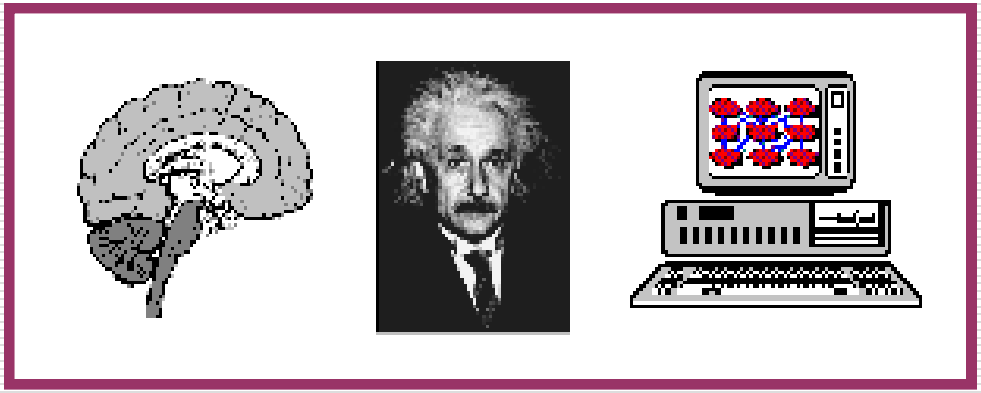
Chapter 15

Neural Networks and the Soft Computing Paradigm

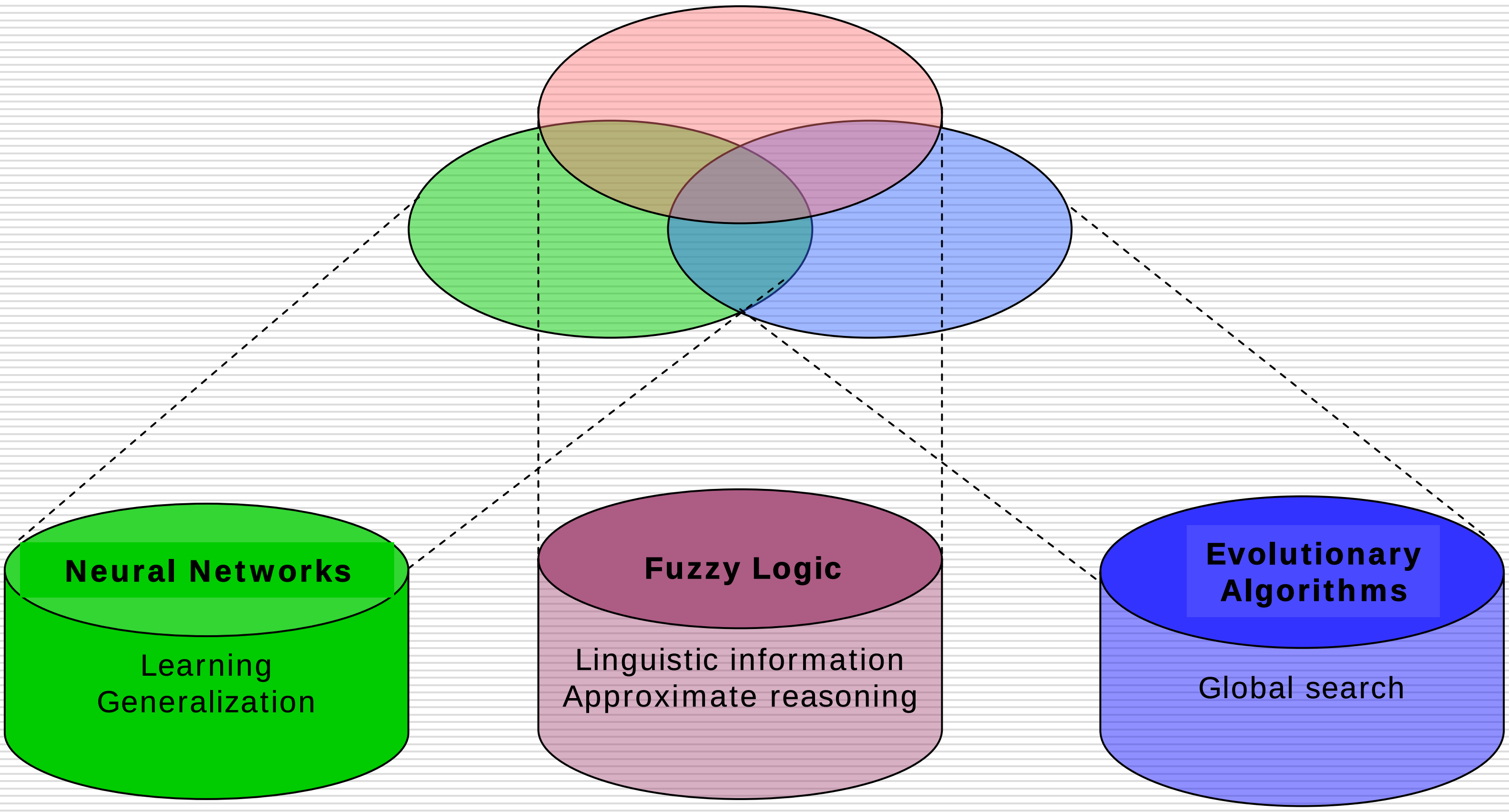
Neural Networks: A Classroom Approach
Satish Kumar
Department of Physics & Computer Science
Dayalbagh Educational Institute (Deemed University)

The Quest for Computational Intelligence?



- *Soft computing*, a term coined by Zadeh is basically a synergistic integration of three computing paradigms:
  - neural networks
  - fuzzy logic
  - probabilistic reasoning (which subsumes belief networks genetic algorithms and chaotic systems)
- Bezdek called the synergism of fuzzy logic, neural networks and genetic algorithms as *computational intelligence*.

Soft Computing/ Computational Intelligence



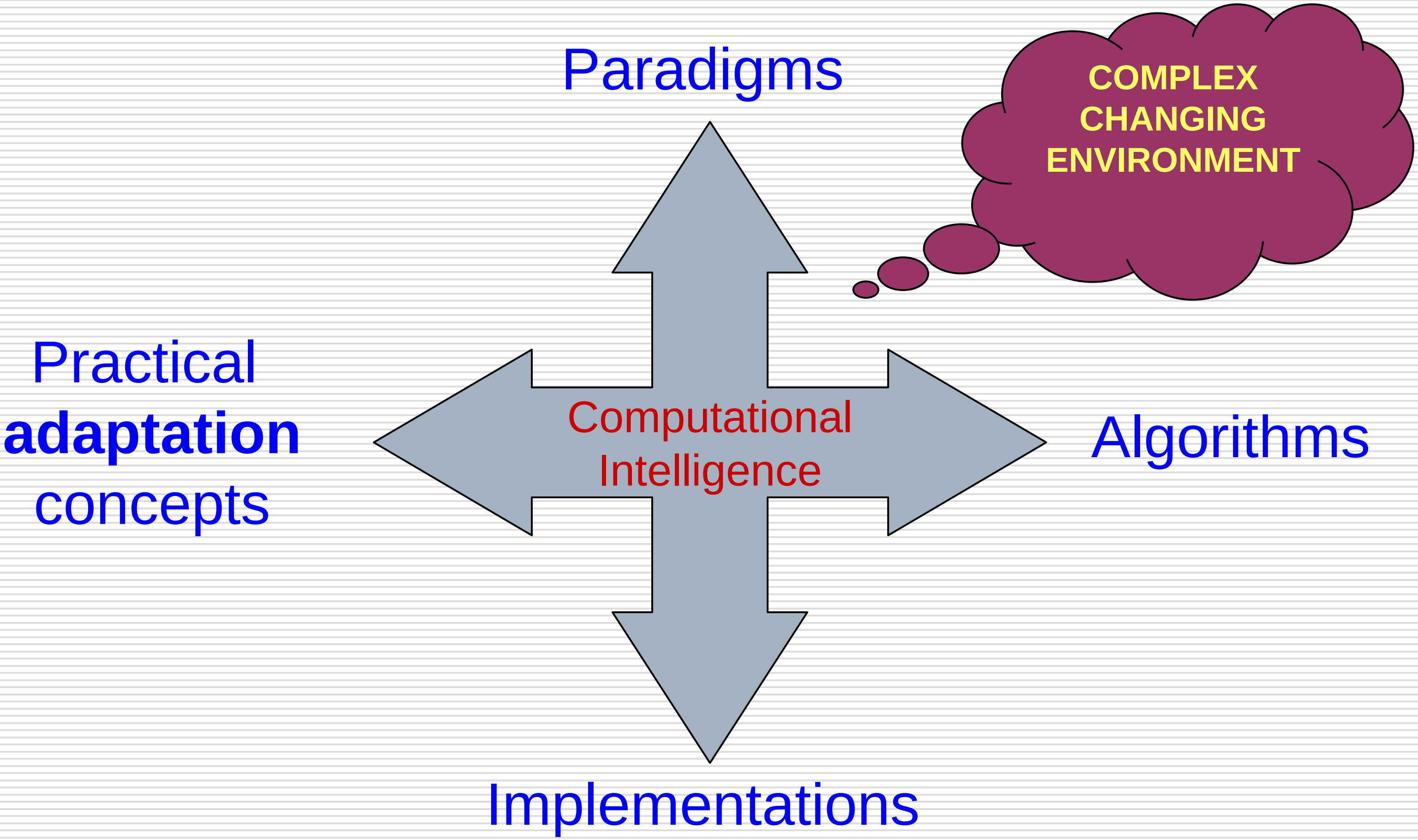
Machine Intelligence Quotient

- Provides a framework for flexible information processing applications designed to operate in the real world.
- Soft computing technologies
  - are robust by design
  - operate by trading off precision for tractability.
  - can handle uncertainty with ease
  - conform better with real world situations
  - provide lower solution costs.
- Primary focus and major contribution is to increase the *machine intelligence quotient* (MIQ)

Computational Intelligence (CI)

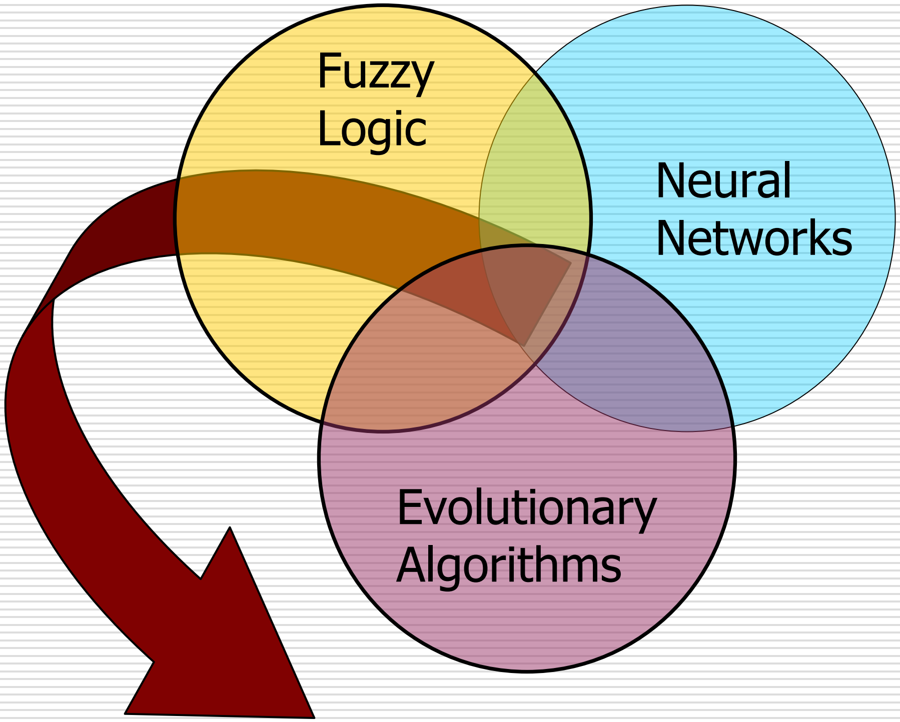
- Involves computing that exhibits the ability to learn and/or to deal with new situations
- Applications where the system is perceived to possess one or more attributes of *reason*, such as
  - generalization
  - discovery,
  - association and
  - abstraction.

Computational Intelligence



Silicon-based Computational Intelligence Systems

- Usually comprise hybrids of paradigms such as:
  - artificial neural networks
  - fuzzy systems
  - evolutionary algorithms
- Augmented with expert knowledge elements
- Often designed to mimic one or more aspects of carbon-based biological intelligence.



Neuro-Fuzzy-Evolutionary Systems

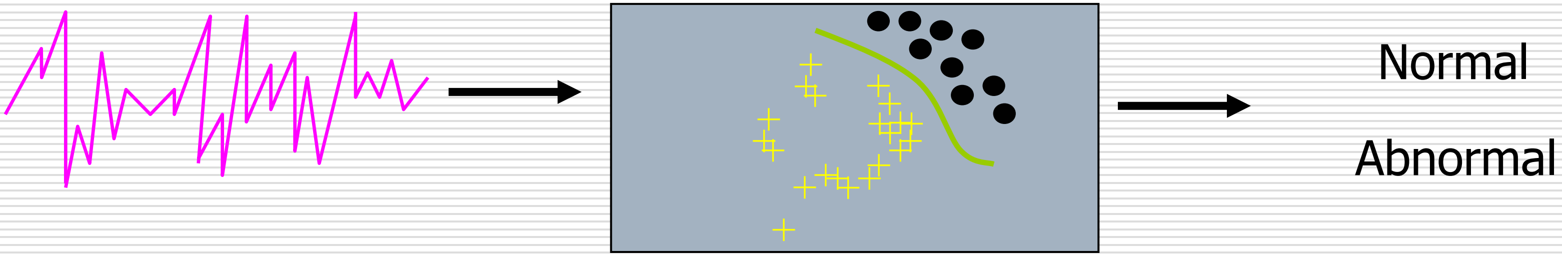
Good candidates for CI

- Fuzzy, imprecise or imperfect data
- No available mathematical algorithm
- Optimal solution unknown
- Rapid prototyping required
- Only domain experts available
- Robust system required

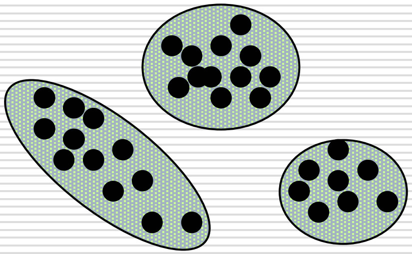
Hints

- Acquire basic knowledge before experimenting
- Pay special attention to data representation and preprocessing
- Components such as ANNs can be on “front end”, “back end”, or in middle
- Combinations of concepts, paradigms, and architectures are feasible, but can be difficult to implement successfully

Application Domains (1)

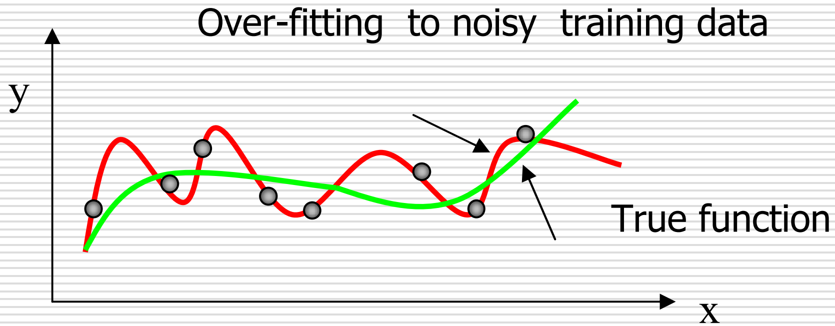


(1) CLASSIFIER DESIGN

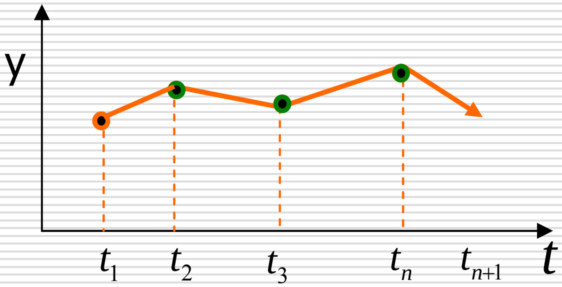


(2) CLUSTERING

Application Domains (2)



(3) FUNCTION APPROXIMATION

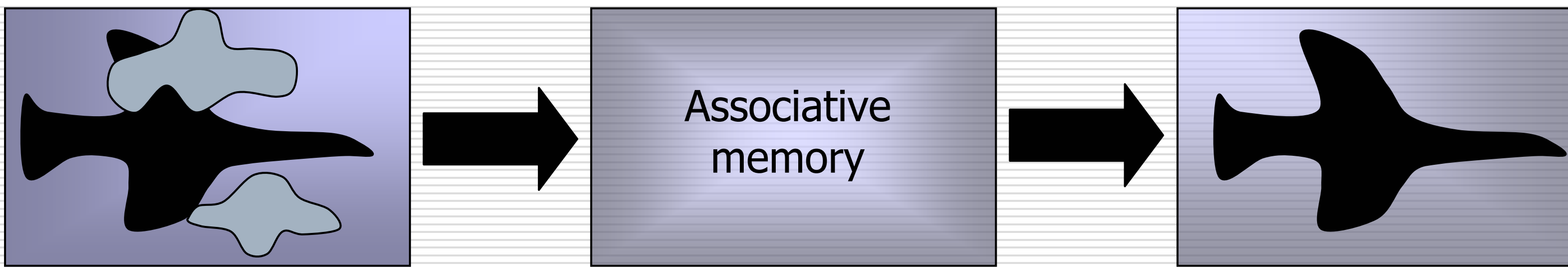


(4) FORECASTING

Application Domains (3)

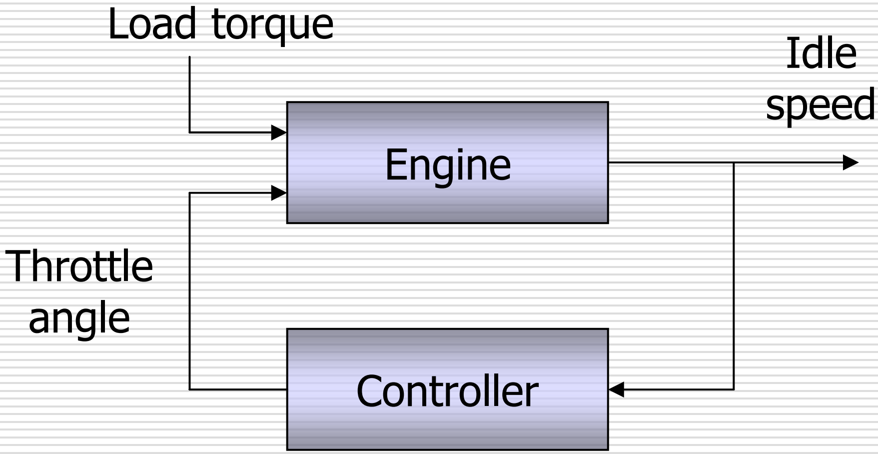
Airplane image partially occluded by clouds

Retrieved airplane image

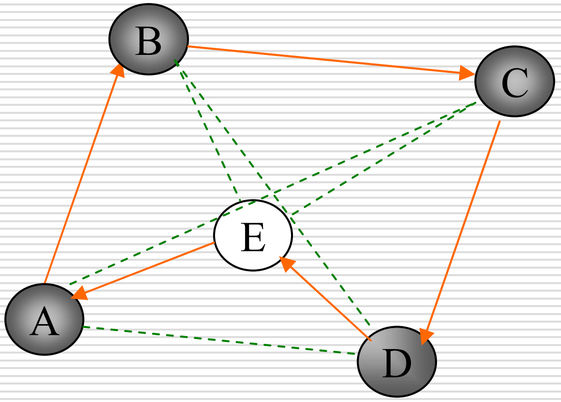


(5) CONTENT ADDRESSABLE MEMORY

Application Domains (4)



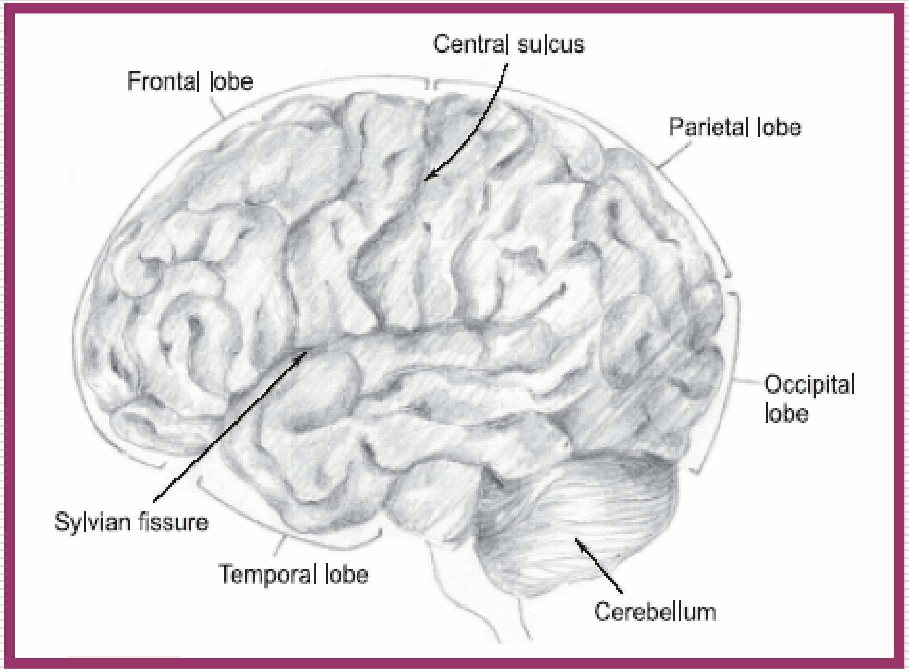
(6) CONTROL



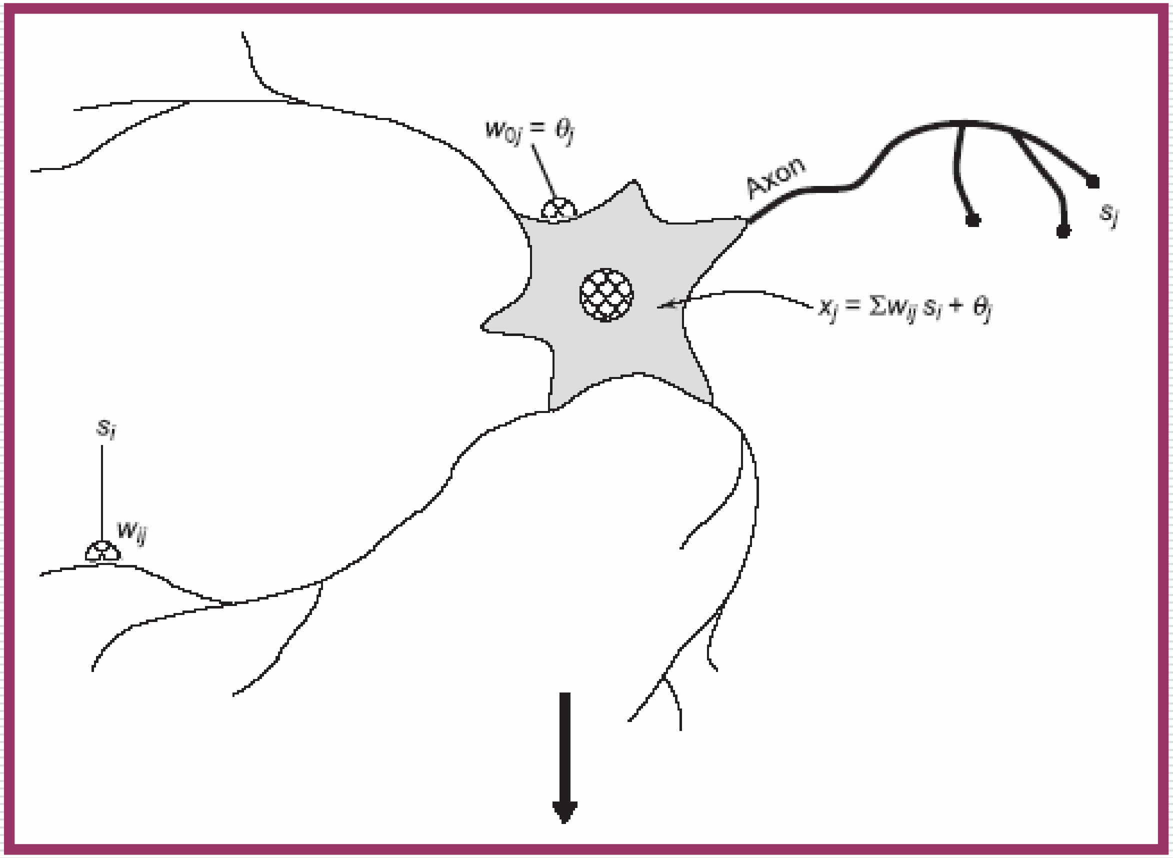
(7) OPTIMIZATION

Inspiration for Computational Intelligence

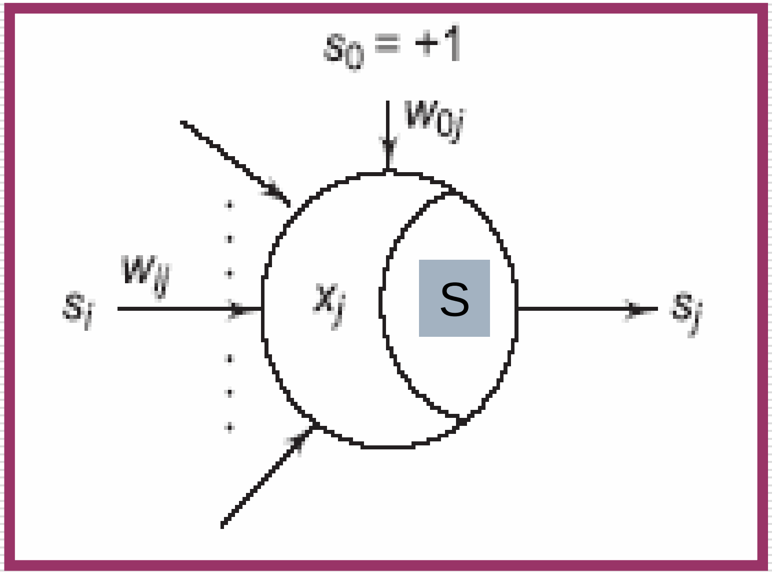
- CI has been inspired by two fundamental questions:
  - How does the human brain work ?
  - How can we exploit the brain metaphor to build intelligent machines ?



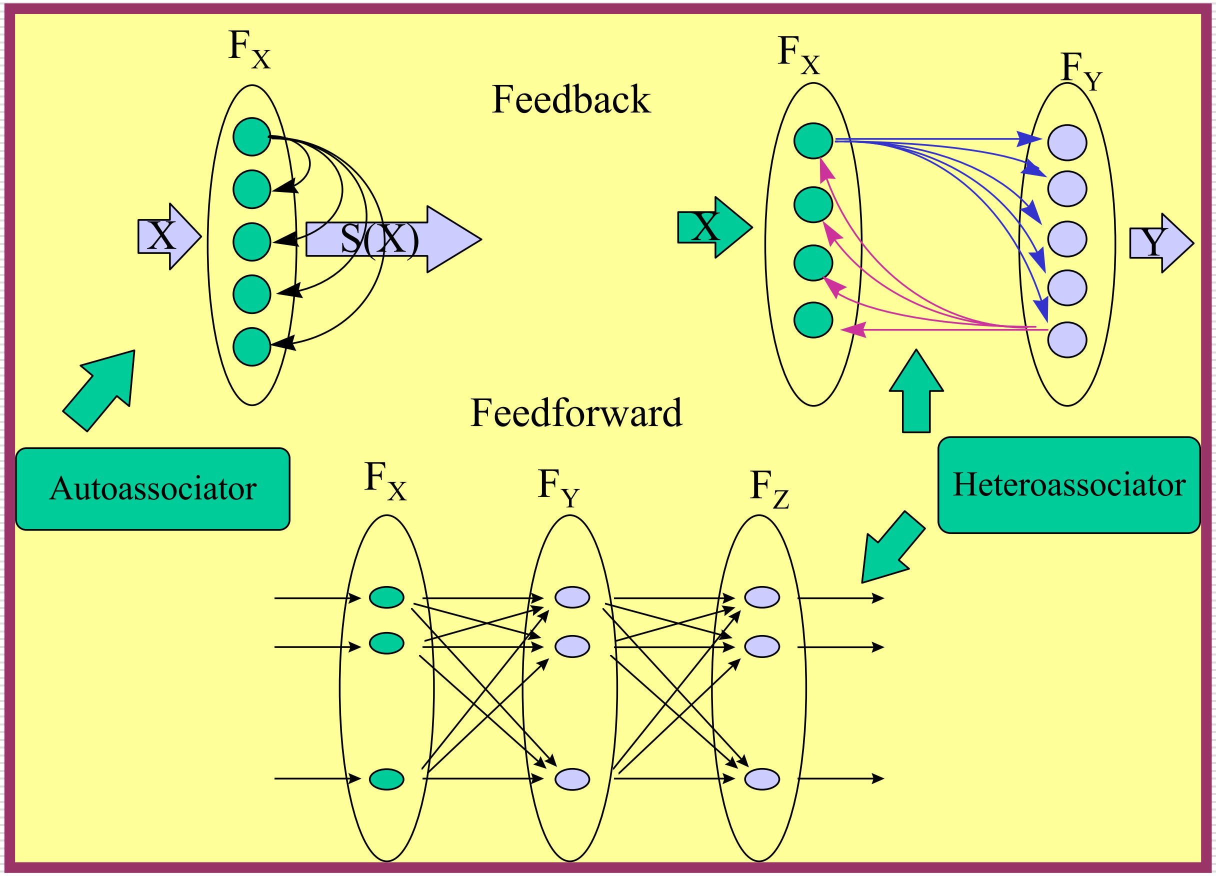
Biological Neuron: Computing Device



Artificial Neuron

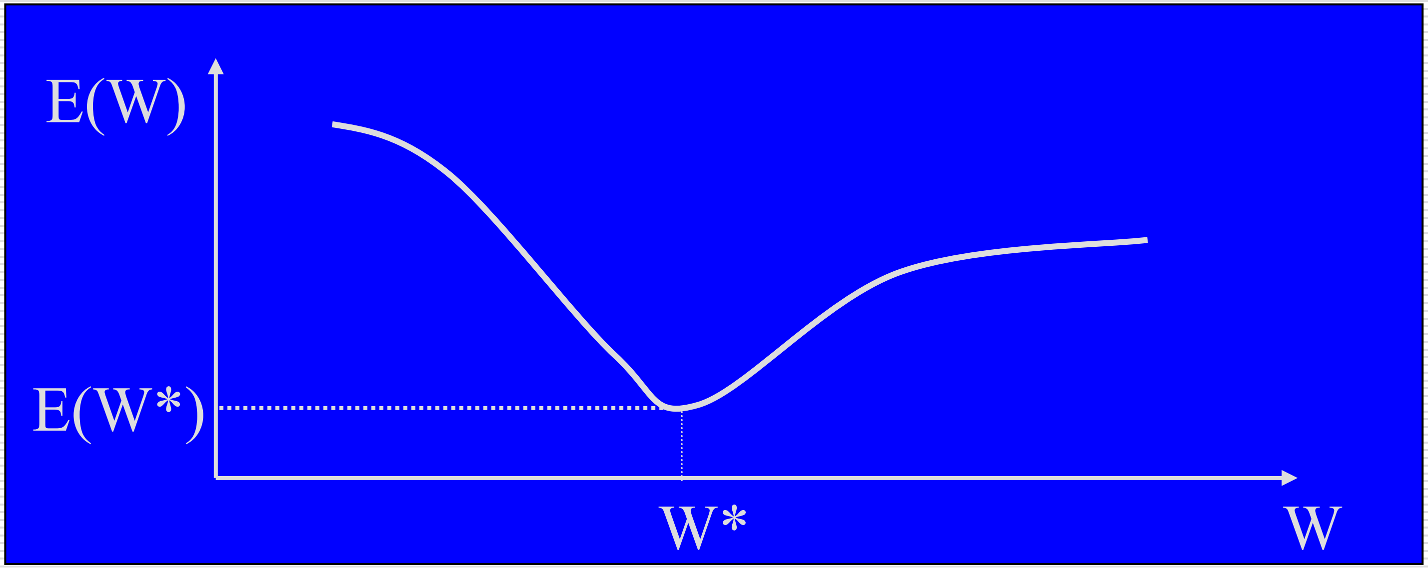


Architectures



Power of NNs: Learning

- Training set: $T = \left\{ \left(x^q, d^q \right) \quad q = 1, 2, \ldots, m \right\}$
- Error measure: $\xi(W) = f\left(o_i^q - d_i^q \right)$



Fuzziness in the Real World

- How is one to represent notions like:
  - *large* profit
  - *high* pressure
  - *tall* man
  - *wealthy* woman
  - *moderate* temperature.
- And statements like:
  - Most experts believe that the likelihood of a severe earthquake in the near future is very low.
  - Usually it takes about an hour to drive from Berkeley to Stanford in light traffic.

Founder of Fuzziness

Lotfi Zadeh, Director, Berkeley Initiative in Soft Computing (BISC)
University of California Berkeley, CA 94720 -1776
http://www.cs.berkeley.edu/People/Faculty/Homepages/zadeh.html

In 1965, Lotfi Zadeh introduced the *theory of fuzzy sets*: A fuzzy set is a collection of objects that might belong to the set to a degree, varying from 1 for full belongingness to 0 for full non-belongingness, through all intermediate values.

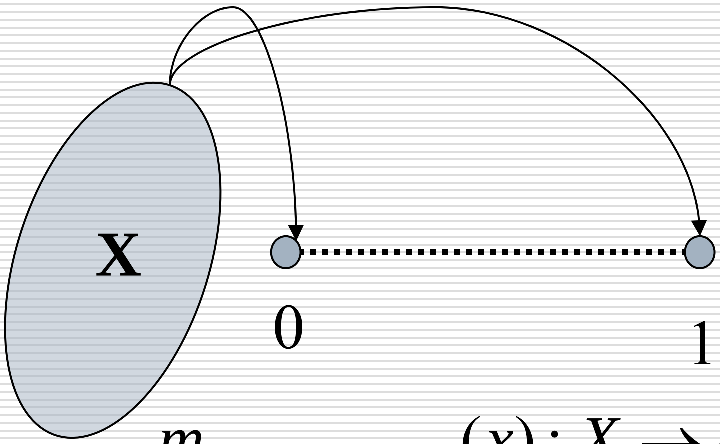
Fuzzy Sets are Functions

Mathematically we have a *membership function* :

$$\mu_{MIDDLE-AGED}(x) : X \rightarrow [0,1]$$

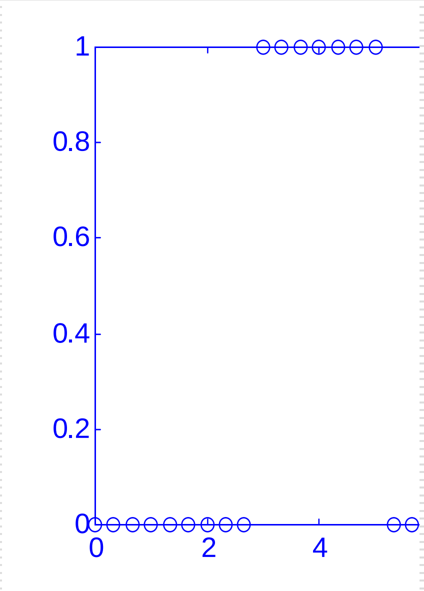
where X is the universe of discourse (UOD)

Example: Classical Set

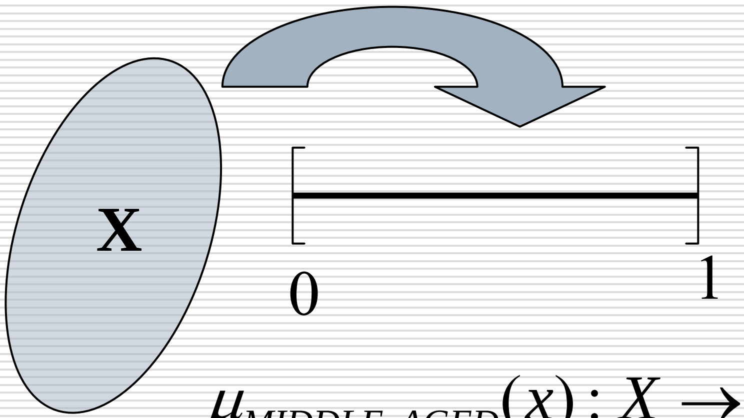


$$m_{MIDDLE-AGED}(x) : X \rightarrow \{0,1\}$$

Example: Numbers, z, 3 ≤ z ≤5

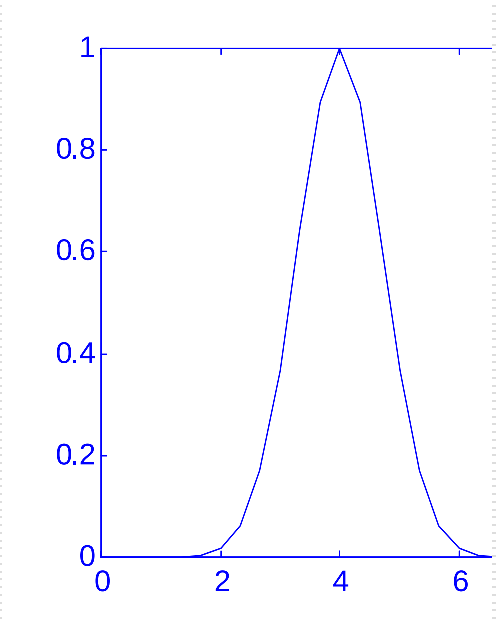


Example: Fuzzy Set

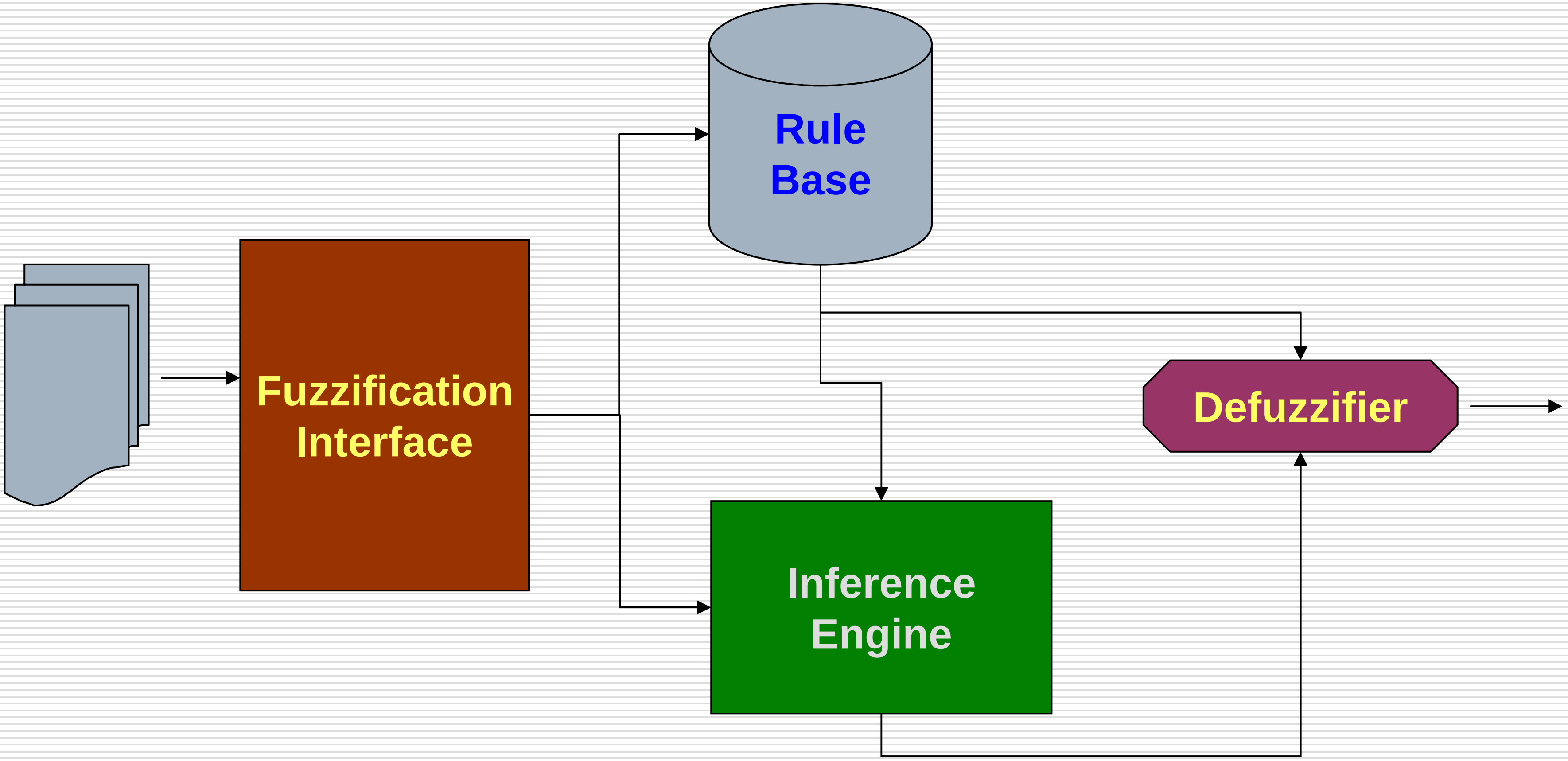


$$\mu_{MIDDLE-AGED}(x) : X \rightarrow [0,1]$$

Example: Numbers ***close to 4***



Fuzzy Systems



Fuzzy-Neural Integration

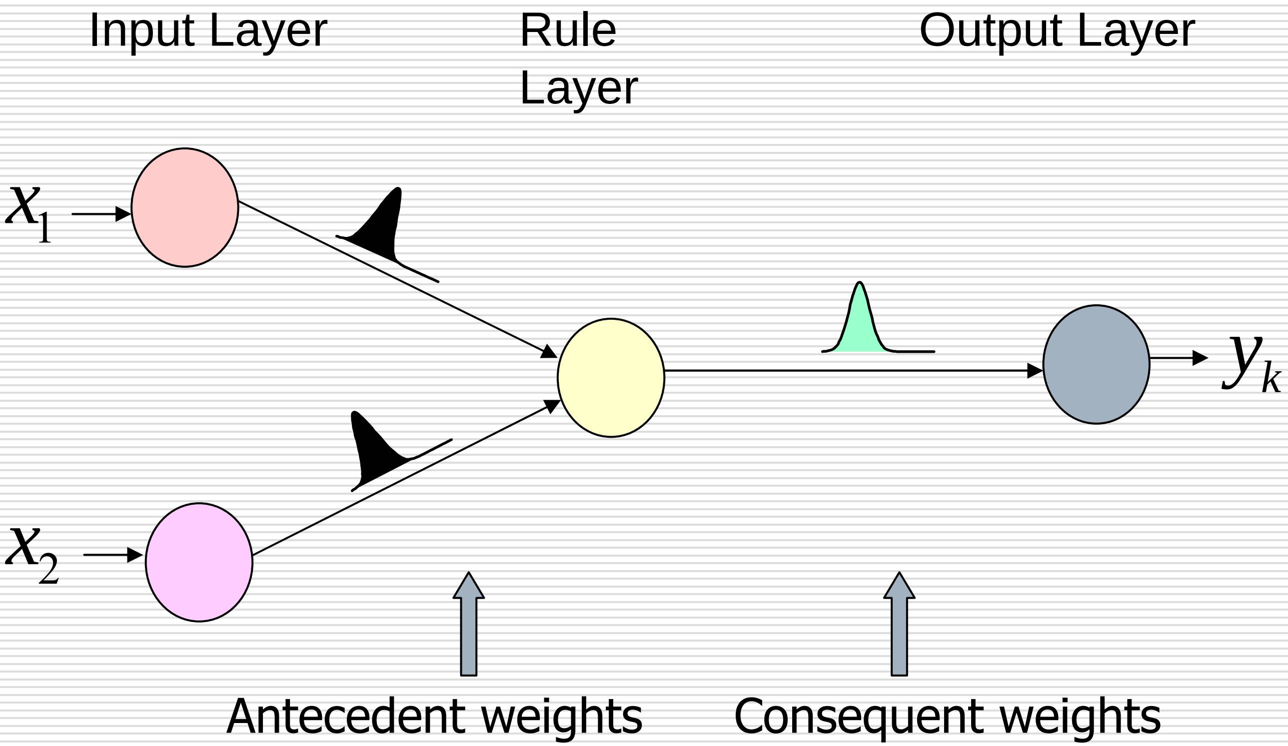
- FL provides
  - a high level framework for approximate reasoning
  - can appropriately handle both the uncertainty and imprecision in linguistic semantics
  - help model expert heuristics
  - provide requisite high level organising principles.

Fuzzy-Neural Integration

- **NN's provide**
  - self-organising substrate architectures
  - low level representation of information
  - on-line adaptation capabilities.
- It is useful to combine these approaches in the development of intelligent systems.
- Such cohesive systems are referred to as *fuzzy-neural systems* or *fuzzy neural networks*.

A Simple Fuzzy Neural Network

If x_1 is MEDIUM and x_2 is LOW then y is HIGH



Evolutionary Computation

- Genetic algorithms
- Evolutionary strategies
- Genetic programming
- Differential evolution

Origins of EAs: Two Books

- 1966, Fogel, Owens, and Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley & Sons
  - Looked at derivation of
    - finite-state machines
    - controllers
    - data reduction

    through successive mutations

- 1975, John H. Holland, *Adaptation in natural and artificial systems*, MIT Press
  - Focus was on natural systems, simulation
  - Introduced current genetic algorithm idea
  - Mostly theory, some applications to:
    - game-playing
    - search programs

Evolutionary Algorithms

- Provide a straightforward approach to solve difficult optimization problems
- Heuristic in nature
- No guarantee to find an optimal solution
- However, can find good "near optimal" solutions fast
- Applications target optimization problems in almost any discipline: engineering, science, finance…

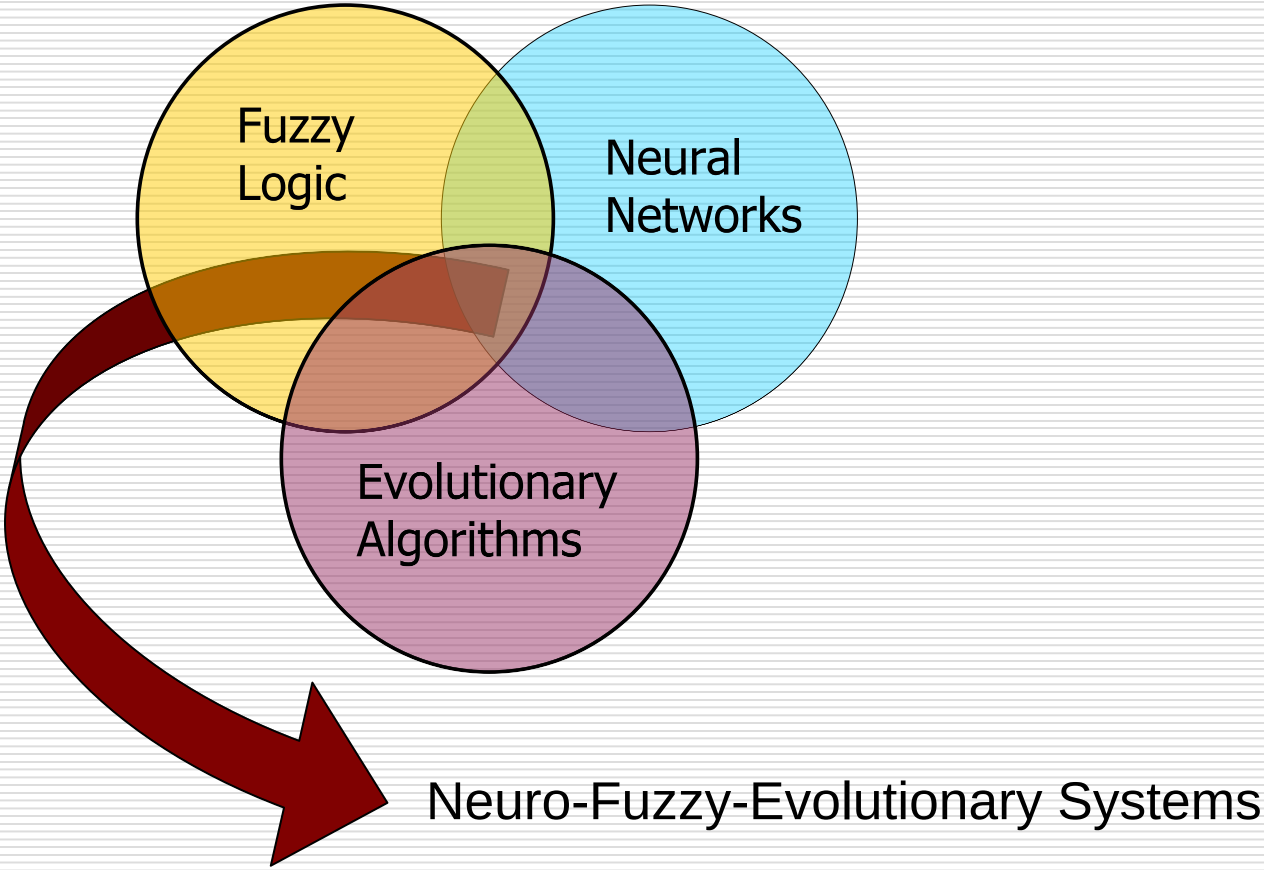
EAs Derive Inspiration from Natural Selection

- Represent the problem as a string of binary or real numbers
- Start with a population of such random solutions
- Focus on the entire population rather than a single individual
- Individuals that are fit enough to survive will reproduce
- Create new individuals from existing ones
  - Crossover
  - Mutation

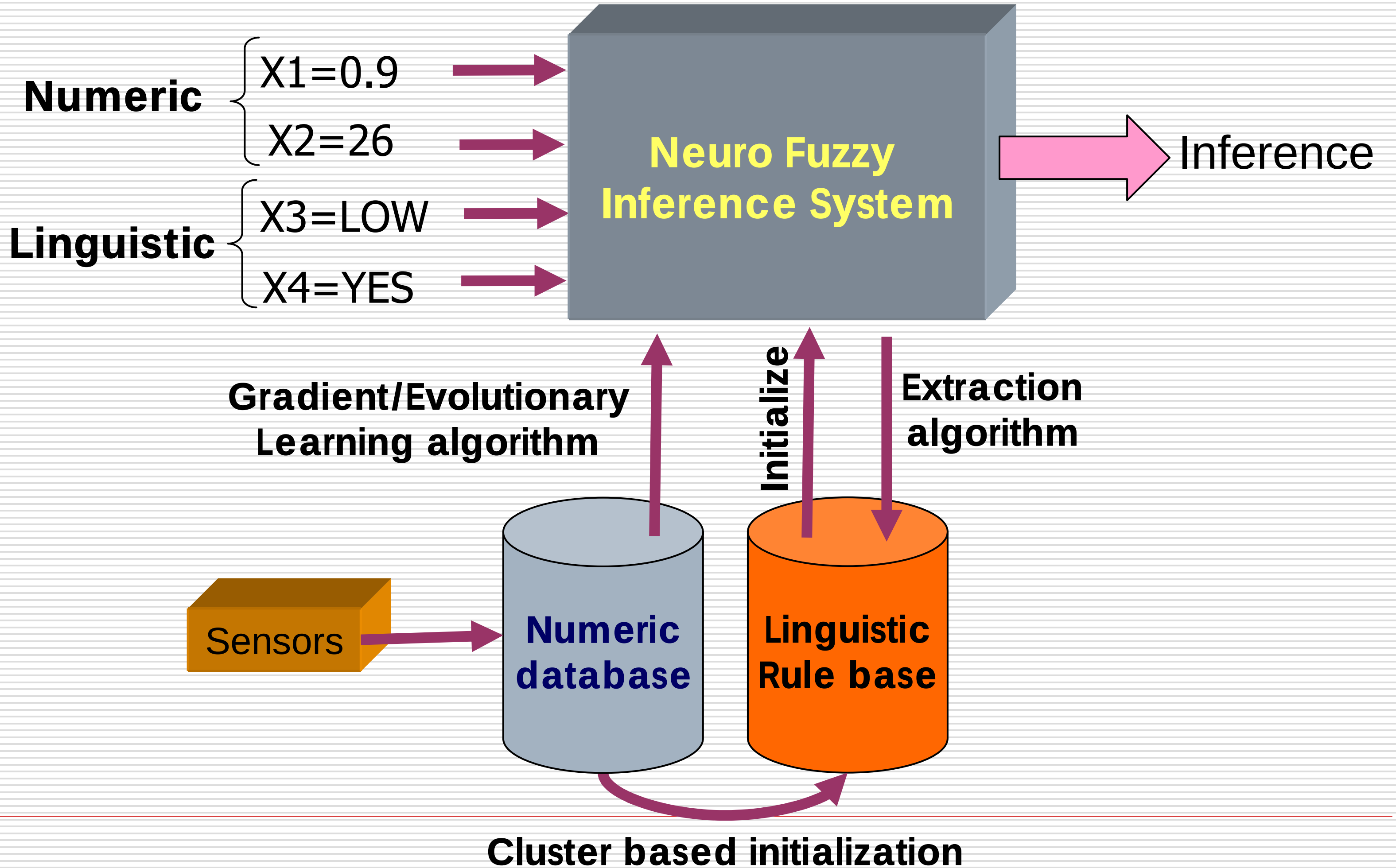
Applications of EAs to NN and FS

- Application to neural networks:
  - Evolve the weights of a neural network
  - Evolve the architecture: how many hidden nodes are enough?
- Application to fuzzy systems
  - Search membership functions automatically
  - Find optimal structures of neuro-fuzzy systems

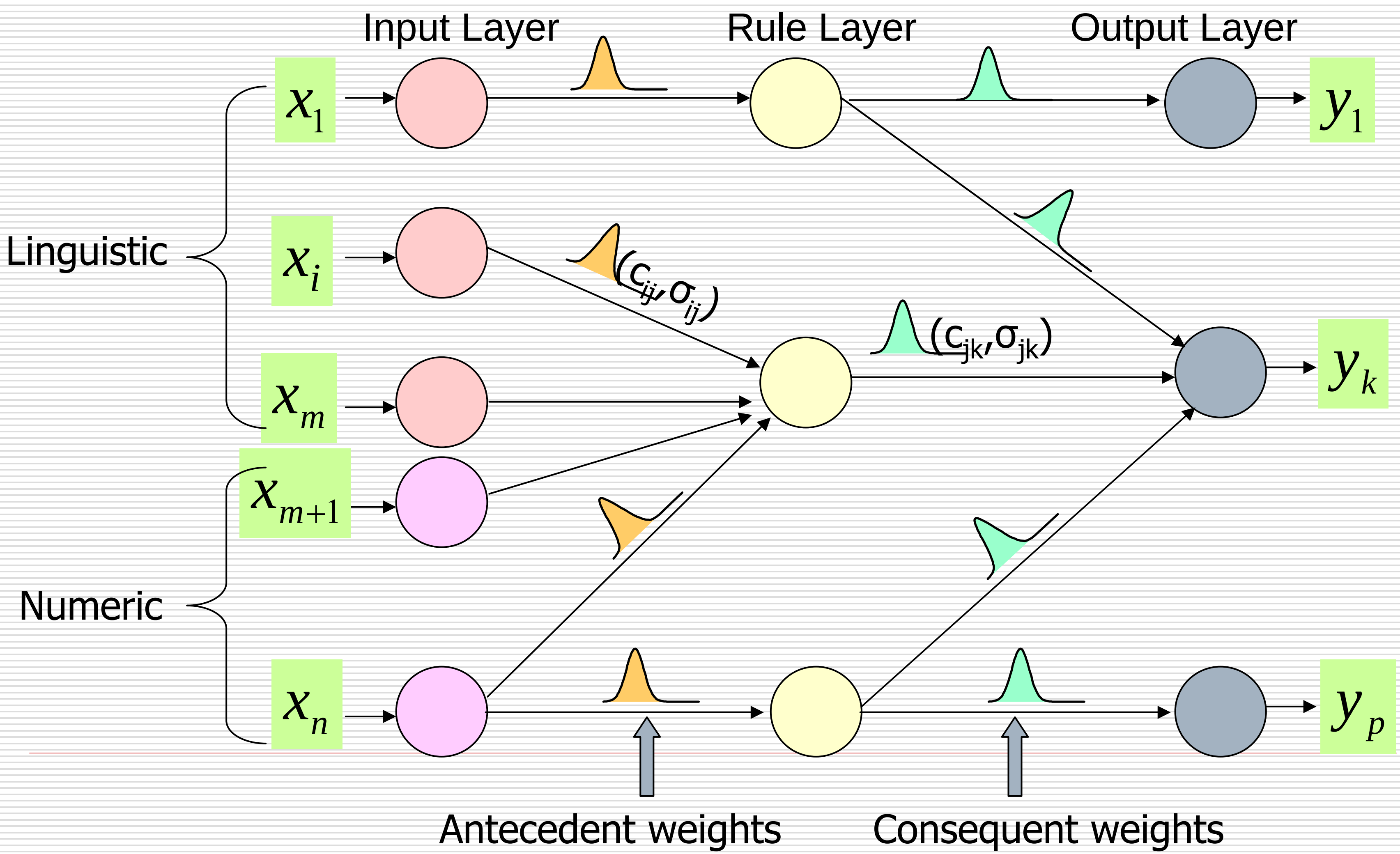
Computational Intelligence Systems



A Neuro Fuzzy System

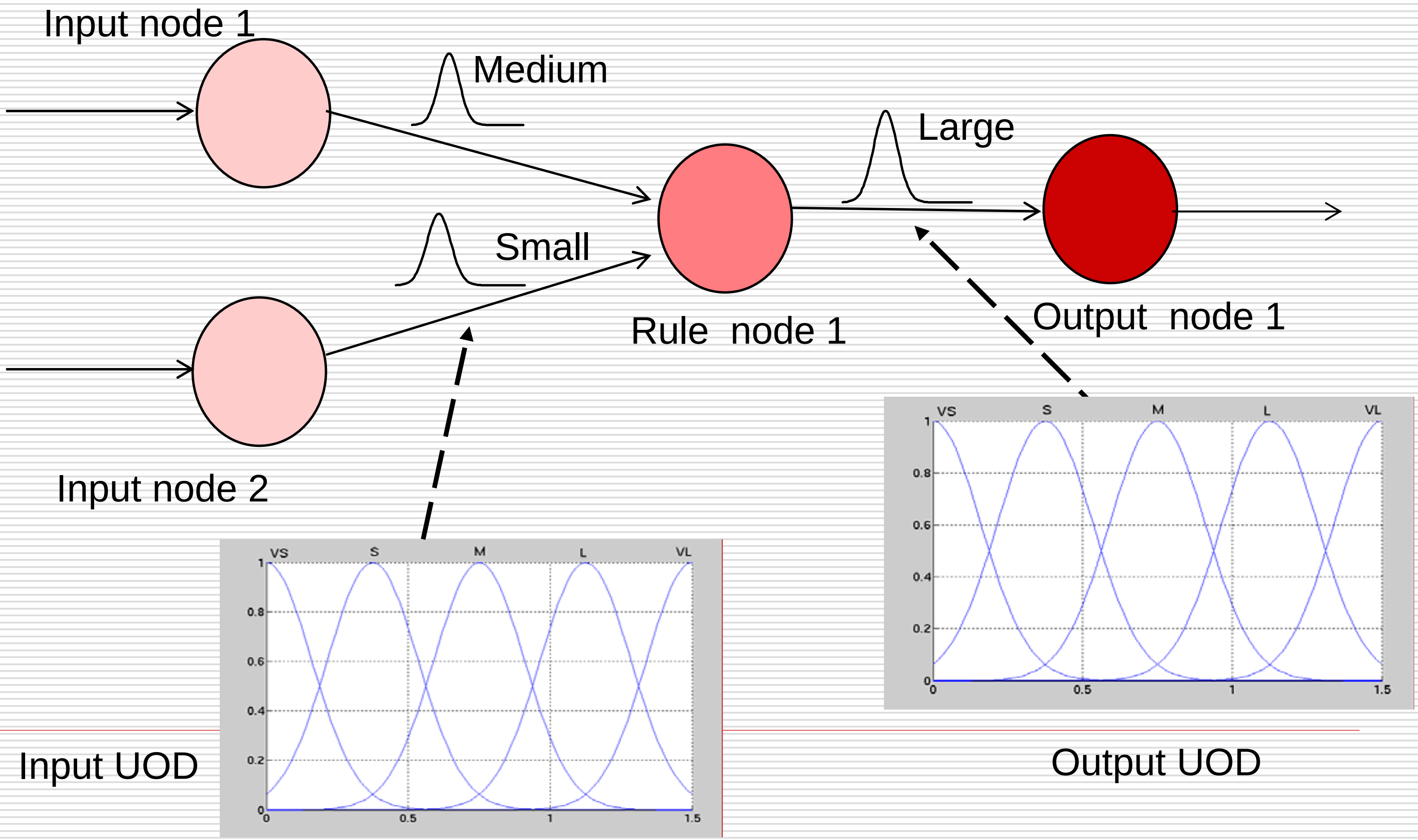


Subsethood Product Fuzzy Neural Inference System (SuPFuNIS)



Issue 1:
Embedding Rule Base Knowledge

R= If x1 is MEDIUM <u>and</u> x2 is SMALL <u>then</u> y1 is LARGE

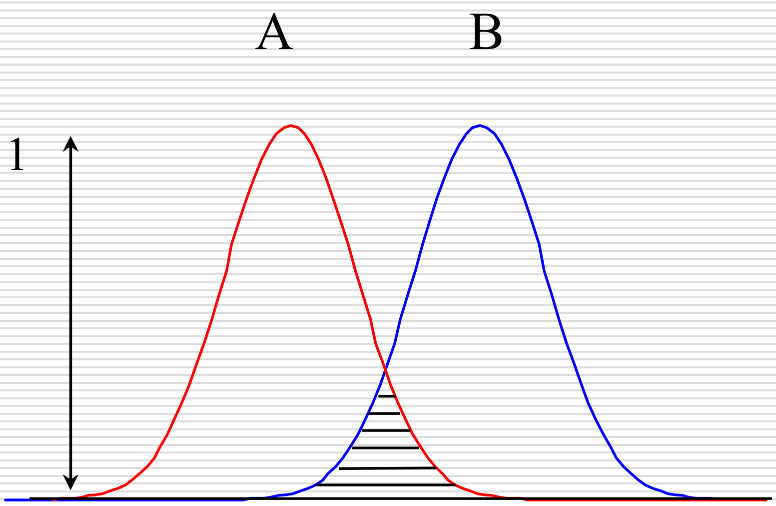


Entropy

$$E(A,B) = \frac{C(A \cap B)}{C(A \cup B)}$$

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

$$E(A,B) = \frac{C(A \cap B)}{C(A) + C(B) - C(A \cap B)}$$

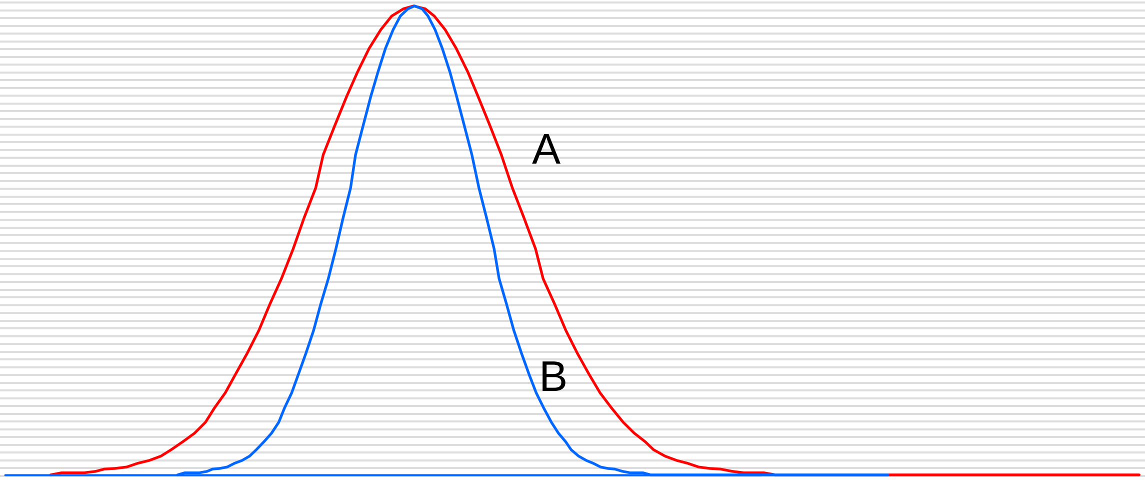


Mutual Subsethood

The mutual subsethood measures the degree to which fuzzy set A equals fuzzy set B.

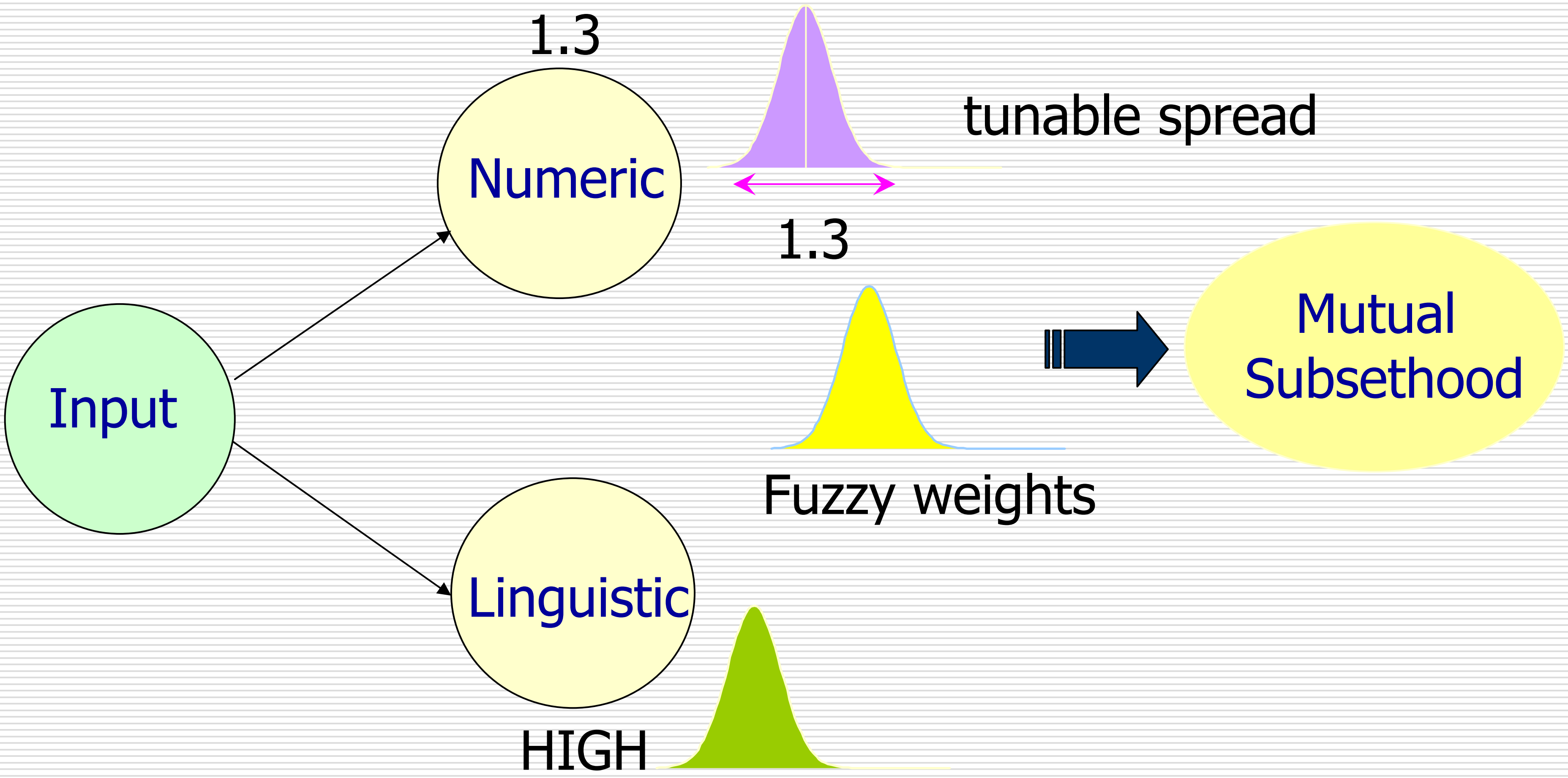
$$E(A,B)=\frac{C(A\cap B)}{C(A\cup B)}$$

$$S(A,B)=\frac{C(A\cap B)}{C(A)}$$

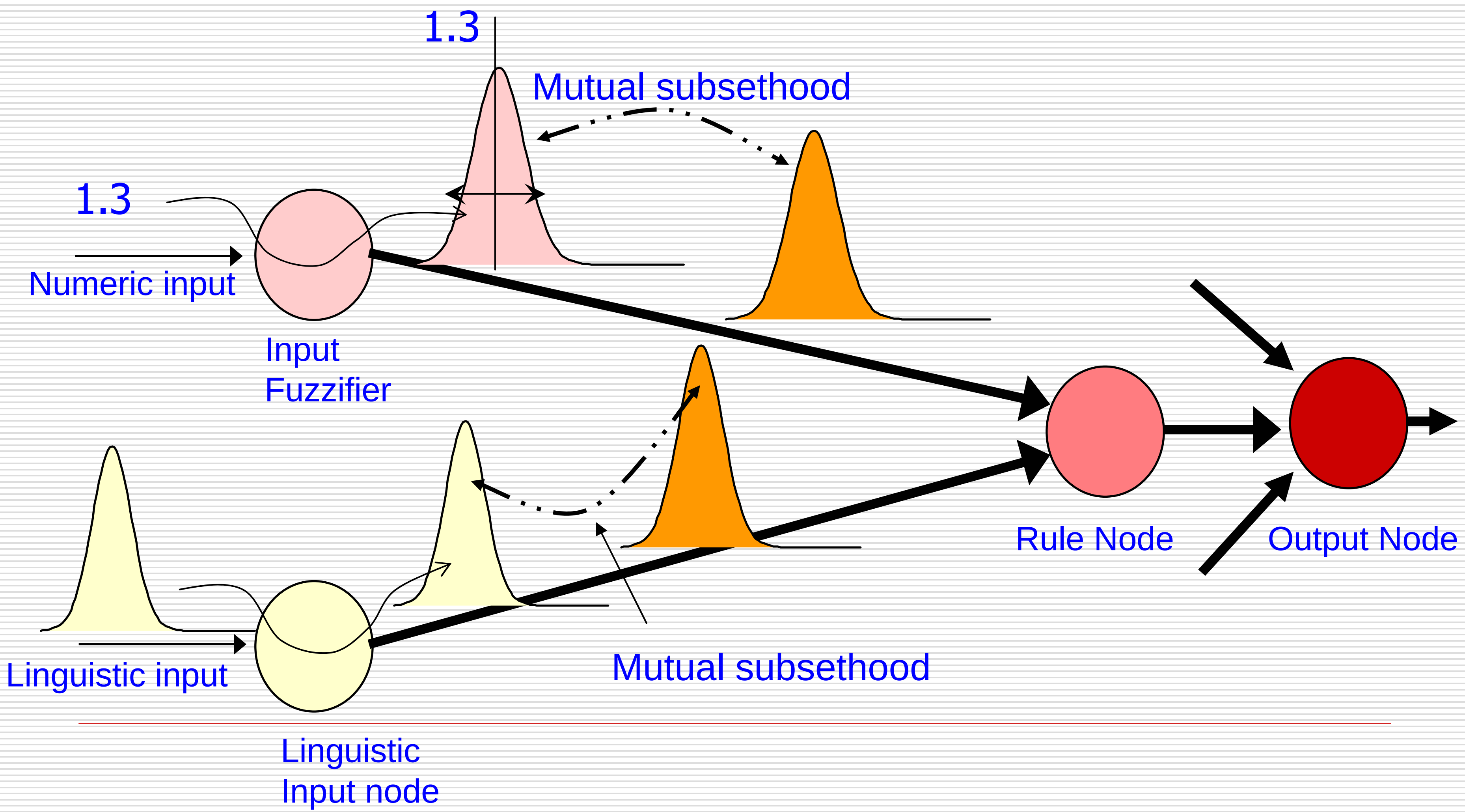


Subsethood measures the degree to which set A belongs to or is a subset of set B

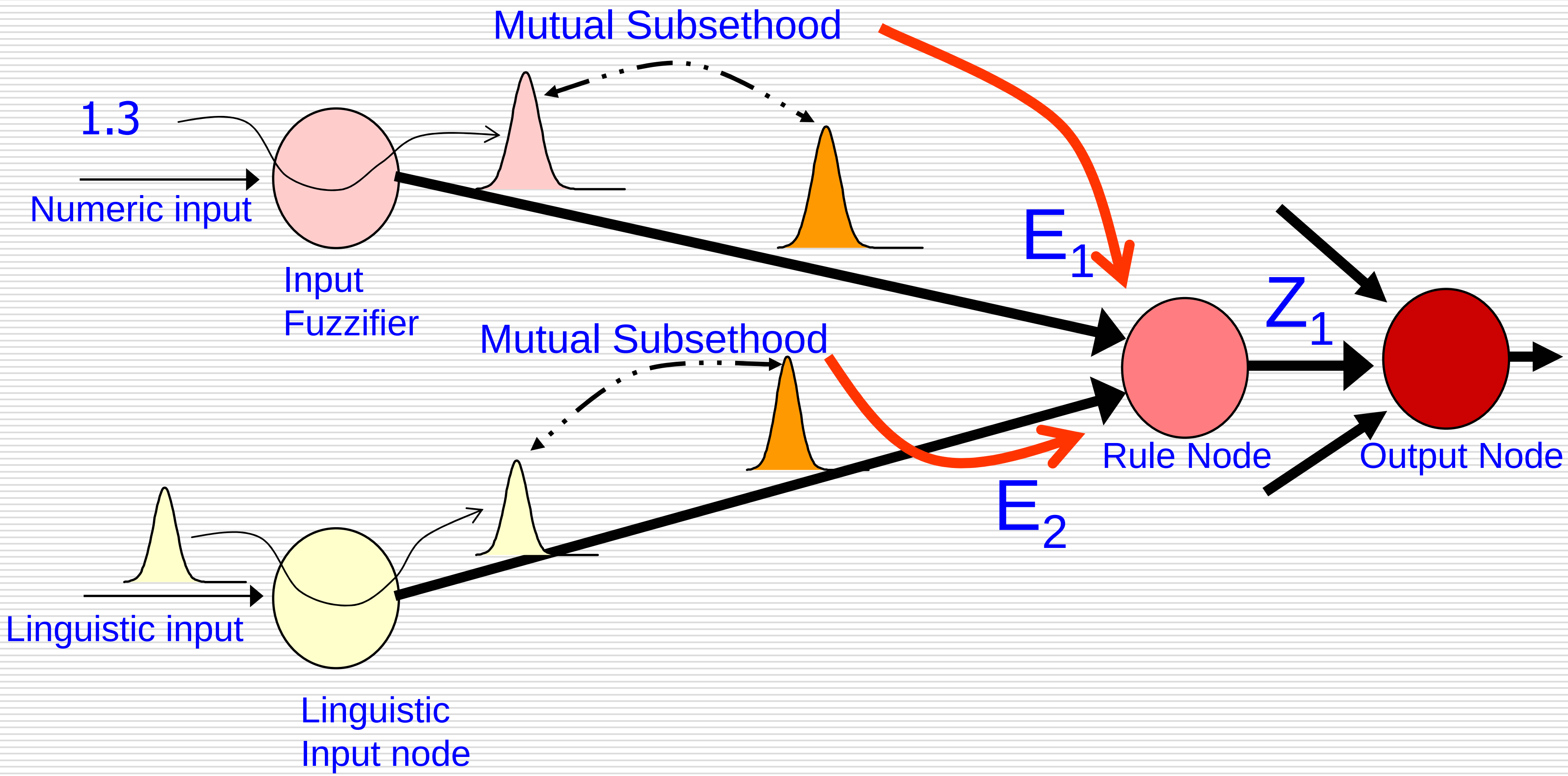
Fuzzification of numeric inputs



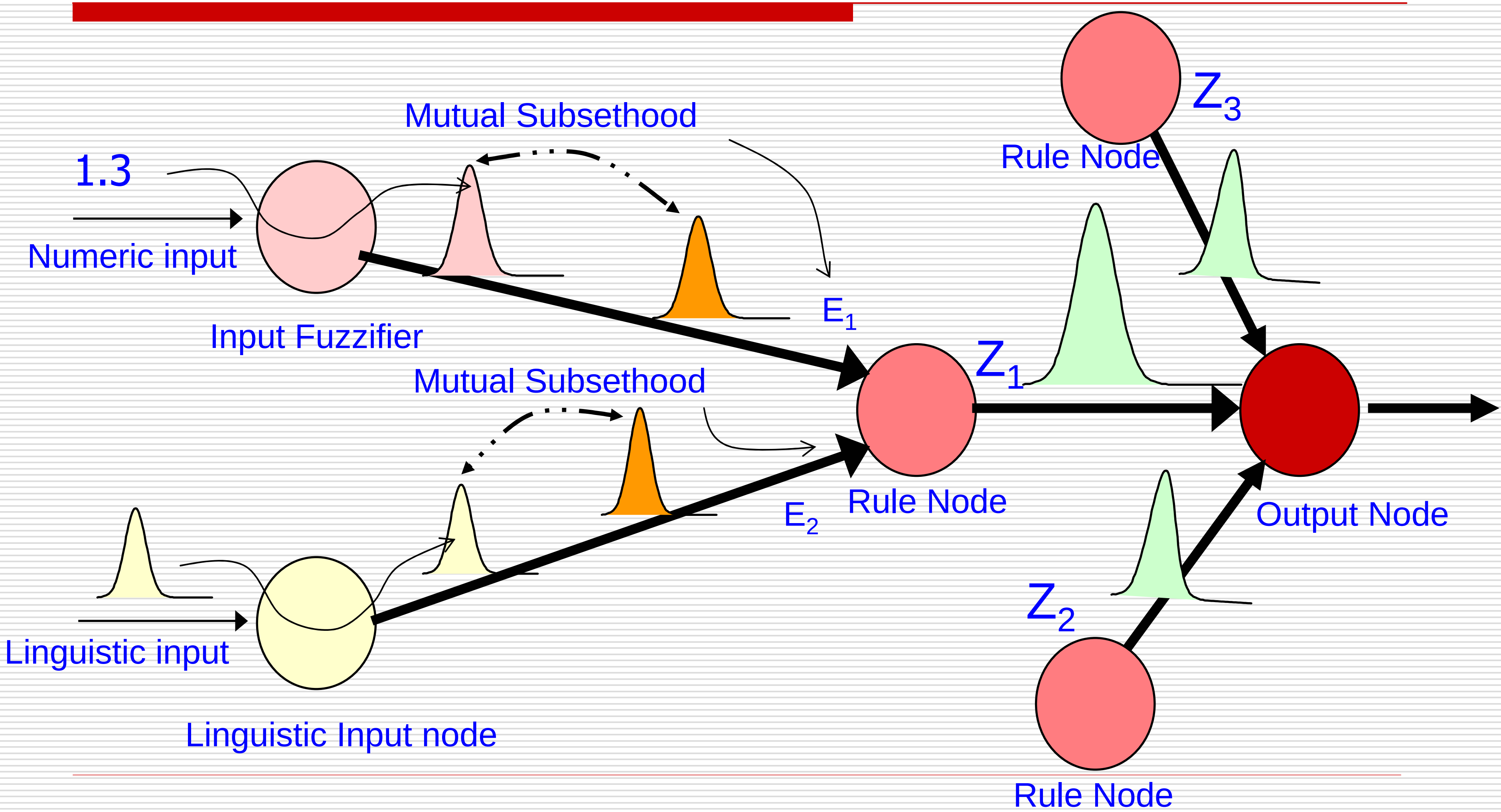
Composition of mixed inputs with fuzzy weights



Activity Aggregation



Defuzzification



Activity Calculation at Output Layer

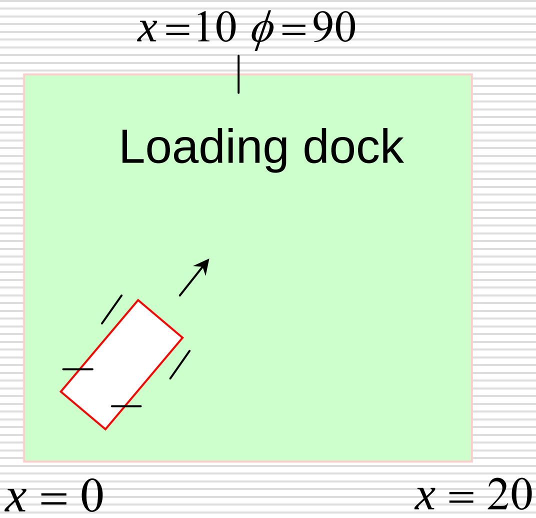
Defuzzification is done by each node in this layer using the volume based defuzzification (modified COG)

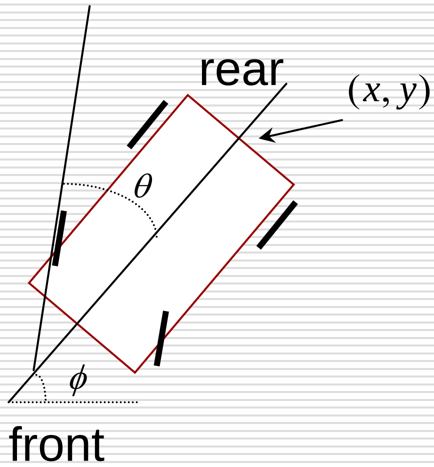
$$y_k = \frac{\sum_{j=1}^{q} z_j \, v_{jk}^{c} \, v_{jk}^{\sigma}}{\sum_{j=1}^{q} z_j \, v_{jk}^{\sigma}}$$

The output signal of node *k* is $S(y_k)=y_k$

Truck Backer Upper Control Problem

- Backing up a truck to loading dock





Φ : Angle of the truck with the horizontal

x, y : coordinates in the space

Θ : steering angle

b : length of truck

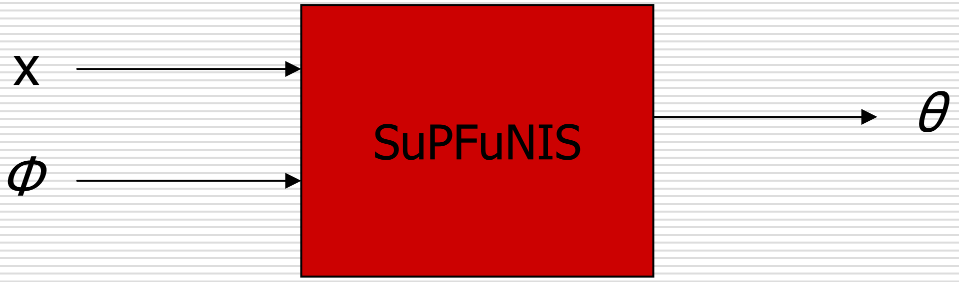
$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)]$$

$$y(t+1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)]\cos[\phi(t)]$$

$$\phi(t+1) = \phi(t) - \sin^{-1}[\tfrac{2\sin[\theta(t)]}{b}]$$

Design of Truck Backer Upper Control

- Enough clearance assumed between the truck and the loading dock such that the coordinate y can be ignored.
- Range of x : 0 to 20
- Range of Φ : -90° to 270°
- Range of θ : -40° to +40°



Control value of θ such that the final state $(x_f, \Phi_f) = (10, 90^\circ)$

Data Set Details

- Numeric data comprises of 238 pairs accumulated from 14 sequences of desired (x,Φ,θ) values (Wang/Mendel, 1992).
- The 14 initial states (x,Φ,θ) are (1,0), (1,90), (1,270), (7,0), (7,90), (7,180), (7,270), (13,0), (13,90), (13,180),(13,270), (19,90), (19,180), (19,270).
- The data was linearly normalized in the range [0,1] and used to train SuPFuNIS for different numbers of rules.
- The free parameters for this application are 6r+2.
- Three initial states, (x,Φ,θ) = (3,30),(10,220), and (13,30) were used to test the performance of the controller.

Sample Data Set

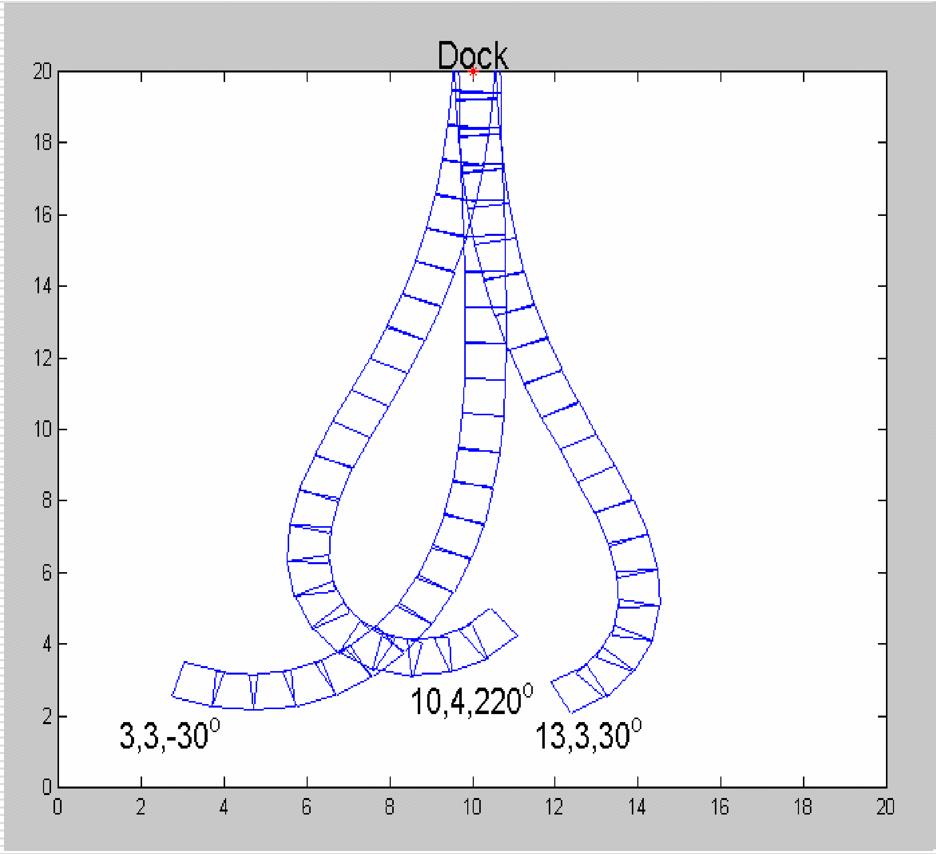
1.00	0.00	-19.00
1.95	9.37	-17.95
2.88	18.23	-16.90
3.79	26.59	-15.85
4.65	34.44	-14.80
5.45	41.78	-13.75
6.18	48.60	-12.70
7.48	54.91	-11.65
7.99	60.71	-10.60
8.72	65.99	-9.55
9.01	70.75	-8.50
9.28	74.98	-7.45
9.46	78.70	-6.40
9.59	81.90	-5.34
9.72	84.57	-4.30
9.81	86.72	-3.25
9.88	88.34	-2.20
9.91	89.44	0.00

Starting Point (1,0)

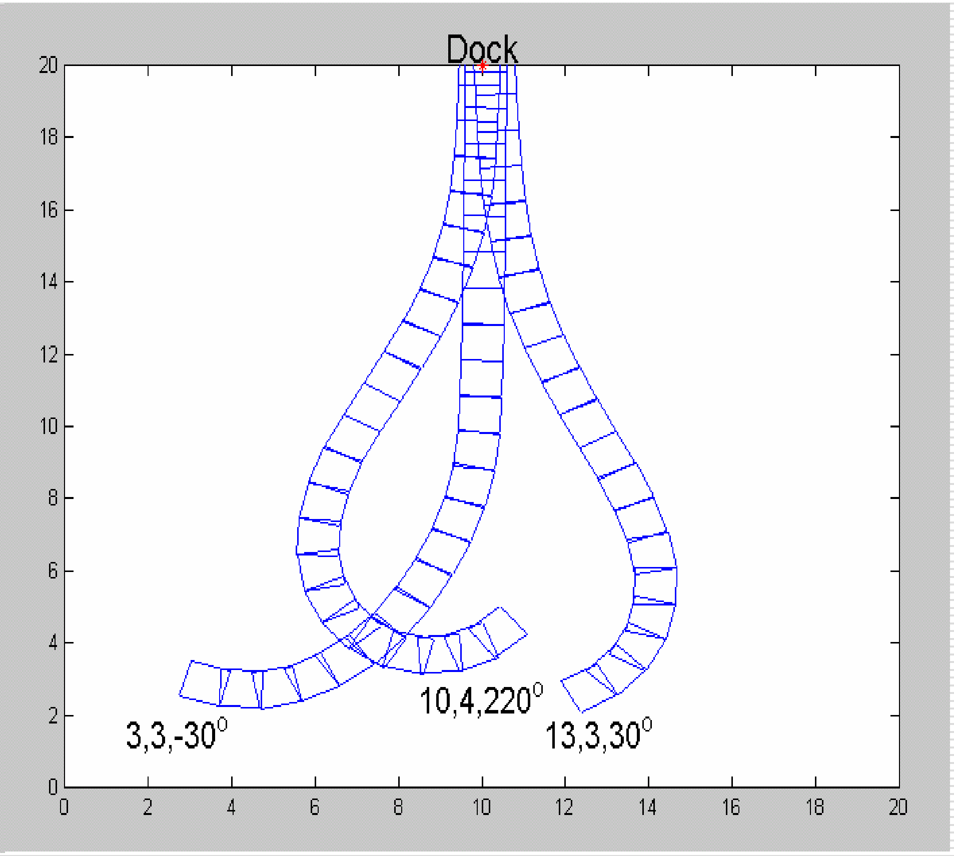
1.00	90.00	18.00
1.15	81.11	16.00
1.43	73.19	14.00
1.83	66.24	12.00
2.31	60.27	10.00
2.88	55.29	8.00
3.50	51.30	6.00
4.16	48.31	4.00
4.86	46.31	2.00
5.56	45.31	0.00
6.26	45.31	-2.00
6.95	46.31	-4.00
7.61	48.31	-6.00
8.23	51.30	-8.00
8.79	55.29	-10.00
9.28	60.27	-12.00
9.67	66.24	-14.00
9.95	73.19	-16.00
10.09	81.11	-18.00
10.09	90.00	0.00

Starting Point (1,90)

Truck Backer Upper Trajectories



3 rules



5 rules

Performance Measure

- The performance of the controller is considered good if a proper balance is maintained between the type of trajectory and trajectory destination.
  - Normalized Docking Error (NDE)
  - Trajectory Error (TE)

$$\mathrm{NDE} = \sqrt{\left(\frac{\phi_f - \phi_a}{360}\right)^2 + \left(\frac{x_f - x_a}{20}\right)^2}$$

$$\mathrm{TE} = \frac{\text{length of truck trajectory}}{\text{distance(initial position, desired final position)}}$$

Docking and Trajectory Errors

Errors when 238 pairs of numeric data are used
For 5 rule SuPFuNIS

Initial point (x, y, ϕ)	Normalized Docking Error	Trajectory Error
(13,3,30)	0.0058	1.0533
(10,4,220)	0.0120	1.2059
(3,3,-30)	0.0088	1.2106

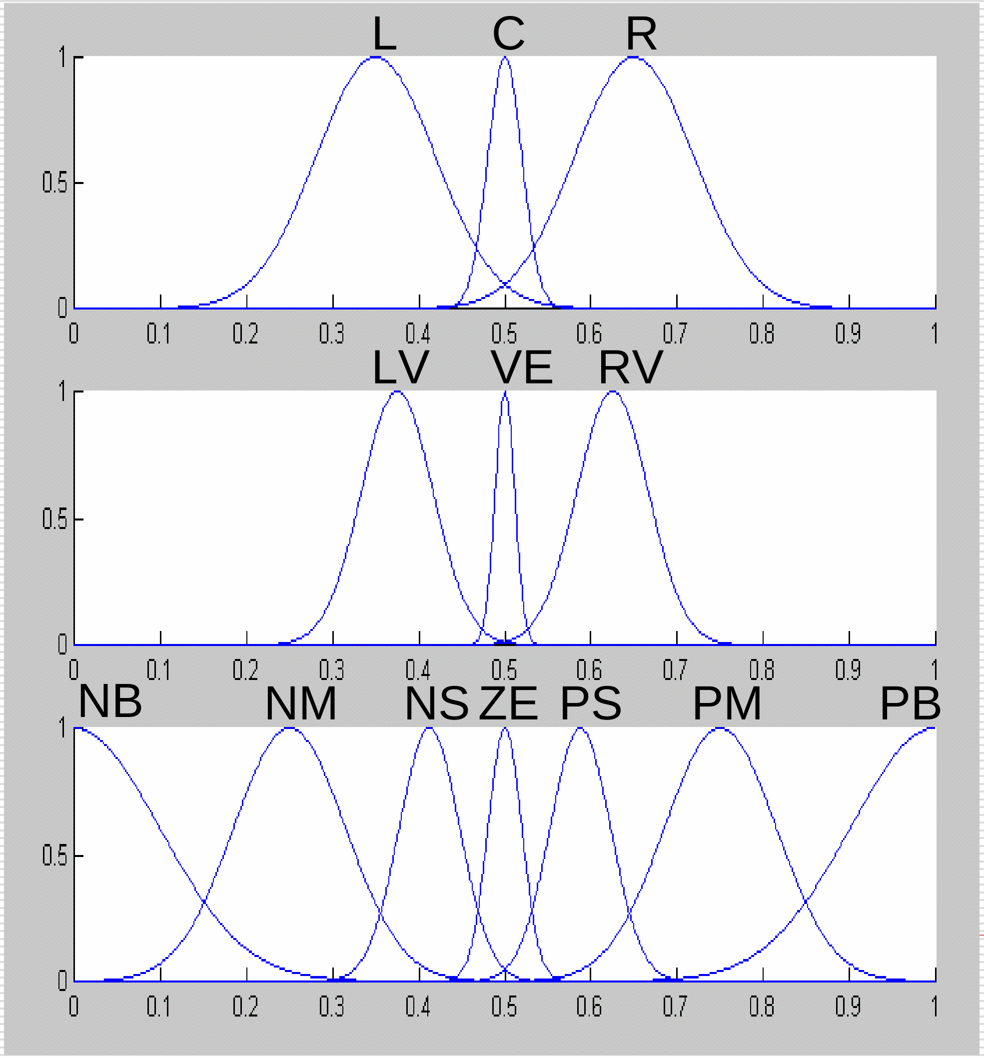
Augmentation of SuPFUNIS with Expert Linguistic Knowledge

- Training is done using reduced set (42 pairs) considering only first three pairs of data from each of the 14 sequences.
- Finer control is done using the linguistic rules constructed from the expert knowledge.

ANGLE ϕ	LEFT (L)	CENTER (C)	RIGHT (R)
RIGHT VERTICAL (RV)	NEGATIVE SMALL (NS)	NEGATIVE MEDIUM (NM)	NEGATIVE BIG (NB)
VERTICAL (VE)	POSITIVE MEDIUM (PM)	ZERO (ZE)	NEGATIVE MEDIUM (NM)
LEFT VERTICAL (LV)	POSTIVIE BIG (PB)	POSITIVE MEDIUM (PM)	POSITIVE SMALL (PS)

POSITION x

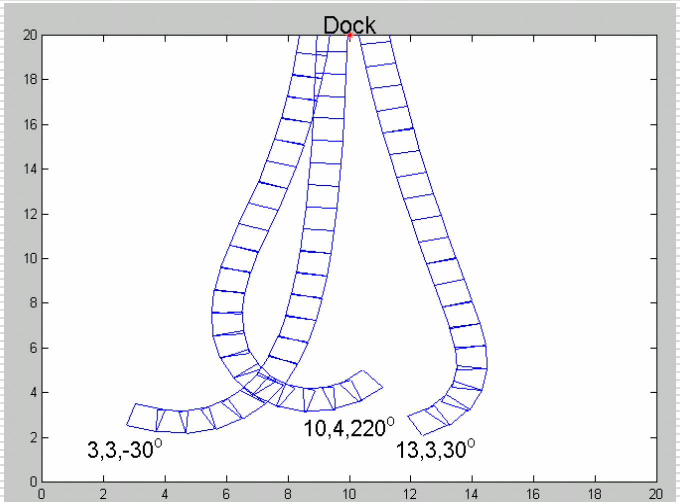
Fuzzy sets for linguistic labels of x, ϕ, θ



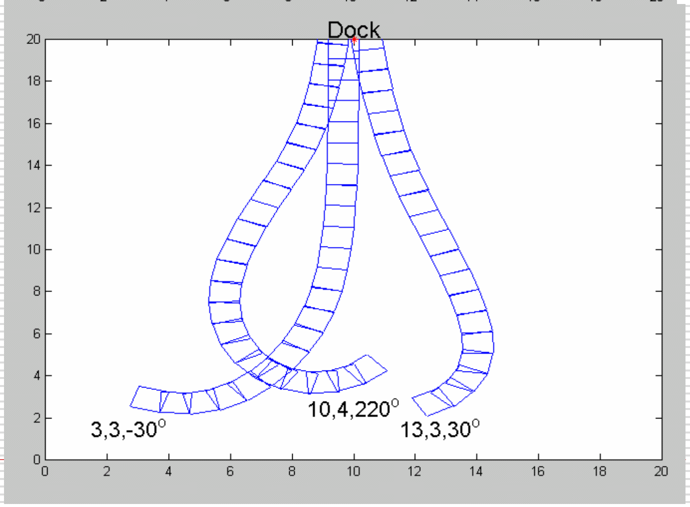
x-position x	(center,spread)
L:Left	(0.35,0.098)
C: Center	(0.5,0.028)
R:Right	(0.65,0.098)
Angle ϕ	(center,spread)
LV: Left Vertical	(0.375,0.059)
VE: Vertical	(0.5,0.016)
RV: Right Vertical	(0.625,0.059)
Steering-angle signal θ	(center,spread)
NB: Negative Big	(0.00,0.14)
NM: Negative Medium	(0.25,0.092)
NS: Negative Small	(0.4125,0.05)
ZE: Zero	(0.5,0.028)
PS: Positive Small	(0.5875,0.05)
PM: Positive Medium	(0.75,0.092)
PB: Positive Big	(1.00, 0.14)

Truck Backer Upper Trajectories

- 5 rules (reduced numeric data)



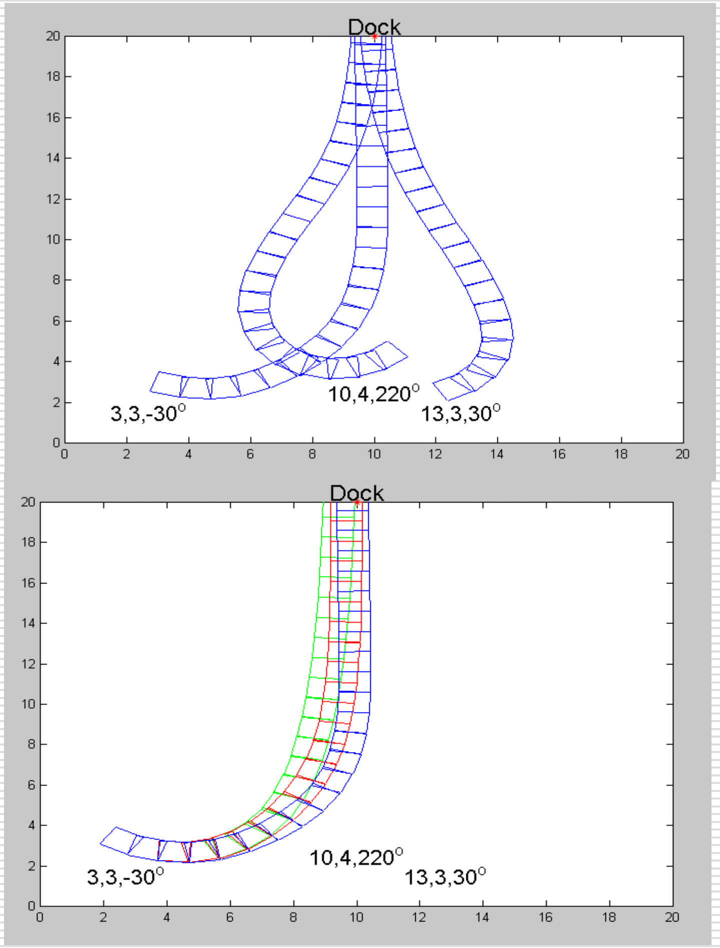
- 5 rules (reduced numeric data) + 5 expert rules



Truck Backer Upper Trajectories

- 5 rules(reduced numeric data) + 9 expert rules

- Comparison of (a), (b), (c) for 3,3,-30



Docking and Trajectory Errors

Initial point (x, y, ϕ)	Rules Numeric+ Linguistic	Normalized Docking Error	Trajectory Error
(3,3,-30)	5+0	0.0277	1.148649
(10,4,220))	5+0	0.0531	1.282923
(13,3,30))	5+0	0.0428	1.059538
(3,3,-30)	5+5	0.0158	1.152186
(10,4,220))	5+5	0.0332	1.282707
(13,3,30))	5+5	0.0259	1.059563
(3,3,-30)	5+9	0.0065	1.155053
(10,4,220))	5+9	0.0166	1.211574
(13,3,30))	5+9	0.0107	1.055119