Chapter 1

Introduction to Fuzzy Logic

1.1 Fuzzy Logic: The word fuzzy means uncertainty. Any particular event which do not result any of the exact value (i.e. true or false) is fuzzy. Fuzzy Logic was introduced in 1965 by Lofti A. In other words, we can say that fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. There can be numerous other examples like this with the help of which we can understand the concept of fuzzy logic.

The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. For example, let us take the statement:

"Rama is old."

If Rama's age was 75, we might assign the statement the truth value of 0.80. The statement could be translated into set terminology as follows:

"Rama is a member of the set of old people."

This statement would be rendered symbolically with fuzzy sets as:

mOLD(Rama) = 0.80

where m is the membership function, operating in this case on the fuzzy set of old people, which returns a value between 0.0 and 1.0.

A **set** is an unordered collection of different elements. It can be written explicitly by listing its elements using the set bracket. If the order of the elements is changed or any element of a set is repeated, it does not make any changes in the set.

Example

- A set of all positive integers.
- A set of all the planets in the solar system.
- A set of all the states in India.
- A set of all the lowercase letters of the alphabet.

1.2 Differences between Fuzzy Logic and Neural Networks

Fuzzy logic allows making definite decisions based on imprecise or ambiguous data, whereas ANN tries to incorporate human thinking process to solve problems without mathematically modelling them. Even though both of these methods can be used to solve nonlinear problems, and problems that are not properly specified, they are not related. In contrast to Fuzzy logic, ANN tries to apply the thinking process in the human brain to solve problems. Further, ANN includes a learning process that involves learning algorithms and requires training data whereas

fuzzy logic includes development of membership functions and rules to relate them. Fuzzy logic basically deals with fixed and approximate (not exact) reasoning and the variables in fuzzy logic can take values from 0 to 1, this is contradicting to the traditional binary sets which takes value either 1 or 0 and since it can take any values in the range 0 to 1, it means that it is partially true and it is widely used for applications in control systems. Neural network on the other hand is based on biological neural network which is made up of artificial neurons interconnecting one another working in unison to produce outputs and it adapts to system with the data given to it by making adjustments to the synaptic connections that exist between the neurons.

Fuzzy logic makes decision based on the raw and ambigous data given to it whereas Neural network tries to learn from the data, incorporating the same way involved in the biological neural network. Both of these system are used to solve non-linear and complex problems and are no where related to each other.

1.3 Applications of Fuzzy System:

Aerospace

In aerospace, fuzzy logic is used in the following areas −

- Altitude control of spacecraft
- Satellite altitude control
- Flow and mixture regulation in aircraft deicing vehicles

Automotive

In automotive, fuzzy logic is used in the following areas −

- Trainable fuzzy systems for idle speed control
- Shift scheduling method for automatic transmission
- Intelligent highway systems
- Traffic control
- Improving efficiency of automatic transmissions

Business

In business, fuzzy logic is used in the following areas −

- Decision-making support systems
- Personnel evaluation in a large company

Defense

In defense, fuzzy logic is used in the following areas −

- Underwater target recognition
- Automatic target recognition of thermal infrared images
- Naval decision support aids
- Control of a hypervelocity interceptor
- Fuzzy set modeling of NATO decision making

Electronics

In electronics, fuzzy logic is used in the following areas −

- Control of automatic exposure in video cameras
- Humidity in a clean room
- Air conditioning systems
- Washing machine timing
- Microwave ovens
- Vacuum cleaners

Finance

In the finance field, fuzzy logic is used in the following areas −

- Banknote transfer control
- Fund management
- Stock market predictions

Industrial Sector

In industrial, fuzzy logic is used in following areas −

- Cement kiln controls heat exchanger control
- Activated sludge wastewater treatment process control
- Water purification plant control
- Quantitative pattern analysis for industrial quality assurance
- Control of constraint satisfaction problems in structural design
- Control of water purification plants

Manufacturing

In the manufacturing industry, fuzzy logic is used in following areas −

- Optimization of cheese production
- Optimization of milk production

Marine

In the marine field, fuzzy logic is used in the following areas −

- Autopilot for ships
- Optimal route selection
- Control of autonomous underwater vehicles
- Ship steering

Medical

In the medical field, fuzzy logic is used in the following areas −

- Medical diagnostic support system
- Control of arterial pressure during anesthesia
- Multivariable control of anesthesia
- Modeling of neuropathological findings in Alzheimer's patients
- Radiology diagnoses
- Fuzzy inference diagnosis of diabetes and prostate cancer

Securities

In securities, fuzzy logic is used in following areas −

- Decision systems for securities trading
- Various security appliances

Transportation

In transportation, fuzzy logic is used in the following areas −

- Automatic underground train operation
- Train schedule control
- Railway acceleration
- Braking and stopping

Pattern Recognition and Classification

In Pattern Recognition and Classification, fuzzy logic is used in the following areas −

- Fuzzy logic based speech recognition
- Fuzzy logic based
- Handwriting recognition
- Fuzzy logic based facial characteristic analysis
- Command analysis
- Fuzzy image search

Psychology

In Psychology, fuzzy logic is used in following areas −

- Fuzzy logic based analysis of human behavior
- Criminal investigation and prevention based on fuzzy logic reasoning

A **set** is an unordered collection of different elements. It can be written explicitly by listing its elements using the set bracket. If the order of the elements is changed or any element of a set is repeated, it does not make any changes in the set.

Example

- A set of all positive integers.
- A set of all the planets in the solar system.
- A set of all the states in India.
- A set of all the lowercase letters of the alphabet.

1.4 Historical Evolution Fuzzy Set:

Human Reasoning was dominated for centuries by the fundamental "Laws of Thought" (Korner, 1967), introduced by Aristotle (384-322 BC) and the philosophers that preceded him, which include:

• The principle of identity
• The law of the excluded middle
• The law of contradiction

In particular, the second of the above laws, stating that every proposition has to be either "True" or "False", was the basis for the genesis of the Aristotle's bi-valued Logic. The precision of the traditional mathematics owes undoubtedly a large part of its success to this Logic.

However, even when Parmenides proposed, around 400 BC, the first version of the law of the excluded middle, there were strong and immediate objections. For example, Heraclitus opposed that things could be simultaneously true and not true, whereas the Buddha Sidhartha Gautama, who lived in India a century earlier, had already indicated that almost every notion contains elements from its opposite one. The ancient Greek philosopher Plato (427-377 BC) laid the foundation of what it was later called FL by claiming that there exists a third area beyond "True" and "False", where these two opposite notions can exist together. More modern philosophers like Hegel, Marx, Engels and others adopted and further cultivated the above Plato's belief.

The Polish philosopher Jan Lukasiewicz (1878-1956) was the first to propose a systematic alternative of the bi-valued logic introducing in the early 1900's a three valued logic by adding the term "Possible" between "True" and "False" (Lejewski, 1967). Eventually he developed an entire notation and axiomatic system from which he hoped to derive modern mathematics. Later he also proposed four and five valued

Logics and he finally arrived to the conclusion that axiomatically nothing could prevent the derivation of an infinite valued Logic.
But it was not until relatively recently that an infinite-valued Logic was introduced (Zadeh, 1973), called FL, because it is based on the notion of FS initiated in 1965 (Zadeh, 1965) by Lotfi (2018), Professor at the University of Berkeley, California. An important goal of FL is that through it algorithmic procedures can be devised which translate the "fuzzy" terminology into numerical values, perform reliable operations upon those values and then return natural language statements in a reliable manner.

Zadeh (1921–2017) (Wikipedia, retrieved from the Web on February, 2012) was born in Baku, Azerbaijan of USSR, to a Russian Jewish mother (FanyaKoriman), who was a pediatrician, and an Iranian Azeri father (Rahim Aleskerzade), who was a journalist on assignment from Iran.

At the age of 10, when Stalin introduced collectivization of farms in USSR, the Zadeh family moved to Iran. In 1942 Zadeh graduated from the University of Tehran with a degree in electrical engineering and moved to the USA in 1944. He received a MS from MIT in 1946 and a Ph.D. in electrical engineering from Columbia University in 1949. He taught for ten years in Columbia, promoted to a Full Professor in 1957, before moving to Berkeley in 1959. Among others he introduced jointly with J.R. Ragazzini in 1962 the pioneering *z-transform method* used today in the digital analysis (Brule, 2016) whereas his more recent works include computing with words and perceptions (Zadeh, 1984;2005a) and an outline towards a generalized theory of uncertainty (Zadeh, 2005b). It has been estimated that Zadeh, who died in Berkeley on 6 September 2017, aged 96, counted in 2011 more than 950 000 citations by other researchers!
As it was expected, the far-reaching theory of fuzzy systems aroused some objections to the scientific community. While there have been generic complaints about the fuzziness of assigning values to linguistic terms, the most cogent criticisms come from Haak (1979). She argued that there are only two areas – the nature of Truth and Falsity and the fuzzy systems' utility – in which FL could be possibly needed, and then maintained that in both cases it can be shown that FL is unnecessary.

Fox (1981) responded to her objections indicating that FL is useful in three areas: To handle real-world relationships which are inherently fuzzy, to calculate the frequently existing in real world situations fuzzy data and to describe the operation of some inferential systems which are inherently fuzzy. His most powerful arguments were that traditional and FL need not be seen as competitive, but as complementary and that FL, despite the objections of classical logicians, has found its way into practical applications and has proved very successful there.

1.5 The concept of Fuzzy Set

Real life situations appear frequently where some definitions have not clear boundaries, like "the young people of a city", "the good players of a team", "the diligent students of

a class", etc. The need to model mathematically such kind of situations was one of the main reasons that laid to the development of the FS theory.
Let *U* be the set of the discourse. Then, according to Zadeh (1965), a *fuzzy subset A of U* (or for brevity a FS in *U*) can be defined with the help of its *membership function mA: U [0,1],* which assigns to each element *x* of *U* a real value *mA(Fox)* in [0, 1], called the *membership degree of x in A*. The closer is *mA(Fox)* to 1, the more *x* satisfies the characteristic property of *A*. Then one defines *A* as a set of ordered pairs of the form: *A = {(x, mA(x)) :x€U*
Many authors, for reasons of simplicity, identify the FS *A* with its membership function *mA*. A FS can be also denoted in the form of a symbolic sum, or a symbolic power series, or a symbolic integral, when *U* is a finite or numerable set or it has the power of the continuous respectively. For general facts on FS and the uncertainty connected to them we refer to the book of Klir and Folger (1988).
Example 1: The young human ages
Let *U* be the set of the non negative integers not exceeding 140 (considered as the upper bound of human life) representing the human ages. The set of all ages not exceeding a given integer in *U*, e.g. 20, is a crisp subset of *U*. On the contrary the set *A* of the young human ages, being not precisely defined, is a FS in *U*. The membership function of *A* can be defined by

$$m_A(x) = \begin{cases} \left[1+(0.04x)^2\right]^{-1}, & x \le 40 \\ 0, & x > 40 \end{cases}$$

Therefore, the age of a recently born baby has membership degree m*A*(0) = 1, the age of 25 years has membership degree:

$M_A(25) = (1 + 1^2)^{-1} = 0.5$, etc.

Mathematical Representation of a Set

Sets can be represented in two ways −

Roster or Tabular Form

In this form, a set is represented by listing all the elements comprising it. The elements are enclosed within braces and separated by commas.

Following are the examples of set in Roster or Tabular Form −

- Set of vowels in English alphabet, A = {a,e,i,o,u}
- Set of odd numbers less than 10, B = {1,3,5,7,9}

Set Builder Notation

In this form, the set is defined by specifying a property that elements of the set have in common. The set is described as A = {x:p(x)}

Example 1 − The set {a,e,i,o,u} is written as

$$A = \{x:x \text{ is a vowel in English alphabet}\}$$

Example 2 − The set {1,3,5,7,9} is written as

$B = \{x:1 \leq x < 10 \text{ and } (x\%2) \neq 0\}$

If an element x is a member of any set S, it is denoted by $x \in S$ and if an element y is not a member of set S, it is denoted by $y \notin S$.

Example − If $S = \{1,1.2,1.7,2\}, 1 \in S$ but $1.5 \notin S$

Cardinality of a Set

Cardinality of a set S, denoted by |S||S|, is the number of elements of the set. The number is also referred as the cardinal number. If a set has an infinite number of elements, its cardinality is ∞∞.

Example − $|\{1,4,3,5\}| = 4, |\{1,2,3,4,5,\ldots\}| = \infty$

If there are two sets X and Y, $|X| = |Y|$ denotes two sets X and Y having same cardinality. It occurs when the number of elements in X is exactly equal to the number of elements in Y. In this case, there exists a bijective function 'f' from X to Y.

$|X| \leq |Y|$ denotes that set X's cardinality is less than or equal to set Y's cardinality. It occurs when the number of elements in X is less than or equal to that of Y. Here, there exists an injective function 'f' from X to Y.

$|X| < |Y|$ denotes that set X's cardinality is less than set Y's cardinality. It occurs when the number of elements in X is less than that of Y. Here, the function 'f' from X to Y is injective function but not bijective.

If $|X| \leq |Y|$ and $|X| \leq |Y|$ then $|X| = |Y|$. The sets X and Y are commonly referred as **equivalent sets**.

Types of Sets

Sets can be classified into many types; some of which are finite, infinite, subset, universal, proper, singleton set, etc.

Finite Set

A set which contains a definite number of elements is called a finite set.

Example − S = {x|x∈ N and 70 > x > 50}

Infinite Set

A set which contains infinite number of elements is called an infinite set.

Example − S = {x|x∈ N and x > 10}

Subset

A set X is a subset of set Y (Written as X ⊆ Y) if every element of X is an element of set Y.

Example 1 − Let, X = {1,2,3,4,5,6} and Y = {1,2}. Here set Y is a subset of set X as all the elements of set Y is in set X. Hence, we can write Y⊆X.

Example 2 − Let, X = {1,2,3} and Y = {1,2,3}. Here set Y is a subset (not a proper subset) of set X as all the elements of set Y is in set X. Hence, we can write Y⊆X.

Proper Subset

The term "proper subset" can be defined as "subset of but not equal to". A Set X is a proper subset of set Y (Written as X ⊂ Y) if every element of X is an element of set Y and |X| < |Y|.

Example − Let, X = {1,2,3,4,5,6} and Y = {1,2}. Here set Y ⊂ X, since all elements in Y are contained in X too and X has at least one element which is more than set Y.

Universal Set

It is a collection of all elements in a particular context or application. All the sets in that context or application are essentially subsets of this universal set. Universal sets are represented as U.

Example − We may define U as the set of all animals on earth. In this case, a set of all mammals is a subset of U, a set of all fishes is a subset of U, a set of all insects is a subset of U, and so on.

Empty Set or Null Set

An empty set contains no elements. It is denoted by Φ. As the number of elements in an empty set is finite, empty set is a finite set. The cardinality of empty set or null set is zero.

Example – S = {x|x∈ N and 7 < x < 8} = Φ

Singleton Set or Unit Set

A Singleton set or Unit set contains only one element. A singleton set is denoted by {s}.

Example − S = {x|x∈ N, 7 < x < 9} = {8}

Equal Set

If two sets contain the same elements, they are said to be equal.

Example − If A = {1,2,6} and B = {6,1,2}, they are equal as every element of set A is an element of set B and every element of set B is an element of set A.

Equivalent Set

If the cardinalities of two sets are same, they are called equivalent sets.

Example − If A = {1,2,6} and B = {16,17,22}, they are equivalent as cardinality of A is equal to the cardinality of B. i.e. |A| = |B| = 3

Overlapping Set

Two sets that have at least one common element are called overlapping sets. In case of overlapping sets −

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

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

$$n(A) = n(A - B) + n(A \cap B)$$

$$n(B) = n(B - A) + n(A \cap B)$$

Example − Let, A = {1,2,6} and B = {6,12,42}. There is a common element '6', hence these sets are overlapping sets.

Disjoint Set

Two sets A and B are called disjoint sets if they do not have even one element in common. Therefore, disjoint sets have the following properties −

$$n(A \cap B) = \phi$$

$$n(A \cup B) = n(A) + n(B)$$

Example − Let, A = {1,2,6} and B = {7,9,14}, there is not a single common element, hence these sets are overlapping sets.

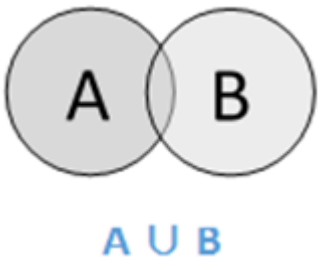
Operations on Classical Sets

Set Operations include Set Union, Set Intersection, Set Difference, Complement of Set, and Cartesian Product.

Union

The union of sets A and B (denoted by A∪ BA ∪ B) is the set of elements which are in A, in B, or in both A and B. Hence, A ∪ B = {x|x∈ A OR x ∈ B}.

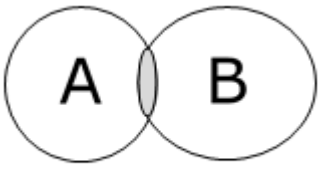
Example − If A = {10,11,12,13} and B = {13,14,15}, then A ∪ B = {10,11,12,13,14,15} – The common element occurs only once.



A ∪ B

Intersection

The intersection of sets A and B (denoted by A ∩ B) is the set of elements which are in both A and B. Hence, A ∩ B = {x|x∈ A AND x ∈ B}.

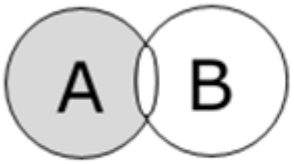


A ∩ B

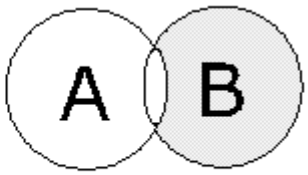
Difference/ Relative Complement

The set difference of sets A and B (denoted by A–B) is the set of elements which are only in A but not in B. Hence, A − B = {x|x∈ A AND x ∉ B}.

Example − If A = {10,11,12,13} and B = {13,14,15}, then (A − B) = {10,11,12} and (B − A) = {14,15}. Here, we can see (A − B) ≠ (B − A)



A - B



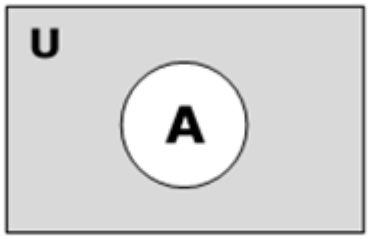
B - A

Complement of a Set

The complement of a set A (denoted by A′) is the set of elements which are not in set A. Hence, A′ = {x|x∉ A}.

More specifically, A′ = (U−A) where U is a universal set which contains all objects.

Example − If A = {x|x belongs to set of add integers} then A′ = {y|y does not belong to set of odd integers}



Complement of set A

Cartesian Product / Cross Product

The Cartesian product of n number of sets A1,A2,…An denoted as A1 × A2...× An can be defined as all possible ordered pairs (x1,x2,…xn) where x1 ∈ A1,x2 ∈ A2,…xn∈ An

Example − If we take two sets A = {a,b} and B = {1,2},

The Cartesian product of A and B is written as − A × B = {(a,1),(a,2),(b,1),(b,2)}

And, the Cartesian product of B and A is written as − B × A = {(1,a),(1,b),(2,a),(2,b)}

Properties of Classical Sets

Properties on sets play an important role for obtaining the solution. Following are the different properties of classical sets −

Commutative Property

Having two sets **A** and **B**, this property states −

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

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

Associative Property

Having three sets **A**, **B** and **C**, this property states −

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

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

Distributive Property

Having three sets **A**, **B** and **C**, this property states −

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

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

Idempotency Property

For any set **A**, this property states −

$$A \cup A = A$$

$$A \cap A = A$$

Identity Property

For set **A** and universal set **X**, this property states −

$$A \cup \varphi = A$$

$$A \cap X = A$$

$$A \cap \varphi = \varphi$$

$$A \cup X = X$$

Transitive Property

Having three sets **A**, **B** and **C**, the property states −

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

, then $A \subseteq C$

De Morgan's Law

It is a very important law and supports in proving tautologies and contradiction. This law states −

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

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

Exercise:

1. What is Fuzzy System? Give Example for the same.
2. Explain the concept of fuzziness with suitable example.
3. State the applications of fuzzy sets.
4. What is fuzzy set/
5. Give comparison between fuzzy system and neural networks.
6. Give the historical evolution of fuzzy system.

Chapter 2

Fuzzy Sets

Introduction:

Fuzzy logic starts with and builds on a set of user-supplied human language rules. The fuzzy systems convert these rules to their mathematical equivalents. This simplifies the job of the system designer and the computer, and results in much more accurate representations of the way systems behave in the real world.

Additional benefits of fuzzy logic include its simplicity and its flexibility. Fuzzy logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity. "If you don't have a good plant model, or if the system is changing, then fuzzy will produce a better solution than conventional control techniques," says Bob Varley, a Senior Systems Engineer at Harris Corp., an aerospace company in Palm Bay, Florida.

You can create a fuzzy system to match any set of input-output data. The Fuzzy Logic Toolbox makes this particularly easy by supplying adaptive techniques such as adaptive neuro-fuzzy inference systems (ANFIS) and fuzzy subtractive clustering.

Fuzzy logic models, called fuzzy inference systems, consist of a number of conditional "if-then" rules. For the designer who understands the system, these rules are easy to write, and as many rules as necessary can be supplied to describe the system adequately (although typically only a moderate number of rules are needed).

In fuzzy logic, unlike standard conditional logic, the truth of any statement is a matter of degree. (How cold is it? How high should we set the heat?) We are familiar with inference rules of the form p -> q (p implies q). With fuzzy logic, it's possible to say (.5* p) -> (.5 * q). For example, for the rule if (weather is cold) then (heat is on), both variables, cold and on, map to ranges of values. Fuzzy inference systems rely on membership functions to explain to the computer how to calculate the correct value between 0 and 1. The degree to which any fuzzy statement is true is denoted by a value between 0 and 1.

Not only do the rule-based approach and flexible membership function scheme make fuzzy systems straightforward to create, but they also simplify the design of systems and ensure that you can easily update and maintain the system over time.

Fuzzy Set Theory was formalised by Professor LoftiZadeh at the University of California in 1965. What Zadeh proposed is very much a paradigm shift that first gained

acceptance in the Far East and its successful application has ensured its adoption around the world.

A paradigm is a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. For example the use of transistors instead of vacuum tubes is a paradigm shift - likewise the development of Fuzzy Set Theory from conventional bivalent set theory is a paradigm shift.

Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically. For example, Fig 1 below illustrates bivalent sets to characterise the temperature of a room.

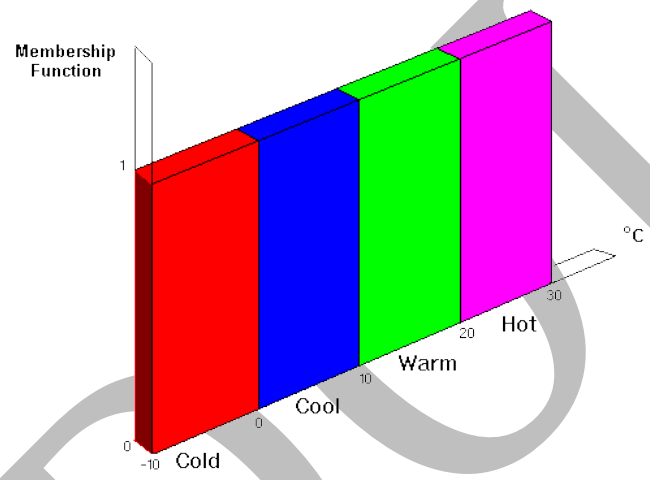


Fig. 1 : Bivalent Sets to Characterize the Temp. of a room.

The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' hence the expert knowledge we need to define our system is mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Fahrenheit of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur.

This natural phenomenon can be described more accurately by Fuzzy Set Theory. Fig.2 below shows how fuzzy sets quantifying the same information can describe this natural drift.

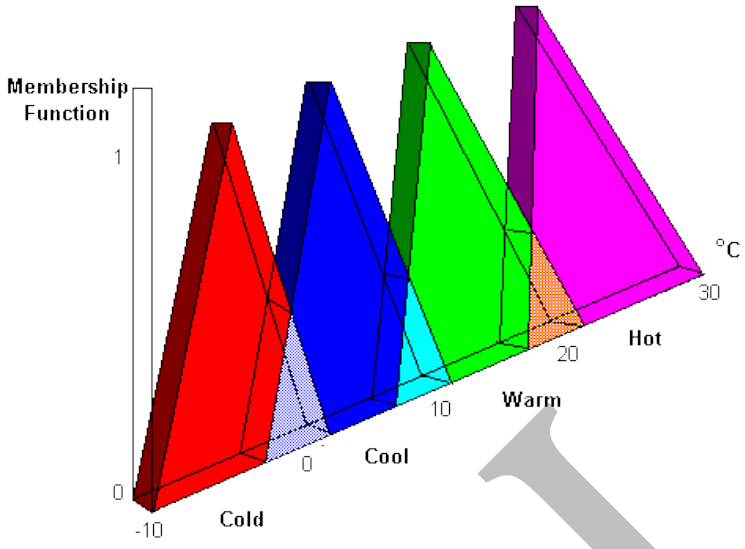


Fig. 2 - Fuzzy Sets to characterize the Temp. of a room.

Fuzzy Logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems.

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot".

2.1 History of Fuzzy Logic

Although, the concept of fuzzy logic had been studied since the 1920's. The term fuzzy logic was first used with 1965 by LotfiZadeh a professor of UC Berkeley in California. He observed that conventional computer logic was not capable of manipulating data representing subjective or unclear human ideas.

Fuzzy logic has been applied to various fields, from control theory to AI. It was designed to allow the computer to determine the distinctions among data which is neither true nor false. Something similar to the process of human reasoning. Like Little dark, Some brightness, etc.

2.2 Characteristics of Fuzzy Logic

Here, are some important characteristics of fuzzy logic:

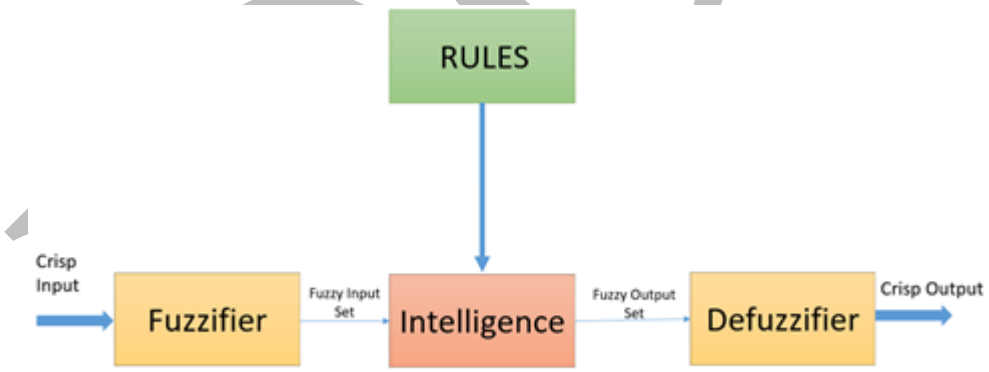
- Flexible and easy to implement machine learning technique
- Helps you to mimic the logic of human thought
- Logic may have two values which represent two possible solutions
- Highly suitable method for uncertain or approximate reasoning
- Fuzzy logic views inference as a process of propagating elastic constraints
- Fuzzy logic allows you to build nonlinear functions of arbitrary complexity.
- Fuzzy logic should be built with the complete guidance of experts

However, fuzzy logic is never a cure for all. Therefore, it is equally important to understand that where we should not use fuzzy logic.

Here, are certain situations when you better not use Fuzzy Logic:

- If you don't find it convenient to map an input space to an output space
- Fuzzy logic should not be used when you can use common sense
- Many controllers can do the fine job without the use of fuzzy logic

2.3 Fuzzy Logic Architecture



Fuzzy Logic architecture has four main parts as shown in the diagram:

Rule Base:

It contains all the rules and the if-then conditions offered by the experts to control the decision-making system. The recent update in fuzzy theory provides various methods for the design and tuning of fuzzy controllers. This updates significantly reduce the number of the fuzzy set of rules.

Fuzzification:

Fuzzification step helps to convert inputs. It allows you to convert, crisp numbers into fuzzy sets. Crisp inputs measured by sensors and passed into the control system for further processing. Like Room temperature, pressure, etc.

Inference Engine:

It helps you to determines the degree of match between fuzzy input and the rules. Based on the % match, it determines which rules need implment according to the given input field. After this, the applied rules are combined to develop the control actions.

Defuzzification:

At last the Defuzzification process is performed to convert the fuzzy sets into a crisp value. There are many types of techniques available, so you need to select it which is best suited when it is used with an expert system.

Fuzzy Logic vs. Probability

Fuzzy Logic	**Probability**
Fuzzy: Tom's degree of membership within the set of old people is 0.90.	Probability: There is a 90% chance that Tom is old.
Fuzzy logic takes truth degrees as a mathematical basis on the model of the vagueness phenomenon.	Probability is a mathematical model of ignorance.

Crisp vs. Fuzzy

Crisp	**Fuzzy**
It has strict boundary T or F	Fuzzy boundary with a degree of membership
Some crisp time set can be fuzzy	It can't be crisp
True/False {0,1}	Membership values on [0,1]
In Crisp logic law of Excluded Middle and Non- Contradiction may or may not hold	In the fuzzy logic law of Excluded Middle and Non- Contradiction hold

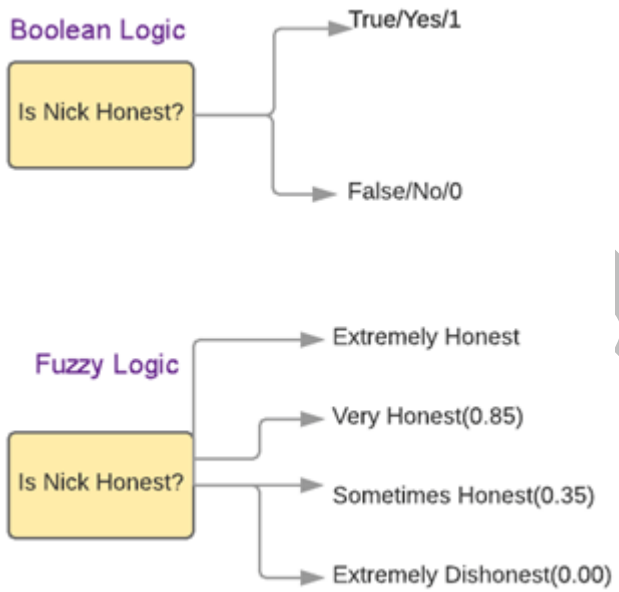
Classical Set vs. Fuzzy set Theory

Classical Set	**Fuzzy Set Theory**
Classes of objects with sharp boundaries.	Classes of objects do not have sharp boundaries.
A classical set is defined by crisp boundaries, i.e., there is clarity about the location of the set	A fuzzy set always has ambiguous boundaries, i.e., there may be uncertainty about the location

boundaries.	of the set boundaries.
Widely used in digital system design	Used only in fuzzy controllers.

Fuzzy Logic Examples

See the below-given diagram. It shows that in fuzzy systems, the values are denoted by a 0 to 1 number. In this example, 1.0 means absolute truth and 0.0 means absolute falseness.



2.4 Application Areas of Fuzzy Logic

The Blow given table shows how famous companies using fuzzy logic in their products.

Product	**Company**	**Fuzzy Logic**
Anti-lock brakes	Nissan	Use fuzzy logic to controls brakes in hazardous cases depend on car speed, acceleration, wheel speed, and acceleration
Auto transmission	NOK/Nissan	Fuzzy logic is used to control the fuel injection and ignition based on throttle setting, cooling water temperature, RPM, etc.
Auto engine	Honda, Nissan	Use to select geat based on engine load, driving style, and road conditions.
Copy machine	Canon	Using for adjusting drum voltage based on picture

		density, humidity, and temperature.
Cruise control	Nissan, Isuzu, Mitsubishi	Use it to adjusts throttle setting to set car speed and acceleration
Dishwasher	Matsushita	Use for adjusting the cleaning cycle, rinse and wash strategies based depend upon the number of dishes and the amount of food served on the dishes.
Elevator control	Fujitec, Mitsubishi Electric, Toshiba	Use it to reduce waiting for time-based on passenger traffic
Golf diagnostic system	Maruman Golf	Selects golf club based on golfer's swing and physique.
Fitness management	Omron	Fuzzy rules implied by them to check the fitness of their employees.
Kiln control	Nippon Steel	Mixes cement
Microwave oven	Mitsubishi Chemical	Sets lunes power and cooking strategy
Palmtop computer	Hitachi, Sharp, Sanyo, Toshiba	Recognizes handwritten Kanji characters
Plasma etching	Mitsubishi Electric	Sets etch time and strategy

2.4.1 Advantages of Fuzzy Logic System

- The structure of Fuzzy Logic Systems is easy and understandable
- Fuzzy logic is widely used for commercial and practical purposes
- It helps you to control machines and consumer products
- It may not offer accurate reasoning, but the only acceptable reasoning
- It helps you to deal with the uncertainty in engineering
- Mostly robust as no precise inputs required
- It can be programmed to in the situation when feedback sensor stops working
- It can easily be modified to improve or alter system performance
- inexpensive sensors can be used which helps you to keep the overall system cost and complexity low
- It provides a most effective solution to complex issues

2.4.2 Disadvantages of Fuzzy Logic Systems

- Fuzzy logic is not always accurate, so The results are perceived based on assumption, so it may not be widely accepted.
- Fuzzy systems don't have the capability of machine learning as-well-as neural network type pattern recognition

- Validation and Verification of a fuzzy knowledge-based system needs extensive testing with hardware
- Setting exact, fuzzy rules and, membership functions is a difficult task
- Some fuzzy time logic is confused with probability theory and the terms

2.5 Features of Membership Functions

We will now discuss the different features of Membership Functions.

Core

For any fuzzy set $\tilde{A}$, the core of a membership function is that region of universe that is characterize by full membership in the set. Hence, core consists of all those elements $y$of the universe of information such that,

$$\mu_{\tilde{A}}(y)=1$$

Support

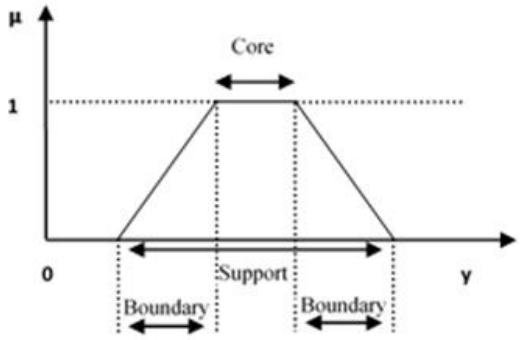
For any fuzzy set $\tilde{A}$, the support of a membership function is the region of universe that is characterize by a nonzero membership in the set. Hence core consists of all those elements $y$of the universe of information such that,

$$\mu_{\tilde{A}}(y)>0$$

Boundary

For any fuzzy set $\tilde{A}$, the boundary of a membership function is the region of universe that is characterized by a nonzero but incomplete membership in the set. Hence, core consists of all those elements $y$of the universe of information such that,

$$1>\mu_{\tilde{A}}(y)>0$$



Features of Membership Function

2.5.1 Fuzzification

It may be defined as the process of transforming a crisp set to a fuzzy set or a fuzzy set to fuzzier set. Basically, this operation translates accurate crisp input values into linguistic variables.

Following are the two important methods of fuzzification −

Support Fuzzification(s-fuzzification) Method

In this method, the fuzzified set can be expressed with the help of the following relation

$$\tilde{A}=\mu_1 Q(x_1)+\mu_2 Q(x_2)+...+\mu_n Q(x_n)$$

Here the fuzzy set $Q(x_i)$is called as kernel of fuzzification. This method isimplemented by keeping μ_i constant and x_i being transformed to a fuzzy set $Q(x_i)$.

Grade Fuzzification (g-fuzzification) Method

It is quite similar to the above method but the main difference is that it kept $x_i$constant and $\mu_i$is expressed as a fuzzy set.

2.5.2 Defuzzification

It may be defined as the process of reducing a fuzzy set into a crisp set or to convert a fuzzy member into a crisp member.

We have already studied that the fuzzification process involves conversion from crisp quantities to fuzzy quantities. In a number of engineering applications, it is necessary to defuzzify the result or rather "fuzzy result" so that it must be converted to crisp result. Mathematically, the process of Defuzzification is also called "rounding it off".

The different methods of Defuzzification are described below −

Max-Membership Method

This method is limited to peak output functions and also known as height method. Mathematically it can be represented as follows −

$$\mu_{\tilde{A}}(x_*)>\mu_{\tilde{A}}(x) for all x \in X$$

Here, $x_*$is the defuzzified output.

Centroid Method

This method is also known as the center of area or the center of gravity method. Mathematically, the defuzzified output $x*$will be represented as −

$$x*=\int\mu A\tilde{}(x).xdx\int\mu A\tilde{}(x).dx$$

Weighted Average Method

In this method, each membership function is weighted by its maximum membership value. Mathematically, the defuzzified output $x*$

will be represented as −

$$x*=\sum\mu\bar{A}\tilde{}(\bar{x_i}).\underline{x}_i\sum\mu A\tilde{}(\bar{x_i})$$

Mean-Max Membership

This method is also known as the middle of the maxima. Mathematically, the defuzzified output $x*$

will be represented as −

$$x*=\sum i=1n\bar{x}_in$$

2.6 Operations on Fuzzy Sets

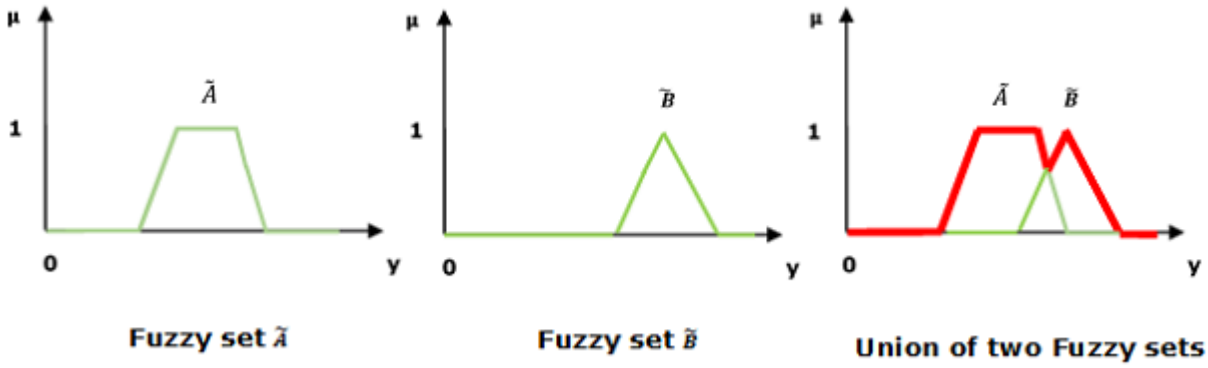
Having two fuzzy sets $A\tilde{}$ and $B\tilde{}$, the universe of information U and an element y of the universe, the following relations express the union, intersection and complement operation on fuzzy sets.

Union/Fuzzy 'OR'

Let us consider the following representation to understand how the **Union/Fuzzy 'OR'** relation works −

$$\mu A\tilde{}\cup B\tilde{}(y)=\mu A\tilde{}\vee\mu B\tilde{}\forall y\in U$$

Here ∨ represents the 'max' operation.



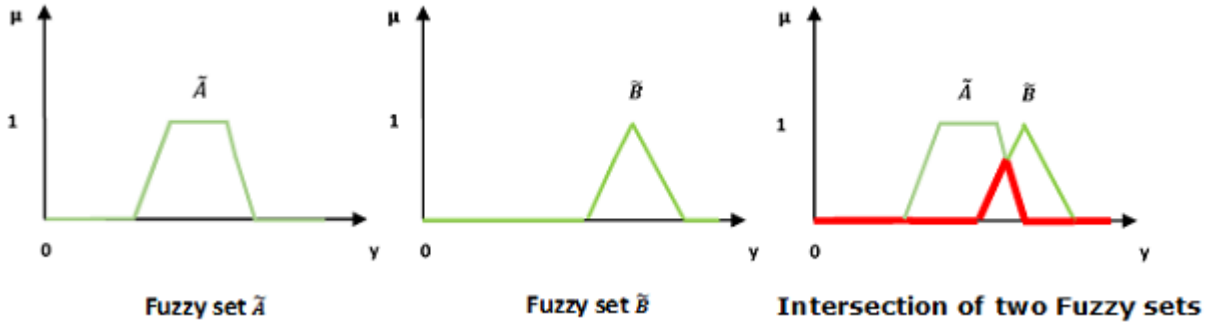
Fuzzy set $\tilde{A}$ Fuzzy set $\tilde{B}$ Union of two Fuzzy sets

Intersection/Fuzzy 'AND'

Let us consider the following representation to understand how the **Intersection/Fuzzy 'AND'** relation works −

$$\mu_{\tilde{A}\cap\tilde{B}}(y)=\mu_{\tilde{A}}\wedge\mu_{\tilde{B}}\forall y\in U$$

Here ∧ represents the 'min' operation.

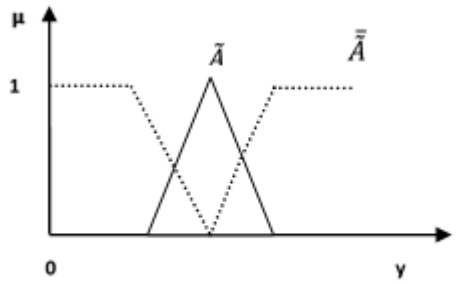


Fuzzy set $\tilde{A}$ Fuzzy set $\tilde{B}$ Intersection of two Fuzzy sets

Complement/Fuzzy 'NOT'

Let us consider the following representation to understand how the **Complement/Fuzzy 'NOT'** relation works −

$$\mu_{\tilde{A}}=1-\mu_{\tilde{A}}(y).y\in U$$



Complement of a fuzzy set

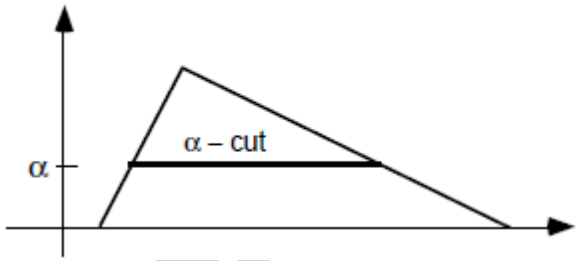
Definition. (support) Let A be a fuzzy subset ofX; the support of A, denoted supp(A), is the crispsubset of X whose elements all have nonzero membership grades in A.

$$\text{supp}(A) = \{x \in X \mid A(x) > 0\}.$$

Definition. (normal fuzzy set) A fuzzy subset A ofa classical set X is called normal if there exists anx $\in$ X such that A(x) = 1. Otherwise A is subnormal.

Definition. (α-cut) An α-level set of a fuzzy setA of X is a non-fuzzy set denoted by $[A]_\alpha$ and isdefined by

$$[A]^\alpha = \begin{cases} \{t \in X | A(t) \geq \alpha\} \text{ if } \alpha > 0 \\ \text{cl}(\text{suppA}) \text{ if } \alpha = 0 \end{cases}$$



*where*cl(*suppA*) *denotes the closure of the supportof A.*

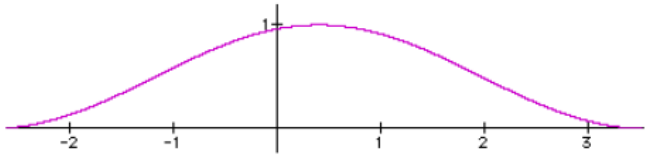
Definition. *(convex fuzzy set) A fuzzy set A of X iscalled* convex *if* $[A]_\alpha$ *is a convex subset of X* $\forall \alpha \in [0, 1]$.An*α*-cut of a triangular fuzzy number.

In many situations people are only able to characterize numeric information imprecisely. For example, people use terms such as, about 5000, near zero, or essentially bigger than 5000. These are examples of what are called *fuzzy numbers*. Using the theory of fuzzy subsets we can represent these fuzzy numbers as fuzzy subsets of the set of real numbers.

Definition. *(fuzzy number) A fuzzy number A is afuzzy set of the real line with a normal, (fuzzy) convex and continuous membership function of bounded support. The family of fuzzy numbers will be denoted by F.*

Definition. *(quasi fuzzy number) A quasi fuzzy numberA is a fuzzy set of the real line with a normal,fuzzy convex and continuous membership functionsatisfying the limit conditions*

$$\lim_{t\to\infty} A(t) = 0, \quad \lim_{t\to-\infty} A(t) = 0.$$

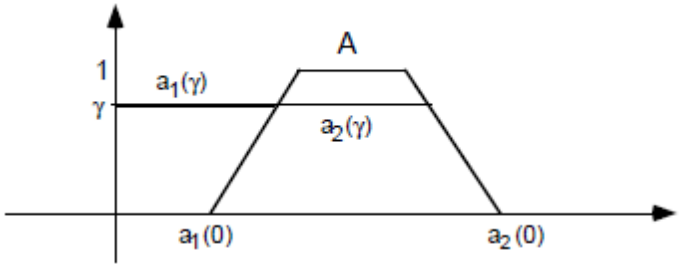


Fuzzy number.

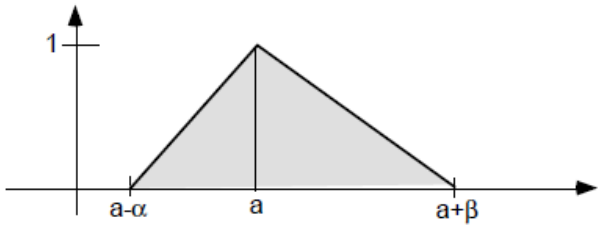
Let A be a fuzzy number. Then $[A]\gamma$ is a closedconvex (compact) subset of R for all $\gamma \in [0, 1]$. Let us introduce the notations $a1(\gamma) = \min[A]^\gamma$, $a2(\gamma) = \max[A]^\gamma$
In other words, $a1(\gamma)$ denotes the left-hand side and$a2(\gamma)$ denotes the right-hand side of the γ-cut. It iseasy to see that If$\alpha \preceq \beta$ then $[A]^\alpha \supset [A]^\beta$

Furthermore, the left-hand side functiona_1: [0, 1] →Ris monoton increasing and lower semi-continuous,and the right-hand side functiona_2: [0, 1] →Ris monoton decreasing and upper semi-continuous.

We shall use the notation$[A]\gamma = [a1(\gamma), a2(\gamma)]$.The support of A is the open interval (a1(0), a2(0)).



If A is not a fuzzy number then there exists a $\gamma \in [0, 1]$ such that $[A]\gamma$ is not a convex subset of R.
Definition. (Triangular fuzzy number) A fuzzy set A is called triangular fuzzy number with peak (or center) a, left width $\alpha > 0$ and right width $\beta > 0$ if its membership function has the following form

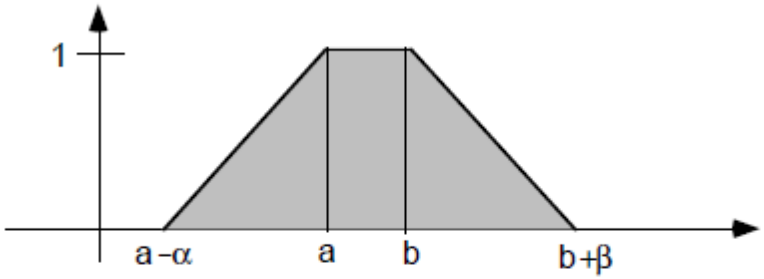


Triangular fuzzy number.

and we use the notation A = (a, α, β). It can easily be verified that
[A]γ = [a − (1 − γ)α, a + (1 − γ)β], ∀γ ∈ [0, 1]. The support of A is (a − α, b + β).
A triangular fuzzy number with center a may be seen as a fuzzy quantity "x is approximately equal to a".

Definition .*(trapezoidal fuzzy number) A fuzzy set A is called trapezoidal fuzzy number with tolerance interval* [*a, b*]*, left width α and right width β if its membership function has the following form*

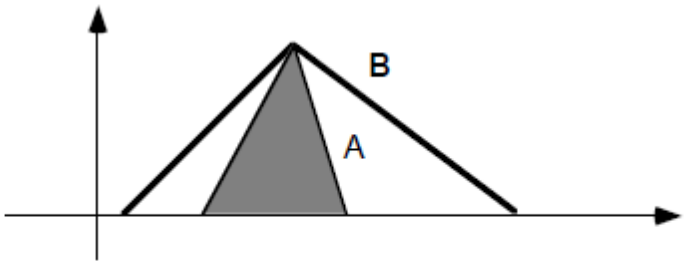
$$A(t) = \begin{cases} 1-(a-t)/\alpha & \text{if } a-\alpha \le t \le a \\ 1 & \text{if } a \le t \le b \\ 1-(t-b)/\beta & \text{if } a \le t \le b+\beta \\ 0 & \text{otherwise} \end{cases}$$



Trapezoidal fuzzy number.

and we use the notation A = (a, b, α, β). It can easily be shown that
[A]γ = [a −(1 −γ)α, b + (1 −γ)β], ∀γ ∈[0, 1]. The support of A is (a −α, b + β).
A trapezoidal fuzzy number may be seen as a fuzzy quantity "x is approximately in the interval [a, b]".

Definition. (subsethood) Let A and B are fuzzy subsets of a classical set X. We say that A is a subset of B if A(t) ≤ B(t), ∀t ∈ X.

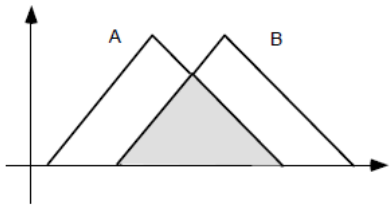


A is a subset of B.

Operations on fuzzy sets

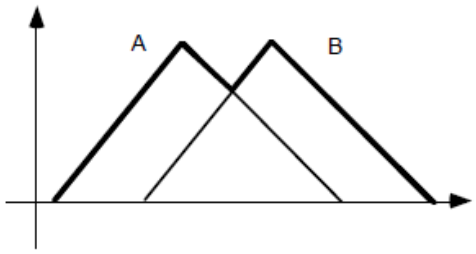
We extend the classical set theoretic operations from ordinary set theory to fuzzy sets. We note that all those operations which are extensions of crisp concepts reduce to theirusual meaning when the fuzzy subsets have membership degrees that are drawn from{0,1}. For this reason, when extending operations to fuzzy sets we use the same symbol as in set theory. Let A and B are fuzzy subsets of a nonempty (crisp) set X.

Definition. *(intersection) The intersection of A and B is defined as*
$(A \cap B)(t) = \min\{A(t), B(t)\} = A(t) \wedge B(t)$, *for all* $t \in X$.



Intersection of two triangular fuzzy numbers.

Definition. *(union) The union of A and B is definedasv*$(A \cup B)(t) = \max\{A(t), B(t)\} = A(t) \vee B(t)$,*for all* $t \in X$.



Union of two triangular fuzzy numbers.

Definition. *(complement) The complement of afuzzy set A is defined as*

$$(\neg A)(t) = 1 - A(t)$$

A closely related pair of properties which hold inordinary set theory are *the law of excluded middle* $A \vee \neg A = X$ and *the law of non-contradiction principle* $A \wedge \neg A = \emptyset$. It is clear that $\neg 1X = \emptyset$ and $\neg \emptyset = 1X$, however, thelaws of excluded middle and non-contradiction arenot satisfied in fuzzy logic.

2.7 FUZZY LOGIC OBJECTIONS

It would be remarkable if a theory as far-reaching as fuzzy systems did not arouse some objections in the professional community. While there have been generic complaints about the "fuzziness" of the process of assigning values to linguistic terms, perhaps the most cogent criticisms come from Haack . A formal logician, Haack argues that there are only two areas in which fuzzy logic could possibly be demonstrated to be "needed," and then maintains that in each case it can be shown that fuzzy logic is not necessary.

The first area Haack defines is that of the nature of Truth and Falsity: if it could be shown, she maintains, that these are fuzzy values and not discrete ones, then a need for fuzzy logic would have been demonstrated. The other area she identifies is that of fuzzy systems' utility: if it could be demonstrated that generalizing classic logic to encompass fuzzy logic would aid in calculations of a given sort, then again a need for fuzzy logic would exist.

In regards to the first statement, Haack argues that True and False are discrete terms. For example, "The sky is blue" is either true or false; any fuzziness to the statement arises from an imprecise definition of terms, not out of the nature of Truth. As far as fuzzy systems' utility is concerned, she maintains that no area of data manipulation is made easier through the introduction of fuzzy calculus; if anything, she says, the calculations become more complex. Therefore, she asserts, fuzzy logic is unnecessary.

Fox has responded to her objections, indicating that there are three areas in which fuzzy logic can be of benefit: as a "requisite" apparatus (to describe real-world relationships which are inherently fuzzy); as a "prescriptive" apparatus (because some data is fuzzy, and therefore requires a fuzzy calculus); and as a "descriptive" apparatus (because some inferencing systems are inherently fuzzy).

His most powerful arguments come, however, from the notion that fuzzy and classic logics need not be seen as competitive, but complementary. He argues that many of Haack's objections stem from a lack of semantic clarity, and that ultimately fuzzy statements may be translatable into phrases which classical logicians would find palatable.

Exercise:

1. Explain in brief about fuzzy logic.
2. Discuss the history of fuzzy system.

3. Explain fuzzy operations with suitable example.
4. What is subsethood? Explain in brief with suitable example.
5. Explain trapezoidal and union operation.
6. Explain intersection and triangular operations.
7. Explain in brief quasi fuzzy number.

Chapter 3

Fuzzy Relations and Implications

3.1 Crisp and Fuzzy Relation

A fuzzy relation generalizes these degrees to membership grades. So, a crisp relation is a restricted case of a fuzzy relation.

Crisp relation:

+ degrees or strengths of relation

Fuzzy relation

- Cartesian product :

$$\underset{i\in N_n}{\times} X_i = \{(x_1,\cdots,x_n) \mid x_i \in X_i, \forall i \in N_n\}$$

$$N_n = \{1,2,\cdots,n\}$$

- n-ary relation: a subset of $\underset{i\in N_n}{\times} X_i$

$$R(X_1, X_2,\cdots, X_n) \subseteq X_1 \times X_2 \times \cdots \times X_n$$

i.e.,

a set the universal set

Characteristic function:

$$\mu_R(x_1, x_1,\cdots x_n) = \begin{cases} 1 & \text{if } (x_1, x_1,\cdots x_n) \in R \\ 0 & \text{otherwise} \end{cases}$$

- Binary, Ternary, Quaternary, Quinary, n-ary

Relations

Definition of Relation

A relation among crisp sets X1,...,Xn is a subset of X1 × ... × Xn denoted as R(X1,...,Xn) or R(Xi | 1 ≤ i ≤ n). So, the relation R(X1,...,Xn) ⊆ X1 × ... ×Xn is set, too. The basic concept of sets can be also applied to relations: • containment, subset, union, intersection, complement Each crisp

relation can be defined by its characteristic function R(x1,...,xn) =(1, if and only if (x1,...,xn) ∈ R, 0, otherwise.

The membership of (x1,...,xn) in R signifies that the elements of (x1,...,xn) are related to each other.

Relation as Ordered Set of Tuples

A relation can be written as a set of ordered tuples. Thus R(X1,...,Xn) represents n-dim. membership array R = [ri1,...,in]. • Each element of i1 of R corresponds to exactly one member of X1. • Each element of i2 of R corresponds to exactly one member of X2. • And so on...

If (x1,...,xn) ∈ X1 × ... ×Xn corresponds to ri1,...,in ∈ R, then ri1,...,in =(1, if and only if (x1,...,xn) ∈ R, 0, otherwise.

Fuzzy Relations

The characteristic function of a crisp relation can be generalized to allow tuples to have degrees of membership. • Recall the generalization of the characteristic function of a crisp set!

Then a fuzzy relation is a fuzzy set defined on tuples (x1,...,xn) that may have varying degrees of membership within the relation. The membership grade indicates strength of the present relation between elements of the tuple. The fuzzy relation can also be represented by an n-dimensional membership array.

Cartesian Product of Fuzzy Sets: n Dimensions

Let n ≥ 2 fuzzy sets A1,...,An be defined in the universes of discourse X1,...,Xn, respectively. The Cartesian product of A1,...,An denoted by A1 × ... × An is a fuzzy relation in the product space X1 × ... × Xn. It is defined by its membership function

$\mu_{A1\times...\times An}(x1,...,xn) = \top(\mu_{A1}(x1),...,\mu_{An}(xn))$

whereas xi ∈ Xi, $1 \leq i \leq n$. Usually ⊤ is the minimum (sometimes also the product).

Cartesian Product of Fuzzy Sets: 2 Dimensions

A special case of the Cartesian product is when n = 2. Then the Cartesian product of fuzzy sets A∈ F(X) and B ∈ F(Y) is a fuzzy relation A × B ∈ F(X × Y) defined by

$\mu_{A\times B}(x,y) = \top[\mu_A(x), \mu_B(y)], \forall x \in X, \forall y \in Y.$

Subsequences

Consider the Cartesian product of all sets in the family

X = {Xi | i∈INn = {1,2,...,n}}.

For each sequence (n-tuple) x = (x1,...,xn) $\in \times i \in INn Xi$ and each sequence (r-tuple, $r \leq n$) y = (y1,...,yr) $\in \times j \in JXj$ where $J \subseteq INn$ and $|J| = r$ y is called subsequence of x if and only if yj = xj, $\forall j \in J$. $y \prec x$ denotes that y is subsequence of x.

Projection

Given a relation R(x1,...,xn). Let [R $\downarrow$ Y] denote the projection of R on Y. It disregards all sets in X except those in the family

Y = {Xj | j ∈ J ⊆INn}.

Then [R $\downarrow$ Y] is a fuzzy relation whose membership function is defined on the Cartesian product of the sets in Y

[R $\downarrow$ Y](y) = max x≻y

R(x).

Under special circumstances, this projection can be generalized by replacing the max operator by another t-conorm.

Cylindric Extension

Another operation on relations is called cylindric extension. Let X and Y denote the same families of sets as used for projection. Let R be a relation defined on Cartesian product of sets in family Y. Let [R $\uparrow$ X \ Y] denote the cylindric extension of R into sets X1, (i∈INn) which are in X but not in Y. It follows that for each x with x ≻ y

[R $\uparrow$ X \ Y](x) = R(y).

The cylindric extension • produces largest fuzzy relation that is compatible with projection, • is the least specific of all relations compatible with projection, • guarantees that no information not included in projection is used to determine extended relation.

Example

Consider again the example for the projection. The membership functions of the cylindric extensions of all projections are already shown in the table under the assumption that their arguments are extended to (x1,x2,x3) e.g.

[R23 $\uparrow$ {X1}](0,0,2) = [R23 $\uparrow$ {X1}](1,0,2) = R23(0,2) = 0.2.

In this example none of the cylindric extensions are equal to the original fuzzy relation. This is identical with the respective projections. Some information was lost when the given relation was replaced by any one of its projections.

Cylindric Closure

Relations that can be reconstructed from one of their projections by cylindric extension exist. However, they are rather rare. It is more common that relation can be exactly reconstructed • from several of its projections (max), • by taking set intersection of their cylindric extensions (min). The resulting relation is usually called cylindric closure. Let the set of projections {Pi | i∈ I} of a relation on X be given. Then the cylindric closure cyl{Pi} is defined for each x ∈ X as

cyl{Pi}(x) = min i∈I [Pi ↑ X \ Yi](x).

Yi denotes the family of sets on which Pi is defined.

Example

Consider again the example for the projection. The cylindric closures of three families of the projections are shown below:

$(x_1.$	$x_2.$	$x_3)$	$R(x_1, x_2, x_3)$	$cyl(R_{12}, R_{13}, R_{23})$	$cyl(R_1, R_2, R_3)$	$cyl(R_{12}, R_3)$
0	0	0	0.4	0.5	0.9	0.9
0	0	1	0.9	0.9	0.9	0.9
0	0	2	0.2	0.2	0.9	0.9
0	1	0	1.0	1.0	1.0	1.0
0	1	1	0.0	0.5	0.9	0.9
0	1	2	0.8	0.8	1.0	1.0
1	0	0	0.5	0.5	0.9	0.5
1	0	1	0.3	0.5	0.9	0.5
1	0	2	0.1	0.2	0.9	0.5
1	1	0	0.0	0.5	1.0	1.0
1	1	1	0.5	0.5	0.9	0.9
1	1	2	1.0	1.0	1.0	1.0

Motivation and Domain

Binary relations are significant among n-dimensional relations. They are (in some sense) generalized mathematical functions. On the contrary to functions from X to Y, binary relations R(X,Y) may assign to each element of X two or more elements of Y. Some basic operations on functions, e.g. inverse and composition, are applicable to binary relations as well. Given a fuzzy relation R(X,Y). Its domain domR is the fuzzy set on X whose membership function is defined for each x ∈ X as

domR(x) = max y∈Y

R(x,y),

i.e. each element of X belongs to the domain of R to a degree equal to the strength of its strongest relation to any y ∈ Y.

Range and Height

The range ran of R(X,Y) is a fuzzy relation on Y whose membership function is defined for each y ∈ Y as

ranR(y) = max x∈X

R(x,y),

i.e. the strength of the strongest relation which each y ∈ Y has to an x ∈ X equals to the degree of membership of y in the range of R. The height h of R(X,Y) is a number defined by

h(R) = max y∈Y

maxx∈X

R(x,y).

h(R) is the largest membership grade obtained by any pair (x,y) ∈ R.

Representation and Inverse

Consider e.g. the membership matrix R = [rxy] with rxy = R(x,y).

Its inverse R−1(Y,X) of R(X,Y) is a relation on Y × X defined by

R−1(y,x) = R(x,y), ∀x ∈ X, ∀y ∈ Y.

R−1 = [r−1 xy] representing R−1(y,x) is tje transpose of R for R(X,Y)

(R−1)−1 = R, ∀R.

Standard Composition

Consider the binary relations P(X,Y), Q(Y,Z) with common set Y. The standard composition of P and Q is defined as

(x,z) ∈ P ◦ Q ⇐⇒∃y ∈Y : {(x,y) ∈ P ∧ (y,z) ∈ Q}.

In the fuzzy case this is generalized by

[P ◦ Q](x,z) = sup y∈Y

min{P(x,y), Q(y,z)}, ∀x ∈ X, ∀z ∈ Z.

If Y is finite, sup operator is replaced by max. Then the standard composition is also called max-min composition.

Inverse of Standard Composition

The inverse of the max-min composition follows from its definition:

[P(X,Y) ◦ Q(Y,Z)]−1 = Q−1(Z,Y) ◦ P−1(Y,X).

Its associativity also comes directly from its definition:

[P(X,Y)] ◦ Q(Y,Z)] ◦ R(Z,W) = P(X,Y) ◦ [Q(Y,Z) ◦ R(Z,W)].

Note that the standard composition is not commutative. Matrix notation: [rij] = [pik] ◦ [qkj] with rij = maxkmin(pik,qkj).

Example

$$P \circ Q = R$$

$$\begin{bmatrix} .3 & .5 & .8 \\ 0 & .7 & 1 \\ .4 & .6 & .5 \end{bmatrix} \circ \begin{bmatrix} .9 & .5 & .7 & .7 \\ .3 & .2 & 0 & .9 \\ 1 & 0 & .5 & .5 \end{bmatrix} = \begin{bmatrix} .8 & .3 & .5 & .5 \\ 1 & .2 & 5 & .7 \\ .5 & .4 & .5 & .5 \end{bmatrix}$$

For instance:

r11 = max{min(p11,q11),min(p12,q21),min(p13,q31)} = max{min(.3,.9),min(.5,.3),min(.8,1)} = .8 r32 = max{min(p31,q12),min(p32,q22),min(p33,q32)} = max{min(.4,.5),min(.6,.2),min(.5,0)} = .4

Example: Types of Airplanes (Speed, Height, Type)

Consider the following fuzzy relations for airplanes: • relation A between maximal speed and maximal height, • relation B between maximal height and the type.

A	h_1	h_2	h_3
s_1	1	.2	0
s_2	.1	1	0
s_3	0	1	1
s_4	0	.3	1

B	t_1	t_2
h_1	1	0
h_2	.9	1
h_3	0	.9

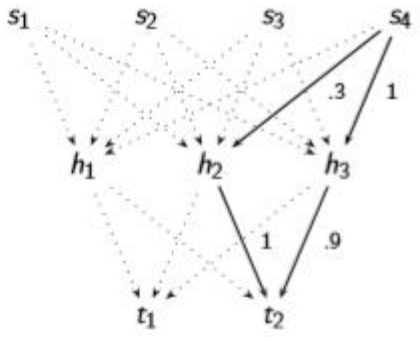
Example (cont.)

matrix multiplication scheme

A	$\circ$	B	1	0
			.9	1
			0	.9
1	.2	0	1	.2
.1	1	0	.9	1
0	1	1	.9	1
0	.3	1	.3	.9

$A \circ B$ speed-type relation

flow scheme



$$(A \circ B)(s_4, t_2) = \max\{\min\{.3,\ 1\},\ \min\{.1,\ .9\}\} = .9$$

Relational Join

A similar operation on two binary relations is the relational join. It yields triples (whereas composition returned pairs). For P(X,Y) and Q(Y,Z), the relational join $P * Q$ is defined by

$[P * Q](x,y,z) = \min\{P(x,y), Q(y,z)\}, \forall x \in X, \forall y \in Y, \forall z \in Z.$

Then the max-min composition is obtained by aggregating the join by the maximum:

$[P \circ Q](x,z) = \max_{y \in Y}$

$[P * Q](x,y,z), \forall x \in X, \forall z \in Z.$

Example

The join $S = P * Q$ of the relations P and Q has the following membership function (shown below on left-hand side). To convert this join into its corresponding composition $R = P \circ Q$ (shown on right-hand side), the two indicated pairs of S(x,y,z) are aggregated using max.

x	y	z	$\mu_S(x,y,z)$
1	a	α	.6
1	a	β	.7*
1	b	β	.5*
2	a	α	.6
2	a	β	.8
3	b	β	1
4	b	β	.4*
4	c	β	.3*

x	z	$\mu_R(x,z)$
1	α	.6
1	β	.7
2	α	.6
2	β	.8
3	β	.1
4	β	.4

For instance,

$R(1,\beta) = \max\{S(1,a,\beta), S(1,b,\beta)\} = \max\{.7, .5\} = .7$

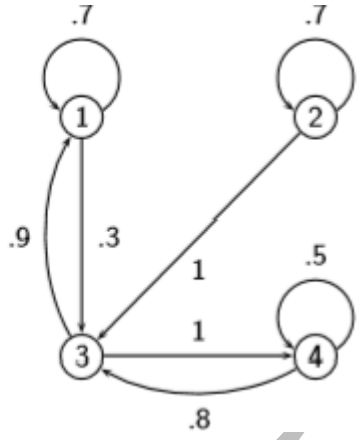
Binary Relations on a Single Set

It is also possible to define crisp or fuzzy binary relations among elements of a single set X. Such a binary relation can be denoted by R(X,X) or R(X2) which is a subset of $X \times X = X2$. These relations are often referred to as directed graphs which is also an representation of them. • Each element of X is represented as node. • Directed connections between nodes indicate pairs of $x \in X$ for which the grade of the membership is nonzero. • Each connection is labeled by its actual membership grade of the corresponding pair in R.

Example

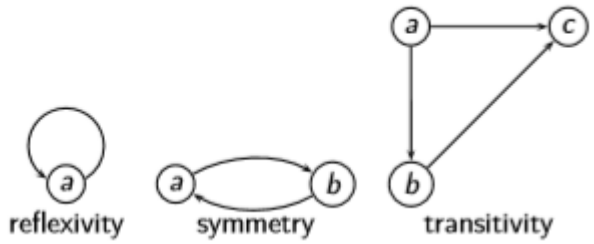
An example of R(X,X) defined on X = 1,2,3,4. Two different representation are shown below.

	1	2	3	4
1	.7	0	.3	0
2	0	.7	1	0
3	.9	0	0	1
3	0	0	.8	.5



Properties of Crisp Relations

A crisp relation R(X,X) is called • reflexive if and only if ∀x ∈ X : (x,x) ∈ R, • symmetric if and only if ∀x,y∈ X : (x,y) ∈ R ↔ (y,x) ∈ R, • transitive if and only if (x,z) ∈ R whenever both (x,y) ∈ R and (y,z) ∈ R for at least one y ∈ X.



Properties of Fuzzy Relations

These properties can be extended for fuzzy relations. So one can define them in terms of the membership function of the relation. A fuzzy relation R(X,X) is called • reflexive if and only if ∀x ∈ X : R(x,x) = 1, • symmetric if and only if ∀x,y∈ X : R(x,y) = R(y,x), • transitive if it satisfies

R(x,z) ≥ max y∈Y

min{R(x,y), R(y,z)}, ∀(x,z) ∈ X2.

Note that a fuzzy binary relation that is reflexive, symmetric and transitive is called fuzzy equivalence relation.

R. Kruse, C. Moewes FS – Fuzzy Relations

· Representation of a relation

$R(\bar{X}_1,...,\bar{X}_n)$

$(r_{i_1,i_2,...,i_n})$: n-D membership array

$r_{i_1,i_2,...,i_n}$ = 1 iff $(x_1,...,x_n) \in R$

0 Otherwise

◯ Example 3.1 :

$R(\bar{X}\bar{Y})Z)Y_1Y_2Y_3Y_4Z_2Z_1Z_3Z_4Z_5$

$R(\{\bar{X}_1,\bar{X}_3\}\cdots,\bar{X}_i\ J \subseteq N_n Y \le X$

$\{\bar{X}_i \mid j \in J \subseteq N_n\} \Rightarrow R_2 = \frac{1}{X,a,*}+\frac{1}{X,a,\$}+\frac{1}{Y,b,*}+\frac{1}{Y,a,\$}+\frac{0.8}{X,b,*}+\frac{0.8}{X,b,\$}+\frac{0.8}{Y,b,*}+\frac{0.8}{Y,b,\$}$

$\Rightarrow y = \{\bar{X}_1,\bar{X}_3\}$

$Y_j = X_j \forall j \in J$

$[R \uparrow X - Y] : [R_2 \uparrow \{\bar{X}_1,\bar{X}_3\}]$

$\Rightarrow y = \{\bar{X}_2\}, X - Y = \{\bar{X}_1,\bar{X}_3\} = \{(*,*),(x,\$),(Y,*),(Y,s)\}$

$(x) = R\ (y) = [R \downarrow \{\bar{X}_i\}](Y) = \max_{x \succ y} R(x)R_{ij}$

$< Y_j \mid j \in J > \in \underset{j\in J}{X}\ \bar{X}_j$

$R_{2,3} = \frac{1}{a,*}+\frac{0.7}{a,\$}+\frac{0.4}{b,*}+\frac{0.8}{b,\$}.$

$\frac{0.9}{X,a,*}+\frac{0.4}{X,b,*}+\frac{1}{Y,a,*}+\frac{0.7}{Y,a,\$}+\frac{0.8}{Y,b,\$}$

$\Rightarrow R_{1,2} = \frac{0.9}{X,a}+\frac{0.4}{X,b}+\frac{1}{Y,a}+\frac{0.8}{Y,b}\ R_{1,3} = \frac{0.9}{X,*}+\frac{0}{X,\$}+\frac{1}{Y,*}+\frac{0.8}{Y,\$}$

$0 \le R(x_1,x_2,...,x_n) \le 1 \frac{1}{(NY,Beijing)}+\frac{0}{(NY,NY)}+\frac{0.6}{(NY,London)}+\frac{0.9}{(Paris,Beijing)}+\frac{0.7}{(Paris,NY)}+(Pa$

$\bar{X}$ ={English , French} , $\bar{Y}$ ={ dollar , pound , franc , mark}

Z={US , France , Canada , Britain , Germary}

$R(\bar{X}\,\bar{Y}\,Z)$ ={(English , dollar , US) , (French , franc , France)

(English , dollar , Canada) , (French , dollar , Canada)

(English , pound , Britain)}

Y_1 Dollar 1 0 1 0 0 Dollar 0 0 1 0 0

Y_2 Pound 0 0 0 1 0 Pound 0 0 0 0 0

Y_3 Franc 0 0 0 0 0 Franc 0 1 0 0 0

Y_4 Mark 0 0 0 0 0 Mark 0 0 0 0 0

US Fran Can Brit Ger US Fran Can Brit Ger

$Z_1\ Z_2\ Z_3\ Z_4\ Z_5\ Z_1\ Z_2\ Z_3\ Z_4\ Z_5$

English Franch

$X_1\ X_2$

· **Fuzzy Relations**

Cartesian Product : $\bar{X}_1 \times \bar{X}_2 \times \cdots \times \bar{X}_n$

tuples : $(x_1, x_2, ..., x_n)$

membership grade :

$0 \le R(x_1, x_2, ..., x_n) \le 1$

Example 3.2: Binary relation R : represents the concept " very far"

$\bar{X}$ = { New York , Paris}

$\bar{Y}$ ={Beijing , New York , London}

Relation in list notation

$R(\bar{X}\,\bar{Y})$ =

$$\frac{1}{(NY, Beijing)} + \frac{0}{(NY, NY)} + \frac{0.6}{(NY, London)} + \frac{0.9}{(Paris, Beijing)} + \frac{0.7}{(Paris, NY)} + \frac{0.3}{(Paris, Londom)}$$

Relation in membership array

	NY	Paris
Beijing	1	0.9
NY	0	0.7
London	0.6	0.3

· Ordinary fuzzy relation

with valuation set [0,1]

L-fuzzy relation

With ordered valuation set L

3.2 Projection and Cyclindric Extensions

· set family X = $\{\underline{\bar{X}}_i \mid i \in N_n\}$

Let X = $< x_i \mid i \in N_n > \in \underset{j\in J}{X} \underline{\bar{X}}_i$

Let Y = $< Y_j \mid j \in J > \in \underset{j\in J}{X} \underline{\bar{X}}_j$

Where $J \subseteq N_n$, | J | = r

Y a subsequence of X , $Y \leq X$

iff $Y_j = X_j \ \forall j \in J$

⊙Projection : $[R \downarrow y]$ the projection of R on Y

$R(\underline{\bar{X}}_1, \underline{\bar{X}}_2, \cdots, \underline{\bar{X}}_n)$: a relation

Y=$\{\underline{\bar{X}}_i \mid j \in J \subseteq N_n\}$

$[R \downarrow y]$: a fuzzy relation (set)

$[R \downarrow y](Y) = \max_{x \succ y} R(x)$

※max can be generalized by other t-conorms

· Example 3.3 $\bar{\bar{X}}_1$ ={X,Y}, $\bar{\bar{X}}_2$ ={a,b}, $\bar{\bar{X}}_3$ ={*,$}

$$R(\bar{\bar{X}}_1, \bar{\bar{X}}_2, \bar{\bar{X}}_3) = \frac{0.9}{X,a,*} + \frac{0.4}{X,b,*} + \frac{1}{Y,a,*} + \frac{0.7}{Y,a,\$} + \frac{0.8}{Y,b,\$}$$

Let $R_{ij} = [R \downarrow \{\bar{\bar{X}}_i, \bar{\bar{X}}_j\}]$, $R_i = [R \downarrow \{\bar{\bar{X}}_i\}]$

$$\Rightarrow R_{1,2} = \frac{0.9}{X,a} + \frac{0.4}{X,b} + \frac{1}{Y,a} + \frac{0.8}{Y,b}$$

$$R_{1,3} = \frac{0.9}{X,*} + \frac{0}{X,\$} + \frac{1}{Y,*} + \frac{0.8}{Y,\$}$$

$$R_{2,3} = \frac{1}{a,*} + \frac{0.7}{a,\$} + \frac{0.4}{b,*} + \frac{0.8}{b,\$}$$

$$R_1 = \frac{0.9}{*} + \frac{1}{y}$$

$$R_2 = \frac{1}{a} + \frac{0.8}{b}$$

$$R_3 = \frac{1}{*} + \frac{0.8}{\$}$$

⊙Cyclindric Extension $[R \downarrow X - Y]$the CE of R into X-Y

X-Y : sets $\bar{\bar{X}}_i$ that are in X but are not in Y

$$[R \downarrow X - Y](x) = R\ (y)$$

R: a relation defined on Y

·Example 3.4 (Refer to example 3.3)

Let X = $\{\bar{X}_1, \bar{X}_2, \bar{X}_3\}$

And R= $R_{1,2} \Rightarrow y = \{\bar{X}_1, \bar{X}_3\}$

∴ X-Y = $\bar{X}_3$ = {*,$}

From example 3.3 $R_{1,2} = \frac{0.9}{X,a} + \frac{0.4}{X,b} + \frac{1}{Y,a} + \frac{0.8}{Y,b}$

$\therefore [R \uparrow X - Y] = [R_{1,2} \uparrow \{\bar{X}_3\}] =$

$$\frac{0.9}{X,a,*} + \frac{0.9}{X,a,\$} + \frac{0.4}{X,b,*} + \frac{0.4}{X,b,\$} + \frac{1}{Y,a,\$} + \frac{1}{Y,a,\$} + \frac{0.8}{Y,b,*} + \frac{0.8}{Y,b,\$}$$

$$[R \uparrow X - Y] : [R_{12} \uparrow \{\bar{X}_3\}][R_{13} \uparrow \{\bar{X}_2\}][R_{23} \uparrow \{\bar{X}_1\}][R_1 \uparrow \{\bar{X}_2, \bar{X}_2\}][R_2 \uparrow \{\bar{X}_1, \bar{X}_3\}]$$

$$[R_3 \uparrow \{\bar{X}_1, \bar{X}_2\}]$$

Consider $[R \uparrow X - Y] = [R_2 \uparrow \{\bar{X}_1, \bar{X}_3\}]$

$\Rightarrow y = \{\bar{X}_2\}, X - Y = \{\bar{X}_1, \bar{X}_3\} = \{(*,*),(x,\$),(Y,*),(Y,s)\}$

{x,y} {x,$}

R= $R_2 = \frac{1}{a} + \frac{0.8}{b}$

$$\therefore [R_2 \uparrow \{\bar{X}_1, \bar{X}_3\}] = \frac{1}{X,a,*} + \frac{1}{X,a,\$} + \frac{1}{Y,b,*} + \frac{1}{Y,a,\$} + \frac{0.8}{X,b,*} + \frac{0.8}{X,b,\$} + \frac{0.8}{Y,b,*} + \frac{0.8}{Y,b,\$}$$

3.4 Cyclindric closure

-A relation may be exactly reconstructed from several of its projections by taking the set intersection of their cyclindricextensions $\{P_i \mid i \in I\}$:a set of projections of a relation on X

$$\Rightarrow cyl\{P_i\}(X) = \min_{i \in I}[P_i \uparrow X - Y_i](X) \geq R$$

Y_i ：The family of sets on which P_i is defined.

- Example：

$$Cyl\{R_{1,2}, R_{1,3}, R_{2,3}\} = \frac{0.9}{x,a,*} + \frac{0.4}{x,b,*} + \frac{1}{y,a,*} + \frac{0.7}{y,a,\$} + \frac{0.4}{y,b,*} + \frac{0.8}{y,b,*}$$

Refer to the original relation $R(\bar{X}_1, \bar{X}_2, \bar{X}_3)$ in example 3.3.

It is not fully reconstructable from its projections become of ignoramus of R_1 , R_2 ,and R_3 .

3.3. Binary Relations R($\bar{X}, \bar{Y}$)

$\bar{X} \neq \bar{Y}$ ：bipartite graph

$\bar{X} = \bar{Y}$ ：directed graph

- Representations

i, matrices $R = [r_{ij}]$, where $r_{ij} = R(x_i, y_j)$

ii, sagittal diagrams

Examples：

i)

	y_1	y_2	y_3	y_4	y_5
x_1	.9	1	0	0	0
x_2	0	.4	0	0	0
x_3	0	0	1	.2	0
x_4	0	0	0	0	.4
x_5	0	0	0	0	.5
x_6	0	0	0	0	.2

ii)

3.9

- Domain：dom R

Crisp – dom R = $\{x \in X \mid (x, y) \in R, \exists y \in Y\}$

Fuzzy – dom R(x) = $\max_{y \in Y} R(x, y)$

The domain of a fuzzy relation R(x,y) is a fuzzy set on X; dom R(x) is its membership function.

e.g. dom R(X_1) = max(0.9, 1) = 1

- Range：ran R

Crisp – ran R = $\{y \in Y \mid (x, y) \in R, \exists x \in X\}$

Fuzzy – ran R(y) = $\max_{x \in X} R(x, y)$

e.g. ran R(y_5) = max(0.4, 0.5, 0.2) = 0.5

・Height： $h(R) = \max_{y \in Y} \max_{x \in X} R(x, y)$

e.g., $h(R) = 1$ normal fuzzy relation

・Inverse： $R^{-1}(Y, X)$

$R^{-1}(y, x) = R(x, y)$

$\therefore R^{-1} = R^T, (R^{-1})^{-1} = R$

e.g. $R = \begin{bmatrix} 0.3 & 0.2 \\ 0 & 1 \\ 0.6 & 0.4 \end{bmatrix}$

$R^{-1} = R^T = \begin{bmatrix} 0.3 & 0 & 0.6 \\ 0.2 & 1 & 0.4 \end{bmatrix}$

・ Composition： $R(X, Z) = P(X, Y) \circ Q(Y, Z)$

$R(x, z) = [P \circ Q](x, z) = \max_{y \in Y} \min[P(x, y), Q(y, z)]$

Max-min composition

Properties： $\begin{cases} P \circ Q \neq Q \circ P \\ (P \circ Q)^{-1} \neq Q^{-1} \circ P^{-1} \\ (P \circ Q) \circ R = P \circ (Q \circ R) \end{cases}$

Matric form： $[r_{ij}] = [p_{ik}] \circ [q_{kj}]$

Where $[r_{ij}] = \max_k \min(p_{ik}, q_{kj})$

3.11

$$R(x,z)=[P\circ Q](x,z)=\max_{y\in Y}[P(x,y)\bullet Q(y,z)]$$

max-product composition

matrix form $[r_{ij}]=[p_{ik}]\circ[q_{kj}]$

Where $[r_{ij}]=\max(p_{ik},q_{kj})$

- Example

$$\begin{bmatrix}0.3 & 0.5 & 0.8\\ 0.0 & 0.7 & 1.0\\ 0.4 & 0.6 & 0.5\end{bmatrix}\circ\begin{bmatrix}0.9 & 0.5 & 0.7 & 0.7\\ 0.3 & 0.2 & 0.0 & 0.9\\ 1.0 & 0.0 & 0.5 & 0.5\end{bmatrix}$$

Max-min = $\begin{bmatrix}0.8 & 0.3 & 0.5 & 0.5\\ 1.0 & 0.2 & 0.5 & 0.7\\ 0.5 & 0.4 & 0.5 & 0.6\end{bmatrix}$

Max-prod = $\begin{bmatrix}0.8 & 0.15 & 0.4 & 0.45\\ 1.0 & 0.14 & 0.5 & 0.63\\ 0.5 & 0.2 & 0.28 & 0.54\end{bmatrix}$

- Relational join： $R(X,Y,Z)=P(X,Y)*Q(Y,Z)$

$$R(x,y,z)=[P*Q](x,y,z)=\min[P(x,y),Q(y,z)]$$

※The max-min composition can be obtained by aggregating appropriate elements of the corresponding join.

3.12・Example

$$※[P \circ Q](x,z) = \max_{y \in Y}[P * Q](x,y,z)$$

5.4Binary Relation on a Simple Set

・Representations

3-13

◎ characteristic Properties (Crisp case)

i, reflexive

irrflexive ∃ ↓

antiflexive ∀ ↓

ii, symmetric

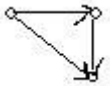
asymmetric ↓ ∃

antisymmetric ↓ $\forall(x,y)\in R,(y,x)\in R \Rightarrow x=y$

strictly antisymmetric $\forall x\neq y,(x,y)\in R$ or $(y,x)\in R$ ↓

iii, transitive

nontransitive ↓ ∃



antitransitive ∀ ↓

◎ Fuzzy Relations

i, reflexive --- $\forall x,\ R(x,x)=1$

irreflexive --- $\exists x,\ R(x,x)\neq 1$

antiflexive --- $\forall x,\ R(x,x)\neq 1$

ε-reflexive --- $\forall x,\ R(x,x)\geq \varepsilon$

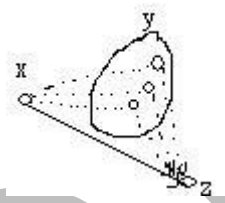
ii, symmetric -- $\forall x, y,\ R(x,x) = R(y,x)$

asymmetric -- $\exists x, y,\ R(x,x) \neq R(y,x)$

antisymmetric -- $\begin{cases} R(x,x) > 0 \\ R(y,x) > 0 \end{cases} \Rightarrow x = y$

iii, max-min transitive -- $\forall x, z$

$$R(x,z) \geq \max_{y \in \bar{Y}} \min\left[R(x,y), R(y,z)\right]$$



max-product transitive -- $\exists x, z$

$$R(x,z) \geq \max_{y \in \bar{Y}} \min\left[R(x,y) \bullet R(y,z)\right]$$

nontransitive -- $\exists x, z$

$$R(x,z) < \max_{y \in \bar{Y}} \min\left[R(x,y), R(y,z)\right]$$

antitransitive -- $\forall x, z$

$$R(x,z) < \max_{y \in \bar{Y}} \min\left[R(x,y), R(y,z)\right]$$

◎ Example 3.7 R:very near

$\Rightarrow$ reflexive, symmetric, nontransitive

3-15

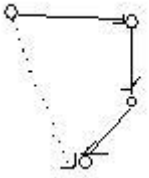
◎Summary

	Reflexive	Antireflexive	Symmetric	Antisymmetric	Transitive
Crisp: equivalence; Fuzzy: similarity	■		■		■
Quasi-equivalence			■		■
Compatibility or Tolerance	■		■		
Partial ordering	■			■	■
Preordering or Quasi-ordering	■				■
Strict ordering		■		■	■

Figure3.6 Some important types of binary relation R(X,X)

◎ transitive closure: $R_T(\bar{X})$

Algorithm for computing R_T



1. $R' = R \cup (R \circ R)$
2. If $R' \neq R$, Let $R' = R$, go to step1
3. Stop, $R_T = R'$

Where $\cup$:component-wise max

◎Example 3.8

$$R=\begin{bmatrix}0.7 & 0.5 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0 & 1.0\\ 0.0 & 0.4 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.8 & 0.0\end{bmatrix}$$

Step1: $R\circ R=\begin{bmatrix}0.7 & 0.5 & 0.0 & 0.5\\ 0.0 & 0.0 & 0.8 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.4\\ 0.0 & 0.4 & 0.0 & 0.0\end{bmatrix}$

$$R\cup(R\circ R)=\begin{bmatrix}0.7 & 0.5 & 0.0 & 0.5\\ 0.0 & 0.0 & 0.8 & 1.0\\ 0.0 & 0.4 & 0.0 & 0.4\\ 0.0 & 0.4 & 0.8 & 0.0\end{bmatrix}=R'$$

Step2: $\because R'\neq R$, Let $R=R'$

repeat step1

$$R\circ R=\begin{bmatrix}0.7 & 0.5 & 0.5 & 0.5\\ 0.0 & 0.4 & 0.8 & 0.4\\ 0.0 & 0.4 & 0.4 & 0.4\\ 0.0 & 0.4 & 0.4 & 0.4\end{bmatrix}$$

$$R\cup(R\circ R)=\begin{bmatrix}0.7 & 0.5 & 0.5 & 0.5\\ 0.0 & 0.4 & 0.8 & 1.0\\ 0.0 & 0.4 & 0.4 & 0.4\\ 0.0 & 0.4 & 0.8 & 0.0\end{bmatrix}=R'$$

Step3: $\because R'\neq R$, Let $R=R'$

repeat step1

$$R'=\begin{bmatrix}0.7 & 0.5 & 0.5 & 0.5\\ 0.0 & 0.4 & 0.8 & 1.0\\ 0.0 & 0.4 & 0.4 & 0.4\\ 0.0 & 0.4 & 0.8 & 0.4\end{bmatrix}=R$$

Step4: Stop

$$R_T = R^{/}$$

3-17

3.5 Fuzzy Equivalence Relation

◎Crisp binary relation

equivalence: reflexive, symmetric, and transitive

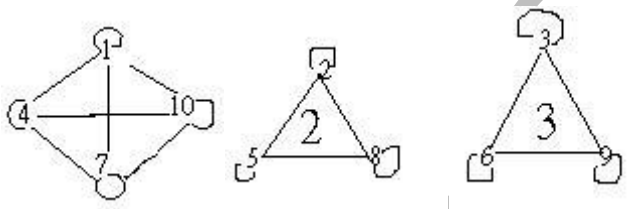
equivalence classes

partition: X/R

◎ Example 3.9:

$\bar{X} = \{1, 2, \cdots, 10\}$

$R(\bar{X} \times \bar{X}) = \{(x, y) \mid x, y$ have the same remainder when divided by 3$\}$



R: reflexive, symmetric, transitive

⇨ equivalence

partition $\bar{X}/R = \{(1, 4, 7, 10), (2, 5, 8), (3, 6, 9)\}$

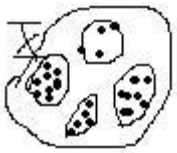
◎Fuzzy Binary Relation

。 Fuzzy

Similarity relation ~~equivalence~~ relation

Similarity classes ~~equivalence~~ classes

2 Interpretations of a similarity relation:



1.Group similar elements into crisp classes

whose members are similar to each other

to some specified degree.

2. $\forall x \in \bar{X}$, associate a fuzzy set A_x

defined on $\bar{X}$.

。 a fuzzy relation $R = \bigcup_{\alpha \in [0,1]} \alpha \square^{\alpha} R$ (Theorem 2.5, Eqs.(2.1)(2.2))

If R: Similarity relation,

$\Rightarrow {}^{\alpha}R$: equivalence relation

。 Let $\Pi({}^{\alpha}R)$: the partition of $\bar{X}$ w.r.t. ${}^{\alpha}R$

$\Rightarrow \Pi(R) = \{\Pi({}^{\alpha}R) \mid \alpha \in (0,1]\}$

$\Pi({}^{\alpha}R)$: nested, i.e.,

$\Pi({}^{\alpha}R)$: aredefinement of $\Pi({}^{\beta}R)$ iff $\alpha \geq \beta$

Prove that :A fuzzy relation $R: X \to X$ is a similarity relation, then ${}^{\alpha}R$ is a equivalent relation

Pf :∵ R : a similar relation

∴R : reflexive, i.e., $\forall x \in X, R(x,x) = 1$

symmetric, i.e., $\forall x, y \in X, R(x,y) = R(y,x)$

transitive, i.e., $\forall x, z \in X^2, R(x,z) \geq \max_{y \in Y}[R(x,y), R(y,z)]$

i, ${}^{\alpha}R$: reflexive

∵ $\forall x \in X, R(x,x) = 1 \geq \alpha \in [0,1], (x,x) \in {}^{\alpha}R$

${}^{\alpha}R$: reflexive

ii, ${}^{\alpha}R$: symmetric

∵R : symmetric

$\forall x, y \in Z, R(x,y) = R(y,x)$

Let $R(x,y) = R(y,x) = \beta$

Then $\beta \geq \alpha$ or $\beta < \alpha$

a, if $\beta \geq \alpha$ => $(x,y),(y,x) \in {}^{\alpha}R$

b, if $\beta < \alpha$ => $(x,y),(y,x) \notin {}^{\alpha}R$

iii, ${}^{\alpha}R$: transitive

∵R : transitive

$\forall x, z \in X^2, R(x,z) \geq \max_{y \in Y}[R(x,y), R(y,z)]$

Let $R(x,y) = \beta_1$, $R(y,z) = \beta_2$

Assume $\beta_1 < \beta_2$

Then $\alpha \leq \beta_1 < \beta_2$, $\beta_1 < \alpha < \beta_2$, or $\beta_1 < \beta_2 \leq \alpha$

a. if $\alpha \leq \beta_1 < \beta_2$ => $(x,y)^{\alpha}R, (y,z) \in {}^{\alpha}R$ --- (A)

$\because R(x,z) \geq \max_{y \in Y}[R(x,y), R(y,z)] \geq \min[\beta_1, \beta_2] = \beta_1 \geq \alpha$

$\therefore R(x,z) \geq \alpha, (x,z) \in {}^{\alpha}R$ --- (B)

(A) , (B) => ${}^{\alpha}R$: *transitive*

b. if $\beta_1 < \alpha < \beta_2$

$\Rightarrow (x,y) \notin {}^{\alpha}R, (y,z) \in {}^{\alpha}R$, don't care (x,z)

c. if $\beta_1 < \beta_2 \leq \alpha$

$\Rightarrow (x,y) \notin {}^{\alpha}R, (y,z) \notin {}^{\alpha}R$, don't care (x,z)

Example 3.10 : $R(X, X)$: a fuzzy relation

R : reflexive , symmetric , transitive ($\because R' = R \cup (R \circ R) = R$)

$\because$level set : $\Lambda_R = \{0.0, 0.4, 0.5, 0.8, 0.9, 1.0\}$

There are five nested partition ${}^{\alpha}\pi's$

- The similarity class for each element is a fuzzy set defined by the row of the membership matrix corresponds to that element

Example : see Example 3.10

For c : $\frac{0}{a} + \frac{0}{b} + \frac{1}{c} + \frac{0}{d} + \frac{1}{e} + \frac{0.9}{f} + \frac{0.5}{g}$

For e : $\frac{0}{a}+\frac{0}{b}+\frac{1}{c}+\frac{0}{d}+\frac{1}{e}+\frac{0.9}{f}+\frac{0.5}{g}$

∴c and e are similar at any level α

3.6 Compatibility Relations ---- reflexive , symmetric

compatibility

Alternatives : tolerance } relation

proximity

- Crisp case :
  Maximal compatibility classes – not properly contained within any other compatibility class

  Complete cover – all the maximal compatibility classes

- Fuzzy case :
  α-compatibility class ---- a subset A of X ,

  s.t. $\forall x, y \in A, if\ \exists R(x, y) => R(x, y) \geq \alpha, R$: fuzzy compatibility relation

maximalα ---- compatibility classes

completeα-cover

Example 3.11 : $R(X, X)$: a fuzzy relation

∵R : reflexive , symmetric

∴a compatibility relation

∵ $A_R = \{0.0, 0.4, 0.5, 0.7, 0.8, 1.0\}$

=>the completeα-covers

$\partial = 0.5$

$\frac{0.7}{b} + \frac{0.9}{d}$

$$\begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix}\begin{bmatrix} 1 & 0.7 & 0 & 1 & 0.7 \\ 0 & 1 & 0 & 0.9 & 0 \\ 0.5 & 0.7 & 1 & 1 & 0.8 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.1 & 0 & 0.9 & 1 \end{bmatrix}$$

$$^{0.5}R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e.q. $\therefore$ $\partial = 0.4$

$\Rightarrow x \le y \Rightarrow xx = y \forall y \ne xx \in X(x,y) \rightarrow \le \in XS = \{x_1, x_2\} \subseteq Xy \in Ax \ge y$

$(x,y) y \in X \exists (x \le y, or \Leftrightarrow y \le xAx \in XA \subseteq Xx \ne yR_{\le [x]}(y) = R(y,x)$

$x \in U(R,A)(x) = \bigcap_{x \in A} R_{\ge [x]} \forall y \in$

$$^{0.4}R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Look for complete subgraphs

(1,2) , (3,4,5),(4,5,6,7),(5,8),(9)

(34,),(4,5,6),(4,5,7),(3,5),(5,6)

(4,5),(5,6,7),(4,6,7),(4,6),(6,7)

∴ maximal compatible classes (the complete 0.4-cover):

(1,2),(3,4,5),(4,5,6,7),(5,8),(9)

These do not partition X.

e.q. $\partial = 0.5$

$$^{0.5}R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ maximal compatible classes

(The complete 0.5-cover)

(1,2),(3,4,5),(4,5,6),(4,5,7),(5,8),(9)

3.7. Ordering Relations

· partial ordering: reflexive , antisummetric , transitive

$X \leq Y$: X : predecessor

precedes Y : successor

if exist First member : if $x \leq y \ \forall y \in X$ (minimum)

unique Last member : if $y \leq x \ \forall y \in X$ (maximam)

may not Minimal member : if $y \leq x \Rightarrow x = y$

be unique Maximal member : if $x \leq y \Rightarrow x = y$

·properties :

1, if $\exists$, at most one first member

if $\exists$, at most one last member

2, There may be several maximal and minimal member

3, if $\exists$ a first member X , $\Rightarrow$ only one minimal member Y exists and x=y

4, if $\exists$ a last member x , $\Rightarrow$ only one maximal member Y exists and x=y.

5, partial the first member $\Leftrightarrow$ the last member inverse

ordering the last member $\Leftrightarrow$ the first member partial ordering

※ In a partial ordering , it does not guarantee that $\forall(x,y)$, $\exists(x \le y, or\ y \le x)$.

If $\exists$, $\Rightarrow$ (x,y) : comparable (total ordering)

Otherwise (x,y) : non comparable

· $A \subseteq X$

If $x \in X$, and $\forall y \in A$, $x \le y$,

$\Rightarrow$ x: lower bound of A on X

If …… , $x \ge y$

$\Rightarrow$ x : upper bound of A on X

·greatest lower bound (orinfimum) GLB

- a lower bound which succeeds every other lower bound

Least upper bound (orsupermum) LUB

- a upper bound which preceeds every other upper bound

·Lattice – A partial ordering on X contains GLB and LUB , $\forall S = \{x_1, x_2\} \subseteq X$

· Connected – a partial ordering is said to be connected

iff $\forall x, y \in X$, $x \neq y \Rightarrow$ x<y or y>x

· Linear ordering (total ordering , simple ordering , complete ordering)

- when a partial ordering is connected , then $\forall(x, y)$: comparable

· Hasse diagrams – representing partial orderings in which $\rightarrow$ indicates $\leq$

· Example 3.12 : Crisp partial orderings

· Fuzzy partial ordering

- reflexive , antisymmetric , and transitive under some form of transitivity.

※ any fuzzy partial ordering can be resolved into a series of crisp partial ordering .

i.e. taking a series of ∂ cut that produce increasing levels of refinement

· In a fuzzy partial ordering , R

$\forall x \in X$, twofuzzy sets are associated with

$R_{\geq[x]}$: dominating class

$R_{\geq[x]}(y) = R(x, y)$

$R_{\leq[x]}$: dominated class

$R_{\leq[x]}(y) = R(y, x)$

· xundominatediff R(x,y) = 0 $\forall y \neq x$

X undominatingiff R(y,x) = 0 $\forall y \neq x$

· Fuzzy upper bound for $A \subseteq X$ is a fuzzy set

$$U(R, A) = \bigcap_{x \in A} R_{\geq [x]}$$

※ If a least upper bound of A exists , it is the unique element $x \in U(R, A)$

s.t. 1, 2,

$U(R, A)(x) > 0$ R(x,y) > 0 ,

$\forall y \in$ support [U(R,A)]

· Example 3.13

a b c d e

Fuzzy partial ordering R:

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 1 & 0.7 & 0 & 1 & 0.7 \\ 0 & 1 & 0 & 0.9 & 0 \\ 0.5 & 0.7 & 1 & 1 & 0.8 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.1 & 0 & 0.9 & 1 \end{bmatrix}$$

1. row : dominating class for each element
   column : dominated class for each element
2. d : undominated , C : undominating
3. For A = {a,b} , U(R,A) = the intersection of
   The dominating classes of a and b = $\frac{0.7}{b} + \frac{0.9}{d}$

4, LUB(A) =b

3-32

3. Crisp ordering captured by the fuzzy ordering

e.g. $\alpha = 0.5$

$$ {}^{\alpha}R \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} $$

is → 2 3 1 5 3

※The ordering become weaken with the increasing α

5-33

Fuzzy preordering – reflexive and transitive

Fuzzy weak ordering –

i, an ordering satisfying the proportion of a fuzzy total ordering except antisymmetry.

ii, a fuzzy preordering in which $\forall x \neq y$, either R(x,y)>0 or R(y,x)>0

Fuzzy strict ordering –

Antireflexive

Antisymmetric

Transitive

3.8. Morphisms

・Crisp homomorphism h from (X,R) to (Y,Q)

Where R(X,X), Q(Y,Y)：binary relations

$$\forall(x_1, x_2) \in R \Rightarrow (h(x_1), h(x_2)) \in Q$$

- Fuzzy homomorphism h

If R(X,X), Q(Y,Y)：Fuzzy binary relations

And $R(x_1, x_2) \leq Q[h(x_1), h(x_2)]$

※ It's possible that a relation $(h(x_1), h(x_2)) \in Q$ which $(x_1, x_2) \notin R$.

※ If this is never the case h is called a strong homomorphism.

3-34

・Crisp strong homomorphism h

If $(x_1, x_2) \in R \Rightarrow (h(x_1), h(x_2)) \in Q$

And $(y_1, y_2) \in Q \Rightarrow (h^{-1}(y_1), h^{-1}(y_2)) \in R$

※ where h : many to one → $h^{-1}(y)$ contains a set of Xs

・Fuzzy strong homomorphism h

H imposes a partition Π_h on X

Let

$A = \{a_1, a_2, \cdots, a_n\}$

$B = \{b_1, b_2, \cdots, b_n\} \in \Pi_h$

R,Q：fuzzy relations

h : strong homomorphism

iff $\max_{i,j}(R(a_i, b_j)) = Q(y_1, y_2)$

where $\begin{cases} y_1 = h(a_i) \forall a_i \in A \\ y_2 = h(b_j) \forall b_j \in B \end{cases}$

3-33

・Example 3.14

R(X,X)

$$R=\begin{bmatrix}0 & 0.5 & 0 & 0\\ 0 & 0 & 0.9 & 0\\ 1 & 0 & 0 & 0.5\\ 0 & 0.6 & 0 & 0\end{bmatrix}$$

Q(Y,Y)

$$Q=\begin{bmatrix}0.5 & 0.9 & 0\\ 1 & 0 & 0.9\\ 1 & 0.9 & 0\end{bmatrix}$$

→h：ordinary fuzzy homomorphism (one way)

$\because \forall R(x_1,x_2)\le Q(h(x_1),h(x_2)) \leftarrow strong$

But $Q(\gamma,\beta)=0.9$ $R(d,c)=0$

i,e, $(\gamma,\beta)\in Q$ while $(d,c)\notin R$ where $h(d)=\gamma, h(c)=\beta$

3-36

R(X,X)

$$\begin{bmatrix} 0.8 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.9 & 0.5 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.8 \end{bmatrix}$$

Q(Y,Y)

$$\begin{bmatrix} 0.7 & 0 & 0.9 \\ 0.4 & 0.8 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

→h：strong fuzzy homomorphism (two way)

3-37

※Q represents a simplification of R

・Isomorphism : (congruence)

h:1-1, onto $\Rightarrow X \neq Y$

Endomorphism : (subgraph)

h:X→Y, $Y \subseteq X$

Automorphism :

Isomorphism and End Endomorphism

i.e.m X=Y nad R=Q

3-38

3.9 SUP-i Compositions of Fuzzy Relations Generalize max-min Composition

i : t-norm

sup : t-conorm

- P(X,Y), Q(Y,Z)：fuzzy relations

$P \overset{i}{o} Q(X \times Z)$:sup-i composition

$$[P \overset{i}{o} Q](X, Z) = \sup_{y \in Y} i[P(x, y), Q(y, z)]$$

- Properties

1. $(P \overset{i}{o} Q) \overset{i}{o} R = P \overset{i}{o} (Q \overset{i}{o} R)$
2. $P \overset{i}{o} (\cup_j Q_j) = \cup_j (P \overset{i}{o} Q_j)$
3. $P \overset{i}{o} (\cap_j Q_j) = \cap_j (P \overset{i}{o} Q_j)$
4. $(\cup_j P_j) \overset{i}{o} Q = \cup_j (P_j \overset{i}{o} Q)$
5. $(\cap_j P_j) \overset{i}{o} Q = \cap_j (P_j \overset{i}{o} Q)$
6. $(P \overset{i}{o} Q)^{-1} = Q^{-1} \overset{i}{o} R^{-1}$

3. Show Eq.(3.16), i.e., $P \overset{i}{\circ}(\bigcap_{j\in J} Q_j) \subseteq \bigcap_{j\in J}(P \overset{i}{\circ} Q_j)$,

Where $P(\bar{X},\bar{Y})$ and $Q(\bar{Y},Z)$ are fuzzy relations.

pf. From Eq.(3.13), i.e., $\left[P \overset{i}{\circ} Q\right](x,z) = \sup_{y\in\bar{Y}} i\left[P(x,y), Q(y,z)\right]$

$$\therefore \left[P \overset{i}{\circ}(\bigcap_{j\in J} Q_j)\right](x,z) = \sup_{y\in\bar{Y}} i\left[P(x,y), \bigcap_{j\in J} Q_j(y,z)\right]$$

Let $Q = \bigcap_{j\in J} Q_j$

$$\Rightarrow Q \subseteq Q_1, Q \subseteq Q_2, \cdots, Q \subseteq Q_{|J|}$$

i.e., $\forall(y,z),\ Q(y,z) \le Q_1(y,z), \cdots, Q(y,z) \le Q_{|J|}(y,z)$

$\because$ *i* is monotonically increasing

$$\therefore \begin{cases} i[P(x,y), \bigcap_{j\in J} Q_j(y,z)] \le i[P(x,y), Q_1(y,z)] \\ \cdots\cdots\cdots \\ i[P(x,y), \bigcap_{j\in J} Q_j(y,z)] \le i[P(x,y), Q_{|J|}(y,z)] \end{cases} \quad \forall(x,y)$$

$$i[P(x,y), (\bigcap_{j\in J} Q_j)(y,z)] \le \bigcap_{j\in J} i[P(x,y), Q_j(y,z)]$$

$$\therefore \sup_{y\in\bar{Y}} i[P(x,y), (\bigcap_{j\in J} Q_j)(y,z)] \le \sup_{y\in\bar{Y}} \bigcap_{j\in J} i[P(x,y), Q_j(y,z)]$$

$$= \bigcap_{j\in J} \sup_{y\in\bar{Y}} i[P(x,y), Q_j(y,z)], \forall(x,y),(y,z)$$

$$\Rightarrow \left[P \overset{i}{\circ}(\bigcap_{j\in J} Q_j)\right](x,z) \le \left[\bigcap_{j\in J}(P \overset{i}{\circ} Q_j)\right](x,z), \forall(x,z)$$

i,e., $P \overset{i}{\circ}(\bigcap_{j\in J} Q_j) \subseteq \bigcap_{j\in J}(P \overset{i}{\circ} Q_j)$

。Sup-i composition monotonically increases

i.e., $P \overset{i}{\circ} Q_1 \subseteq P \overset{i}{\circ} Q_2 ----(5.20)$ if $Q_1 \subseteq Q_2$

$Q_1 \overset{i}{\circ} P \subseteq Q_2 \overset{i}{\circ} P ----(5.21)$

。Identity of $\overset{i}{\circ}$

$$E(x,y) = \begin{cases} 1, x = y \\ 0, x \neq y \end{cases} \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

i.e., $E \overset{i}{\circ} P = P \overset{i}{\circ} E = P$

。Relation *R*on $\underline{\underline{X}}^2$:*i*-transitive

iff $R(x,z) \geq i\left[R(x,y), R(y,z)\right], \forall x, y, z \in \underline{\underline{X}}$

$\Leftrightarrow R \overset{i}{\circ} R \subseteq R$

。*i*-transitive closure $R_{T(i)}$

--- The smallest *i*-transitive relation containing R

。Theorem 3.1: R: any fuzzy relation

$\Rightarrow R_{T(i)} = \bigcup_{n=1}^{\infty} R^{(n)}$, where

$R^{(n)} = R \overset{i}{\circ} R^{(n-1)}$

proof:

By (5.15) (5.17)

i, $R_{T(i)} \overset{i}{\circ} R_{T(i)} = \left(\bigcup_{n=1}^{\infty} R^{(n)}\right) \overset{i}{\circ} \left(\bigcup_{m=1}^{\infty} R^{(m)}\right) = \bigcup_{n=1}^{\infty} \bigcup_{m=1}^{\infty} \left(R^{(n)} \overset{i}{\circ} R^{(m)}\right) = \bigcup_{n,m=1}^{\infty} R^{(n+m)}$

$= \bigcup_{k=2}^{\infty} R^{(k)} \subseteq \bigcup_{k=1}^{\infty} R^{(k)} = R_{T(i)}$

i.e., $R_{T(i)}$: *i*-transitive ($\because R_{T(i)} \overset{i}{\circ} R_{T(i)} \subseteq R_{T(i)}$)

ii, Let S:*i*-transitive, $R \subseteq S$ (5.20)(5.21) monotonically increasing

$\Rightarrow R^{(2)} = R \overset{i}{\circ} R \subseteq S \overset{i}{\circ} S \subseteq S$

If $R^{(n)} \subseteq S$, (i-transitive)

$\Rightarrow R^{(n+1)} = R \overset{i}{\circ} R^{(n)} \subseteq S \overset{i}{\circ} S \subseteq S$

} mathematical induction

$\therefore R^{(k)} \subseteq S, \forall k$

$\therefore R_{T(i)} = \bigcup_{k=1}^{\infty} R^{(k)} \subseteq S$

i.e., $R_{T(i)}$: smallest

。Theorem 3.2: R: reflexive fuzzy relation on $\underline{\underline{X}}^2$, $\left|\underline{\underline{X}}\right| = n$

$\Rightarrow R_{T(i)} = R^{(n-1)} \Leftrightarrow \begin{pmatrix} R^{(m)} \subseteq R^{m+1} \\ R^n = R^{n-1} \end{pmatrix} \forall m$

proof :i, $\because R: reflexive,$

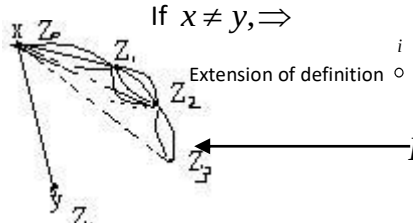
$\therefore E \subseteq R, R = E \overset{i}{\circ} R \subseteq R \overset{i}{\circ} R = R^{(2)}$

$\therefore R^{(n-1)} \subseteq R^{(n)}$ (By repetition)

ii, show $R^{(n-1)} \supseteq R^{(n)}$

proof: If $x = y, \Rightarrow R^{(n-1)}(x,x) = 1$

If $x \neq y, \Rightarrow$



reflexive

Extension of definition $\overset{i}{\circ}$

$$R^{(n)}(x,y) = \sup_{Z_1,\cdots,Z_{n-1}} i\left[R(x,z_1), R(z_1,z_2), \cdots, R(z_{n-1},y)\right]$$

$\because \left|\bar{\bar{X}}\right| = n$

$\Rightarrow X = Z_0, Z_1, \cdots, Z_n = y$ contains

at least 2 identical element.

Say $Z_r = Z_s (r < s)$

$\Rightarrow i\left[R(x,z_1), \cdots, R(z_{r-1}, z_r), \cdots, R(z_s, z_{s+1}), \cdots, R(z_{n-1}, y)\right] \leq R^{(k)}(x,y), (k \leq n-1)$

$\therefore \forall x, y \in \bar{\bar{X}}, R^{(n)}(x,y) \leq R^{(n-1)}(x,y),$

$\therefore R^{(n)} \subseteq R^{(n-1)} ---(B) \therefore R^{(n)} = R^{(n-1)} \leftarrow (A,B)$

$\therefore R_{T(i)} = R^{(n-1)}$

3.10 INF- W_i Compositions of Fuzzy Relations

。 W_i operation:

$a > b --> b$
$a \le b --> 1$

$$w_i(a,b) = \sup\{x \in [0,1] \mid i(a,x) \le b\} \longrightarrow$$

where $a, b \in [0,1]$, i : continuous t-norm

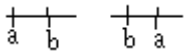
※ If i : logical conjunction (i.e., $\wedge$, and)

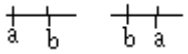
$\Rightarrow W_i$: logical implication (i.e., $\Rightarrow$, if then)

a	b	a⇒b
0	0	1
0	1	1
1	0	0
1	1	1

。 Theorem 3.3

1, $i(a,b) \le d$ iff $w_i(a,b) \ge b$

2, $w_i(w_i(a,b),b) \ge a$ 

3, $w_i(i(a,b),d) = w_i(a,w_i(b,d))$ 

4, $a \le b, \Rightarrow w_i(a,d) \ge w_i(b,d)$ ---- i

$w_i(d,a) \le w_i(d,b)$ --- ii

3, $i(w_i(a,b), w_i(b,d)) \le w_i(a,d)$

6, $w_i(\inf_j a_j, b) \ge \sup_j w_i(a_j, b)$

7, $w_i(\sup_j a_j, b) = \inf_j w_i(a_j, b)$

8, $w_i(b, \sup_j a_j) \ge \sup_j w_i(b, a_j)$

9, $w_i(b, \inf_j a_j) = \inf_j w_i(b, a_j)$

10, $i(a, w_i(a,b)) \le b$

proof: (1) i , If $i(a,b) \le d, \Rightarrow b \in \{x \mid i(a,x) \le d\}$

$(\Rightarrow) \Rightarrow b \le \sup\{x \mid i(a,x) \le d\} = w_i(a,d)$

ii, If $b \le w_i(a,d)$

i: continuous monotone

i: monotone (increasing)

$(\Leftarrow) \Rightarrow i(a,b) \le i(a, w_i(a,d)) = i(a, \sup\{x \mid i(a,x) \le d\}) = \sup\{i(a,x) \mid i(a,x) \le d\} \le d$

(3)

$\because i(a,x) \le w_i(b,d) \Leftrightarrow i(b, i(a,x)) \le d$

By (1)=>

Associativity communitation

$\Leftrightarrow i(i(a,b),x) \le d \Leftrightarrow x \le w_i(i(a,b)d)$

$w_i(i(a,b),d)$

By (A)

$\therefore w_i(a, w_i(b,d)) = \sup\{x \mid i(a,x) \le w_i(b,d)\} = \sup\{x \mid x \le w_i(i(a,b),d)\} = w_i(i(a,b),d)$

(7) Let $S = \sup_j a_j$ ---(B) $\Rightarrow a_j \le s, \forall j$

By(4)

$\Rightarrow w_i(s,b) \le w_i(a_j,b), \forall j$

$\therefore w_i(s,b) \le \inf_j w_i(a_j,b)$ ---- (C)

$\because \inf_j w_i(a_j,b) \le w_i(a_{j_0},b), \forall j_0 \in J$

By(1)

$\Rightarrow i(a_{j_0}, \inf_j w_i(a_j,b)) \le b, \forall j_0$

$\therefore i(s, \inf_j w_i(a_j,b)) = \sup_{j_0} i(a_{j_0}, \inf_j w_i(a_j,b)) \le b$

By(1)

$\Rightarrow w_i(s,b) \ge \inf_j w_i(a_j,b)$ --- (D)

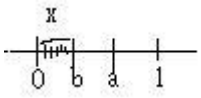
By(B)(C)(D)

$\Rightarrow w_i(\sup_j a_j, b) = w_i(s,b) = \inf_j w_i(a_j,b)$

(2)Show $w_i(w_i(a,b),b) \geq a$ (Theorem 3.3 (2))

proof : $\because w_i(a,b) = Sup\{x \in [0,1] \mid i(a,x) \leq b\}$ and by Theorem 3.10

$i_{\min}(a,b) \leq i(a,b) \leq \min(a,b)$



i, If a>b

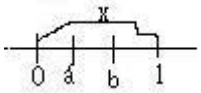
$w_i(a,b) = Sup\{x \in [0,1] \mid i(a,x) \leq b\} \leq Sup\{x \in [0,1] \mid \min(a,x) \leq b\} = b$

$\therefore \begin{aligned} & i(w_i(a,b),a) \\ & \leq i(b,a) \\ & \leq i(b,1) \\ & = b \end{aligned}$

By Axiom i2 $w_i(a,b) \leq b$

By Axiom i2 $(a \leq 1)$

$\Rightarrow i(w_i(a,b),a) \leq b$



i, If $a \leq b$

$w_i(a,b) = Sup\{x \in [0,1] \mid i(a,x) \leq b\} \leq Sup\{x \in [0,1] \mid \min(a,x) \leq b\} = 1$

$\therefore i(w_i(a,b),a) \leq i(w_i(a,b),b)$

$\leq i(1,a)$

By Axiom i2 $w_i(a,b) \leq 1$

$\leq i(b,1)$

By Axiom i2

$= b$

$\Rightarrow i(w_i(a,b),a) \leq b$

By Theorem 3.3 property 1(i.e., $i(a,b) \leq d$ iff $w_i(a,d) \geq b$)

$i(w_i(a,b),a) \leq b \Rightarrow w_i(w_i(a,b),b) \geq a$

(4) prove Theorem 3.3 (4) : $a \le b$ =>i, $W_i(a,d) \ge W_i(b,d)$

ii, $W_i(d,a) \le W_i(d,b)$

proof :i, $W_i(a,d) \ge W_i(b,d)$

$W_i(a,d) = \sup\{x \mid i(a,x) \le d\}$ ---- (A)

$W_i(b,d) = \sup\{x \mid i(b,x) \le d\}$ ---- (B)

a, if $d \le a \le b$ => (A)=d , (B)=d ,

∴ (A)=(B) ----- (1)

b, if $a \le d \le b$ => (A)=1 , (B)=d

∴ (A) ≥ (B) ----- (2)

c, if $a \le b \le d$ => (A)=1 , (B)=1

∴ (A)= (B) ----- (3)

(1),(2),(3) => (A) ≥ (B)

i.e., $W_i(a,d) \ge W_i(b,d)$

ii, see i

5. show $i(W_i(a,d), W_i(b,d)) \le W_i(a,d)$

Proof :∵ if $a > b \Rightarrow W_i(a,b) = b$

if $a \le b \Rightarrow W_i(a,b) = 1$

A, if $a \le b \Rightarrow i(W_i(a,b), W_i(b,d)) = i(b, W_i(b,d))$

$\Rightarrow i(b, W_i(b,d)) = i(b,d) \le \min(b,d) = d$

$W_i(a,d) = d$

$\Rightarrow i(b, W_i(b,d)) = i(b,1) = b$

$W_i(a,d) = 1$

$$=> i(b, W_i(b,d)) = i(b,1) = b$$

$$W_i(a,d) = 1$$

B, if $a \le b => i(W_i(a,b), W_i(b,d)) = i(1, W_i(b,d)) = W_i(b,d)$

$$=> W_i(b,d) = d$$

$$W_i(a,d) = d$$

$$W_i(b,d) = d$$

$$W_i(a,d) = 1$$

$$W_i(b,d) = 1$$

$$W_i(a,d) = 1$$

10. show $i(a, W_i(a,b)) \le b$

Proof :∵ $a > b => W_i(a,b) = b$

$$a \le b => W_i(a,b) = 1$$

A, if $a > b$

$$=> i(a, W_i(a,b)) = i(a,b) \le \min(a,b) = b$$

B, if $a \le b$

$$=> i(a, W_i(a,b)) = i(a,1) = a \le b$$

- $\inf - W_i$ composition

$$(P \overset{W_i}{\circ} Q)(x,z) = \overset{\inf}{y \in Y} W_i(P(x,y), Q(y,z))$$

- Theorem 3.4 :

$$(1)(P \overset{i}{\circ} Q \subseteq R) <=> (Q \subseteq P^{-1} \overset{W_i}{\circ} R) <=> (P \subseteq (Q \overset{W_i}{\circ} R^{-1})^{-1})$$

$$(2)(P \overset{W_i}{\circ} (Q \overset{W_i}{\circ} S) = (P \overset{i}{\circ} Q) \overset{W_i}{\circ} S$$

- Theorem 3.5 :

$$(\cup_j P_j) \overset{W_i}{\circ} Q = \cap_j (P_j \overset{W_i}{\circ} Q)$$

$$(\cap_j P_j) \overset{W_i}{\circ} Q) \supseteq \cup_j (P_j \overset{W_i}{\circ} Q)$$

$$P \overset{W_i}{\circ} (\cap_j Q_j) = \cap_j (P_j \overset{W_i}{\circ} Q_j)$$

$$P \overset{W_i}{\circ} (\cup_j Q_j) \supseteq \cup_j (P_j \overset{W_i}{\circ} Q_j)$$

- Theorem 3.6 : if $Q_1 \subseteq Q_2$ => $P \overset{W_i}{\circ} Q_1 \subseteq P \overset{W_i}{\circ} Q_2$

$Q_1 \overset{W_i}{\circ} R \supseteq Q_2 \overset{W_i}{\circ} R$

Proof : $Q_1 \subseteq Q_2$ => $Q_1 \cap Q_2 = Q_1$, $Q_1 \cup Q_2 = Q_2$

$$\because (P \overset{W_i}{\circ} Q_1) \cap (P \overset{W_i}{\circ} Q_2) = P \overset{W_i}{\circ} (Q_1 \cap Q_2) = P \overset{W_i}{\circ} Q_1$$

$$=> P \overset{W_i}{\circ} Q_1 \subseteq P \overset{W_i}{\circ} Q_2$$

$$\because (Q_1 \overset{W_i}{\circ} R) \cap (Q_2 \overset{W_i}{\circ} R) = (Q_1 \cup Q_2) \overset{W_i}{\circ} R = Q_2 \overset{W_i}{\circ} R$$

$$=> Q_1 \overset{W_i}{\circ} R \supseteq Q_2 \overset{W_i}{\circ} R$$

- Theorem 3.7 :

  1. $P^{-1} \overset{i}{\circ} (P \overset{W_i}{\circ} Q) \subseteq Q$

  2. $R \subseteq P \overset{W_i}{\circ} (P^{-1} \overset{i}{\circ} R)$

  3. $P \subseteq (P \overset{W_i}{\circ} Q) \overset{W_i}{\circ} Q^{-1}$

  4. $R \subseteq (R \overset{W_i}{\circ} Q^{-1}) \overset{W_i}{\circ} Q$

  Proof :

(1) $\because P \overset{W_i}{\circ} Q \subseteq (P^{-1})^{-1} \overset{W_i}{\circ} Q$ ---- (A)

$\because (5.26) <==========> (5.25)$

i.e., $(\vec{Q} \subseteq \vec{P}^{-1} \overset{W_i}{\circ} \vec{R}) <=======> (\vec{P} \overset{i}{\circ} \vec{Q} \subseteq \vec{R})$

let $P \overset{W_i}{\circ} Q = \vec{Q}$, $P^{-1} = \vec{P}$, $Q = \vec{R}$

$\therefore (A) <==> P^{-1} (P \overset{i}{\circ} Q) \subseteq Q$

(2) $\because P^{-1} \overset{i}{\circ} R \subseteq P^{-1} \overset{i}{\circ} R$ ---- (B)

Let $P^{-1} = \vec{P}$, $Q = \vec{R}$, $P^{-1} \overset{i}{\circ} R = \vec{R}$

$\therefore (B) <==> R \subseteq P \overset{W_i}{\circ} (P^{-1} \overset{i}{\circ} R)$

(3) by (3.33) , $[P^{-1} \overset{i}{\circ} (P \overset{W_i}{\circ} Q)]^{-1} \subseteq Q^{-1}$

$=> (P \overset{W_i}{\circ} Q)^{-1} \overset{i}{\circ} P \subseteq Q^{-1}$ ---- (C)

Let $P \overset{W_i}{\circ} Q = \vec{P}, P = \vec{Q}, Q^{-1} = \vec{R}$

$\therefore (C) <==> P \subseteq (P \overset{W_i}{\circ} Q)^{-1} \overset{W_i}{\circ} Q^{-1}$

(4) follows (3)

Chapter 4

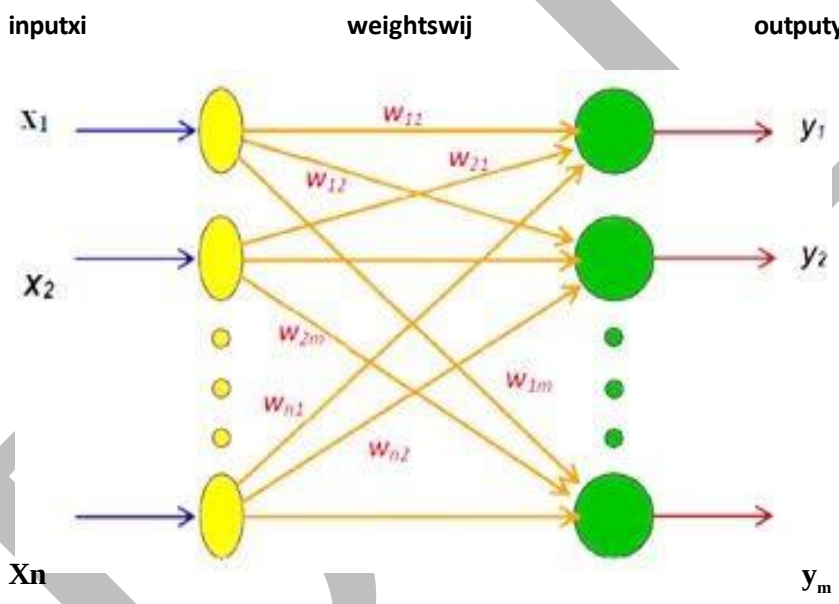
Single Layer Perception: Perceptron convergence theorem, Method of steepest descent-least mean square algorithms.

Single-Layer NN Systems

Here,asimplePerceptronModelandanADALINENetworkModelispresented.

Single layerPerceptron

Definition:Anarrangementofoneinputlayerofneuronsfeedforwardtooneoutputlayerofneurons isknownasSingleLayerPerceptron.



Single layer Perceptron Fig.

Simple PerceptronModel

1 if net j≥0

0 if net j<0

where netj= $\sum_{i=1}^{n}$ **xiwij**

- LearningAlgorithm:TrainingPerceptron

  ThetrainingofPerceptronisasupervisedlearningalgorithmwhereweightsareadjustedtominimize errorwhenevertheoutputdoesnotmatchthedesiredoutput.

  - Iftheoutputiscorrectthennoadjustmentofweightsisdone.

  i.e. $W_{ij}^{K+1} = W_{ij}^{K}$

- If the output is **1** but should have been 0 then the weights are decreased on the active input link

  i.e. $W_{ij}^{K+1} = W_{ij}^{K} - .x_i$

- Iftheoutputis0butshouldhavebeen1thentheweightsareincreasedontheactiveinputlink

i.e. $W_{ij}^{K+1} = W_{ij}^{K} + \propto . xi$

Where

W_{ij}^{K+1} isthenewadjustedweight, W_{ij}^{K} is the oldweight

xi istheinputand$\propto$isthelearningrateparameter.

$\propto$ smallleadstoslowand$\propto$largeleadstofastlearning.

- PerceptronandLinearlySeparableTask

  Perceptroncannothandletaskswhicharenotseparable.

  - Definition:Setsofpointsin2-Dspacearelinearlyseparableifthesetscanbeseparatedbyastraight line.
  - Generalizing,asetofpointsinn-dimensionalspacearelinearlyseparableifthereisahyperplaneof (n-1)dimensionsseparatesthesets.

Example

(a) Linearlyseparablepatterns (b) Not Linearly separablepatterns

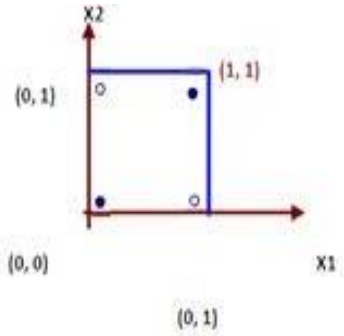
Note:Perceptroncannotfindweightsforclassificationproblemsthatarenotlinearlyseparable.

- XOR Problem

  :Exclusive

Input x1	Input x2	Output	
0	0	0	
1	1	0	Even parity •
0	1	1	
1	0	1	Odd parity °

XOR truth table



  ORoperation

Fig. Output of XOR in x1 , x2 plane

  Evenparityis,evennumberof1bitsintheinput

  Oddparityis,oddnumberof1bitsintheinput

  - Thereisnowaytodrawasinglestraightlinesothatthecirclesareononesideofthelineandthedots ontheotherside.
  - Perceptronisunabletofindalineseparatingevenparityinputpatternsfromoddparityinputpatterns.

- PerceptronLearningAlgorithm

  The algorithm is illustrated step-by-step.

* **Step 1:**

  Createapeceptronwith**(n+1)**inputneurons**x0,x1,……, . xn,**

  where **xo = 1** is the bias input. Let **O** be the output neuron.

* **Step 2:**

  Initialize weight **W=(w_0,w_1,…… ,.wn)** torandomweights.

* **Step 3:**

  Iteratethroughtheinputpatterns**Xj**ofthetrainingsetusingtheweightset;iecomputetheweightedsum of inputs **net j=** $\sum_{i=1}$ xiwi **foreachinputpatternj.**

* **Step 4:**

  Compute the output **y j** using the step function

$$y_j = f(net_j) = \begin{cases} 1 \text{ if } net_j \geq 0 \\ wij0 \text{ if } net_j < 0 \end{cases} \quad \text{where } net_j = \sum_{i=1}^{n} x_i$$

* **Step 5:**

  Compare thecomputedoutput **yj** with thetargetoutput **yj** foreachinputpattern**j**.Ifalltheinputpatternshavebeenclassifiedcorrectly,thenoutput(read)theweightsandexit.

* **Step 6:**

  Otherwise, update the weights as given below :

  Ifthecomputedoutputs**yj**is**1**butshouldhavebeen0,

  Then **wi = wi - ∝xi, i= 0,1,2,.....,n**

  Ifthecomputedoutputs**yj**is**0**butshouldhavebeen**1**,

  Then**wi=wi+∝xi, i= 0,1,2,.....,n**

  where ∝is the learning parameter and is constant.

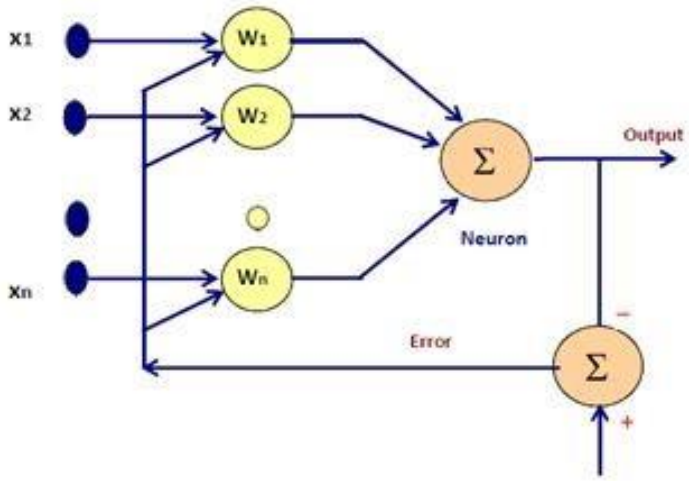
* **Step 7:**

  goto step 3

* **END**

AnADALINEconsistsofasingleneuronoftheMcCulloch-Pittstype,whereitsweightsaredetermined bythenormalizedleastmeansquare(LMS)traininglaw.TheLMSlearningruleisalsoreferredtoas deltarule.Itisawell-establishedsupervisedtrainingmethodthathasbeenusedoverawiderangeof diverseapplications.

- **ArchitectureofasimpleADALINE**



ThebasicstructureofanADALINEissimilartoaneuronwithalinearactivationfunctionandafeedback loop.Duringthetrainingphaseof ADALINE,theinputvectoraswellasthedesiredoutputarepresented to thenetwork.

[The complete training mechanism has been explained in the next slide.]

- **ADALINETraining Mechanism**

  (Ref. Fig. in the previous slide - Architecture of a simple ADALINE)

* ThebasicstructureofanADALINEissimilartoalinearneuronwithanextrafeedbackloop.
* DuringthetrainingphaseofADALINE,theinputvector$X=[x_1,x_2,...,x_n]^T$aswellasdesired outputarepresentedtothenetwork.
* Theweightsareadaptivelyadjustedbasedondeltarule.
* AftertheADALINEistrained,aninputvectorpresentedtothenetworkwithfixedweightswillresult inascalaroutput.
* Thus,thenetworkperformsanndimensionalmappingtoascalarvalue.
* The activation function is not used during the training phase. Once the weights are properly adjusted, he response of the trained unit can be tested by applying various inputs, which are not in the training set.Ifthenetworkproducesconsistentresponsestoahighdegreewiththetestinputs,itissaidthatthe etworkcouldgeneralize.Theprocessoftrainingandgeneralizationaretwoimportantattributesofthis network.

Usage of ADLINE :

In practice, an ADALINE is used to

- Makebinarydecisions;theoutputissentthroughabinarythreshold.
- RealizationsoflogicgatessuchasAND,NOTandOR.
- Realizeonlythoselogicfunctionsthatarelinearlyseparable.

Applications of Neural Network

Neural Network Applications can be grouped in following categories:

* **Clustering:**

  Aclusteringalgorithmexploresthesimilaritybetweenpatternsandplacessimilarpatternsinacluster. Bestknownapplicationsincludedatacompressionanddatamining.

* **Classification/Pattern recognition:**

  Thetaskofpatternrecognitionistoassignaninputpattern(likehandwrittensymbol)tooneofmany classes.Thiscategoryincludesalgorithmicimplementationssuchasassociativememory.

* **Function approximation:**

  Thetasksoffunctionapproximationistofindanestimateoftheunknownfunctionsubjecttonoise. Variousengineeringandscientificdisciplinesrequirefunctionapproximation.

* **Prediction Systems:**

  Thetaskistoforecastsomefuturevaluesofatime-sequenceddata.Predictionhasasignificantimpact ondecisionsupportsystems.Predictiondiffersfromfunctionapproximationbyconsideringtimefactor. System may be dynamic and may produce different results for the same input data based on system state(time).

Multilayer Perceptron: Derivation of the back-propagation, Algorithm , Learning Factors

MULTI-LAYER PERCEPTRONS

Inthepreviussectionweshowedthatbyaddinganextrahiddenunit,theXORproblemcanbesolved.For binaryunits,onecanprovethatthisarchitectureisabletoperformanytransformationgiventhecorrectconnections andweights.Themostprimitiveisthenextone.Foragiventransformationy=d(x),wecandividethesetofall possibleinputvectorsintotwoclasses:X^+={x|d(x)=1}andX^-={x|d(x).1}

Since there are Ninput units, the total number of possible input vectors x is 2N. For every x^p c X+ ahiddenunithcanbereservedofwhichtheactivationyhis1ifandonlyifthespecificpatternpispresentattheinput: wecanchooseitsweightsw_{ih}equaltothespecificpatternxandthebias$U_h$equalto1-Nsuchthat

$$y_h^p = \mathrm{sgn}\left[\sum_i w_{ih} x_i^p - N + \frac{1}{2}\right]$$

isequalto1forx^p=wonly.Similarly,theweightstotheoutputneuroncanbechosensuchthattheoutputisoneas soonasoneoftheMpredicateneuronsisone:

$$y_o^p = \mathrm{sgn}\left[\sum_{h=1}^{M} w_h + M - \frac{1}{2}\right]$$

Thisperceptronwillgiveo=1onlyifx$\in X^+$:itperformsthedesiredmapping.Theproblemisthelargenumberof predicateunits,whichisequaltothenumberofpatternsinX^+,whichismaximally2^N.ofcoursewecandothesame trickforX^-,andwewillalwaystaketheminimalnumberofmaskunits,whichismaximally2^{N-1}.Amoreelegantproof isgivenbyMinskyandPapert,butthepointisthatforcomplextransformationsthenumberofrequiredunitsinthe hiddenlayerisexponentialinN.

Back-Propagation

Aswehaveseeninthepreviouschapter,asingle-layernetworkhassevererestrictions:theclassoftasksthat canbeaccomplishedisverylimited.Inthischapterwewillfocusonfeedforwardnetworkswithlayersofprocessing units.

MinskyandPapertshowedin1969thatatwolayerfeed-forwardnetworkcanovercomemanyrestrictions, butdidnotpresentasolutiontotheproblemofhowtoadjusttheweightsfrominputtohiddenunits.Ananswertothis questionwaspresentedbyRumelhart,HintonandWilliamsin1986,andsimilarsolutionsappearedtohavebeen publishedearlier(Parker,1985;Cun,1985).

Thecentralideabehindthissolutionisthattheerrorsfortheunitsofthehiddenlayeraredeterminedbyback-propagatingtheerrorsoftheunitsoftheoutputlayer.Forthisreasonthemethodisoftencalledtheback-propagation learningrule.Back-propagationcanalsobeconsideredasageneralizationofthedeltarulefornon-linearactivation functionsandmultilayernetworks.

MULTI - LAYER FEED - FORWARD NETWORKS

A feed-forward network has a layered structure. Each layer consists of units, which receive their input from units from a layer directly below and send their output to units in a layer directly above the unit. There are no connections within a layer.

The N_i inputs are fed into the first layer of N_h,1 hidden units. The input units are merely 'fan-out' units; no processing takes place in these units. The activation of a hidden unit is a function F_i of the weighted inputs plus a bias, as given in eq.(10.4). The output of the hidden units is distributed over the next layer of N_h,2 hidden units, until the last layer of hidden units, of which the outputs are fed into a layer of N_o output units (see Fig.12.1).

Although back-propagation can be applied to networks with any number of layers, just as for networks with binary units (section 11.7) it has been shown (Cybenko,1989; Funahashi,1989; Hornik, Stinchcombe, & White, 1989; Hartman, Keeler, & Kowalski,1990) that only one layer of hidden units suffices to approximate any function with finitely many discontinuities to arbitrary precision, provided the activation functions of the hidden units are non-linear (the universal approximation theorem).

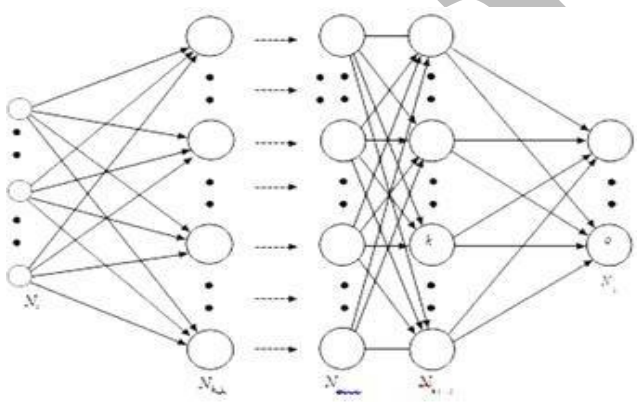


Fig. A multi-layer network with layers of units.

In most applications a feed-forward network with a single layer of hidden units is used with a sigmoid activation function for the units.

THE GENERALISED DELTA RULE

Since we are now using units with nonlinear activation functions, we have to generalise the delta rule, which was presented in chapter 11 for linear functions to the set of non-linear activation functions. The activation is a differentiable function of the total input, given by

$$y_k^p = F(S_k^p) \quad \text{..... (12.1)}$$

in which

$$S_k^p = \sum_j w_{jk} y_k^p + \theta_k \quad \text{....(12.2)}$$

To get the correct generalization of the delta rule as presented in the previous chapter, we must set

$$\Delta_p w_{jk} = -\gamma \frac{\partial E^p}{\partial W_{jk}} \quad \text{....(12.3)}$$

The error E^p is defined as the total quadratic error for pattern p at the output units :

$$E^p = \frac{1}{2} \sum_{o=1}^{No} (d_o^p - y_o^p)^2 \quad \text{.....(12.4)}$$

where d_o^p is the desired output for unit **0** when pattem**p** is clamped.We further set $E = \sum_{p} E^p$ as the summed squared error. We can write

$$\frac{\partial E^p}{\partial w_{jk}} \quad \frac{\partial E^p}{\partial S_k^p} \quad \frac{\partial S_k^p}{\partial w_{jk}} \qquad ...(12.5)$$

By equation (12.2) we sec thal the scoond factor is

$$\frac{\partial S_k^p}{\partial w} = y_j^{pjk} \qquad ...(12.6)$$

When we define

$$\delta^p = \frac{\partial E^p}{\partial S_k^p} \qquad ...(12.7)$$

wewillgetanupdaterulewhichisequivalenttothedeltaruleasdescribedintheprevious chapter,resultingina gradientdescentontheerrorsurfaceifwemaketheweightchangesaccordingto:

$$\Delta_p W_{jk} = \gamma \delta^p \gamma^p{}_j \qquad ...(12.8)$$

The trick is to figure out what δ_k^p should be for each unit k in the network. The interesting result, whichwe nowderive,isthatthereisasimplerecursive computationoftheseδ'swhichcanbeimplemented bypropagatingerrorsignalsbackwandthroughthenetwork.

To computeδ^pweapplythechainruletowritethispartialderivativeastheproductoftwofactors,onefactor reflectingthechangeinerrorasafunctionoftheoutputoftheunitandonereflectingthechangeinthecutputasa functionofchangesintheinput.Thus,wehave

$$\delta^p = \frac{\partial E^p}{\partial w^p_k} \quad \frac{\partial E^p}{\partial S^p_k} \quad \frac{\partial y^p_k}{\partial S^p_k} \qquad ...(12.9)$$

Let us compute the scoond factor. By equation (12.1) we see that

$$\frac{\partial y^p}{\partial S^p_k} = F(S^p)_k \qquad ...(12.10)$$

whichissimplythederivativeofthesquashingfunctionFforthekthunit,evaluatedattheactinputSftothatunit.To computethefirstfactorofequation(12.9),weconsidertwocases.First,assumethatunitkisanoutputunttk=oof thenetwork.Inthiscase,itfollowsfromthedefinitionofE^{P}that

$$\frac{\partial E^P}{\partial y_o^P} = -(d^P - y^P) \qquad \text{...(12.11)}$$

which is the same result as we obtained with the standard delta rule. Substituting this and equation (12.10)inequation(12.9),weget

$$\delta_o^P = (d^p - y^p) F^t (S_o^p) \qquad \text{...(12.12)}$$

foranyoutputunito,Secondly,ifkisnotanoutputunitk-h,wedonotreadilyknowthecontributionoftheunitto theoutputerrorofthenetwork.However,theerrormeasurecanbewrittenasafunctionofthenetinputsfromhidden tooutputlayerEo-Ep(xp,xp,.....sp,)andweusethechainruletowrite.

$$\frac{\partial E^P}{\partial y_k^P} = \sum_{0=1}^{N} \frac{\partial E^P}{\partial S^P_k} \quad \frac{\partial S^P_o}{\partial S^P_k} = \sum_{0=1}^{N} \frac{\partial E^P \partial}{\partial S^P_n} \quad \partial y_n^P \sum_{j=1}^{N} W_{in} \quad y_j^P = \sum_{j=1}^{N} \frac{\partial E^P}{\partial S^P_n} w_{ij} = \sum_{j=1}^{N} \delta_o^P w_{ij} \qquad \text{...(12.13)}$$

Substituting this in equation (12.9) yields.

$$\delta_k^P = F(S^p)_k \sum_{j=1}^{N_o} \delta_o^P w_{ho}$$

Equations (12.12) and (12.14) give a recursive procedure for computing the 6's for all units in the network, whicharethenusedtocomputetheweightchangesaccordingtoequation(12.8).Thisprocedureconstitutesthegeneralized delta rule for a feed-forward network of non-linearunits.

12.3.1 UnderstandingBack-Propagation

The equations derived in the previous section may be mathematically correct, but what do they actually mean ?

Is there a way of understanding back-propagation other than reciting the necessary equations ?

The answer is, of course, yes. In fact, the whole back-propagation process is intuitively very clear. What happensintheaboveequationsisthefollowing.Whenalearningpatternisclamped,theactivationvaluesarepropagated totheoutputunits,andtheactualnetworkoutputiscomparedwiththedesiredoutputvalues,we usuallyendupwithan error in each of the output units. Let's call this error $e_o$for a particular output unit o. We have to bring $e_o$tozero.

The simplest method to do this is the greedy method: we strive to change the connections in the neural network insuchawaythat,nexttimearound,theerror$e_o$willbezeroforthisparticularpattern.Weknowfromthedeltarulethat, in order to reduce an error, we have to adapt its incoming weightsaccording to

$$Ow_{ho}=(d_o- y_o)\ y_h$$

Thatisstepone.Butitaloneisnotenough:whenweonlyapplythisrule,theweightsfrominputtohiddenunits are never changed, and we do not have the full representational power of the feed-forward network as promised by the universal approximationtheorem.

In order to adapt the weights from input to hidden units, we again want to apply the delta rule. In this case, however, we do not have a value for 6 for the hidden units. This is solved by the chain rule which does the following: distribute the error of an output unit o to all the hidden units that is it connected to, weighted by this connection. Differentlyput,ahiddenunithreceivesadeltafromeachoutputunitoequaltothedeltaofthatoutputunitweightedwith (= multiplied by) the weight of the connection between thoseunits.

In symbols: $6_h = \backslash\ 6_O w_{ho}$Well, not exactly: we forgot the activation

0

function of the hidden unit; F' has to be applied to the delta, before the backpropagation process can continue.

WORKINGWITH BACK-PROPAGATION

Theapplicationofthegeneraliseddeltarulethusinvolvestwophases:Duringthefirstphasetheinputxis presentedandpropagatedforwardthroughthenetworktocomputetheoutputvaluesypforeachoutputunit.This outputiscomparedwithitsdesiredvaluedo,resultinginanerrorsignal6pforeachoutputunit.

Thesecondphaseinvolvesabackwardpassthroughthenetworkduringwhichtheerrorsignalispassedto eachunitinthenetworkandappropriateweightchangesarecalculated.

12.4.1 WeightAdjustmentswithSigmoidActivationFunction

The results from the previous section can be summarised in three equations:

* Theweightofaconnectionisadjustedbyanamountproportionaltotheproductofanerrorsignal6, ontheunitkreceivingtheinputandtheoutputoftheunitjsendingthissignalalongtheconnection:

$$\Delta_p W_{kj} = \gamma \delta^P \gamma_k^P \quad _j \qquad \text{...(12.16)}$$

* Iftheunitisanoutputunit,theerrorsignalisgivenby

$$\delta_o^P \text{ -(d}^p \text{ -}_o \text{ y}^p\text{) F}_o^t \text{ (S}_o^p\text{)} \quad _o \qquad \text{...(12.17)}$$

Take as the activation function F the 'sigmoid' function as defined in chapter 2 :

$$y^P = F(S^P) = \frac{1}{1 + e^{-tP}} \qquad \text{...(12.18)}$$

Inthiscasethederivativeisequalto

$$F^{-1}(S^P) = \frac{\partial}{\partial S^P} \frac{1}{1+e^{-S^P}} = \frac{1}{(1+e^{-S^P})^2}(-e^{-t})^p = -\frac{1}{(1+e^{-S^P})^2}(-e^{-t^P}) = y^p(1-y^p) \qquad \text{...(12.19)}$$

such that the error signal for an output unit can be written as :

$$\delta_o^P \text{ -(d}^p \text{ -}_o \text{y}^p\text{) y}^p_o \quad _{oo} \text{ (1-y}^p\text{)}_o \qquad \text{...(12.20)}$$

* Theerrorsignalforahiddenunitisdeterminedrecursivelyintermsoferrorsignalsoftheunitstowhichit directlyconnectsandtheweightsofthoseconnections.Forthesigmoidactivationfunction:

$$\delta_k^P = F^p(S^p)_k \sum_{j=1}^{N_o} d_o^P w_{ho} - y^P{}_k(1 - y^p)_k \sum_{j=1}^{N_o} d_o^P w_{ho} \qquad \text{...(12.21)}$$

Learning Rate And Momentum

The learning procedure requires that the change in weight is 6E . True gradient descent proportional to

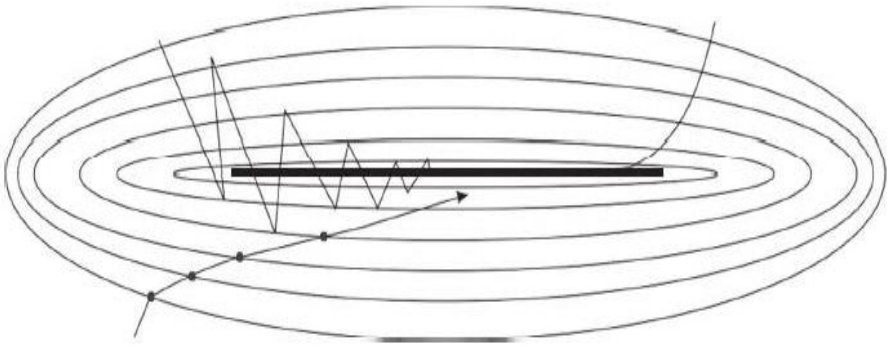
$$\frac{\partial E^p}{\partial w}$$

. requiresthatinfinitesimalstepsaretaken.Theconstantofproportionalityisthelearningratey.Forpracticalpurposes wechoosealearningratethatisaslargeaspossiblewithoutleadingtooscillation.Onewaytoavoidoscillationat large,istomakethechangeinweightdependentofthepastweightchangebyaddingamomentumterm:

$$\Delta w_{jk}(t+1) = \eta \delta_k^p o_j^p + \alpha \Delta w_{jk}(t) \quad (12.22)$$

where t indexes the presentation number and a is a constant which determines the effect of the previous weight change.

Theroleofthemomentumtermisshown inFig.12.2.Whennomomentumtermisused,ittakesalongtime beforetheminimumhasbeenreachedwithalowlearningrate,whereasforhighlearningratestheminimumisnever reachedbecauseoftheoscillations.Whenaddingthemomentumterm,theminimumwillbereachedfaster.



The descent in weight space. (a) for small learning rate; (b) for large learning rate: note the oscillations, and (c) with large learning rate and momentum term added.

Learning Per Pattern

Although,theoretically,theback-propagationalgorithmperformsgradientdescentonthetotalerroronlyiftheweights areadjustedafterthefullsetoflearningpatternshasbeenpresented,moreoftenthannotthelearningruleisapplied toeachpatternseparately,i.e.,apatternpisapplied,E^piscalculated,andtheweightsareadapted(p=1,2,…,P). Thereexistsempiricalindicationthatthisresultsinfasterconvergence.Carehastobetaken,however,withtheorder inwhichthepatternsaretaught.Forexample,whenusingthesamesequenceoverandoveragainthenetworkmay becomefocusedonthefirstfewpatterns.Thisproblemcanbeovercomebyusingapermutedtrainingmethod.

Example12.1:Afeed-forwardnetworkcanbeusedtoapproximateafunctionfromexamples.Supposewehavea system(forexampleachemicalprocessorafinancialmarket)ofwhichwewanttoknowthecharacteristics.The inputofthesystemisgivenbythetwo-dimensionalvectorxandtheoutputisgivenbytheone-dimensionalvectord. Wewanttoestimatetherelationshipd=f(x)from80examples{x^p,d^p}asdepictedinFig.12.3(topleft).Afeed-forwardnetworkwasprogrammedwithtwoinputs,10hiddenunitswithsigmoidactivationfunctionandanoutput unitwithalinearactivationfunction.Checkforyourselfhowequation(4.20)shouldbeadaptedforthelinearinstead ofsigmoidactivationfunction.Thenetworkweightsareinitializedtosmallvaluesandthenetworkistrainedfor5,000 learningiterationswiththeback-propagationtrainingrule,describedinthepreviousection.Therelationshipbetween xanddasrepresentedbythenetworkisshowninFig.12.3(topright),whilethefunctionwhichgeneratedthe learningsamplesisgiveninFig.12.3(bottomleft).Theapproximationerrorisdepictedin Fig.12.3(bottomright). Weseethattheerrorishigherattheedgesoftheregionwithinwhichthelearningsampleswere generated.The networkisconsiderablybetteratinterpolationthanextrapolation.

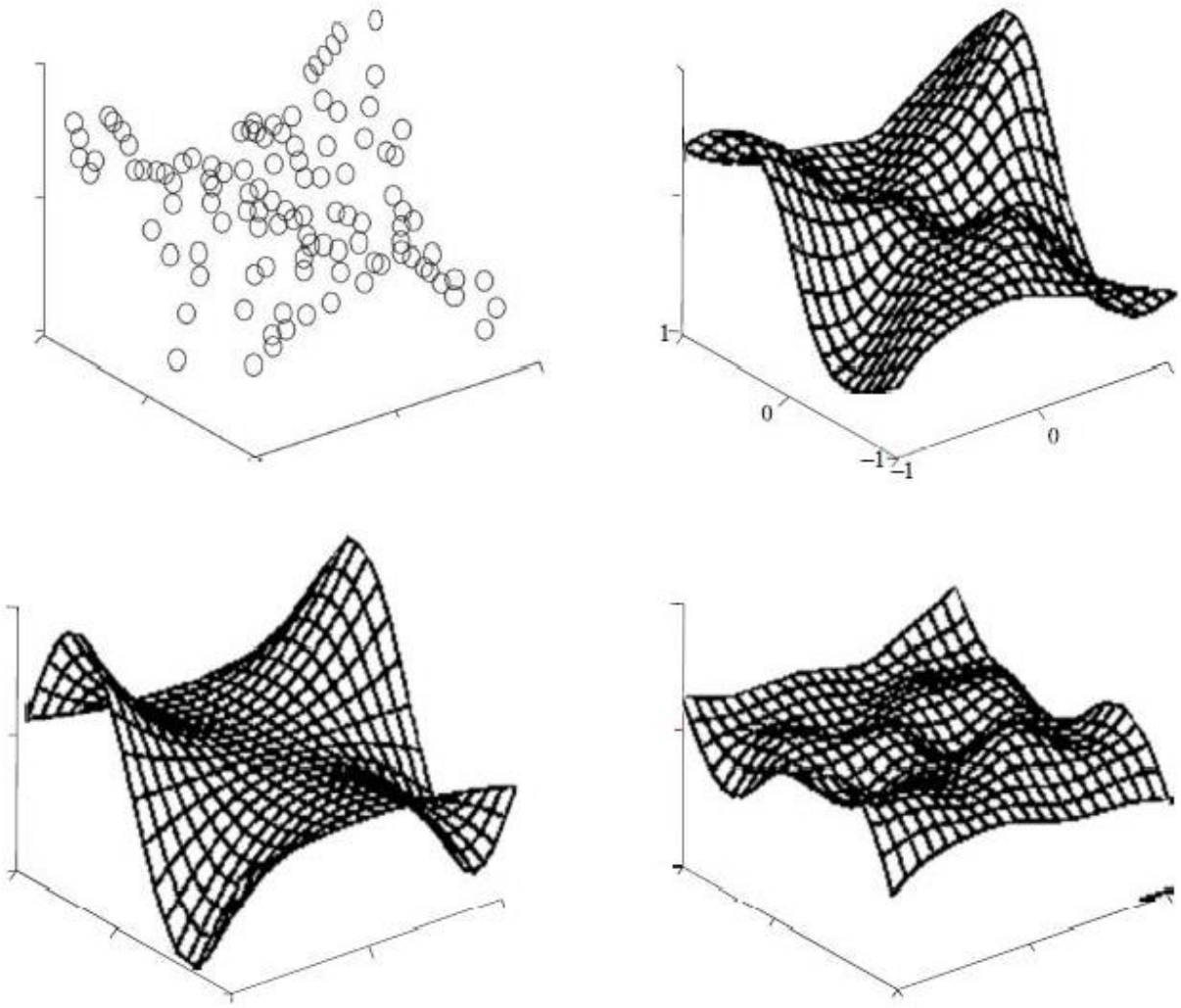


Fig. Example of function approximation with a feed forward network. Top left: The original learning samples; Top right: The approximation with the network; Bottom left: The function which generated the learning samples; Bottom right: The error in the approximation.

Exercise:

Q1. What is mean by Single-Layer NN Systems.
Q2. Explain Architecture of a simple ADALINE.
Q3. What are the use of ADLINE.
Q4. What are the Applications of Neural Network.
Q5. What is mean by Learning Algorithm.
Q6. Explain MULTI-LAYER PERCEPTRONS
Q7. What is the process of Back Propagation.
Q8. Write a short note on MULTI - LAYER FEED - FORWARD NETWORKS.
Q 9. List out the GENERALISED DELTA RULE.
Q10. How we can understand the Back Propagation.

Q11. Explain WORKINGWITH BACK-PROPAGATION.
Q12. What is mean by Learning Rate And Momentum.

Chapter 5

Radial Basis and Recurrent Neural Networks: RBF network structure , theorem and the reparability of patterns, RBF learning strategies

RadialBasisandRecurrentNeuralNetworks:RBFnetworkstructure,theoremandthereparabilityofpatternsRBF learningstrategies,K-meansandLMSalgorithms,comparisonofRBFandMLPnetworks:energyfunction,spurious states, errorperformance.

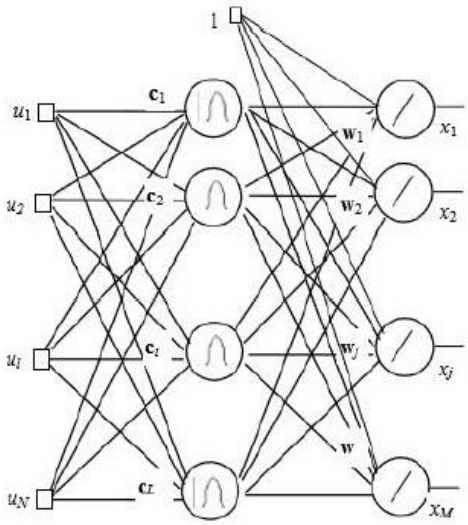
Radialbasisfunction(RBF)networksarefeed-forwardnetworkstrainedusingasupervisedtrainingalgorithm.They aretypicallyconfiguredwithasinglehiddenlayerofunitswhoseactivationfunctionisselectedfromaclassof functionscalledbasisfunctions.Whilesimilartobackpropagationinmanyrespects,radialbasisfunctionnetworks haveseveraladvantages.Theyusuallytrainmuchfasterthanbackpropagationnetworks.Theyarelesssusceptibleto problemswithnon-stationaryinputsbecauseofthebehaviouroftheradialbasisfunctionhiddenunits.

PopularizedbyMoodyandDarken(1989),RBFnetworkshaveproventobeausefulneuralnetworkarchitecture. ThemajordifferencebetweenRBFnetworksandbackpropagationnetworks(thatis,multilayerperceptrontrained byBackPropagationalgorithm)isthebehaviourofthesinglehiddenlayer.Ratherthanusingthesigmoidalor S-shapedactivationfunctionasinbackpropagation,thehiddenunitsinRBFnetworksuseaGaussianorsomeother basiskernelfunction.Eachhiddenunitactsasalocallytunedprocessorthatcomputesascoreforthematchbetween theinputvectoranditsconnectionweightsorcentres.Ineffect,thebasisunitsarehighlyspecializedpatterndetectors. Theweightsconnectingthebasisunitstotheoutputsareusedtotakelinearcombinationsofthehiddenunitsto producttthefinalclassificationoroutput.Inthischapterfirstthestructureofthenetworkwillbeintroducedanditwill beexplainedhowitcanbeusedforfunctionapproximationanddatainterpolation.Thenitwillbeexplainedhowitcan betrained.

The Structure of the RBF Networks

RadialBasisFunctionsarefirstintroducedinthesolutionoftherealmultivariableinterpolationproblems.BroomheadandL owe(1988),andMoodyandDarken(1989)werethefirsttoexploittheuseofradialbasisfunctionsinthe designofneuralnetworks.

The structure of an RBF networks in its most basic form involves three entirely different layers



Structure of the Standart RBF network

Theinputlayerismadeupofsourcenodes(sensoryunits)whosenumberisequaltothedimensionpoftheinput vectoru.

Hidden layer

Thesecondlayeristhehiddenlayerwhichiscomposedofnonlinearunitsthatareconnecteddirectlytoallofthe nodesintheinputlayer.Itisofhighenoughdimension,whichservesadifferentpurposefromthatinamultilayer perceptron.

Eachhiddenunittakesitsinputfromallthenodesatthecomponentsattheinputlayer.Asmentionedabovethehidden unitscontainsabasisfunction,whichhastheparameterscenterandwidth.Thecenterofthebasisfunctionforanode iatthehiddenlayerisavectorciwhosesizeistheastheinputvectoruandthereisnormallyadifferentcenterforeach unitinthenetwork.

First,theradialdistancedi,betweentheinputvectoruandthecenterofthebasisfunctionciiscomputedforeachunit iinthehiddenlayeras

$$d_i = || u - c_i ||$$ (5.1.1)

using the Euclidean distance.

The output $h_i$of each hidden unit t is then computed by applying the basis function G to this distance.

$$h_i = G(d_i, \sigma_i)$$ (5.1.2)

AsitisshowninFigure5.2.thebasisfunctionisacurve(tipicallyaGaussianfunction,thewidthcorrespondingtothe variance.σ_i)whichhasapeakatzerodistanceanditdecreasesasthedistancefromthecenterincreases.

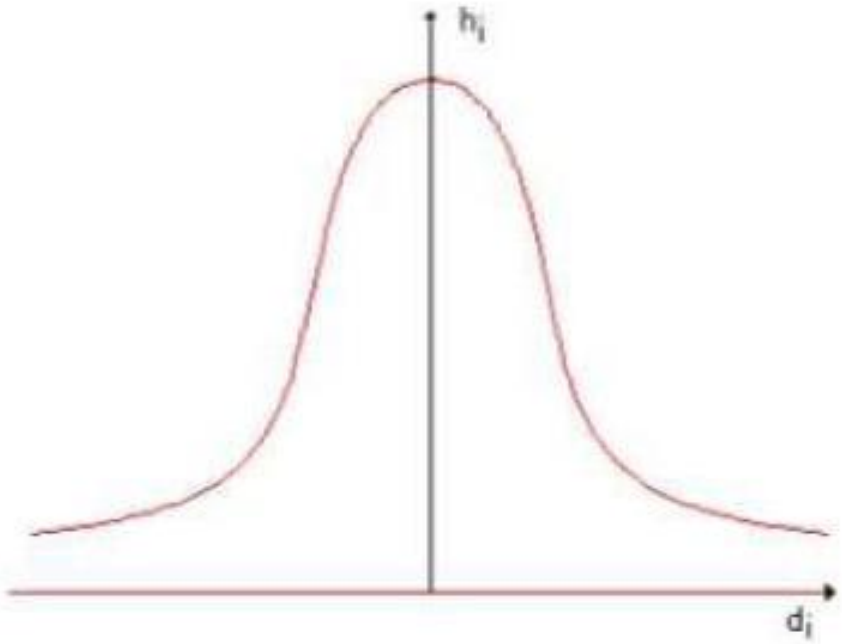


Figure5.2.TheresponseregionofanRBFhiddennodearoundits centerasafunctionofthedistancefromthiscenter.

Foraninputspaceu$\in R^2$,thatisM=2,thiscorrespondstothetwodimensionalGaussiancenteredatciontheinput space,wherealsoc$\in_j R^2$,asitisshowninFigure5.3

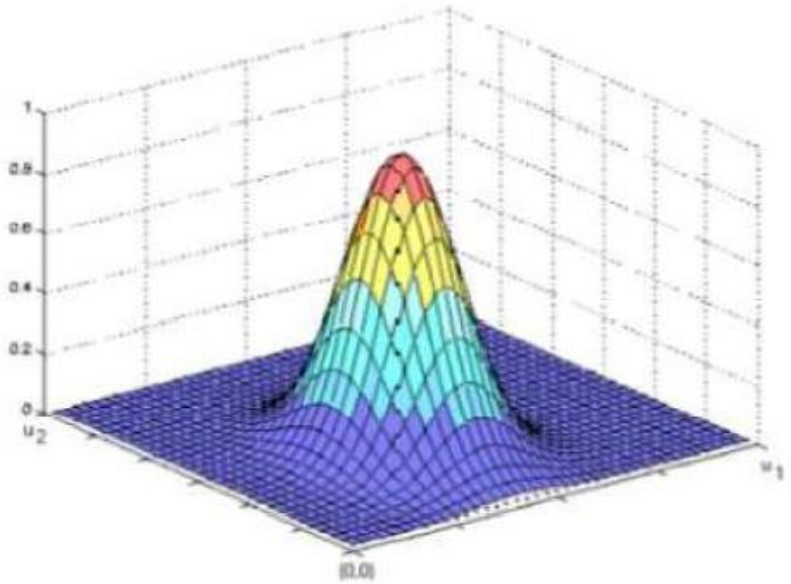


Figure5.3Responseofahiddenunitontheinputspaceforu$\in R^2$

5.1.2 Outputlayer

Thetransformationfromtheinputspacetothehiddenunitspaceisnonlinear,whereasthetransformationtothe hiddenunitspacetotheoutputspaceislinear.

The jth output is computed as

$$x_j = f_j(u) = w_{oj} + \sum_{(w)}^{t} w_y h_i \qquad j = 1,2,.....,M \tag{5.1.3}$$

Mathematicalmodel

In summary, the mathematical model of the RBF network can be expressed as;

$$x=f(u), f:R^N \rightarrow R^M \tag{5.1.4}$$

$$x_j = f_j(u) = w_{oj} + \sum_{|u|}^{t} w_g G(|u-ct|) \; j=1,2,.....,M \tag{5.1.5}$$

where is the Euclidean distance between u and c*i*

5.2 Functionapproximation

Let y=g(u) be a given function of u, y∈R, u∈R, g:R R, G i=1.. L, be a finite set of basis functions.

The function g can be written in terms of the given basis functions as

$$y = g(u) = \sum_{|u|}^{t} w_t G_t(u) + r(u) \quad (5.2.1)$$

where r(u) is the residual.

The function y can be approximated as

$$y = g(u) \cong \sum_{|u|}^{t} w_t G_t(u) \quad (5.2.2)$$

The aim is to minimize the error by setting the parameters of Giappropriately. A possible choice for the error definition is the L2 norm of the residual function r(u) which is defined as

$$||y(u)||_{1.2} = \sum r(u)^2 \quad (5.2.3)$$

5.2.1 Approximation byRBFNN

Now,considerthesingleinputsingleoutputRBFnetworkshowninFigure5.4.Thenxcanbewrittenas t

$$x = f(u) = \sum_{|u|} w_t G_t(||u - c_t||) \quad (5.2.4)$$

Bytheuseofsuchanetwork,ycanbewritten t

$$y = \sum_{|u|} w_t G(||u - c_t||) + r(u) = f(u) + r(u) \quad (5.2.5)$$

where f(u) is the output of the RBFNN given in Figure 5.4 and r(u) is the residual.

By setting the center c_i, the variance σ_i, and the weight $w_i$the error appropriately, the error can be

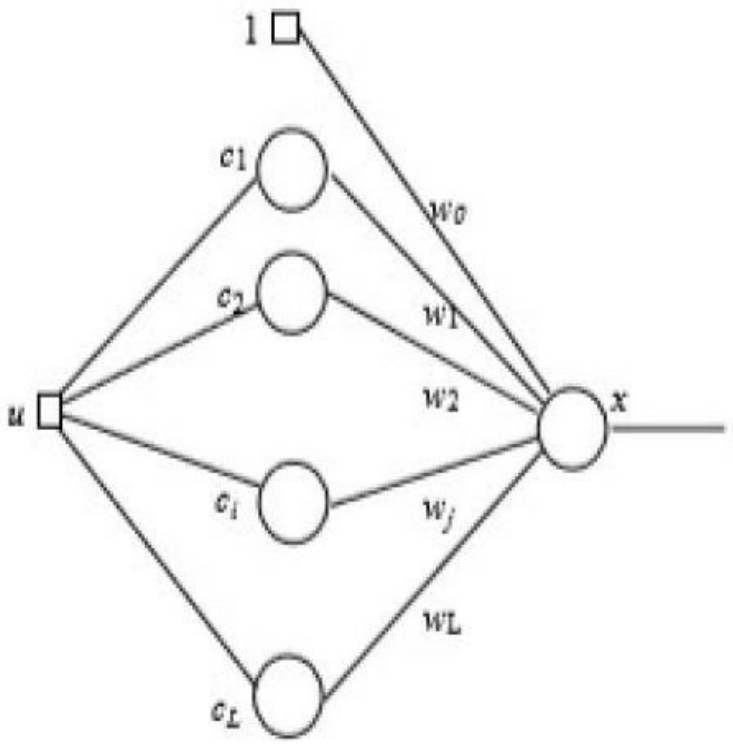


Figure 5.4 Single input, single output RBF network

Whatever we discussed here for g:R→R, can be generalized to g:R^N→R^M easily by using an **N** input, M output RBFNN given in figure 5.1 previously.

5.2.2 DataInterpolation

Giveninputoutputtrainingpatterns(u^k,y^k)$_t$k=1,2...K,theaimofdatainterpolationistoapprimatethefunctiony fromwhichthedataisgenerated.Sincethefunctionyisunknown,theproblemcanbestatedasaminimization problemwhichtakesonlythesamplepointsintoconsideration:

Choose **$w_{ij}$and c_i, i=1,2 ... L. j=1,2 ... M** so as to minimize

$$J(wtc)=\sum_{|u|}^{t} ||y^k-f(u^k)||^2 \qquad (5.2.6)$$

Asanexample,theoutputofanRBFnetworktrainedtofitthedatapointsgiveninTable5.1isgiveninFigure5.5.

TABLEI:13datapointsgeneratedbyusingsumofthreegaussisnswithc_1=0.2000 c_2 = 0.6000

c_3=0.9000 w_1= 0.2000 w = $_2$0.5000 w = 0.$_3$3000 σ =0.1000

data no	1	2	3	4	5	6	7	9	10	11	12	13
x	0.0500	0.2000	0.2500	0.3000	0.4000	0.4300	0.4800	0.6000	0.7000	0.8000	0.9000	0.9500
f(x)	0.863	0.2662	0.2362	0.1687	0.1260	0.1756	0.3290	0.6694	0.4573	0.3320	0.4063	0.3535

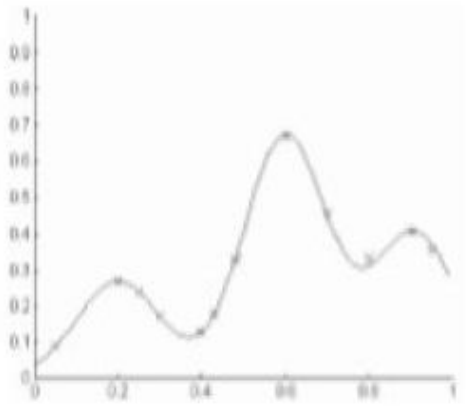


Figure 5.5 Output of the RBF network trained to fir the datapoints given in Table 5.1

5.3 Training RBFNetworks

ThetrainingofaRBFnetworkcanbeformulatedasanomilinearunconstrainedoptimizationproblemgivenbelow:

Giveninputoutputtrainingpatterns$(u^k,y^k)_t$k=1,2...K,Choose**w_{ij}andc$_i$,i=1,2...L.j=1,2...M**

soastominimize

Note that the training problem becomes quadratic once if ci's (radial basis function centers) are known.

5.3.1 Adjusting thewidths

Initssimplestform,allhiddenunitsintheRBFnetworkhavethesamewidthordegreeofsensitivitytoinputs. However,inportionsoftheinputspacewheretherearefewpatterns,itissometimedesirabletohavehiddenunits withawideareaofreception.Likewise,inportionsoftheinputspace,whicharecrowded,itmightbedesirableto haveveryhighlytunedprocessorswithnarrowreceptionfields.Computingtheseindividualwidthsincreasesth e performanceoftheRBFnetworkattheexpenseofamorecomplicatedtrainingprocess.

5.3.2 Adjusting thecenters

Rememberthatinabackpropagationnetwork,allweightsinallofthelayersareadjustedatthesametime.Inradial basisfunctionnetworks,however,theweightsintothehiddenlayerbasisunitsareusuallysetbeforethesecondlaye r
ofweightsisadjusted.Astheinputmovesawayfromtheconnectionweights,theactivationvaluefallsoff.This behaviorleadstotheuseoftheterm"center"forthefirst-layerweights.Thesecenterweightscanbecomputedusing KohonenfeatureMaps,statisticalmethodssuchasK-
Meansclustering,orsomeothermeans.Inanycase,theyare
thenusedtosetthe areasofsensitivityfortheRBFnetwork'shiddenunits,whichthenremainfixed.

5.3.3 Adjusting theweights

Oncethehiddenlayerweightsareset,asecondphaseoftrainingisusedtoadjusttheoutputweights.Thisprocess typicallyusesthestandardsteepestdescentalgorithm.Notethatthetrainingproblembecomesquadraticonceifci' s(radialbasisfunctioncenters)areknown.

Exercise

Q1. Write a short on RadialBasisandRecurrentNeuralNetworks
Q2. Explain the Structure of the RBF Networks.
Q3. What is mean by Hidden layer.
Q4. What is mean by Function approximation.
Q5. Write a short note on DataInterpolation.
Q6. Write a short note on Adjusting the centers.
Q7. Write a short note on Adjusting the widths.

Chapter 6

K-means and LMS algorithms, Comparison of RBF and MLP networks

k-means clustering algorithm

k-meansisoneofthesimplestunsupervisedlearningalgorithmsthatsolvethewellknownclusteringproblem.The procedurefollowsasimpleandeasywaytoclassifyagivendatasetthroughacertainnumberofclusters(assumek clusters)fixedapriori.Themainideaistodefinekcenters,oneforeachcluster.Thesecentersshouldbeplacedina cunningwaybecauseofdifferentlocationcausesdifferentresult.So,thebetterchoiceistoplacethemasmuchas possiblefarawayfromeachother.Thenextstepistotakeeachpointbelongingtoagivendatasetandassociateitto thenearestcenter.Whennopointispending,thefirststepiscompletedandanearlygroupageisdone.Atthispoint weneedtore-calculateknewcentroidsasbarycenteroftheclustersresultingfromthepreviousstep.Afterwehave theseknewcentroids,anewbindinghastobedonebetweenthesamedatasetpointsandthenearestnewcenter.A loophasbeengenerated.Asaresultofthisloopwemaynoticethatthekcenterschangetheirlocationstepbystep untilnomorechangesaredoneorinotherwordscentersdonotmoveanymore.Finally,this algorithmaimsat minimizinganobjectivefunctionknowassquarederrorfunctiongivenby:

$$J(V) = \sum_{i=1}^{C} \sum_{j=1}^{C_i} (||x_i - v_j||)^2$$

where,

'$\|x_i - v_j\|$'istheEuclideandistancebetweenxiandvj.

'c_i'isthenumberofdatapointsinithcluster.

'c' is the number of cluster centers.

Algorithmic steps for k-means clustering

Let$X=\{x_1,x_2,x_3,\ldots\ldots\ldots\ldots,xn\}$bethesetofdatapointsand$V=\{v_1,v_2,\ldots\ldots\ldots\ldots\ldots\ldots v_c\}$bethesetofcenters.

1) Randomly select 'c' clustercenters.
2) Calculalethedistancebetweeneachdatapointandclustercenters.
3) Assignthedatapointtotheclustercenterwhosedistancefromtheclustercenterisminimumofalltheclusterc enters.
4) Recalculatethenewclustercenterusing:

$$V_i = (1/c_i) \sum_{j=1}^{C_i} x_j$$

where, 'c_i' represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

Advantages

1) Fast,robustandeasiertounderstand.
2) Relativelyefficient:O(tknd),wherenis#clusters,dis#dimensionofeachobject,andtis#iterations. Normally,k,t,d<<n.
3) Givesbestresultwhendatasetaredistinctorwellseparatedfromeachother.

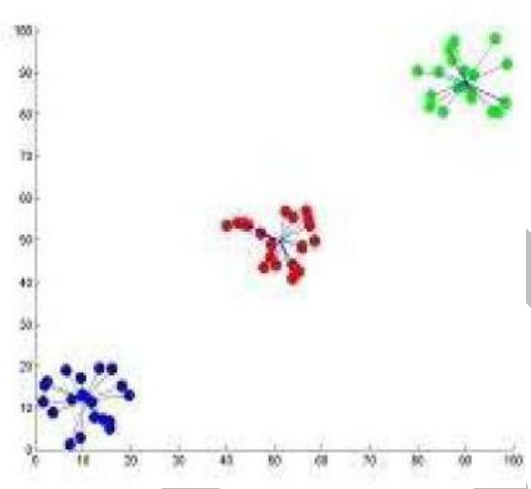


Fig. I : Showing the result of k-means for 'N' = 60 and 'c' = 3

Note : For more detailed figure for k-means algorithm please refer to **k-means figure**sub page.

Disadvantages

1) Thelearningalgorithmrequiresopriorispecificationofthenumberofclustercenters.
2) TheuseofExclusiveAssigmnent-Iftherearetwohighlyoverlappingdatathenk-meanswillnotbeableto resolvethattherearetwoclusters.
3) Thelearningalgorithmisnotinvarianttonon-lineartransformationi.e.withdifferentrepresentationofdata wegetdifferentresults(datarepresentedinformofcartesianco-ordinatesandpolarco-ordinateswillgive differentresults).
4) Euclideandistancemeasurescanunequallyweightunderlyingfactors.
5) Thelearningalgorithmprovidesthelocaloptimaofthesqurederrorfunction.

5) Thelearningalgorithmprovidesthelocaloptimaofthesquarederrorfunction.

6) Randomlychoosingoftheclustercentercannotleadustothefruitfulresult.Pl.referFig.

7) Applicableonlywhenmeanisdefinedi.e.failsforcategoricaldata.

8) Unabletohandlenoisydataandoutliers.

9) Algorithmfailsfornon-lineardataset.

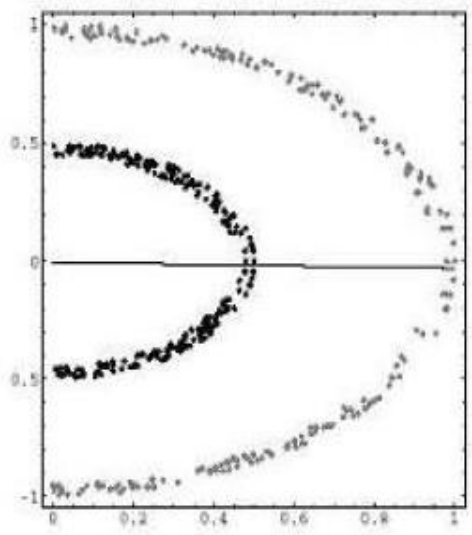


Fig II :Show.the non-linear data set where k-means algorithm fails.

References :

1) AnEfficientk-meansClusteringAlgorithm:AnalysisandImplementationbyTapasKanungo,DavidM. Mount,NathanS.Netanyahu,ChristineD.Piatko,RuthSilvermanandAngelaY.Wu.

2) ResearchissuesonK-meansAlgorithm: AnExperimentalTrialUsingMatlabbyJoaquinPerezOrtega,Ma Del.RocioBooneRojasandMariaJ.SomodevillaGarcia.

3) Thek-méansalgorithm-NotesbyTan,Steinbach,KumarGhosh.

4) http://home.dei.polimi.it/matteucc/Clustering/tutorialhtml/kmeans.html

5) k-meansclusteriiigbykechen.

References

https://sites.google.com/site/dataclusteringalgorithm/k-means-clustering-algorithm

Theleast-mean-square(LMS)algorithmisbasedontheuseofinstantaneousestimatesoftheautocorrelationfunction rt(j.k)andthecross-correlationfunctionrul(k).Theseestimatesarededucteddirectlyfromthedefiningequations (6.8) and (6.7) as follows:

$$p_1(j, k, n) \; -x_1(m)x_1(n) \tag{6.20}$$

and

$$p_{tt}(k, n) - x_1(m)\, x_1(n)$$

Theuseofahatinptandpuisintendedtosignifythatthesequantitiesare"estimates."Thedefinitionsintroducedin Eqs.(6.20)and(6.21)havebeengeneralizedtoincludeanonstationaryenvironment,inwhichcaseallthesensory signalsandthedesiredresponseassumetime-varyingformstoo.Thus,substitutingp$_1$(j,k,n)-x$_1$(m)x$_1$(n)andp$_{tt}$(k, n) inplaceofr,(j,k)andfm(k)inEq.(6.17).Weget

$$w_y^t(n + 1) = w_y^t(n) + \eta[\, x_t(n) f(n) - \sum_{j-1}^{t} w_y(n)x_j(n)\, x_j(n)\,]$$

$$= w_y^t(n) + \eta[d(n) - \sum_{j-1} w_y(m)x_j(n)\,]\, x_j(n)$$

$$= w_y^t(n) + \eta[d(n) - y(m)]x_j(n)]x_j(n),\ k\text{-}1,2,\ldots\ldots P. \tag{6.22}$$

where y(n) is the output of the spatial filter computed at iteration n in accordance with the LMS algorithm; that is,

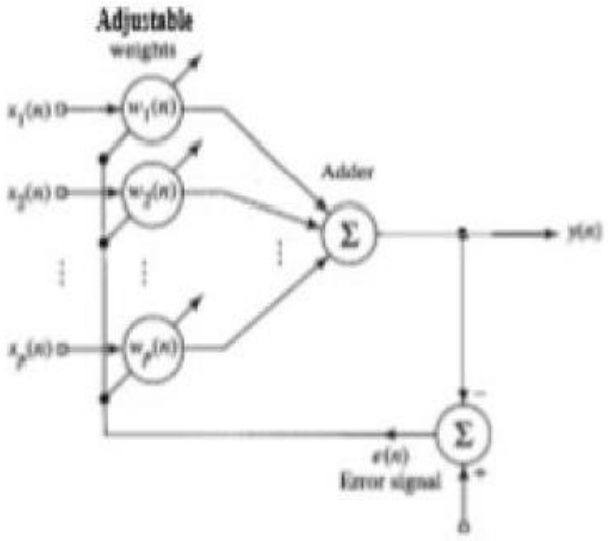
$$d(n) = \sum_{j-1}^{t} w_y(m)x_j(n) \tag{6.23}$$

NotethatinEq.(6.22)wehaveusedwy(n)inplaceofsy(n)toemphasizethefactthatEq.(6.22)invilves"estimates" oftheweightsofthespatialfilter.

Figure6.3illustratestheoperationalenvironmentoftheLMSalgorithm,whichiscompletelydescribedby Eqs.(6.22)and(6.23).AsummaryoftheLMSalgorithmispresentedinTable6.1,whichclearlyillustratesthe simplicityofthealgorithm.Asindicatedinthistable,fortheinitializationofthealgorithm,itiscustomarytosetallthe initialvaluesoftheweightsofthefilterequaltozero.

Inthemethodofsteepestdescentappliedtoa"known"environment,theweightvectorw(n),madeupofthe weights wy(n), wy(n),,w, (n). starts at some initial value w(i), and then follows a precisely defined trajectory (alongtheerrorsurface)thateventuallyterminatesontheoptimumsolutionw,providedthatthelearning-rateparameter nischosenproperly.Inconstrast,intheLMSalgorithmappliedtoan"unknown"environment,theweightvector w(n),respresentingan"estimate"ofw(n),followsarandomtrajectory.Forthisreason,theLMSalgorithmis sometimesreferredtoasa"stochasticgradientalgorithm."AsthenumberofiterationsintheLMSalgorithmapproaches

infinity,w(n)performsarandomwalk(Brownianmotion)abouttheoptimumsolutionw,;seeAppendixD.



Desired response

FIGURE 6.3 Adaptive spatial filter.

Anotherwayofstatingthebasicdifferencebetweenthemethodofsteepestdescentandthe LMSalgorithm isintermsoftheerrorcalculationsinvolved.Atanyiterationn,themethodofsteepestdescentminimizesthemean-squarederrorj(n).Thiscostfunctioninvolvesensembleaveraging,theeffectofwhichistogivethemethodof steepestdescentan"exact"gradientvectorthatimprovesinpointingaccuracywithincreasing*n*.TheLMSalgorithm, ontheotherhand,minimizesaninstantaneousestimateofthecostfunctionj(n).Consequently,thegradientvectorin theLMSalgorithmis"random,"anditspointingaccuracyimproves"ontheaverage"withincreasingn.

ThebasicdifferencebetweenthemethodofsteepestdescentandtheLMSalgorithmmayalsobestatedin termsoftime-domainideas,emphasizingotheraspectsoftheadaptivefilteringproblem.Themethodofsteepest discentminimizesthesumoferrorsquaresx, (n),integratedoverallpreviousiterationsofthealgorithmuptoand including estimates of the autocorrelation function y, and cross-correlation function yn. In constract, theLMS algorithmsimplyminimizestheinstantaneouserrorsquaredy(n),definedas($\frac{1}{2}$)e^2(n),therebyreducingthestorage requirementtotheminimumpossible.Inparticular,itdoesnotrequirestoringanymoreinformationthanispresentin theweightsofthefilter.

TABLE 6.1 Summary of the LMS Algorithm

1. **Initialization.Set**

   $w_y(1) - 0$ **fork-1,2, ... p.**

2. **Filtering. For time n-1,2, ...,compute.**

$$y(n) = \sum_{j-1}^{p} w_y(n)x_j(n)$$

e(n) - d(n) - y(n)

$w^t_1(n + 1) - wx(n) + ny(n)\, x_2(n)$ **fork-1,2, ... p**

ItisalsoimportanttorecognizethattheLMSalgorithmcanoperateinastationeryornonstationaryenvironment. Bya“nonstationery”environmentwemeanoneinwhichthestatisticsvarywithtime.Insuchasituationtheoptimum solutionassumesatime-varyingform,andtheLMSalgorithmthereforehasthetaskofnotonlyseekingtheminimum pointoftheerrorsurfacebutalsotruckingit.Inthiscontext,thesmallerwemakethelearning-rateparametern,the betterwillbethetrackingbehaviourofthealgorithm.However,thisimporovementinperformanceisattainedatthe costofaslowadaptationrate(Haykin,1991;WidrowandStearns,1985).

Signal-Flow Graph Representation of the LMS Algorithm

Equation(6.22)providesacompletedescriptionofthetimeevolutionoftheweightsintheLMSalgorithm.Rewriting thesecondlineofthisequationinmatricform,wemayexpressitasfollows:

$$W(n+1)=w(n)+n[d(n)-x2(n)w(n)]x(n) \quad (6.24)$$

where

$$w(n) - [w (n), w (n), ...,w]^t \quad (6.25)$$

and

$$x(n) - [w (n), w (n), ...,x(n)]^t \quad (6.26)$$

Rearranging terms in Eq. (6.24), we have

$$W(n+1)=[I-nx(n)x^2(n)]w(n)+nx(n)d(n) \quad (6.27)$$

whereIistheidentitymatrix.InusingtheLMSalgorithm,wenotethat

$$W(n) = z^{-1} [w(n)+1)] \quad (6.28)$$

wherez^{-1}istheunit-delayoperatorimplyingstorage.UsingEqs.(6.27_and(6.28),wemaythusrepresenttheLMS algorithmbythesignal-flowgraphdepictedinFig.6.4.

Thesignal-flowgraphofFig.6.4revealsthattheLMSalgorithmisanexampleofastochasticfeedback system.Thepresenceoffeedbackhasaprofoundimpactontheconvergencebehaviouroftheadaptivealgorithm,as discussednext.

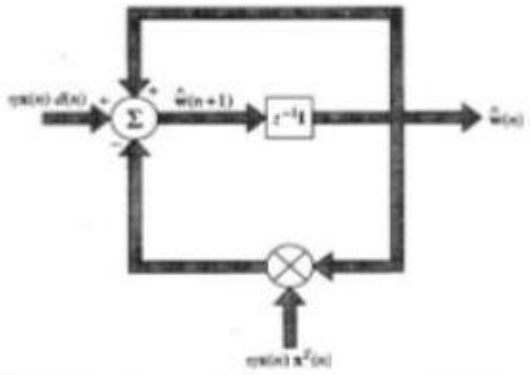


FIGURE 6.4 Signal-flow graph representation of the LMS algorithm.

Q1. What is mean by k-means clustering algorithm.
Q2.What are the Algorithmic steps for k-means clustering.
Q3. Write a Advantages and Dis Advantage of k-means clustering.
Q4. Explain Least-Mean-Square Algorithm.
Q5. Explain Signal-Flow Graph Representation of the LMS Algorithm

Chapter 7

Hopfield networks: energy function, spurious states, error performance.

Hopfield Network

HopfieldneuralnetworkwasinventedbyDr.JohnJ.Hopfieldin1982.Itconsistsofasinglelayerwhichcontains one or more fully connected recurrent neurons. The Hopfield network is commonly used for auto-association and optimizationtasks.

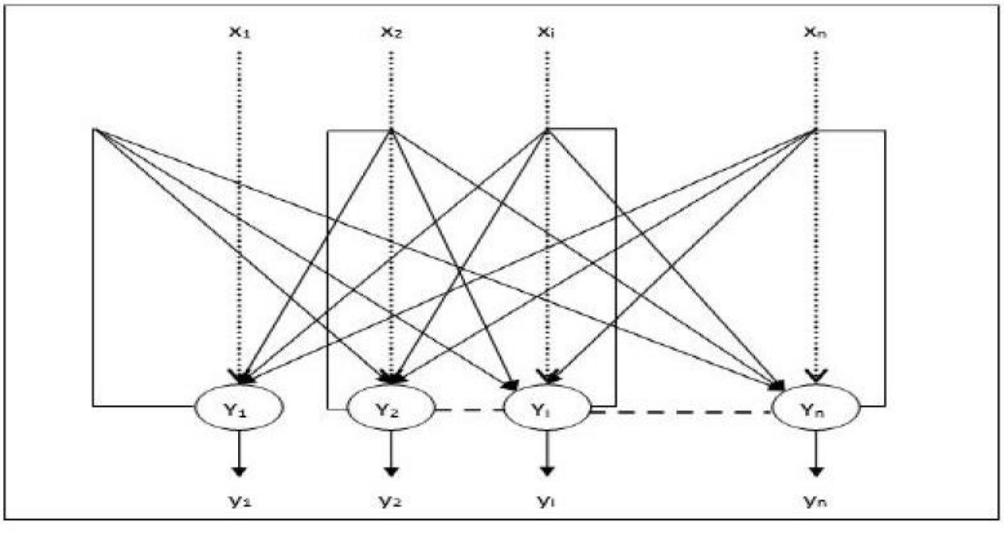
Discrete Hopfield Network

A Hopfield network which operates in a discrete line fashion or in other words, it can be said the input and outputpatternsarediscretevector,whichcanbeeitherbinary(0,1)orbipolar(+1,-1)innature.Thenetworkhas symmetrical weights with no selfconnectionsi.e., $\mathbf{w_{ij} = w_{ji}}$ **and** $\mathbf{w_{ii} = 0}$.

Architecture

Following are some important points to keep in mind about discrete Hopfield network -

* Thismodelconsistsofneuronswithoneinvertingandonenon-invertingoutput.
* Theoutputofeachneuronshouldbetheinputofotherneuronsbutnottheinputofself.
* Weight/connectionstrengthisrepresentedby$\mathbf{w_{ij}}$.
* Connectionscanbeexcitatoryaswellasinhibitory.Itwouldbeexcitatory,iftheoutputoftheneuronissame astheinput,otherwiseinhibitory.
* Weightsshouldbesymmetrical,i.e.$\mathbf{w_{ij}=w_{ji}}$



The output from Y_1 going to Y_2, Y_i and Y_n have the weights w_{12}, w_{1i} and w_{1n} respectively.

Similarly, other arcs have the weights on them.

DuringtrainingofdiscreteHopfieldnetwork,weightswillbeupdated.Asweknowthatwecanhavethebinary input vectors as well as bipolar input vectors. Hence, in both the cases, weight updates can be done with the followingrelation.

Case 1 - Binary input patterns

For a set of binary patterns **s(p), p = 1 to P**

Here, **s(p) =$s_i(p), s_2(p), \ldots\ldots, S_i(P), \ldots\ldots, S_n(P)$**

Weight Matrix is given by

$$w_{ij} = \sum_{p=1} [2s_i(p) - 1] [2s_j(p) - 1] \text{ for } i \neq j$$

Case2- Bipolar inputpatterns

For a set of binary patterns **s(p), p = 1 to P**

Here,s(p)=s1(p),....,s2(p),....,si(p), ..., Sn(P)

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P} [s_i(p)] [s_j(p)] \text{ for } i \neq j$$

Step1- Initializetheweights,whichareobtainedfromtrainingalgorithmbyusingHebbianprinciple.

Step2- Performsteps3-9,iftheactivationsofthenetworkisnotconsolidated.

Step3- Foreachinputvectorx,performsteps4-8.

Step4- Makeinitialactivationofthenetworkequaltotheexternalinputvectorxasfollows-

$$y_i = x_i \text{ for } i = 1 \text{ to}$$

nStep5- ForeachunitY$_i$,performsteps6-9.

Step6- Calculatethenetinputofthenetworkasfollows-

$$Y_{ini} = x_i + \sum_j y_j w_{ji}$$

Step7- Applytheactivationasfollowsoverthenetinputtocalculatetheoutput-

$$\theta i = \begin{cases} 1 & \text{if } yini > \theta i \\ yi & \text{if } yini = \theta i \\ 0 & \text{if } yini < \theta i \end{cases}$$

Here θi is the threshold.

Step9- Testthenetworkforconjunction.

Energy Function Evaluation

An energy function is defined as a function that is bonded and non-increasing function of the state of the system. EnergyfunctionalsocalledLyapunovfunctiondeterminesthestabilityofdiscreteHopfieldnetwork,andischaracterized as follows-

$$E_f = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j w_{ij} - \sum_{i=1}^{n} x_{ii} y + \sum_{i=1}^{n} \theta_i y$$

Condition :In a stable network, whenever the state of node changes, the above energy function will decrease. Suppose when node **i**has changed state from $\mathbf{y}_i^{(k)}$ to $\mathbf{y}_i^{(k+1)}$ then the Energy change $\Delta\mathbf{E}_f$ is given by the following relation.

$$\Delta E_f = E(y_f^{(k+1)}{}_i - E(y^{(k)})_f{}_i$$

$$= -\left(\sum_{j=1}^{n} w\, y^{(k)}_{iji} + x_i - \theta_i\right)\left(y^{(k+1)}_i - y^{(k)}\right)_i$$

$$= -(net)_i \Delta y_i$$

Here $\Delta y_i = y_i^{(k+1)} - y_i^{(k)}$

The change in energy depends on the fact that only one unit can update its activation at a time.

Continuous Hopfield Network

IncomparisonwithDiscreteHopfieldnetwork,continuousnetworkhastimeasacontinuousvariable.Itisalsoused inautoassociationandoptimizationproblemssuchastravellingsalesmanproblem.

Model: Themodelorarchitecturecanbebuildupbyaddingelectricalcomponentssuchasamplifierswhichcan maptheinputvoltagetotheoutputvoltageoverasigmoidactivationfunction.

Energy Function Evaluation :

$$E_f = \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n} y_{ij} y w_{ij} - \sum_{i=1}^{n} x y_i + \frac{1}{\Box}\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ijri} 9_0 \int_0^{y_i} a^{-1}(y)dy$$

Here $\Box$ is gain parameter and $\mathbf{9_{ri}}$input conductance.

Some important points about Boltzmann Machine -

* Theyuserecurrentstructure.
* Theyconsistofstochasticneurons,whichhaveoneofthetwopossiblestates,either1or0.
* Someoftheneuronsinthisareadaptive(freestate)andsomeareclamped(frozenstate).
* IfweapplysimulatedannealingondiscreteHopfieldnetwork,thenitwouldbecomeBoltzmannMachine.

Reference :

https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_hopfield.htm

spurious states

purious states are patterns $x_s \notin P$ $x_s \notin P$, where P_p is the set of patterns to be memorized. In other words, they correspondtolocalminimaintheenergyfunctionthatshouldn't bethere.Theycanbecomposedofvariouscombinations oftheoriginalpatternsorsimplythenegationofanypatternintheoriginalpatternset.Thesetendtobecomepresent when $\alpha = |P|/N$ $\alpha = |P|/N$ (where N_N isthenumberofneurons)becomestoohighforacertainlearningrule.

Itturns outthatspurious statesareimportant forderivingáá inHopfieldnetworks.Becauseweknowthatthe dynamicalupdateequationsalwaysreducetheenergyofasystem,spuriousminimawilltrapthenetworkandretur n incorrectorincompleteresults.Typicallythesespuriousminimahaveahigherenergyandsmallerbasinthanrea l patterns(thoughthisisnotguaranteedwhen $|P| \approx |P|$ istoolarge).Thisnaturallyleadstoastochasticsolutionusinga MonteCarlotypeapproach,whereenoughenergyisgiventotheneuronssothattheyaren't stuckinthelocalminima butdon't jumpoutoftheclosestcorrectpatternminimum(theseareBoltzmannmachines).

Here'ssomehand-wavyintuition.Thelearningrulesprojectthecurrentconfigurationofthenetworkintothesubspace spannedbythepatternvectorsandthencalculatethepatternvectorthatliesclosesttotheprojectedconfiguration vector.Butevenifyouhadcompletelyorthogonalpatterns,youcannotspecifymorepatternsthanthenumberof neurons(becausetheneitheryouduplicateapatternorthenextpatternyouaddisn't orthogonal).

Therealproblemisthatmostlearningrulesgive $\alpha << N$ $\alpha << N$ (e.g.theHebbrulegives $\alpha \approx 0.138$ $\alpha \approx 0.138$ using mean-fieldderivations)becausetheprojectionintothesubspaceisnotorthogonal.Thisisnotanissueifthepatterns

themselvesareorthogonal(i.e.completelyuncorrelated),butthatisveryrarelythecaseinpractice.

Therearewaysto“unlearn”thesespuriousminimatoo.Seethisquestionforgoodreferences,especiallycheckthe Rojasbookwhichisavailableforfreeonline.Also,ifyoucangetyourhandsontheHertzbook,lookatEq.(10.22) whichisthemeanfieldequationwhosesolutionsgivethepossiblestates,includingspuriousones(theyalsogivean explanationforhowtofindthemspecifically).

Exercise:

Q1. Write short note onHopfield Network.
Q2. Explain Discrete Hopfield Network.
Q3. What are the important points of discrete Hopfield network.
Q4. Write a note on Training Algorithm with its cases
Q5. Explain Energy Function Evaluation.
Q6. What is mean by Continuous Hopfield Network.

Chapter 8

Simulated Annealing: The Boltzmann machine, Boltzmann learning rule, Bidirectional Associative Memory.

Simulated Annealing

Statistical Mechanics and the Simulated Annealing

Thestartingpointofstatisticalmechanicsisanenergyfunction.Weconsideraphysicalsystemwithasetof probabilisticstates*x*={x},eachofwhichhasenergyE(x).Forasystemataatemperature$T>0$,itsstate*x*varieswith time,andquantitiessuchasEthatdependonthestatefluctuates.Althoughtheremustbesomedrivingmechanismfor thesefluctuations,partoftheideaoftemperatureinvolvestreatingthemasrandom.Whenasystemisfirstprepared, orafterachangeofparameters,thefluctuationshasonaverageadefinitedirectionsuchthattheenergyEdecreases. However,sometimeslater,anysuchtrendceasesandthesystemjustfluctuatesaroundaconstantaveragevalue. Thenwesaythatthesystemisinthermalequilibrium.

Afundamentalresultfromphysicstellsusthatinthermalequilibriumeachofthepossiblestatesxoccurs withprobability,determinedaccordingtoBoltzmann-Gibbsdistribution,

$$P(x)=\frac{1}{Z}e^{-\frac{E(x)}{T}}$$

where the normalizing factor

$$Z=\sum_{x} e^{-\frac{E(x)}{T}}$$

is called the partition function and it is independent of the state x but temperature.

TheBoltzmann-Gibbsdistributionisusually derivedfromvery general assumptionsaboutmicroscopic dynamics ofmaterials.ThecoefficientTisrelatedtoabsolutetemperatureTaofthesystemas

$$T=K_B T_a$$

wherecoefficientkBisBoltzmann'sconstanthavingvalue1.38x10-16erg/K.Interestinglyenough,thesame

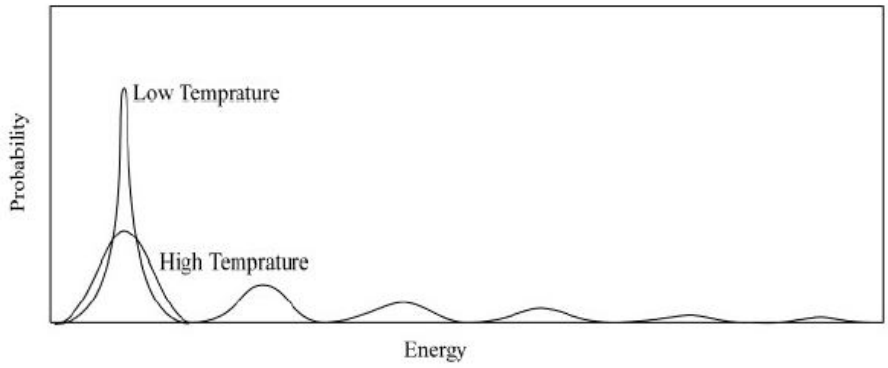
distributioncanalsobeachievedintheviewpointofinformationtheory.AlthoughthetemperatureThasnophysical meaningininformationtheory,itisinterpretedasapseudotemperatureinanabstractmanner.

Givenastatedistributionfunctionfd(x),letP(x(k)=xi)betheprobabilityofthesystembeingatstatexiat thepresenttimek.Furthermore,letP(x(k+1)=xj|x(k)=xi)representtheconditionalprobabilityofnextstatexjgiven thepresentstateisxi.ThenotationP(xi)andP(xj□xi)willbeusedsimplytosdenotetheseprobabilitiesrespectively. Inequilibriumthestatedistributionandthestatetransitionreachesabalancesatisfying:

$$P(x^j \mid x^i)\, P(x^i) = P(x^i \mid x^j)\, P(x^j)$$

Therefore, in equilibrium the Boltzmann Gibbs distribution given by equation (8.1.1) results in :

$$P(x^j / x^i) = \frac{1}{1+e^{\square\square\square}}$$



where □ □ □ □ (x^j) □ E(x^i)

Figure 8.1 Relation between temperature and probability of the states [Kung 93]

TheMetropolisalgorithmprovidesasimplemethodforsimulatingtheevolutionofphysicalsysteminaheat bathtothermalequilibrium[Metropolisetal].ItisbasedonMonteCarloSimulationtechnique,whichaimsto approximatetheexpectedvalue<g(x)>ofsomefunctiong(x)ofarandomvector$x$withagivendensityfunction fd(x). For this purpose several x vectors, say x=Xk k=1..K, are randomly generated according to the density functionfd(x)andthenYkisfoundasYk=g(Xk).Byusingthestronglawoflargenumbers:

$$\lim_{K\to\square} \frac{1}{K} \sum Y^k = < Y^k > = < g(x) >$$

the average of generated Y vectors can be used as an estimate of <g(x)> [Sheldon 1989].

In each step of the Metropolis algorithm, an atom (unit) of the system is subjected to a small random displacement,andtheresultingchangeΔEintheenergyofthesystemisobserved.IfΔE≤0,thenthedisplacementis accepted,andthenewsystemconfigurationwiththedisplacedatomisusedasthestartingpointforthenextstepof

thealgorithm.If,ontheotherhand,ΔE>0,thenthealgorithmproceedsinaprobabilisticmannersothattheconfiguration withthedisplacedatomisacceptedwithaprobabilitygivenby:

$$P(\Delta E) = e^{-\Delta \square \square}$$

Providedenoughnumberoftransitionsin theMetropolisalgorithm,thesystemreachesthermalequilibrium. Thus,byrepeatingthebasicstepsofMetropolisalgorithm,weeffectivelysimulatethemotionsoftheatomsofa physicalsystemattemperatureT.Moreover,thechoiceofP(ΔE)definedinEq.(8.1.8)ensuresthatthermalequilibrium ischaracterizedbytheBoltzmann-GibbsdistributionprovidedinEq.(8.1.5).

ReferringtoEq.(8.1.5),noticethatifP(xi)>P(xj)impliesE(xi)<E(xj),andviceversa.Somaximizingthe probabilityfunctionisequivalentto minimizingtheenergy function.Furthermore,noticethatthis propertyisindependent ofthetemperature,althoughthediscriminationbecomesmoreapparentasthetemperaturedecreases(Figure8.1).

Therefore,thetemperatureparameterTprovidesanewfreeparameterforsteeringthestepsizetowardthe globaloptimum.Withahightemperature,theequilibriumcanbereachedmorerapidly.However,ifthetemperature istoohigh,allthestateswillhaveasimilarlevelofprobability.Ontheotherhand,whenT0,theaverageestate becomesvery closetotheglobalminimum.Thisidea,thoughveryattractiveatthefirstglance,cannotbeimplemented directlyinpractice.Infact,withalowtemperature,itwilltakeaverylongtimetoreachequilibriumand,more seriously,thestateismoreeasilytrappedbylocalminima.Therefore,itisnecessarytostartatahightemperatureand thendecreaseitgradually.Correspondingly,theprobablestatethengraduallyconcentratearoundtheglobalminimum (Figure8.2).

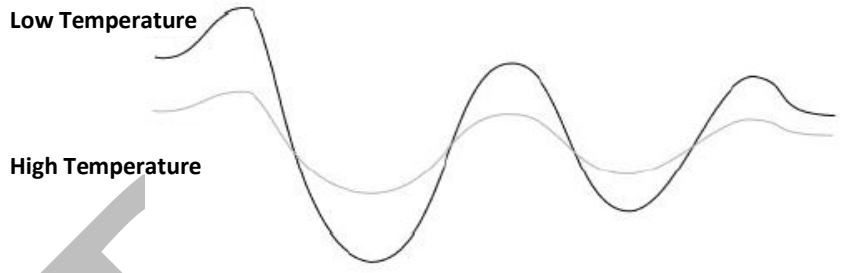


Figure 8.2 The energy levels adjusted for high and low temperature

Thishasananalogywithmetallurgicalannealing,inwhichabodyofmetalisheatedneartoitsmeltingpoint andisthenslowlycooledbackdowntoroomtemperature.Thisprocesseliminatesdislocationsandothercrystal latticedisruptionsbythermalagitationathightemperature.Furthermore,itpreventstheformation ofnewdislocations bycoolingthemetalveryslowly.Thisprovidesnecessarytimetorepairanydislocationsthatoccurasthetemperature drops.Theessenceofthisprocessisthatglobalenergyfunctionofthemetalwilleventuallyreachanabsoluteminimum value.

Ifthematerialiscooledrapidly,itsatomsareoftencapturedinunfavorablelocationsinthelattice.Oncethe temperaturehasdroppedfarbelowthemeltingpoint,thesedefectssurviveforever,sinceanylocal rearrangementsof atomscostsmoreenergythanwhateveravailableinthermalfluctuations.Theatomiclatticethusremainscapturedin alocalenergyminimum.Inordertoescapefromlocalminimaandtohavethelatticeintheglobalenergyminimum,the thermalfluctuationscanbeenhancedbyreheatingthematerialuntilenergy-consuminglocalrearrangementsoccurat areasonablerate.Thelatticeimperfectionsthenstarttomoveandannihilate,untiltheatomiclatticeisfreeofdefects- exceptforthosecausedbythermalfluctuations.Thesecanbegraduallyreducedifthetemperatureisdecreasedso slowly thatthermalequilibriumismaintainedatalltimesduringthecoolingprocess.Howmuchtime mustbespentfor thecoolingprocessdependsonthespecificsituation.A greatdealofexperienceisrequiredtoperformtheannealing inanoptimalway.Ifthetemperatureisdecreasedquickly,somethermalfluctuationsarefrozenin.Ontheotherhand, ifoneproceedstooslowly,theprocessneverends.

Theamazingthingaboutannealingisthatthestatisticalprocessofthermalagitationleadstoapproximatelythesamef inalenergystate.Thisresultisindependentoftheinitialconditionofthemetalandanyofthedetailsofthe statisticalannealingprocess.Themathematicalconceptofsimulatedannealingderivesfromananalogywiththis physicalbehavior.

Thesimulatedannealingalgorithm,isavariantoftheMetropolisalgorithminwhichthetemperatureistime dependent.Inanalogywithmetallurgicalannealing,itstartswithahightemperatureandgraduallydecreasesit.At eachtemperature,itappliesseveraltimestheupdaterulegivenbyEq.(8.1.8).Anannealingschedulespecifiesafinite sequenceoftemperaturevaluesandafinitenumberoftransitionsattemptedateachvalueofthetemperature.The annealingscheduledevelopedby[Kirkpatricketal1983]isasfollows.

TheinitialvalueT0ofthetemperatureischosenhighenoughtoensurethatvirtuallyallproposedtransitionsbe acceptedbythesimulatedannealingalgorithm.Thenthecoolingisperformed.Ateachtemperature,enoughtransitions areattemptedsothatthereisapredeterminednumberoftransitionsperexperimentontheaverage.Attheend,the systemisfrozenandannealingstopsifthedesirednumberofacceptancesisnotachievedatpredeterminednumberof successivetemperatures.Inthefollowing,weprovidetheannealingprocedureinmoredetail:

SIMULATED ANNEALING

Step1.SetInitialvalues:assignahighvaluetotemperatureasT(0)=T_0,decide onconstantsK_T,$K_A$andK_S,Typicalvaluesforwhichare0.8<K_T<0.99, K_A=10. and K_S=3.

Step2.Decrementthetemperature:T(k)=K_T,(k-1)whereK_T,isaconstant smallerbutclosetounity.

Step3.Attemptenoughnumberoftransitionsateachtemperature,sothatthere are$K_A$acceptedtransitionsperexperimentontheaverage.

Step4.Stopifthedesirednumberofacceptancesisnotachievedat$K_S$successive temperatureselserepeatsteps2and3.

AveryimportantpropertyofsimulatedannealingisitsasymptoticconvergenceIthasbeenprovedin[GemanandGe man84]thatifT(k)atiterationkischosensuchthatitsatisfies

$$T(k) \geq \frac{T_0}{\log(1+k)}$$

providedtheinitialtemperatureT_0ishighenough,thenthesystemwillconvergetotheminimumenergyconfiguration. Themaindrawbackofsimulatedannealingisthelargeamountofcomputationaltimenecessaryforstochasticrelaxation.

Manyelementarytransformationsareperformedateachtemperaturestepinordertoreachanearequilibriumstate.

Themaindrawbackofsimulatedannealingisthelargeamountofcomputationaltimenecessaryforstochastic relaxation.Many elementary transformationsareperformedateachtemperaturestepinordertoreachnearequilibrium state.

Exercise:

Q1. What is mean by Simulated Annealing.

Q2. Explain Statistical Mechanics.

Q3. Explain Simulated Annealing.

Q4. Write a short note on Boltzmann learning rule.

Q5. Explain Bidirectional Associative Memory.

Chapter 9

The Boltzmann machine, Boltzmann learning rule

9.2 Boltzmann Machine

Boltzmannmachine[Hintonetal83]isaconnectionistmodelhavingstochasticnature.Thestructureofthe BoltzmannmachineissimilartoHopfieldnetwork,butitaddssomeprobabilisticcomponenttotheoutputfunction.It usessimulatedannealingconcepts,inspiteofthedeterministicnatureinstatetransitionoftheHopfieldnetwork [Hintonetal83,Aartsetal1986,AllwrightandCarpenter1989,LaarhovenandAarts1987].

ABoltzmannmachinecanbeviewedasarecurrentneuralnetworkconsistingofNtwostateunits.Depending onthepurpose,thestatescanbechosenfrombinaryspace,thatis$x\in\{0,1\}^N$orfrombipolarspace$x\in\{-1,1\}^N$. TheenergyfunctionoftheBoltzmannmachineis:

$$E_{(x)} = -\frac{1}{2}\sum_i\sum_j w_{ij}x_i x_j - \sum_i \theta_i x_i$$

Theconnectionsaresymmetricalbydefinition,thatis$w_{ij}=w_{ji}$.Furthermoreinthebipolarcase,theconvergence ofthemachinerequires$w_{ii}=0$(orequivalentlly$\theta_i=0$).However inthebinarycaseself-loopsareallowed.

TheobjectiveofaBoltzmannmachineistoreachtheglobalminimumofitsenergyfunction,whichisthestate havingminimumenergy.Similartosimulatedannealingalgorithm,thestatetransitionmechanismofBoltzmannMachine usesastochasticacceptancecriterion,thusallowingittoescapefromitslocalminima.InasequentialBoltzmann machine,unitschangetheirstatesonebyone,whiletheychangestatealltogetherinaparallelBoltzmannmachine.

LetX denotethestatespaceofthemachine,thatis thesetofallpossiblestates.Amongthese,thestate vectors differing only one bit are called neighboring states. The neighborhood $N_x\subset X$ is definedas the set ofallneighboringstatesofx.Letaxjtodenotetheneighboringstatethatisobtainedfromxbychangingthestate ofneuronj.Hence,inbinarycasewehave

$$x_i^j = \begin{cases} x_i & \text{if } i\neq j \\ 1-x_i & \text{if } i=j \end{cases} \qquad x\in(0,1)^n,\ x^j\in N_x$$

In bipolar case, this becomes:

$$x_i^j = \begin{cases} x_i & \text{if } i\neq j \\ -x_i & \text{if } i=j \end{cases} \qquad x\in(-1,1)^n,\ x^j\in N_x$$

The difference in energy when the global state of the machine is changed from x to x^j is :

Note that the contribution of the connections w_{km} k≠j, m≠j, to $E(x)$ and $E(x^j)$ is identical, furthermore wij=wij. For the binary case, by using equations (9.2.1) and (9.2.2), we obtain

$$\Delta E(x^j | x) = (2x_j - 1)\left(\sum_i w_{ij}x_i + \theta_j\right) \qquad x \in \{0,1\}^N$$

For the bipolar case it is

$$\Delta E(x^j | x) = (2x_j)\left(\sum_i w_{ij}x_i + \theta_j\right) \qquad x \in \{-1,1\}^N, \; w_{ii} = 0$$

Therefore,thechangeintheenergycanbecomputedbyconsideringonlylocalinformation.Inasequential Boltzmannmachine,atrialforastatetransitionisatwo-stepprocess.Givenastatex,firstaunitjisselectedasa candidatetochangestate.Theselectionprobabilityusuallyhasuniformdistributionovertheunits.Thenaprobabilistic functiondetermineswhetherastatetransitionwilloccurornot.Thestatexjisacceptedwithprobability

$$P(x^j | x) = \frac{1}{1+e^{\Delta E(x^j|x)/T}}$$

whereTisacontrolparameterhavinganalogyintemperature.Initiallythetemperatureissetlargeenoughtoaccept almostallstatetransitionswithprobabilitycloseto0.5,andthenTisdecreasedintimetozero(Figure9.3).Witha propercoolingschedule,thesequentialBoltzmannmachineconvergesasymptoticallytoastatehavingminimum energy.

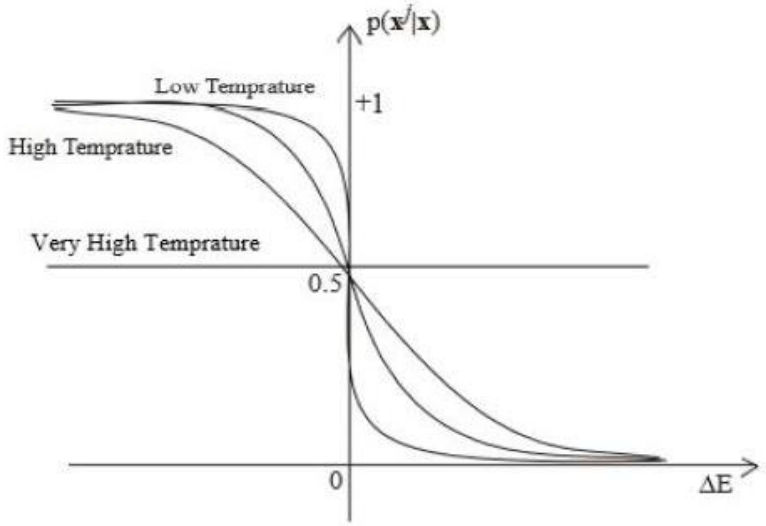


Figure 9.3. Acceptance probability in Boltzmann machine for different temperatures

ABoltzmannmachinestartsexecutionwitharandominitialconfiguration.Initially,thevalueofTisverylarge. Acoolingscheduledetermineshowandwhentodecrementthecontrolparameter.AsT → 0,lessandlessstate transitionsoccur.Ifnostatetransitionsoccurforaspecifiednumberoftrials,itisdecidedthattheBoltzmannmachine hasreachedthefinalstate.

A state x* ∈ ¸ *X* is called a locally minimal state, if

$\square E(x^{*j} | x^*) \geq 0$ **j = 1..N**

Notethat,alocalminimumisastatewhoseenergycannotbeincreasedbyasinglestatetransition.Lettheset ofalllocalminimabedenotedbyX*.WhiletheHopfieldnetworkistrappedmostlyinoneoftheselocalminima,the Boltzmannmachinecanescapefromthelocalminimabecauseofitsprobabilisticnature.Althoughthemachin e asymptoticallyconvergestoaglobalminimum,thefinite-timeapproximationoftheBoltzmannMachineprevents guaranteeingconvergencetoastatewithminimumenergy.However,stillthefinalstateofthemachinewillbeanearly minimumoneamongX*.

UseofBoltzmannmachineasaneuraloptimizerinvolvestwophasesasitisexplainedfortheHopfield networkinChapter4.Inthefirstphase,theconnectionweightsaredetermined.Forthispurpose,anenergyfunctio n forthegivenapplicationisdecided.Inthenon-constrainedoptimizationapplications,theenergyfunctioncanbe directlyobtainedbyusingthecostfunction.However,inthecaseofconstrainedoptimization,theenergyfunctio n mustbederivedusingboththeoriginalcostfunctionandtheconstraints.Thenextstepistodeterminetheconnection weights{w_{ij}}byconsideringthisenergyfunction.Theninthesecondphase,themachinesearchestheglobalminimu m throughtheannealingprocedure.

Exercise:

Q1. Write short note on Boltzmann Machine.
Q2. Explain what is mean by Boltzmann learning.
Q3. What are the rule of Boltzmannlearning rule.
Q4. Write a short note on Learning Per Pattern.
Q5. Explain the Structure of the RBF Networks.

Chapter 10

Bidirectional Associative Memory

BIDIRECTIONALASSOCIATIVE MEMORY (BAM)

Severalversionsoftheheteroassociativerecurrentneuralnetwork,orbidirectionalassociativememory(BAM), developedbyKosko(1988,1992a).-Abidirectionalassociativememory[Kosko,1988]storesasetofpattern associationsbysummingbipolarcorrelationmatrices(annbymouterproductmatrixforeachpatterntobestored). - Thearchitectureofthenetconsistsoftwolayersofneurons,connectedbydirectionalweightedconnectionpaths. - Thenetiterates,sendingsignalsbackandforthbetweenthetwolayersuntilallneuronsreachequilibrium(i.e.,until eachneuron'sactivationremainsconstantforseveralsteps).-Bidirectionalassociativememoryneuralnetscan respondtoinputtoeitherlayer.-Becausetheweightsarebidirectionalandthealgorithmalternatesbetweenupdating theactivationsforeachlayer,weshallrefertothelayersastheX-layerandtheY-layer(ratherthantheinputand outputlayers).

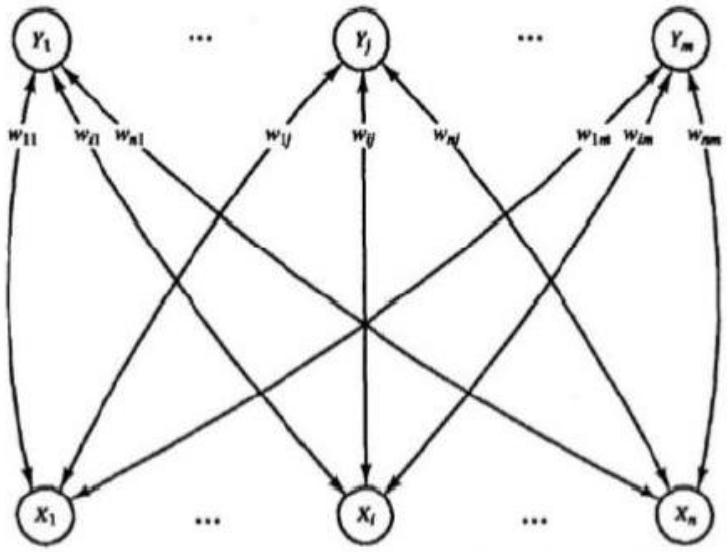


Figure : Bidirectional Associative Memory **(BAM)**

Architecture-Thesingle-layernonlinearfeedbackBAMnetwork(withheteroassociativecontentaddressablememory) hasnunitsinitsXlayerandmunitsinitsY-layer.-Theconnectionsbetweenthelayersarebidirectional;i.e.,ifthe weightmatrixforsignalssentfromtheX-layertotheY-layerisW,theweightmatrixforsignalssentfromtheY-layer totheX-layerisW^T.

Discrete BAM

Thetwobivalent(binaryorbipolar)formsofBAMarecloselyrelated.Ineach,theweightsarefoundfromthesum oftheouterproductsofthebipolarformofthetrainingvectorpairs.Also,theactivationfunctionisastepfunction, withthepossibilityofanonzerothreshold.Thebipolar

vectors improve the performance of the net.

* The weight matrix to store a set of input and target vectors s(p) : t(p), p = 1, . . . , P, where

  s(p) = (s1(p), …., si(p), …., sn(p)) ;

  t(p) = (t1(p), ……., tj(p), ……., tm(p))

  can be determined by the Hebb rule.

* Theformulasfortheentriesdependonwhetherthetrainingvectorsarebinaryorbipolar.

  For binary input vectors, the weight matrix

  W = {wij} is given by

$$w_{ij} = \sum_{p=1}^{p} (2s_i(p) - 1)\,(2t_j(p) - 1)$$

- Forbipolarinputvectors,theweightmatrixW=***{wij}***isgivenby

$$w_{ij} = \sum_{p=1}^{p} s_i(p)t_j(p)$$

ActivationFunction:Theactivationfunctionforthediscrete BAMistheappropriatestepfunction,dependingon whetherbinaryorbipolarvectorsareused.

♦ Forbinaryinputvectors,theactivationfunctionfortheY-layeris

$$y_j = \begin{cases} 1 & \text{if } y\_in_j > 0 \\ y_j & \text{if } y\_in_j = 0 \\ 0 & \text{if } y\_in_i < 0 \end{cases}$$

and the activation function for the X-layer is

$$x_j = \begin{cases} 1 & \text{if } x\_in_j > 0 \\ x_j & \text{if } x\_in_j = 0 \\ 0 & \text{if } x\_in_i < 0 \end{cases}$$

♦ Forbinaryinputvectors,theactivationfunctionfortheY-layeris

$$y_j = \begin{cases} 1 & \text{if } y\_in_j > 0_j \\ y_j & \text{if } y\_in_j = 0_j \\ -1 & \text{if } y\_in_j < 0_j. \end{cases}$$

and the activation function for the X-layer is

$$x_i = \begin{cases} 1 & \text{if } x\_in_i > 0_i \\ x_i & \text{if } x\_in_i = 0_i \\ -1 & \text{if } x\_in_i < 0_i. \end{cases}$$

Notethatifthenetinputisexactlyequaltothethresholdvalue,theactivationfunction"decides"toleavetheactivation ofthatunitatitspreviousvalue.

□ Theactivationsofallunitsareinitializedtozero.

□ ThefirstsignalistobesentfromtheX-layertotheY-layer.However,iftheinputsignalfortheX-layeristhe zerovector,theinputsignaltotheY-layerwillbeunchangedbytheactivationfunction,andtheprocesswillbe thesameasifthefirstpieceofinformationhadbeensentfromtheY-layertotheX-layer.

* Signals are sent only from one layer to the other at any step of the process, not simultaneously in both directions.

Algorithm

1. Initializetheweightstostoreasetof Pvectors;initializeallactivationsto0
2. Foreachtestinginput,doSteps3-7.

3a. PresentinputpatternxtotheX-layer,(i.e.,setactivationsofX-layertocurrentinputpattern). 3b. PresentinputpatternytotheY-layer,(Eitheroftheinputpatternsmaybethezerovector.)

4. Whileactivationsarenotconverged,doSteps5-7.
5. UpdateactivationsofunitsinY-layer:

Compute net inputs :

$$y\_in_j = \sum_i w_{ij} x_i$$

Compute activations :

$$y_j = f(y.in_j)$$

Send signal to X-layer.

6. UpdateactivationsofunitsinX-layer:

Compute net inputs :

$$x\_in_i = \sum_j w_{ij} y_j$$

Compute net inputs :

$$x_i = f(x.in_i)$$

Send signal to Y-layer.

7. Test for convergence:

If the activation vectors x and y have reached equilibrium, then stop; otherwise, continue.

Continuous BAM

Acontinuousbidirectionalassociativememory[Kosko,1988]transformsinputsmoothlyandcontinuouslyintooutput intherange[0,1]usingthelogisticsigmoidfunctionastheactivationfunctionforallunits.

- Forbinaryinputvectors(s(p),t(p)),p=1,2,...,P,theweightsaredeterminedbytheformula

$$w_{ij} = \sum_{p=1}^{p} (2s_i(p) - 1)\ (2t_j(p) - 1)$$

- Theactivationfunctionisthelogisticsigmold

$$f(y, in_j) = \frac{1}{1 + \exp(-y\_in_j)}$$

- whereabiasisincludedincalculatingthenetinputtoanyunitandcorrespondingformulas applyfortheunitsintheX-layer.

$$y\_in_j = b_j + \sum_i w_{ij} x_i$$

AnumberofotherformsofBAMshavebeendeveloped.Insome,theactivationschangebasedonadifferential equationknownasCohenGrossbergactivationdyanamics(Cohen&Grossberg,1983).

Application

Example-11 :A BAM net to associate letters with simple bipolar codes

Considerthepossibilityofusinga(discrete)BAMnetwork(withbipolarvectors)tomaptwosimpleletters (givenby5x3patterns)tothefollowingbipolarcodes:

he target output vector t for letter A is [-1 1] and for the letter C is [1 1],

The weight matrices are :

(to store $A \rightarrow$ -1 1)

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$$

($C \rightarrow$ 1 1)

$$\begin{pmatrix} -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

(w, tostore both)

$$\begin{pmatrix} 0 & -2 \\ 0 & 2 \\ 2 & 0 \\ 0 & 2 \\ 0 & -2 \\ -2 & 0 \\ 0 & 2 \\ -2 & 0 \\ -2 & 0 \\ 0 & 2 \\ 0 & -2 \\ -2 & 0 \\ -2 & 0 \\ 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Toillustratetheuseofa BAM,wefirstdemonstratethatthenetgivesthecorrectyvectorwhenpresentedwiththex vectorforeitherthepatternAorthepatternC:

INPUT PATTERN A

[-12-1 1-11 111 1 -11 1 -1 1] w =[-1416] → [-11]

INPUT PATTERN C

[-111 1-1 -1 1-1 -1 1 -1 -1 -1 1 1] w= [-1416] → [11]

Toseethebidirectednatureofthenet,observethattheYvectorscanalsobeusedasinput.Forsignalssentfromthe Y-layertotheX-layer,theweightmatrixisthetransposeofthematrixW,i.e.w^T.

For the input vector associated with pattern A. namely. (-1, 1), we have

[-1 1] w^T = [-2 2 -2 2 -2 2 2 2 2 2 -2 2 2 2 -2 2]

→ [-11-1 1-11 111 1-11 1 -11]

This is pattern A.

Similarly, if we input the vector associated with pattern C, namely, (1,

1).We obtain→[1 1] w^T [-111 1-1-1 1-1-1 -1 11]

Exercise:

Q1. Write a short on Bidirectional Associative Memory.
Q2. What is mean by Discrete BAM.
Q3. Explain Algorithm of BAM.
Q4. Explain Continuous BAM.
Q5. What are the Application of BAM.

Chapter 11

Fuzzy Set, Properties, Operations on Fuzzy sets, Fuzzy relations

Introduction

Fuzzy Logic

Fuzzy set theory was developed by Lotfi A. Zadeh [Zadeh, 1965], professor for computer science at the University of California in Berkeley, to provide a mathematical tool for dealing with the concepts used in natural language (linguistic variables). Fuzzy Logic is basically a multivalued logic that allows intermediate values to be defined between conventional evaluations.

However, the story of fuzzy logic started much more earlier . To devise a concise theory of logic, and later mathematics, Aristotle posited the so-called ¡¨Laws of Thought¡¨. One of these, the ¡¨Law of the Excluded Middle,¡¨ states that every proposition must either be True (T) or False (F). Even when Parminedes proposed the first version of this law (around 400 Before Christ) there were strong and immediate objections: for example, Heraclitus proposed that things could be simultaneously True and not True. It was Plato who laid the foundation for what would become fuzzy logic, indicating that there was a third region (beyond T and F) where these opposites ¡¨tumbled about.¡¨ A systematic alternative to the bi-valued logic of Aristotle was first proposed by ¢Gukasiewicz around 1920, when he described a three-valued logic, along with the mathematics to accompany it. The third value he proposed can best be translated as the term ¡¨possible,¡¨ and he assigned it a numeric value between T and F. Eventually, he proposed an entire notation and axiomatic system from which he hoped to derive modern mathematics.

Later, he explored four-valued logics, five-valued logics, and then declared that in principle there was nothing to prevent the derivation of an infinite-valued logic.¢Gukasiewicz felt that three- and infinite-valued logics were the most intriguing, but he ultimately settled on a fourvalued logic because it seemed to be the most easily adaptable to Aristotelian logic. It should be noted that Knuth also proposed a threevalued logic similar to Lukasiewicz¡¦s, from which he speculated that mathematics would become even more elegant than in traditional bi-valued logic. The notion of an infinite-valued logic was introduced in Zadeh¡¦s seminal work ¡¨Fuzzy Sets¡¨ where he described the mathematics of fuzzy set theory, and by extension fuzzy logic. This theory proposed making the membership function (or the values F and T) operate over the range of real numbers [0, 1].

New operations for the calculus of logic were proposed, and showed to be in principle at least a generalization of classic logic. Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. The conventional approaches

to knowledge representation lack the means for representating the meaning of fuzzy concepts. As a consequence, the approaches based on first order logic and classical probability theory do not provide an appropriate conceptual framework for dealing with the representation of commonsense knowledge, since such knowledge is by its nature both lexically imprecise and noncategorical. The developement of fuzzy logic was motivated in large measure by the need for a conceptual framework which can address the issue of uncertainty and lexical imprecision. Some of the essential characteristics of fuzzy logic relate to the following (Zadeh, 1992): In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning. In fuzzy logic, everything is a matter of degree. In fuzzy logic, knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables. Inference is viewed as a process of propagation of elastic constraints. Any logical system can be fuzzified. There are two main characteristics of fuzzy systems that give them better performance for specific applications. Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system with a mathematical model that is difficult to derive. Fuzzy logic allows decision making with estimated values under incomplete or uncertain information.

Theory has been attacked several times during its existence. For example, in 1972 Zadeh's colleague R. E. Kalman (the inventor of Kalman filter) commented on the importance of fuzzy logic: "...Zadeh's proposal could be severely, fericiously, even brutally criticized from a technical point of view. This would be out of place here. But a blunt question remains: Is Zadeh presenting important ideas or is he indulging in wishful thinking?..."

The heaviest critique has been presented by probability theoreticians and that is the reason why many fuzzy logic authors (Kosko, Zadeh and Klir) have included the comparison between probability and fuzzy logic in their publications. Fuzzy researchers try to separate fuzzy logic from probability theory, whereas some probability theoreticians consider fuzzy logic a probability in disguise.

Fuzzy Set

Since set theory forms a base for logic, we begin with fuzzy set theory in order to ¡§pave the way¡¨ to fuzzy logic and fuzzy logic systems.

6.1.1 Set - Theoretical Operations and Basic Definitions

In classical set theory the membership of element x of a set A (A is a crisp subset of universe X) is defined by

$$_A(x) = \begin{cases} 0, \text{ if } x \notin A, \\ 1, \text{ if } x \in A \end{cases}$$

6.1

The element either belongs to the set or not. In fuzzy set theory, the element can belong to the set partially with a degree and the set does not have crisp boundaries. That leads to the following definition.

Definition 6.1.1 (fuzzy set, membership function) Let X be a nonempty set, for example $X=R^n$, and be called theuniverse of discourse. A fuzzy $A \subset X$ is characterized by the membership function

$$A : X \rightarrow [0, 1] =$$

6.2

Where $\mu_A(X)$ is a grade (degree) of membership of X *x* in set ***A***.

From the definition we can see that the fuzzy set theory is a generalized set theory that includes the classical set theory as a special case. Since {0,1} *x* [0,1] , crisp sets are fuzzy sets. Membership function (2.2) can also viewed as a distribution of truth of a variable. In literature fuzzy set A is often presented as a set of ordered pairs :

$$\boldsymbol{A} = \{(x, \mu_A(x)) \text{ x } \chi X\}$$

6.3

where the first part determines the element and the second part determines the grade of membership. Another way to describe fuzzy set has been presented by Zadeh, Dubois and Prade. If ***X*** is infinite, the fuzzy set ***A*** can be expressed by

$$A = \int \mu_A(x)/ x$$

6.4

If ***X*** is finite, ***A*** can be expressed in the form

$$\boldsymbol{A} = \mu_A(x_i)/x_1 + \ldots + \mu_A(x_n) = \sum_{i=1}^{n} \mu_A(x_i)/x_i$$

6.5

Note : Symbol$\int$in (2.4) has nothing to do with integral (it denotes an uncountable enumeration) and / denotes atuple. The plus sign represents the union. Also note that fuzzy sets are membership functions. Nevertheless, we may still use the set theoretic notations like ***A*** ***B***.

This is the name of a fuzzy set given by $\mu_{A \cup B}$.

Example 6.1.1

Discrete case: $\mu_A = 0.1/x_1 + 0.4/x_2 + 0.8/x_3 + 1.0/x_4 + 0.8/x_5 + 0.4/x_6 + 0.1/x_7$

Continuous case : $\mu_A(x) = \begin{cases} 1 - \dfrac{x - c}{h} & , x \chi [c - h, h] \\ \dfrac{x - c}{h} & , x \chi [c, c + h] \end{cases}$

0 , otherwise

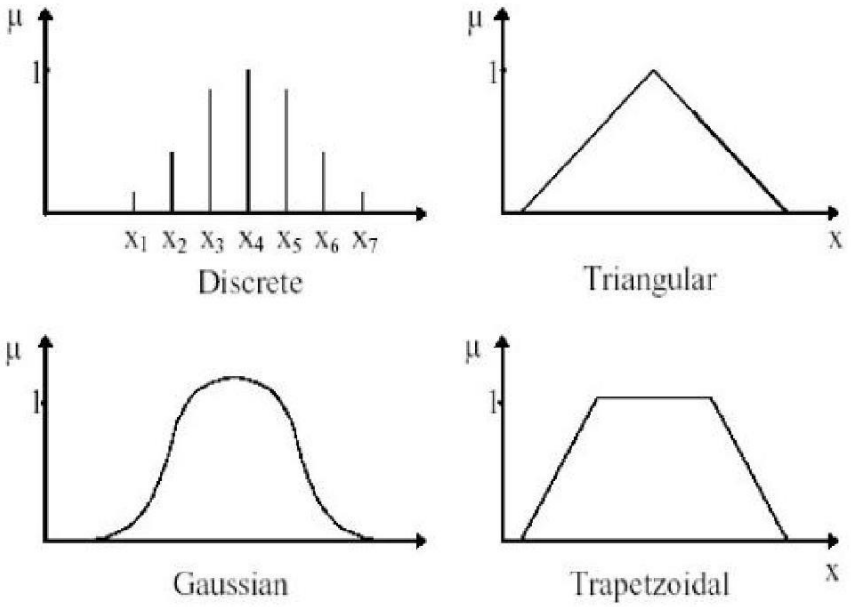


Figure 6.1 A discrete and continuous membership functions

Definition 6.1.2 (support) The support of a fuzzy set A is the crisp set that contains all elements of A with non-zeromembership grade :

$$\text{supp}(\boldsymbol{A}) = \{x \chi X \mu_A | (x) > 0\}$$

6.6

If the support is finite, it is called compact support. If the support of fuzzy set A consists of only one point, it is called a fuzzy singleton. If the membership grade of this fuzzy singleton is one, A is called a crisp singleton 'Zimmermann, 1985'.

Definition 6.1.3 (core) The core (nucleus, center) of a fuzzy set A is defined by

$$\text{core}(\boldsymbol{A}) = \{x \chi X \mu(x) = 1\}$$

Definition 6.1.4 (*height*) The height of a fuzzy setAonXis defined by

$$\text{hgt}(\boldsymbol{A}) = \sup_{xz^{X}} \mu_A(x)$$

and *A*is called normal if$\text{hgt}(\boldsymbol{A}) = 1$, and subnormal if**$\text{hgt}(\boldsymbol{A}) < 1$.**

Note : Non-empty fuzzy set can be normalized by dividing$\mu_A(x)$by $\sup_X \mu_A(x)$. Normalizing of A can beregarded as a mapping from fuzzy sets to possibility distributions :

Normal **(*A*)**$e\nu \div \mu_A$ [Joslyn, 1994].

The relation between fuzzy set membership function , possibility distribution and probability distribution

p : The definition $p_A(x) \div \mu_A(x)$ could hold, if$\mu$is additively normal. Additively normal means here that the stochasticnormalization

$$\int \mu_A(x)dx = 1$$

would have to be satisfied. So it can be concluded that any given fuzzy set could define either a probability distribution or a possibility distribution, depending on the properties of μ. Both probability distributions and possibility distributions are special cases of fuzzy sets. All general distributions are in fact fuzzy sets [Joslyn, 1994].

Definition 6.1.5 (convex fuzzy set)***A***fuzzy set***A***is convex if

$$6x, y \quad x \quad and \quad 6Z, \quad [0,1]$$

$$\mu_A(Zx + (1 - Z)\, y) \square \min(\mu_A(x), \mu_A(y))$$

Definition 6.1.6 (width of a convex fuzzy set) The width of a convex fuzzy set***A***is defined by

width (A) = sup(supp (A))- inf(supp (A))

Definition 6.1.7 (α-cut) Theα-cut of a fuzzy set $\boldsymbol{A}$ is defined by

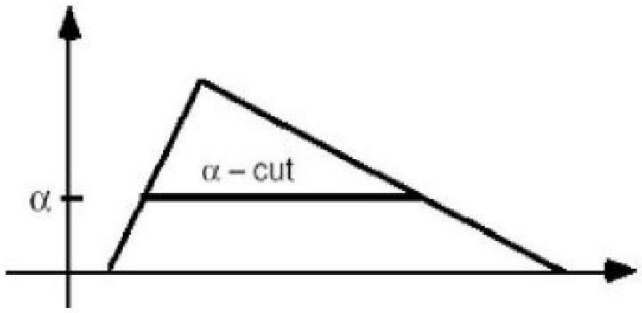


Figure 2.2 An fÑf{ cut of a triangular fuzzy number

Definition 6.1.8 (fuzzy partition) $\boldsymbol{A}$ set of fuzzy sets is called fuzzy partition if

$$6x, y \quad x$$

$$\sum_{i=1}^{N_A} \mu \mid_{Ai} = 1$$

provided A*j* are nonempty and subsets of *x*.

Definition 6.1.9 (*fuzzy number*) ***A*** fuzzy set (subset of a real line **R**) is a fuzzy number, if the fuzzy set is convex,normal, membership function is piecewise continuous and the core consists of *G.* one value only. The family of fuzzy numbers is . In many situations people are only able to characterize numeric information imprecisely. For example, people use terms such as, about 5000, near zero, or essentially bigger than 5000. These are examples of what are called *fuzzy* numbers. Using the theory of fuzzy subsets we can represent these fuzzy numbers as fuzzy subsets of the set of real numbers.

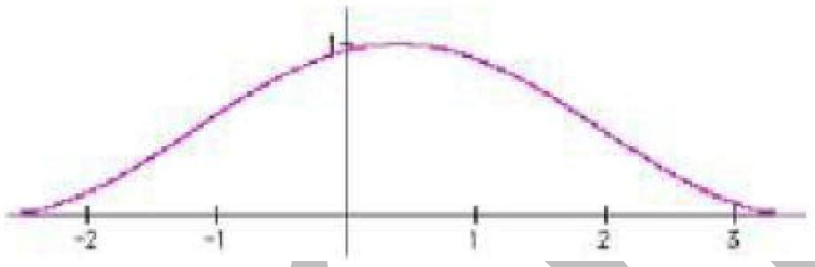


Figure 6.3 Fuzzy number

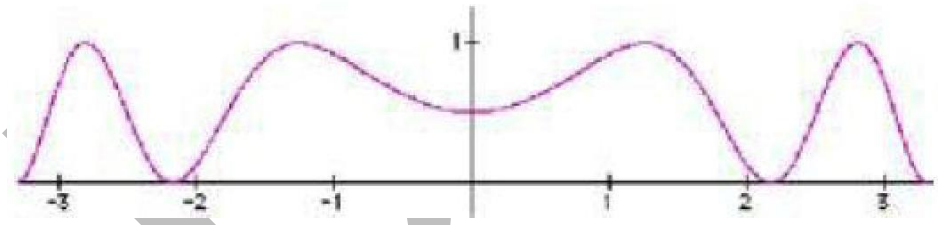


Figure 6.4 Non-fuzzy number

Note : Fuzzy number is always a fuzzy set, but a fuzzy set is not always a fuzzy number.

An example of fuzzy number is 'about 1' that is defined by $\mu(x) = \exp(-j(x-1)^2)$

It is also a quasi fuzzy number because $\lim_{x \to \pm\infty} \mu_A(x) = 0$.

Definition 6.1.10 (fuzzy interval)***A***fuzzy interval is a fuzzy set with the same restrictions as in definition 6.1.9, but thecore does not need to be a one point only.

Fuzzy intervals are direct generalizations of crisp intervals **[a, b]** **R.**

Definition 6.1.11 (*LR-representation of fuzzy numbers*) Any fuzzy number can be described by

$$\mu_A(x) = \begin{cases} L((a - x) / \alpha & , \ x \ \chi \ [a - \alpha, a] \\ 1 & , \ x \ \chi \ [a, b] \\ R((x - b) / \ ) & , \ x \ \chi \ [b, b + \ ] \\ 0 & , \text{otherwise} \end{cases}$$

Where $[a, b]$ is the core of ***A*** , and L:[0,1] → [0,1], R :[0,1] →[0,1] are shape functions (called brieflys-functions) that are continuous and nonincreasing such that $L(0) = R(0) = 1, L(1) = R(1) = 0$, where ***L*** stands for lefthand side and ***R*** stands for right-hand side of membership function [Zimmermann, 1993].

Definition 6.1.12 (*LR-representation of quasi fuzzy numbers*) Any quasi fuzzy number can be described by

$$\mu_A(x) = \begin{cases} L((a - x) / \alpha & , \ x \ \chi \ a \\ 1 & , \ x\chi \ [a, b] \\ R((x - b) / \ ) & , \ x \ \chi \ b \end{cases}$$

Where $[a, b]$ is the core of ***A*** , and L:[0,□) „_ [0,1],R:[0,□) → [0,1]

are shape functions that are continuous and non-increasing such that L(0) = R(0) = 1

and they approach zero :lim$L$(x) = 0 , lim R(x) = 0.

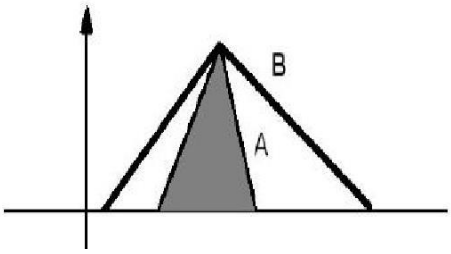
For example, f(x) = e-^{x}f, (x) = e-^{x_2} and f(x) = max (0.1 - x) are such shape functions. In the following the classical set theoretic operations are extended to fuzzy sets.

Definition 6.1.13 (*set theoretic operations*)

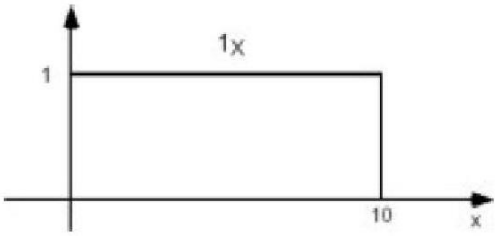
$\mu\emptyset \div 0$		(empty set)
$\mu X \div 1$		(basic set, universe)
$A = B\varepsilon\mu_A(x) = \mu_B(x)$	$6x\chi\ X$	(identity)
$A\chi B\varepsilon\mu_A(x) = \mu_B(x)$	$6x\chi\ X$	(subsethood)
$6x\chi\ X: \mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$		(union)
$6x\chi\ X: \mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$		(intersection)
$6x\chi\ X: \mu_A\text{-}(x) = 1 - \mu_A(x)$		(complement)

Union could also be represented by $A\cup A = \{(x, \max(\mu_A(x), \mu_B(x)))x\chi X\}$

Same notation could also be used with intersection and complement.



A is a subset of



The graph of the universal fuzzy subset in $X = [0, 10]$.

Figure 6.5 Fuzzy sets

Theorem 2.1.1 The following properties of set theory are valid

$=$

$A = A$ (involution)

$A \; B = B \; A$

$A \vee B = B \vee A$ (commutativity)

$(A \; B) \; C = A \; (B \; C) \; (A \vee B) \vee C = A \vee (B \vee)$ (associativity)

$A \vee (B \; C) = (A \vee B) \; (A \vee)$

$A \; (B \vee C) = (A \; B) \vee (A \; C)$ (distributivity)

$A \; A = A$

$A \vee A = A$ (idempotence)

$A \quad (A \vee B) = A$

$A \vee (A \; B) = A$ (absorption)

$(A \; B) = A \cap B$

$(A \vee B) = A \; B$ (De Morgan's laws)

Proof : Above properties can be proved by simple direct calculations. For example,

$=$

$A = 1 - (1 - A) = A$.

Fuzzy Relations

Definition 2.1.17 (*fuzzy relation*) Fuzzy relation is characterized by a function

$\mu_R : X_1 \times \ldots \times X_m \rightarrow [0,1]$ where X_i are the universes of discourse and $X_1 \times \ldots \times X_m$

is the product space. If we have two finite universes, the fuzzy relation can be presented as a matrix (fuzzy matrix) whose elements are the intensities of the relation and ***R*** has the membership function

μ_R (u,ϖ) , where u χ X_1 , ϖχ X_2 . Two fuzzy relations are combined by a so called sup-*- or max-min composition, which will be given in the definition 2.1.19.

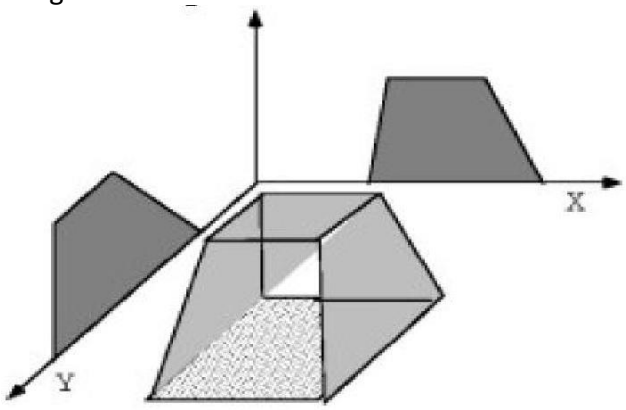


Figure 2.7 Shadows of a fuzzy relation

Note: Fuzzy relations are fuzzy sets, and so the operations of fuzzy sets (union, intersection, etc.) can be applied to them.

Example 2.1.2 Let the fuzzy relation***R***= "approximately equal" correspond to the equality of two numbers.The intensity of cell ***R***(*u*,ϖ) of the following matrix can be interpreted as the degree of membership of the ordered pair in ***R***. The numbers to be compared are {1,2,3,4} and {3,4 ,5,6}.

u \	1	2	3	4
3	.6	.8	1	.8
4	.4	.6	.8	1
5	.2	.4	.6	.8
6	.1	.2	.4	.6

The matrix shows, that the pair (4, 4) is approximately equal with intensity 1 and the pair (1, 6) is approximately equal with intensity 0.1.

Definition 2.1.18 (*Cartesian product*) Let*Ai x X*$_I$be fuzzy sets. Then the Cartesian product isdefined

$$\mu_{A1} \text{ x } ... \text{ x } A_n (x_1 ,..., x_n) = \min\{\mu_A (x_i)\}$$

Note: min can be replaced by a more general t-norm.

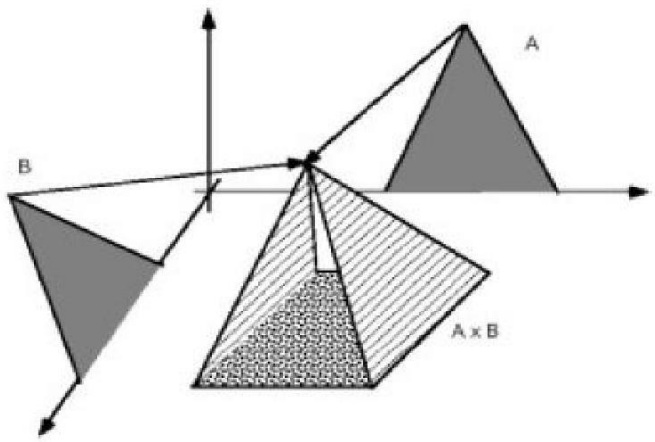


Figure 2.8 Cartesian product of two fuzzy sets is a fuzzy relation in *X*x*Y*

Definition 2.1.19 (*sup-*-composition, max-min composition*) The composition of two relations***R o S***is defined as a membership function in ***XxY***

$$\mu_{ROS}(x) = \sup_{\varpi} T(\mu_R(x,\varpi),\mu_S(\varpi,y))$$

where ***R*** is relation in ***XxV*** and ***S*** is relation in **V x Y, x X, y Y** and ***T*** is a t-norm sup is

replaced by the sum. If **S** is just a fuzzy set (not a relation) in ***V*** , then (2.15) becomes

$$\mu_{ROS}(x) = \sup_{\varpi} T(\mu_R(x,\varpi),\mu_S(\varpi))$$

Example 2.1.3 Let***X*= {**1,2,3,4},fuzzy set***A*= small = {(1,1), (2,0.6), (3,0.2), (4.0)}**

and fuzzy relation *R* = "approximately equal".

$$R = \begin{bmatrix} 1 & .5 & 0 & 0 \\ .5 & 1 & .5 & 0 \\ 0 & .5 & 1 & .5 \\ 0 & 0 & .5 & 1 \end{bmatrix}$$

$$A \circ B = \max_x \min \{\mu_A(x), \mu_R(x, y)\}$$

$$= \{(1,1), (2,.6), (3,.5), (4,.2)\}$$

The interpretation of example: x is small. If x and y are approximately equal, y is more or less small.

Exercise:

Q1.What is mean by Fuzzy Logic.
Q2.Explain Fuzzy Set.
Q3. Explain Fuzzy Relation.
Q4. Write a definition of Fuzzy Number.
Q5. Write a definition of LR-representation of fuzzy numbers.

Chapter 12

The extension principle, fuzzy measures, Membership function's

The Extension Principle

The extension principle is said to be one of the most important tools in fuzzy logic. It gives means to generalize non-fuzzy concepts, e.g., mathematical operations, to fuzzy sets. Any fuzzifying generalization must be consistent with the crisp cases.

Definition 2.1.20 (extension principle) Let $A_1,..., A_n$ be fuzzy sets, defined on $X_1 ... X\eta$ and let f be a function $f : X_i x ... x X_n V$ The extension of f operating on $A_1 ,..., A_n$ gives a membership function (fuzzy set F)

$$\mu_F(\varpi) = \sup_{u_1 \ldots u_n y\, f^{-1}(\varpi)} \min(\mu_A(u_i), \ldots, \mu_{An}(\mu_A(u_n)))$$

when the inverse of f exists. Otherwise define $\mu_F(\varpi) = 0$. Function f is called inducing mapping.

If the domain is either discrete or compact, sup-min can be replaced by max-min. On continuous domains sup-operation and the operation that satisfies criterion

$$S_W(x, y) = \begin{cases} x, & y = 0 \\ y, & x = 0 \\ 1, & \text{otherwise} \end{cases}$$

should be used [Driankov et. al., 1993].

Fuzzy Rules

Fuzzy logic was originally meant to be a technique for modeling the human thinking and reasoning, which is done by fuzzy rules. This idea has been replaced by the thought that the fuzzy rules form an interface between humans and computers [Brown & Harris, 1994]. Humans explain their actions and knowledge using linguistic rules and fuzzy logic is used to represent this knowledge on computers. There are three principal ways to obtain these rules :

1. human experts provide rules
2. data driven: rules are formed by training methods
3. combination of 1. and 2.

The first way is the ideal case for fuzzy systems. Although the rules are not precise, they contain important information about the system. In practice human experts may not provide a sufficient number of rules and especially in the case of complex systems the amount of knowledge may be very small or even non-existent. Thus the second way must be used instead of the first one (provided the data is available). The third way is suited for the cases when some knowledge exists and sufficient amount of data for training is available. In this case fuzzy rules got from experts roughly

approximate the behavior of the system, and by applying training this approximation is made more precise. Rules provided by the expert's form an initial point for the training and thus exclude the necessity of random initialization and diminish the risk of getting stuck in a local minimum (provided the expert knowledge is good enough).

It has been shown in [Mouzouris, 1996] that linguistic information is important in the absence of sufficient numerical data but it becomes less important as more data become available.

Fuzzy rules define the connection between input and output fuzzy linguistic variables and they can be seen to act as associative memories. Resembling inputs are converted to resembling outputs. Rules have a structure of the form :

IF (antecedent) **THEN** (consequent)

In more detail, the structure is

IF (1 x is Ai 1 AND .... AND d x is Aid THEN (y is i B)

where Ajii and Bi are fuzzy sets (they define complete fuzzy partitions) in % U R % V "R, respectively. Linguistic variable x is a vector of dimension d in d UU CC ...1 and linguistic variable V y# ". Vector x is an input to the fuzzy system and y is an output of the fuzzy system. Note that Bi can also be a singleton (consequence part becomes: ...THEN (y is i z)). Further, if fuzzy system is used as a classifier, the consequence part becomes: ...THEN class is c.

A fuzzy rule base consists of a collection of rules $\{R^1, R^2, ..., R^M\}$, where each rule i R can be considered to be of the form (2.34). This does not cause a loss of generality, since multi-input-multioutput (MIMO) fuzzy logic system can always be decomposed into a group of multi-input-singleoutput (MISO) fuzzy logic systems. Furthermore (2.34) includes the following types of rules as a special case [Wang, 1994] :

- **IF** (x_1 is A^i_1 **AND ... AND** x_i is A^i_k) **THEN** (y is $\boldsymbol{B}^i$) where $k>d$
- **IF** (x_1 is A^i_1 **AND ... AND** x_i is A^i_k) **OR** (x_{k+1} is $\boldsymbol{A}^i_{k+1}$**AND... AND**x_d is $\boldsymbol{A}^i_d$)**) THEN** (y is $\boldsymbol{B}^i$)
- (y is $\boldsymbol{B}^i$)
- (y is $\boldsymbol{B}^i$) **UNLESS** (x_i is A^i_1) **AND...AND**x_d is $\boldsymbol{A}^i_d$)
- non-fuzzy rules

In control systems the production rules are of the form

IF <process state> **THEN** <control action>

where the <process state> part contains a description of the process output at the kth sampling instant. Usually this description contains values of error and change-of-error. The <control action> part describes the control output (change-in-control) which should be produced given the particular <process state>. For example, a fuzzified PI-controller is of the type (2.35). Important properties for a set of rules are completeness, consistency and continuity. These are defined in the following.

Definition 6.1.27 (*completeness*) A rule base is said to be complete if any combination of input values results in anappropriate output value:

$$\mathbf{6\ x\ \chi X :hgt(out(x)) >0}$$

Definition 6.1.28 (inconsistency) A rule base is inconsistent if there are two rules with the same rule antecedent butdifferent rule consequences.

This means that two rules that have the same antecedent map to two non-overlapping fuzzy output sets. When the output sets are non-overlapping, then there is something wrong with the output variables or the rules are inconsistent or discontinuous.

Definition 6.1.29 (continuity) A rule base is continuous if the neighboring rules do not have fuzzy output sets that have

empty intersection.

Exercise:

Q1.What is mean by The Extension Principle.
Q2.What are the Fuzzy Rule.
Q3. Write a definition of completeness.
Q4. Write a Definition of inconsistency.

Chapter 13

Fuzzification and defuzzyfication methods, Fuzzy controllers

Fuzzifier and Defuzzifier

The *fuzzifier* maps a crisp point x into a fuzzy set

$$\mu_{A'}(x) = \begin{cases} 1, & \text{if } x = x' \\ 0, & \text{otherwise} \end{cases}$$

Where ***X'*** is the input. Fuzzifier of the form (2.43) is called a singleton fuzzifier. If the input contains noise it can be modeled by using fuzzy number. Such fuzzifier could be called a nonsingleton fuzzifier. Because of simplicity, singleton fuzzifier is preferred to nonsingleton fuzzifier.

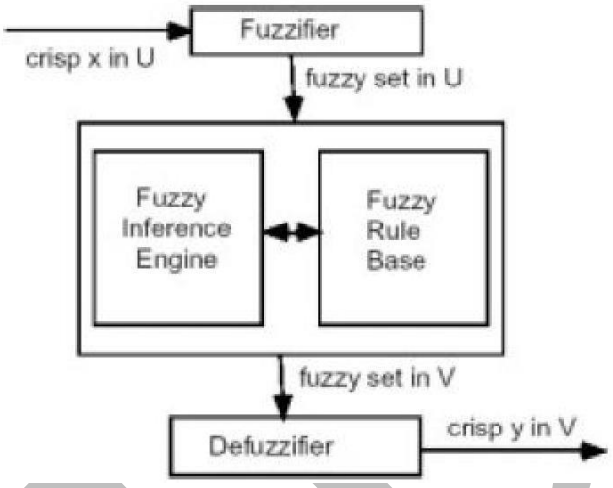


Figure 6.15 Fuzy logic controller

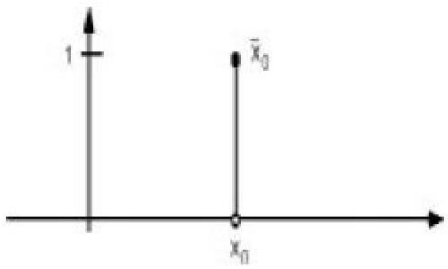


Figure 2.16 Fuzzy singleton as fuzzifier

The *defuzzifier* maps fuzzy sets to a crisp point. Several defuzzification methods have been suggested. The following five are the most common :

- **Center of Gravity** (CoG) : In the case of 1-dimensional fuzzy sets it is often called the Center of Area (CoA)method. Some authors (for example, [Driankov et. al., 1993]) regard CoG and CoA as a same method, when other (for example, [Jager, 1995]) give them different forms. If CoA is calculated by dividing the area of combination of output membership functions by two and then taking from the left so much that we get an area equal to the right one, then it is clearly a distinct method. CoG determines center of gravity of the mass, which is formed as a combination of the clipped or scaled output fuzzy membership functions. The intersection part of these membership functions can be taken once or twice into calculation. Driankov [1993] separates the Center of Gravity methods such that, if the

intersection is calculated once the method is Center of Area and if it is calculated twice the method is called Center of Sums (CoS). In Fig. 2.17 defuzzified value obtained by CoS is slightly smaller than obtained by the CoA - method.

- **Height method** (HM) : Can be considered as a special case of CoG, whose output membership functions aresingletons. If symmetric output sets are used in CoG, they have the same centroid no matter how wide the set is and CoG reduces to HM. HM calculates a normalized weighted sum of the clipped or scaled singletons. HM is sometimes called Center average defuzzifier or fuzzy-mean defuzzifier.
- **Middle of Maxima** (MoM) : Calculates the center of maximum in clipped membership function. The rest of thedistribution is considered unimportant.
- **First of Maxima** (FoM): As MoM, but takes the leftmost value instead of center value.

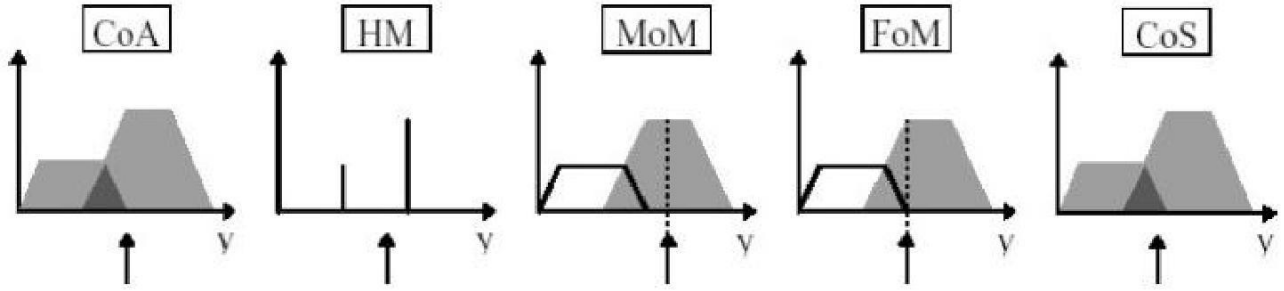


Figure 6.17 Defuzzification methods

Defuzzification methods can be compared by some criteria, which might be the continuity of the output and the computational complexity. HM, CoA and CoS produce continuous outputs. The simplest and quickest method of these is the HM method and for large problems it is the best choice. The fuzzy systems using it have a close relation to some well-known interpolation methods (we will return to this relation later).

The maximum methods (MoM, FoM) have been widely used. The underlying idea of MoM (with max-min inference) can be explained as follows. Each input variable is divided into a number of intervals, which means that the whole input space is divided into a large number of d-dimensional boxes. If a new input point is given, the corresponding value for y is determined by finding which box the point falls in and then returning the average value of the corresponding y-interval associated with that input box. Because of the piecewise constant output, MoM is inefficient for approximating nonlinear continuous functions. Kosko has shown [Kosko, 1997] that if there are many rules that fire simultaneously, the maximum function tends to approach a constant function. This may cause problems especially in control.

The property of CoS is that the shape of final membership function used as a basis for defuzzification will resemble more and more normal density function when the number of functions used in summation grows. Systems of this type that sum up the output membership functions to form final output set are called *additive fuzzy systems*.

Fuzzy logic is widely used in machine control. The term “fuzzy” refers to the fact that the logic involved can deal with concepts that cannot be expressed as the “true” or “false” but rather as “partially true”. Although alternative approaches such as genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and

so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

The most common shape of membership functions is triangular, although trapezoidal and bell curves are also used, but the shape is generally less important than the number of curves and their placement. From three to seven

curves are generally appropriate to cover the required range of an input value, or the “universe of discourse” in fuzzy jargon.

As discussed earlier, the processing stage is based on a collection of logic rules in the form of IF THEN statements, where the IF part is called the “antecedent” and the THEN part is called the “consequent”. Typical fuzzy control systems have dozens of rules.

Consider a rule for a thermostat :

IF (temperature is “cold”) THEN turn (heater is “high”)

This rule uses the truth value of the “temperature” input, which is some truth value of “cold”, to generate a result in the fuzzy set for the “heater” output, which is some value of “high”. This result is used with the results of other rules to finally generate the crisp composite output. Obviously, the greater the truth value of “cold”, the higher the truth value of “high”, though this does not necessarily mean that the output itself will be set to “high” since this is only one rule among many. In some cases, the membership functions can be modified by “hedges” that are equivalent to adverbs. Common hedges include “about”, “near”, “close to”, “approximately”, “very”, “slightly”, “too”, “extremely”, and “somewhat”. These operations may have precise definitions, though the definitions can vary considerably between different implementations. “Very”, for one example, squares membership functions; since the membership values are always less than 1, this narrows the membership function. “Extremely” cubes the values to give greater narrowing, while “somewhat” broadens the function by taking the square root.

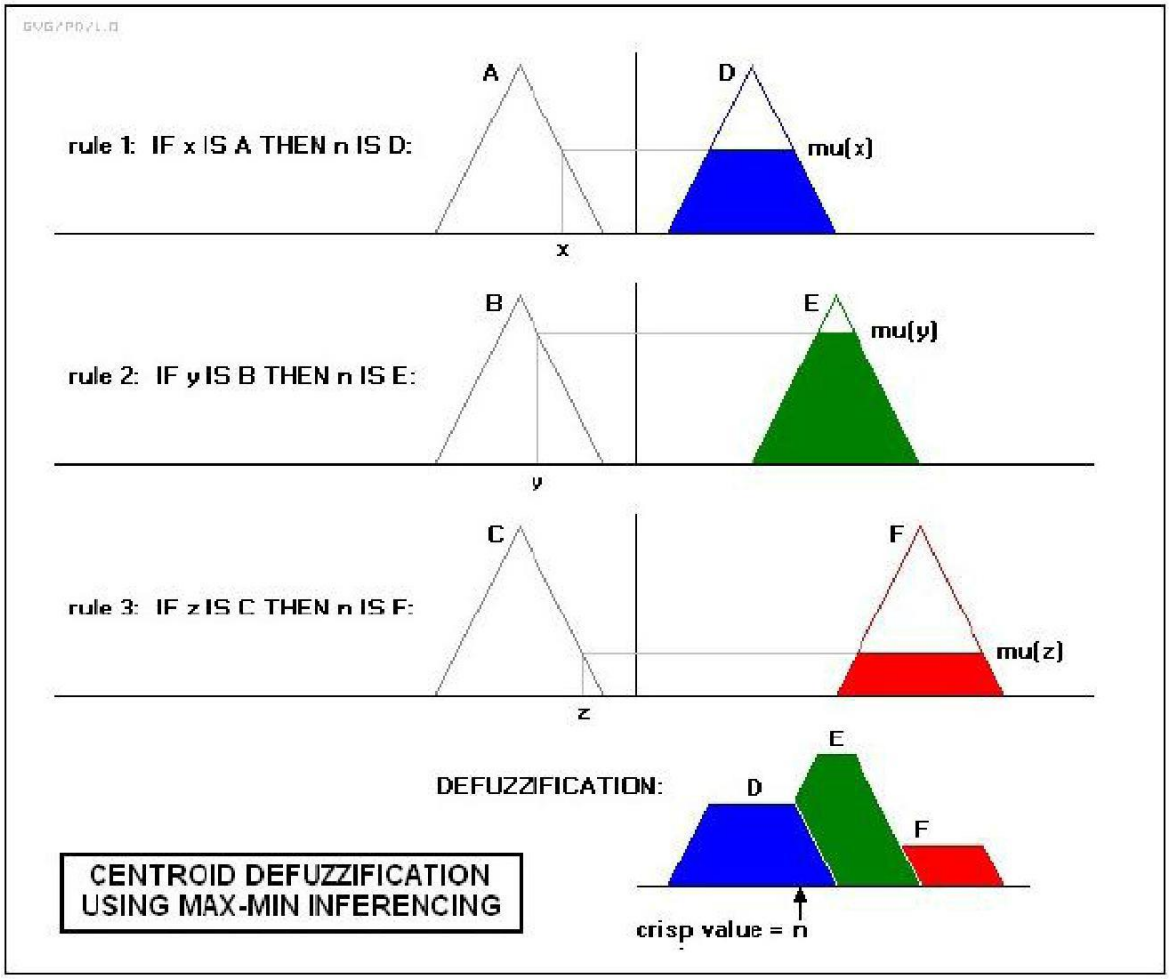
In practice, the fuzzy rule sets usually have several antecedents that are combined using fuzzy operators, such as AND, OR, and NOT, though again the definitions tend to vary: AND, in one popular definition, simply uses the minimum weight of all the antecedents, while OR uses the maximum value. There is also a NOT operator that subtracts a membership function from 1 to give the “complementary” function.

There are several ways to define the result of a rule, but one of the most common and simplest is the “max-min” inference method, in which the output membership function is given the truth value generated by the premise.

Rules can be solved in parallel in hardware, or sequentially in software. The results of all the rules that have fired are “defuzzified” to a crisp value by one of several methods. There are dozens, in theory, each with various advantages or drawbacks.

The “centroid” method is very popular, in which the “center of mass” of the result provides the crisp value. Another approach is the “height” method, which takes the value of the biggest contributor. The centroid method favors the rule with the output of greatest area, while the height method obviously favors the rule with the greatest output value.

The diagram below demonstrates max-min inferencing and centroid defuzzification for a system with input variables “x”, “y”, and “z” and an output variable “n”. Note that “mu” is standard fuzzylogic nomenclature for “truth value”:

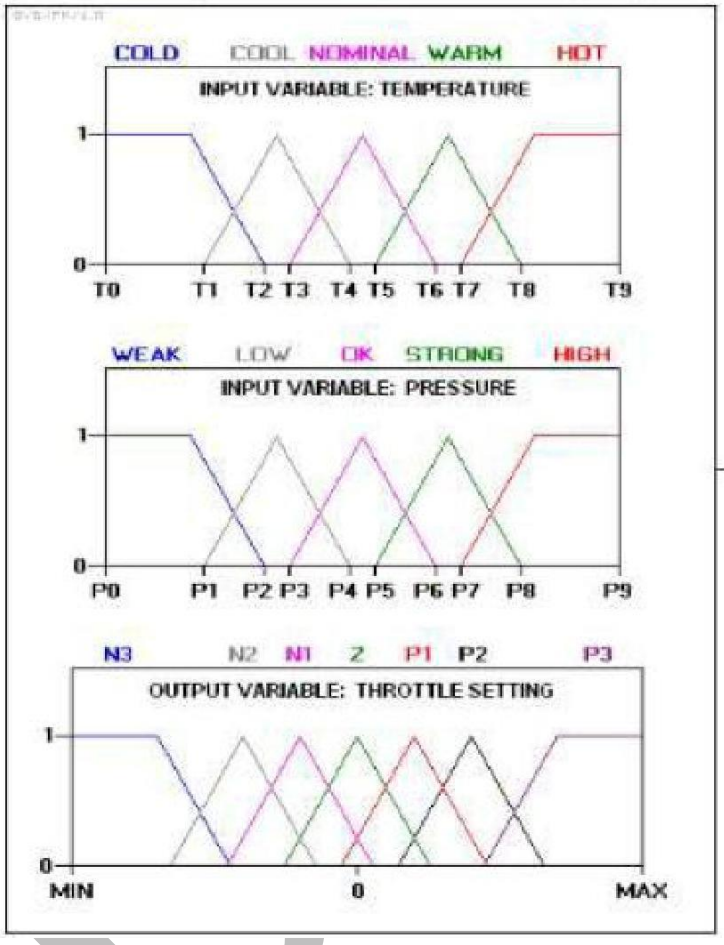


Notice how each rule provides a result as a truth value of a particular membership function for the output variable. In centroid defuzzification the values are OR'd, that is, the maximum value is used and values are not added, and the results are then combined using a centroid calculation. Fuzzy control system design is based on empirical methods, basically a methodical approach to trial-and-error. The general process is as follows :

- Document the system's operational specifications and inputs and outputs.
- Document the fuzzy sets for the inputs.
- Document the rule set.
- Determine the defuzzification method.
- Run through test suite to validate system, adjust details as required.
- Complete document and release to production.

As a general example, consider the design of a fuzzy controller for a steam turbine. The block diagram of this control system appears as follows :

The input and output variables map into the following fuzzy set :



where:

N3 : Large negative.

N2 : Medium negative.

N1 : Small negative.

Z : Zero.

P1 : Small positive.

P2 : Medium positive.

P3 : Large positive.

The rule set includes such rules as:

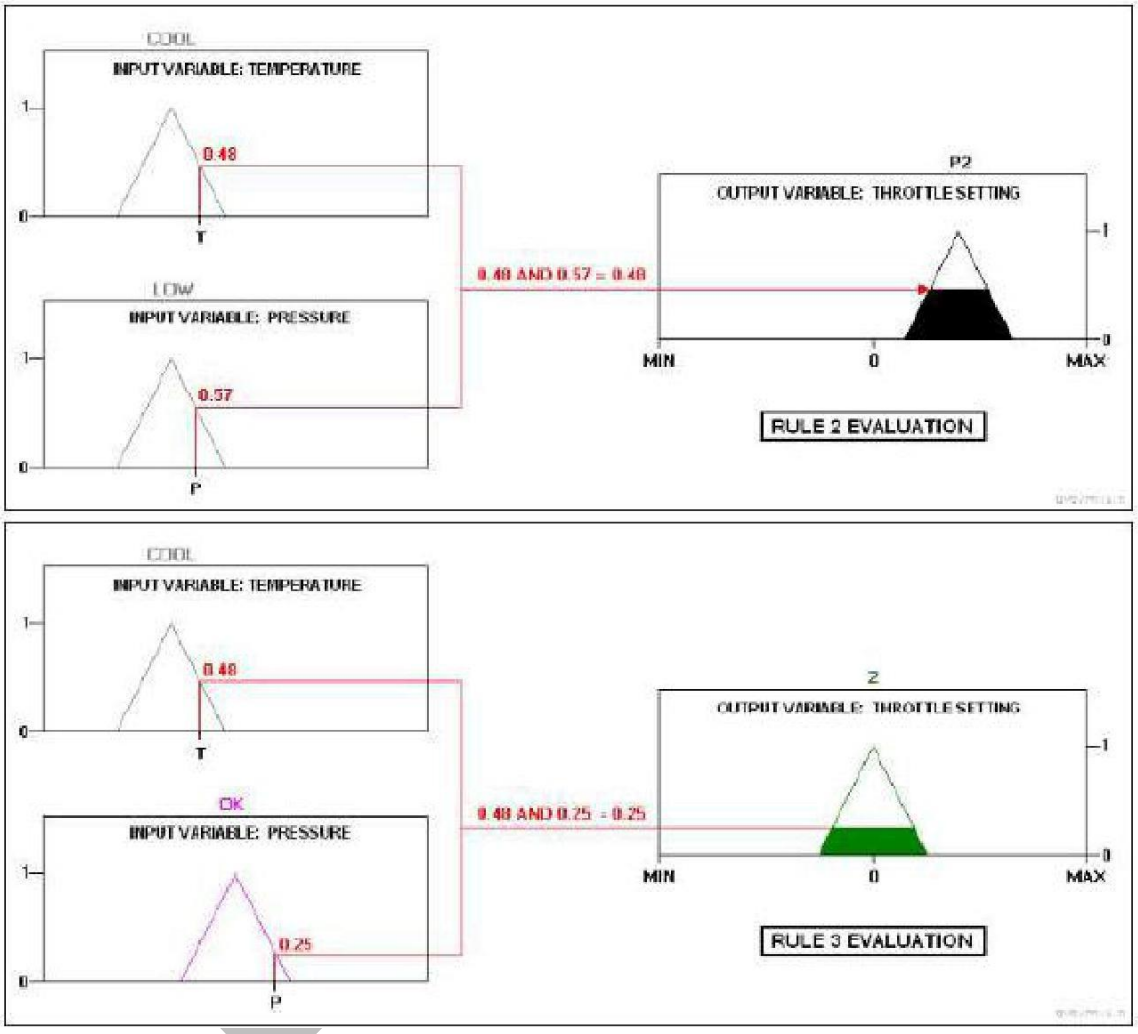
rule 1 : IF temperature IS cool AND pressure IS weak, THEN throttle is P3.

rule 2 : IF temperature IS cool AND pressure IS low, THEN throttle is P2.

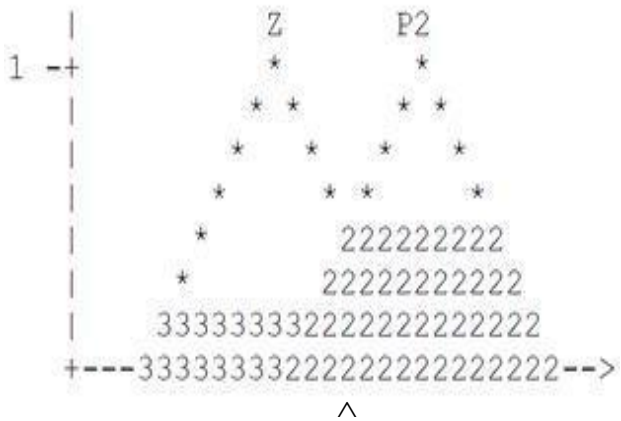
rule 3 : IF temperature IS cool AND pressure IS ok, THEN throttle is Z.

rule 4 : IF temperature IS cool AND pressure IS strong, THEN throttle is N2.

In practice, the controller accepts the inputs and maps them into their membership functions and truth values. These mappings are then fed into the rules. If the rule specifies an AND relationship between the mappings of the two input variables, as the examples above do, the minimum of the two is used as the combined truth value; if an OR is specified, the maximum is used. The appropriate output state is selected and assigned a membership value at the truth level of the premise. The truth values are then defuzzified. For an example, assume the temperature is in the "cool" state, and the pressure is in the "low" and "ok" states. The pressure values ensure that only rules 2 and 3 fire:



The two outputs are then defuzzified through centroid defuzzication :

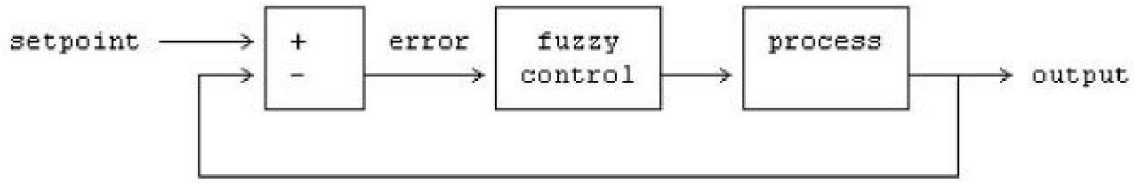


+150

The output value will adjust the throttle and then the control cycle will begin again to generate the next value .

Building a fuzzy controller[edit]

Consider implementing with a microcontroller chip a simple feedback controller :



A fuzzy set is defined for the input error variable "e", and the derived change in error, "delta", as well as the "output", as follows:

LP : large positive
SP : small positive
ZE : zero
SN : small negative
LN : large negative

If the error ranges from -1 to +1, with the analog-to-digital converter used having a resolution of 0.25, then the input variable's fuzzy set (which, in this case, also applies to the output variable) can be described very simply as a table, with the error / delta / output values in the top row and the truth values for each membership function arranged in rows beneath :

	-1	-0.75	-0.5	-0.25	0	0.25	0.5	0.75	1
mu (LP)	0	0	0	0	0	0	0.3	0.7	1
mu (SP)	0	0	0	0	0.3	0.7	1	0.7	0.3
mu (ZE)	0	0	0.3	0.7	1	0.7	0.3	0	0
mu (SN)	0.3	0.7	1	0.7	0.3	0	0	0	0
mu (LN)	1	0.7	0.3	0	0	0	0	0	0

or, in graphical form (where each "X" has a value of 0.1):

```
          LN          SN          ZE          SP          LP
          +-----------------------------------------------+
-1.0      | XXXXXXXXXX  XXX         :           :           :           |
-0.75     | XXXXXXX     XXXXXXX     :           :           :           |
-0.5      | XXX         XXXXXXXXXX  XXX         :           :           |
-0.25     | :           XXXXXXX     XXXXXXX     :           :           |
0.0       | :           XXX         XXXXXXXXXX  XXX         :           |
0.25      | :           :           XXXXXXX     XXXXXXX     :           |
0.5       | :           :           XXX         XXXXXXXXXX  XXX         |
0.75      | :           :           :           XXXXXXX     XXXXXXX     |
1.0       | :           :           :           XXX         XXXXXXXXXX  |
```

Suppose this fuzzy system has the following rule base :

rule 1 : IF e	= ZE	AND	delta = ZE	THEN output = ZE
rule 2 : IF e	= ZE	AND	delta = SP	THEN output = SN
rule 3 : IF e	= SN	AND	delta = SN	THEN output = LP
rule 4 : IF e	= LP	OR	delta = LP	THEN output = LN

These rules are typical for control applications in that the antecedents consist of the logical combination of the error and error-delta signals, while the consequent is a control command output. The rule outputs can be defuzzified using a discrete centroid computation :

SUM (I = 1 TO 4 OF (mu(I) * output(I))) / SUM(I = 1 TO 4 OF mu(I))

Now, suppose that at a given time we have:

e = 0.25

delta = 0.5

Then this gives :

	e	delta
mu(LP)	0	0.3
mu(SP)	0.7	1
mu(ZE)	0.7	0.3
mu(SN)	0	0
mu(LN)	0	0

Plugging this into rule 1 gives :

rule 1 : IF e = ZE AND delta = ZE THEN output = ZE
mu (1) = MIN (0.7, 0.3) = 0.3
output(1) = 0

-- **where :**

- mu(1) : Truth value of the result membership function for rule 1. In terms of a centroid calculation, this is the “mass” of this result for this discrete case.
- output(1) : Value (for rule 1) where the result membership function (ZE) is maximum over the output variable fuzzy set range. That is, in terms of a centroid calculation, the location of the “center of mass” for this individual result. This value is independent of the value of “mu”. It simply identifies the location of ZE along the output range.

The other rules give :

rule 2 : IF e = ZE AND delta = SP THEN output = SN
mu (2) = MIN (0.7, 1) = 0.7
output (2) = -0.5
rule 3 : IF e = SN AND delta = SN THEN output = LP
mu (3) = MIN (0.0, 0.0) = 0
output (3) = 1

rule 4 : IF e = LP OR delta = LP THEN output = LN
mu (4) = MAX (0.0, 0.3) = 0.3
output (4) = -1

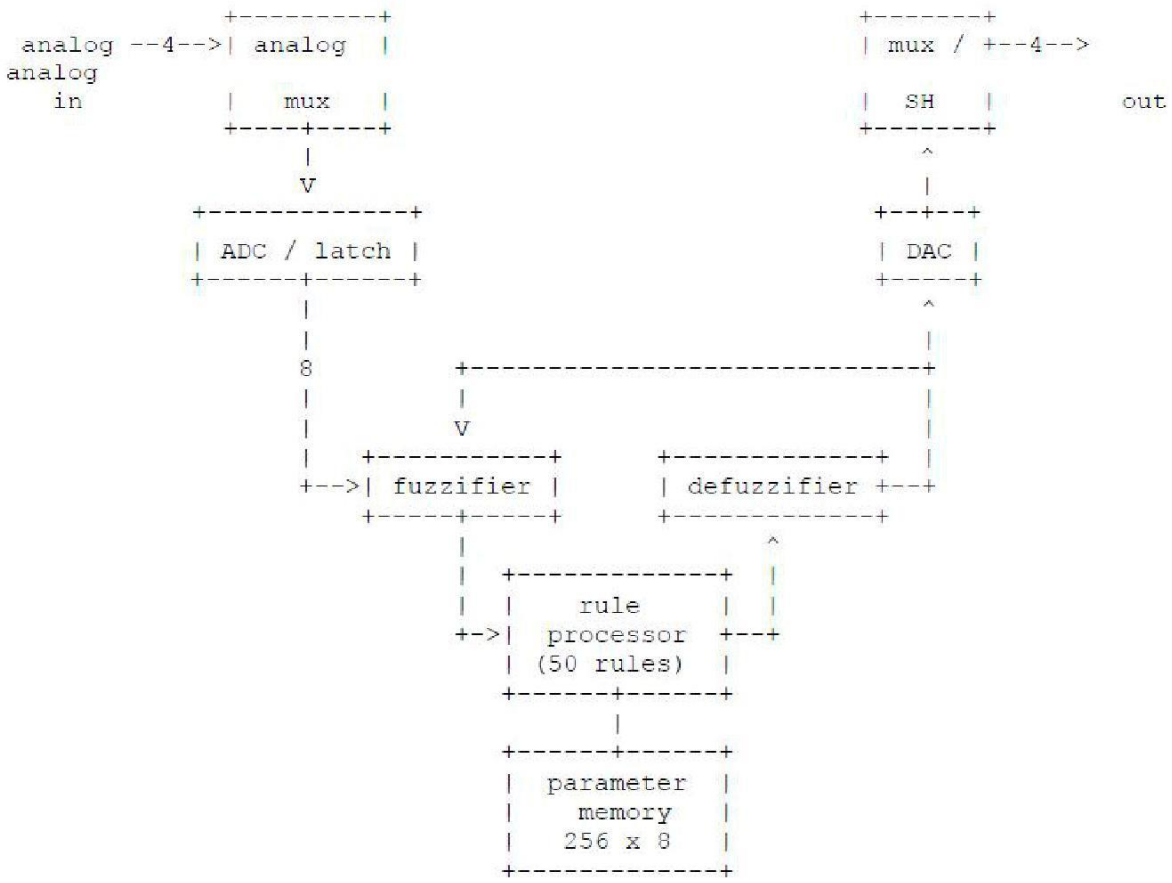
The centroid computation yields :

-for the final control output. Simple. Of course the hard part is figuring out what rules actually work correctly in practice.

If you have problems figuring out the centroid equation, remember that a centroid is defined by summing all the moments (location times mass) around the center of gravity and equating the sum to zero. So if is the center of gravity, is the location of each mass, and is each mass, this gives :

In our example, the values of mu correspond to the masses, and the values of X to location of the masses (mu, however, only 'corresponds to the masses' if the initial 'mass' of the output functions are all the same/equivalent. If they are not the same, i.e. some are narrow triangles, while others maybe wide trapizoids or shouldered triangles, then the mass or area of the output function must be known or calculated. It is this mass that is then scaled by mu and multiplied by its location X_i).

This system can be implemented on a standard microprocessor, but dedicated fuzzy chips are now available. For example, Adaptive Logic INC of San Jose, California, sells a "fuzzy chip", the AL220, that can accept four analog inputs and generate four analog outputs. A block diagram of the chip is shown below :



ADC : analog-to-digital converter

DAC : digital-to-analog converter

SH

:
sampl
e/hold

Antilock brakes

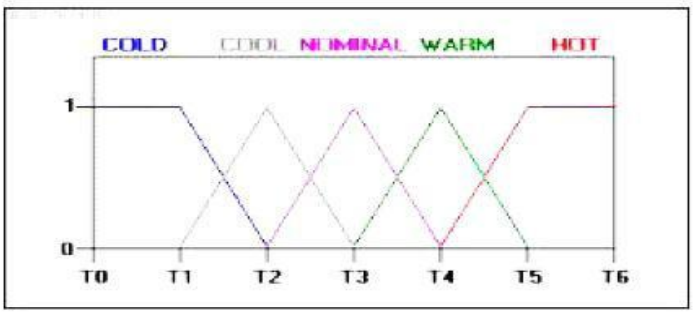
As a first example, consider an anti-lock braking system, directed by a microcontroller chip. The microcontroller has to make decisions based on brake temperature, speed, and other variables in the system.

The variable "temperature" in this system can be subdivided into a range of "states": "cold", "cool", "moderate", "warm", "hot", "very hot". The transition from one state to the next is hard to define.

An arbitrary static threshold might be set to divide "warm" from "hot". For example, at exactly 90 degrees, warm ends and hot begins. But this would result in a discontinuous change when the input value passed over that threshold. The transition wouldn't be smooth, as would be required in braking situations.

The way around this is to make the states *fuzzy*. That is, allow them to change gradually from one state to the next. In order to do this there must be a dynamic relationship established between different factors.

We start by defining the input temperature states using "membership functions":



With this scheme, the input variable's state no longer jumps abruptly from one state to the next. Instead, as the temperature changes, it loses value in one membership function while gaining value in the next. In other words, its ranking in the category of cold decreases as it becomes more highly ranked in the warmer category.

At any sampled timeframe, the "truth value" of the brake temperature will almost always be in some degree part of two membership functions: i.e.: '0.6 nominal and 0.4 warm', or '0.7 nominal and 0.3 cool', and so on.

The above example demonstrates a simple application, using the abstraction of values from multiple values. This only represents one kind of data, however, in this case, temperature.

Adding additional sophistication to this braking system, could be done by additional factors such as traction, speed, inertia, set up in dynamic functions, according to the designed fuzzy system.[9]

Logical interpretation of fuzzy control

In spite of the appearance there are several difficulties to give a rigorous logical interpretation of the IF-THEN rules. As an example, interpret a rule as IF (*temperature is "cold"*) THEN (*heater is "high"*) by the first order formula *Cold(x)→High(y)* and assume that r is an input such that *Cold(r)* is false. Then the formula *Cold(r)→High(t)* is true for any t and therefore any t gives a correct control given r. A rigorous logical justification offuzzy control is given in Hajek's book (see Chapter 7) where fuzzy control is represented as a theory of Hajek's basic logic.[2] Also in Gerla 2005 [10] another logical approach to fuzzy control is proposed based on fuzzy logic programming.Indeed, denote by f the fuzzy function arising of an IF-THEN systems of rules. Then we can translate this system into a fuzzy program P containing a series of rules whose head is "Good(x,y)". The interpretation of this predicate in the least fuzzy Herbrand model of P coincides with f. This gives further useful tools to fuzzy control.

Exercise:

Q1. Explain Fuzzifier and Defuzzifier.
Q2.Write a short note onFuzzification.
Q3. Explain the concept of defuzzyfication methods.
Q4.What is the process of Building a fuzzy controller.
Q5.Write a short note on centroid computation yields.
Q6. Write a short note on Antilock brakes.