OEC552-SOFT COMPUTING (SC)

UNIT-I
Introduction to Soft Computing

Text book

1. J.S.R.Jang, C.T. Sun and E.Mizutani, "**Neuro-Fuzzy and Soft Computing",** PHI / Pearson Education 2004
2. S.N. Sivanandam & S.N. Deepa, "**Principles of Soft Computing, 2nd Edition**"
by Copyright © 2011 Wiley India Pvt. Ltd. All rights reserved

SOFT COMPUTING (SC)

- Solutions for a complex problems –obtained by incorporating certain processes resembling biological and nature inspired phenomena.
- used intelligently where solutions in polynomial time and remain intractable to conventional mathematical and analytical methods.

Deals with

Imprecision, uncertainty, partial truth and approximation to achieve practicability, robustness and low solution cost.

SC techniques includes Expert systems, Artificial Neural Networks, Fuzzy Logic systems and evolutionary computations.

Evolutionary computation techniques include Evolutionary algorithms, metaheuristics and Swarm intelligent techniques.

Swarm intelligent techniques such as ant colony optimisation, Particle swarm optimisation, bees algorithms and cuckoo search.

SOFT COMPUTING (SC)

Soft Computing (SC):

The symbiotic use of many emerging problem-solving disciplines.

According to Prof. Zadeh:

"...in contrast to traditional hard computing, soft computing exploits the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution-cost, and better rapport with reality"

Soft Computing Main Components:

- Approximate Reasoning
- Search & Optimization
  - ✓ Neural Networks, Fuzzy Logic, Evolutionary Algorithms

PROBLEM SOLVING TECHNIQUES

HARD COMPUTING

SOFT COMPUTING

Precise Models

Approximate Models

Symbolic Logic Reasoning

Traditional Numerical Modeling and Search

Approximate Reasoning

Functional Approximation and Randomized Search

Hard computing vs Soft computing

Hard Computing	Soft computing
Precisely stated analytical model required	Imprecision is tolerable
More Computation time required	As it involves intelligent computational steps, computational time required is less
It involves binary logic crisp systems and numerical analysis	It involves nature inspired systems such as neural networks, fuzzy logic systems and swarm intelligent system.
Precision is observed within the computation	Approximation is obtained in the computation
Imprecision and uncertainity are undesirable properties	Tolerance for imprecision and uncertainty is exploited to achieve tractability, lower cost, high Machine intelligence quotient and economy of communication.
It produces precise answer	It can produce approximate answers

Hard computing vs Soft computing

Hard Computing	Soft computing
Programs are written which follow standard rules of programming	Programs are evolved which require new laws and theories to be created and justified while programming
The outcome is deterministic(i.e., Every trial run, the output is same)	The outcome is stochastic or random in nature and need not be deterministic
It requires exact input data	It can deal with ambiguous and noisy input data
It strictly follows sequential computations	It allows parallel computations
It produces precise answer	It can produce approximate answers

OVERVIEW OF TECHNIQUES IN SOFT COMPUTING

- **Neural Networks**
- **Fuzzy Logic**
- **Genetic Algorithm**
- **Hybrid Systems**

NEURAL NETWORKS

DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

According to Haykin (1994), p. 2:

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network through a learning process.
- Interneuron connection strengths known as synaptic weights are used to store the knowledge

According to Nigrin (1993), p. 11:

A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information.
Furthermore each element operates asynchronously; thus there is no overall system clock.

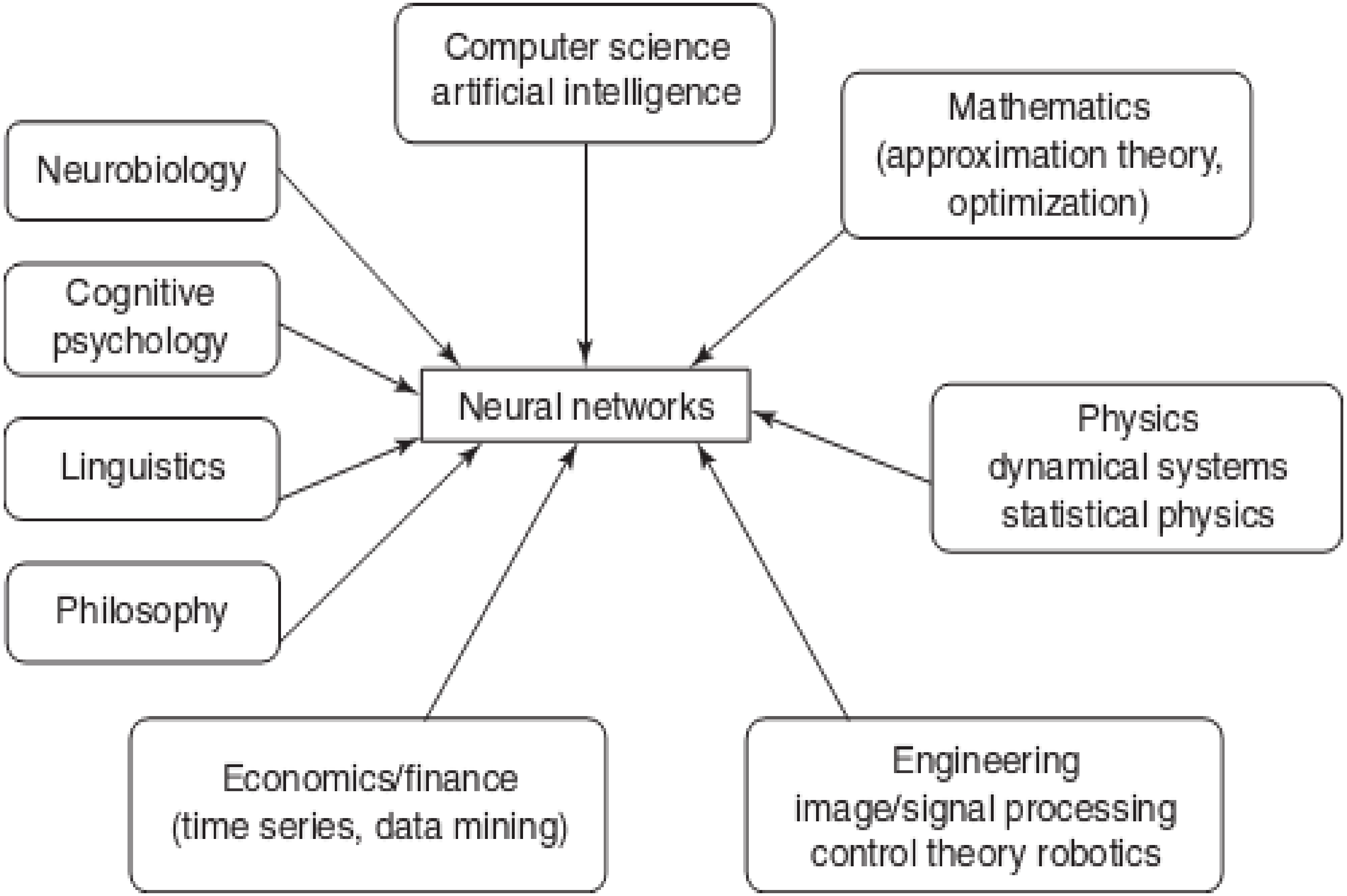
According to Zurada (1992):

Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store and utilize experiential knowledge.

Advantages of Neural Networks

- **Adaptive learning-** -based on data given for training / initial experience
- **Self organization** – create own representation of information it receives during training
- **Real time operation -** computation carried in parallel. Special H/W designed and manufactured
- **Fault tolerance via redundant information coding -** partial destruction leads to corresponding degradation of performance. Some network capabilities may be retained even after major network damage.

MULTIDISCIPLINARY VIEW OF NEURAL NETWORKS



Conventional Computing vs Neuro Computing

Conventional Computing	Neuro Computing
Computational process is sequential and deterministic	Computational process is not sequential and necessarily deterministic
A single processing unit is present-complex central processor	Many simple processing unit-present. They only takes the weighted sum of their inputs from other processors.
Respond to any programmed instruction	Do not respond to programmed instruction. But respond in parallel such as simulated or actual responses for the pattern of inputs present to it.
Separate addresses for storing data	No separate memory addresses for storing data; however information is contained in the overall activation state of the network
Knowledge is centrally located : if certain parts of data is lost, retrieval in not possible	Knowledge is distributed, data retrieval may be possible if a certain part of data is lost. (similar to the memory retrieval in human brain)
Not suited in situations where there are no clear cut algorithmic solutions	Well suited to situations where algorithmic solutions are not possible.

FUZZY LOGIC

- **Origins:** Multivalued Logic for treatment of imprecision and vagueness
  - **1930s**: **Post, Kleene, and Lukasiewicz** attempted to represent undetermined, unknown, and other possible intermediate truth-values.
  - **1937**: **Max Black** suggested the use of a consistency profile to represent vague (ambiguous) concepts.
  - **1965: Zadeh** proposed a complete theory of fuzzy sets (and its isomorphic fuzzy logic), to represent and manipulate ill-defined concepts.

FUZZY LOGIC – LINGUISTIC VARIABLES

- Fuzzy logic gives us a language (with syntax and local semantics) in which we can translate our qualitative domain knowledge.

- Linguistic variables to model dynamic systems

- These variables take linguistic values that are characterized by:

- a label - a sentence generated from the syntax
- a meaning - a membership function determined by a local semantic procedure

FUZZY LOGIC – REASONING METHODS

- The meaning of a linguistic variable may be interpreted as an elastic constraint on its value.
- These constraints are propagated by fuzzy inference operations, based on the generalized modus-ponens.
- An FL Controller (FLC) applies this reasoning system to a Knowledge Base (KB) containing the problem domain heuristics.
- The inference is the result of interpolating among the outputs of all relevant rules.
- The outcome is a membership distribution on the output space, which is defuzzified to produce a crisp output.

GENETIC ALGORITHM **EVOLUTIONARY PROCESS**

Definition of GA : The genetic algorithm is a **probabalistic search algorithm** that iteratively transforms a set (called a population) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the **Darwinian principle of natural selection** and using **operations** that are patterned after naturally occurring genetic operations, such as crossover (sexual recombination) and mutation.

STEPS INVOLVED IN GENETIC ALGORITHM

The genetic algorithms follow the evolution process in the nature to find the better solutions of some complicated problems. Foundations of genetic algorithms are given in Holland (1975) and Goldberg (1989) books.

Genetic algorithms consist the following steps:

- Initialization
- Selection
- Reproduction with crossover and mutation

Selection and reproduction are repeated for each generation until a solution is reached.

During this procedure a certain strings of symbols, known as chromosomes, evaluate toward better solution.

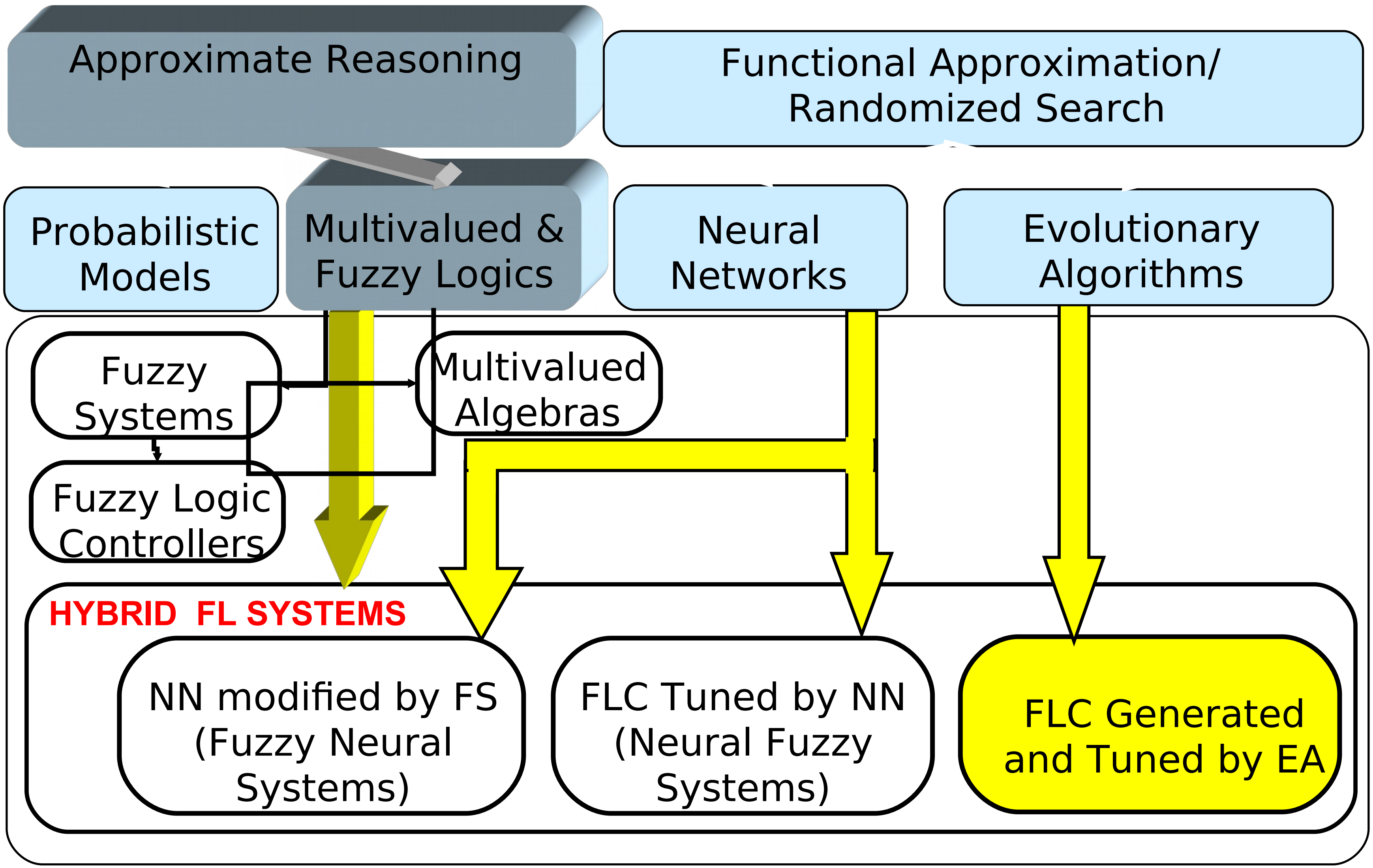
HYBRID SYSTEMS

Hybrid systems enables one to combine various soft computing paradigms and result in a best solution.

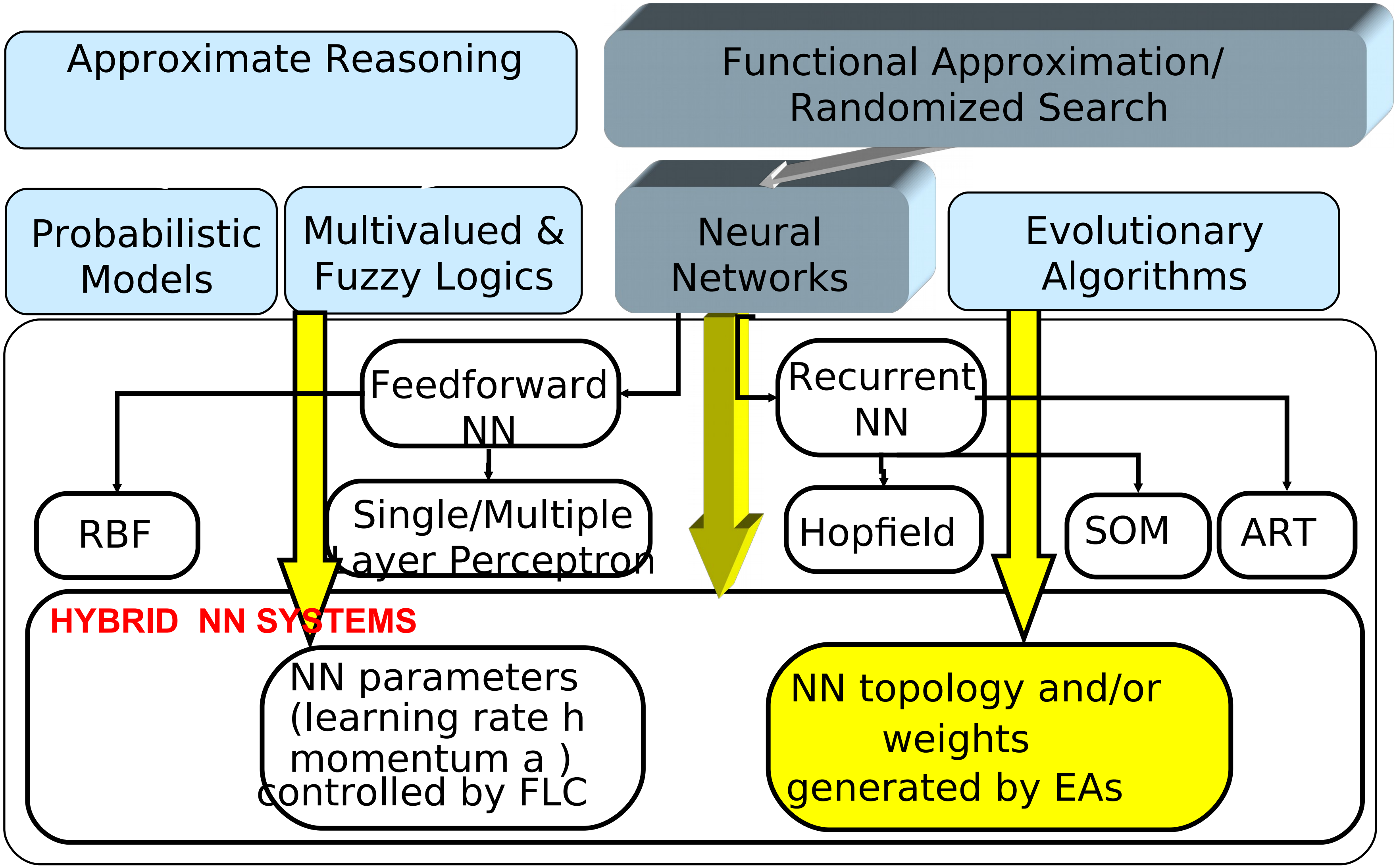
The major three hybrid systems are as follows:

- Hybrid Fuzzy Logic (FL) Systems
- Hybrid Neural Network (NN) Systems
- Hybrid Evolutionary Algorithm (EA) Systems

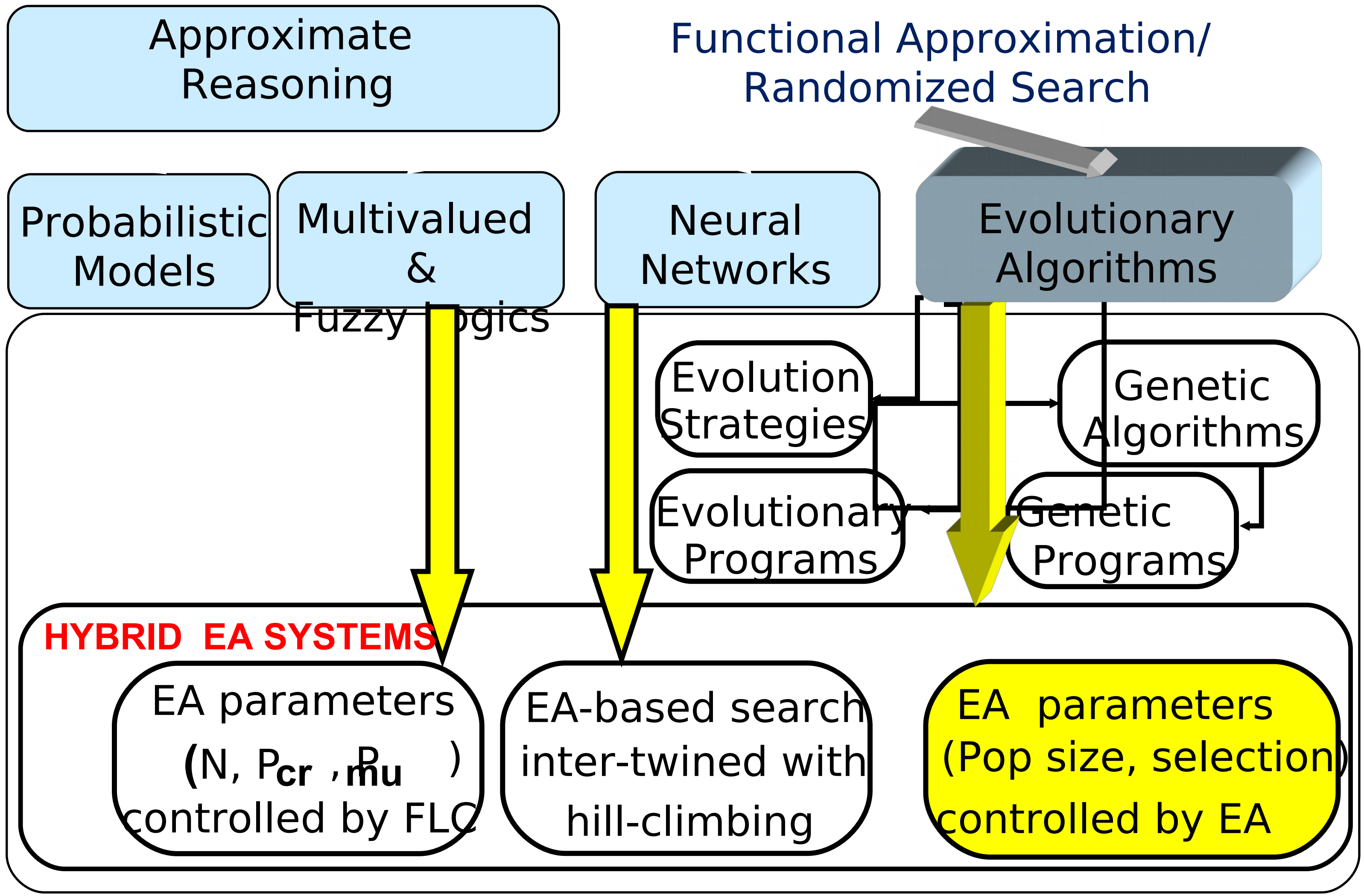
SOFT COMPUTING: HYBRID FL SYSTEMS



SOFT COMPUTING: HYBRID NN SYSTEMS



SOFT COMPUTING: HYBRID EA SYSTEMS



PROPERTIES OF SOFT COMPUTING

- Learning from experimental data
- Soft computing techniques derive their power of generalization from approximating or interpolating to produce outputs from previously unseen inputs by using outputs from previous learned inputs
- Generalization is usually done in a high dimensional space.

ADVANTAGES OF SOFT COMPUTING

- Models based on human reasoning.
- Models can be
- linguistic,
- simple (no number crunching),
- comprehensible (no black boxes),
- fast when computing,
- good in practice.

APPLICATIONS OF SOFT COMPUTING

- Handwriting Recognition
- Image Processing and Data Compression
- Automotive Systems and Manufacturing
- Soft Computing to Architecture
- Decision-support Systems
- Soft Computing to Power Systems
- Neuro Fuzzy systems
- Fuzzy Logic Control
- Machine Learning Applications
- Speech and Vision Recognition Systems
- Process Control and So on

SOFT COMPUTING APPLICATIONS: CONTROL

- Heavy industry (Matsushita, Siemens, Stora-Enso).
- Home appliances (Canon, Sony, Goldstar, Siemens).
- Automobiles (Nissan, Mitsubishi, Daimler-Chrysler, BMW, Volkswagen).
- Spacecrafts (NASA).

SOFT COMPUTING APPLICATIONS: BUSINESS

- supplier evaluation for sample testing,
- customer targeting,
- sequencing,
- scheduling,
- optimizing R&D,
- projects,
- knowledge-based prognosis,
- fuzzy data analysis
- hospital stay prediction,
- TV commercial slot evaluation,
- address matching,
- fuzzy cluster analysis,
- sales prognosis for mail order house,
- multi-criteria optimization, etc.

SOFT COMPUTING APPLICATIONS: FINANCE

- fuzzy scoring for mortgage applicants,
- creditworthiness assessment,
- fuzzy-enhanced score card for lease risk assessment,
- risk profile analysis,
- insurance fraud detection,
- cash supply optimization,
- foreign exchange trading,
- insider trading,
- trading surveillance,
- investor classification, etc.

SOFT COMPUTING APPLICATIONS: OTHERS

Statistics, Social sciences, Behavioral sciences, Robotics, Biology, Medicine

ARTIFICIAL NEURAL NETWORKS: AN INTRODUCTION

DEFINITION OF NEURAL NETWORKS

According to the DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

- … a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

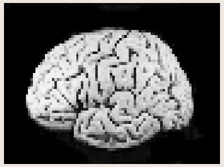
According to Haykin (1994), p. 2:

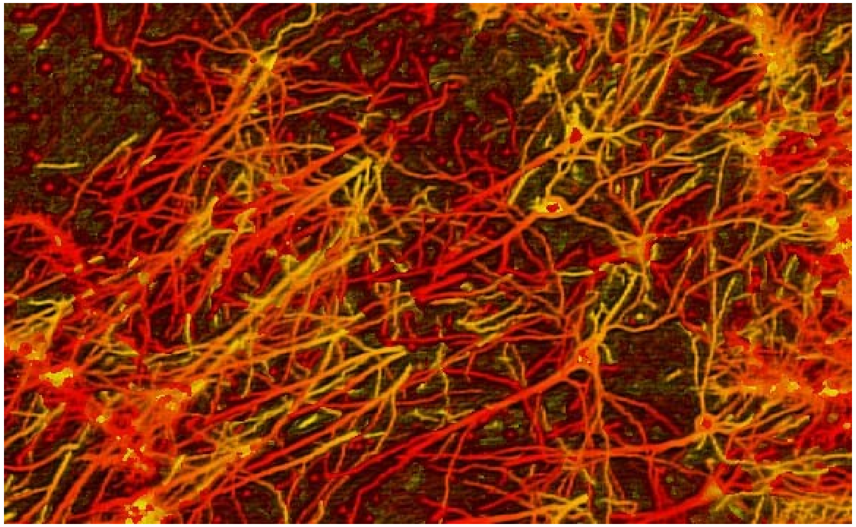
A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network through a learning process.
- Interneuron connection strengths known as synaptic weights are used to store the knowledge.

BRAIN COMPUTATION

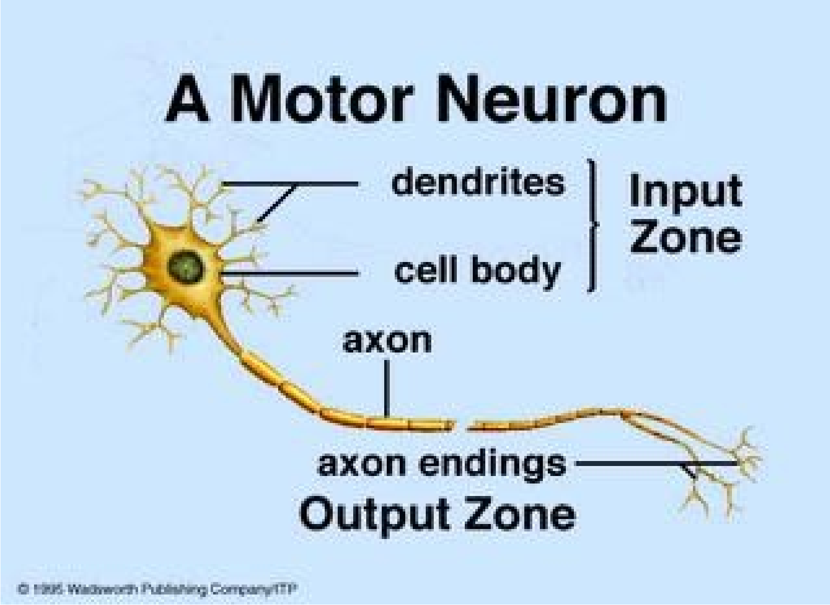
The **human brain** contains about 10 billion nerve cells, or neurons. On average, each neuron is connected to other neurons through approximately 10,000 synapses.

	processing elements	element size	energy use	processing speed	style of computation	fault tolerant	learns	intelligent, conscious
	10^{14} synapses	10^{-6} m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^{8} transistors	10^{-6} m	30 W (CPU)	10^{9} Hz	serial, centralized	no	a little	not (yet)



INTERCONNECTIONS IN BRAIN

BIOLOGICAL (MOTOR) NEURON



ARTIFICIAL NEURAL NET

- Information-processing system.
- Neurons process the information.
- The signals are transmitted by means of connection links.
- The links possess an associated weight.
- The output signal is obtained by applying activations to the net input.

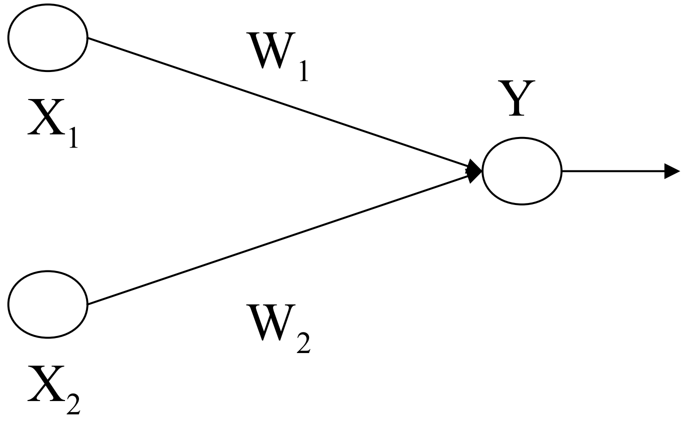
MOTIVATION FOR NEURAL NET

- Scientists are challenged to use machines more effectively for tasks currently solved by humans.
- Symbolic rules don't reflect processes actually used by humans.
- Traditional computing excels in many areas, but not in others.

The major areas being:

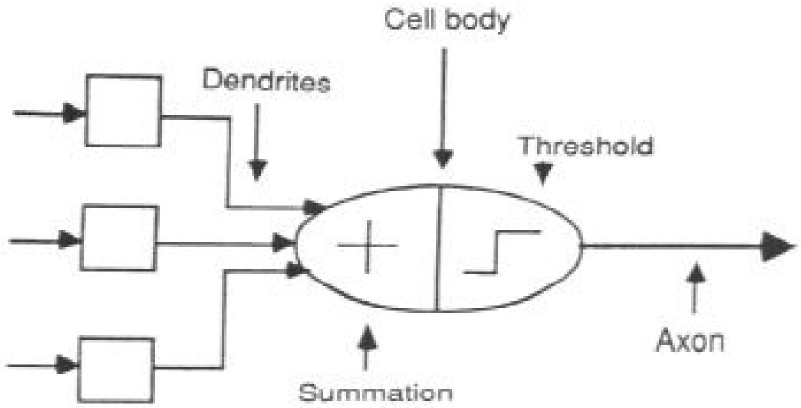
- Massive parallelism
- Distributed representation and computation
- Learning ability
- Generalization ability
- Adaptivity
- Inherent contextual information processing
- Fault tolerance
- Low energy consumption.

ARTIFICIAL NEURAL NET



Simple artificial neural net with two input neurons (X_1, X_2) and one output neuron (Y). The inter connected weights are given by W_1 and W_2.

ASSOCIATION OF BIOLOGICAL NET WITH ARTIFICIAL NET



PROCESSING OF AN ARTIFICIAL NET

The neuron is the basic information processing unit of a NN. It consists of:

1. A set of links, describing the neuron inputs, with weights $W_1, W_2, \ldots, W_m$.
2. An adder function (linear combiner) for computing the weighted sum of the inputs (real numbers):
3. Activation function for limiting the amplitude of the neuron output.

$$u = \sum_{j=1}^{m} w_j x_j$$

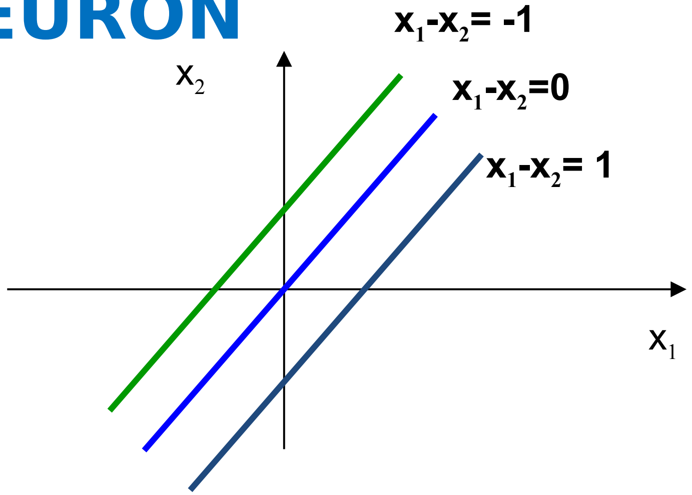
$$y = \varphi(u + b)$$

BIAS OF AN ARTIFICIAL NEURON

The bias value is added to the weighted sum $\sum w_i x_i$ so that we can transform it from the origin.

$$Y_{in} = \sum w_i x_i + b,$$

where b is the bias



MULTI LAYER ARTIFICIAL NEURAL NET

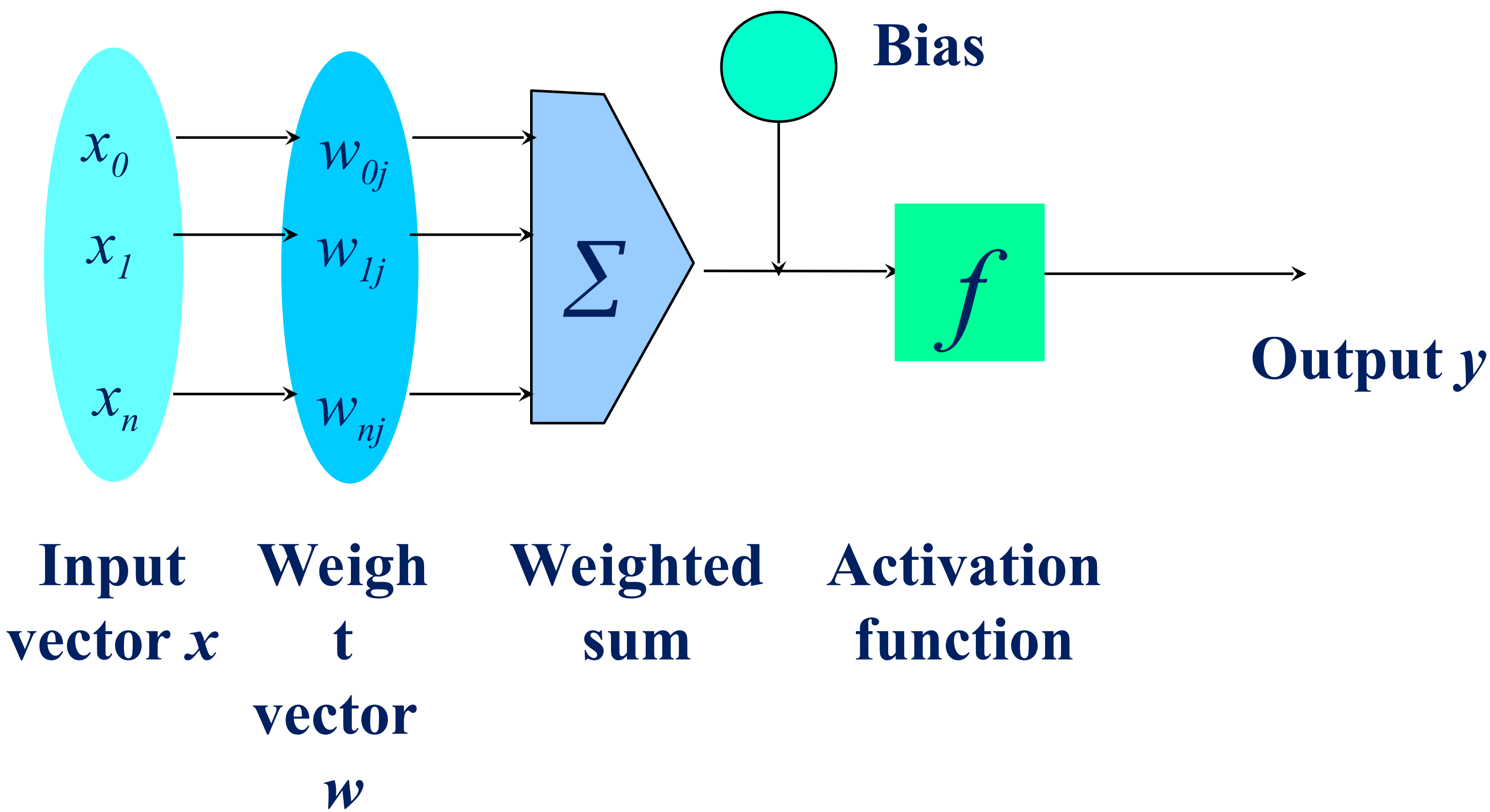
INPUT: records without class attribute with normalized attributes values.

INPUT VECTOR: $X = \{ x_1, x_2, \ldots, x_n \}$ where n is the number of (non-class) attributes.

INPUT LAYER: there are as many nodes as non-class attributes, i.e. as the length of the input vector.

HIDDEN LAYER: the number of nodes in the hidden layer and the number of hidden layers depends on implementation.

OPERATION OF A NEURAL NET



WEIGHT AND BIAS UPDATION

Per Sample Updating

•updating weights and biases after the presentation of each sample.

Per Training Set Updating (Epoch or Iteration)

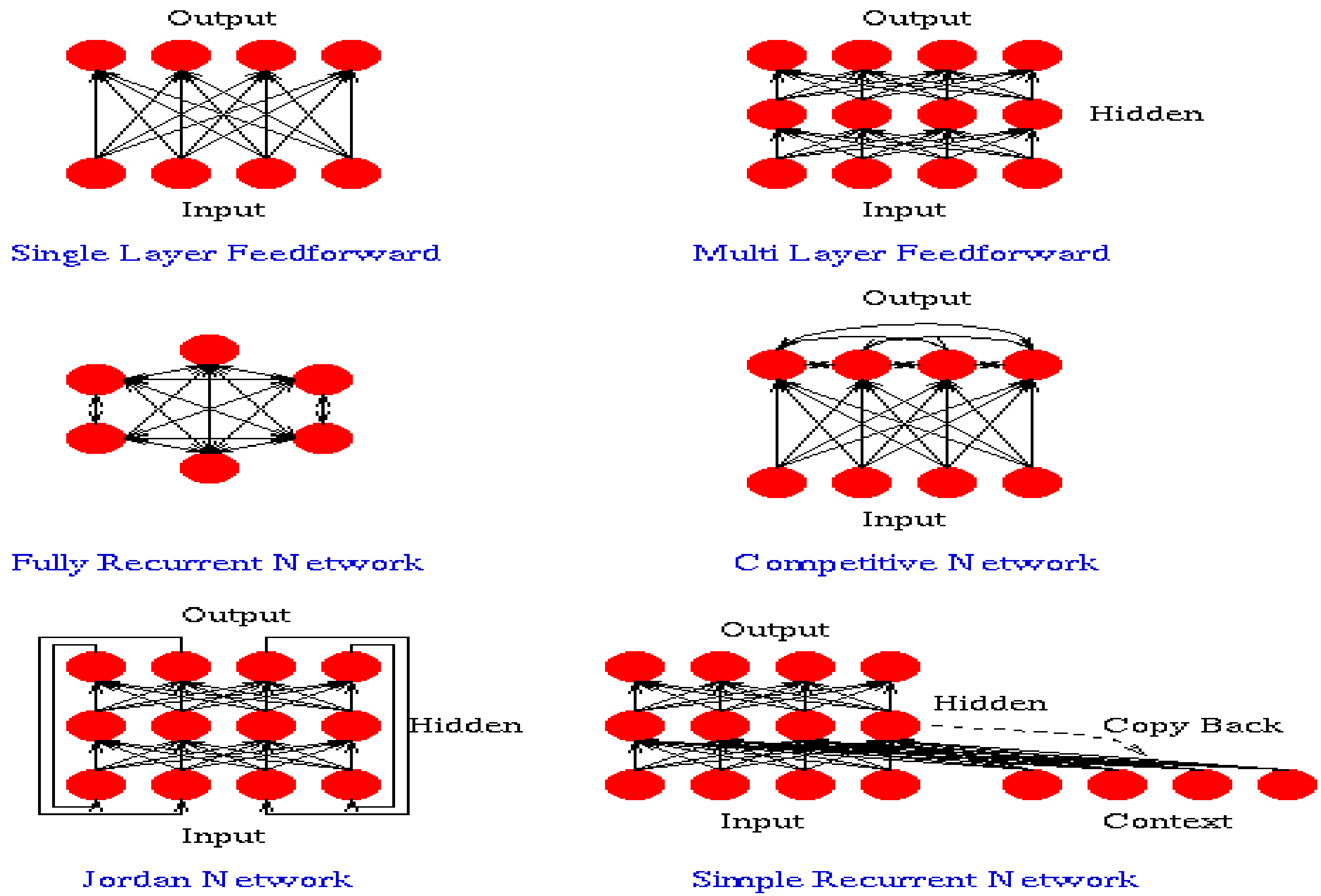
•weight and bias increments could be accumulated in variables and the weights and biases updated after all the samples of the training set have been presented.

STOPPING CONDITION

- All change in weights (Δwij) in the previous epoch are below some threshold, or
- The percentage of samples misclassified in the previous epoch is below some threshold, or
- A pre-specified number of epochs has expired.
- In practice, several hundreds of thousands of epochs may be required before the weights will converge.

BUILDING BLOCKS OF ARTIFICIAL NEURAL NET

- Network Architecture (Connection between Neurons)
- Setting the Weights (Training)
- Activation Function



LAYER PROPERTIES

- **Input Layer:** Each input unit may be designated by an attribute value possessed by the instance.
- **Hidden Layer:** Not directly observable, provides nonlinearities for the network.
- **Output Layer:** Encodes possible values.

TRAINING PROCESS

- **Supervised Training** - Providing the network with a series of sample inputs and comparing the output with the expected responses.
- **Unsupervised Training** - Most similar input vector is assigned to the same output unit.
- **Reinforcement Training** - Right answer is not provided but indication of whether ‘right’ or ‘wrong’ is provided.

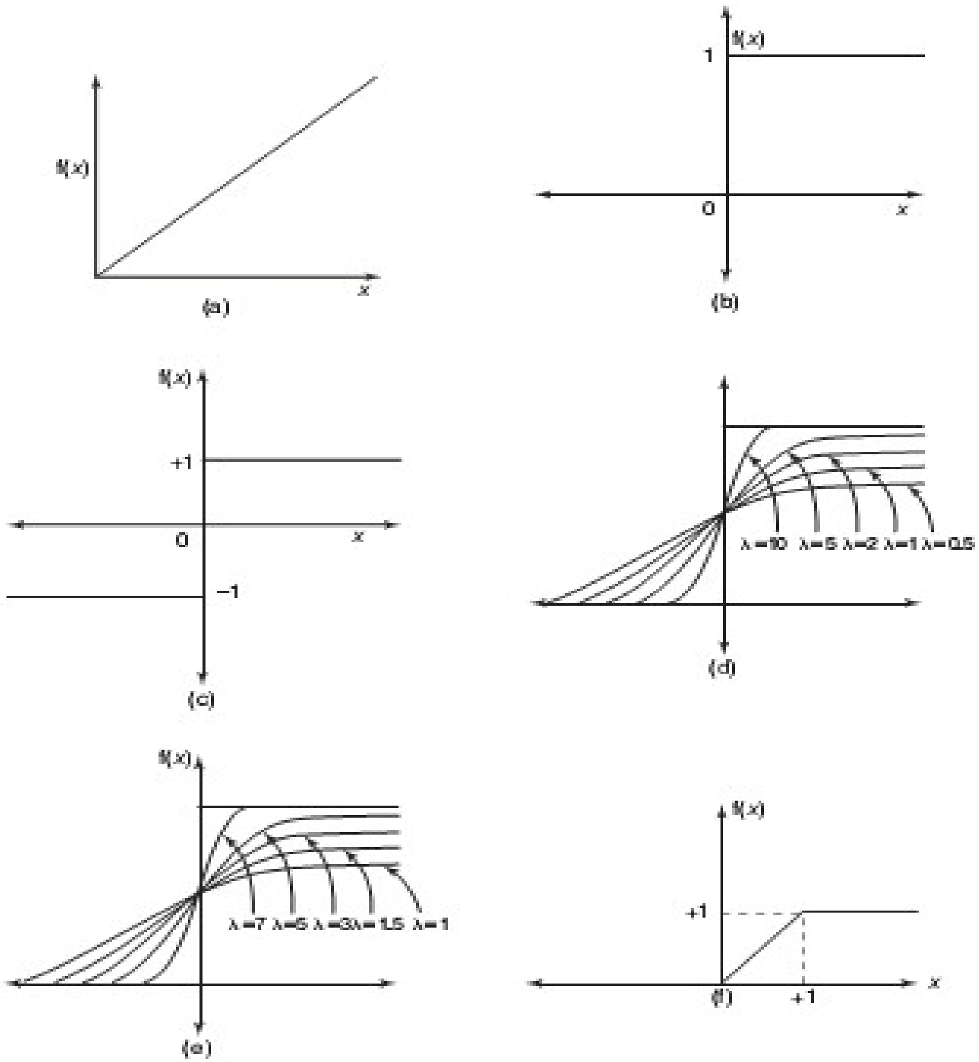
ACTIVATION FUNCTION

- **ACTIVATION LEVEL – DISCRETE OR CONTINUOUS**

- **HARD LIMIT FUCNTION (DISCRETE)**
  - Binary Activation function
  - Bipolar activation function
  - Identity function

- **SIGMOIDAL ACTIVATION FUNCTION (CONTINUOUS)**
  - Binary Sigmoidal activation function
  - Bipolar Sigmoidal activation function

ACTIVATION FUNCTION



Activation functions:

(A) Identity

(B) Binary step

(C) Bipolar step

(D) Binary sigmoidal

(E) Bipolar sigmoidal

(F) Ramp

CONSTRUCTING ANN

- Determine the network properties:
  - Network topology
  - Types of connectivity
  - Order of connections
  - Weight range

- Determine the node properties:
  - Activation range

- Determine the system dynamics
  - Weight initialization scheme
  - Activation – calculating formula
  - Learning rule

PROBLEM SOLVING

- Select a suitable NN model based on the nature of the problem.
- Construct a NN according to the characteristics of the application domain.
- Train the neural network with the learning procedure of the selected model.
- Use the trained network for making inference or solving problems.

NEURAL NETWORKS

- **Neural Network** learns by adjusting the weights so as to be able to correctly classify the training data and hence, after testing phase, to classify unknown data.
- **Neural Network** needs long time for training.
- **Neural Network** has a high tolerance to noisy and incomplete data.

SALIENT FEATURES OF ANN

- Adaptive learning
- Self-organization
- Real-time operation
- Fault tolerance via redundant information coding
- Massive parallelism
- Learning and generalizing ability
- Distributed representation

FEW APPLICATIONS OF NEURAL

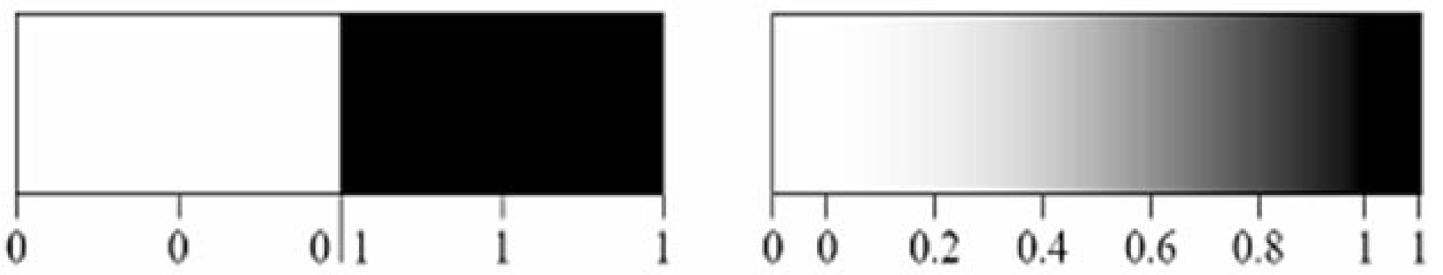
- Aerospace
- Automotive
- Banking
- Credit Card Activity Checking
- Defense
- Electronics
- Entertainment
- Financial
- Industrial
- Insurance
- Insurance
- Manufacturing
- Medical
- Oil and Gas
- Robotics
- Speech
- Securities
- Telecommunications
- Transportation

INTRODUCTION TO FUZZY LOGIC, CLASSICAL SETS AND FUZZY SETS

FUZZY LOGIC

- Fuzzy logic is **the logic** underlying **approximate**, rather than exact, **modes of reasoning**.
- It is an extension of multivalued logic: **Everything**, including truth, **is a matter of degree**.
- It contains as special cases **not only** the classical two-value logic and multivalue logic systems, **but also** probabilistic logic.
- A proposition ***p*** has a **truth value**
  - 0 or 1 in two-value system,
  - element of a set T in multivalue system,
  - **Range over the fuzzy subsets of T** in fuzzy logic.

- Boolean logic uses sharp distinctions.
- Fuzzy logic reflects how people think.
- Fuzzy logic is a set of mathematical principles for



(a) Boolean Logic. (b) Multi-valued Logic.

- Fuzzy logic is a set of mathematical principles for knowledge representation and reasoning based on degrees of membership.

TYPES AND MODELING OF UNCERTAINTY

Stochastic Uncertainty:

- The probability of hitting the target is 0.8

Lexical Uncertainty:

- "Tall Men", "Hot Days", or "Stable Currencies"
- We will probably have a successful business year.
- The experience of expert A shows that B is Likely to Occur. However, expert C is convinced This Is Not True.

Example: One finds in desert two bottles of fluids with the following labels:

- ✓ bottle 1: there is a probability of 5% that this bottle is poisoned.
- ✓ bottle 2: this bottle contains a liquid which belongs to the set of drinkable water with membership function value of 0.95.

FUZZY vs PROBABILITY

- Let L=set of all liquids
  - £ be the subset ={all drinkable liquids}
- Suppose you had been in desert (you must drink!) and you come up with two bottles marked C and A.
- **Bottle C is labeled $\mu_{£}(C)=0.95$ and bottle A is labeled $Pr[A \in £]=0.95$**

• C could contain swamp water, but would not contain any poison. Membership of 0.95 means that the contents of C are fairly similar to perfectly drinkable water.

• The probability that A is drinkable is 0.95, means that over a long run of experiments, the context of A are expected to be drinkable in about 95% of the trials. In other cases it may contain poison.

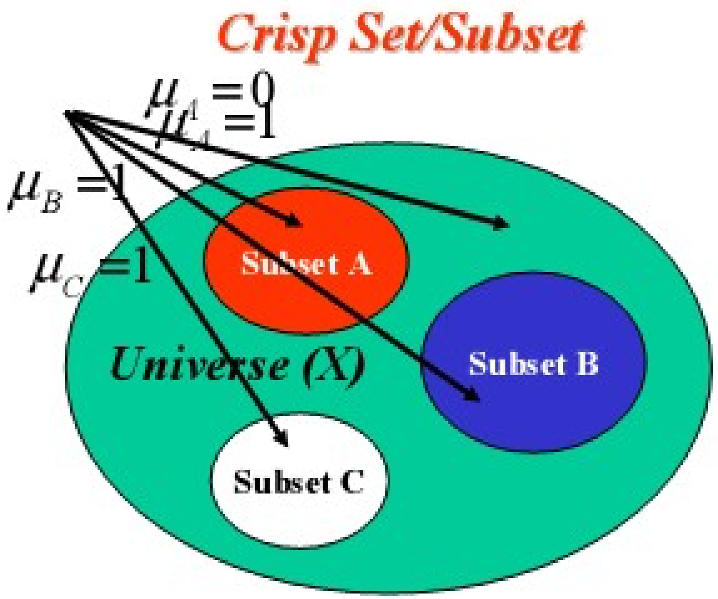
NEED OF FUZZY LOGIC

- Based on intuition and judgment.
- No need for a mathematical model.
- Provides a smooth transition between members and nonmembers.
- Relatively simple, fast and adaptive.
- Less sensitive to system fluctuations.
- Can implement design objectives, difficult to express mathematically, in linguistic or descriptive rules.

CLASSICAL SETS (CRISP SETS)

Conventional or crisp sets are Binary. An element either belongs to the set or does not.

{True, False} **{1, 0}**



OPERATIONS ON CRISP SETS

- UNION: $A \cup B = \{x | x \in A \text{ or } x \in B\}$
- INTERSECTION: $A \cap B = \{x | x \in A \text{ and } x \in B\}$
- COMPLEMENT: $\bar{A} = \{x | x \notin A, x \in X\}$
- DIFFERENCE:

$$\begin{aligned} A|B \text{ or } (A - B) &= \{x | x \in A \text{ and } x \notin B\} \\ &= A - (A \cap B) \end{aligned}$$

PROPERTIES OF CRISP SETS

The various properties of crisp sets are as follows:

1. Commutativity

$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$

2. Associativity

$$A \cup (B \cup C) = (A \cup B) \cup C$$
$$A \cap (B \cap C) = (A \cap B) \cap C$$

3. Distributivity

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

4. Idempotency

$$A \cup A = A$$
$$A \cap A = A$$

5. Transitivity

$$\text{If } A \subseteq B \subseteq C, \text{ then } A \subseteq C$$

PROPERTIES OF CRISP SETS

6. Identity

$$A \cup \phi = A, \quad A \cap \phi = \phi$$
$$A \cap X = A, \quad A \cap X = X$$

7. Involution (double negation)

$$\bar{\bar{A}} = A$$

8. Law of excluded middle

$$A \cup \bar{A} = X$$

9. Law of contradiction

$$A \cap \bar{A} = \phi$$

10. DeMorgan's law

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$
$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

FUZZY SETS

Rules of thumb frequently stated in "fuzzy" linguistic terms.

John is *tall.*

If someone is *tall* **and** *well-built*

then his basketball skill is good.

<u>Fuzzy Sets</u>

$0 \leq \mu S(x) \leq 1$ ------ $\mu S(x)$ (or $\mu(S, x)$) is the **degree** of membership of x in set S

$\mu S(x) = 0$ x is not at all in S

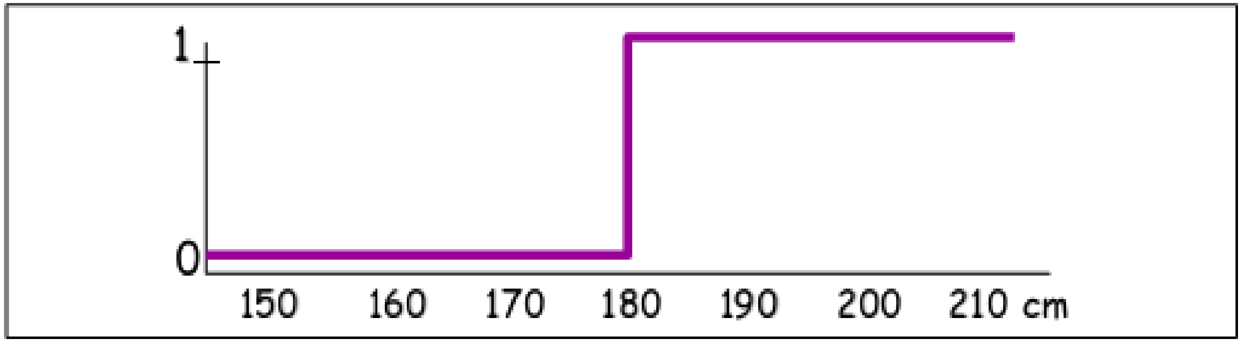
$\mu S(x) = 1$ x is fully in S.

If $\mu S(x) = 0$ or 1, then the set S is **crisp.**

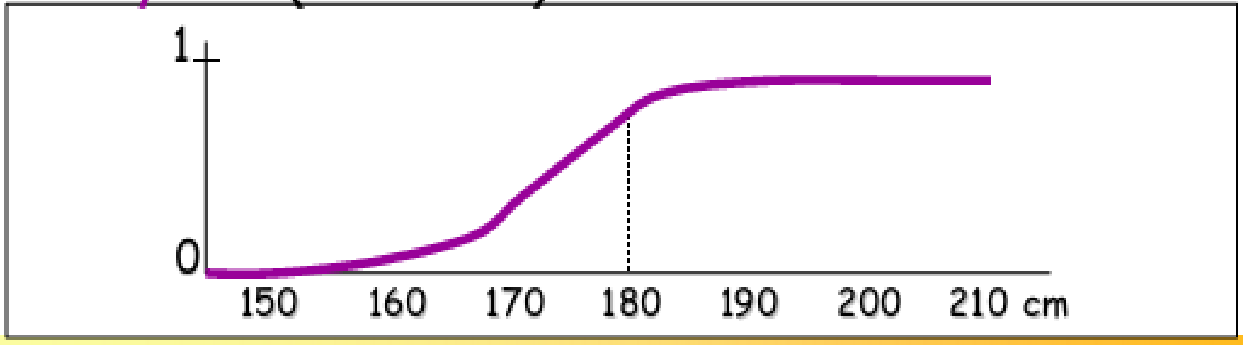
Fuzzy set

- Is a function f: domain → [0,1]

Crisp set (tall men):



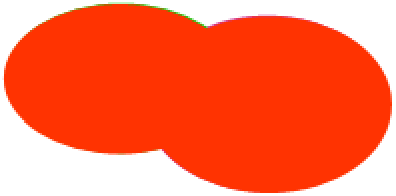
Fuzzy set (tall men):



OPERATIONS ON FUZZY SETS

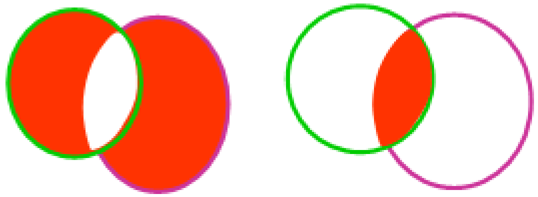
- Union: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$
- Intersection: $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
- Complement: $\mu_{\neg A}(x) = 1 - \mu_A(x)$

Fuzzy union operation or fuzzy *OR*



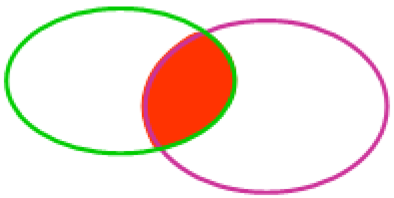
$$\mu_{A+B}(x) = \max\left[\mu_A(x), \mu_B(x)\right]$$

Complement operation



$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Fuzzy intersection operation or fuzzy *AND*



$$\mu_{A \cdot B}(x) = \min\left[\mu_A(x), \mu_B(x)\right]$$

PROPERTIES OF FUZZY SETS

The same as for crisp sets

- Commutativity
- Associativity
- Distributivity
- Idempotency
- Identity
- De Morgan’s Laws
- …

1. Commutativity

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup \underset{\sim}{A}$$
$$\underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{B} \cap \underset{\sim}{A}$$

2. Associativity

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cup \underset{\sim}{C}$$
$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cap \underset{\sim}{C}$$

3. Distributivity

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cap (\underset{\sim}{A} \cup \underset{\sim}{C})$$
$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cup (\underset{\sim}{A} \cap \underset{\sim}{C})$$

4. Idempotency

$$\underset{\sim}{A} \cup \underset{\sim}{A} = \underset{\sim}{A}$$
$$\underset{\sim}{A} \cap \underset{\sim}{A} = \underset{\sim}{A}$$

5. Identity

$$\underset{\sim}{A} \cup \phi = \underset{\sim}{A} \text{ and } \underset{\sim}{A} \cup U = U(\text{universal set})$$
$$\underset{\sim}{A} \cap \phi = \phi \text{ and } \underset{\sim}{A} \cap U = \underset{\sim}{A}$$

6. Involution (double negation)

$$\bar{\bar{\underset{\sim}{A}}} = \underset{\sim}{A}$$

7. Transitivity

$$\text{If } \underset{\sim}{A} \subseteq \underset{\sim}{B} \subseteq \underset{\sim}{C}, \text{ then } \underset{\sim}{A} \subseteq \underset{\sim}{C}$$

8. Demorgan's law

$$\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = \overline{\underset{\sim}{A}} \cap \overline{\underset{\sim}{B}}$$
$$\overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = \overline{\underset{\sim}{A}} \cup \overline{\underset{\sim}{B}}$$

CLASSICAL RELATIONS AND FUZZY RELATIONS

RELATIONS

- Relations represent mappings between sets and connectives in logic.
- A classical binary relation represents the presence or absence of a connection or interaction or association between the elements of two sets.
- Fuzzy binary relations are a generalization of crisp binary relations, and they allow various degrees of relationship (association) between elements.

CRISP CARTESIAN PRODUCT

Lets consider properties of crisp relations first and then extend the mechanism to fuzzy sets.

Definition of (crisp) Product set: Let A and B be two non-empty sets, the product set or Cartesian product $A \times B$ is defined as follows,

$$A \times B = \{(a, b) \mid a \in A, b \in B\}$$

(a set of ordered pairs a,b)

CRISP RELATIONS

Cartesian product of *n* sets

$$A_1 \times A_2 \times \ldots \times A_n = \prod_{i=1}^{n} A_i = \{(a_1, \ldots, a_n) \mid a_1 \in A_1, \ldots, a_n \in A_n\}$$

Definition of Binary Relation

If A and B are two sets and there is a specific property between elements x of A and y of B, this property can be described using the ordered pair (x, y). A set of such (x, y) pairs, $x \in A$ and $y \in B$, is called a relation R.

$$R = \{ (x,y) \mid x \in A, y \in B \}$$

Definition of *n*-ary relation

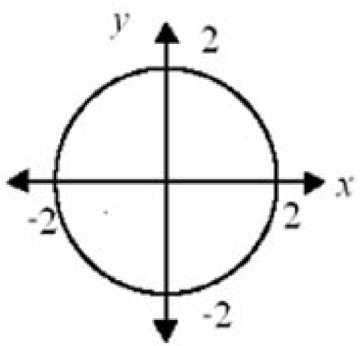
For sets $A_1, A_2, A_3, \ldots, A_n$, the relation among elements $x_1 \in A_1, x_2 \in A_2, x_3 \in A_3, \ldots, x_n \in A_n$ can be described by n-tuple $(x_1, x_2, \ldots, x_n)$. A collection of such n-tuples $(x_1, x_2, x_3, \ldots, x_n)$ is a relation R among $A_1, A_2, A_3, \ldots, A_n$.

$$(x_1, x_2, x_3, \ldots, x_n) \in R,$$
$$R \subseteq A_1 \times A_2 \times A_3 \times \ldots \times A_n$$

CRISP BINARY RELATIONS

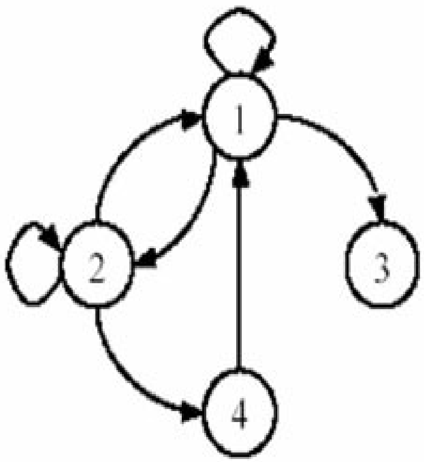
Examples of binary relations

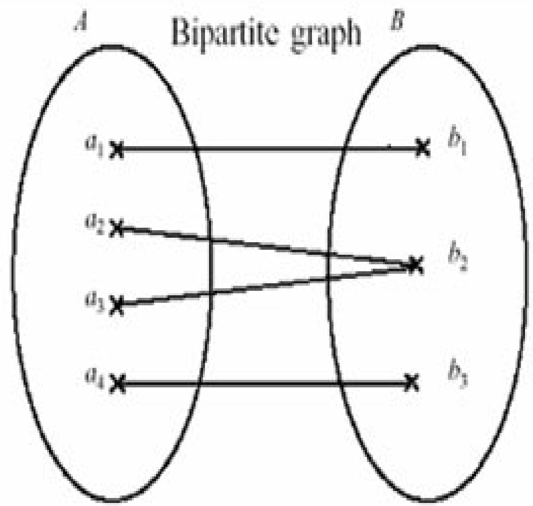
Coordinate diagram



Relation of $x^2 + y^2 = 4$

Binary relation from A to B





Relation matrix

R	b_1	b_2	b_3
a_1	1	0	0
a_2	0	1	0
a_3	0	1	0
a_4	0	0	1

OPERATIONS ON CRISP RELATIONS

Function-theoretic operations for the two crisp relations (R, S) are defined as follows:

1. Union

$$R \cup S \to \chi_{R \cup S}(x, y) : \chi_{R \cup S}(x, y) = \max[\chi_R(x, y), \chi_S(x, y)]$$

2. Intersection

$$R \cap S \to \chi_{R \cap S}(x, y) : \chi_{R \cap S}(x, y) = \min[\chi_R(x, y), \chi_S(x, y)]$$

3. Complement

$$\overline{R} \to \chi_{\overline{R}}(x, y) : \chi_{\overline{R}}(x, y) = 1 - \chi_{\overline{R}}(x, y)$$

4. Containment

$$R \subset S \to \chi_R(x, y) : \chi_R(x, y) \leq \chi_S(x, y)$$

5. Identity

$$\phi \to \phi_R \text{ and } X \to E_R$$

PROPERTIES OF CRISP RELATIONS

The properties of crisp sets (given below) hold good for crisp relations as well.

- Commutativity,
- Associativity,
- Distributivity,
- Involution,
- Idempotency,
- DeMorgan's Law,
- Excluded Middle Laws.

COMPOSITION ON CRISP RELATIONS

The composition operations are of two types:

1. Max-min composition
2. Max-product composition.

The max-min composition is defined by the function theoretic expression as

$$T = R \circ S$$
$$\chi_T(x, z) = \bigvee_{y \in Y} [\chi_R(x, y) \wedge \chi_S(y, z)]$$

The max-product composition is defined by the function theoretic expression as

$$T = R \circ S$$
$$\chi_T(x, z) = \bigvee_{y \in Y} [\chi_R(x, y) \cdot \chi_S(y, z)]$$

FUZZY CARTESIAN PRODUCT

Let R be a fuzzy subset of M and S be a fuzzy subset of N. Then the Cartesian product R × S is a fuzzy subset of N × M such that

$$\forall \vec{x} \in M, \vec{y} \in N \ \mu_{R \times S}(\vec{x}, \vec{y}) = \min\left(\mu_R(\vec{x}), \mu_S(\vec{y})\right)$$

Example:

Let R be a fuzzy subset of {a, b, c} such that R = a/1 + b/0.8 + c/0.2 and S be a fuzzy subset of {1, 2, 3} such that S = 1/1 + 3/0.8 + 2/0.5. Then R x S is given by

$$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 0.5 & 0.9 \\ 0.8 & 0.5 & 0.8 \\ 0.2 & 0.2 & 0.2 \end{bmatrix} \end{array}$$

FUZZY RELATION

A fuzzy relation R is a mapping from the Cartesian space X x Y to the interval [0,1], where the strength of the mapping is expressed by the membership function of the relation $\mu_R(x,y)$

$$\mu_R : A \times B \rightarrow [0,\ 1]$$
$$R = \{((x,\ y),\ \mu_R(x,\ y)) \mid \mu_R(x,\ y) \geq 0\ ,\ \ x \in A,\ \ y \in B\}$$

Crisp relation vs. Fuzzy relation

(a) Crisp relation (b) Fuzzy relation

Corresponding fuzzy relation matrix

A \ A	a	b	c	d
a	0.0	0.0	0.8	0.0
b	1.0	0.0	0.0	0.0
c	0.0	0.9	0.0	1.0
d	0.0	0.0	0.0	0.0

OPERATIONS ON FUZZY RELATION

The basic operation on fuzzy sets also apply on fuzzy relations.

1. Union:

$$\mu_{\underset{\sim}{R} \cup \underset{\sim}{S}}(x, y) = \max[\mu_{\underset{\sim}{R}}(x, y), \mu_{\underset{\sim}{S}}(x, y)]$$

2. Intersection:

$$\mu_{\underset{\sim}{R} \cap \underset{\sim}{S}}(x, y) = \min[\mu_{\underset{\sim}{R}}(x, y), \mu_{\underset{\sim}{S}}(x, y)]$$

3. Complement:

$$\mu_{\overline{\underset{\sim}{R}}}(x, y) = 1 - \mu_{\underset{\sim}{R}}(x, y)$$

4. Containment:

$$\underset{\sim}{R} \subset \underset{\sim}{S} \Rightarrow \mu_{\underset{\sim}{R}}(x, y) \leq \mu_{\underset{\sim}{S}}(x, y)$$

5. Inverse:
The inverse of a fuzzy relation R on $X \times Y$ is denoted by R^{-1}. It is a relation on $Y \times X$ defined by $R^{-1}(y, x) = R(x, y)$ for all pairs $(y, x) \in Y \times X$.

6. Projection:
For a fuzzy relation $R(X, Y)$, let $[R \downarrow Y]$ denote the projection of R onto Y. Then $[R \downarrow Y]$ is a fuzzy relation in Y whose membership function is defined by

$$\mu_{[R\downarrow Y]}(x, y) = \max_{x} \mu_{\underset{\sim}{R}}(x, y)$$

The projection concept can be extended to an n-ary relation $R(x_1, x_2, \ldots, x_n)$.

PROPERTIES OF FUZZY RELATIONS

The properties of fuzzy sets (given below) hold good for fuzzy relations as well.

- Commutativity,
- Associativity,
- Distributivity,
- Involution,
- Idempotency,
- DeMorgan's Law,
- Excluded Middle Laws.

COMPOSITION OF FUZZY RELATIONS

Two fuzzy relations R and S are defined on sets A, B and C. That is, $R \subseteq A \times B$, $S \subseteq B \times C$. The composition $S \bullet R = SR$ of two relations R and S is expressed by the relation from A to C:

For $(x, y) \in A \times B$, $(y, z) \in B \times C$,

$$\mu_{S \bullet R}(x, z) = \max_y [\min(\mu_R(x, y), \mu_S(y, z))]$$

$$= \vee_y [\mu_R(x, y) \wedge \mu_S(y, z)]$$

$M_{S \bullet R} = M_R \bullet M_S$ (matrix notation)

(max-min composition)

Example:

$$X=\{x_1,x_2\},\ Y=\{y_1,y_2\},\quad \text{and} \quad Z=\{z_1,z_2,z_3\}$$

Consider the following fuzzy relations:

$$\tilde{R}=\begin{array}{c} \\ x_1 \\ x_2 \end{array}\begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.7 & 0.5 \\ 0.8 & 0.4 \end{bmatrix} \end{array} \quad \text{and} \quad \tilde{S}=\begin{array}{c} \\ y_1 \\ y_2 \end{array}\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.9 & 0.6 & 0.5 \\ 0.1 & 0.7 & 0.5 \end{bmatrix} \end{array}$$

Using max-min composition,

$$\left.\begin{aligned} \mu_{\tilde{T}}(x_1,z_1) &= \bigvee_{y\in Y}(\mu_{\tilde{R}}(x_1,y)\wedge\mu_{\tilde{S}}(y,z_1)) \\ &= \max[\min(0.7,0.9),\min(0.5,0.1)] \\ &= 0.7 \end{aligned}\right\} \tilde{T}=\begin{array}{c} \\ x_1 \\ x_2 \end{array}\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.8 & 0.6 & 0.4 \end{bmatrix} \end{array}$$

Two fuzzy relations R and S are defined on sets A, B and C. That is, $R \subseteq A \times B$, $S \subseteq B \times C$. The composition $S \bullet R = SR$ of two relations R and S is expressed by the relation from A to C:

For $(x, y) \in A \times B$, $(y, z) \in B \times C$,

$$\mu_{S\bullet R}(x, z) = \max_y [\mu_R(x, y) \bullet \mu_S(y, z)]$$

$$= \bigvee_y [\mu_R(x, y) \bullet \mu_S(y, z)]$$

$M_{S\bullet R} = M_R \bullet M_S$ (matrix notation)

(max-product composition)

Max-product example:

$X \equiv \{x_1, x_2\}$, $Y \equiv \{y_1, y_2\}$, and $Z = \{z_1, z_2, z_3\}$

Consider the following fuzzy relations:

$$\tilde{R} = \begin{array}{c} \\ x_1 \\ x_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.7 & 0.5 \\ 0.8 & 0.4 \end{bmatrix} \end{array} \quad \text{and} \quad \tilde{S} = \begin{array}{c} \\ y_1 \\ y_2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.9 & 0.6 & 0.5 \\ 0.1 & 0.7 & 0.5 \end{bmatrix} \end{array}$$

Using max-product composition,

$$\left.\begin{aligned} \mu_{\tilde{T}}(x_2, z_2) &= \bigvee_{y \in Y} (\mu_{\tilde{R}}(x_2, y) \bullet \mu_{\tilde{S}}(y, z_2)) \\ &= \max[(0.8, 0.6), (0.4, 0.7)] \\ &= 0.48 \end{aligned}\right\} \quad \tilde{T} \equiv \begin{array}{c} \\ x_1 \\ x_2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} .63 & .42 & .25 \\ .72 & .48 & .20 \end{bmatrix} \end{array}$$

CLASSICAL EQUIVALENCE RELATION

Let relation R on universe X be a relation from X to X. Relation R is an equivalence relation if the following three properties are satisfied.

1. Reflexivity
2. Symmetry
3. Transitivity

The function theoretic forms of representation of these properties are as follows:

1. Reflexivity:

$$\chi_R(x_i, x_i) = 1 \text{ or } (x_i, x_i) \in R$$

2. Symmetry:

$$\chi_R(x_i, x_j) = \chi_R(x_j, x_i)$$
$$\text{i.e., } (x_i, x_j) \in R \Rightarrow (x_j, x_i) \in R$$

3. Transitivity:

$$\chi_R(x_i, x_j) \text{ and } \chi_R(x_j, x_k) = 1, \text{ so } \chi_R(x_i, x_k) = 1$$
$$\text{i.e., } (x_i, x_j) \in R(x_j, x_k) \in R, \text{ so } (x_i, x_k) \in R$$

CLASSICAL TOLERANCE RELATION

A tolerance relation R_1 on universe X is one where the only the properties of reflexivity and symmetry are satisfied. The tolerance relation can also be called proximity relation. An equivalence relation can be formed from tolerance relation R_1 by $(n-1)$ compositions within itself, where n is the cardinality of the set that defines R_1, here it is X, i.e.

$$\underbrace{R_1^{n-1}}_{\text{Tolerance relation}} = R_1 \circ R_1 \circ \cdots \circ R_1 = \underbrace{R}_{\text{Equivalence relation}}$$

FUZZY EQUIVALENCE RELATION

Let $\underset{\sim}{R}$ be a fuzzy relation on universe X, which maps elements from X to X. Relation $\underset{\sim}{R}$ will be a fuzzy equivalence relation if all the three properties – reflexive, symmetry and transitivity – are satisfied. The membership function theoretic forms for these properties are represented as follows:

1. Reflexivity:

$$\mu_{\underset{\sim}{R}}(x_i, x_i) = 1 \quad \forall x \in X$$

If this is not the case for few $x \in X$, then $R(X, X)$ is said to be irreflexive.

2. Symmetry:

$$\mu_{\underset{\sim}{R}}(x_i, x_j) = \mu_{\underset{\sim}{R}}(x_j, x_i) \text{ for all } x_i, x_j \in X$$

If this is not satisfied for few $x_i, x_j \in X$, then $R(X, X)$ is called asymmetric.

3. Transitivity:

$$\mu_{\underset{\sim}{R}}(x_i, x_j) = \lambda_1 \text{ and } \mu_{\underset{\sim}{R}}(x_j, x_k) = \lambda_2$$

$$\Rightarrow \ \mu_{\underset{\sim}{R}}(x_i, x_k) = \lambda$$

where

$$\lambda = \min\,[\lambda_1, \lambda_2]$$

$$\text{i.e., } \mu_{\underset{\sim}{R}}(x_i, x_j) \geq \max_{x_j \in X} \min[\mu_{\underset{\sim}{R}}(x_i, x_j), \mu_{\underset{\sim}{R}}(x_j, x_k)] \qquad \forall (x_i, x_k) \in X^2$$

FUZZY TOLERANCE RELATION

A binary fuzzy relation that possesses the properties of reflexivity and symmetry is called fuzzy tolerance relation or resemblance relation.

The equivalence relations are a special case of the tolerance relation. The fuzzy tolerance relation can be reformed into fuzzy equivalence relation in the s ... on is reformed into crisp equiva

$$\underbrace{\underset{\sim}{R}_1^{n-1}}_{\substack{\text{Fuzzy}\\ \text{tolerance}\\ \text{relation}}} = \underset{\sim}{R}_1 \circ \underset{\sim}{R}_1 \circ \cdots \circ \underset{\sim}{R}_1 = \underbrace{\underset{\sim}{R}}_{\substack{\text{Fuzzy}\\ \text{equivalence}\\ \text{relation}}}$$

where ‘n’ is the cardinality of the set that defines R1.

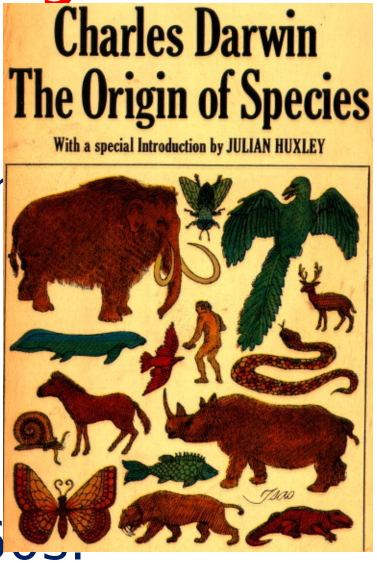
GENETIC ALGORITHM

General Introduction to GAs

- Genetic algorithms (GAs) are a technique to solve problems which need optimization.

- GAs are a subclass of **Evolutionary Computing** and are random search algorithms.

- GAs are based on Darwin's theory of evolution.



History of GAs:

- Evolutionary computing evolved in the 1960s.
- GAs were created by John Holland in the mid-1970s.