2.3 Adaptive Neuro Fuzzy Inference System

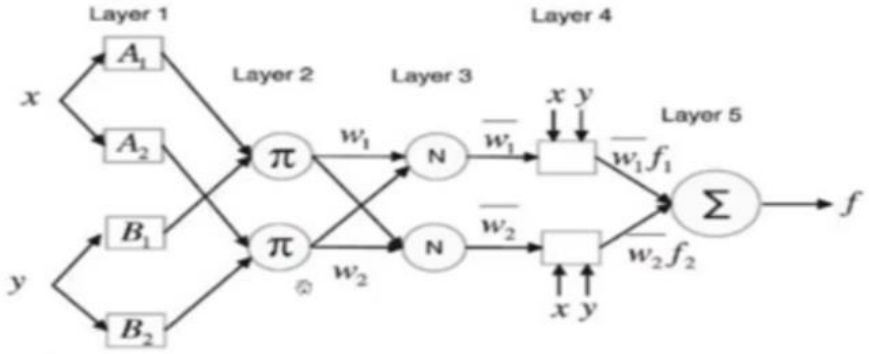
ANFIS (Adaptive Neuro Fuzzy Inference System)

- A computational model that integrates **artificial neural networks** (ANNs) and **fuzzy logic** principles thereby creating a **hybrid system** capable of modeling complex relationships between input and output data.
- Introduced by Jang in the early 1990s and has since found applications in various fields including control systems, pattern recognition, forecasting, and decision-making.
- ANFIS can handle nonlinear relationships, adapt to different types of data, and integrate of human-like reasoning through fuzzy logic.
- Careful consideration of input variable selection, fuzzy set design, and training data quality are required to design an ANFIS model.
- ANFIS provides a powerful framework for building intelligent systems capable of modeling complex systems and making accurate predictions based on imprecise or uncertain data.

ANFIS: Components and Functions

1. **Fuzzy Inference System (FIS)**: ANFIS starts with a FIS, which comprises fuzzy sets, linguistic variables, fuzzy rules, and a rule-based inference mechanism. Fuzzy logic is employed to represent and process linguistic information in a mathematical framework.
2. **Neural Network Architecture**: ANFIS utilizes a specific type of neural network known as a Takagi-Sugeno-Kang (TSK) type neural network. This neural network structure is well-suited for mapping fuzzy rules to input-output relationships.
3. **Layer Structure**:
    1. **Input Layer**: Each node in the input layer corresponds to a different input variable.
    2. **Fuzzification Layer**: This layer fuzzifies crisp input data by determining the membership grade of each input variable in the associated fuzzy sets.
    3. **Rule Layer**: Nodes in this layer represent fuzzy if-then rules. Each node computes the firing strength of a rule based on the degree of membership of input variables in corresponding fuzzy sets.
    4. **Normalization Layer**: This layer normalizes the firing strengths of rules to ensure consistency.
    5. **Consequent Layer**: This layer calculates the output using the weighted sum of the firing strengths of rules. It also acts as a defuzzification layer by computing a crisp output value.
4. **Hybrid Learning Algorithm**: ANFIS employs a hybrid learning algorithm that combines the principles of gradient descent and least-squares estimation. This algorithm adjusts the parameters of the fuzzy inference system and the neural network simultaneously to minimize the error between the actual and predicted outputs.
5. **Training**: ANFIS is typically trained using a supervised learning approach, where input-output pairs are used to update the parameters of the fuzzy sets, fuzzy rules, and neural network weights.

ANFIS Architecture



Layer 1: *Input layer.*

- Comprises neurons for transmitting external crisp signals to the next layer.
- Each neuron or node is adaptive with a membership function.
- Fuzzy membership functions can be of any shape: Gaussian, Triangular, Trapezoidal etc.

Layer 2: *Fuzzification layer.*

- A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron's fuzzy set.
- Calculates the firing strength of a rule using product $\prod$ operation.

Layer 3: *Normalization layer.*

- Calculates the normalized firing strength of a rule from previous layer

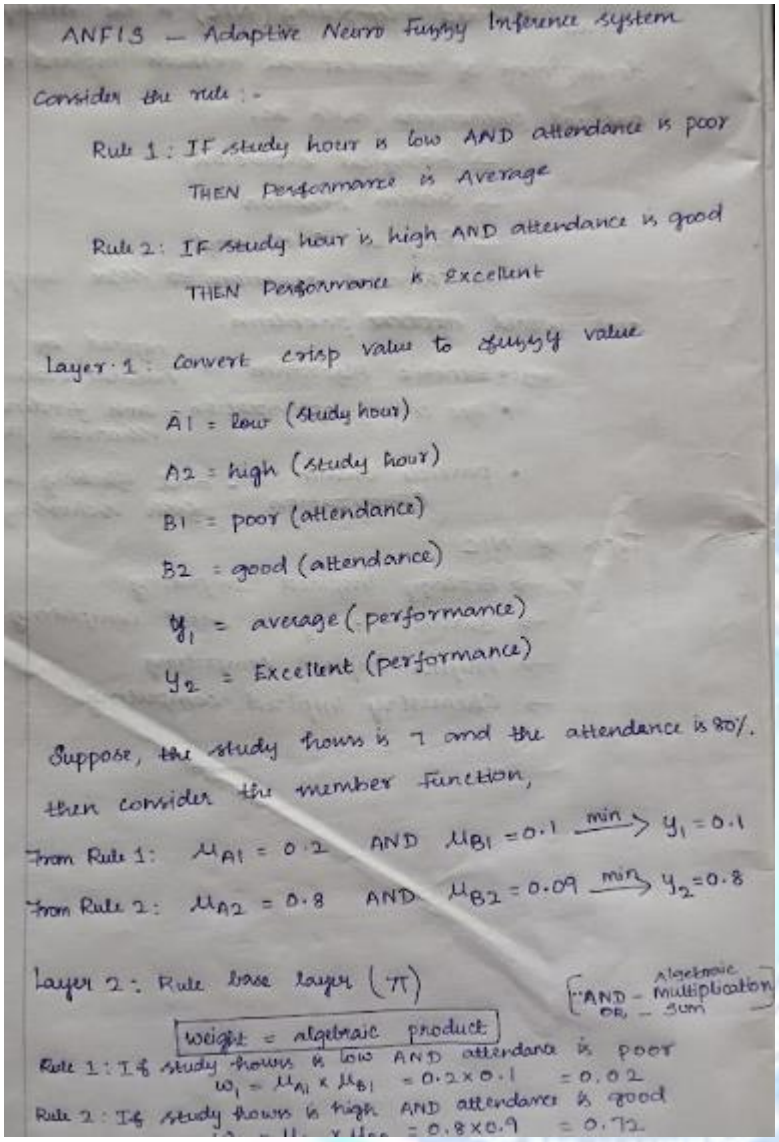
Layer 4: *Consequent layer.*

- Each node represents consequent part of fuzzy rule.
- The features x and y are associated with linear parameters/weights that are trainable.

Layer 5: *Defuzzification layer.*

- Defuzzified output obtained by summing all the outputs emanating from the consequent layer.

Example:



ANFIS – Adaptive Neuro Fuzzy Inference System

Consider the rule :-

Rule 1 : IF study hour is low AND attendance is poor
THEN Performance is Average

Rule 2 : IF study hour is high AND attendance is good
THEN Performance is Excellent

Layer 1 : Convert crisp value to fuzzy value

A1 = Low (Study hour)

A2 = high (study hour)

B1 = poor (attendance)

B2 = good (attendance)

y_1 = average (performance)

y_2 = Excellent (performance)

Suppose, the study hours is 7 and the attendance is 80%.
then consider the member function,

From Rule 1: $\mu_{A1} = 0.2$ AND $\mu_{B1} = 0.1 \xrightarrow{\min} y_1 = 0.1$

From Rule 2: $\mu_{A2} = 0.8$ AND $\mu_{B2} = 0.09 \xrightarrow{\min} y_2 = 0.8$

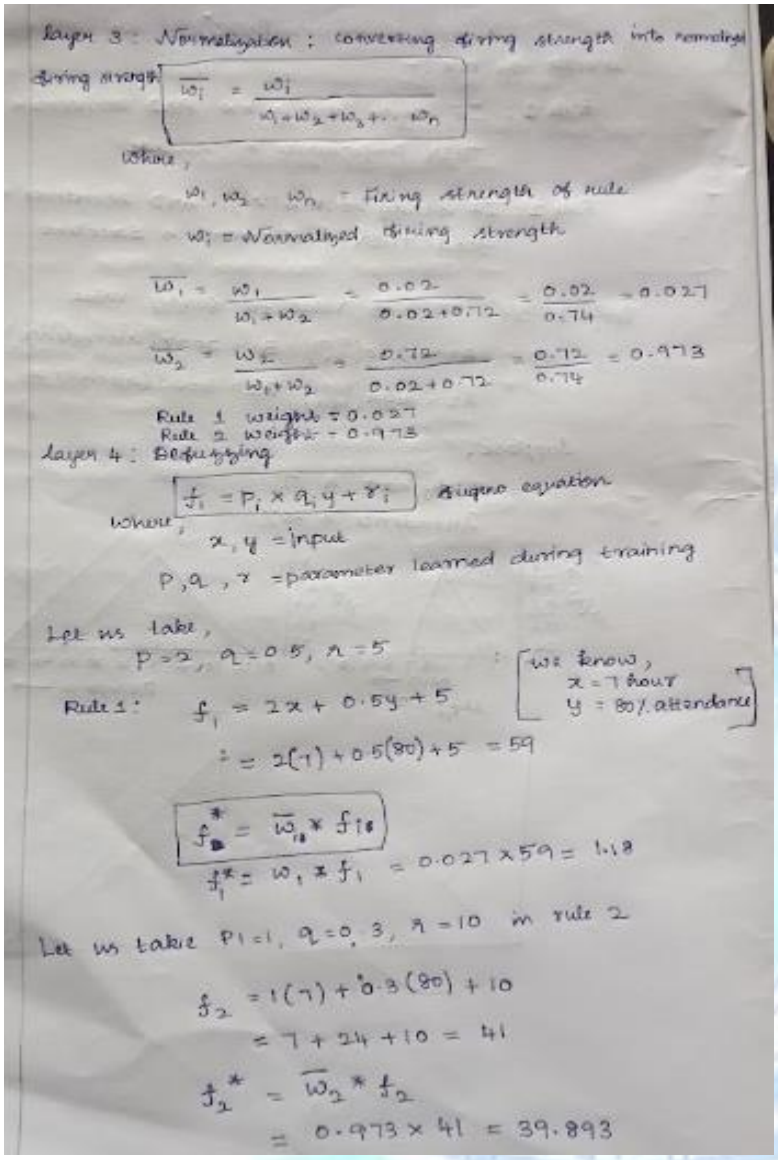
Layer 2 : Rule base layer (π)

∵ AND – Algebraic Multiplication
OR – Sum

weight = algebraic product

Rule 1 : If study hours is low AND attendance is poor
$w_1 = \mu_{A1} \times \mu_{B1} = 0.2 \times 0.1 = 0.02$

Rule 2 : If study hours is high AND attendance is good
$= 0.8 \times 0.9 = 0.72$



Layer 3 : Normalization : converting firing strength into normalized firing strength

$$\overline{w_i} = \frac{w_i}{w_1 + w_2 + w_3 + \dots w_n}$$

where,

$w_1, w_2 \dots w_n$ = Firing strength of rule

$\overline{w_i}$ = Normalized firing strength

$$\overline{w_1} = \frac{w_1}{w_1 + w_2} = \frac{0.02}{0.02 + 0.72} = \frac{0.02}{0.74} = 0.027$$

$$\overline{w_2} = \frac{w_2}{w_1 + w_2} = \frac{0.72}{0.02 + 0.72} = \frac{0.72}{0.74} = 0.973$$

Rule 1 weight = 0.027
Rule 2 weight = 0.973

Layer 4 : Defuzzying

$f_i = P_i \times a_i + q_i y + r_i$ Sugeno equation

where,

x, y = input

P, q, r = parameter learned during training

Let us take,

$p = 2, \quad q = 0.5, \quad r = 5$

We know,
x = 7 hour
y = 80% attendance

Rule 1 : $f_1 = 2x + 0.5y + 5$

$= 2(7) + 0.5(80) + 5 = 59$

$f_1^* = \overline{w_1} * f_1$

$f_1^* = w_1 * f_1 = 0.027 \times 59 = 1.18$

Let us take $P_1 = 1, \quad q = 0.3, \quad r = 10$ in rule 2

$f_2 = 1(7) + 0.3(80) + 10$

$= 7 + 24 + 10 = 41$

$f_2^* = \overline{w_2} * f_2$

$= 0.973 \times 41 = 39.893$

Layer 5 : Combining / Consequent layer $\sum \overline{w_i} f_i$

$$f = f_1 + f_2$$

$$= 1.18 + 39.893 = 41.073$$

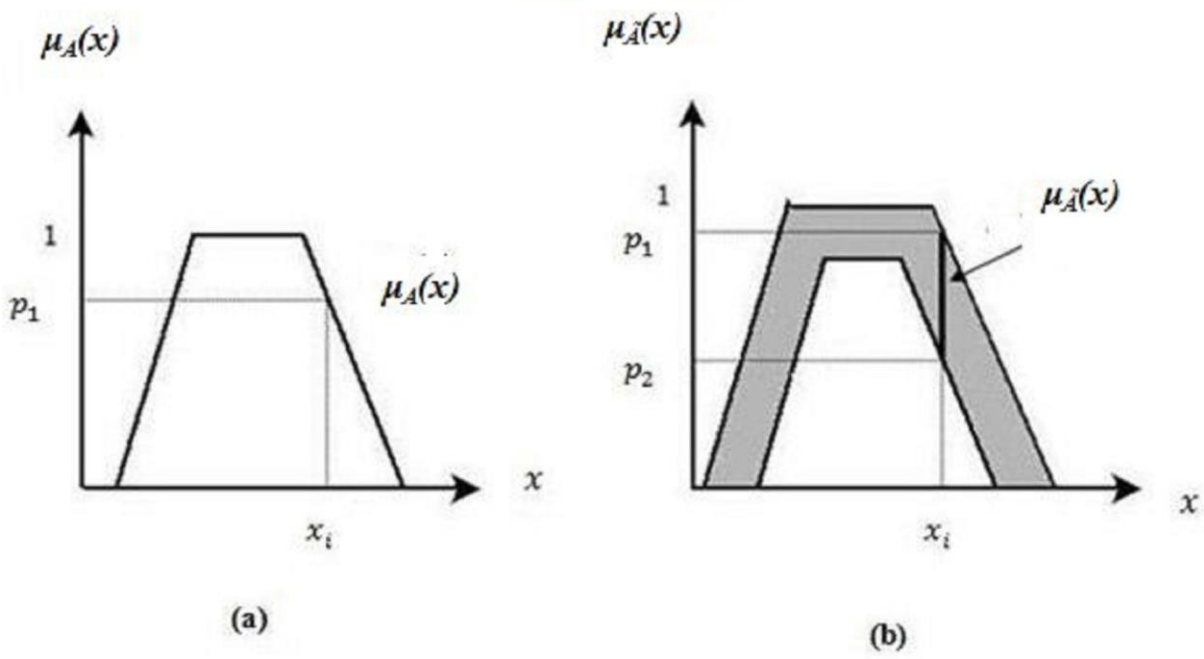
ANFIS Output is a weighted combination of the rule output Therefore, the predicted output of the ANFIS model is 41.073. It mean the student performance is 41.073 (approximately 41/100)

Type-2 Fuzzy Logic

Type-2 Fuzzy Logic (T2FL) is an extension of **Type-1 Fuzzy Logic** in which the **membership value itself is fuzzy**.

In Type-1 fuzzy logic, each input has a precise membership value between **0 and 1**.

In Type-2 fuzzy logic, the membership value is represented by a **range of possible values**, allowing the system to handle **uncertainty and imprecise information**.



Real-World Example – Air Conditioner

Consider an air conditioner that determines whether a room is **"Hot."**

Different people may have different perceptions of what *hot* means. A Type-1 system might assign:

30°C → Hot = 0.8

But because the perception of *hot* is uncertain, a Type-2 system could represent it as:

30°C → Hot = [0.7, 0.9]

The system can therefore make decisions even when the membership information is uncertain.