (Incremental Structure)

Aggregated Structure

Principal Idea about Aggregated Structure

In an aggregated structure, input values are incorporated as groups at the lowest level, where each input group is fed into a FIS. The outputs of the lower level fuzzy systems are combined (aggregated) using the higher level fuzzy systems.
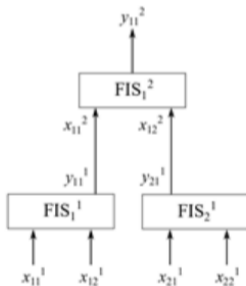

Figure 5: Aggregated FIS Tree

Aggregated Structure

Example

An autonomous robot navigation task combines obstacle avoidance and target reaching sub-tasks for collision-free navigation. To achieve the navigation task, the FIS tree can use four inputs:

- distance to the closest obstacle,
- angle of the closest obstacle,
- distance to the target,
- and angle of the target.

Distances and angles are measured with respect to the current position and heading direction of the robot. In this case, at the lowest level, the inputs naturally group as shown in the following figure: obstacle distance and obstacle angle (group 1) and target distance and target angle (group 2). Two fuzzy systems separately process individual group inputs and then another fuzzy system combines their outputs to produce a collision-free heading for the robot.

Aggregated Structure

In an aggregated FIS tree, input values are naturally grouped together for specific decision-making.
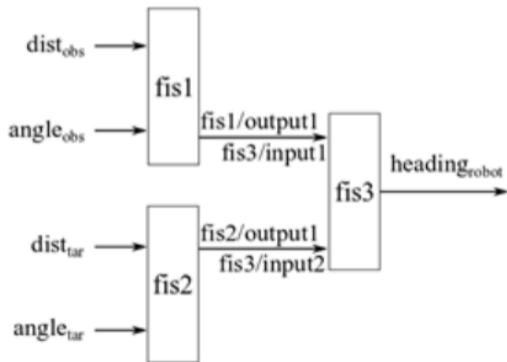
Parallel Structure

A parallel structure is a variation of the aggregated structure. In a parallel structure, the outputs of the low-level fuzzy systems are not combined using another fuzzy system. Instead, they are combined using a mathematical function, such as weighted average or weighted sum.
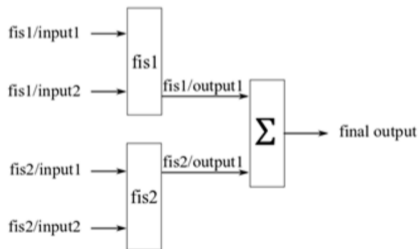

Figure 6: Parallel FIS Tree

Cascaded or Combined Structure

Principal Idea about Cascaded Structure

A cascaded structure, also known as combined structure, combines both incremental and aggregated structures to construct a FIS tree. This structure is suitable for a system that includes both correlated and uncorrelated inputs. The tree groups the correlated inputs in an aggregated structure, and adds uncorrelated inputs in an incremental structure. The following figure shows an example of a cascaded tree structure, where the first four inputs are grouped pairwise in an aggregated structure and the fifth input is added in an incremental structure.
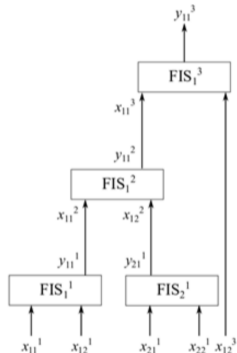

Figure 7: Cascaded FIS Tree

Cascaded or Combined Structure

Example

consider the robot navigation task discussed in Aggregated Structure. Suppose that task includes another input, the previous heading of the robot, taken into account to prevent large changes in the robot heading. You can add this input using the incremental structure of the following diagram.
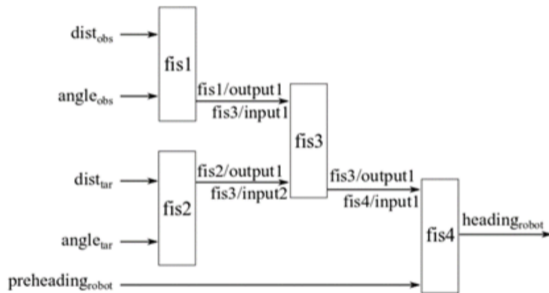

Figure 8: Cascaded FIS Tree Example

Agenda

Add or Remove FIS Tree Outputs

Outputs Operations

When you evaluate a fistree object, it returns results for only the open outputs, which are not connected to any FIS inputs in the FIS tree. You can optionally access other outputs in the tree. For instance, in the following diagram of an aggregated FIS tree, you might want to obtain the output of fis2 when you evaluate the tree.
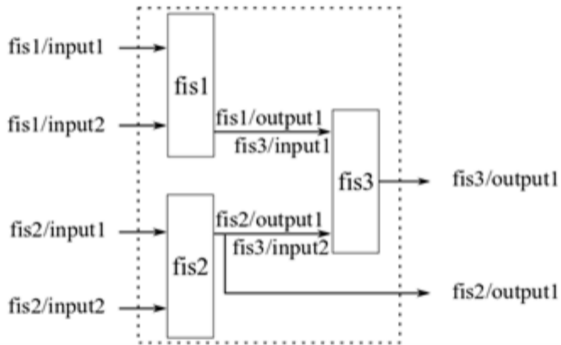

Figure 9: FIS Tree Outputs

Agenda

Use the same Value of Multiple Inputs of FIS Tree

Inputs Operations

A fistree object allows using the same value for multiple inputs. For instance, in the following figure, input2 of fis1 and input1 of fis2 use the same value during evaluation.
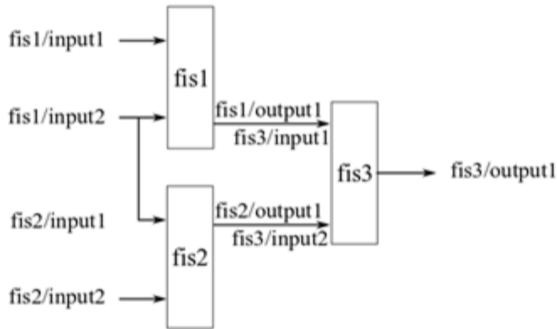

Figure 10: FIS Tree Inputs

Agenda

Conclusion

A FIS tree is a hierarchical structure of interconnected fuzzy inference systems. A FIS tree is more computationally efficient and easier to understand than a single FIS with the same number of inputs. You can create a FIS tree using a fistree object or the Fuzzy Logic Designer app. There are several FIS tree structures that you can use for your application: an incremental, aggregated, or cascaded structure.