UCS2735 BIOINFORMATICS TECHNOLOGIES

Unit 3

Machine Learning in Bioinformatics

Machine Learning in Bioinformatics - Artificial Neural Network - Neural network architecture and applications - Genetic algorithm - Fuzzy system

* Machine Learning in Bioinformatics

→ ML models can be easily adapted to a changing environment. In bioinformatics:

(i) New data is continuously generated in molecular biology research

(ii) Biological data often has missing and noisy data. - ML techniques can be used to handle this.

(iii) ML models can handle huge volumes of biological data, and can extract hidden relationships & correlations in the data.

* Research Areas in Bioinformatics that used ML Tools

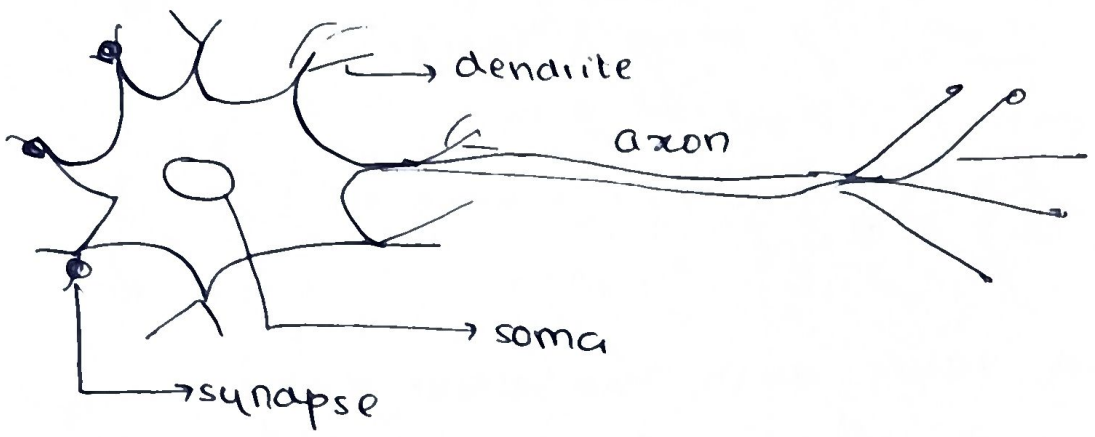
Research Area	Tool
(i) Sequence Alignment	BLAST FASTA
(ii) Multiple Sequence Alignment	Clustal W MultAlin DiAlign

Research Area	Tool
(iii) Gene Finding	GenScan
	Genome Scan
	Gene Mark
(iv) Protein domain analysis and identification	Pfam
	Blocks
	Pro Dom
(v) Pattern Identification	Gibbs Sampler
	Align ACE
	MEME
(vi) Protein Folding Prediction	Predict Protein
	Swiss Modle

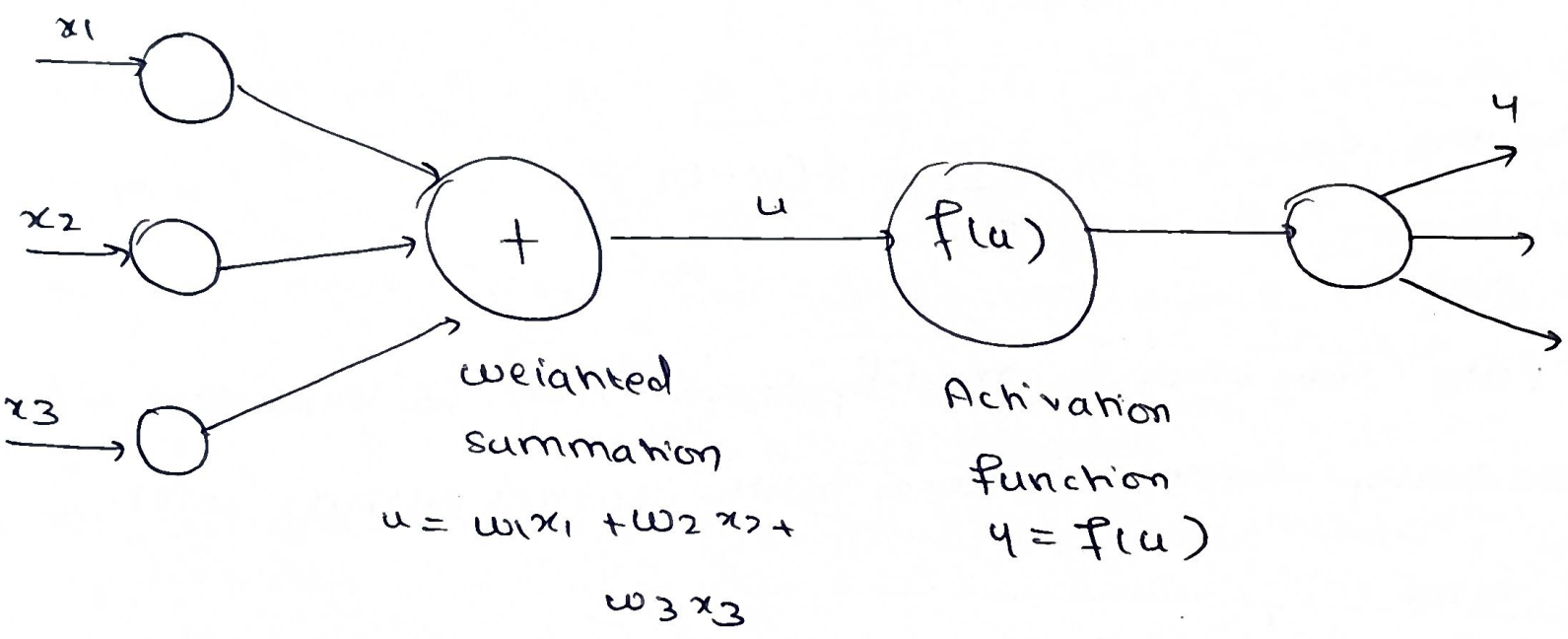
Unit -3

* Artificial Neural Networks

→ analogous to a neuron in the brain



→ ANNs can be represented as:



Function of each component	Biological Networks	ANN
Accept Inputs	Dendrite	Input
Process the inputs	Soma	Neuron
Turn the processed inputs into outputs	Axon	Output
Involve learning process	Synapse	weight

Working of ANNs

1. Choose the initial weights randomly for sample input values
2. Compute the weighted sum

$$u = \Sigma w_i x_i$$

3. Apply an activation function

$$f(u) = f(\Sigma w_i x_i) = h_i$$

4. Compare the actual output value with the target output value

t_i = target output

O_i = actual output

5. Compute the error $E = \frac{1}{2} \sqrt{\Sigma (t_i - O_i)^2}$

6. Backpropate the error to modify the weights so that the actual output gradually becomes closer to the target output, with smaller error.

* Supervised Learning in Bioinformatics

→ SVMs have been shown to perform well in multiple areas of biological analysis. These include:

(i) detecting remote protein homologies

(ii) analyze expression data

↳ usually a high dimensional data that poses a problem for many ML methods, but SVM is able to generalize well. (For other models, can do dimensionality reduction, but may lead to info loss / perf degradation)

* Unsupervised Learning Methods

→ can be categorized as:

(i) self-organizing maps (SOM)

(ii) self-organizing tree (SOTA) and

(iii) adaptive resonance theory (ART)

→ SOMs (Kononent networks) are the most commonly used algorithm

Working of SOMs

→ works on the basis of grouping data by patterns.

→ Initialize a 2D grid of neurons initialized with random weights

→ For each input vector, find the neuron closest to the data point using a distance measure.

→ This neuron is called the Best Matching Unit (BMU)

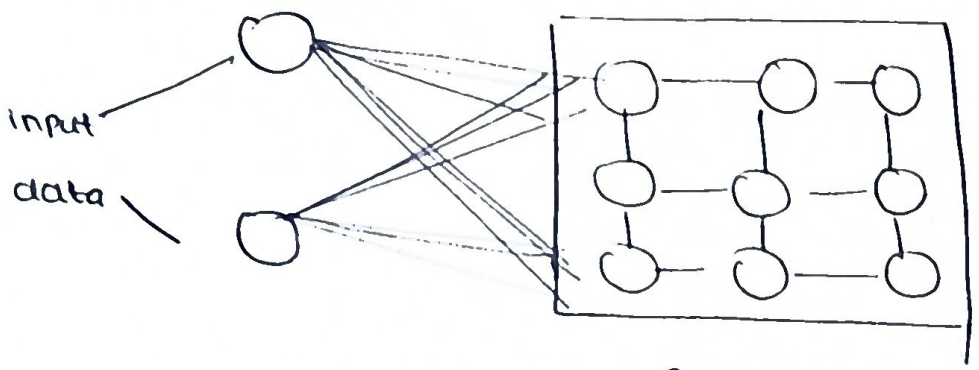
→ Update BMU and neighbor's weights to be closer to the data point

→ Neighborhood influence decreases with distance from BMU.

→ learning rate and neighborhood radius decrease with iterations

Output : Organized map with similar data points close together

Uses : Clustering, visualization & finding patterns in high dimensional data.



SOM

* Advantages, Disadvantages and Enhancements of SOM

Advantages

1. Dimensionality Reduction - valuable for mapping complex, high-dimensional data into simpler, lower-dimensional spaces.
2. Fast Execution - SOMs can process large datasets efficiently.

Disadvantages

1. Fixed Clusters - SOMs require a set number of clusters from the start, which may not align with the actual data structure.
2. Lack of Hierarchy - without a tree structure, SOMs cannot capture complex, higher-order relationships between clusters.
3. Convergence Issues: SOMs don't always guarantee convergence to a stable solution
4. Non-Deterministic Results: The results of SOMs can vary depending on the chosen learning rates.

Enhancements to overcome SOM limitations

1. Combine with hierarchical clustering : helps create a nested structure, helping analyze complex data like gene expressions

2. Use Fuzzy Kohonen Networks: combine SOM w/ fuzzy c-means, to address the fixed cluster and non-deterministic outcome issues.

* Benefits of using SOTAs

1. Proportional Clustering - clustering reflects the data's heterogeneity,

2. Binary / Nested Structure - binary topology enables a layered structure, where nodes at each level are averages of the items below them

* Neural Network Applications in Bioinformatics

→ prediction translation initiation sites in DNA sequences

→ predict imunologically interesting peptides

→ carry out pattern classification & signal processing

→ perform protein sequence classification

→ analyze the gene expression patterns

→ prediction of secondary protein structure

Example on SOM

Assume we have gene expression data for a set of genes suspected to be linked to a brain disorder.

Each gene has 3 features:

(i) Expression level (EL)

(ii) Connectivity (CN)

(iii) Mutation Rate (MR)

The data table is as follows:

Gene	EL	CN	MR
G1	0.6	0.4	0.8
G2	0.1	0.9	0.3
G3	0.5	0.6	0.2
G4	0.8	0.3	0.7

Group these genes using SOM with a 2x2 grid. The initial weight vectors are:

Neuron	Weight vector (EL, CN, MR)
(0,0)	[0.5, 0.3, 0.7]
(0,1)	[0.8, 0.2, 0.6]
(1,0)	[0.4, 0.5, 0.3]
(1,1)	[0.3, 0.7, 0.4]

Let the learning rate be 0.5

Apply SOM to group genes for one iteration.

Ans Step 1 : Normalize the data

min max normalization = $\frac{\text{value} - \text{min}}{\text{max} - \text{min}}$

$$(G_1, EL) = \frac{0.6 - 0.1}{0.8 - 0.1} = \frac{0.5}{0.7} = 0.71$$

$$(G_1, CN) = \frac{0.4 - 0.3}{0.9 - 0.3} = \frac{0.1}{0.6} = 0.17$$

$$(G_1, MR) = \frac{0.8 - 0.2}{0.8 - 0.2} = \frac{0.6}{0.6} = 1$$

$$(G_2, EL) = \frac{0.1 - 0.1}{0.8 - 0.1} = 0$$

$$(G_2, CN) = \frac{0.9 - 0.3}{0.9 - 0.3} = \frac{0.5}{0.5} = 1$$

$$(G_2, MR) = \frac{0.3 - 0.2}{0.8 - 0.2} = \frac{0.1}{0.6} = 0.17$$

$$(G_3, EL) = \frac{0.5 - 0.1}{0.8 - 0.1} = \frac{0.4}{0.7} = 0.57$$

$$(G_3, CN) = \frac{0.6 - 0.3}{0.9 - 0.3} = \frac{0.3}{0.6} = 0.5$$

$$(G_3, MR) = \frac{0.2 - 0.2}{0.8 - 0.2} = 0$$

$$(G_4, EL) = \frac{0.8 - 0.1}{0.8 - 0.1} = 1.0$$

$$(G_4, CN) = \frac{0.3 - 0.3}{0.9 - 0.3} = 0.0$$

$$(G_4, NR) = \frac{0.83 - 0.2}{0.8 - 0.2} = 0.833$$

The normalized data is:

	EL	CN	NR
G_1	0.71	0.17	1.0
G_2	0.0	1.0	0.17
G_3	0.57	0.5	0.0
G_4	1.0	0.0	0.83

Step 2 : Identify the BMU for each gene

G1

To (0,0)

$$\sqrt{(0.71-0.5)^2 + (0.17-0.3)^2 + (1.0-0.7)^2} = 0.39$$

To (0,1)

$$\sqrt{(0.71-0.8)^2 + (0.17-0.2)^2 + (1.0-0.6)^2} = 0.41$$

To (1,0)

$$= 0.61$$

G1 - closest to (0,0)

To (1,1)

$$= 0.78$$

G2

To(0,0) = 0.93

To(0,1) = 1.10

To(1,0) = 0.75

To(1,1) = 0.54

G2 closest to (1,1)

G3

To(0,0) = 0.74

To(0,1) = 0.72

To(1,0) = 0.57

To(1,1) = 0.53

G3 closest to (1,1)

G4

To(0,0) = 0.54

To(0,1) = 0.30

To(1,0) = 0.86

To(1,1) = 1.03

G4 closest to (0,1)

Step3 : Update the weights of the BMU

For BMU(0,0) G1

new wt = old wt + α (gene vector - old wt)

EL = 0.5 + 0.5 (0.71 - 0.5) = 0.605

CN = 0.3 + 0.5 (0.17 - 0.3) = 0.235

MR = 0.7 + 0.5 (1.0 - 0.7) = 0.85

Updated vector for (0,0) : ~~[0.5, 0.3~~

[0.605, 0.235, 0.85]

For BMU (1,1) (G2 and G3)

EL: $0.3 + 0.5\ (0.0 - 0.3) = 0.15$

CN: $0.7 + 0.5\ (1.0 - 0.7) = 0.85$

MR: $0.4 + 0.5\ (0.17 - 0.4) = 0.285$

Updated weight vector = [0.15, 0.85, 0.285]

For BMU (0,1) G4

EL: $0.8 + 0.5\ (1.0 - 0.8) = 0.9$

CN: $0.2 + 0.5\ (0.0 - 0.2) = 0.1$

MR: $0.6 + 0.5\ (0.83 - 0.6) = 0.715$

Updated weight vector = [0.9, 0.1, 0.715]

The clusters are

Neuron (0,0) → G1

Neuron (0,1) → G4

Neuron (1,1) → G2, G3

* Genetic Algorithm (read theory and steps from Pattern Recognition notes)

Advantages

→ can do parallel search

→ generates robust & optimized solutions

→ can provide good solutions even if very little info. about the problem is provided.

Limitations

→ encoding in a suitable representation is difficult

→ Natural evolution does not always produce a good solution (frequently converges to a local optimum).

→ difficult to choose parameters such as representation, population size and fitness function

* Applications of Genetic Algorithms in BioinFormatics

→ used to solve multiple sequence alignment problems

→ done with SAGA (Sequence Alignment by Genetic Algorithm)

Working of SAGA

→ Create a random initial population of alignments (Generation 0 or G0)

→ Each generation evolves by selecting and improving alignments

→ Over generations, the fitness of the population (alignment quality is improved)

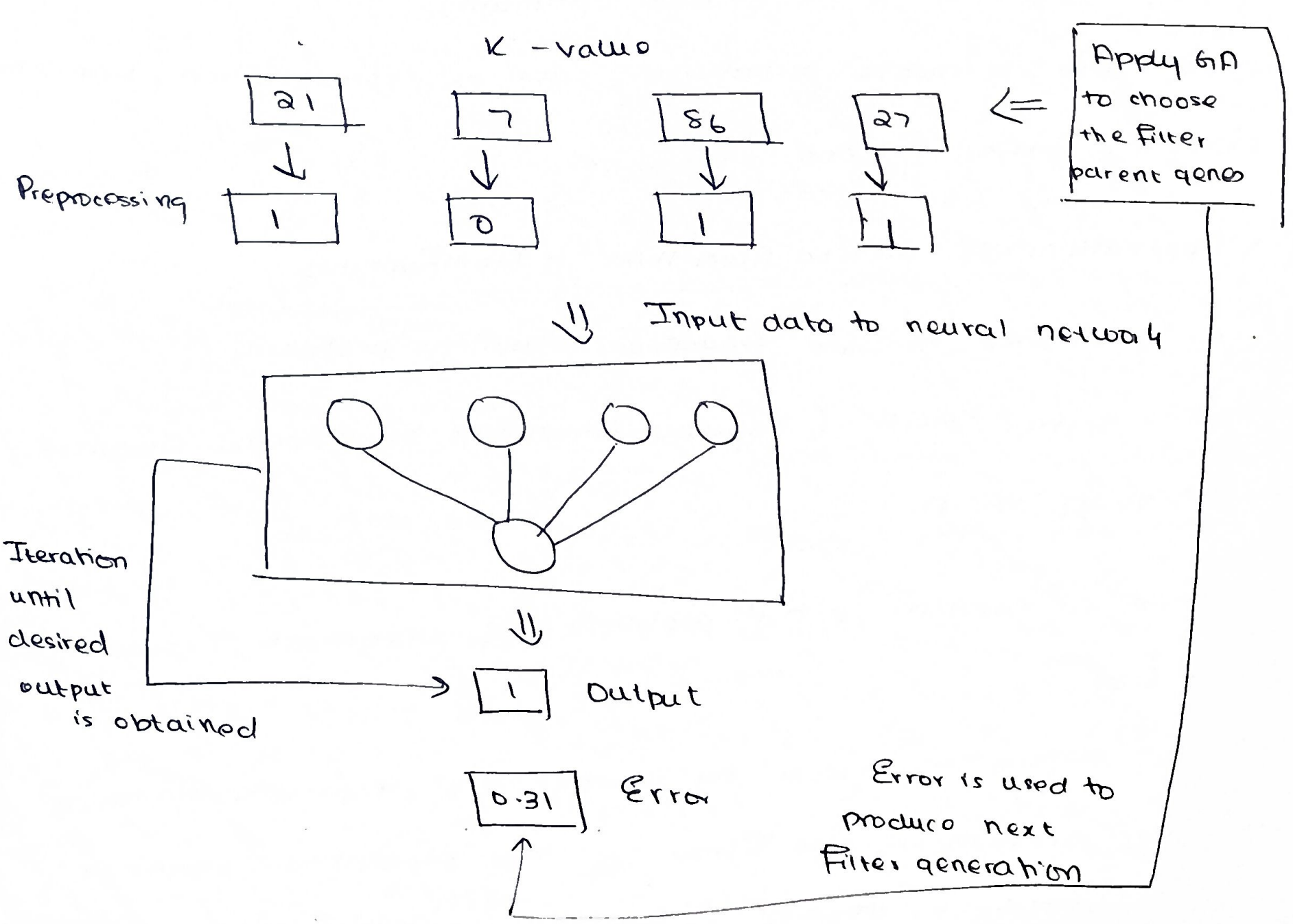
→ Alignments with higher fitness are chosen as parents.

→ New alignments are created by mixing and modifying selected parent alignments w/ mutation and crossover

→ The algorithm repeats this process until no further improvement is possible.

* Neural Genetic Hybrid Models

→ Combine genetic algorithms with ML algorithms ~~with~~ (like KNN) and neural networks, to analyze complex data like gene expression



* Fuzzy Systems

→ A fuzzy system is an expert system that uses fuzzy membership functions and rules to make decisions, in contrast to traditional Boolean logic, where there is strictly only 0 or 1.

→ It allows reasoning with partial truth, handling vagueness and ambiguity - qualities inherent to human think.

* Key Features of Fuzzy Logic

1. Superset of Conventional logic - Fuzzy logic extends Boolean logic by allowing statements to be partially true or false.

2. Mathematical Basis In classical set theory, characteristic fns. define membership, where elements either fully belong or don't. In fuzzy sets, a membership function is used, allowing for partial inclusion based on a real interval

3. Linguistic Variables - A variable whose values are not numbers, but words or sentences in natural language. These are used to construct the fuzzy rules (eg. Temperature)

4. Linguistic Values - Values that the linguistic variables can take (eg. hot, cold, fine, rainy)

5. Fuzzy Rules : conditional statements used within fuzzy logic systems

A fuzzy rule can be:

If x is A, then y is B x, y = linguistic variables
A, B = linguistic values.

→ Other operators like EQUAL, COMPLEMENT (NOT), OR, AND etc. can also be used in fuzzy rules.

* Steps in Fuzzy Inference

Under fuzzification, apply membership functions to the numeric values, to determine the degree of truth for each of the fuzzy sets

Fuzzification

↓

Under rule evaluation, the truth value for the antecedents of each rule is computed, and applied to the consequents of each rule.

Rule Evaluation

↓

Under aggregation, all the fuzzy output sets from each output variable are combined to form a single output fuzzy set for each output variable.

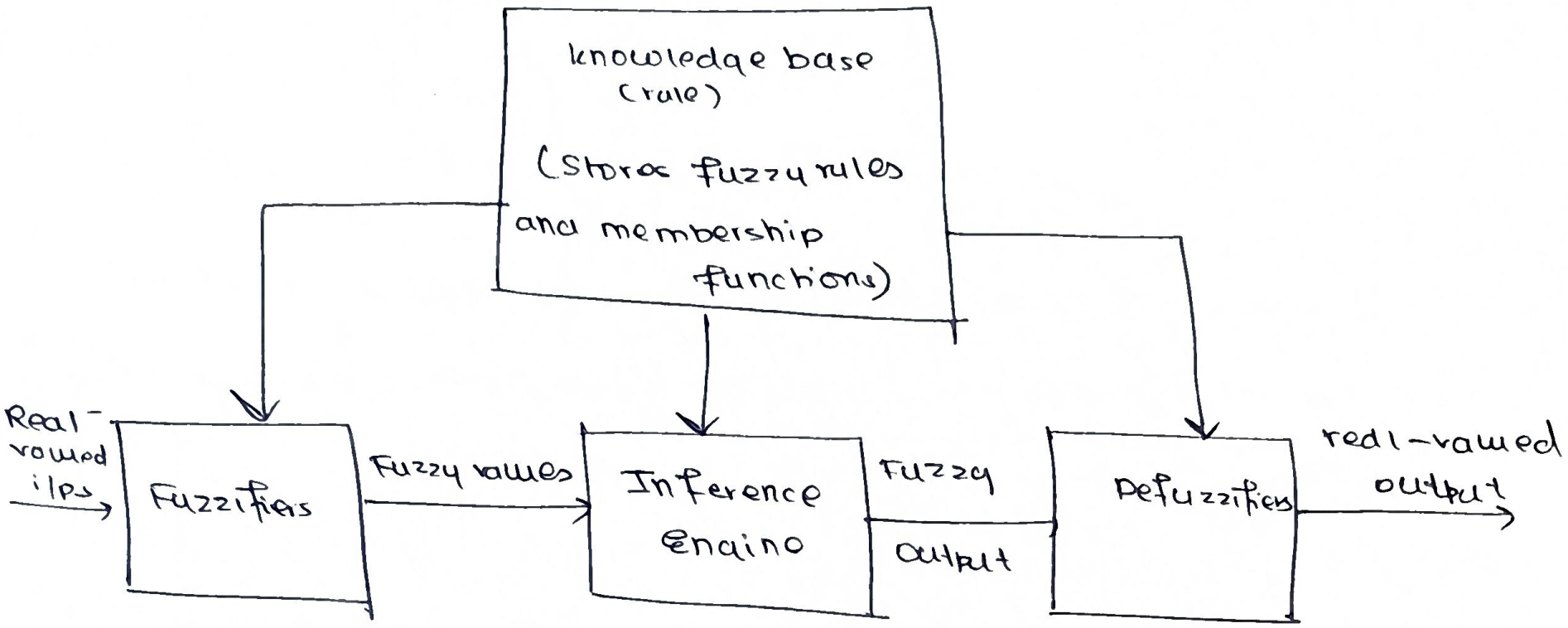
Aggregation of rule consequents

↓

Under defuzzification, convert the fuzzy output back into a numerical value.

Defuzzification

* Components of Fuzzy Expert Systems



Example for Fuzzy Inference Systems