# Elementos de Inteligencia Artificial

Amaury Caballero Ph.D., P.E.

Universidad Internacional de la Florida

# Artificial intelligence (AI) (Wikipedia)

is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior. Major AI researchers and textbooks define this field as "the study and design of intelligent agents",[1] in which an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.[2] John McCarthy, who coined the term in 1955,[3] defines it as "the science and engineering of making intelligent machines".[

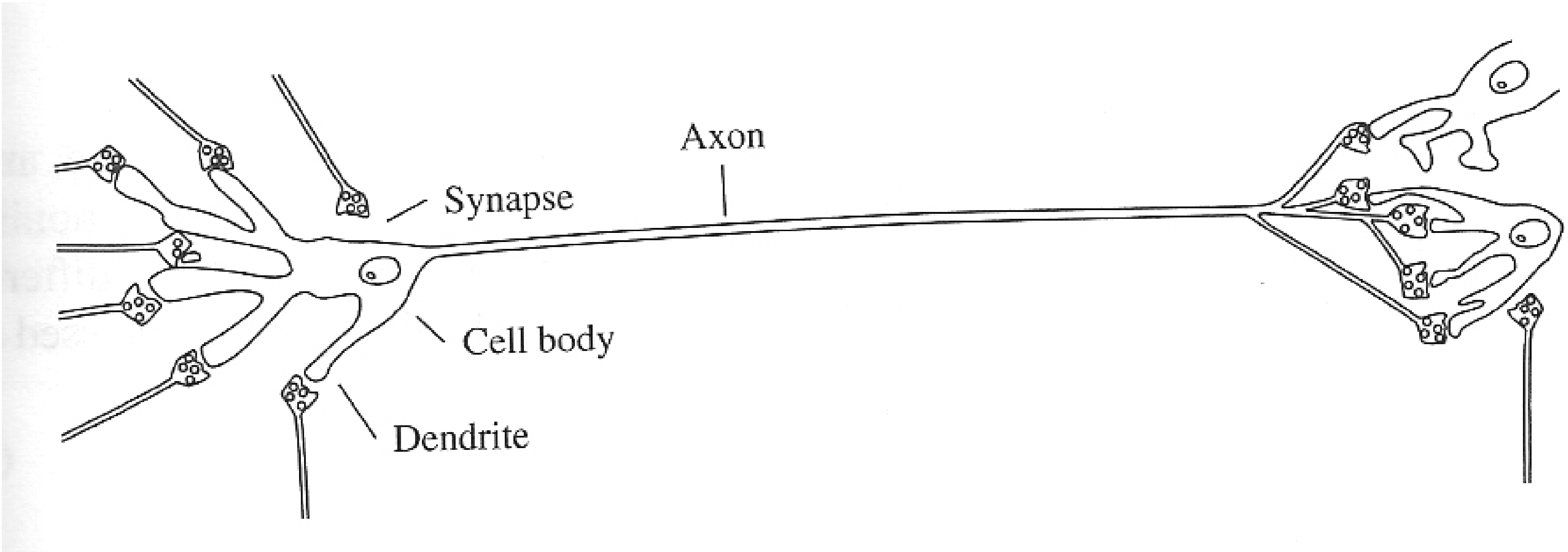
# Some Tools

- **Search and optimization:** Mathematical optimization, Evolutionary computation (Ex: Genetic algorithms)
- **Logic:** is used for knowledge representation and problem solving (Ex: Fuzzy logic)
- **Probabilistic methods for uncertain reasoning:** Bayesian networks, Markov models
- **Classifiers and statistical learning methods**
- **Neural networks**
- **Control theory**

# Neural Networks

# The Pyramidal Cell

- A neuron is made up of several protrusions called dendrites and a long branch called the axon. A neuron is joined to other neurons through the dendrites. The dendrites of different neurons meet to form synapses, the areas where messages pass. The neurons receive the impulses via the synapses. If the total of the impulses received exceeds a certain threshold value, then the neuron sends an impulse down the axon where the axon is connected to other neurons through more synapses.



# What is a Neural Network?

- A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
- 1. Knowledge is acquired by the network from its environment through a learning process.
- 2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

# Areas of Application of Neural networks (1)

| | |
|---|---|
| **Biology** | **Brain research, retina research, blood cell classification, classification of sub-species (Iris database), etc** |
| **Industrial Engineering** | **Geological formations (oil fields, etc), Database management, Optimization of air plane schedule and occupations, Pattern recognition** |
| **Electrical Engineering** | **Network security (dynamic, static, voltage), Prediction (load, harmonics), Fault detection in primary circuits and transmission lines, rotating machines, transformers, Fault diagnosis in perturbations, Load dispatching, Line overloads control and detection, State estimation, Optimal capacitor and filter switching, Parameter estimation, Design of stabilizers and other devices** |

# Areas of Application of Neural networks (2)

| | |
|---|---|
| Environmental Engineering | Analysis of tendencies and patterns, Weather prediction |
| Finances | Price development prediction, Stock market analysis and prediction, Credit risk evaluation, Fraud detection, Signature interpretation |
| Manufacturing | Specialized robots and control systems (artificial vision, pressure sensors, temperature,gas, etc), Production control in industrial processes, Quality control |
| | Specialized robots and control systems (artificial vision, pressure sensors, temperature,gas, etc), Production control in industrial processes, Quality control |

# Areas of Application of Neural networks (3)

| | |
|---|---|
| **Biomedical Engineering** | **Epilepsy, auditory/comprehension tests, Voice analysis for helping deaf individuals, Diagnosis and treatment of diseases (epilepsy, etc) according to symptoms and/or analytical data (EEG, EKG, blood analysis, etc), Surgery monitoring, Prediction of adverse side effects of medicines, Biomedical image registration and related topics** |
| **Military** | **Radar signal classification, Weapons, Optimization in the use of spare resources, continuous recognition in target shooting** |

# Areas of Application of Neural networks (4)

| | |
|---|---|
| **Aviation** | **Noise reduction applied to cockpit voice, Modeling of airplanes stability during flights, etc, Gate assignment for aircrafts (United/TWA) (size/capacity-gate matching + minimizing distances between connecting flights, foreign vs. domestic)** |
| **Signal processing** | **Noise cancellation, pattern recognition and extraction, etc** |

# Properties of ANN

# Properties of ANN

- Nonlinearity
- Input-output mapping
- Adaptivity
- Evidential response
- Contextual information
- Fault tolerance
- VLSI implementability
- Uniformity of analysis and design
- Neurobiological analogy

# 1. Nonlinearity

- An artificial neuron can be linear or nonlinear.
- The nonlinearity is distributed through the network

# 2. Input—Output Mapping

- A popular paradigm of learning called learning with a teacher or supervised learning involves modification of the synaptic weights of a neural network by applying a set of labeled training samples or task examples. Each example consists of a unique input signal and a corresponding desired response. The network is presented with an example picked at random from the set, and the synaptic weights (free parameters) of the network are modified to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion. The training of the network is repeated for many examples in the set until the network reaches a steady state where there are no further significant changes in the synaptic weights. The previously applied training examples may be reapplied during the training session but in a different order. Thus the network learns from the examples by constructing an input—output mapping for the problem at hand.

# 3. Adaptivity

- Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover,when it is operating in a nonstationary environment (i.e., one where statistics change with time), a neural network can be designed to change its synaptic weights in real time.

# 4. Evidential Response

- In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the decision made. This latter information may be used to reject ambiguous patterns, should they arise, and thereby improve the classification performance of the network.

# 5. Contextual Information

- Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally by a neural network.

# 6. Fault Tolerance

- A neural network, implemented in hardware form, has the potential to be inherently fault tolerant, or capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions. For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality. However, due to the distributed nature of information stored in the network, the damage has to be extensive before the overall response of the network is degraded seriously. Thus, in principle, a neural network exhibits a graceful degradation in performance rather than catastrophic failure.

# 7. VLSI Implementability

- The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network well suited for implementation using very-large-scale-integrated (VLSI) technology. One particular beneficial virtue of VLSI is that it provides a means of capturing truly complex behavior in a highly hierarchical fashion.

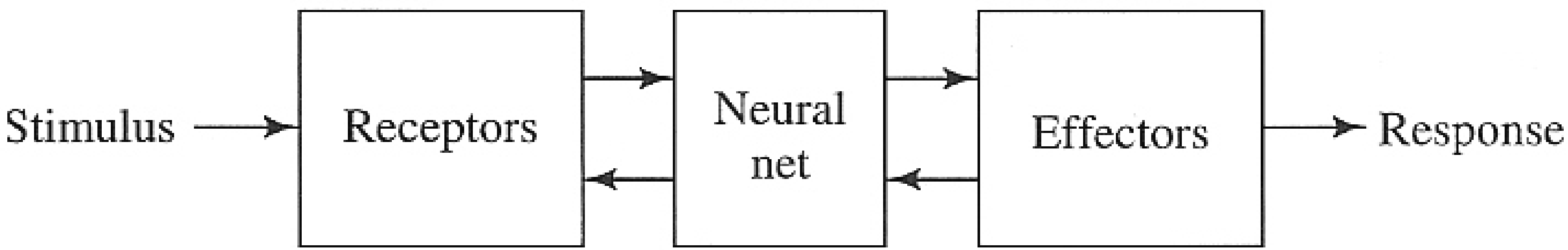
# 8. Uniformity of Analysis and Design

- Basically, neural networks enjoy universality as information processors. We say this in the sense that the same notation is used in all domains involving the application of neural networks.

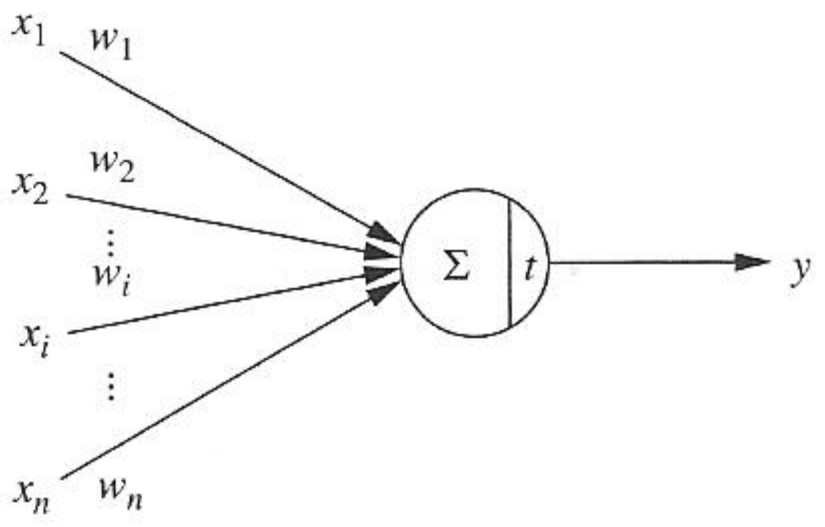
# 9. Neurobiological Analogy

- The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful. Neurobiologists look to (artificial) neural networks as a research tool for the interpretation of neurobiological phenomena. On the other hand, engineers look to neurobiology for new ideas to solve problems more complex than those based on conventional hard-wired design techniques.

# Human Brain



# Model of a Neuron




$$y = F\left(\sum w_i x_i - t\right)$$

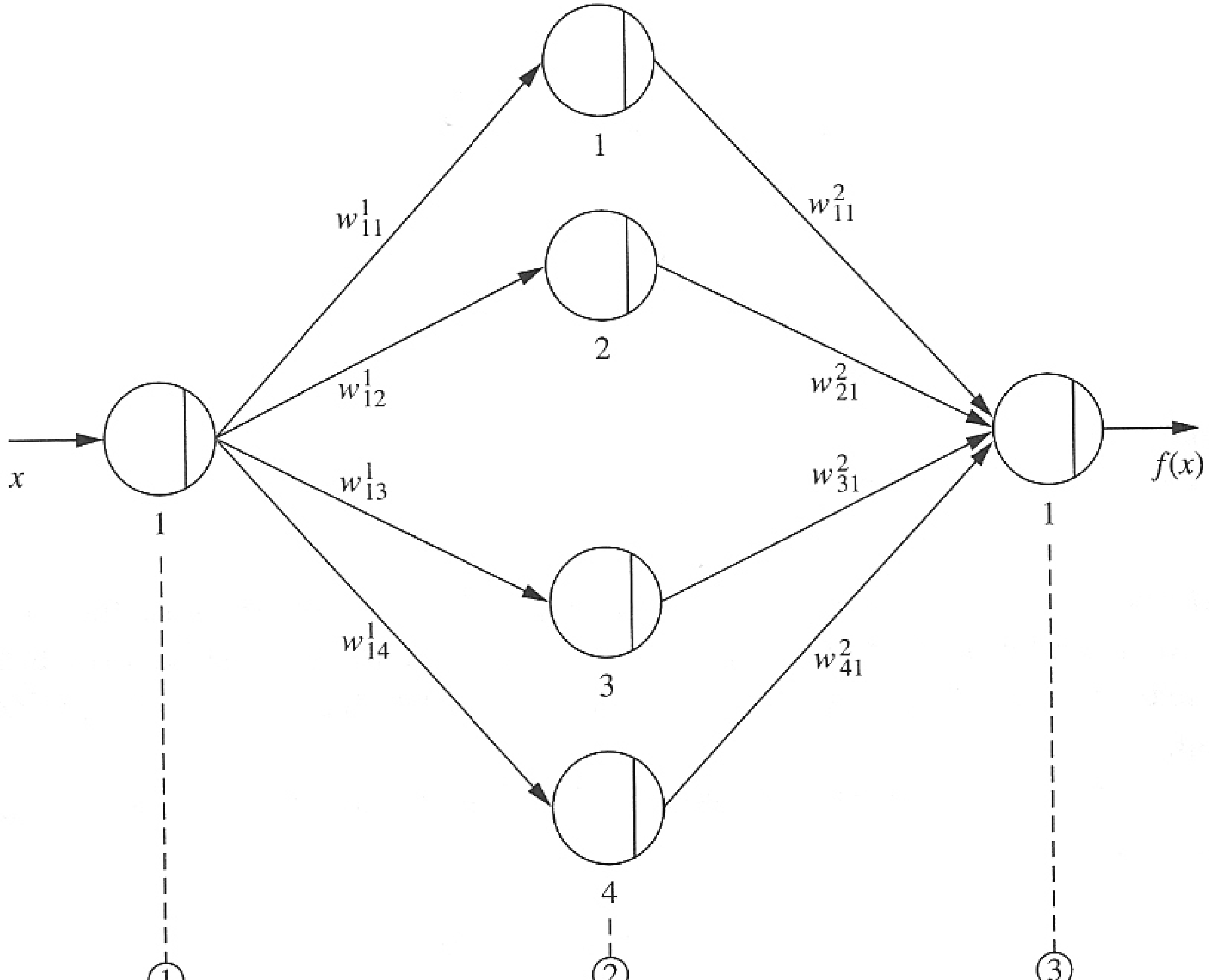
where $x_i$ signal input $(i = 1, 2, \ldots, n)$

$w_i$ weight associated with the signal input $x_i$

$t$ threshold level prescribed by user

$F(s)$ is a nonlinear function, e.g., a sigmoid function $F(s) = \dfrac{1}{1 + e^{-s}}$

# Network Architecture



# How Do ANN Work?

Neural systems solve problems by adapting to the nature of the data (signals) they receive. One of the ways to accomplish this is to use a training data set and a checking data set of input and output data/signals $(x, y)$ (for a multiple-input, multiple-output system using a neural network, we may use input–output sets comprised of vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$). We start with a random assignment of weights $w^i_{jk}$ to the paths joining the elements in the different layers Then an input $x$ from the training data set is passed through the neural network. The neural network computes a value $(f(x)_{\text{output}})$, which is compared with the actual value $(f(x)_{\text{actual}} = y)$. The error measure $E$ is computed from these two output values as

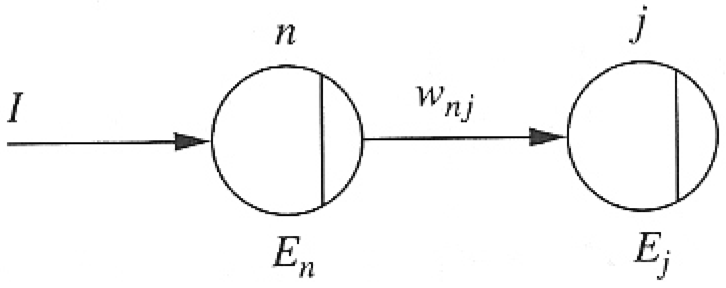
$$E = f(x)_{\text{actual}} - f(x)_{\text{output}}$$

# Error for the Element n

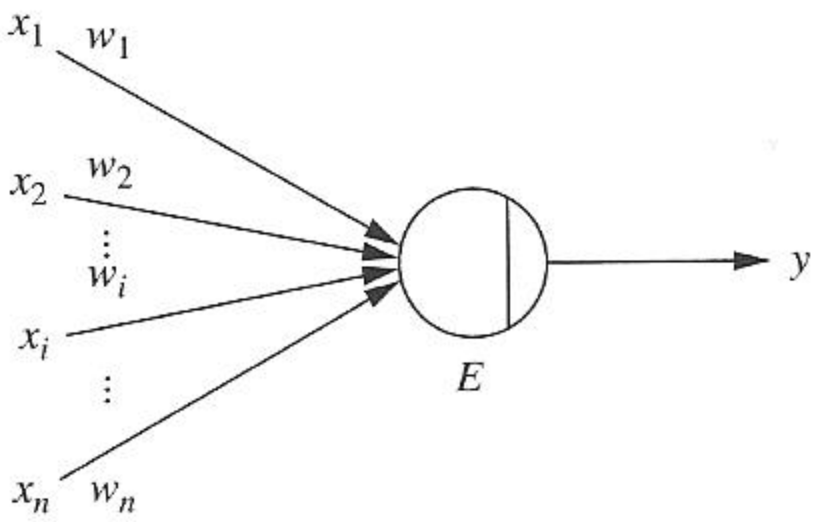
$$E_n = F'(I) w_{nj} E_j$$

where, for $F(I) = 1/(1 + e^{-I})$, the sigmoid function, we have

$$F'(I) = F(I)(1 - F(I))$$



# Weights Update

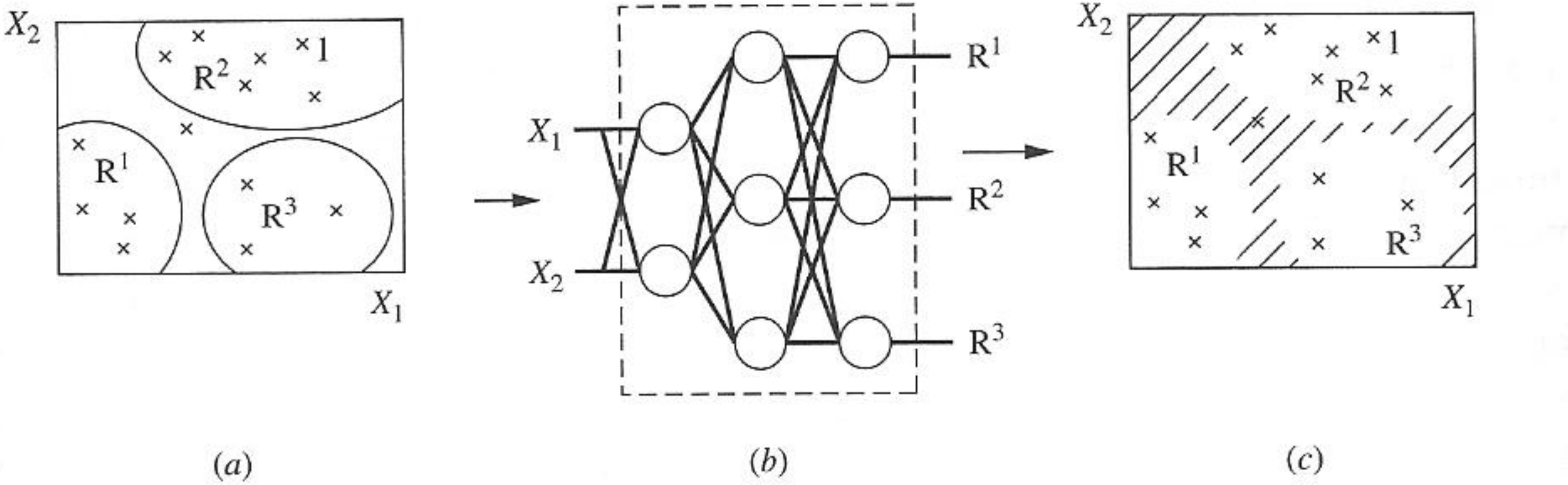



For an element with an error $E$ associated with it, as shown in Fig., the associated weights may be updated as

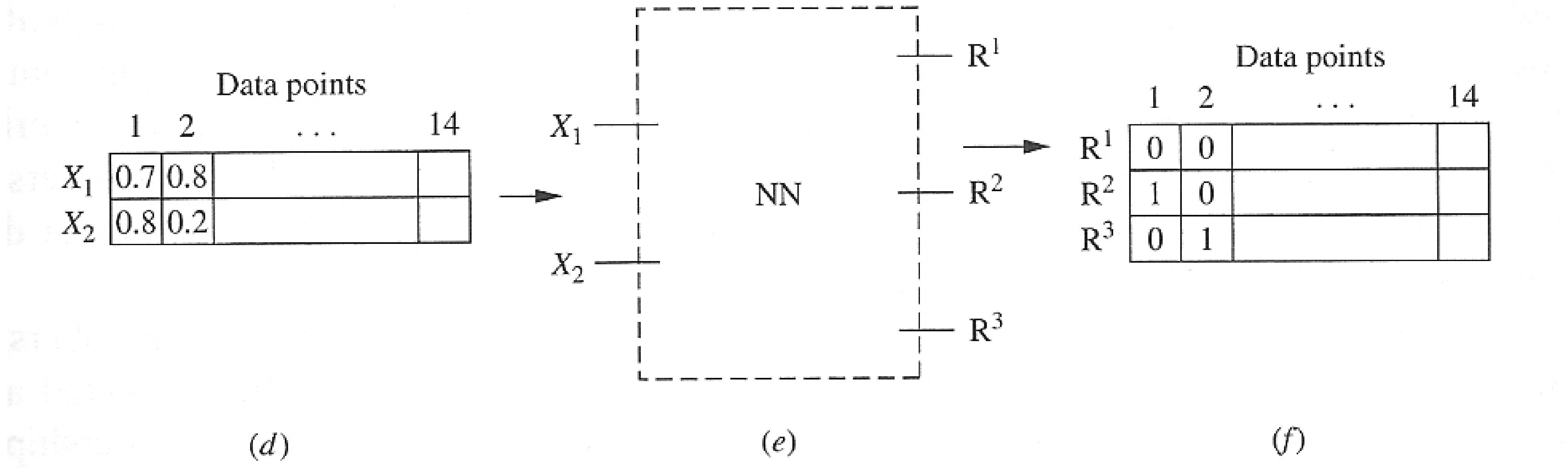
$$w_i \text{ (new)} = w_i \text{ (old)} + \alpha E x_i$$

where $\alpha$ = learning constant
$E$ = associated error measure
$x_i$ = input to the element

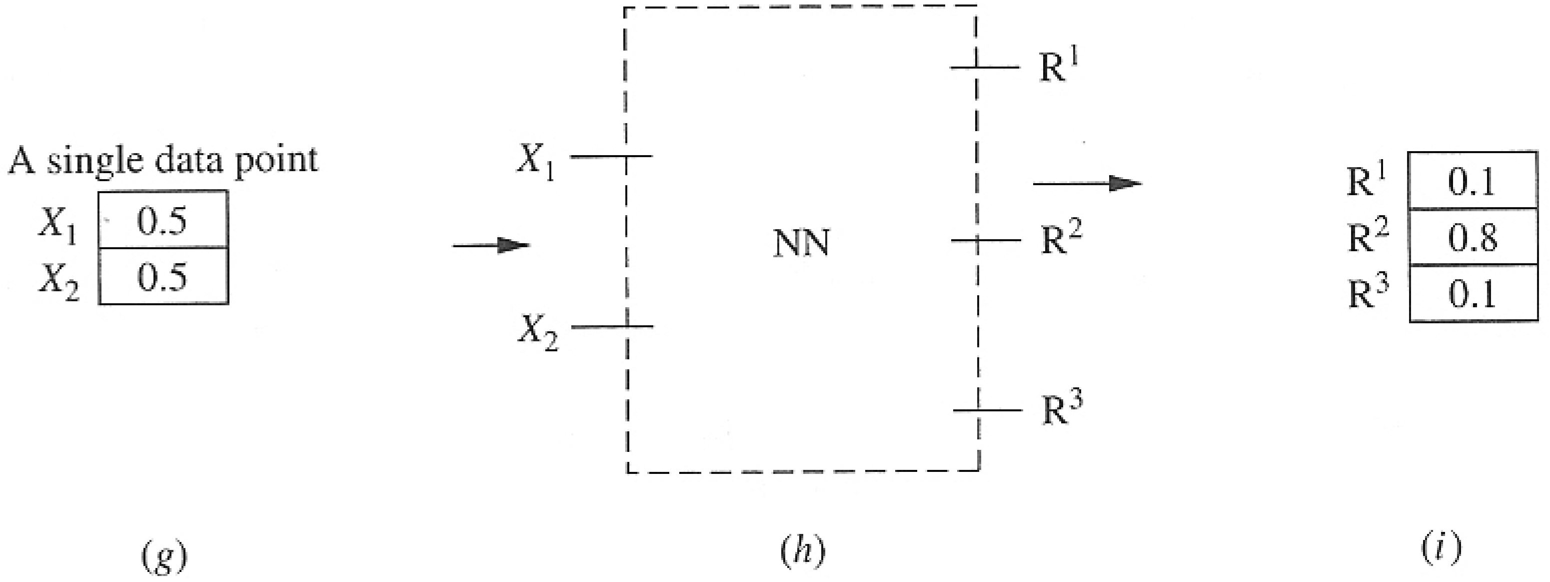
# Generation of Membership Functions Using ANN (1)



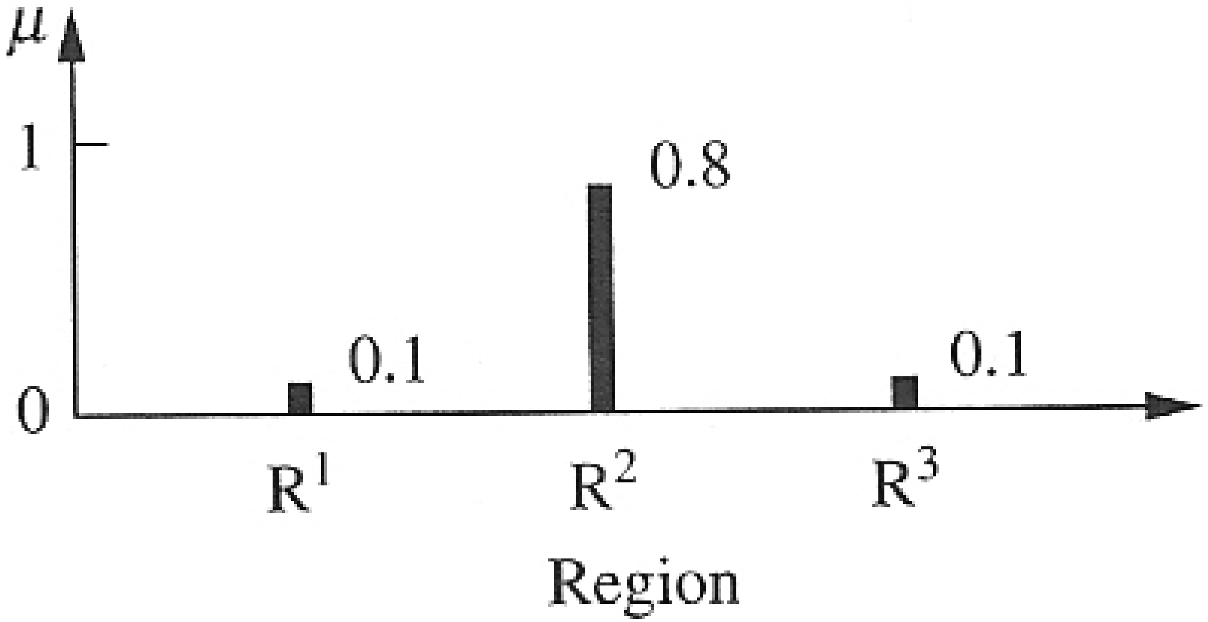
# Generation of Membership Functions Using ANN (2)



# Generation of Membership Functions Using ANN (3)



# Membership Function for Data Point (0.5, 0.5)



# Example

Variables describing the data points to be used as a training data set

| Data point | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.05 | 0.09 | 0.12 | 0.15 | 0.20 | 0.75 | 0.80 | 0.82 | 0.90 | 0.95 |
| $x_2$ | 0.02 | 0.11 | 0.20 | 0.22 | 0.25 | 0.75 | 0.83 | 0.80 | 0.89 | 0.89 |

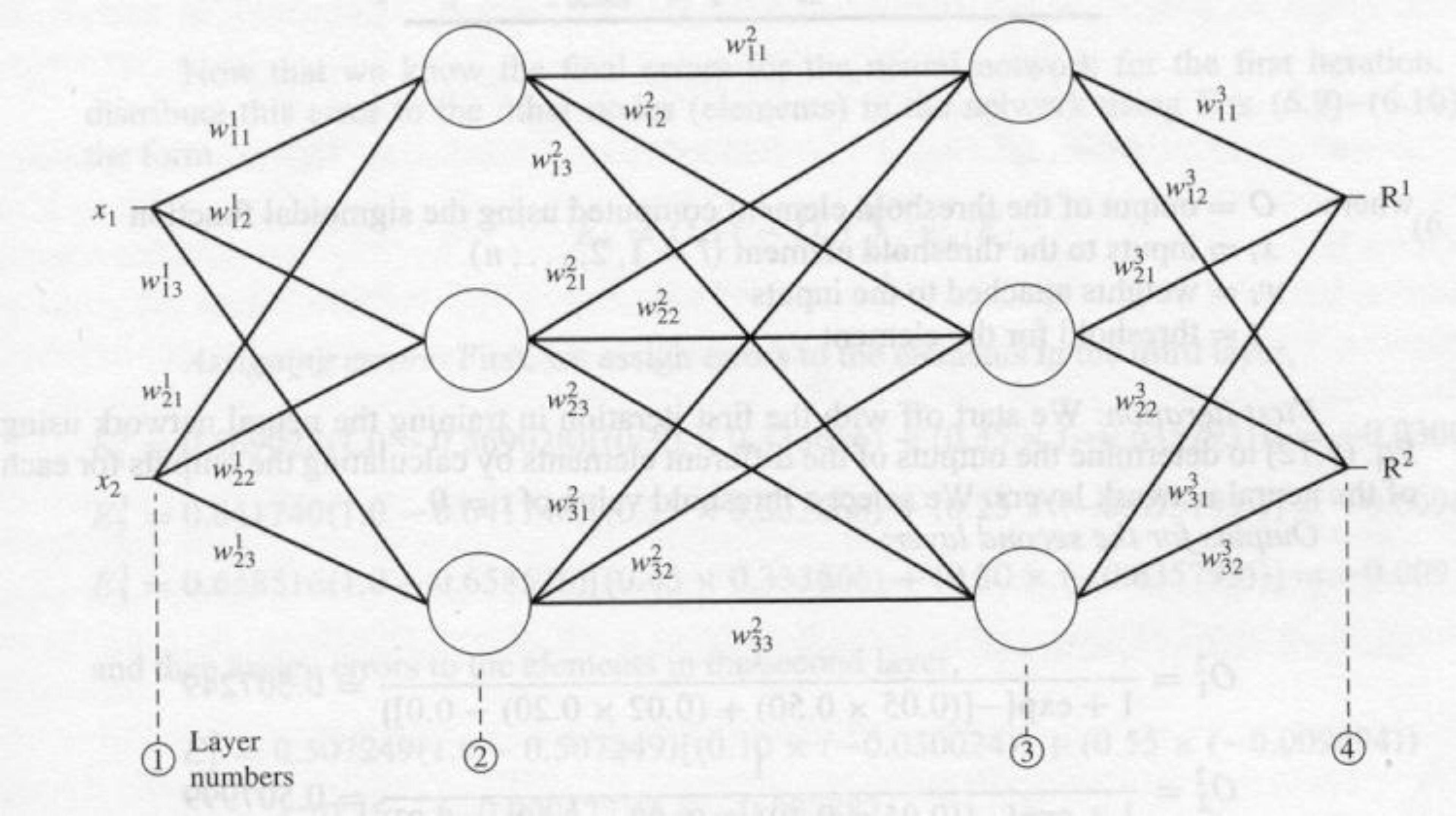
Variables describing the data points to be used as a checking data set

| Data point | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.09 | 0.10 | 0.14 | 0.18 | 0.22 | 0.77 | 0.79 | 0.84 | 0.94 | 0.98 |
| $x_2$ | 0.04 | 0.10 | 0.21 | 0.24 | 0.28 | 0.78 | 0.81 | 0.82 | 0.93 | 0.99 |

Membership Values of the Data Points in the Training and checking Data Sets to be Used for Training and Checking the Performance of the ANN. The data points that are to be used for training and checking the performance of the ANN have been assigned membership values of “1” for the classes they have been originally assigned

| Data points | 1 & 11 | 2 & 12 | 3 & 13 | 4 & 14 | 5 & 15 | 6 & 16 |
|---|---|---|---|---|---|---|
| $R_1$ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 |
| $R_2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| | **7 & 17** | **8 & 18** | **9 & 19** | **10 & 20** | | |
| | 0.0 | 0.0 | 0.0 | 0.0 | | |
| | 1.0 | 1.0 | 1.0 | 1.0 | | |

# ANN to be Trained



Assigned Weights. As input to the NN will be used the eq. $O = \frac{1}{1+\exp[-(\sum x_i w_i - t)]}$

| | | |
|---|---|---|
| $w^1_{11} = 0.5$ | $w^2_{11} = 0.10$ | $w^3_{11} = 0.30$ |
| $w^1_{12} = 0.4$ | $w^2_{12} = 0.55$ | $w^3_{12} = 0.35$ |
| $w^1_{13} = 0.1$ | $w^2_{13} = 0.35$ | $w^3_{21} = 0.35$ |
| $w^1_{21} = 0.2$ | $w^2_{21} = 0.20$ | $w^3_{22} = 0.25$ |
| $w^1_{22} = 0.6$ | $w^2_{22} = 0.45$ | $w^3_{31} = 0.45$ |
| $w^1_{23} = 0.2$ | $w^2_{23} = 0.35$ | $w^3_{32} = 0.30$ |
| | $w^2_{31} = 0.25$ | |
| | $w^2_{32} = 0.15$ | |
| | $w^2_{33} = 0.60$ | |

where $O$ = output of the threshold element computed using the sigmoidal function
$x_i$ = inputs to the threshold element ($i = 1, 2, \ldots, n$)
$w_i$ = weights attached to the inputs
$t$ = threshold for the element

# First Iteration

*First iteration*: We start off with the first iteration in training the neural network using Eq. (6.12) to determine the outputs of the different elements by calculating the outputs for each of the neural network layers. We select a threshold value of $t = 0$.

*Outputs for the second layer*:

$$O_1^2 = \frac{1}{1 + \exp\{-[(0.05 \times 0.50) + (0.02 \times 0.20) - 0.0]\}} = 0.507249$$

$$O_2^2 = \frac{1}{1 + \exp\{-[(0.05 \times 0.40) + (0.02 \times 0.60) - 0.0]\}} = 0.507999$$

$$O_3^2 = \frac{1}{1 + \exp\{-[(0.05 \times 0.10) + (0.02 \times 0.20) - 0.0]\}} = 0.502250$$

# Outputs for the Third Layer

$$O_1^3 = \frac{1}{1 + \exp\{-[(0.507249 \times 0.10) + (0.507999 \times 0.20) + (0.502250 \times 0.25) - 0.0]\}}$$
$$= 0.569028$$

$$O_2^3 = \frac{1}{1 + \exp\{-[(0.507249 \times 0.55) + (0.507999 \times 0.45) + (0.502250 \times 0.15) - 0.0]\}}$$
$$= 0.641740$$

$$O_3^3 = \frac{1}{1 + \exp\{-[(0.507249 \times 0.35) + (0.507999 \times 0.35) + (0.502250 \times 0.60) - 0.0]\}}$$
$$= 0.658516$$

# Outputs for the Fourth Layer

*Outputs for the fourth layer*:

$$O_1^4 = \frac{1}{1 + \exp\{-[(0.569028 \times 0.30) + (0.641740 \times 0.35) + (0.658516 \times 0.45) - 0.0]\}}$$
$$= 0.666334$$

$$O_2^4 = \frac{1}{1 + \exp\{-[(0.569028 \times 0.35) + (0.641740 \times 0.25) + (0.658516 \times 0.30) - 0.0]\}}$$
$$= 0.635793$$

# Determining Errors

$$\mathrm{R}_1 : E_1^4 = O_{1\,\mathrm{actual}}^4 - O_1^4 = 1.0 - 0.666334 = 0.333666$$

$$\mathrm{R}_2 : E_2^4 = O_{2\,\mathrm{actual}}^4 - O_2^4 = 0.0 - 0.635793 = -0.635793$$

# Assigning Errors: $E_n = O_n\ (1 - O_n\ ) \sum W_{nj}\ E_j\ )$

*Assigning errors*: First, we assign errors to the elements in the third layer,

$$E_1^3 = 0.569028(1.0 - 0.569028)[(0.30 \times 0.333666) + (0.35 \times (-0.635793))] = -0.030024$$

$$E_2^3 = 0.641740(1.0 - 0.641740)[(0.35 \times 0.333666) + (0.25 \times (-0.635793))] = -0.009694$$

$$E_3^3 = 0.658516(1.0 - 0.658516)[(0.45 \times 0.333666) + (0.30 \times (-0.635793))] = -0.009127$$

and then assign errors to the elements in the second layer,

$$E_1^2 = 0.507249(1.0 - 0.507249)[(0.10 \times (-0.030024)) + (0.55 \times (-0.009694)) + (0.35 \times (-0.009127))] = -0.002882$$

$$E_2^2 = 0.507999(1.0 - 0.507999)[(0.20 \times (-0.030024)) + (0.45 \times (-0.009694)) + (0.35 \times (-0.009127))] = -0.003390$$

$$E_3^2 = 0.502250(1.0 - 0.502250)[(0.25 \times (-0.030024)) + (0.15 \times (-0.009694)) + (0.60 \times (-0.009127))] = -0.003609$$

# Updating 0weights

$$w^i_{jk}(\text{new}) = w^i_{jk}(\text{old}) + \alpha E^{i+1}_k x_{jk}$$

where
$w^i_{jk}$ = represents the weight associated with the path connecting the $j$th element of the $i$th layer to the $k$th element of the $(i+1)$th layer
$\alpha$ = learning constant, which we will take as 0.3 for this example
$E^{i+1}_k$ = error associated with the $k$th element of the $(i+1)$th layer
$x_{jk}$ = input from the $j$th element in the $i$th layer to the $k$th element in the $(i+1)$th layer $(O^i_j)$

# Updating Weights Connecting Elements of the 3rd and 4th Layers

$$w_{11}^{3} = 0.30 + 0.3 \times 0.333666 \times 0.569028 = 0.356960$$

$$w_{21}^{3} = 0.35 + 0.3 \times 0.333666 \times 0.641740 = 0.414238$$

$$w_{31}^{3} = 0.45 + 0.3 \times 0.333666 \times 0.658516 = 0.515917$$

$$w_{12}^{3} = 0.35 + 0.3 \times (-0.635793) \times 0.569028 = 0.241465$$

$$w_{22}^{3} = 0.25 + 0.3 \times (-0.635793) \times 0.641740 = 0.127596$$

$$w_{32}^{3} = 0.30 + 0.3 \times (-0.635793) \times 0.658516 = 0.174396$$

# Updating Weights Connecting Elements of the 2nd and 3rd Layers

$$w_{11}^{2} = 0.10 + 0.3 \times (-0.030024) \times 0.507249 = 0.095431$$

$$w_{21}^{2} = 0.20 + 0.3 \times (-0.030024) \times 0.507999 = 0.195424$$

$$w_{31}^{2} = 0.25 + 0.3 \times (-0.030024) \times 0.502250 = 0.245476$$

$$w_{12}^{2} = 0.55 + 0.3 \times (-0.009694) \times 0.507249 = 0.548525$$

$$w_{22}^{2} = 0.45 + 0.3 \times (-0.009694) \times 0.507999 = 0.448523$$

$$w_{32}^{2} = 0.15 + 0.3 \times (-0.009694) \times 0.502250 = 0.148540$$

$$w_{13}^{2} = 0.35 + 0.3 \times (-0.009127) \times 0.507249 = 0.348611$$

$$w_{23}^{2} = 0.35 + 0.3 \times (-0.009127) \times 0.507999 = 0.348609$$

$$w_{33}^{2} = 0.60 + 0.3 \times (-0.009127) \times 0.502250 = 0.598625$$

# Updating Weights Connecting Elements of the 1st and 2nd Layers

$$w_{11}^{1} = 0.50 + 0.3 \times (-0.002882) \times 0.05 = 0.499957$$

$$w_{12}^{1} = 0.40 + 0.3 \times (-0.003390) \times 0.05 = 0.399949$$

$$w_{13}^{1} = 0.10 + 0.3 \times (-0.003609) \times 0.05 = 0.099946$$

$$w_{21}^{1} = 0.20 + 0.3 \times (-0.002882) \times 0.02 = 0.199983$$

$$w_{22}^{1} = 0.60 + 0.3 \times (-0.003390) \times 0.02 = 0.599980$$

$$w_{23}^{1} = 0.20 + 0.3 \times (-0.003609) \times 0.02 = 0.199978$$

# Fuzzy Logic

# Principle  of Incompatibility

As the complexity of a system increases, our ability to make *precise* and yet *significant* statements about its behavior diminishes until a threshold is reached beyond which precision and significance become almost mutually exclusive characteristics.

# Fuzzy Logic

- Fuzzy logic may be viewed as a bridge between the extensively wide gap between the precision crisp logic and the imprecision of both real world and human interpretation.
  - Prof. Zadeh

- As its name implies, the theory of *fuzzy sets* is, basically, a theory of *graded* concepts - a theory in which everything is a matter of degree or, to put it figuratively, everything has elasticity.
  - H.J. Zimmermann

# Fuzzy Logic

A technology which enhances model-based system designs using both intuition and engineering heuristics.

Decisions based on “degree of truth”

A new paradigm of systems engineering which helps achieve robust & fault-tolerant systems

An efficient way of designing, optimizing & maintaining highly complex systems transparently

# Fuzzy Logic

The center of the fuzzy modeling is the idea of linguistic variables.

***Example***:

- If project duration is *long*, the completion risk is increased.
- The engine temperature is *hot*.
- A cruise missile has a *long* range at a *high* speed.
- If you are *tall*, you are quite likely *heavy*.
- Tom is rather *tall* but Judy is *short*.
- Disposable incomes in the *middle* tax payer is *adequate*.
- That project requires a *large* manpower commitment.

## Fuzzy Logic

***Example*: *For*** the fuzzy variable "*tall*" of men, the degrees of membership depend on their heights.

| Height | Degree of membership |
|---|---|
| 5'0" | 0.0 |
| 5'4" | 0.08 |
| 5'8" | 0.32 |
| 6'0" | 0.50 |
| 6'4" | 0.82 |
| 6'8" | 0.98 |
| 7'0" | 1.00 |

The variable "tall" is defined by the range of values for heights (5'0", 5'4", ..., 7'0") and the degrees of membership (0.00, 0.08, ..., 1.00).

$$tall: heights \rightarrow [0, 1]$$

"*Heights*" is the domain of "*tall*," and [0, 1] the range.

# Types of Uncertainty

- Probabilistic uncertainty
  - example: rolling a dice
- Linguistic uncertainty
  - examples: low price, tall people, young age
- Informational uncertainty
  - example: credit worthiness, honesty

# Fuzzy vs. Probability

- Probabilistic Reasoning
  - “There is an 75% chance that Betty is old.”
  - Betty is either old or not old (the law of the excluded middle).
- Fuzzy Reasoning
  - “Betty's degree of membership within the set of old people is 0.75.”
  - Betty is like an old person, but could also have some characteristics of a young person.

# Fuzzy vs Probability

- Crisp Facts – distinct boundaries
- Fuzzy Facts – imprecise boundaries
- Probability – incomplete facts

***Example***: Scout reporting an enemy

- "Two tanks at grid NV 54" (Crisp)
- "A few tanks at grid NV 54"   (Fuzzy)
- "There might be 2 tanks at grid NV 54" (Probabilistic)

# What is Fuzzy Logic?

- It is a multi-valued Boolean logic.
- It uses rule-based control.
- It is based on a form of Artificial Intelligence.
- It uses human intuition for control.

## What is Fuzzy Logic?

Fuzzy logic methodology is basically characterized by three traits:

(1) It does not consider whether something is true or false, but rather how true it is.

(2) Because it's similar to human reasoning, its implementation tends to be based on natural language.

(3) It's flexible and can model complex, nonlinear systems by using imprecise information.

# Fuzzy Logic - What is it NOT?

- Not the solution to ALL problems. Some problems are better solved with conventional methods.

# When should Fuzzy Logic be used?

If no adequate mathematical model for a given problem can be easily found.

If non-linearity, time-constraints or multiple parameters exist.

- If engineering know-how about the given problem is available or can be acquired during the design process.

# When should not Fuzzy Logic be used?

When the problem can be easily solved using conventional control techniques, such as a PID controller.

When there is a simple, clear-defined and fast-to-solve mathematical model for the given problem available.

When the problem cannot be solved at all. There are some problems with which even fuzzy logic can not help you.

# Why Fuzzy Logic?

- Human knowledge is fuzzy: expressed in “Fuzzy”
  - Linguistic Terms - young, old, big, cheap are fuzzy words
  - Temperature is expressed as cold, warm or hot. No quantitative meaning.

- The world is *Not* binary: gradual transitions & ambiguities at the boundaries

# Why Use Fuzzy Logic?

- No complex mathematical models for system development
- Simpler and more effective implementation
- More descriptive
- Higher fault-tolerance and a better trade-off between system robustness and system sensitivity

# Fuzzy Logic Applications

**(1) <u>Automotive</u>**

(a) fuzzy engine control
(b) fuzzy cruise control
(c) fuzzy anti-lock braking systems
(d) fuzzy transmission systems

**(2) <u>Appliances</u>**

(a) washing machines
(b) air conditioners
(c) cameras
(d) VCRs
(e) microwave ovens

## Fuzzy Logic Applications

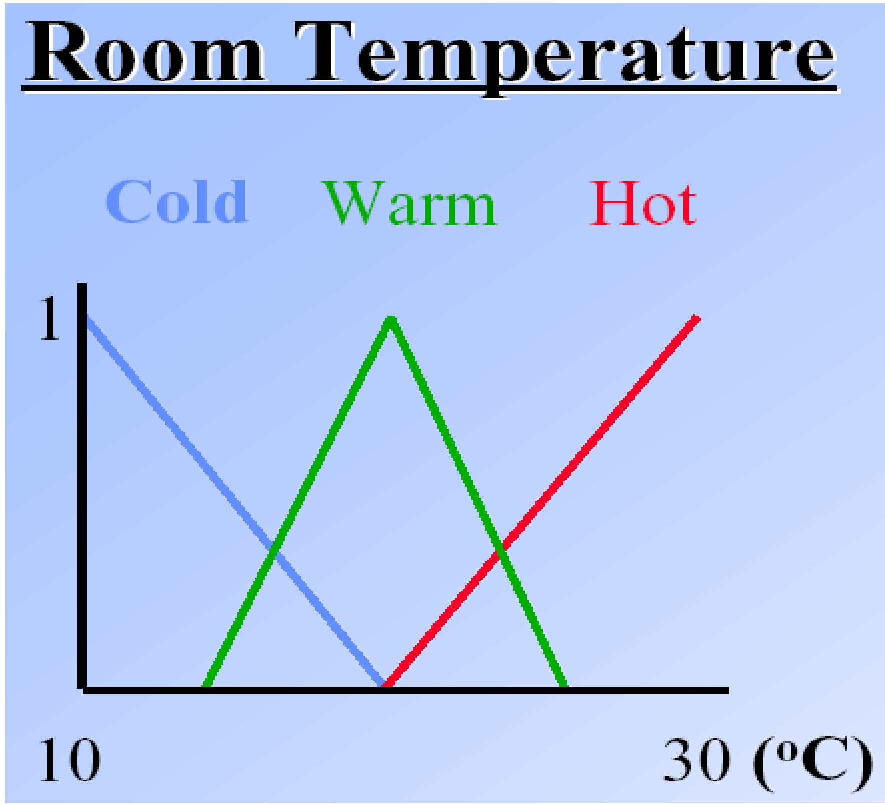
**(3) <u>Industries</u>**

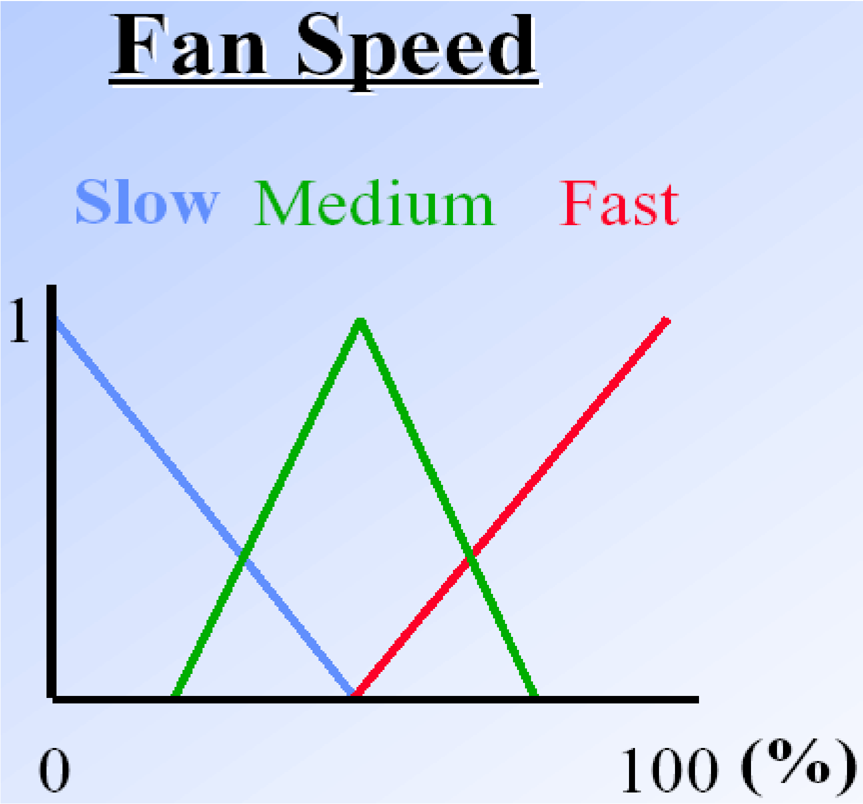
(a) chemical plants
(b) nuclear power plants
(c) elevators
(d) motor control
(e) water quality control
(f) automatic train operation systems

**(4) <u>Aerospace</u>**

(a) flexible wing control
(b) jet engine failure diagnostics
(c) spacecraft positioning control

# Example: Fuzzy Temperature Control

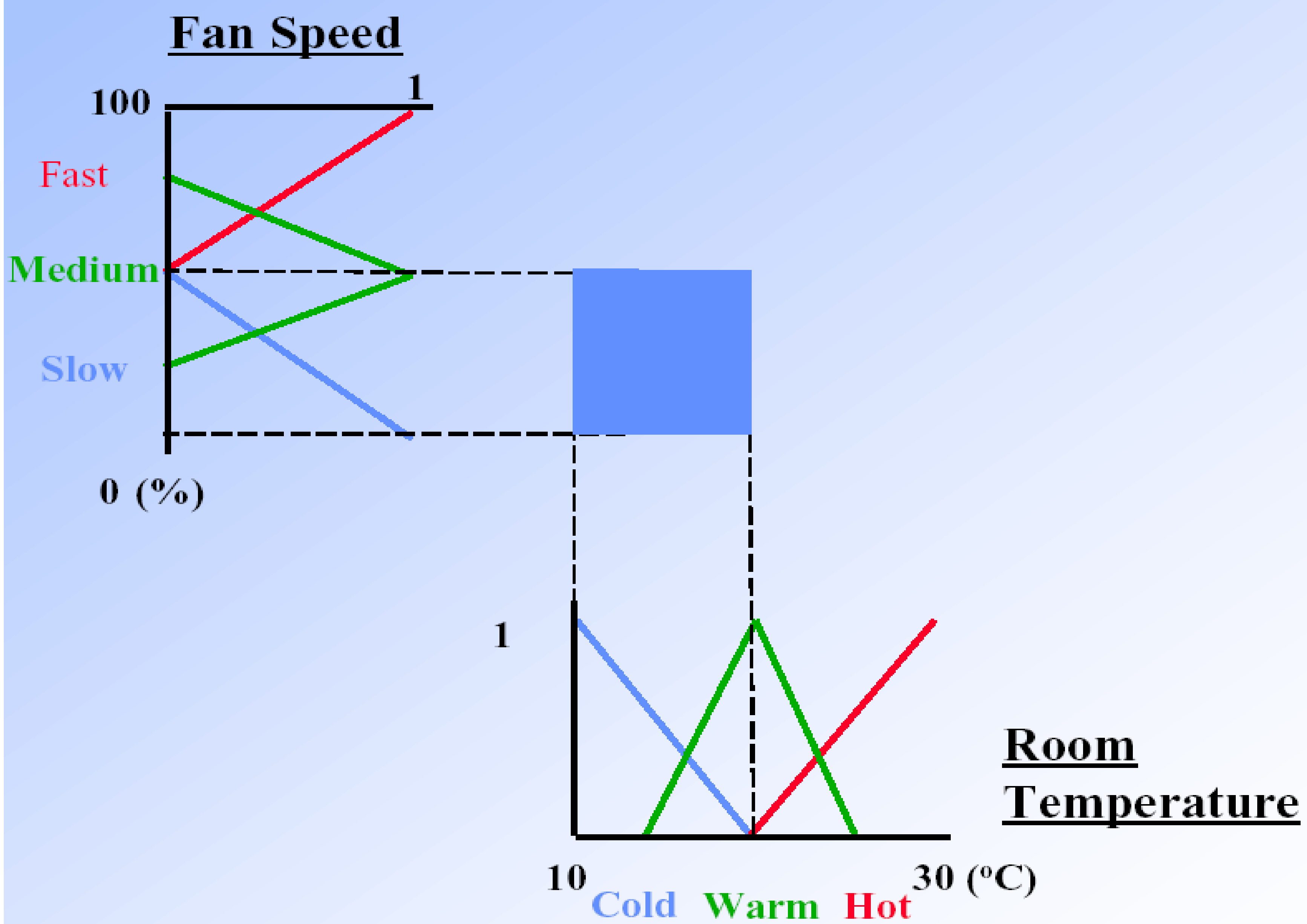


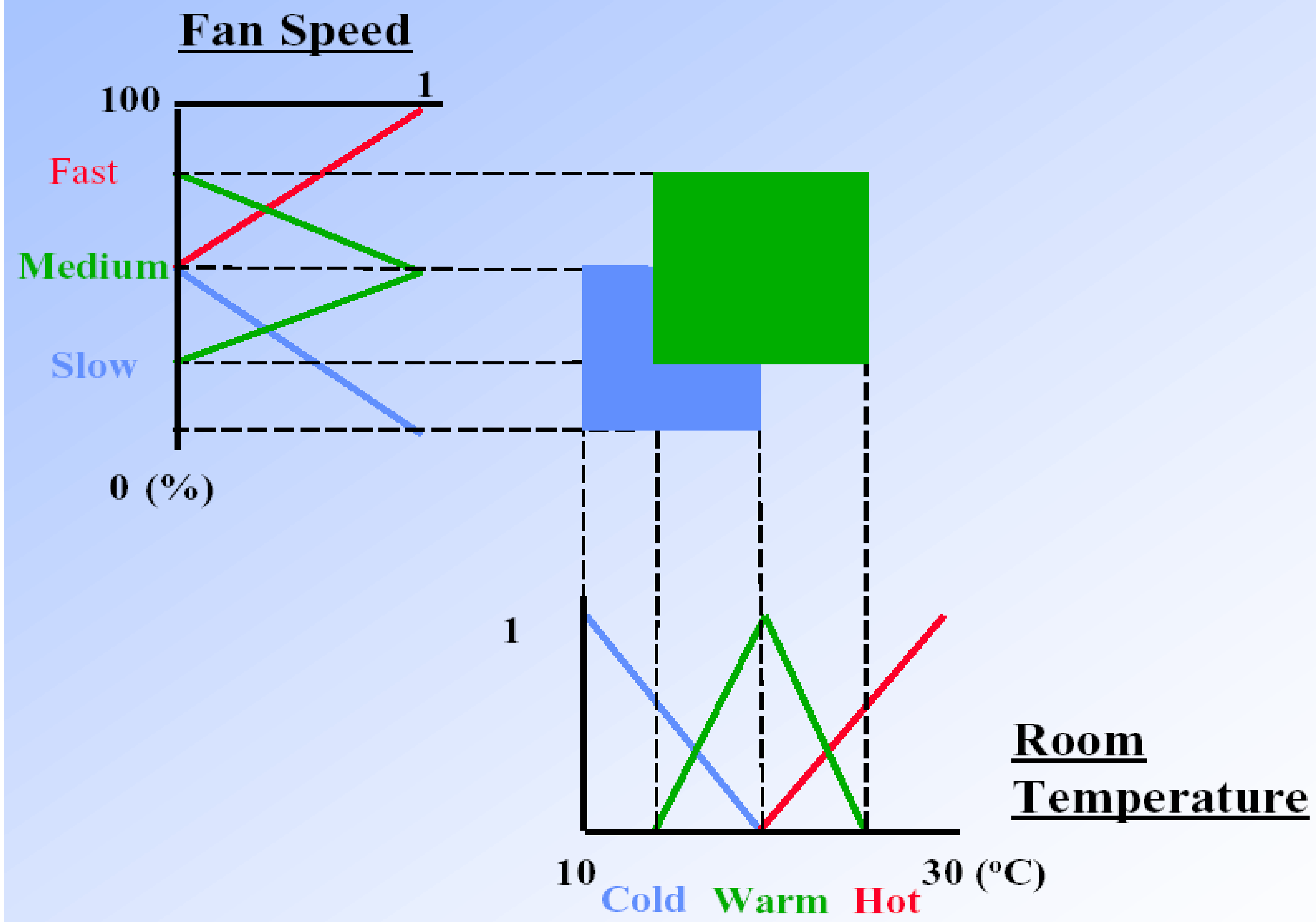
# Fuzzy Temperature Control

- If Room Temperature is **Cold** then Fan Speed is **Slow**
- If Room Temperature is **Warm** then Fan Speed is **Medium**
- If Room Temperature is **Hot** then Fan Speed is **Fast**

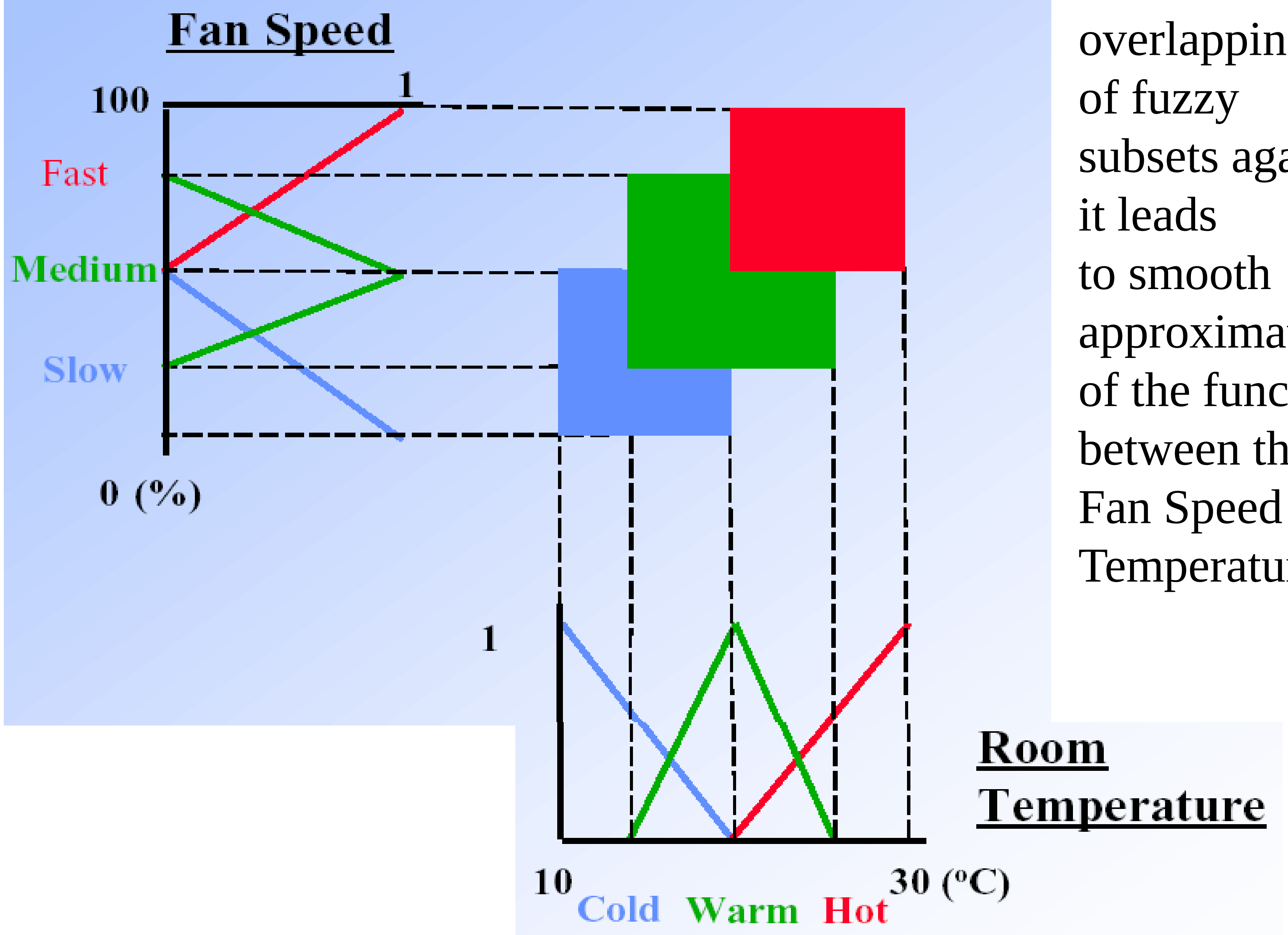
# The Fuzzy Patches



# The Fuzzy Patches

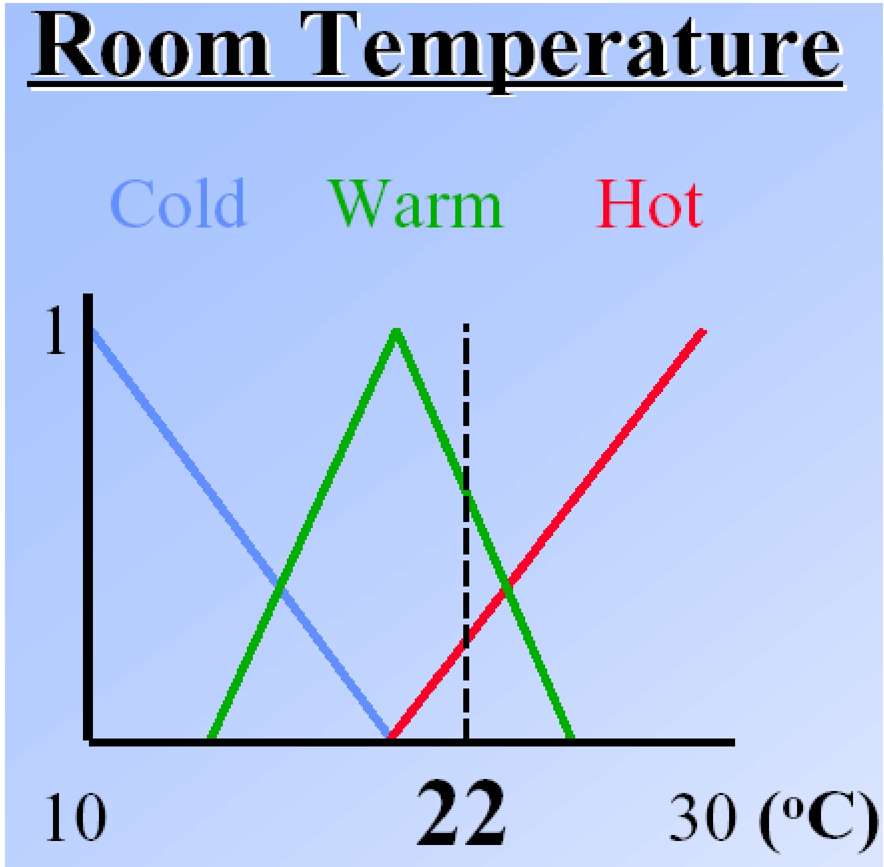


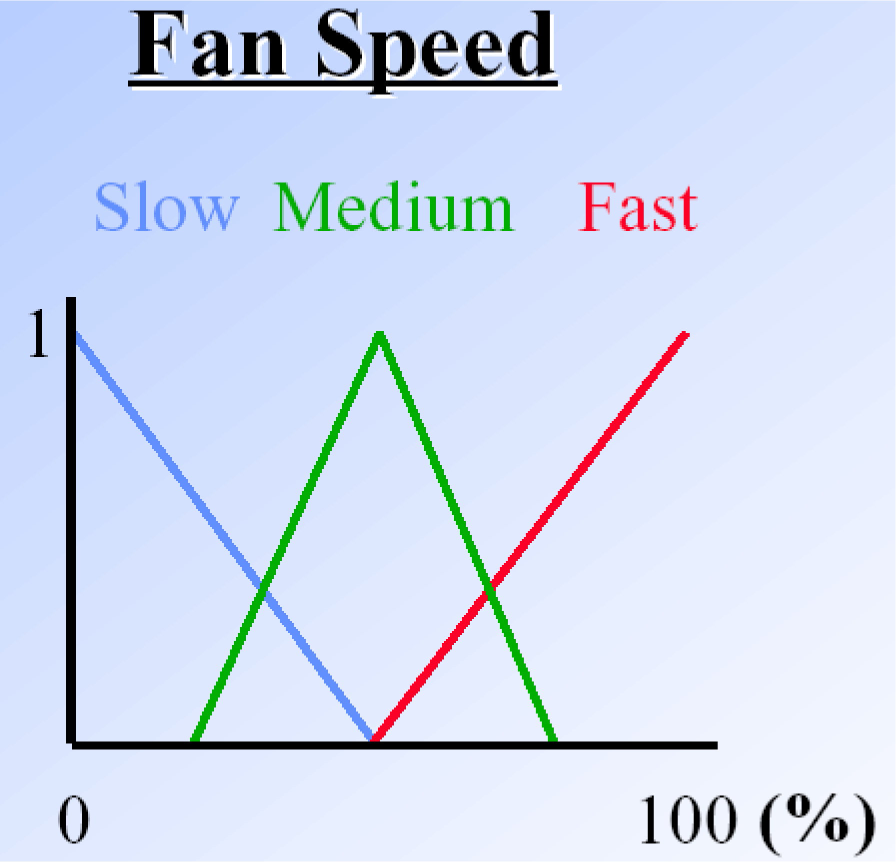
# The Fuzzy Patches




Note the overlapping of fuzzy subsets again - it leads to smooth approximation of the function between the Fan Speed & Temperature

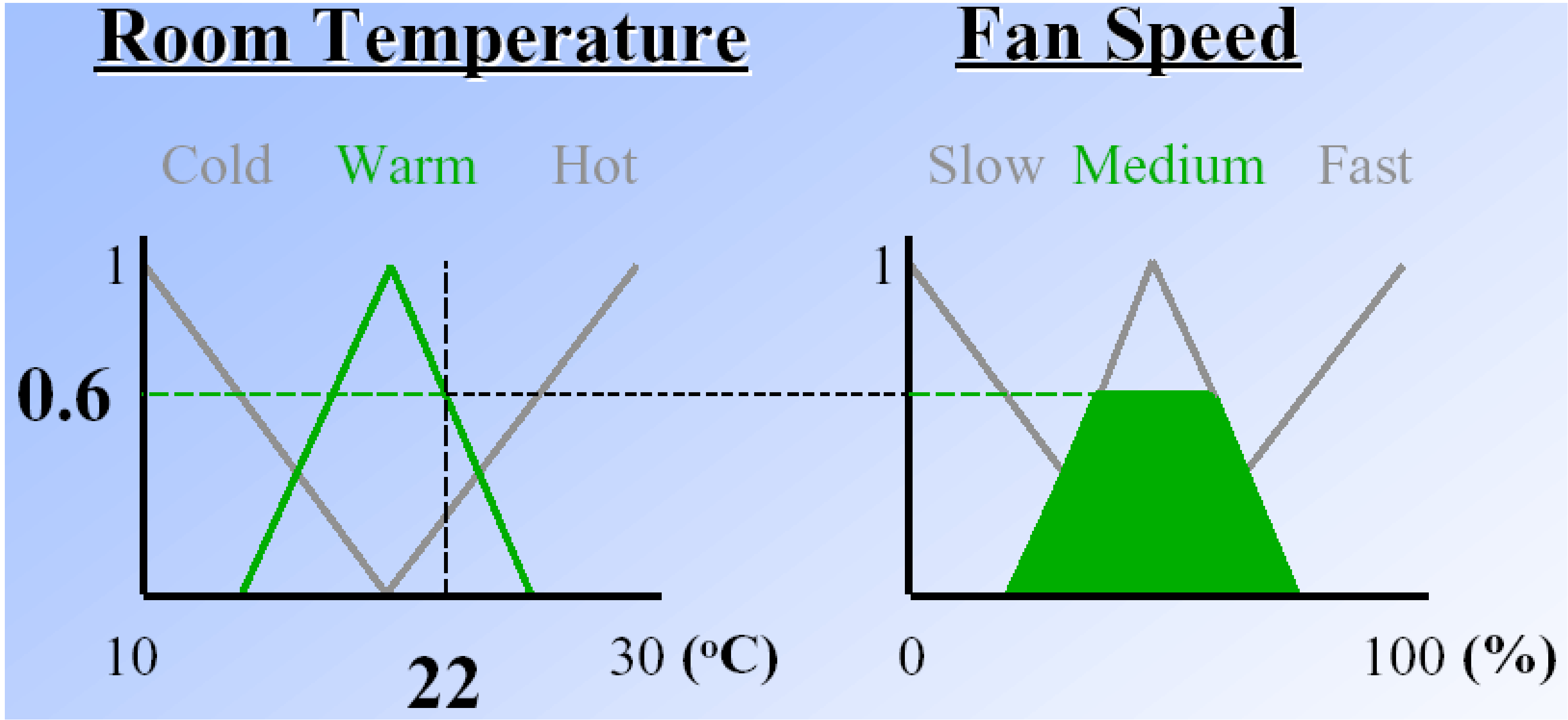
# Calculation of the output:



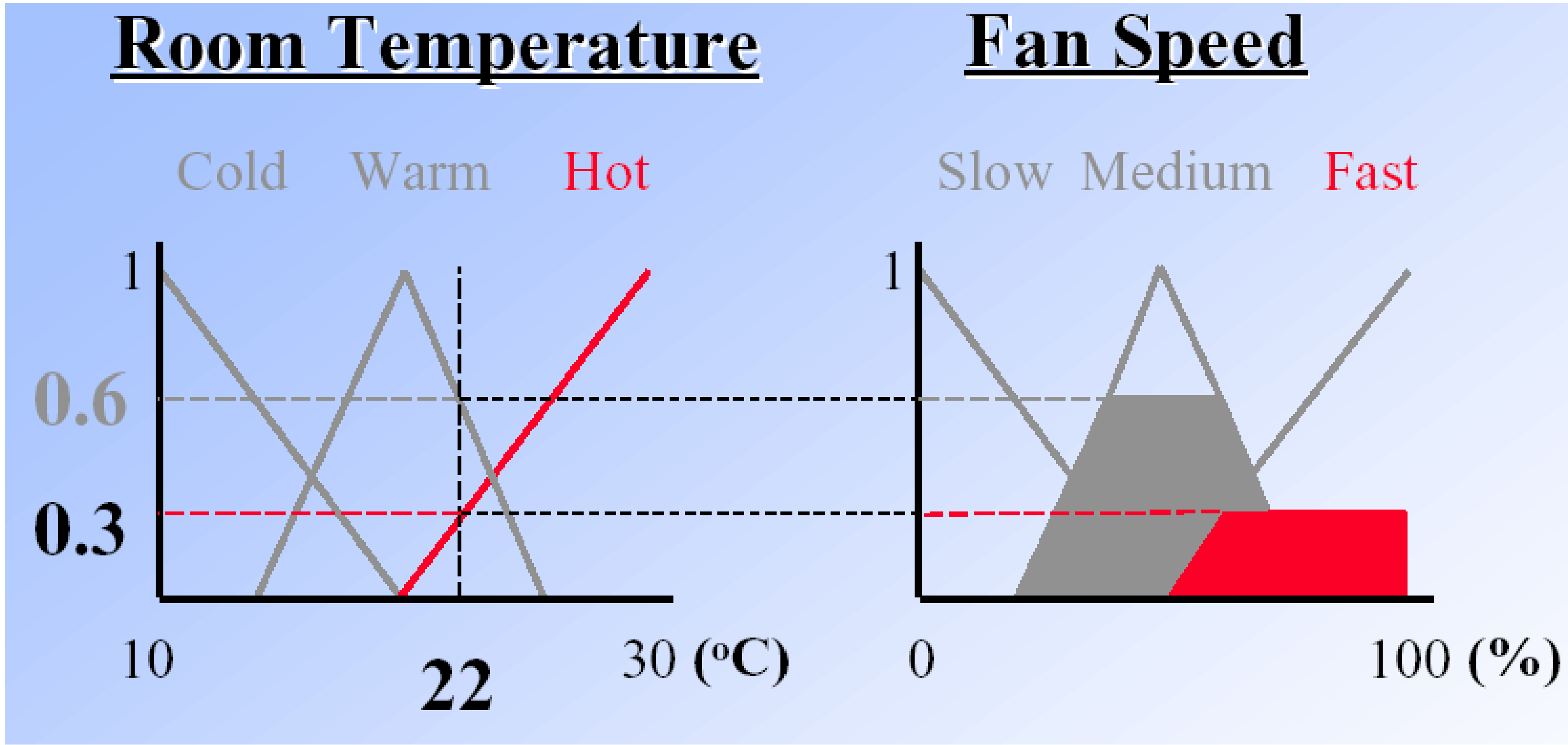

After the fuzzy modeling is done there is an operation phase:
Calculate the Fan Speed when Room Temperature = 22 $^0$C .
NOTE! 22 $^0$C belongs to the Subsets Warm and Hot

# Fuzzification and Inference




If Room Temperature is **Warm** Then Fan Speed is **Medium**

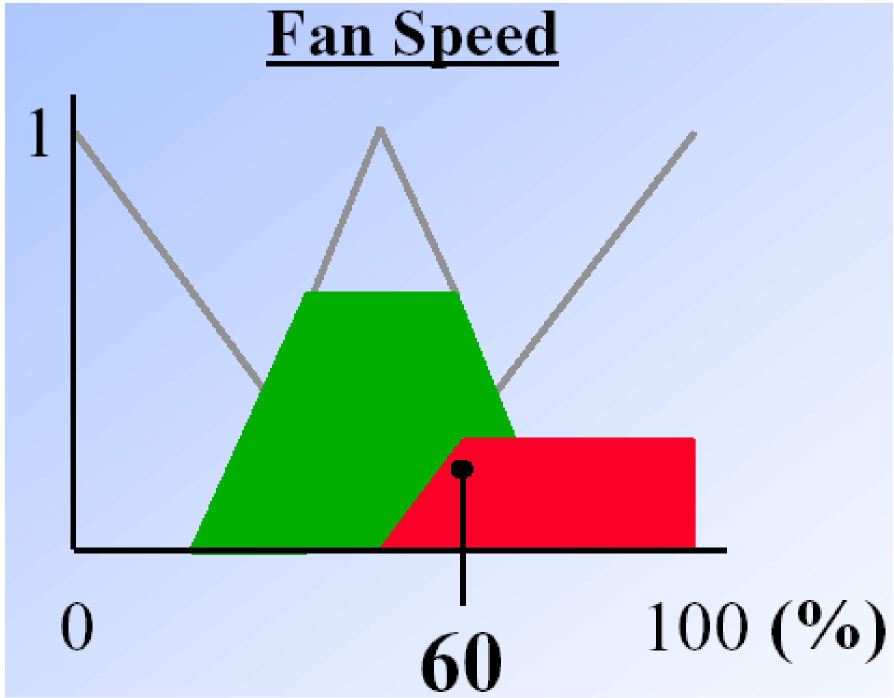
# Fuzzification and Inference




If Room Temperature is **Hot** then Fan Speed is **Fast**

**The question now is: What is the output value?**

# Defuzzification




- The Result of the Fuzzy Inference is a Fuzzy Subset Composed of the Slices of Fan Speed: Medium (green) and Fast (red)

- How to Find a Crisp (for the real world application useful) Value?

- One out of several different methods is to find the Centroid of Area to Obtain a Crisp Output

# Example of Crisp Operations

A, B: sets of annual personal income

$A = \{x \mid 100\text{ K} \le x \le 200\text{ K}, x \in U\}$

$B = \{x \mid 50\text{ K} \le x \le 120\text{ K}, x \in U\}$

where $U = [0, 1000\text{ K}]$ : universe of discourse

Operations:

$A \cup B = \{x \mid 50\text{ K} \le x \le 200\text{ K}, x \in U\}$

$A \cap B = \{x \mid 100\text{ K} \le x \le 120\text{ K}, x \in U\}$

$\overline{A} = \{x \mid 0\text{ K} \le x < 100\text{ K} \text{ or } 200\text{ K} < x \le 1000\text{ K}, x \in U\}$

# Fuzzy Sets Notation

- A notation convention for fuzzy sets when the universe of discourse, X, is discrete

  and finite, is as follows for a fuzzy set A:
- Discrete:

$$\underset{\sim}{A} = \left\{ \frac{\mu_{\underset{\sim}{A}}(x_1)}{x_1} + \frac{\mu_{\underset{\sim}{A}}(x_2)}{x_2} + \cdots \right\} = \left\{ \sum_i \frac{\mu_{\underset{\sim}{A}}(x_i)}{x_i} \right\}$$

- Analogous

$$\underset{\sim}{A} = \left\{ \int \frac{\mu_{\underset{\sim}{A}}(x)}{x} \right\}$$

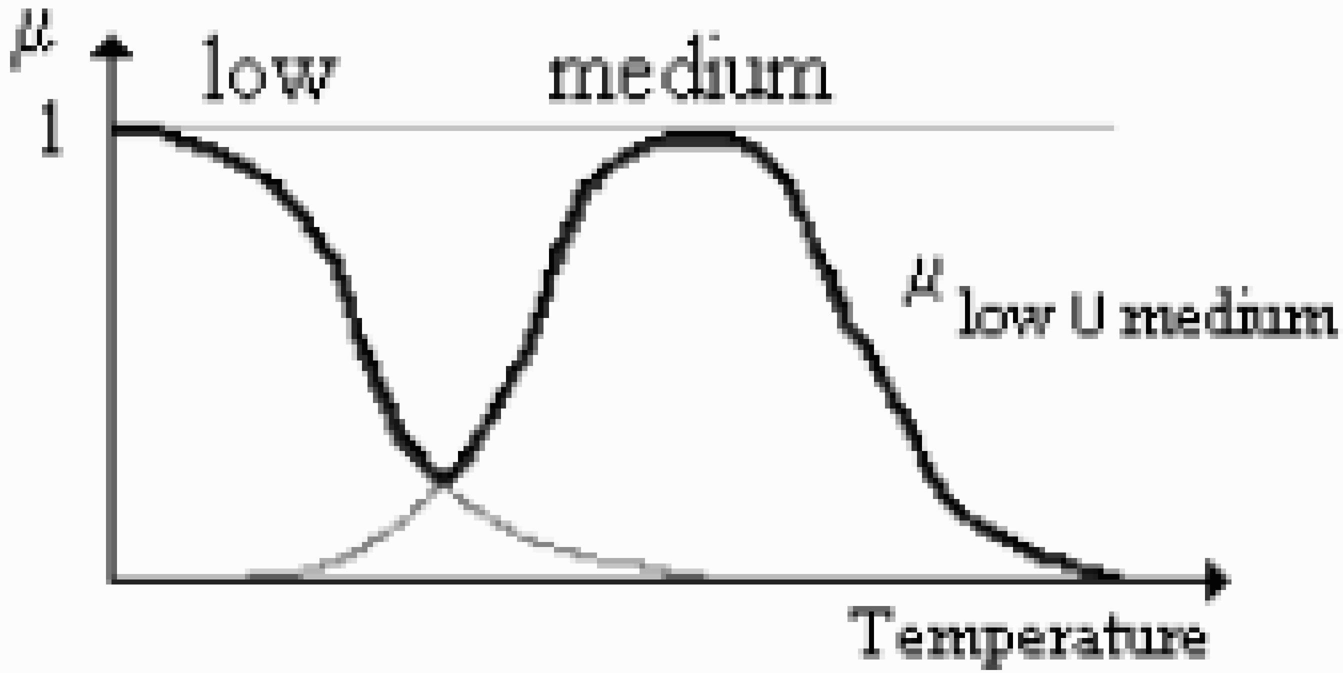
# Fuzzy Sets Operations

| | |
|---|---|
| *Union* | $\mu_{\underset{\sim}{A} \cup \underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \vee \mu_{\underset{\sim}{B}}(x)$ |
| *Intersection* | $\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x)$ |
| *Complement* | $\mu_{\overline{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{A}}(x)$ |

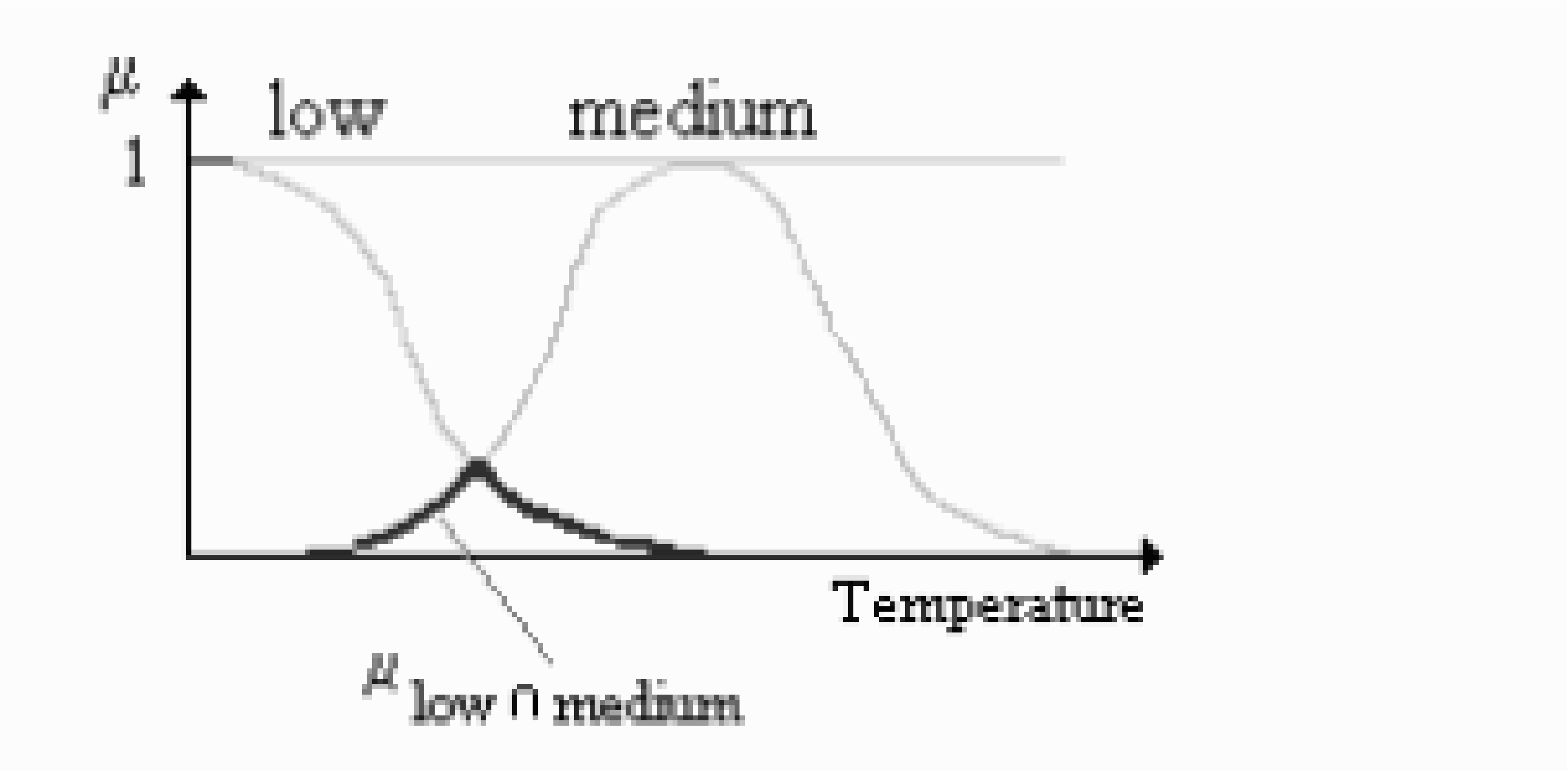
# Fuzzy Union

Defined as the maximum operator:

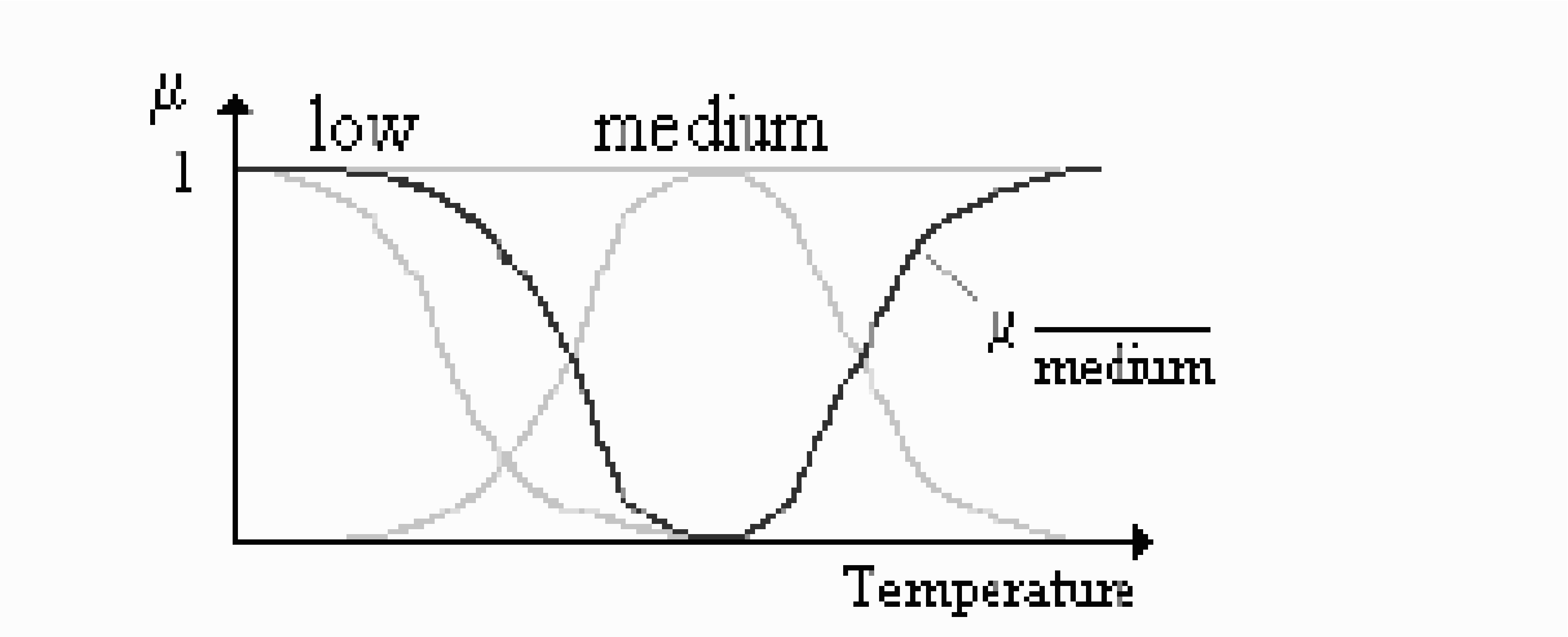
$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$



# Fuzzy Intersection (Conjunction)



# Fuzzy Complement



# Fuzzy Sets

- Fuzzy set theory is a generalization of classical set theory – a generalization that deals excellently with imprecision.

- The power of fuzzy logic is that it enables you to accurately describe a process or behavior without using mathematics.

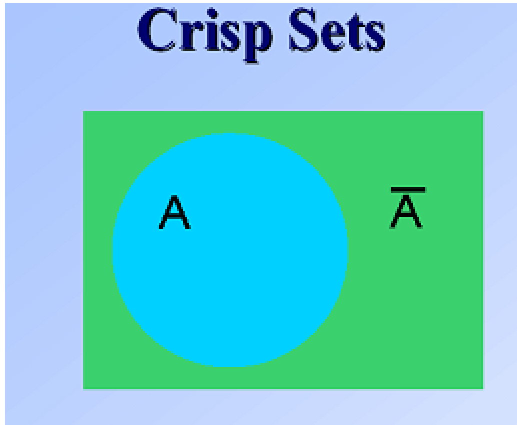
# Fuzzy Sets

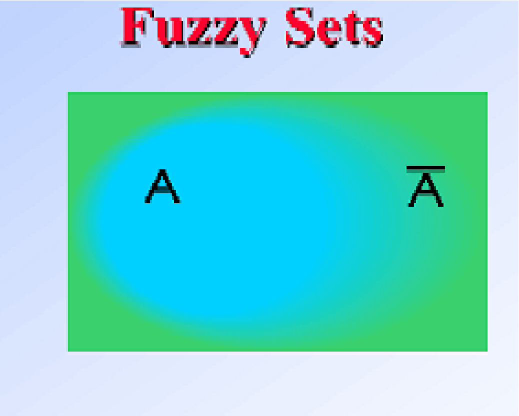
- Classical set theory: An object is either in or not in the set.
  - Can’t talk about non-sharp distinctions

- Fuzzy set theory
  - An object is in a set by matter of degree
    - membership $h = 1.0 \Rightarrow$ in the set
    - membership $h = 0.0 \Rightarrow$ not in the set
    - $0.0 <$ membership $h < 1.0 \Rightarrow$ partially in the set

- Fuzzy sets have a smooth boundary
  - Not completely in or out

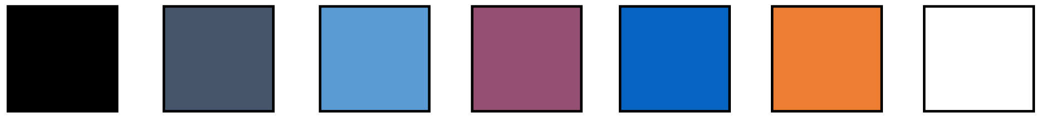
# Crisp Sets and Fuzzy Sets



# Fuzzy Sets

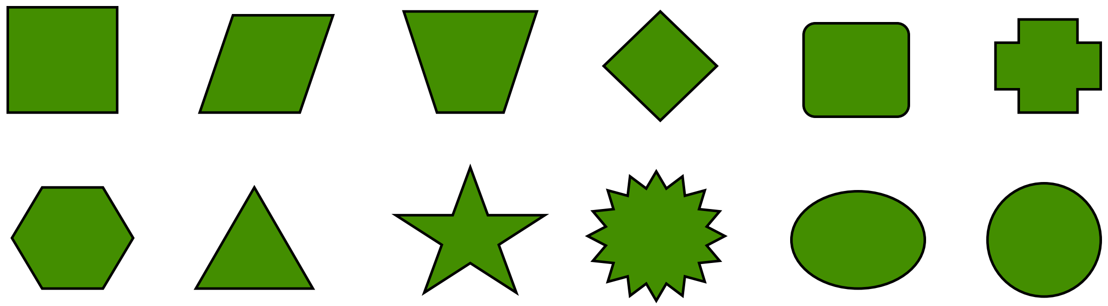
**Shade of Gray**



1 0.8 0.7 0.6 0.5 0.2 0

**Degree of being round** (roundness)

0.5 .6 0.65 0.5 0.65 0.4



0.7 0.55 0.4 0.8 0.9 1.0

# Fuzzy Sets

The basic idea of the *fuzzy set theory* is that an element belongs to a fuzzy set with a certain degree of membership, with fuzzy boundaries

A proposition is neither true nor false, but may be partly true (or partly false) to any degree. This degree is usually taken as a real number in the interval [0,1]

Fuzzy logic is an extension of classic two-valued logic – the truth value of a sentence is not restricted to true or false.

# Fuzzy Sets

- A linguistic variable is a fuzzy variable.

  For example, the statement "John is tall" implies that the linguistic variable *John* takes the linguistic value *tall*.

- The range of possible values of a linguistic variable represents the universe of discourse of that variable.

  For example, the universe of discourse of the linguistic variable *speed* might have the range between 0 and 220 km/h and may include such fuzzy subsets as *very slow*, *slow*, *medium*, *fast*, and *very fast*.

# Fuzzy Sets

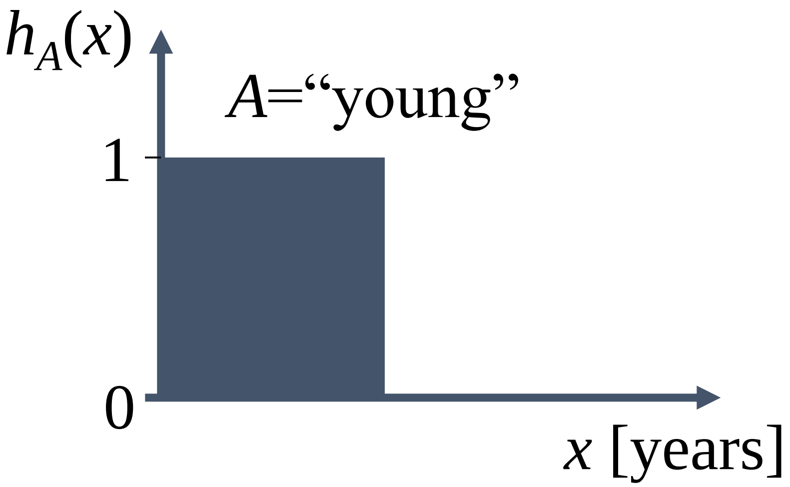
- A fuzzy set has a membership function that allows various degrees of membership for the elements of a given set.
- The membership function may be defined in terms of discrete values, or more commonly by a graph.
- When membership function is described by an analytic expression, we can just use the membership function to describe the fuzzy subset.

# Fuzzy Sets

**_Classical Logic_**

Element $x$ belongs to set $A$ or it does not:

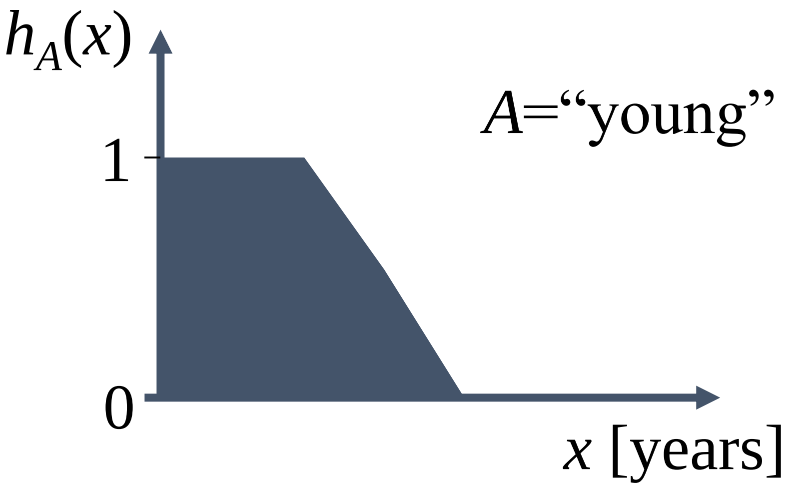
$$h(x) \in \{0,1\}$$




**_Fuzzy Logic_**

Element $x$ belongs to set $A$ with a certain ***degree of membership*****:**

$$h(x) \in [0,1]$$



# Example: Crisp set Tall

- Fuzzy sets and concepts are commonly used in natural language

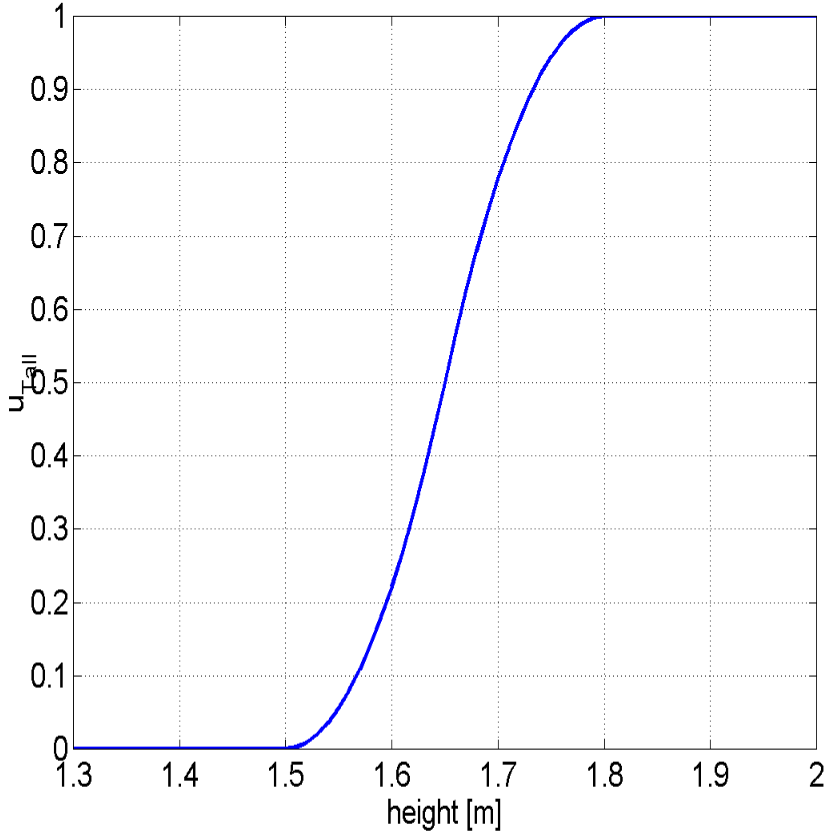
  *John is **tall***
  *Dan is **smart***
  *Alex is **happy***
  *The class is **hot***

- The crisp set ***Tall*** can be defined as $\{x \mid \text{height } x > 1.8$ meters$\}$. But what about a person with a height = 1.79 meters? What about 1.78 meters? … What about 1.52 meters?

# Example: Fuzzy set Tall

- In a fuzzy set a person with a height of 1.8 meters would be considered tall to a **high degree.**
- A person with a height of 1.7 meters would be considered tall to
  a lesser degree etc.

- The function can change for basketball players, women, children etc.



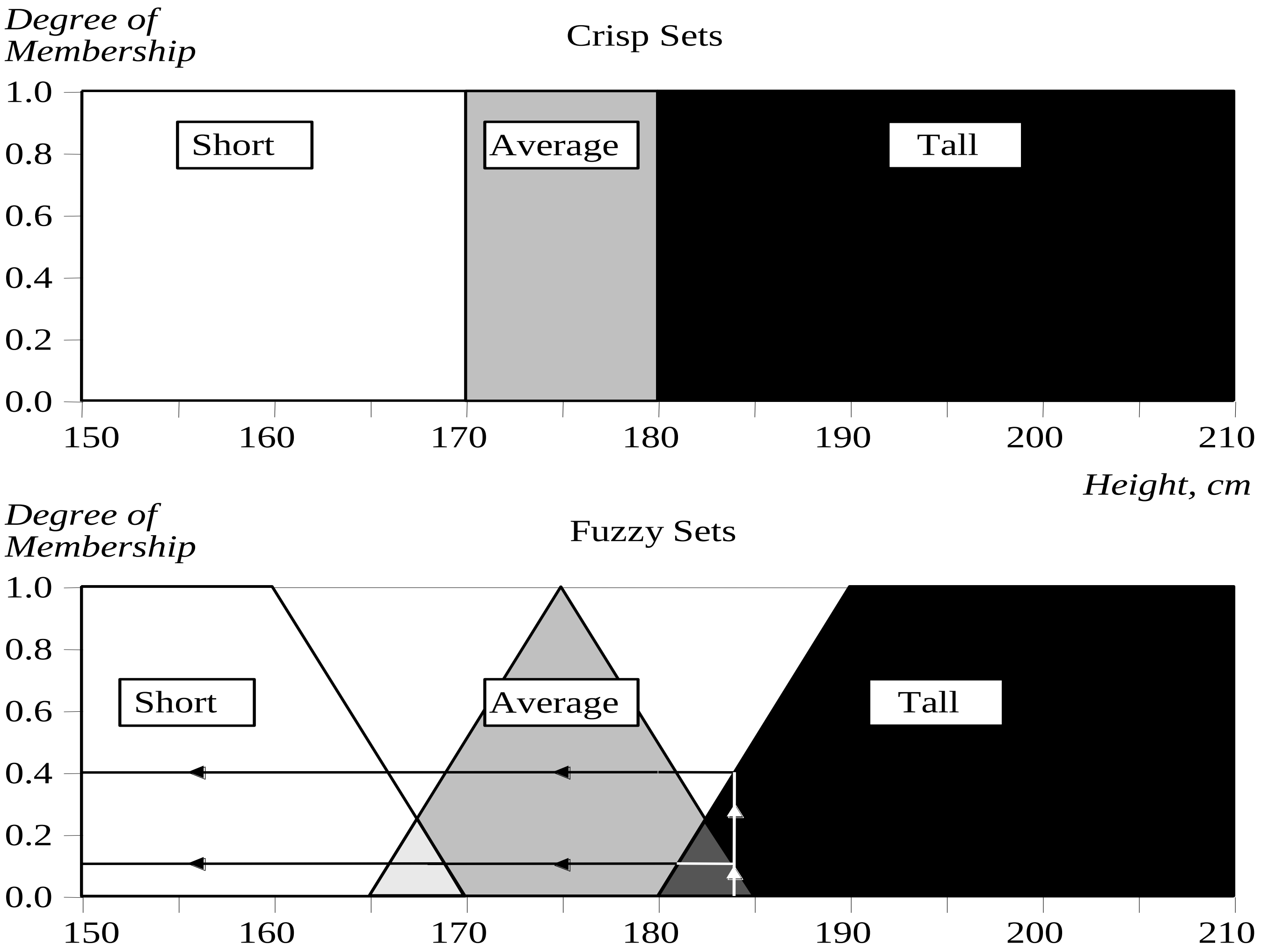
# Fuzzy Sets

- The *x*-axis represents the universe of discourse – the range of all possible values applicable to a chosen variable.

- The *y*-axis represents the membership value of the fuzzy set.

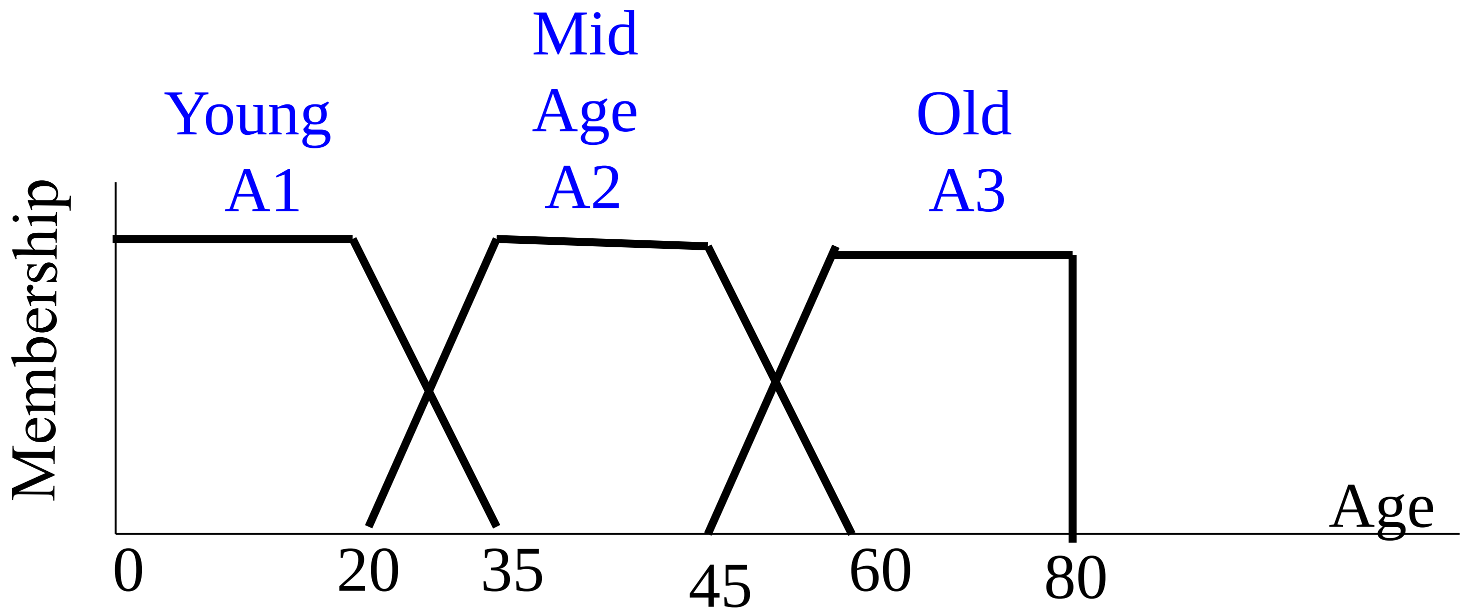
# Fuzzy Sets

- crisp and fuzzy subsets defined on the universe



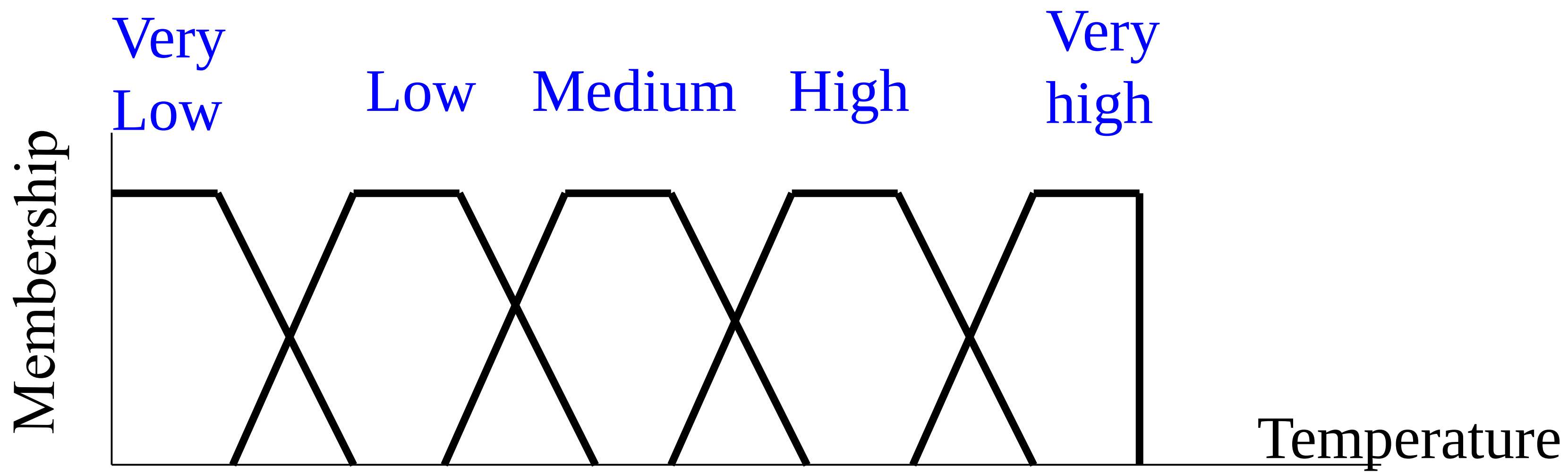
# Fuzzy Set

- Fuzzy sets : young, middle aged and old

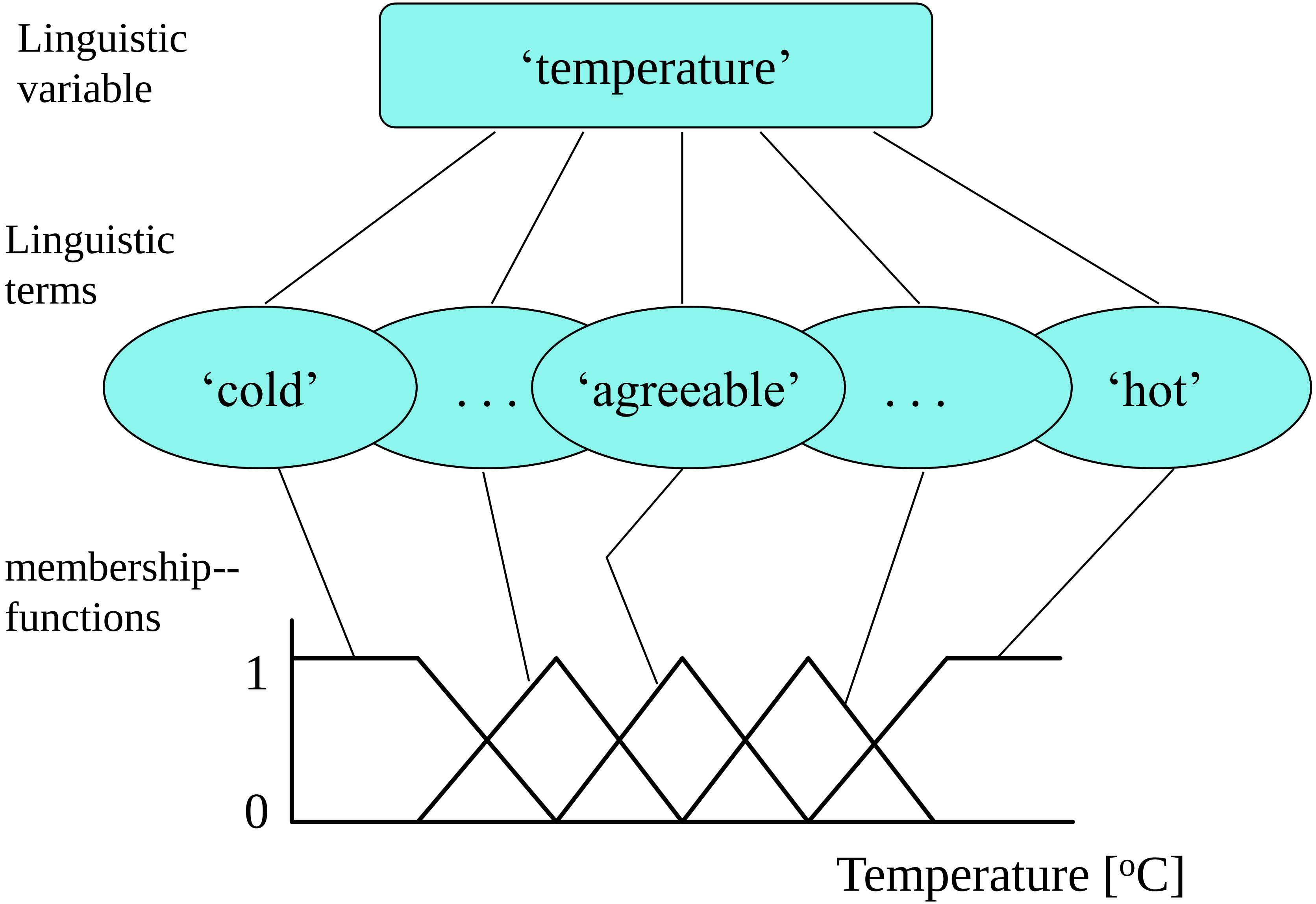


# Fuzzy Sets

- Variables whose states are defined by linguistic concepts like *low*, *medium*, *high*.
- These linguistic concepts are fuzzy sets themselves.




Trapezoidal membership functions


Linguistic variable
'temperature'
Linguistic terms
'cold'
. . .
'agreeable'
. . .
'hot'
membership--functions
1
0
Temperature [°C]

# Fuzzy Sets

- A fuzzy subset has a membership function that allows various degrees of membership for the elements of a given set.

- The membership function may be defined in terms of discrete values, or more commonly by a graph.

## Fuzzy Sets

If $X$ is the *universe of discourse* with element $x$, then a *fuzzy subset*, denoted by $\underline{A}$, on $X$ is a set of ordered pairs such that

$$\underline{A} \equiv \{(x, h_{\underline{A}}(x)) | \; x \in X\}$$

where $h_{\underline{A}}(x)$ is the *membership function* of $x$ in $\underline{A}$ and is defined by $h_{\underline{A}}: X \rightarrow [0, 1]$

Elements with zero membership are usually not listed.

## Fuzzy Sets

- If the universe of discourse $X$ is discrete and finite, $X = \{x_1, x_2, \ldots, x_n\}$, there are three different ways to describe the fuzzy subset $\underline{A}$.

  (a) $\underline{A} = h_{\underline{A}}(x_1)/x_1 + h_{\underline{A}}(x_2)/x_2 + \ldots + h_{\underline{A}}(x_n)/x_n$

  (b) $\underline{A} = \{(x_1, h_{\underline{A}}(x_1)), (x_2, h_{\underline{A}}(x_2)), \ldots, (x_n, h_{\underline{A}}(x_n))\}$

  (c) $\underline{A} = \{h_{\underline{A}}(x_1), h_{\underline{A}}(x_2), \ldots, h_{\underline{A}}(x_n)\}$

- If the universe of discourse $X$ is continuous, we can represent the fuzzy subset $\underline{A}$ by

$$\underline{A} \equiv \int_{x \in X} [h_{\underline{A}}(x)/x]$$

# Fuzzy Set

- Usefulness of fuzzy sets depends on our capability to construct appropriate membership functions for various given concepts in various contexts.

- Constructing meaningful membership functions is a difficult problem.

# Fuzzification

# Fuzzification

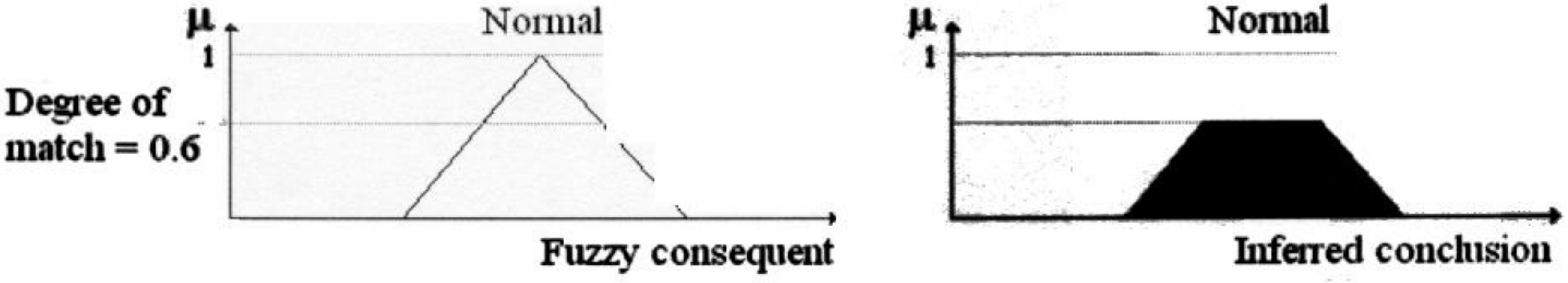
The process of *fuzzification* transforms a set (fuzzy or crisp) to an approximating set that is more fuzzy.

This process is a generation of the dilation operation.

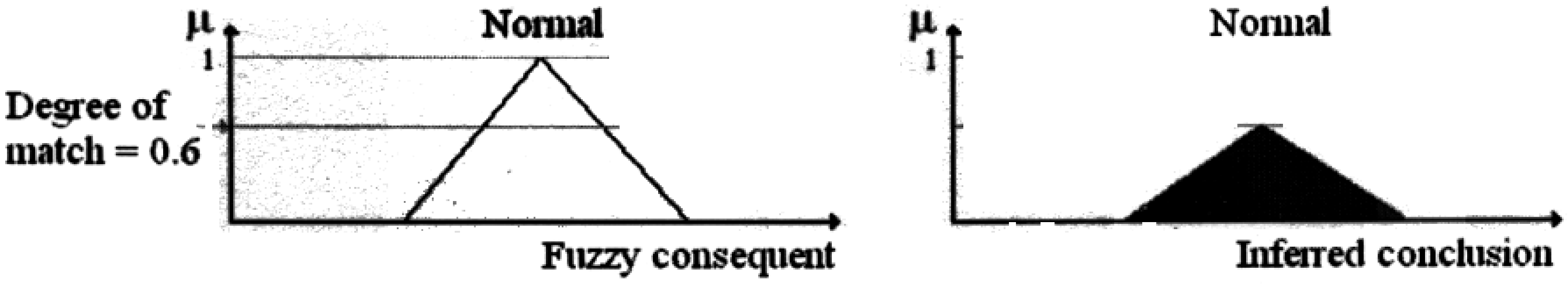
The essence of the fuzzification process is *point fuzzification*. Point fuzzification transforms a singleton set $\{1/u\}$ in $X$ to a fuzzy set that varies around $u$.

The symbol $\sim$ is used to denote a *fuzzifier*.

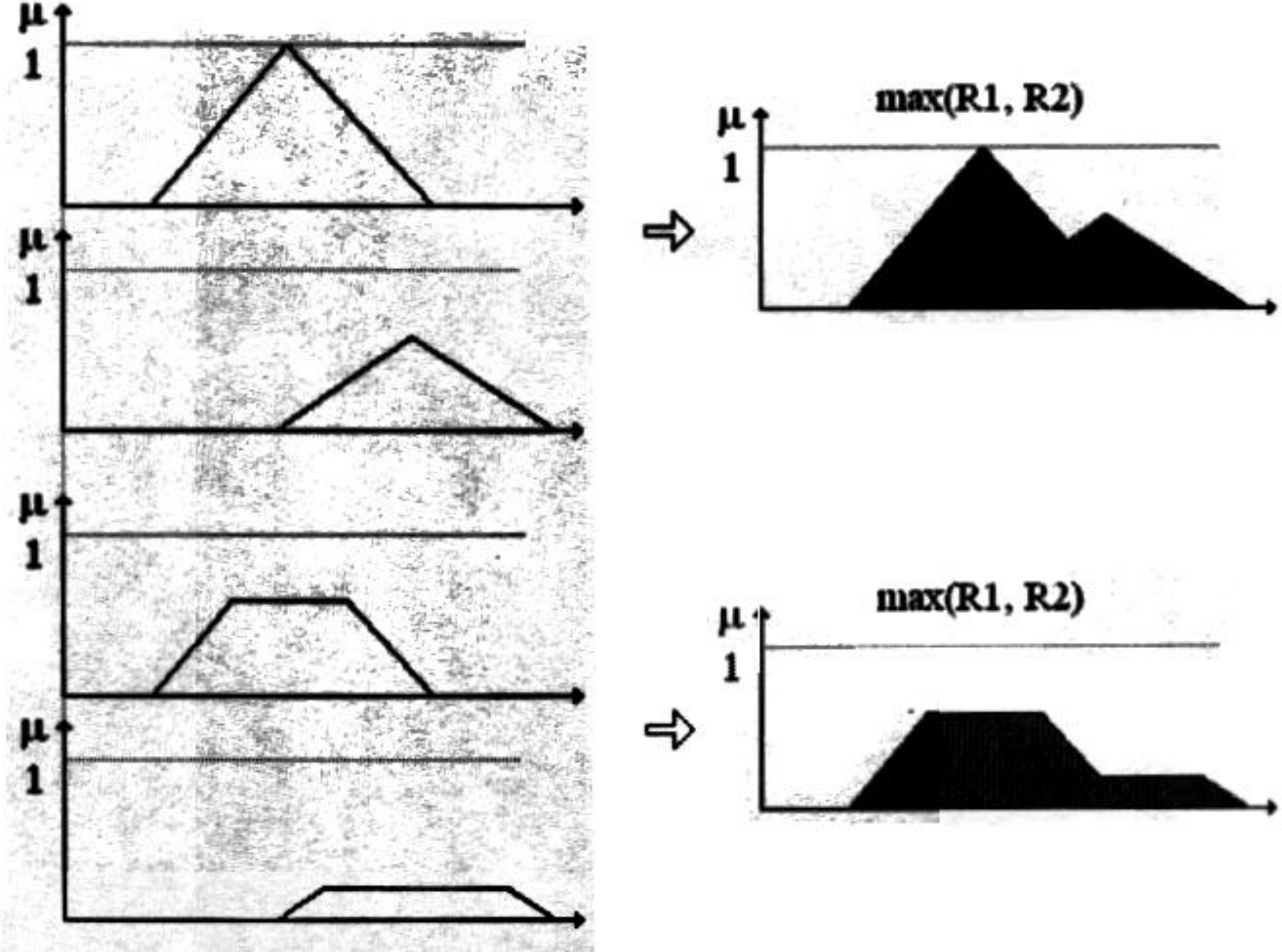
# Clipping Method



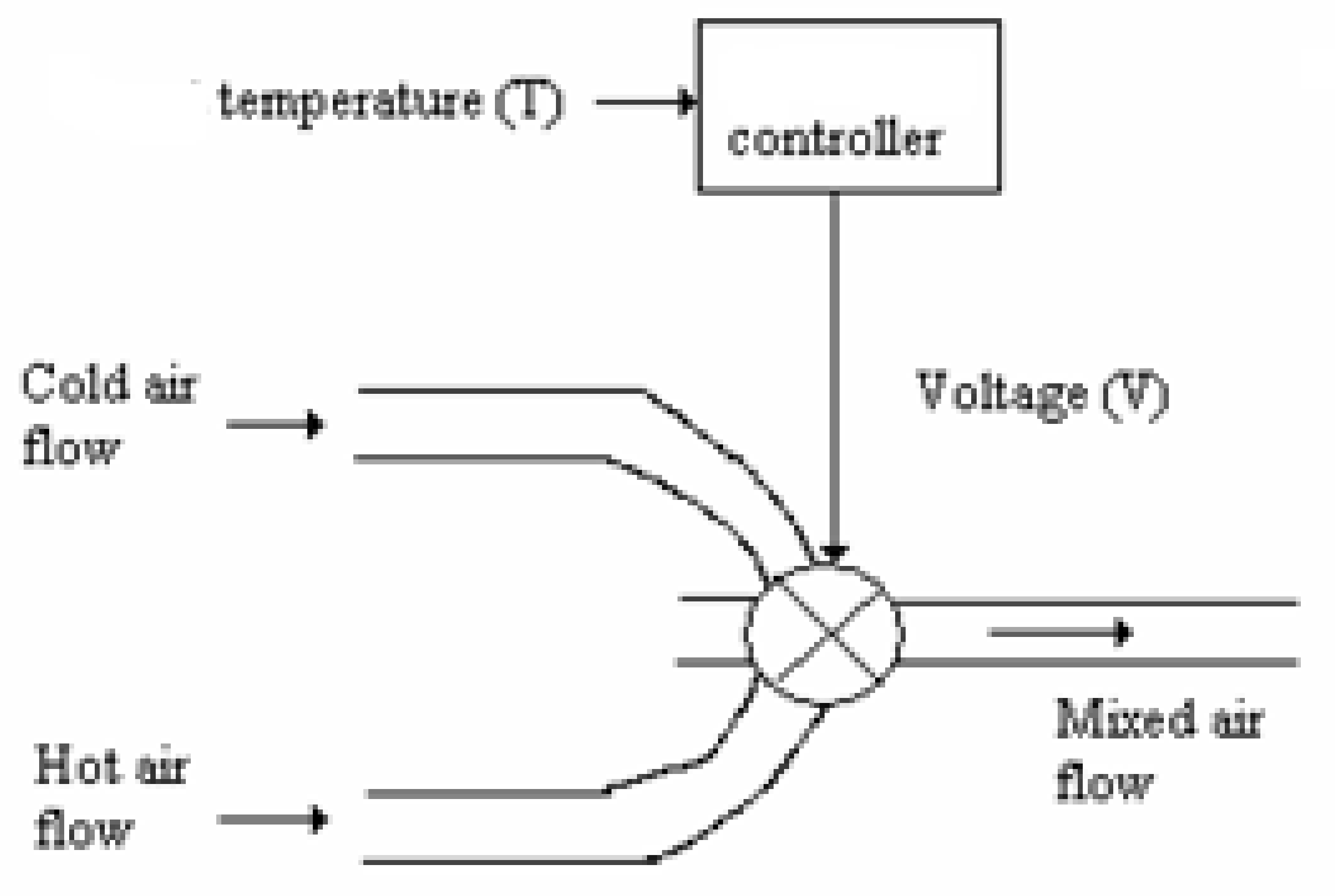
# Scaling Method



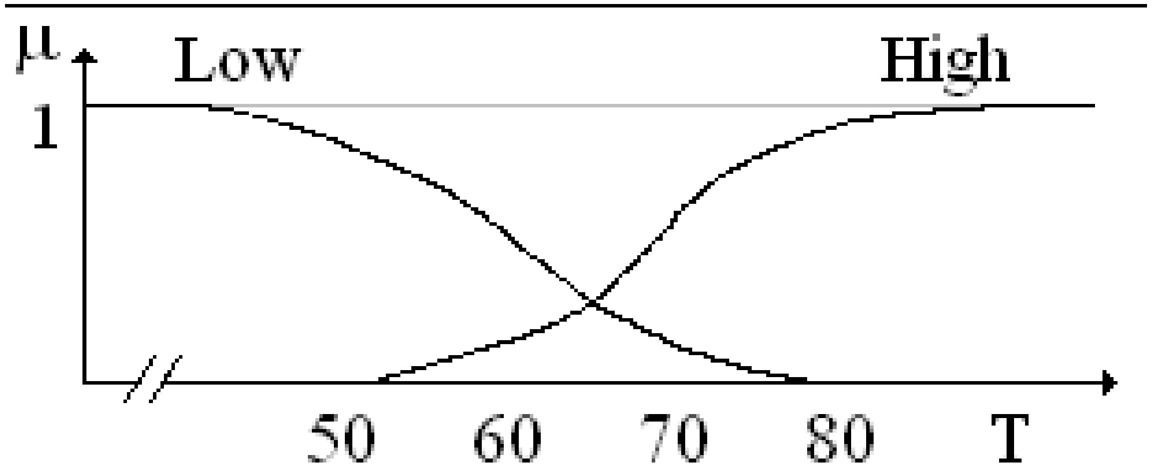
# Combining Fuzzy Conclusions

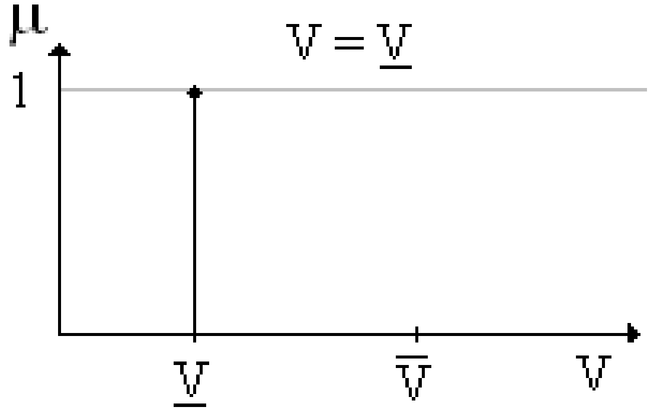


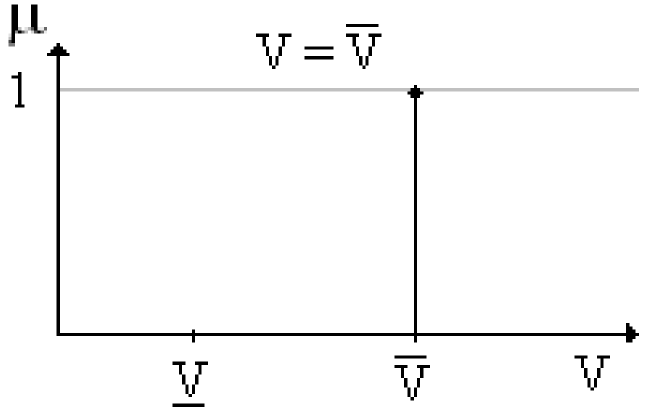
# Example: Air Flow Mixing System for Room Temperature Control



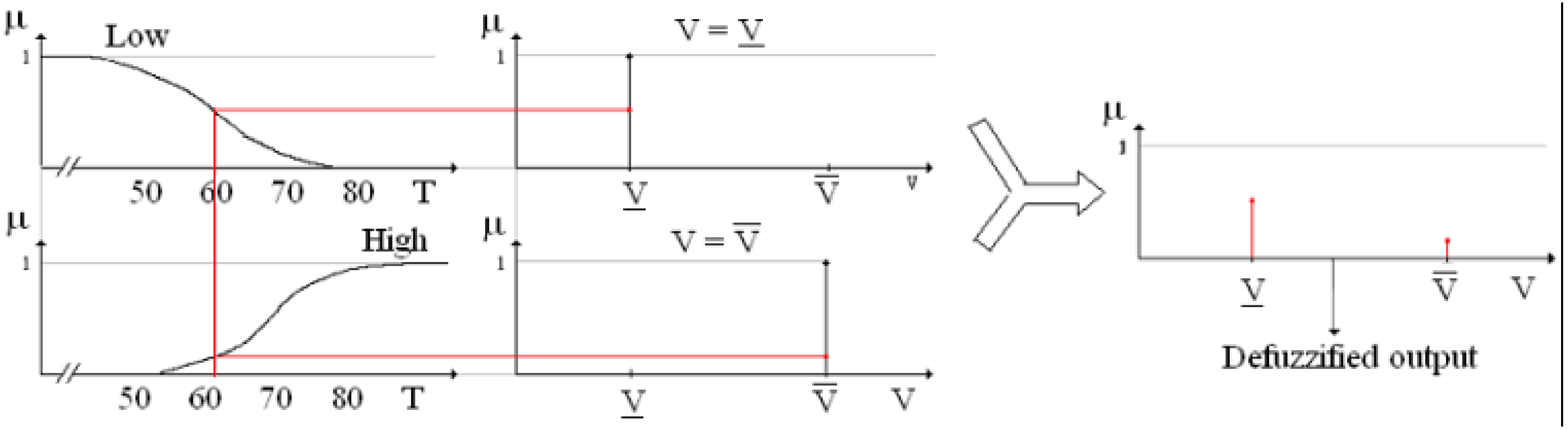
# Input and Output Membership Functions



# Defuzzification

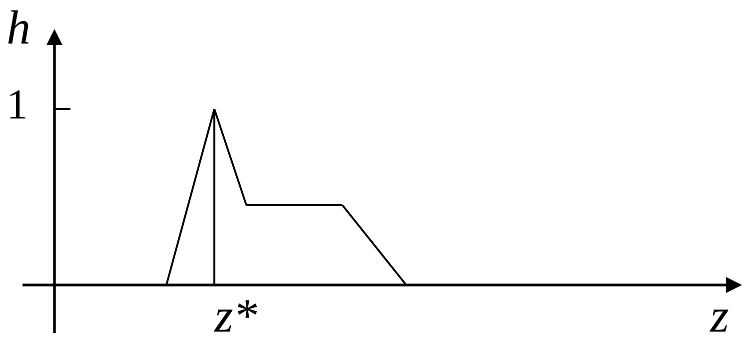


# Defuzzification

## Max-membership principle -

Also known as the height method, this scheme is limited to peaked output functions. This method is given by the algebraic expression

$$h_{\underset{\sim}{C}}(z^*) \geq h_{\underset{\sim}{C}}(z) \qquad \forall\, z \in Z$$

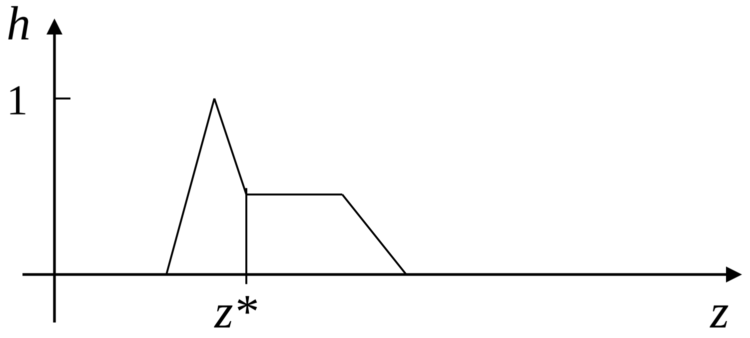


# Defuzzification

## Centroid method -

This procedure (also known as center of gravity) is the most prevalent and physically appealing of all the defuzzification method and is given by the algebraic expression

$$z^* = \frac{\int h_{\underset{\sim}{C}}(z) z\,dz}{\int h_{\underset{\sim}{C}}(z)\,dz} \qquad \forall\ z \in Z$$



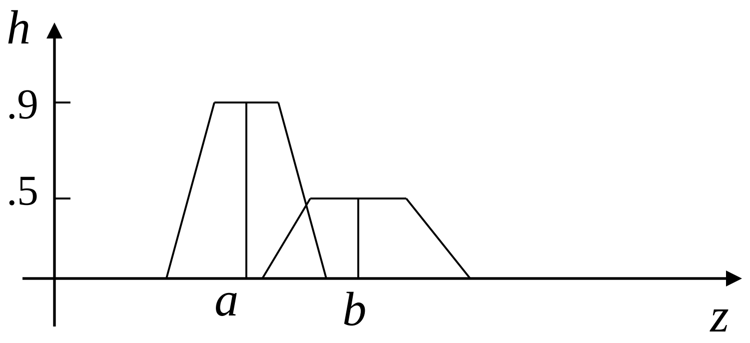
# Defuzzification

## Weighted average method -

This method is only valid for *symmetrical* output membership functions. This is given by the algebraic expression

$$z^* = \frac{\sum h_{\underset{\sim}{C}}(\bar{z})\bar{z}}{\sum h_{\underset{\sim}{C}}(\bar{z})}$$

This method is formed by weighting each membership function in the output by its respective maximum membership value.



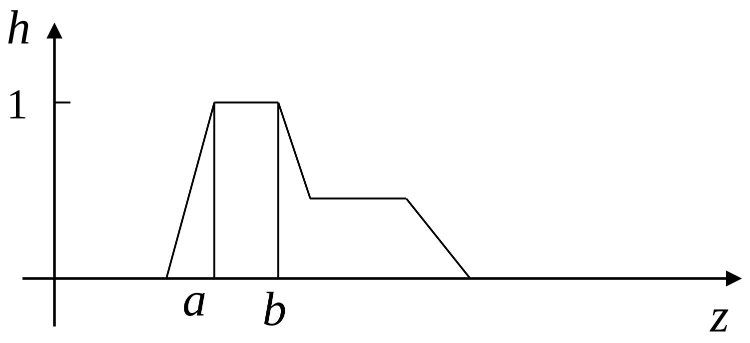

$$z^* = \frac{a(0.5) + b(0.9)}{0.5 + 0.9}$$

# Defuzzification

## Mean-max membership -

This method (also called middle-of-maxima) is closely related to the max-membership principle, except that the locations of the maximum membership can be non-unique. This is given by the algebraic expression

$$z^* = \frac{a+b}{2}$$



# Defuzzification

## Center of Sums -

This method is faster than many defuzzification methods that are presently in used. This process involves the algebraic sum of individual output fuzzy sets, say $\underline{C}_1$ and $\underline{C}_2$, instead of their union. This is given by the algebraic expression

$$z^* = \frac{\int z[\sum_{k=1}^{n} h_{\underline{C}_k}(z)]dz}{\int [\sum_{k=1}^{n} h_{\underline{C}_k}(z)]dz}$$

One drawback to this method ls that the intersecting areas are added twice.

# Defuzzification

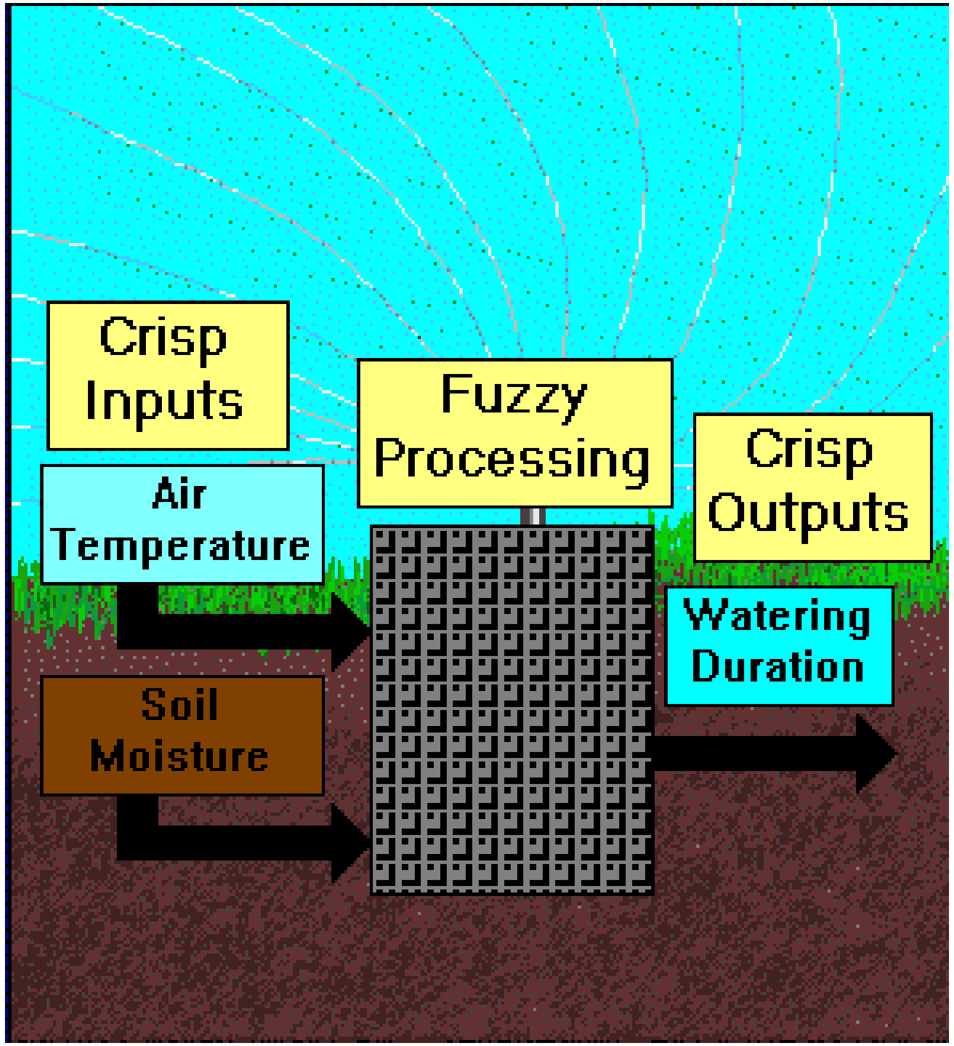
## Center of largest area

If the output fuzzy set has at least two convex subregions, then the center of gravity of the convex fuzzy subregion with the largest area is used to obtain the defuzzified value z* of the output. This is given by the algebraic expression

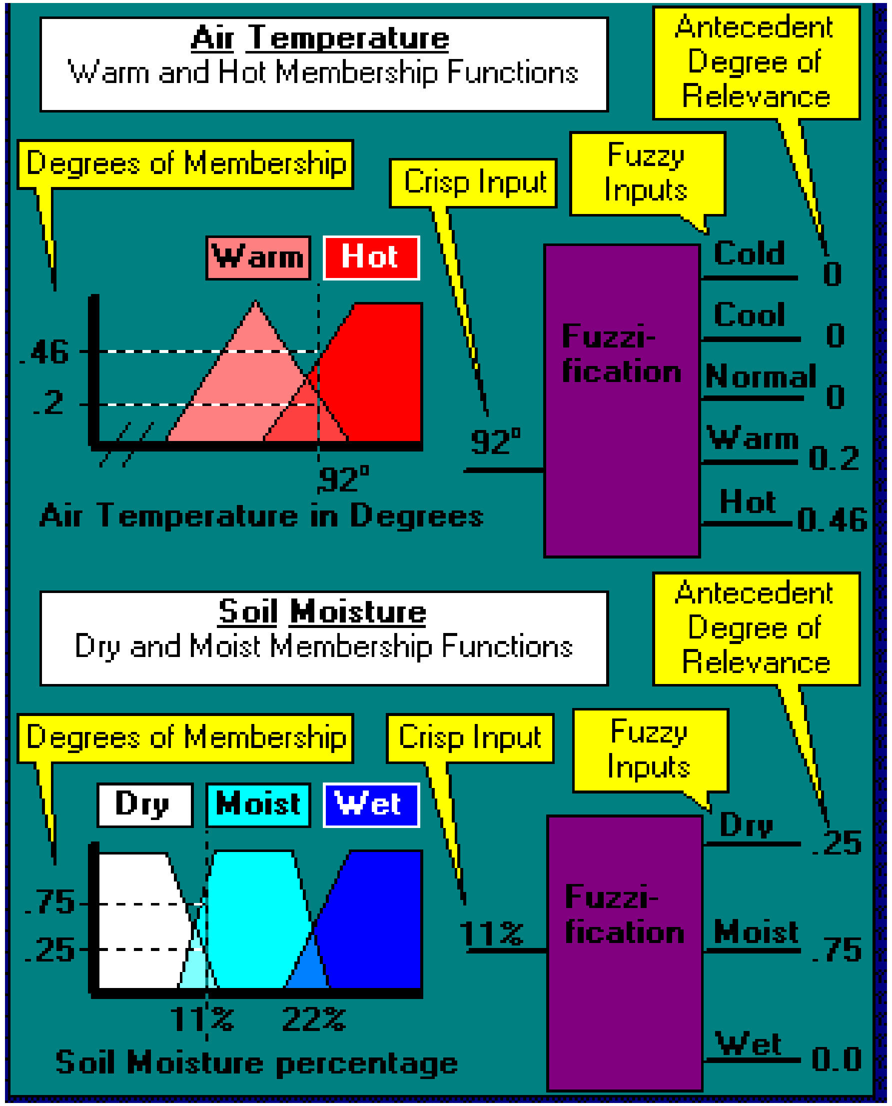
$$z^* = \frac{\int z h_{\underset{\sim}{C}_m}(z)\,dz}{\int h_{\underset{\sim}{C}_m}(z)\,dz}$$

where $\underline{C}_m$ is the convex subregion that has the largest area making up $\underline{C}_k$.

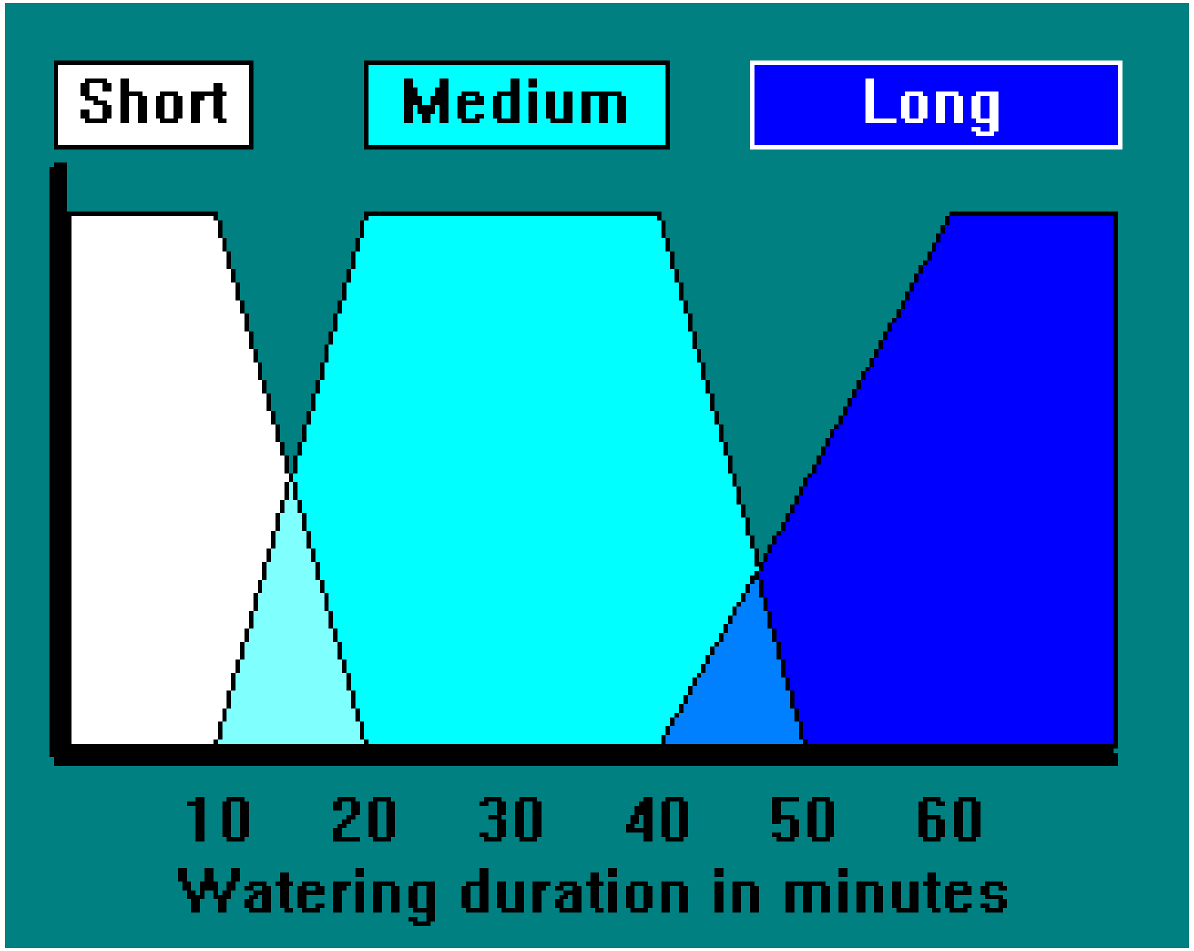
# Fuzzy Logic Application



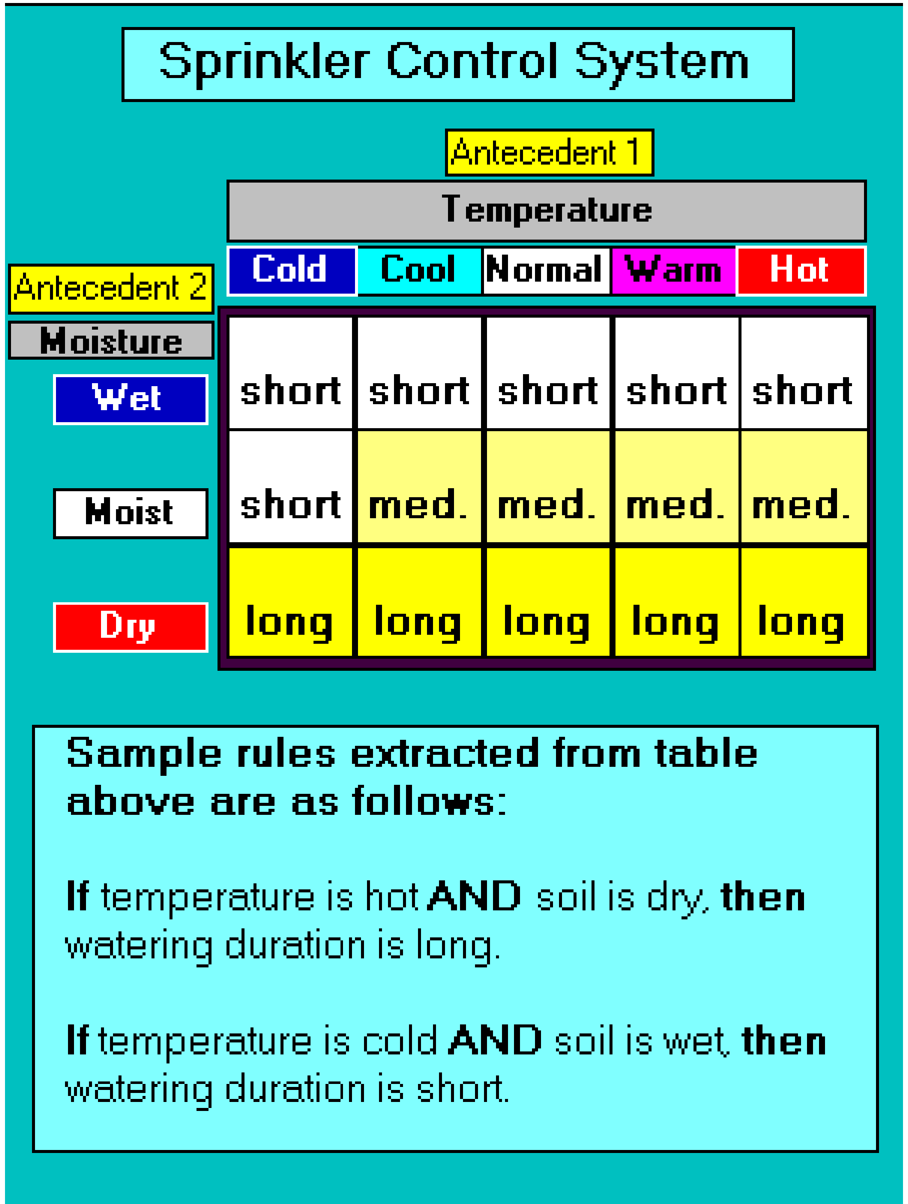
# Fuzzy Processing



# Output Membership Functions



# Rules



Sprinkler Control System

Antecedent 1: Temperature

Antecedent 2: Moisture

| Moisture \ Temperature | Cold | Cool | Normal | Warm | Hot |
|---|---|---|---|---|---|
| Wet | short | short | short | short | short |
| Moist | short | med. | med. | med. | med. |
| Dry | long | long | long | long | long |

**Sample rules extracted from table above are as follows:**

**If** temperature is hot **AND** soil is dry, **then** watering duration is long.

**If** temperature is cold **AND** soil is wet, **then** watering duration is short.

# Rule Strength

For air temp= 92° F.
For soil moisture= 11%

| | Rule Strengths |
|---|---|
| **Rule 1**<br>"**If** temperature is hot (.46) **AND** soil is dry (.25), **then** water duration is long." | .25 |
| **Rule 2**<br>"**If** temperature is warm (.2) **AND** soil is moist (.75), **then** water duration is medium." | .2 |
| **Rule 3**<br>"**If** temperature is warm (.2) **AND** soil is dry (.25), **then** water duration is long." | .2 |
| **Rule 4**<br>"**If** temperature is hot (.46) **AND** soil is moist (.75), **then** water duration is medium." | .46 |

**Fuzzy Output:**
**Water duration is .25 long, and .46 medium.**

# Output Membership Function