Application of AI

Chapter_6

Expert Systems

What is an Expert System?

- An **expert system** is like a **smart computer program** that acts as a **virtual expert** in a specific field. It uses a collection of **knowledge** (facts and rules) to solve problems or make decisions, just like a human expert would.
- Think of it as a **robot doctor**, **robot mechanic**, or **robot financial advisor** that can give advice or solve problems in its area of expertise.

Key Components of an Expert System

1. **Knowledge Base**:
    1. This is like the **brain** of the expert system.
    2. It stores all the **facts** and **rules** about a specific topic.
    3. Example: A medical expert system might have rules like:
        1. *If the patient has a fever and a cough, then they might have the flu.*

Key Components of an Expert System

2. Inference Engine:

1. This is the **thinking part** of the system.
2. It uses the rules from the knowledge base to **reason** and come up with answers.
3. Example: If you tell the system, "I have a fever and a cough," it will use the rules to say, "You might have the flu."

Key Components of an Expert System

3. User Interface:

1. This is how **you interact** with the system.
2. It could be a chat window, a form, or even a voice assistant.
3. Example: You type your symptoms into a medical app, and it gives you advice.

Key Components of an Expert System

4. **Explanation Facility**:
    1. This explains **how the system reached its conclusion**.
    2. Example: The system might say, "I think you have the flu because you have a fever and a cough.”
5. **Knowledge Acquisition Facility**:
    1. This is how the system **learns new information**.
    2. Example: A doctor might add new rules to the system, like, "If the patient has a rash, they might have chickenpox."

How Does an Expert System Work?

- Let’s use a **real-world example** to explain this:
- **Example: Medical Diagnosis System**

1. **You Input Data**:
    1. You tell the system, "I have a fever, a cough, and a sore throat."
2. **Inference Engine Works**:
    1. The system checks its **knowledge base** for rules that match your symptoms.
    2. It finds a rule: *If the patient has a fever, cough, and sore throat, then they might have strep throat.*

How Does an Expert System Work?

3. System Gives Output:

1. The system says, "You might have strep throat. Please see a doctor for a test."

4. Explanation Facility:

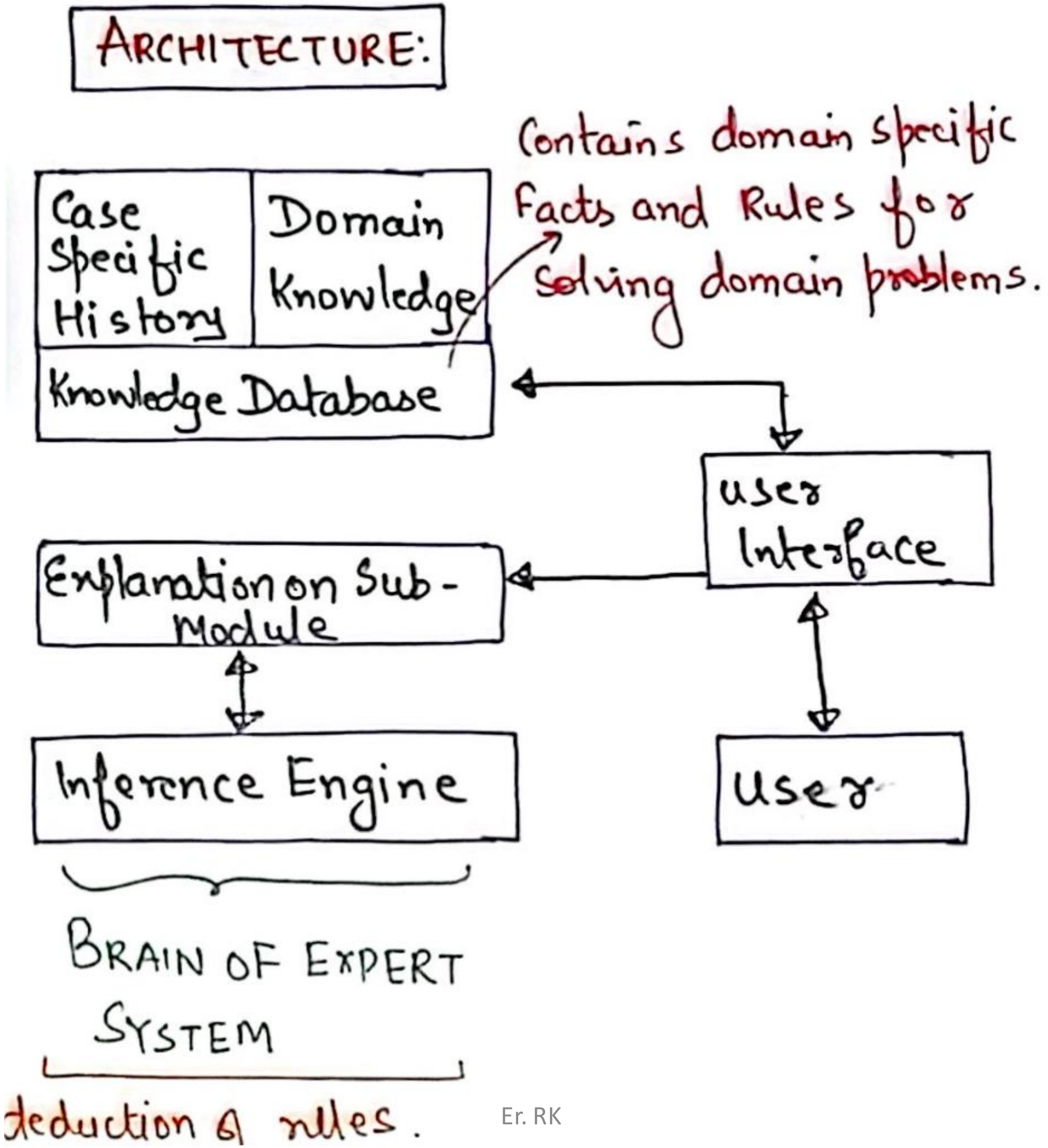
1. The system explains, "I think you have strep throat because you have a fever, cough, and sore throat."

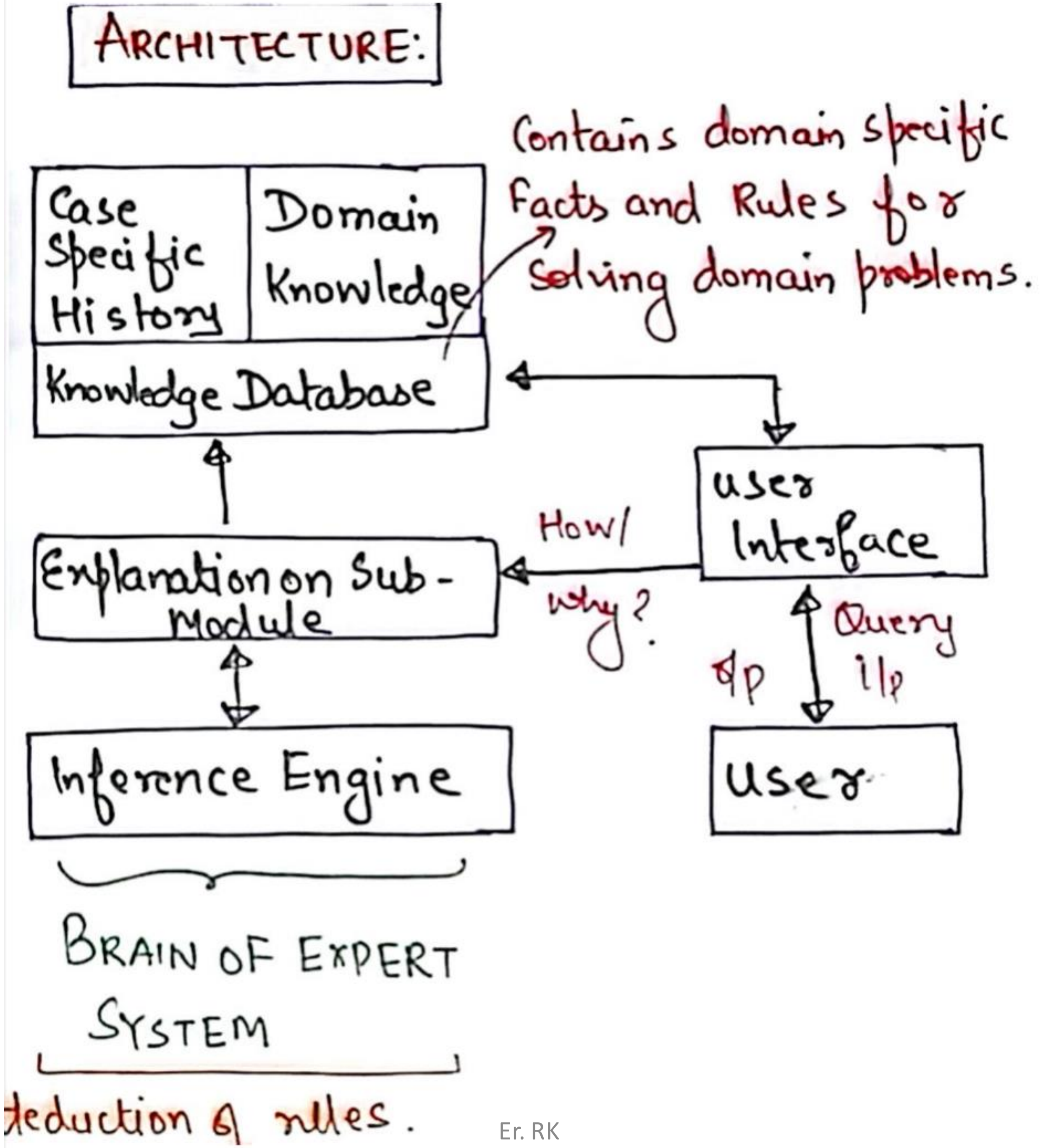
Real-Time Examples of Expert Systems

1. **MYCIN**:
   1. A medical expert system that diagnoses bacterial infections and recommends antibiotics.
   2. Example: If a patient has a fever and specific symptoms, MYCIN suggests the best antibiotic.
2. **Chatbots**:
   1. Many customer support chatbots are expert systems.
   2. Example: If you ask a chatbot, "How do I reset my password?" it uses rules to guide you step-by-step.
3. **Traffic Light Control Systems**:
   1. Some traffic lights use expert systems to manage traffic flow.
   2. Example: If there’s heavy traffic on one road, the system changes the lights to reduce congestion.

Characteristics:

- i) High Performance
- ii) Reliable
- iii) Highly Responsive
- iv) Understandable.


ARCHITECTURE:
Contains domain specific facts and Rules for solving domain problems.
Case Specific History
Domain Knowledge
Knowledge Database
user Interface
Explanation on Sub-Module
Inference Engine
User
BRAIN OF EXPERT SYSTEM
deduction of rules.


ARCHITECTURE:
Contains domain specific facts and Rules for solving domain problems.
Case Specific History
Domain Knowledge
Knowledge Database
User Interface
How/ why?
Query i/p
o/p
Explanation Sub-Module
Inference Engine
User
BRAIN OF EXPERT SYSTEM
deduction of rules.

Advantages of Expert System:

- i) Helps in Increasing the expert Knowledge availability.
- ii) Can be used for training future experts.
- iii) Helps in dealing with uncertainity
  - Cost effective.
- iv) Helps in giving Consistent answer
  - Based on Rules.
- v) Can Review all transactions.
  - while Human Can review Sample of transactions.

Disadvantages of Expert System:

- i) Lack of Common Sense, inflexible
  - modification is difficult.
- ii) Can be used in limited areas.
  - Domain is restricted
- iii) Can't be relied in Areas like medical diagonosis, treatment.
- iv) Can't be used where input is Sensory in nature.
- v) Can't respond creatively to unusual situation.

Applications of Expert System:

- i) Medical Diagnosis
- ii) Help Desk Managem
- iii) Loan analysis
- iv) Virus detection
- v) Stock Market Tradin
- vi) Warehouse optimization

Conventional Computer System	Expert System
① Control Sol^n Process = Data is used	Knowledge is used
② Data is Stored alongwith control program.	Knowledge is Stored Separately in encoded form.
③ How and explanation modules are not present.	Capable of explaining how a decision/ Conclusion is drawn.
④ Data Forms → Numeric, → Alphabetic, → Audio, → Video ...	Knowledge → Rules, → Networks, → Trees.
⑤ Metadata may or may not be present. (→ data about data.)	Meta Knowledge ✓

Development of Expert Systems

1. **Identify the Problem**:
    1. Decide what the system will do (e.g., diagnose diseases, recommend products).
2. **Gather Knowledge**:
    1. Talk to experts (e.g., doctors, engineers) and learn how they solve problems.
3. **Create Rules**:
    1. Write down the rules the system will use (e.g., *If A, then B*).
4. **Build the System**:
    1. Use programming tools to create the knowledge base and inference engine.
5. **Test the System**:
    1. Try it out with real users and see if it works correctly.
6. **Update and Improve**:
    1. Add new rules and fix any problems.

Development of Expert System: General Steps

- The process of ES development is iterative. It includes :-
- 1. Identify Problem Domain
  - The problem must be suitable for ane xpert sytem to solve it.
  - Find the experts in the task toamin for the ES project.
  - Establish cost-effectiveness of the system.
- 2. Design the system:
  - Identify the ES Technology
  - Know and establish the degree of integration with the other systems and database
  - Realize how the concepts can represents the domain knowledge best.

Development of Expert System: General Steps

- 3. Develop the Prototype

From the KB: The Knowledge engineer works to :-

  - acquire domain knowledge from the expert.
  - Represent it in the form of If-THEN-ELSE rules.

- 4. Test and Refine the Prototype
  - The Knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
  - End users test the prototypes of the ES.

Development of Expert System: General Steps

- 5. Develop and Complete the ES
  - Test and esusre the interaction of the ES with all elements of its environment, including end users, database and other informatin systems.
  - Document the ES project well
  - Train the user to use ES.
- 6. Maintain the System.
  - Keep the KB up to date by regular review and update.
  - Carter for new interface with other information system, as those system evolve.

Knowledge Engineering

- This is the process of **building the knowledge base** for an expert system. It involves:
- **Talking to Experts**: Learning from people who know a lot about the topic.
- **Writing Rules**: Turning what the experts say into rules the system can use.
- **Testing Rules**: Making sure the rules work correctly.

What is Case-Based Reasoning (CBR)?

- Case-Based Reasoning (CBR) is a problem-solving method where a system **solves new problems** by looking at **past problems** (called **cases**) and using their solutions as a guide. It’s like learning from experience!
- Think of it as a **smart assistant** that remembers how it solved similar problems in the past and uses that knowledge to help you with new problems.

How Does CBR Work?

- CBR works in **4 simple steps**:

1. **Retrieve**: Find past cases that are similar to the new problem.
2. **Reuse**: Use the solution from the past case to solve the new problem.
3. **Revise**: Check if the solution works for the new problem. If not, adjust it.
4. **Retain**: Save the new case (problem and solution) for future use.

Example: A Doctor Using CBR

- Imagine a **doctor** who uses CBR to diagnose patients.

1. **Retrieve**:
    1. A new patient comes in with symptoms: fever, cough, and sore throat.
    2. The doctor remembers a past case where a patient had similar symptoms and was diagnosed with **strep throat**.
2. **Reuse**:
    1. The doctor thinks, "Since the past patient had strep throat, maybe this patient does too."
    2. They suggest the same treatment: antibiotics.
3. **Revise**:
    1. The doctor runs a test and finds out the patient actually has the **flu**, not strep throat.
    2. They adjust the treatment: rest, fluids, and antiviral medication.
4. **Retain**:
    1. The doctor saves this new case (fever, cough, sore throat = flu) in their memory for future use.

Example: A Car Mechanic Using CBR

- Imagine a **car mechanic** who uses CBR to fix cars.

1. **Retrieve**:
    1. A customer brings in a car that won’t start.
    2. The mechanic remembers a past case where a car wouldn’t start because of a **dead battery**.
2. **Reuse**:
    1. The mechanic thinks, "Maybe this car also has a dead battery."
    2. They try replacing the battery.
3. **Revise**:
    1. The car still won’t start. The mechanic realizes the problem is actually a **faulty starter motor**.
    2. They fix the starter motor instead.
4. **Retain**:
    1. The mechanic saves this new case (car won’t start = faulty starter motor) for future use.

Why is CBR Useful?

1. **Learning from Experience**:
    1. The system gets better over time as it saves more cases.
2. **Handling Complex Problems**:
    1. It works well for problems that are hard to solve with simple rules.
3. **Adaptability**:
    1. It can adjust solutions based on new information.

Applications of CBR

1. **Medical Diagnosis**:
    1. Doctors use past cases to diagnose new patients.
    2. Example: A system that suggests treatments based on past patient records.
2. **Customer Support**:
    1. Chatbots use past cases to answer customer questions.
    2. Example: If a customer asks, "How do I reset my password?" the chatbot finds a similar past case and provides the same solution.
3. **Legal Systems**:
    1. Lawyers use past cases to argue new ones.
    2. Example: A system that suggests legal strategies based on past court rulings.
4. **Engineering**:
    1. Engineers use past designs to create new ones.
    2. Example: A system that suggests design improvements based on past projects.
5. **Education**:
    1. Teachers use past lesson plans to teach new topics.
    2. Example: A system that suggests teaching methods based on past student performance.

Advantages of CBR

1. **Easy to Understand**:
    1. It’s based on real cases, so it’s intuitive.
2. **Improves Over Time**:
    1. The more cases it saves, the better it gets.
3. **Handles Uncertainty**:
    1. It can adapt solutions even if the new problem isn’t exactly the same as past cases.

Limitations of CBR

- **Depends on Past Cases**:
- If there are no similar past cases, the system might struggle.
- **Storage Issues**:
- Storing too many cases can make the system slow.
- **Bias**:
- If past cases are biased, the system might give biased solutions.
-

Natural Language Processing (NLP)?

What is Natural Language Processing (NLP)?

- NLP is a branch of artificial intelligence (AI) that helps computers **understand**, **interpret**, and **respond** to human language. It’s what makes things like **voice assistants**, **chatbots**, and **translation tools** possible.
- Think of NLP as a **bridge** between humans and computers. Humans speak or write in natural language (like English, Hindi, or Spanish), and NLP helps computers make sense of it.

Why is NLP Important?

- **Communication**: It helps computers talk to us in a way we understand.
- **Automation**: It can automate tasks like answering customer questions or translating languages.
- **Insights**: It can analyze large amounts of text to find useful information (e.g., customer feedback).
-

Key Steps in NLP

1. **Tokenization**:
    1. Breaking down a sentence into smaller parts, like words or phrases.
    2. Example:
        1. Sentence: *"I love ice cream."*
        2. Tokens: *["I", "love", "ice", "cream", "."]*
2. **Morphological Analysis**:
    1. Understanding the structure of words (e.g., root words, prefixes, suffixes).
    2. Example:
        1. Word: *"running"*
        2. Root: *"run"*
        3. Suffix: *"-ing"* (shows continuous action).

Key Steps in NLP

3. Syntax Analysis (Parsing):

1. Analyzing the grammar and structure of a sentence.
2. Example:
   1. Sentence: *"The cat sat on the mat."*
   2. Syntax: *The cat (subject) sat (verb) on the mat (prepositional phrase).*

4. Semantic Analysis:

1. Understanding the meaning of words and sentences.
2. Example:
   1. Sentence: *"She gave him a book."*
   2. Meaning: A female person handed a book to a male person.

Key Steps in NLP

5. Pragmatic Analysis:

1. Understanding the context and intent behind the words.
2. Example:
    1. Sentence: *"Can you open the window?"*
    2. Intent: A request to open the window, not just a question.

Examples of NLP

1. **Voice Assistants**:
   1. Example: *Alexa, play my favorite song.*
   2. NLP helps Alexa understand your request and play the song.
2. **Chatbots**:
   1. Example: A chatbot on a website answers questions like, *"What’s my order status?"*
   2. NLP helps the chatbot understand your question and provide the right answer.

Examples of NLP

3. Translation Tools:

1. Example: Google Translate converts *"Hello"* to *"Hola"* in Spanish.
2. NLP helps the tool understand and translate languages.

4. Spell Checkers:

1. Example: When you type *"teh"*, it suggests *"the"*.
2. NLP helps identify and correct spelling mistakes.

5. Sentiment Analysis:

1. Example: A company analyzes tweets to see if people like their product.
2. NLP helps determine if the tweets are positive, negative, or neutral.

Example: How NLP Works in a Chatbot

- Let’s say you ask a chatbot: *"What’s the weather today?"*

1. **Tokenization**:
    1. Breaks the sentence into: *["What’s", "the", "weather", "today", "?"]*
2. **Syntax Analysis**:
    1. Identifies the structure: *"What’s"* (question), *"the weather"* (subject), *"today"* (time).
3. **Semantic Analysis**:
    1. Understands the meaning: You’re asking about the weather for today.
4. **Pragmatic Analysis**:
    1. Recognizes the intent: You want to know the current weather.
5. **Response**:
    1. The chatbot replies: *"Today’s weather is sunny with a high of 25°C."*

Challenges in NLP

1. **Ambiguity**:
    1. Words can have multiple meanings.
    2. Example: *"Bank"* can mean a riverbank or a financial institution.
2. **Sarcasm and Humor**:
    1. Computers struggle to understand sarcasm or jokes.
    2. Example: *"Oh great, another Monday!"* (sarcasm).
3. **Slang and Informal Language**:
    1. People use slang, abbreviations, and informal language.
    2. Example: *"BRB"* means *"Be Right Back"*.
4. **Context**:
    1. Understanding the context of a conversation can be hard.
    2. Example: *"She gave him a ring."* (Is it a jewelry ring or a phone ring?)

How Does NLP Solve These Challenges?

1. **Machine Learning**:
   1. NLP systems learn from large amounts of text data to improve accuracy.
2. **Contextual Models**:
   1. Tools like **BERT** and **GPT** understand the context of words in a sentence.
3. **Human Feedback**:
   1. Systems improve by learning from corrections made by humans.

Fuzzy Logic Introduction

- The term **fuzzy** refers to things that are not clear or are vague.
- In the real world many times we encounter a situation when we can't determine whether the state is true or false, their fuzzy logic provides very valuable flexibility for reasoning.
- Fuzzy Logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1, instead of just the traditional values of true or false.

Fuzzy Logic Introduction

- Fuzzy Logic is based on the idea that in many cases, the concept of true or false is too restrictive, and that there are many shades of gray in between.
- It allows for partial truths, where a statement can be partially true or false, rather than fully true or false.
- Fuzzy Logic is used in a wide range of applications, such as control systems, image processing, natural language processing, medical diagnosis, and artificial intelligence.

Fuzzy Logic Introduction

- The fundamental concept of Fuzzy Logic is the membership function, which defines the degree of membership of an input value to a certain set or category.
- The membership function is a mapping from an input value to a membership degree between 0 and 1, where 0 represents non-membership and 1 represents full membership.

Fuzzy Logic Introduction

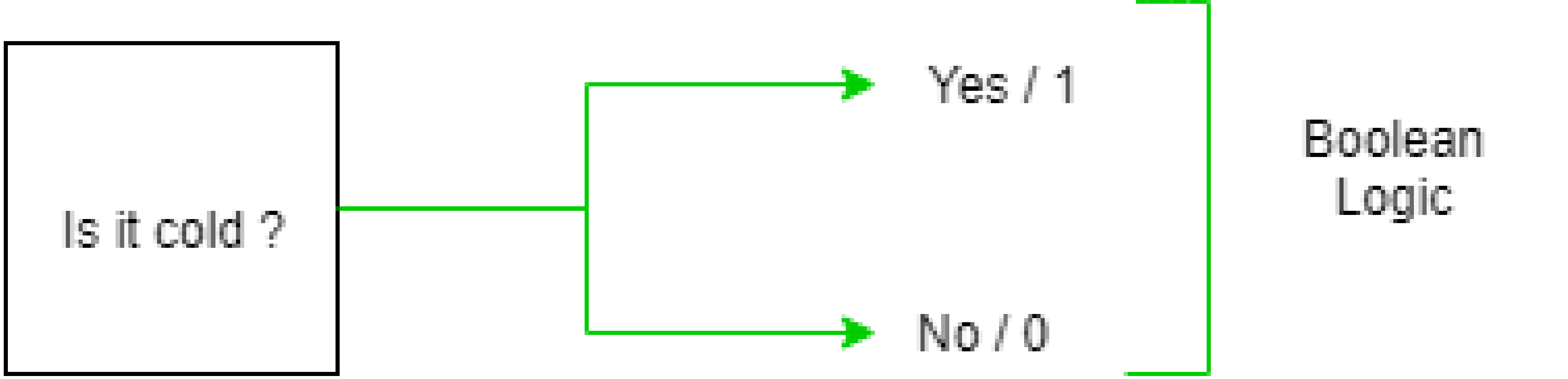
- Fuzzy Logic is implemented using Fuzzy Rules, which are if-then statements that express the relationship between input variables and output variables in a fuzzy way.
- The output of a Fuzzy Logic system is a fuzzy set, which is a set of membership degrees for each possible output value.

Fuzzy Logic Introduction

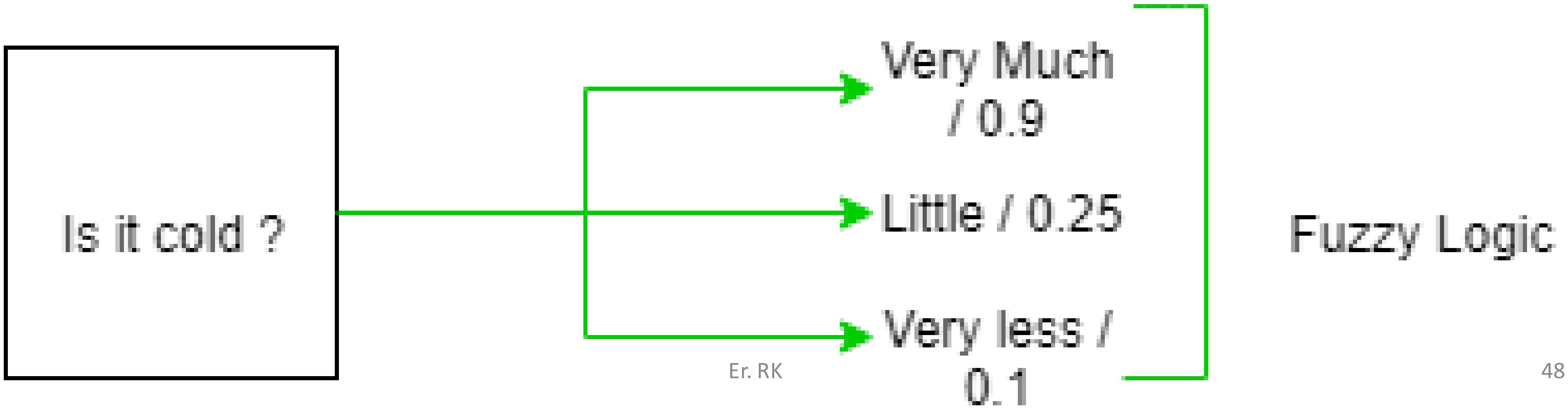
- In summary, Fuzzy Logic is a mathematical method for representing vagueness and uncertainty in decision-making, it allows for partial truths, and it is used in a wide range of applications. It is based on the concept of membership function and the implementation is done using Fuzzy rules.

Example

- In the boolean system truth value, 1.0 represents the absolute truth value and 0.0 represents the absolute false value.
- But in the fuzzy system, there is no logic for the absolute truth and absolute false value.
- But in fuzzy logic, there is an intermediate value too present which is partially true and partially false.



Is it cold ?
Yes / 1
No / 0
Boolean Logic



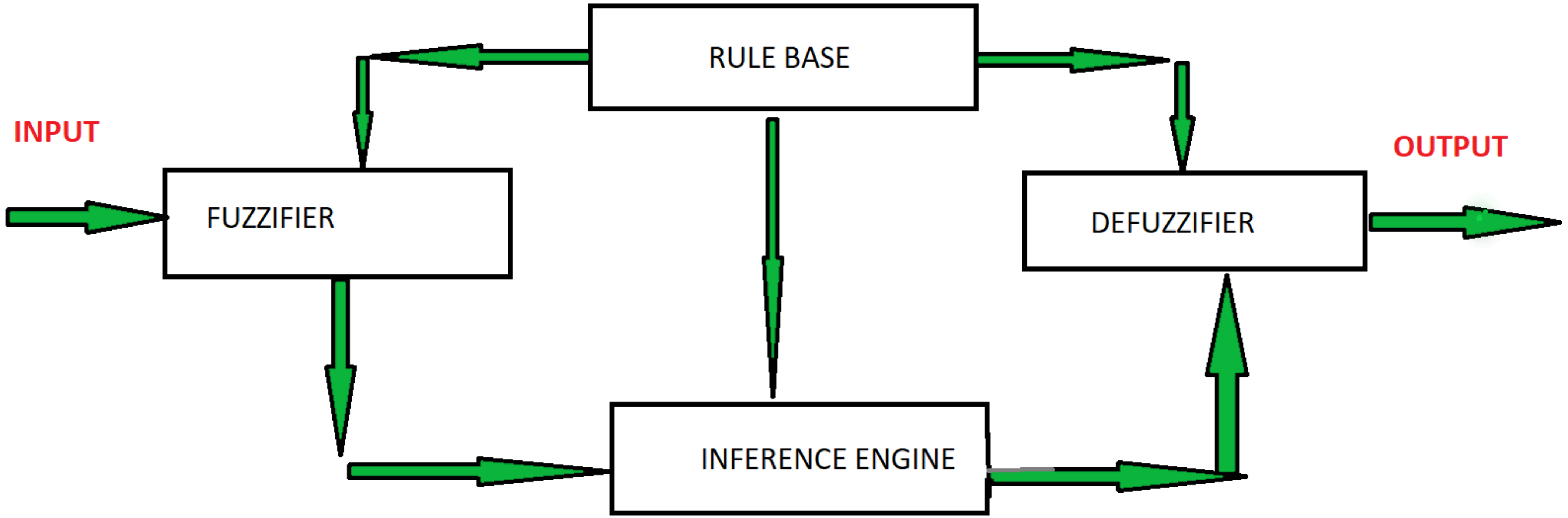
Is it cold ?
Very Much / 0.9
Little / 0.25
Very less / 0.1
Fuzzy Logic

ARCHITECTURE

- Its Architecture contains four parts :
- RULE BASE: It contains the set of rules and the IF-THEN conditions provided by the experts to govern the decision-making system, on the basis of linguistic information. Recent developments in fuzzy theory offer several effective methods for the design and tuning of fuzzy controllers. Most of these developments reduce the number of fuzzy rules.
- FUZZIFICATION: It is used to convert inputs i.e. crisp numbers into fuzzy sets. Crisp inputs are basically the exact inputs measured by sensors and passed into the control system for processing, such as temperature, pressure, rpm’s, etc.

ARCHITECTURE

- INFERENCE ENGINE: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules are to be fired according to the input field. Next, the fired rules are combined to form the control actions.
- DEFUZZIFICATION: It is used to convert the fuzzy sets obtained by the inference engine into a crisp value. There are several defuzzification methods available and the best-suited one is used with a specific expert system to reduce the error.



FUZZY LOGIC ARCHITECTURE

Membership function

- **Definition:** A graph that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred to as the universe of discourse or universal set (u), which contains all the possible elements of concern in each particular application.
- There are largely three types of fuzzifiers:
- Singleton fuzzifier
- Gaussian fuzzifier
- Trapezoidal or triangular fuzzifier

What is Fuzzy Control?

- It is a technique to embody human-like thinkings into a control system.
- It may not be designed to give accurate reasoning but it is designed to give acceptable reasoning.
- It can emulate human deductive thinking, that is, the process people use to infer conclusions from what they know.
- Any uncertainties can be easily dealt with the help of fuzzy logic.
-

Advantages of Fuzzy Logic System

- This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.
- The construction of Fuzzy Logic Systems is easy and understandable.
- Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.
- It provides a very efficient solution to complex problems in all fields of life as it resembles human reasoning and decision-making.
- The algorithms can be described with little data, so little memory is required.

Disadvantages of Fuzzy Logic Systems

- Many researchers proposed different ways to solve a given problem through fuzzy logic which leads to ambiguity. There is no systematic approach to solve a given problem through fuzzy logic.
- Proof of its characteristics is difficult or impossible in most cases because every time we do not get a mathematical description of our approach.
- As fuzzy logic works on precise as well as imprecise data so most of the time accuracy is compromised.

Application

- It is used in the aerospace field for altitude control of spacecraft and satellites.
- It has been used in the automotive system for speed control, traffic control.
- It is used for decision-making support systems and personal evaluation in the large company business.
- Fuzzy logic is used in Natural language processing and various intensive applications in Artificial Intelligence.
- Fuzzy logic is extensively used in modern control systems such as expert systems.
- Fuzzy Logic is used with Neural Networks as it mimics how a person would make decisions, only much faster.

Fuzzy Logic - Inference System

- Fuzzy Inference System is the key unit of a fuzzy logic system having decision making as its primary work. It uses the “IF…THEN” rules along with connectors “OR” or “AND” for drawing essential decision rules.

Characteristics of Fuzzy Inference System

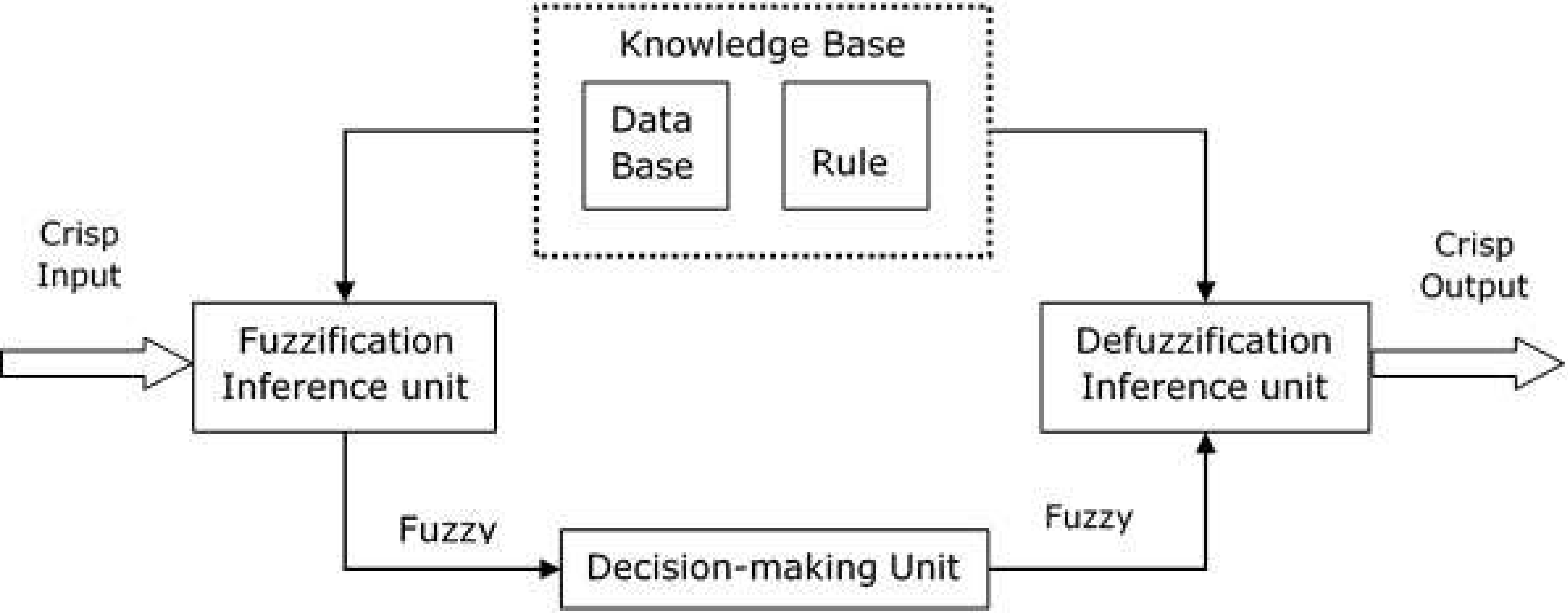
Following are some characteristics of FIS −

- The output from FIS is always a fuzzy set irrespective of its input which can be fuzzy or crisp.
- It is necessary to have fuzzy output when it is used as a controller.
- A defuzzification unit would be there with FIS to convert fuzzy variables into crisp variables.

Functional Blocks of FIS

- The following five functional blocks will help you understand the construction of FIS −
- **Rule Base** − It contains fuzzy IF-THEN rules.
- **Database** − It defines the membership functions of fuzzy sets used in fuzzy rules.
- **Decision-making Unit** − It performs operation on rules.
- **Fuzzification Interface Unit** − It converts the crisp quantities into fuzzy quantities.
- **Defuzzification Interface Unit** − It converts the fuzzy quantities into crisp quantities. Following is a block diagram of fuzzy interference system.

Functional Blocks of FIS



Working of FIS

- The working of the FIS consists of the following steps –
- A fuzzification unit supports the application of numerous fuzzification methods, and converts the crisp input into fuzzy input.
- A knowledge base - collection of rule base and database is formed upon the conversion of crisp input into fuzzy input.
- The defuzzification unit fuzzy input is finally converted into crisp output.

Genetic Algorithm – Machine Learning

- A **genetic algorithm** is a **heuristic search** algorithm that is inspired by Charles Darwin's theory of natural evolution.
- This algorithm reflects the process of **natural selection** where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.
- Genetic Algorithms are being widely used in different **real-world applications**, for example, **image processing**, **Designing electronic circuits**, **code-breaking**, and **artificial creativity**.

How Genetic Algorithm works?

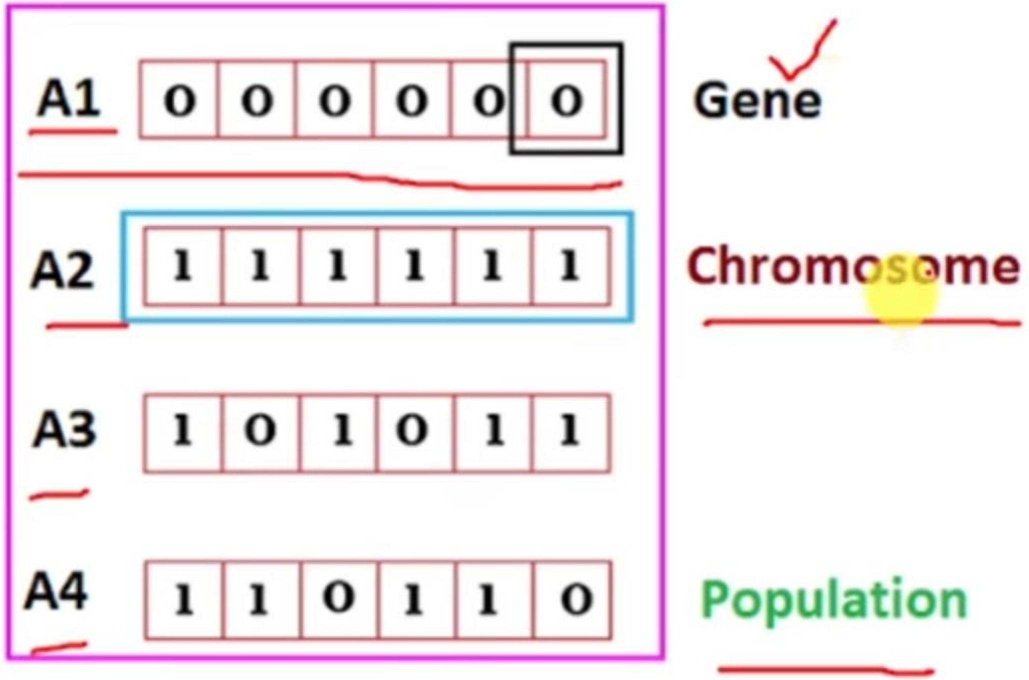
There are five phases in Genetic Algorithm:

- Initialization
- Fitness Assignment
- Selection
- Crossover (Reproduction)
- Termination

How Genetic Algorithm works?

Initial Population

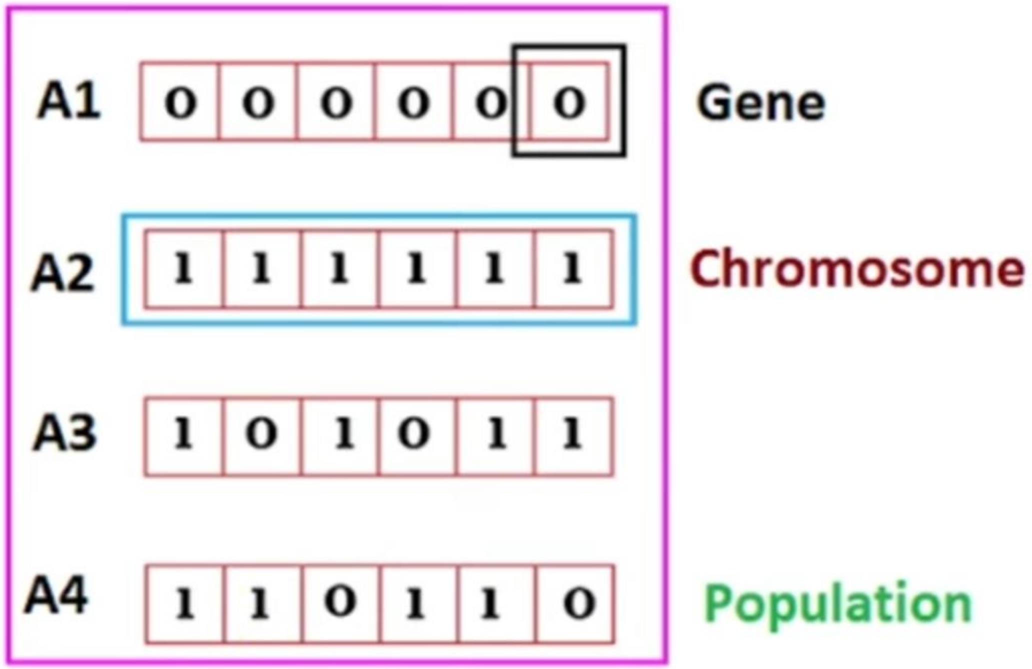
- The process begins with a set of individuals which is called a Population.
- Each individual is a solution to the problem you want to solve known as Chromosome.
- An individual is characterized by a set of parameters (variables) known as Genes.
- Genes are joined into a string to form a Chromosome (solution).



How Genetic Algorithm works?

Fitness Function

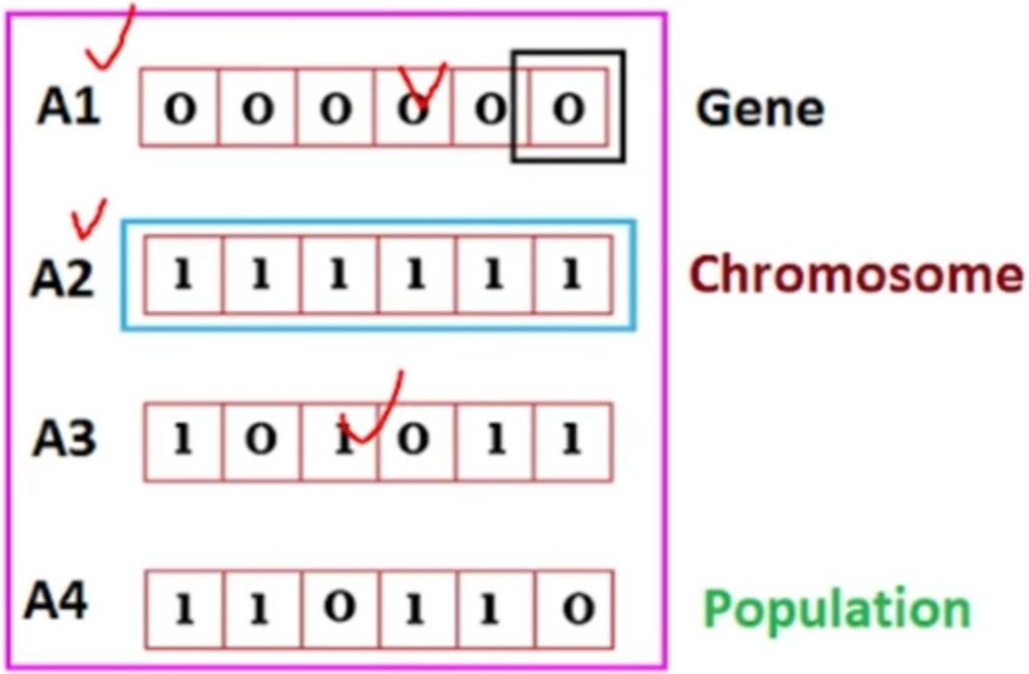
- The fitness function determines **how fit** an individual is? (the ability of an individual to compete with other individuals).
- It gives a fitness score to each individual.
- The probability that an individual will be selected for reproduction is based on its **fitness score**.



How Genetic Algorithm works?

Selection

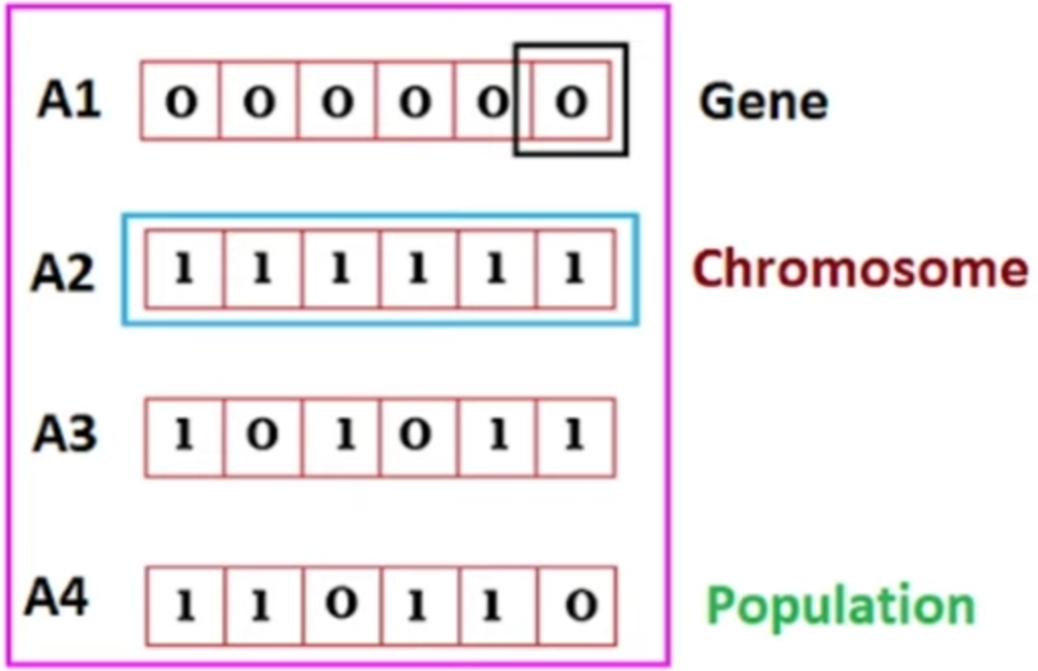
- The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation.
- Two pairs of individuals (parents) are selected based on their fitness scores.
- Individuals with high fitness have more chance to be selected for reproduction.



How Genetic Algorithm works?

There are three types of Selection methods available, which are:

1. Roulette wheel selection
2. Tournament selection
3. Rank-based selection



How Genetic Algorithm works?

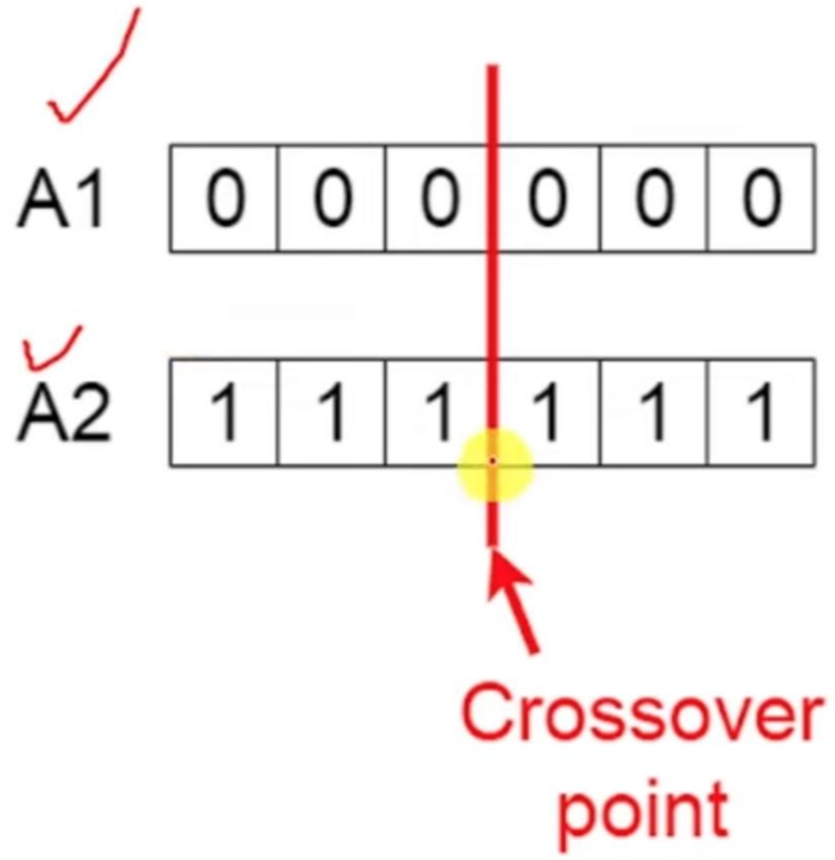
Crossover

- Crossover is the most significant phase in a genetic algorithm.
- For each pair of parents to be mated, a crossover point is chosen at random from within the genes.
- For example, consider the crossover point to be 3 as shown.

How Genetic Algorithm works?

Crossover

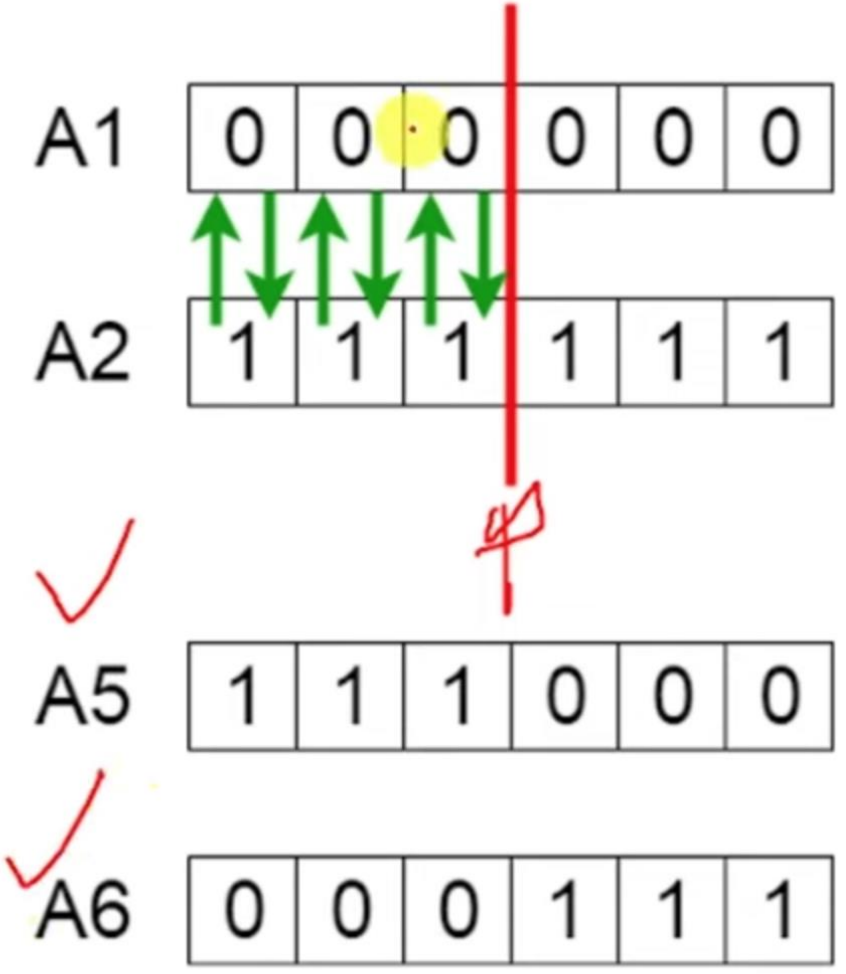
- Crossover is the most significant phase in a genetic algorithm.
- For each pair of parents to be mated, a crossover point is chosen at random from within the genes.
- For example, consider the crossover point to be 3 as shown.



How Genetic Algorithm works?

Offspring

- **Offspring** are created by exchanging the genes of parents among themselves until the crossover point is reached.
- The new offspring are added to the population.



How Genetic Algorithm works?

Mutation

- In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability.
- This implies that some of the bits in the bit string can be flipped.

Before Mutation

A5	1	1	1	0	0	0

After Mutation

A5	1	1	0	1	1	0

How Genetic Algorithm works?

Types of mutation styles available,

1. Flip bit mutation
2. Gaussian mutation
3. Exchange/Swap mutation

Before Mutation

A5	1	1	1	0	0	0

After Mutation

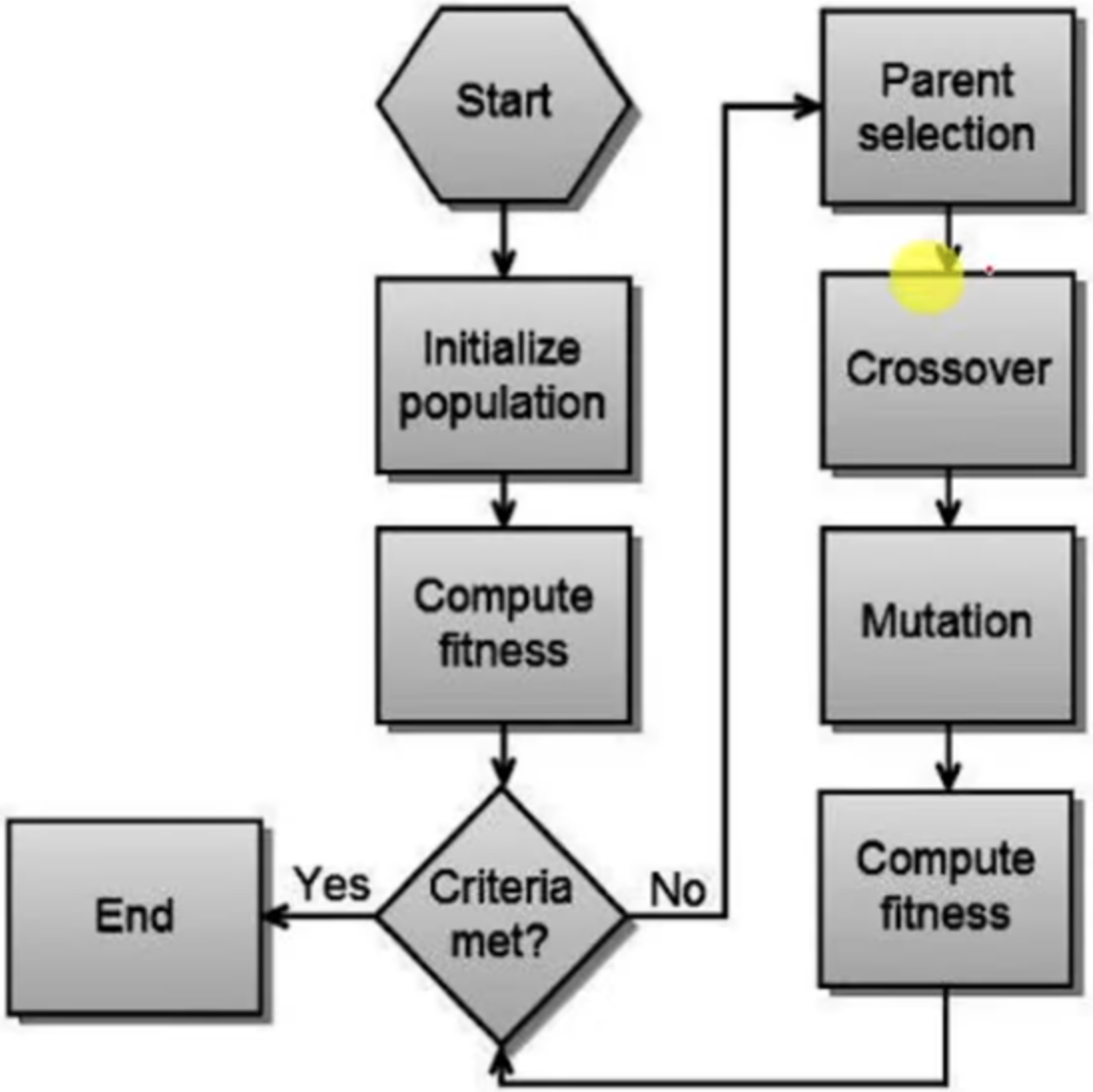
A5	1	1	0	1	1	0

How Genetic Algorithm works?

Termination

- The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation).
- Then it is said that the genetic algorithm has provided a set of solutions to our problem.

How Genetic Algorithm works?



Advantages of Genetic Algorithm

- The parallel capabilities of genetic algorithms are best.
- It helps in optimizing various problems such as discrete functions, multi-objective problems, and continuous functions.
- It provides a solution for a problem that improves over time.
- A genetic algorithm does not need derivative information.

Limitations of Genetic Algorithms

- Genetic algorithms are not efficient algorithms for solving simple problems.
- It does not guarantee the quality of the final solution to a problem.
- Repetitive calculation of fitness values may generate some computational challenges.

Bayes' Theorem

Simple Explanation

- Imagine you’re trying to guess if it's raining outside. Before you look outside, you have some idea of how likely it is to rain today based on the weather report—this is your **initial guess**, or **prior probability**.
- But now, you get new information: your friend comes inside wearing a wet raincoat. This new information, called **evidence**, makes you reconsider the chances that it's raining. How likely is it now?
- Bayes' Theorem helps you **update your guess** about the chances of rain, given the new information (the wet raincoat).

Basic Idea of Bayes' Theorem

1. **Start with a belief**: You know how likely something is (like rain) before getting new evidence.
    1. For example, the weather forecast said there's a **10% chance** of rain today.
2. **Get new evidence**: You see your friend with a wet raincoat, which suggests it might be raining.
3. **Ask two questions**:
    1. How often would you expect to see a wet raincoat if it was raining? (Maybe **90%** of people wear wet coats in the rain.)
    2. How often would you expect to see a wet raincoat if it wasn't raining? (Maybe **5%** of people still have wet coats from something else.)
4. **Update your belief**: You combine your starting belief (10% chance of rain) with the new evidence (the wet coat) to get a new, **better estimate** of the chance it's raining.

Bayes' Theorem

- Bayes' Theorem is a fundamental concept in probability theory and statistics that describes how to update the probability of a hypothesis based on new evidence.
- It is named after the Reverend Thomas Bayes and provides a way to revise existing predictions or theories in light of new data.

The theorem is stated mathematically as:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Where:

- $P(A|B)$ is the posterior probability: the probability of hypothesis A given the evidence B.
- $P(B|A)$ is the likelihood: the probability of evidence B given that hypothesis A is true.
- $P(A)$ is the prior probability: the initial probability of hypothesis A before observing the evidence B.
- $P(B)$ is the marginal likelihood: the total probability of evidence B under all possible hypotheses.

Explanation of Terms

1. **Prior Probability P(A)**:
   1. The initial probability of the hypothesis before any evidence is observed.
2. **Likelihood P(B|A)**:
   1. The probability of observing the evidence assuming the hypothesis is true.
3. **Posterior Probability P(A|B)**:
   1. The updated probability of the hypothesis after considering the new evidence.
4. **Marginal Probability P(B)**:
   1. The probability of observing the evidence under all possible scenarios.

Why Bayes' Theorem Works

- Bayes' Theorem relies on the **Law of Total Probability**, which tells us how to calculate the overall probability of an event by breaking it down into smaller conditional probabilities.
- Imagine you're trying to figure out how likely event A is after seeing B. You know two things:

1. How strongly event B supports A (the likelihood, $P(B|A)$).
2. How common or rare AAA is in general (the prior, $P(A)$).

- You also need the marginal probability $P(B)$, which accounts for how often you'd see B, regardless of whether A is true or not.
- In essence, **Bayes' Theorem balances the evidence (likelihood) with your prior knowledge to refine your belief**.

BAYE'S THEOREM: Describes the probability of an event, based on prior Knowledge of conditions that might be related to the event.

↳ In Probability theory it relates the Conditional probability & marginal probabilities of two random events.

↳ $P(H|E) = \frac{\text{no. of time H and E}}{\text{no. of times E}}$

↳ Calculate $P(B|A)$ with Knowledge of $P(A|B)$.

$P(H/E) = \frac{P(H \cap E)}{P(E)}$ } Prob. of H when E is true.

$P(A \cap B) = P(A|B).P(B)$ — (i)

$P(A \cap B) = P(B|A).P(A)$ — (ii)

} From (i) and (ii) L.H.S are equal.

↳ $P(A|B).P(B) = P(B|A).P(A)$

Posterior (Prob. of A when B is true)

So, $\boxed{P(B|A) = \frac{P(A|B).P(B)}{P(A)}}$ Baye's theorem formula.

Likeli hood (Prob. of evidence)

Prior Prob (Prob. of hypothesi

Subtitles/closed captions (c)

BAYE'S THEOREM: Describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

↳ In Probability theory it relates the Conditional probability & marginal probabilities of two random events.

↳ $P(H|E) = \dfrac{\text{no. of time H and E}}{\text{no. of times E}}$

↳ Calculate $P(B|A)$ with Knowledge of $P(A|B)$.

$P(H|E) = \dfrac{P(H \cap E)}{P(E)}$ } Prob. of H when E is true.

$P(A \cap B) = P(A|B).P(B)$ — (i)

$P(A \cap B) = P(B|A).P(A)$ — (ii)

} From (i) and (ii) L.H.S are equal.

↳ $P(A|B).P(B) = P(B|A).P(A)$

So, $\boxed{P(B|A) = \dfrac{P(A|B).P(B)}{P(A)}}$

Posterior (Prob. of A when B is true)

marginal Prob. (Prob. of evidence

Baye's theorem formula.

Likeli hood (Prob. of evidence)

Prior Prob (Prob. of hypothesis

Baye's theorem Example 1:

Ques1:- what is the probability that Person has disease dengue with neck pain?

↳ Given:-
- 80% of time dengue causes neck pain. $P(a|b) = 0.8$
- ↳ $P(\text{dengue}) = 1/30{,}000$ ($P(b) \Rightarrow 1/30{,}000$)
- ↳ $P(\text{neckpain}) = .02$ $P(a) = .02$

a = Proposistion that Person has neck pain

b = Person has dengue.

$P(b|a) = ?$

$$P(b|a) = \frac{P(a|b).P(b)}{P(a)}$$

$$= \frac{0.8 . 1/30000}{0.02} = 0.00133$$

Applications

- **Medical Diagnosis**: Updating the probability of a disease based on test results.
- **Spam Detection**: Classifying emails as spam or not based on features like words in the email.
- **Machine Learning (Naive Bayes)**: A simple but powerful classifier that assumes independence between features.
- **Risk Analysis**: Assessing the probability of an event (e.g., defaulting on a loan) given new data.
- **Fraud Detection**: Identifying fraudulent transactions by updating prior knowledge of normal behavior with new evidence.

Application of Baye's theorem in AI:

→ ① Robot/Automatic machine] next step is Calculated based on Prev. step

② Forecasting.

→ Weather

WHY BAYESIAN METHODS ARE IMPORTANT?

- Bayesian learning algorithms, like the naive Bayes classifier which calculates the explicit probabilities for hypotheses.
- Bayesian classifiers, the training data are utilized to calculate an observed probability of each class based on feature values.
- These naïve bayes classifier is equally powerful as decision tree and neural network algorithms. Some times it may give even better results.
- It is highly practical approaches to certain types of learning problems.

Advantages of Bayes' Theorem:

- **Incorporation of Prior Knowledge:** Bayes' Theorem allows the incorporation of prior knowledge or beliefs into the analysis, which can be particularly useful when data is scarce.
- **Dynamic Updating:** It provides a systematic method for updating probabilities as new evidence becomes available, making it suitable for real-time applications.
- **Flexibility:** It can handle complex situations where multiple hypotheses and pieces of evidence are involved.
- **Foundation for Bayesian Methods:** Bayes' Theorem is the foundation of Bayesian statistics, which is widely used in various scientific and engineering fields.

Disadvantages of Bayes' Theorem:

- **Dependence on Prior:** The result can be heavily influenced by the choice of the prior, which can be subjective and may not always be easy to determine.
- **Computational Complexity:** Calculating the posterior distribution can be computationally intensive, especially for large datasets or complex models.
- **Interpretation:** The results can be difficult to interpret, particularly for those not familiar with Bayesian methods.
- **Sensitivity to Model Assumptions:** The accuracy of the results depends on the correctness of the model assumptions and the form of the prior distribution.

Example: Email Spam Filtering

- One of the most practical applications of **Bayes' Theorem** is in spam filtering.
- Email services like Gmail and Outlook use machine learning models (including **Naive Bayes** classifiers) to determine if an incoming email is **spam** or **not spam** based on the words and patterns in the email.

Scenario

- You receive an email with the subject line: **"Congratulations! You've won a free trip to Hawaii!"**
- Your spam filter needs to decide if this email is spam. The filter uses **Bayes' Theorem** to calculate the probability that the email is spam given the word "free" appears in the subject.

- **Given Data**

1. **Prior Probability P(Spam)** :
    1. About **40%** of emails are spam: $P(Spam)=0.40$.
2. **Likelihood P(Free|Spam):**
    1. The word "free" appears in **70%** of spam emails: $P(Free|Spam)=0.70$.
3. **False Positive Rate P(Free|Not Spam)**:
    1. The word "free" appears in only **5%** of non-spam emails: $P(Free|Not\ Spam)=0.05$.
4. **Complementary Probability P(Not Spam)** :
    1. $P(Not\ Spam)=1-P(Spam)=0.60$.

- **Goal**
- We want to calculate the probability that the email is spam given that the word "free" appears in the subject, i.e., P(Spam|Free).

1. Calculate the Marginal Probability $P(\text{Free})$:

- This is the total probability of the word "free" appearing in any email, whether spam or not:

$$P(\text{Free}) = P(\text{Free}|\text{Spam}) \cdot P(\text{Spam}) + P(\text{Free}|\text{Not Spam}) \cdot P(\text{Not Spam})$$

Substituting the values:

$$P(\text{Free}) = (0.70 \times 0.40) + (0.05 \times 0.60)$$

$$P(\text{Free}) = 0.28 + 0.03 = 0.31$$

2. **Apply Bayes' Theorem:**

We now calculate $P(\text{Spam}|\text{Free})$:

$$P(\text{Spam}|\text{Free}) = \frac{P(\text{Free}|\text{Spam}) \cdot P(\text{Spam})}{P(\text{Free})}$$

Substituting the values:

$$P(\text{Spam}|\text{Free}) = \frac{0.70 \times 0.40}{0.31} = \frac{0.28}{0.31} \approx 0.9032$$

- **Interpretation**
- The spam filter calculates that there is approximately a **90.32%** chance that the email is spam based on the presence of the word "free." This probability is high enough for the filter to classify the email as spam.