UNIT 6

INT426:GENERATIVE ARTIFICIAL INTELLIGENCE

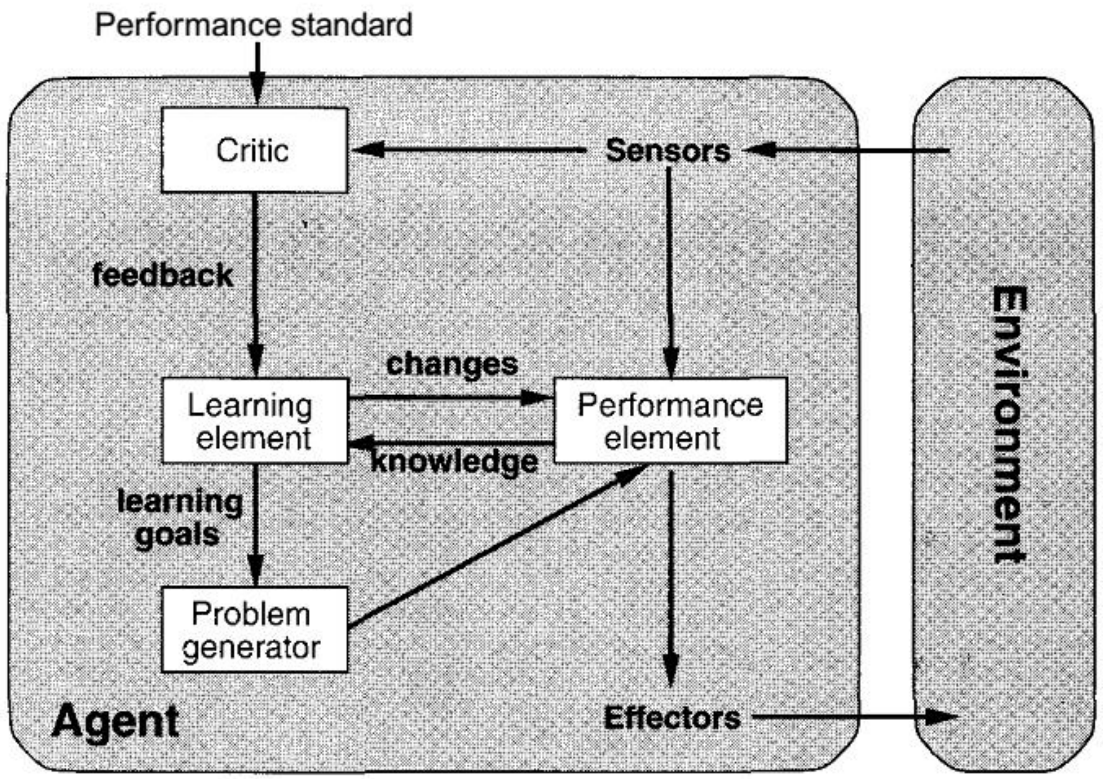
Learning:

- The idea behind learning is that percepts should be used not only for acting, but also for improving the agent's ability to act in the future.
- Learning takes place as a result of the interaction between the agent and the world, and from observation by the agent of its own decision-making processes.

A GENERAL MODEL OF LEARNING AGENTS

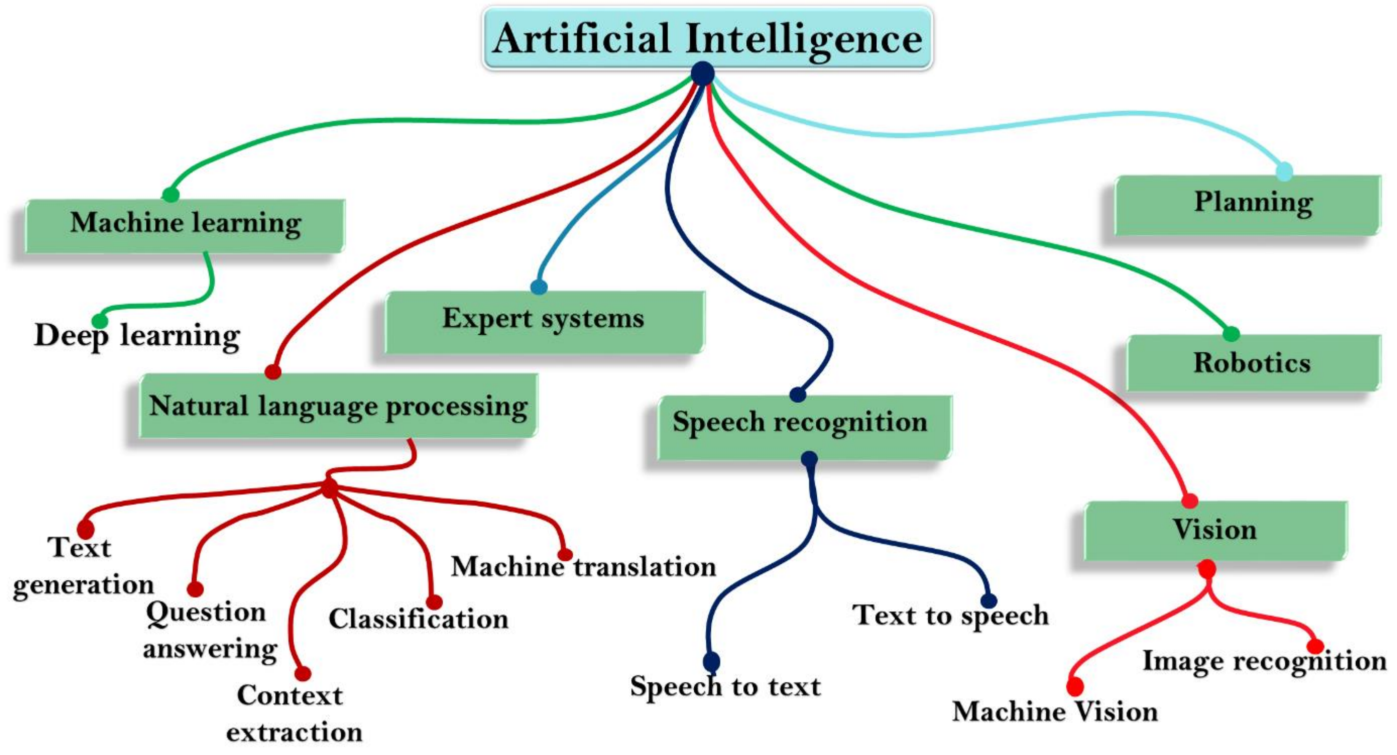
A learning agent can be divided into four conceptual components

- **Learning element:** is responsible for making improvements
- **Performance element:** is responsible for selecting external actions
- **Critic:** is designed to tell the learning element how well the agent is doing
- **Problem Generator:** is responsible for suggesting actions that will lead to new and informative experiences


Performance standard
Critic
Sensors
feedback
changes
Learning element
Performance element
knowledge
learning goals
Problem generator
Agent
Effectors
Environment

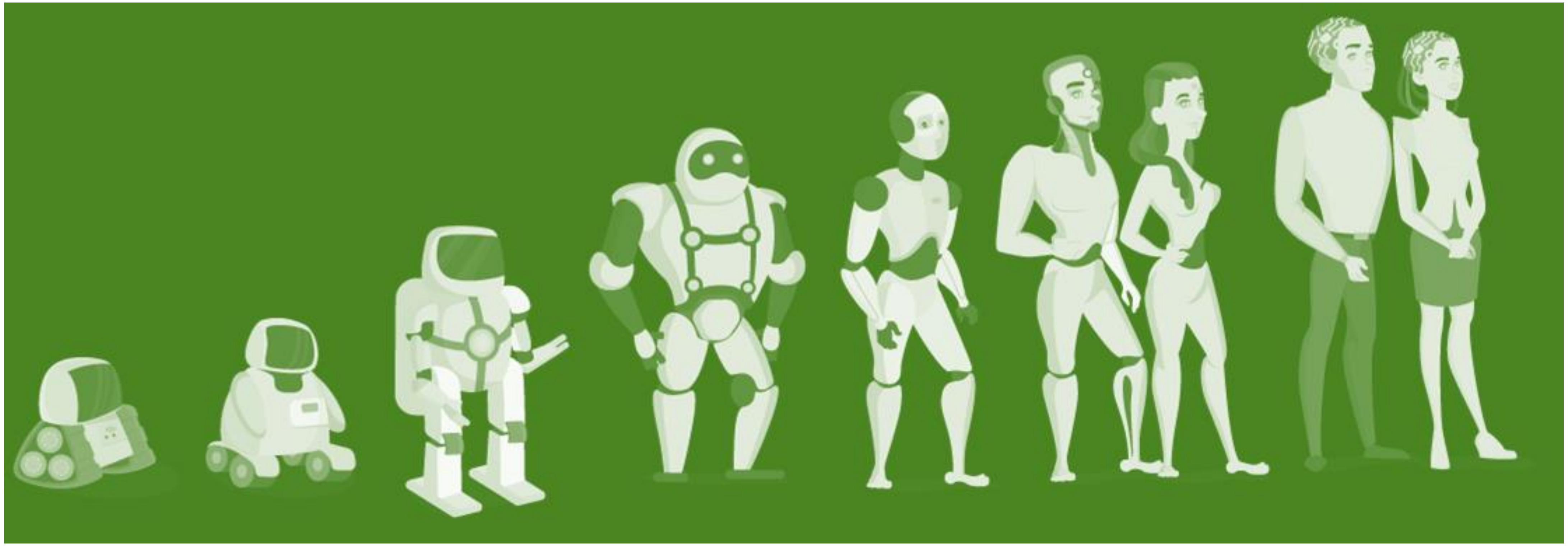
What is Machine Learning?

Machine Learning is a subset of artificial intelligence which focuses mainly on machine learning from their experience and making predictions based on its experience.

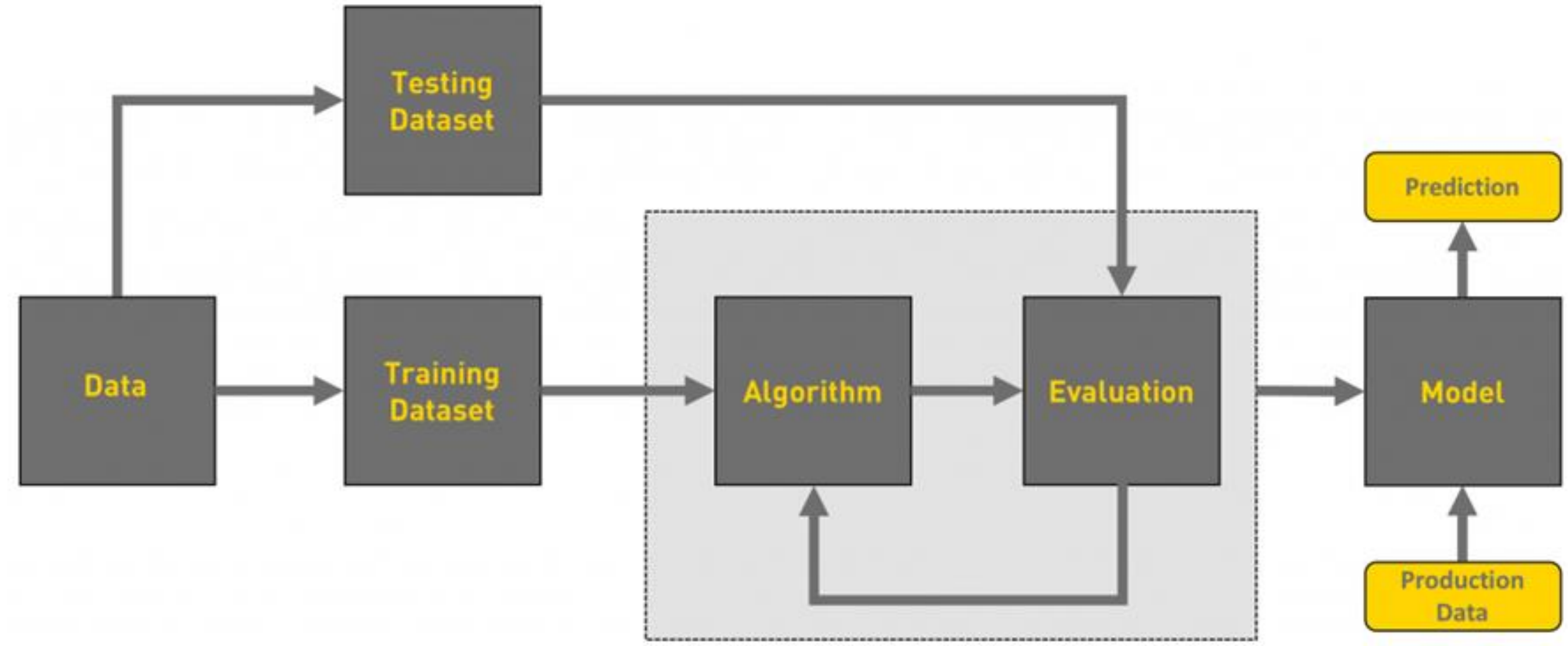


- It enables the computers or the machines to make data-driven decisions rather than being explicitly programmed for carrying out a certain task.
- These programs or algorithms are designed in a way that they learn and improve over time when are exposed to new data.

What is Machine Learning - Evolution of Machines



Machine Learning Workflow



Machine Learning Workflow Contd.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

1.Gathering data

We can also use some free data sets which are present on the internet. **Kaggle** and **UCI Machine learning Repository** are the repositories that are used the most for making Machine learning models.

Types of Data

1. **Numeric** e.g. income, age

2. Categorical :- sometimes called qualitative data, are data whose values describe some characteristic or category(e.g. gender, nationality)

3. Ordinal:- type of data that follows a natural order. The main features of ordinal data are that the difference between data values cannot be determined (e.g. low/medium/high , **socio economic status** ("low income","middle income","high income"), **education level** ("high school","BS","MS","PhD"), **income level** ("less than 50K", "50K-100K", "over 100K"), **satisfaction rating** ("extremely dislike", "dislike", "neutral", "like", "extremely like").

2. Data pre-processing

- Data pre-processing is one of the most important steps in machine learning.
- It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule.
- Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.

- Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set.

Most of the real-world data is messy, some of these types of data are:

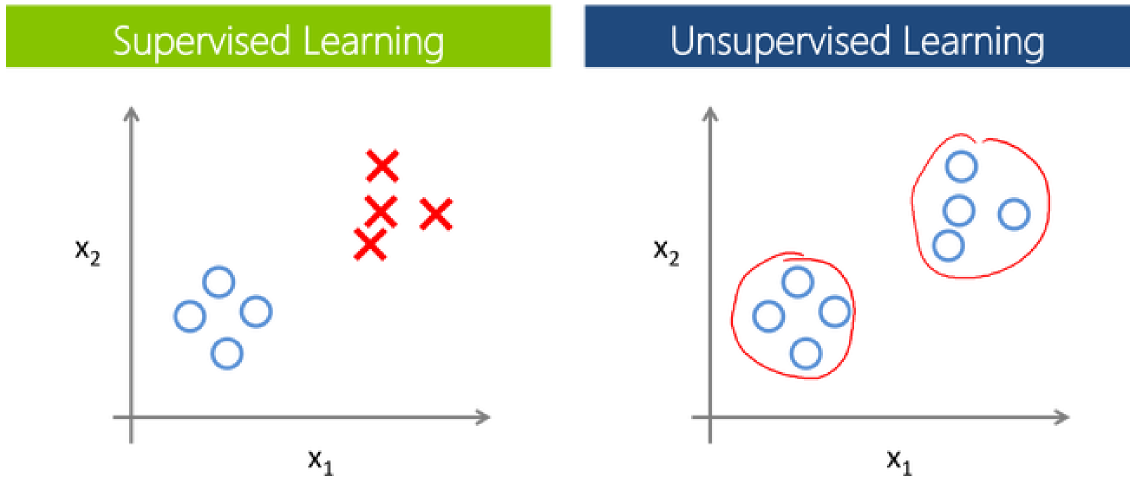
1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).
2. **Noisy data:** This type of data is also called outliners, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.
3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

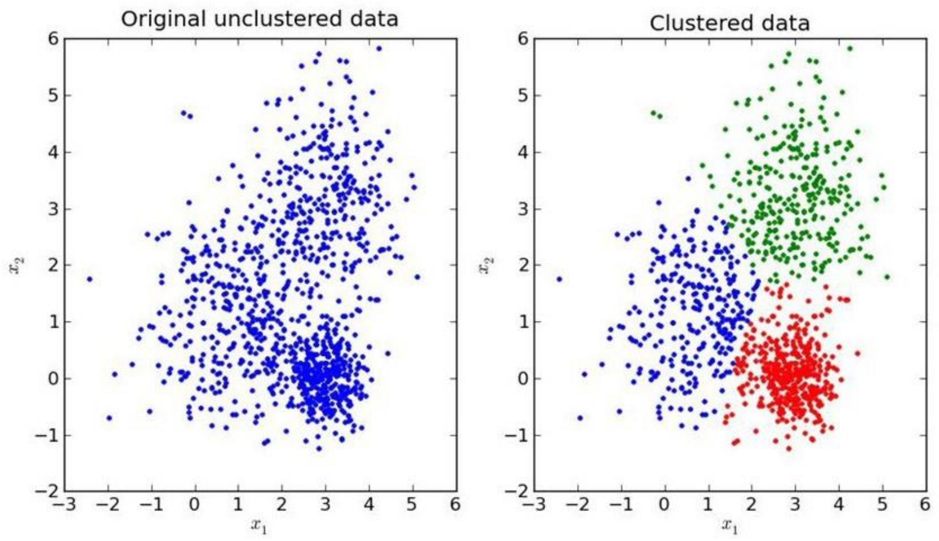
How can data pre-processing be performed ?

- **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.
- **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn’t be performed if there are a lot of missing values in the dataset.
- **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.
- **Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.
- **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0]

3. Researching the model that will be best for the type of data

- ➢ Supervised Learning – Train Me!
- ➢ Unsupervised Learning – I am self sufficient in learning
- ➢ Reinforcement Learning – My life My rules! (Hit & Trial)

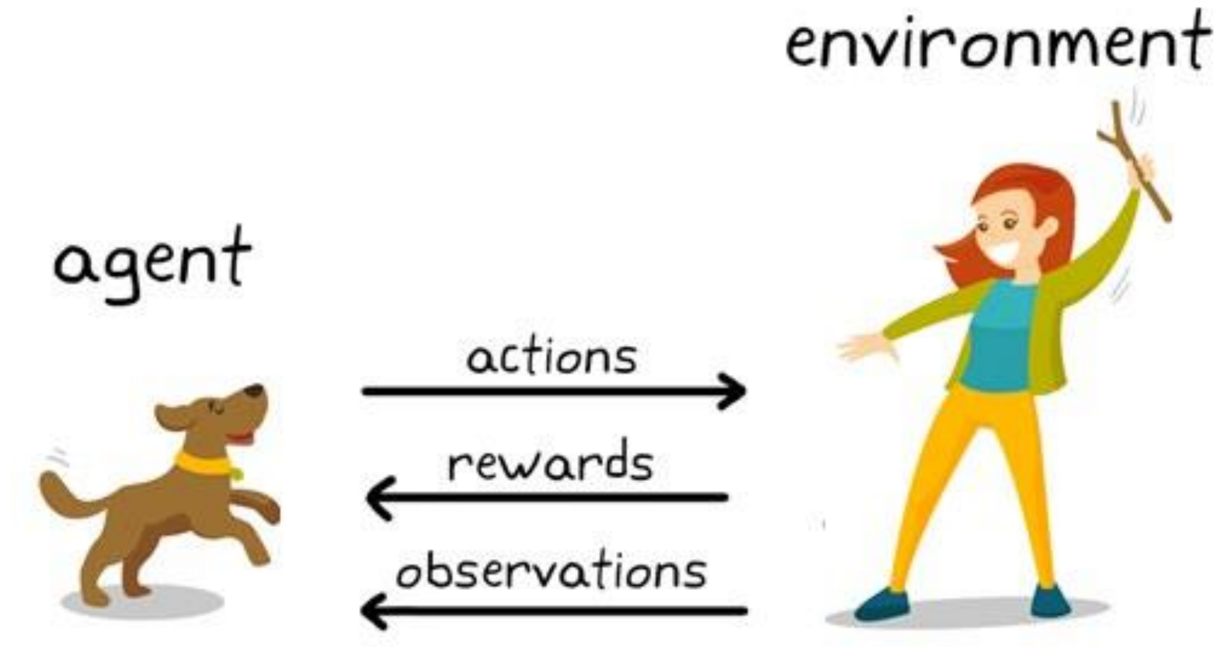




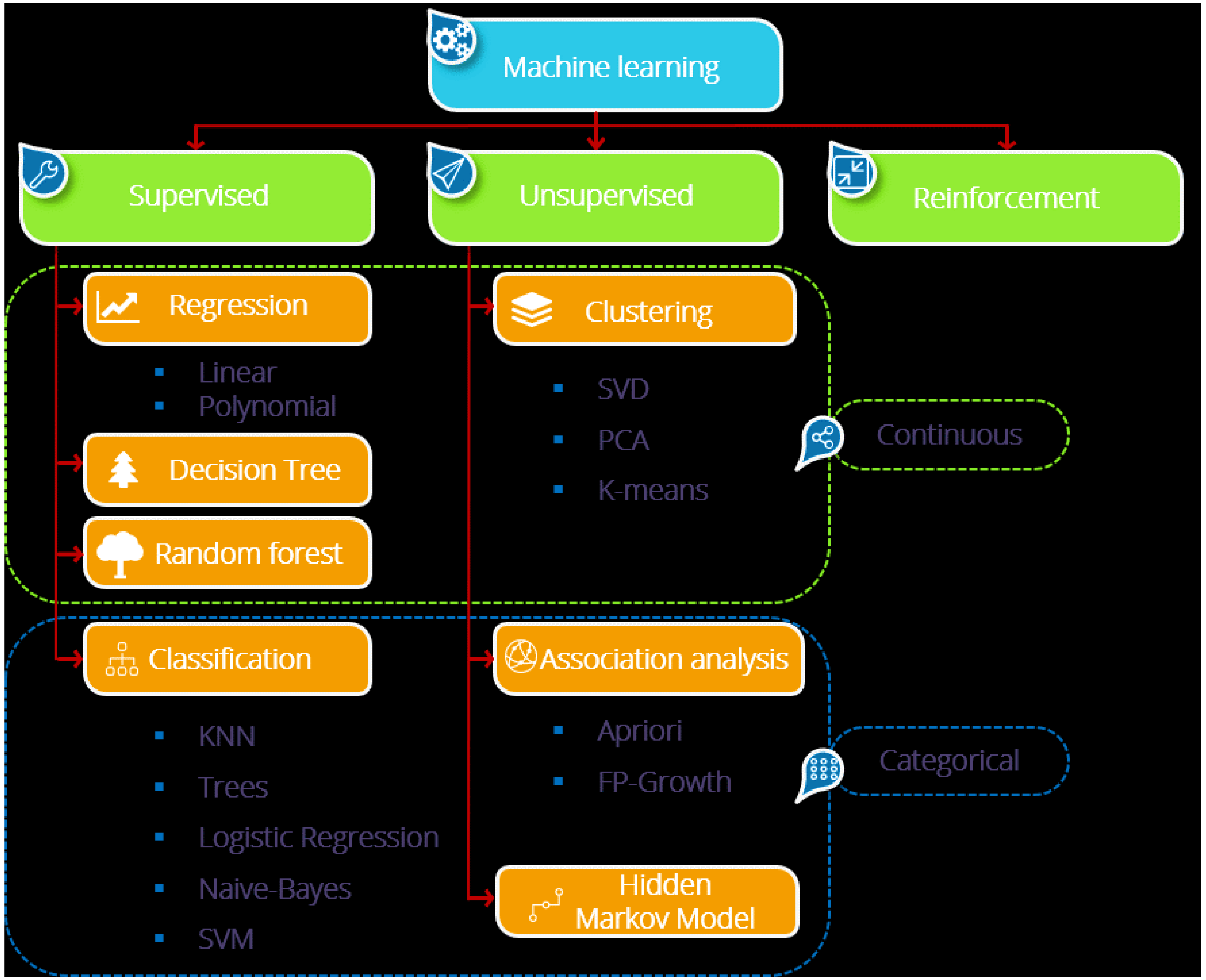
Unsupervised Learning

Reinforcement Learning?

- Reinforcement learning is a type of machine learning in which a computer learns to perform a task through repeated trial-and-error interactions with a dynamic environment.
- This learning approach enables the computer to make a series of decisions that maximize a reward metric for the task without human intervention and without being explicitly programmed to achieve the task



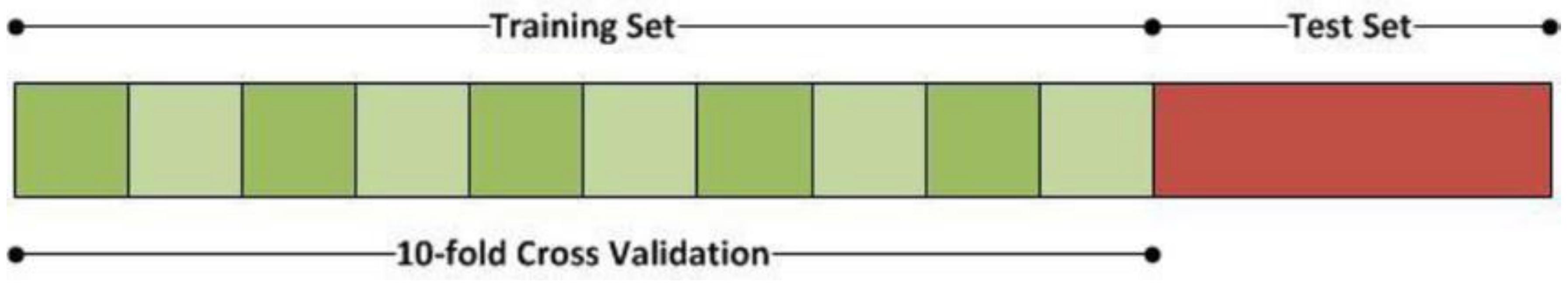
Reinforcement learning in dog training

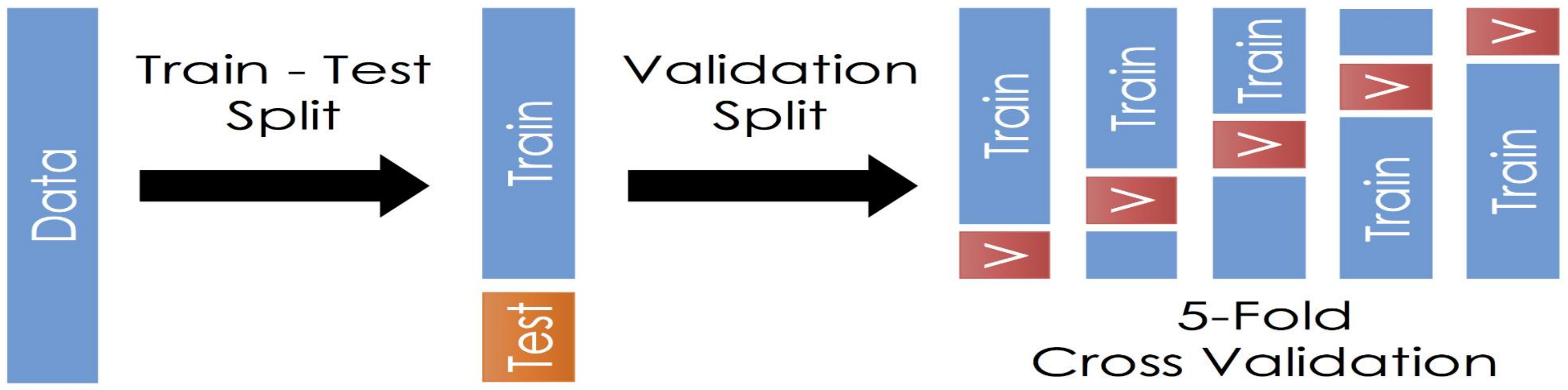

Machine learning
Supervised
Unsupervised
Reinforcement
Regression
Linear
Polynomial
Decision Tree
Random forest
Clustering
SVD
PCA
K-means
Continuous
Classification
KNN
Trees
Logistic Regression
Naive-Bayes
SVM
Association analysis
Apriori
FP-Growth
Hidden Markov Model
Categorical

4. Training and testing the model on data

For training a model we initially split the model into 3 three sections which are ‘**Training data**’ ,‘**Validation data**’ and ‘**Testing data**’.

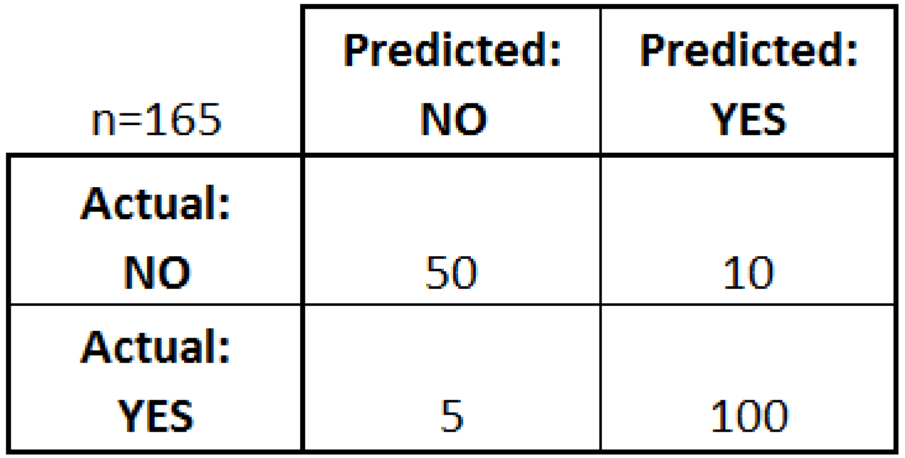
- train the classifier using ‘**training data set**’,
- tune the parameters using ‘**validation set**’ and
- then test the performance of your classifier on unseen ‘**test data set**’.





4. Training and testing the model on data Contd…

Develop a confusion matrix



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

A confusion matrix has 4 parameters, which are '**True positives'**, **'True Negatives'**, **'False Positives'** and '**False Negative'**.

Accuracy = (True Positives +True Negatives) / (Total number of classes)

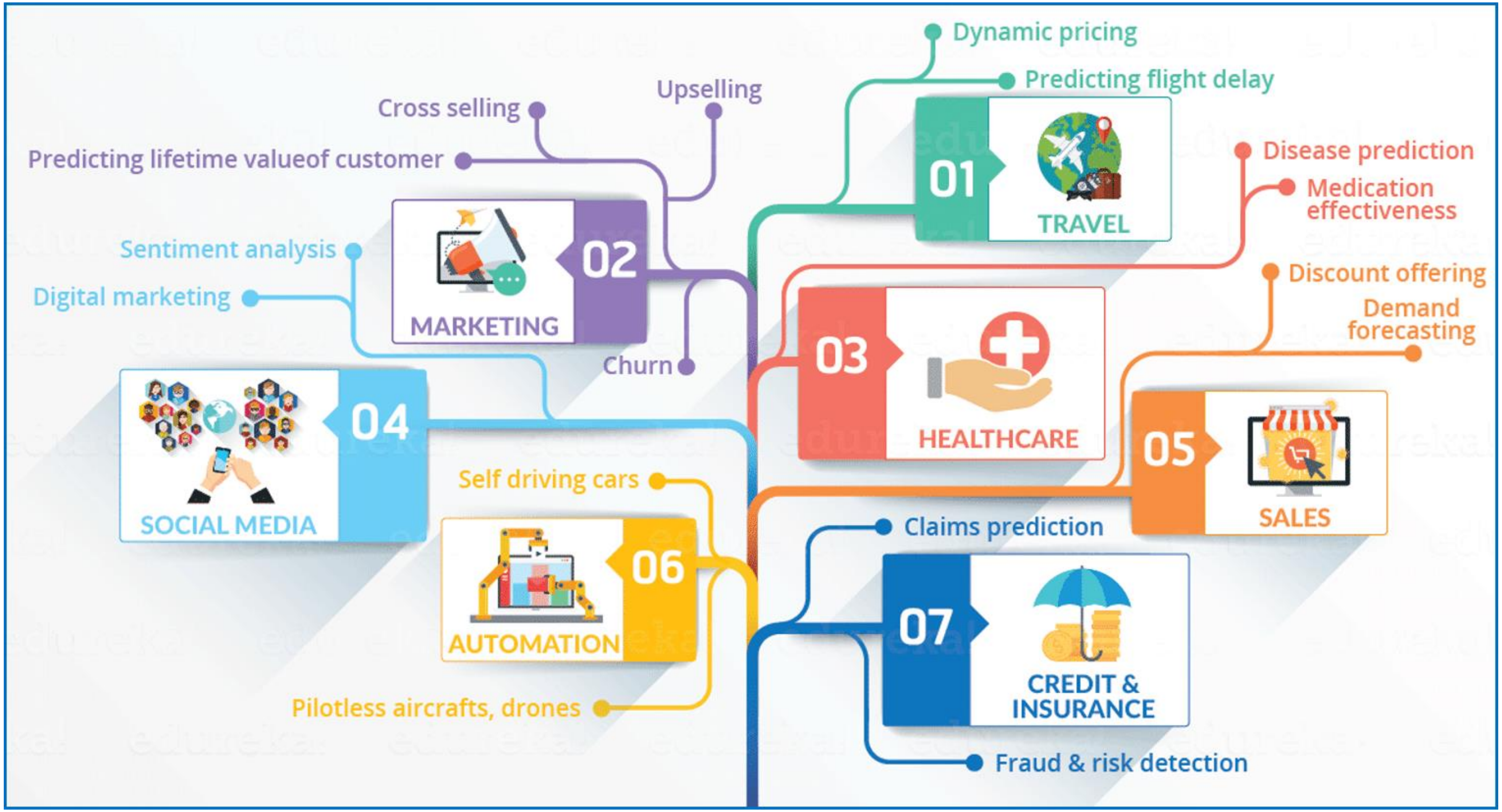
Example:

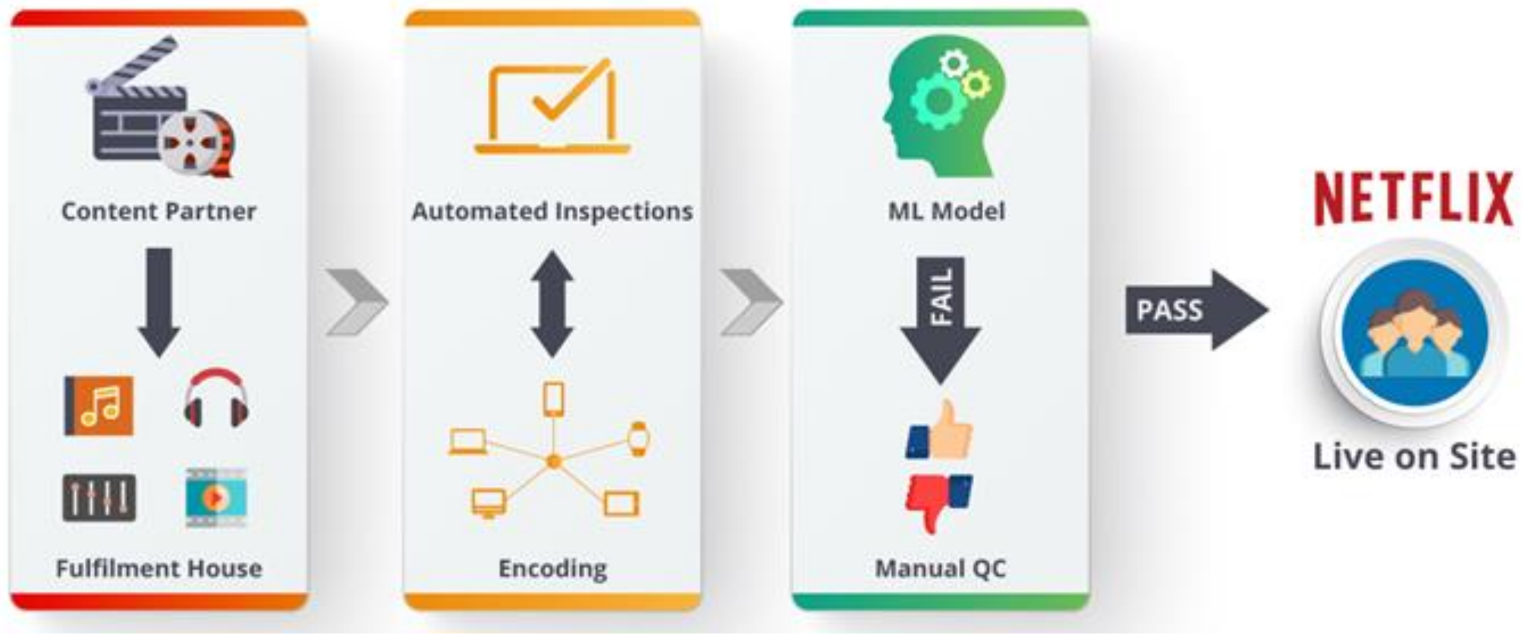
Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

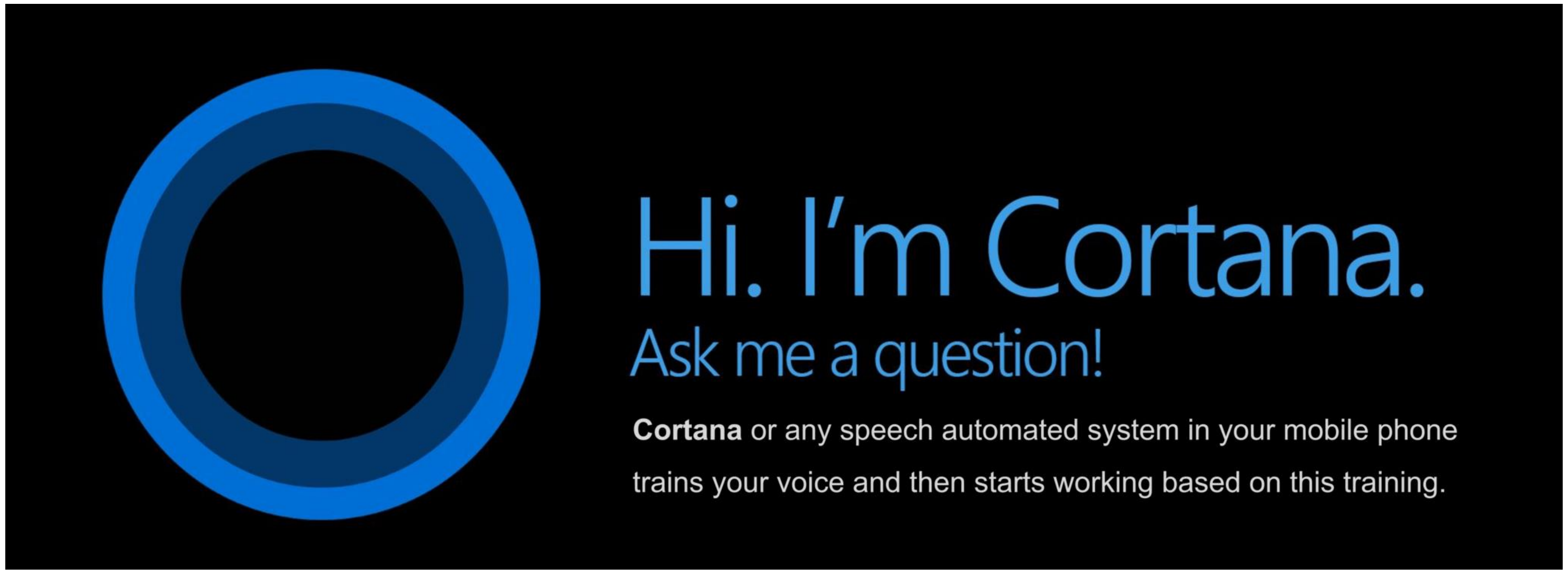
5. Evaluation

- Model Evaluation is an integral part of the model development process.
- It helps to find the best model that represents our data and how well the chosen model will work in the future.

Machine Learning Use Case







Cortana or any speech automated system in your mobile phone trains your voice and then starts working based on this training.

Advanced Topics in Artificial Intelligence

UNIT- VI

INTRODUCTION

- As we have noted, a glimpse into the natural world reveals that even a small child is able to do numerous tasks at once.

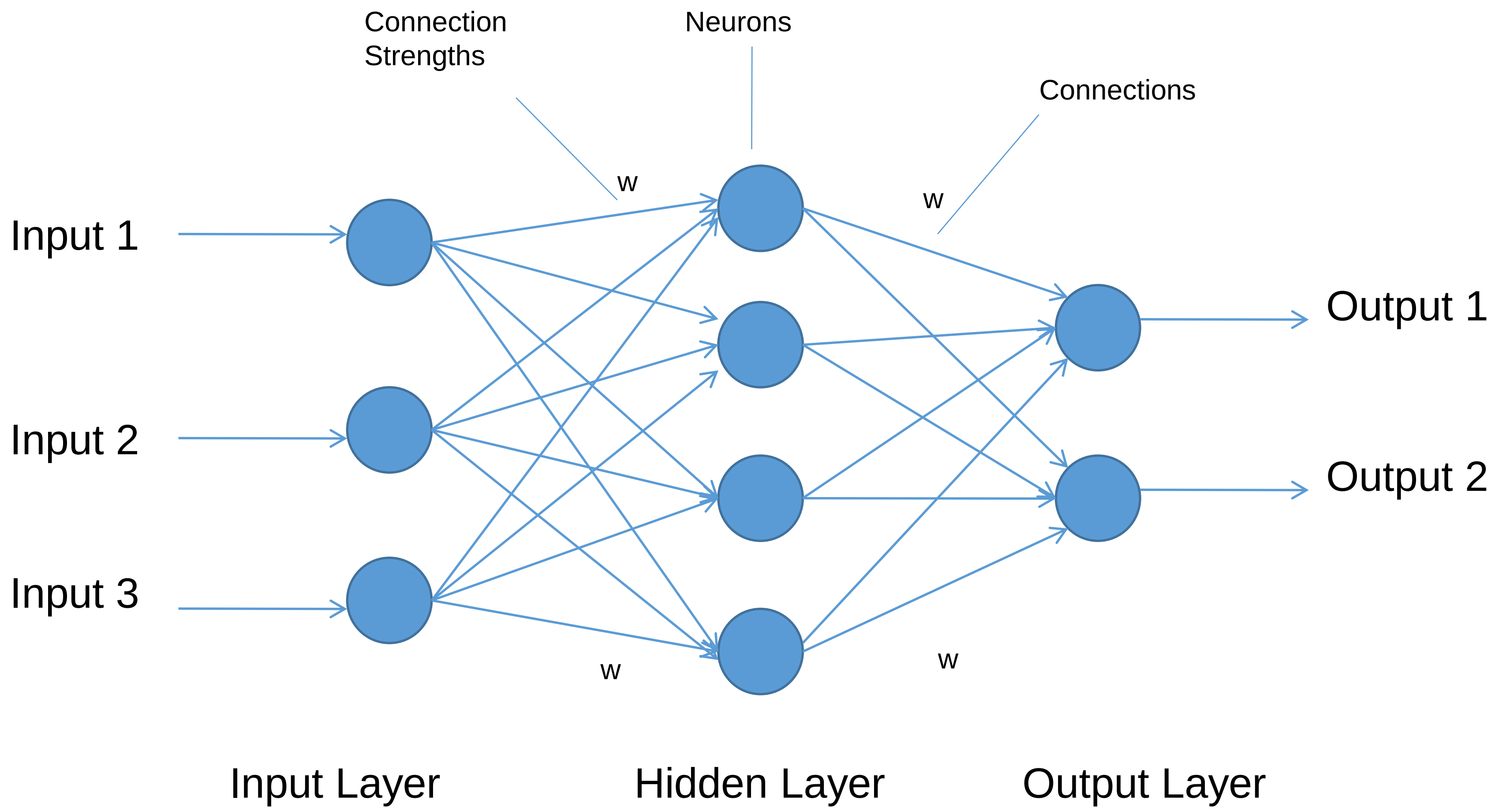
- The example of a child walking, probably the first time that child sees an obstacle, he/she may not know what to do. But afterward, whenever he/she meets obstacles, she simply takes another route.

- It is natural for people to both appreciate the observations and learn from them. An intensive study by people coming from multidisciplinary fields marked the evolution of what we call the artificial neural network (ANN).

ANNs- How They Operate

- ANNs represent a highly connected network of neurons - the basic processing unit
- They operate in a highly parallel manner.
- Each neuron does some amount of information processing.
- It derives inputs from some other neuron and in return gives its output to other neuron for further processing.
- This layer-by-layer processing of the information results in great computational capability.
- As a result of this parallel processing, ANNs are able to achieve great results when applied to real-life problems.

ANN Architecture



There are three fundamental classes of ANN architectures:

- Single layer feed forward architecture
- Multilayer feed forward architecture
- Recurrent networks architecture

Historical Note

- Warren McCulloch and mathematical genius Walter Pitts gave **McCulloch-Pitts theory of formal neural networks**.
- In 1949, Donald Hebb further extended the work in this field, when he described how neural pathways are strengthened each time they are used
- In 1954, Marvin Minsky presented his thesis "**Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem**" and also wrote a paper titled "**Steps Toward Artificial Intelligence**."
- Later John von Neumann invented the von Neumann machine.
- 1958, Frank Rosenblatt, a neurobiologist, proposed the perceptron, which is believed to be the **first physical ANN**.
- Between 1959 and 1960, Bernard Wildrow and Marcian Hoff developed the Adaptive Linear Elements (ADALINE) and the Multiple Adaptive Linear Elements (MADELINE) models.
- In 1986, Rumelhart, Hinton and Williams proposed the back propagation algorithm.

The Biological Neuron

- The entire human brain consists of small interconnected processing units called *neurons and are* connected to each other by nerve fibers

- The interneuron information communication makes it possible for multilevel hierarchical information processing, which gives the brain all its problem-solving power.

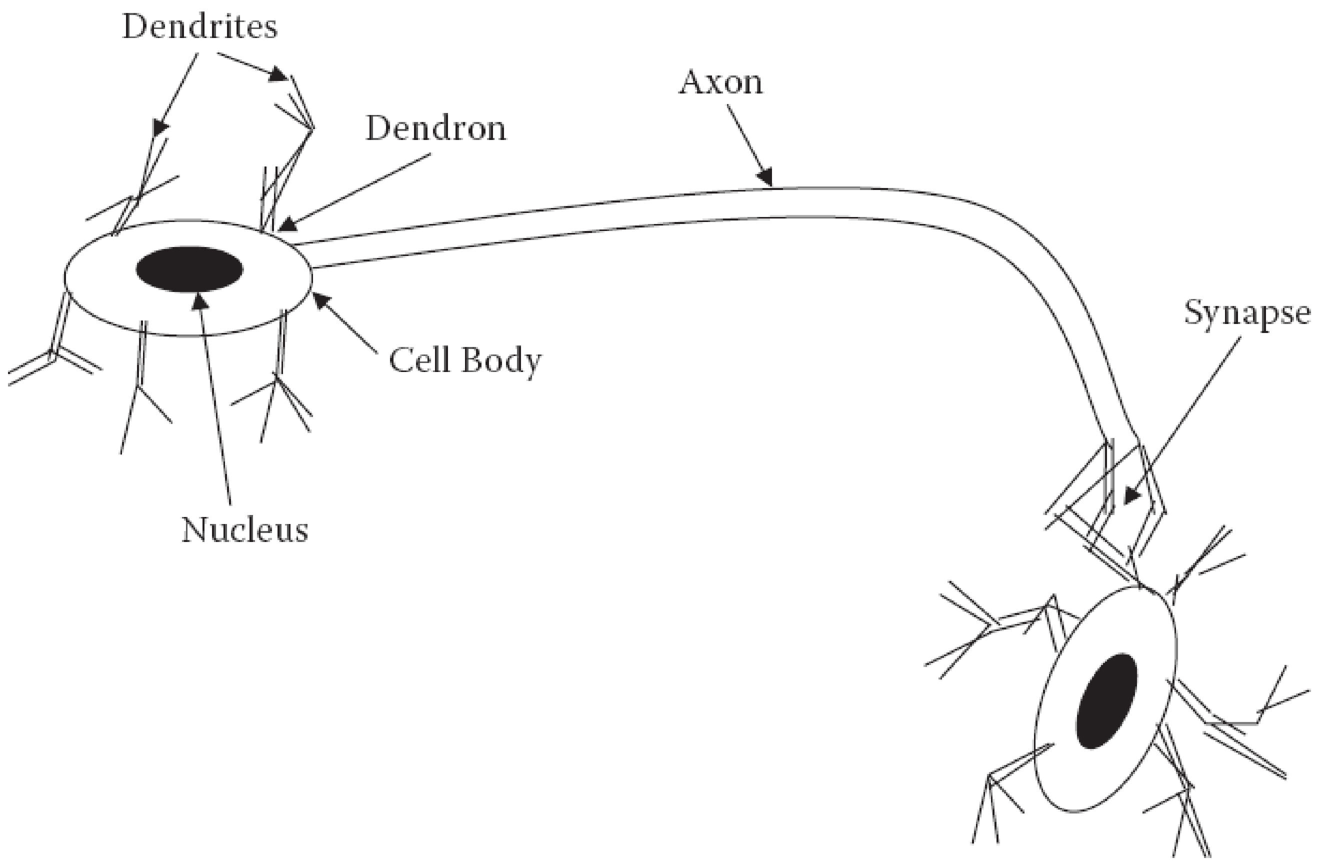
- Each neuron is provided with many inputs, each of which comes from other neurons. Each neuron takes a weighted average of the various inputs presented to it.

- The weighted average is then made to pass over a nonlinear inhibiting function that limits the neuron’s output. The nonlinearity in biological neurons is provided by the presence of potassium ions within each neuron cell and sodium ions outside the cell membrane.

- The difference in concentrations of these two elements causes an electrical potential difference, which in turn causes a current to flow from outside the cell to inside the cell. This is how the neuron takes its inputs

Structural Components of a Neuron

- A neuron has four main structural components - the *dendrites*, the *cell body*, the *axon*, and the *synapse.*



Structural Components of a Neuron

- Dendrites are responsible for receiving the signals from other neurons.
- These signals are passed through a small, thick fiber called a *dendron.*
- *The received signals collected at* different dendrites are processed within the cell body, and the resulting signal is transferred through a long fiber called the *axon.*
- *At the other end of the axon exists an inhibiting unit called a synapse,* which controls the flow of neuronal current from the originating neuron to the receiving dendrites of neighboring neurons.

Biological Neuron	Artificial Neuron
Dendrites	Input
Cell Nucleus(Soma)	Node
Axon	Output
Synapse	Interconnections

The Artificial Neuron

- The neural network, by its simulating a biological neural network, is a novel computer architecture and a novel algorithmization architecture relative to conventional computers.

- It allows using very simple computational operations (additions, multiplication, and fundamental logic elements) to solve complex, mathematically ill-defined problems, nonlinear problems, or stochastic problems.

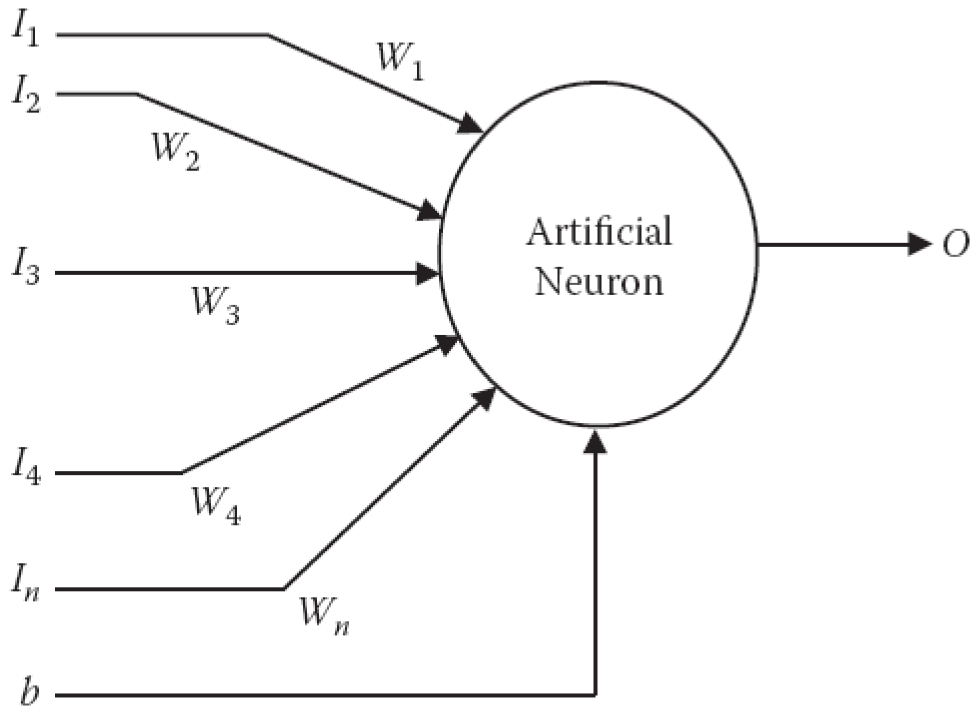
- The artificial neuron is the most basic computational unit of information processing in ANNs.

- Each neuron takes information from a set of neurons, processes it, and gives the result to another neuron for further processing.

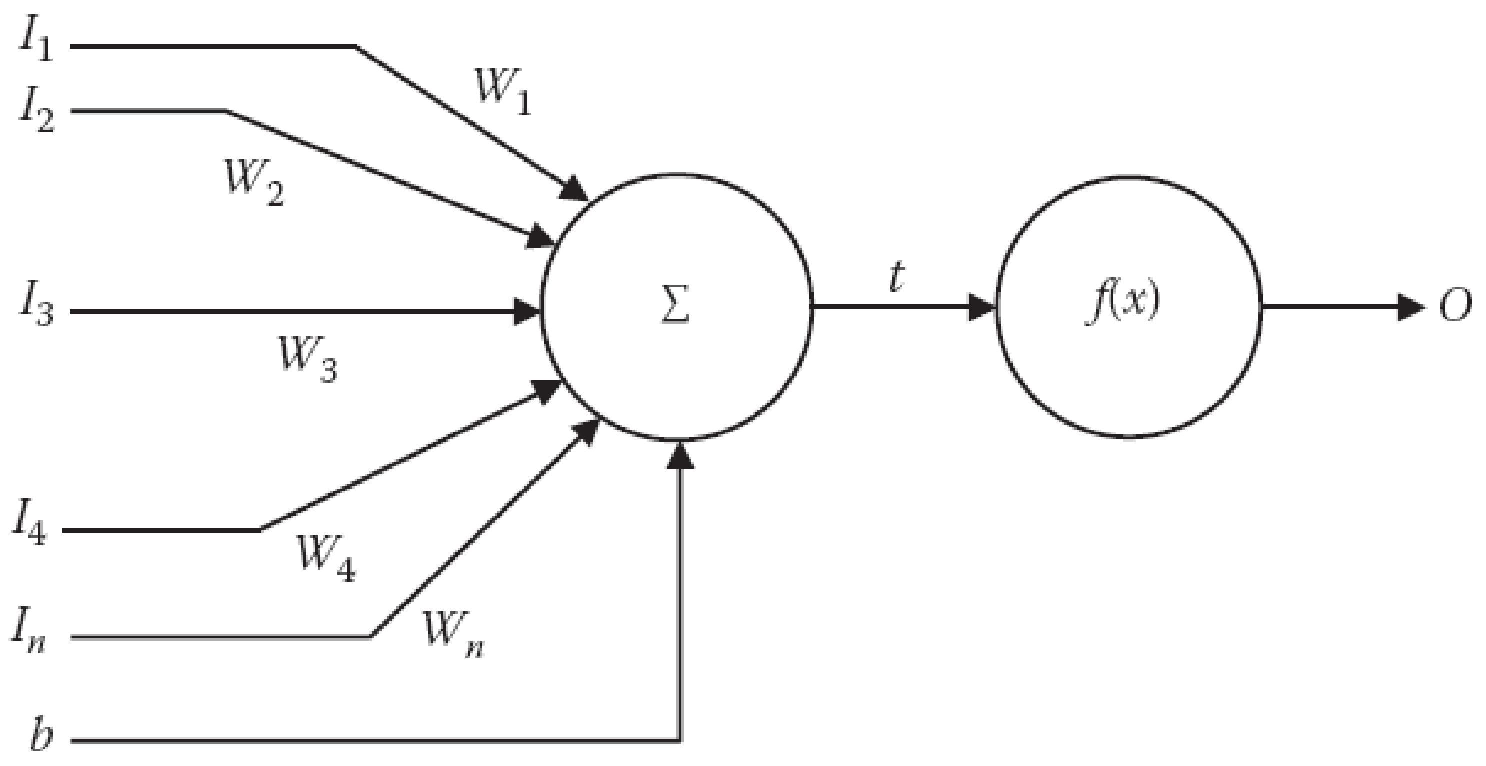
- These neurons are a direct result of the study of the human brain and attempts to imitate the biological neuron.

Structure of an Artificial Neuron

- The neuron consists of a number of inputs. The information is given as inputs via input connections, each of which has some weight associated with it.
- An additional input, known as *bias, is given to* the artificial neuron.



Structure of an Artificial Neuron (continued)



the inputs are marked *I1, I2, I3, … , In; the weights associated with each Connection* are given by *W1, W2, W3, … , Wn; b denotes the bias; and the output is denoted by O. Because there is* one weight for every input, the number of inputs is equal to the number of weights in a neuron.

The Processing of the Neuron

- **Functionality of neuron that is performed can be broken down into two steps**. The **first is the weighted summation**, and the second is the **activation function**. The two steps are applied one after the other, as shown in the previous slide.

- The function of the summation block is given by Equation

$$t = \sum_{i=1}^{n} W_i * I_i + b$$

- The summation forms the input to the next block. **This is the block of the activation function, where the input is made to pass through a function called the *activation function*.**

The Activation Function

- The activation function performs several important tasks
- One of the more important of which is to introduce nonlinearity to the network.
- Another important feature of the activation function is its ability to limit the neuron's output.
- The complete mathematical function of the artificial neuron can be given as:

$$O = f(t)$$

where *f is any activation function*

$$O = f\left(\sum_{i=1}^{n} W_i * I_i + b\right)$$

The Perceptron

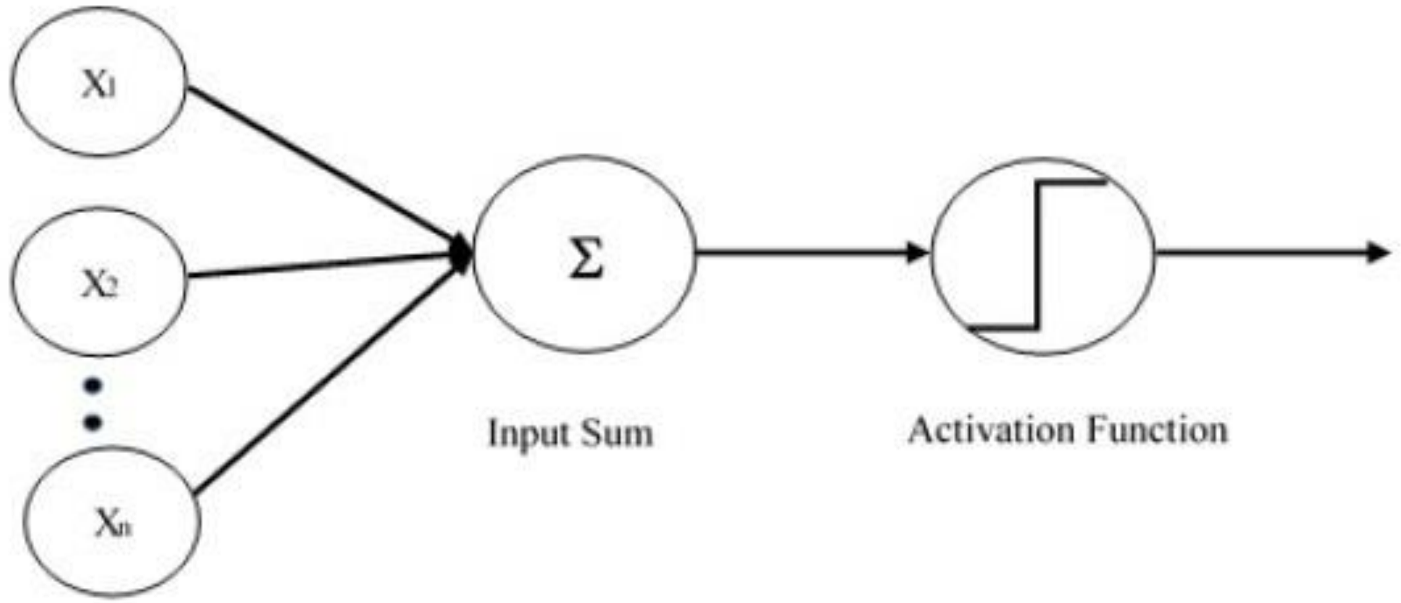
- The perceptron is the most basic model of the ANN. It consists of a single neuron. The perceptron may be seen as a binary classifier that maps every input to an output that is either 0 or 1.

- The perceptron is given by the function represented by Equation

$$f(x) = \begin{matrix} 1 & \text{if } w.x + b > 0 \\ 0 & \text{otherwise} \end{matrix}$$

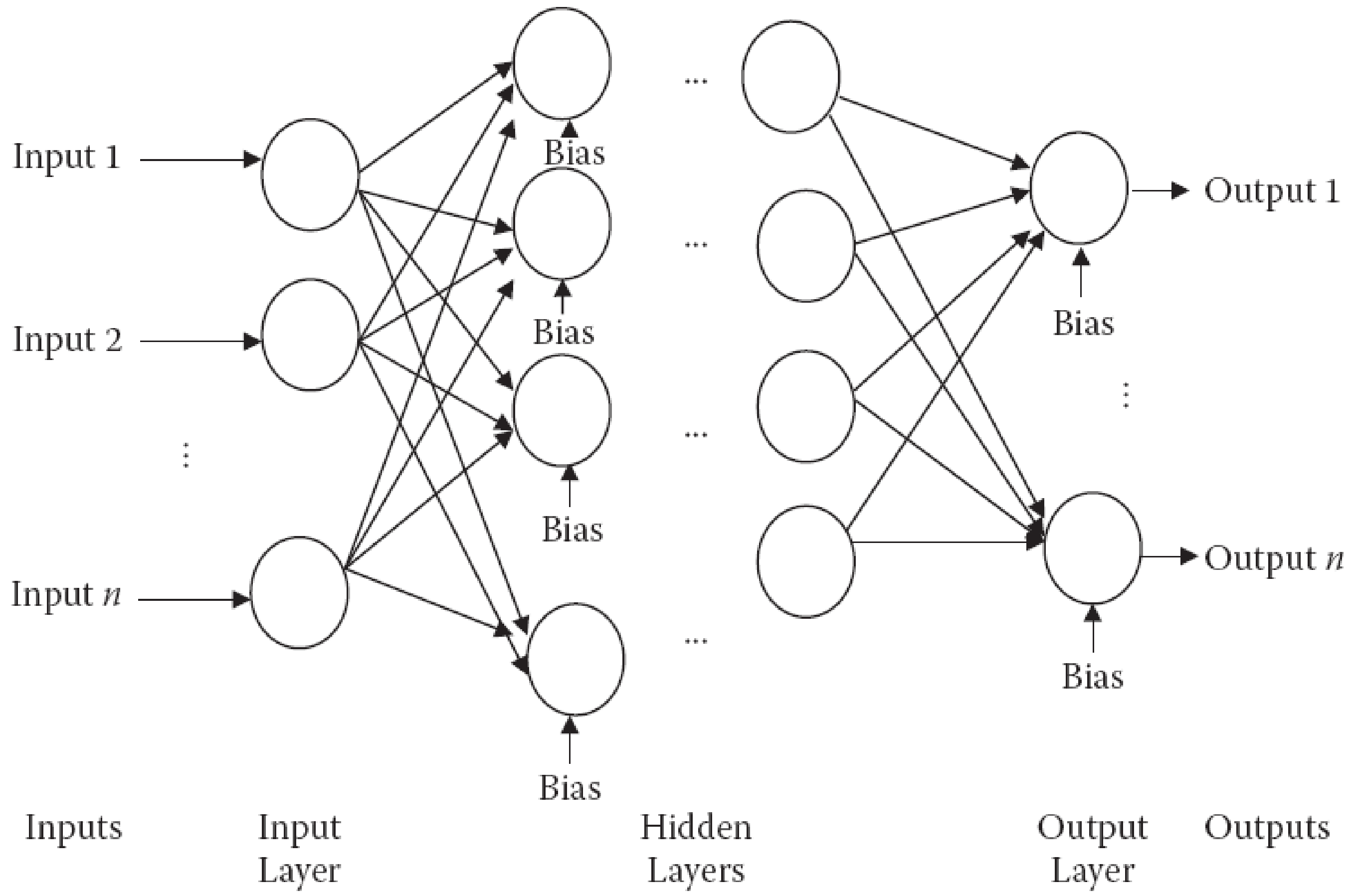
- where *w is the vector of real-world weights, x is the input, "." is the dot product of the two vectors,* and *b is the bias.*

- The perceptron has learning capabilities in that it can learn from the inputs to adapt itself. As a result, the perceptron is able to learn historical data.


X_1
X_2
X_n
Σ
Input Sum
Activation Function

Multilayer Perceptron

- A multilayer perceptron is a class of feedforward artificial neural network.
- Multilayer perceptrons are networks used to overcome the linear separability limitation of the perceptrons.



Multilayer Perceptron

- An MLP consists of
  1. An input layer,
  2. At least one intermediate or “hidden” layer,
  3. And one output layer

- The neurons from each layer being fully connected (in some particular applications, partially connected) to the neurons from the next layer.

- The number of neurons in the input layer is equal to the number of inputs of the system. Each neuron corresponds to one input.

- In output layer, the number of neurons is equal to the number of outputs that the system is supposed to provide.

- The number of layers decides the computational complexity of the ANN.

Example

- Calculate the output for a neuron. The inputs are (2, 3), and the corresponding weights are (0, 1). Bias is given to be 4. The activation function is **sigmoid** . Also draw the neuron architecture.

Solution:

Using Equation, we have

$$O = f\left(\sum_{i=1}^{n} W_i * I_i + b\right)$$

where *x1 = 2, w1 = 0, x2 = 3 , w2 = 1 and b = 4.*

$$\phi(z) = \frac{1}{1+e^{-z}}$$

Hence the neuron's output is

$$\begin{aligned}(w \cdot x) + b &= ((w_1 * x_1) + (w_2 * x_2)) + b \\ &= 0 * 2 + 1 * 3 + 4 \\ &= 7\end{aligned}$$

$$y = f(w \cdot x + b) = f(7) = \boxed{0.999}$$

Stopping Condition

- The algorithm stops according to the stopping condition. Normally one or more of the following criteria are used as stopping conditions:

  • **Time:** The algorithm may be stopped when the time taken to execute exceeds more than a threshold.

  • **Epoch**: The algorithm has a specified maximum number of epochs. Upon exceeding this number, the algorithm may be stopped

  • **Goal**: The algorithm may be stopped if the error measured by the system reduces to more than a specific value. It may not be useful to continue training after this point.

  • **Validating data:** If the error on validation data starts increasing, even if there is a decrease in the testing data, it would be better to stop further training.

  • **Gradient:** Gradient refers to the improvement of the performance or the lowering of the error in between epochs.

Activation Functions

- One of the more important of which is to introduce nonlinearity to the network.
- Another important feature of the activation function is its ability to limit the neuron's output.

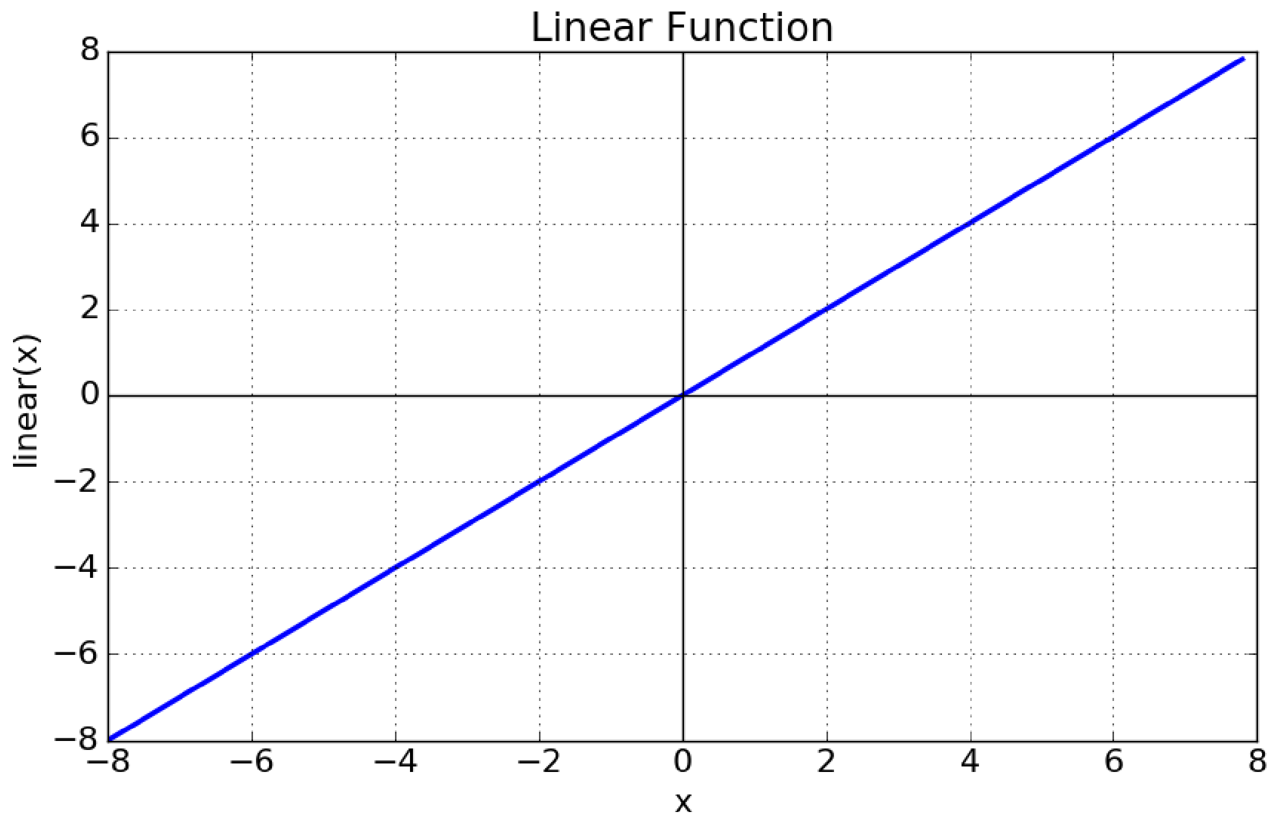
The Activation Functions can be basically divided into 2 types-

1. Linear Activation Function
2. Non-linear Activation Functions

Linear or Identity Activation Function

Equation : f(x) = x

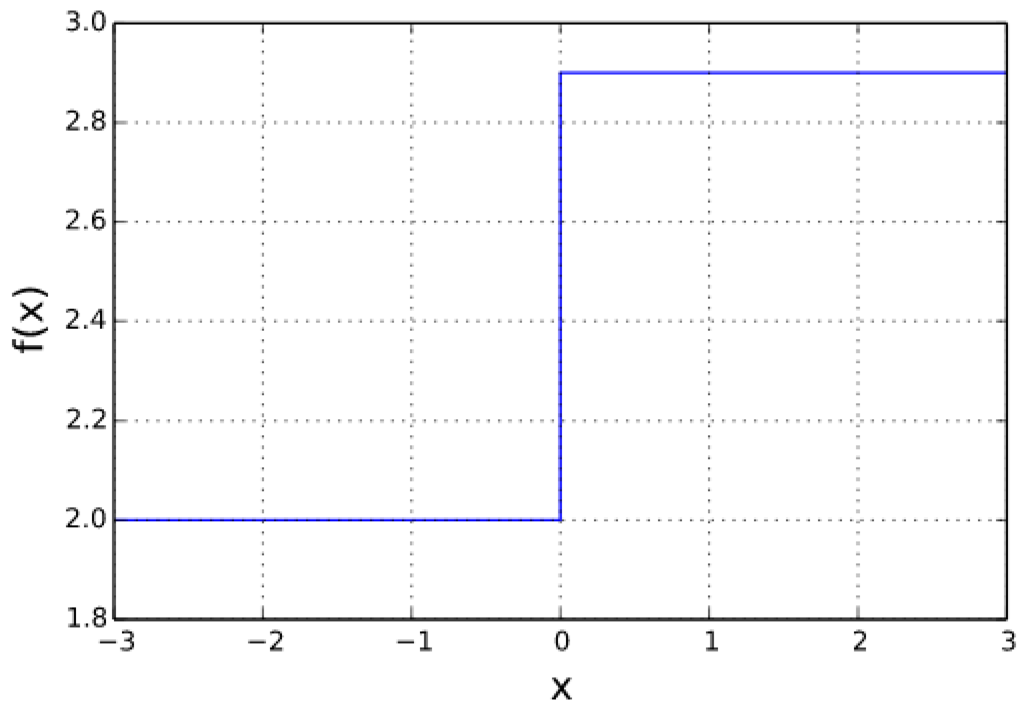
Range : (-infinity to infinity)



Binary step

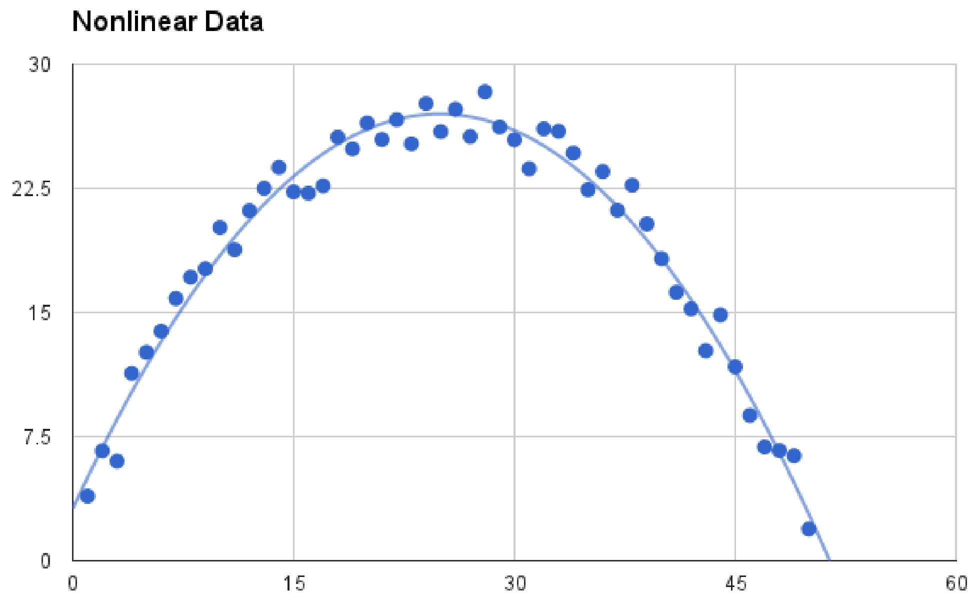
- A binary step function is generally used in the Perceptron linear classifier. It thresholds the input values to 1 and 0, if they are greater or less than zero, respectively.
- This activation function is useful when the input pattern can only belong to one or two groups, that is, binary classification.

$$a_j^i = f(z_j^i) = \begin{cases} 0 & \text{if } z_j^i < 0 \\ 1 & \text{if } z_j^i > 0 \end{cases}$$



Non-linear Activation Function

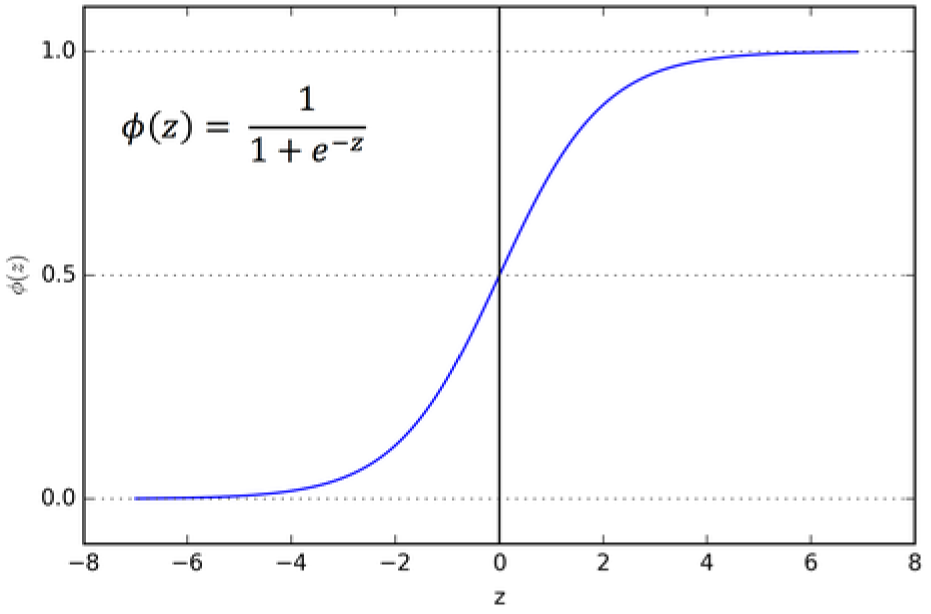
The Nonlinear Activation Functions are the most used activation functions. Nonlinearity helps to makes the graph look something like this



The main terminologies needed to understand for nonlinear functions are:
Derivative or Differential: *Change in y-axis w.r.t. change in x-axis.It is also known as slope.*
Monotonic function: *A function which is either entirely non-increasing or non-decreasing.*

1. Sigmoid or Logistic Activation Function

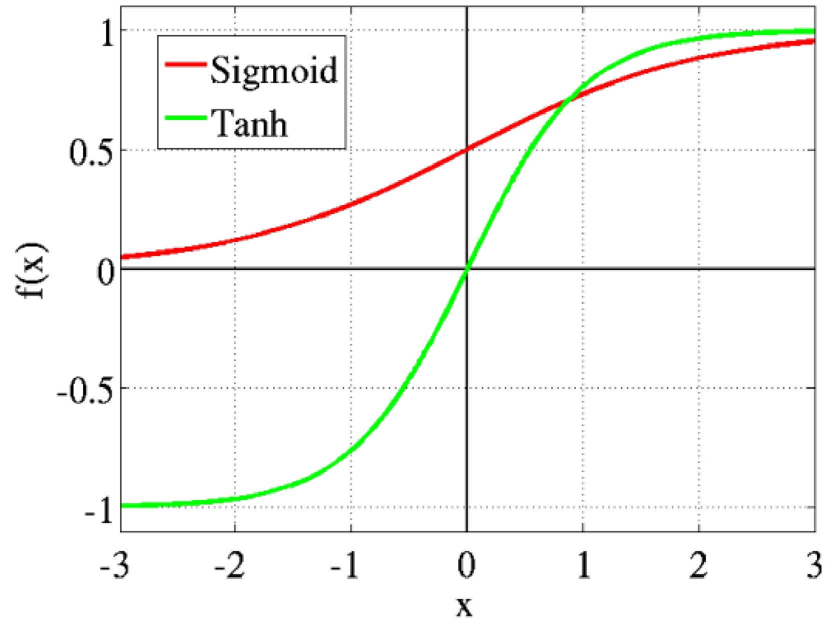
The Sigmoid Function curve looks like a S-shape.



- The main reason why we use sigmoid function is because it exists between **(0 to 1).** Therefore, it is especially used for models where we have to **predict the probability** as an output. Since probability of anything exists only between the range of **0 and 1,** sigmoid is the right choice.

2. Tanh or hyperbolic tangent Activation Function

tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).



The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.

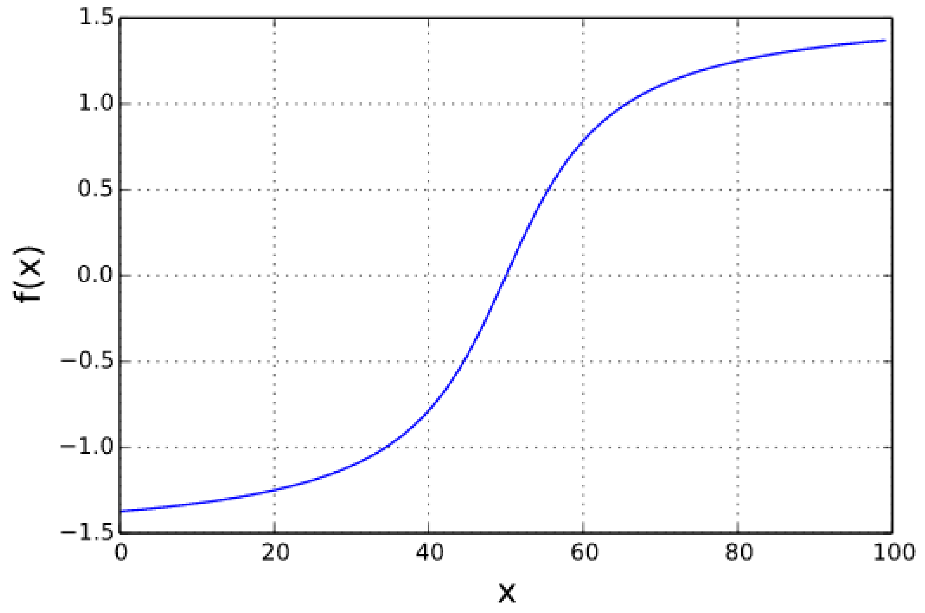
An alternative equation for the tanh activation function is:

$$a_j^i = f(x_j^i) = \frac{2}{1 + \exp(-2x_j^i)} - 1$$

3. ArcTan

$$a_j^i = f(x_j^i) = \tanh^{-1}(x_j^i)$$

This activation function maps the input values in the range $(-\pi/2, \pi/2)$

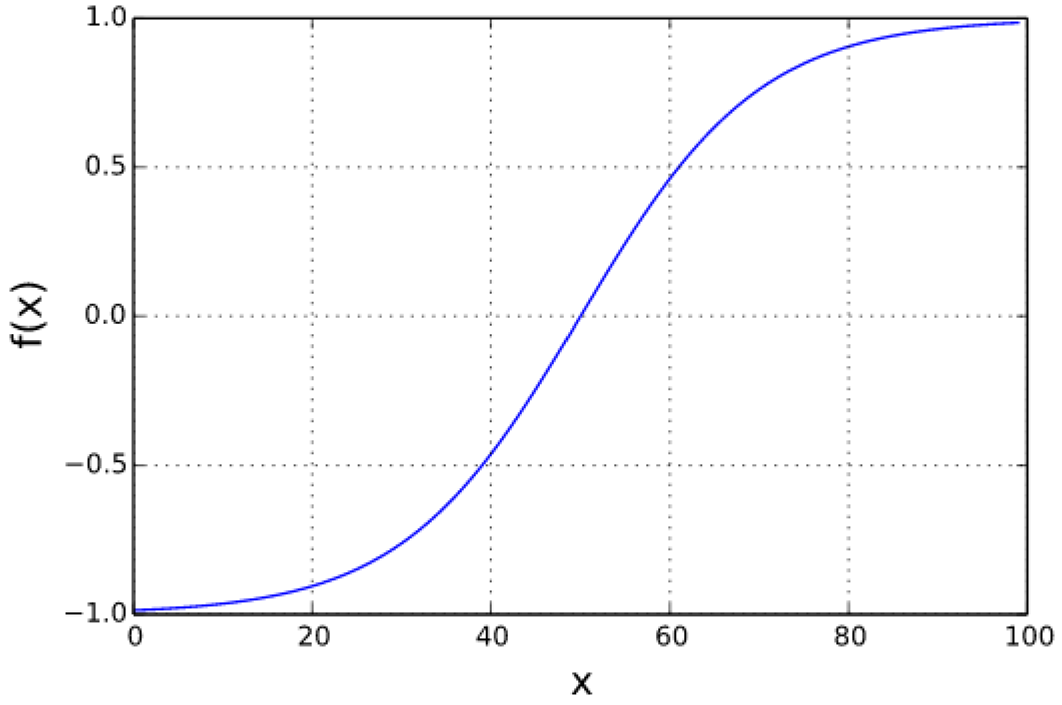


Its graph is slightly flatter than **tanh**, so it has a better tendency to differentiate between similar input values.

4. Bipolar Sigmoid

$$a_j^i = f(x_j^i) = \frac{1 - \exp(-x_j^i)}{1 + \exp(-x_j^i)}$$

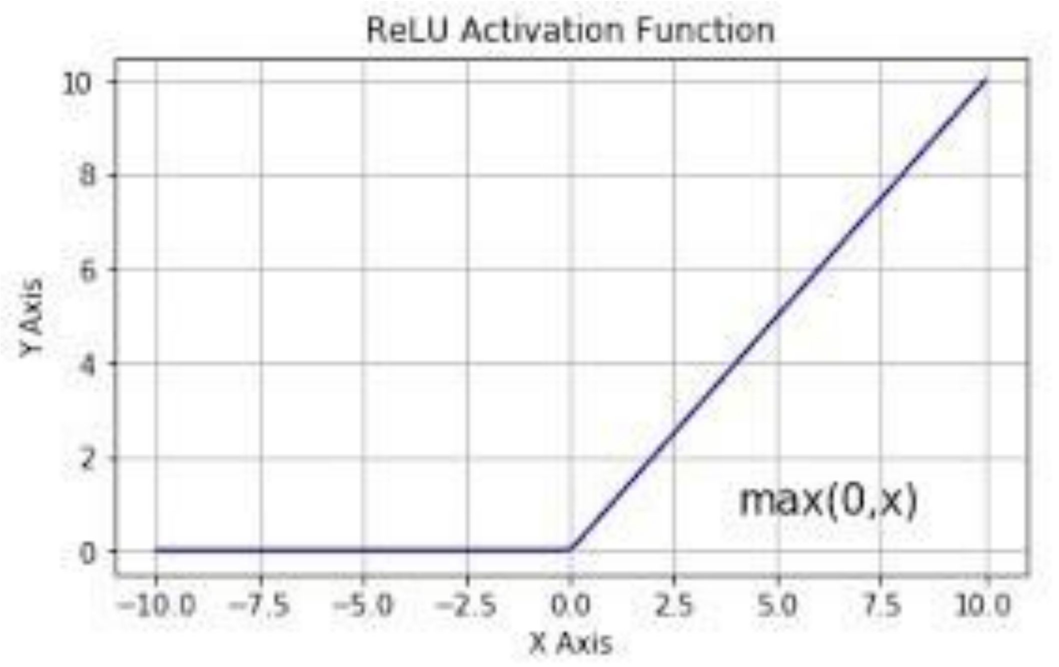
- The sigmoid function can be scaled to have any range of output values, depending upon the problem. When the range is from −1 to 1, it is called a bipolar sigmoid.



5. ReLU (Rectified Linear Unit) Activation Function

The ReLU is the most used activation function in the world right now.Since, it is used in almost all the convolutional neural networks or deep learning.

$$f(x) = \max(0, x) = \begin{cases} x_i \; if \; x_i > 0 \\ 0 \; if \; x_i < 0 \end{cases}$$



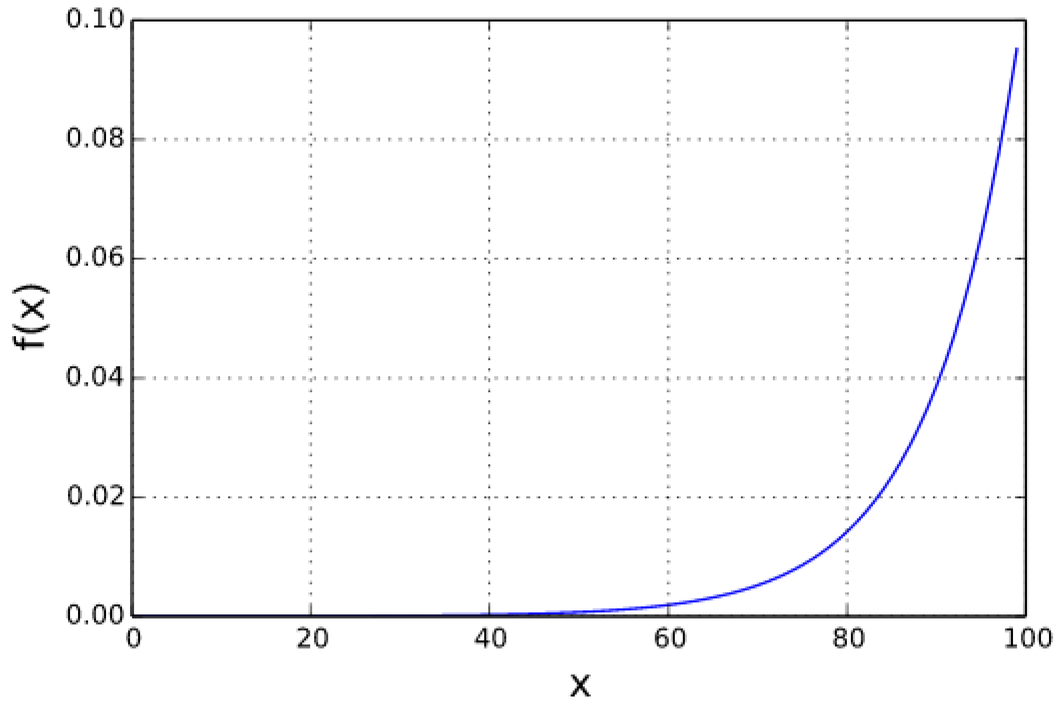
The main idea behind the ReLu activation function is to perform a threshold operation to each input element where values less than zero are set to zero

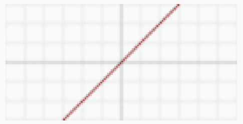
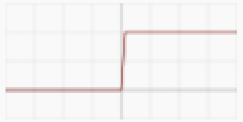
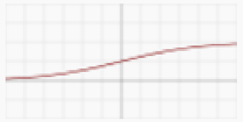
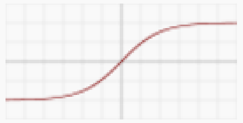
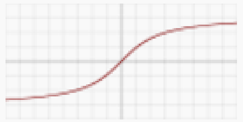
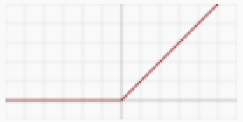
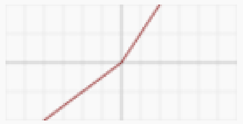
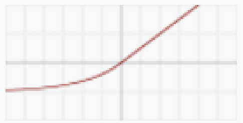
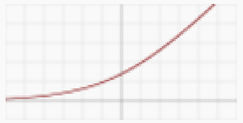
Range: [0 to infinity)
The function and its derivative **both are monotonic**.

6. Softmax

$$a_j^i = f(x_j^i) = \frac{\exp(z_j^i)}{\sum_k \exp(z_k^i)}$$

- The **softmax** function gives us the probabilities that any of the classes are true. It produces values in the range (0,1).
- It also highlights the largest value and tries to suppress values which are below the maximum value; its resulting values always sum to 1.
- This function is widely used in multiple classification logistic regression models.



Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

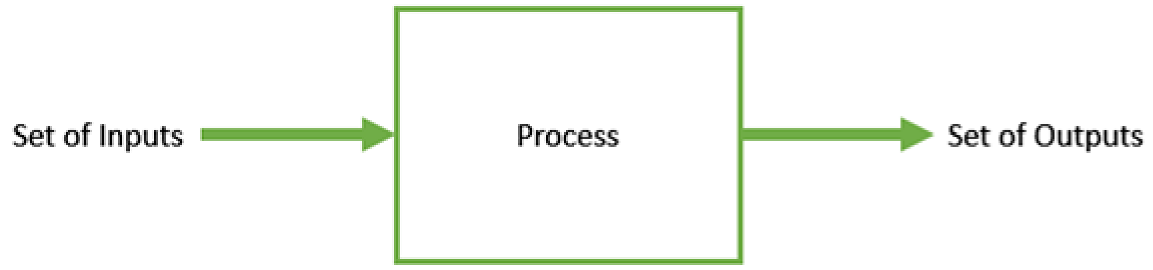
Advanced Topics in Artificial Intelligence

UNIT- VI

Introduction to Genetic Algorithm

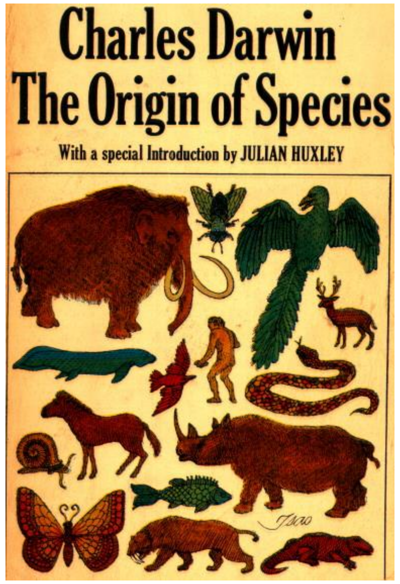
Introduction to Optimization

- Optimization is the process of **making something better**.
- Optimization refers to finding the values of inputs in such a way that we get the “best” output values.
- The definition of “best” varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.



General Introduction to GAs

- Genetic algorithms (GAs) are a technique to solve problems which need optimization.
- GAs are a subclass of Evolutionary Computing and are random search algorithms.
- GAs are based on Darwin’s theory of evolution.
- History of GAs:
  - Evolutionary computing evolved in the 1960s.
  - GAs were created by John Holland in the mid-1970s.



Advantages of GAs

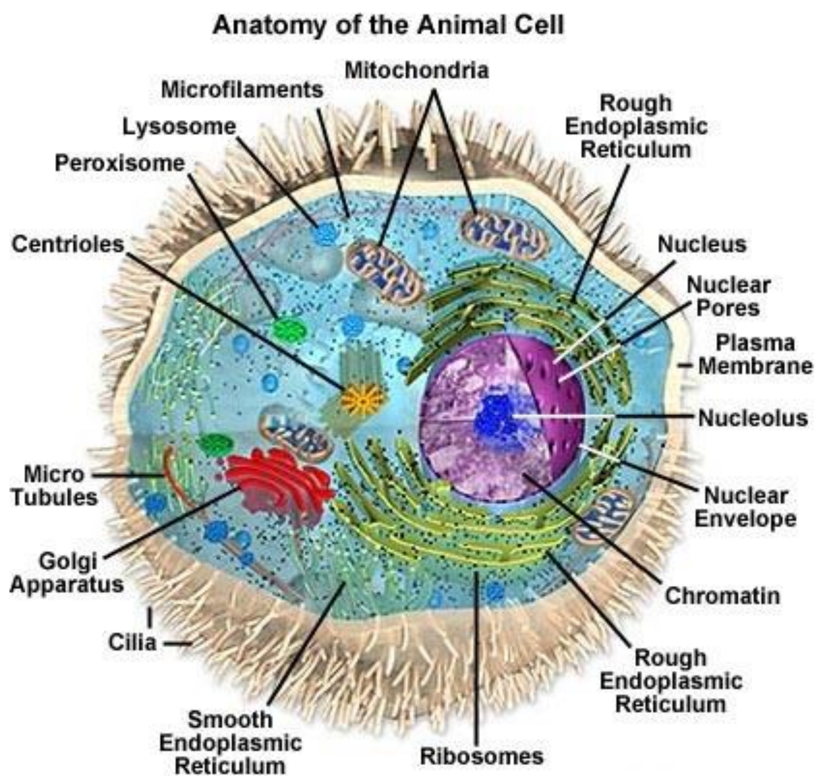
- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Always gets an answer to the problem, which gets better over the time.
- Useful when the search space is very large and there are a large number of parameters involved.

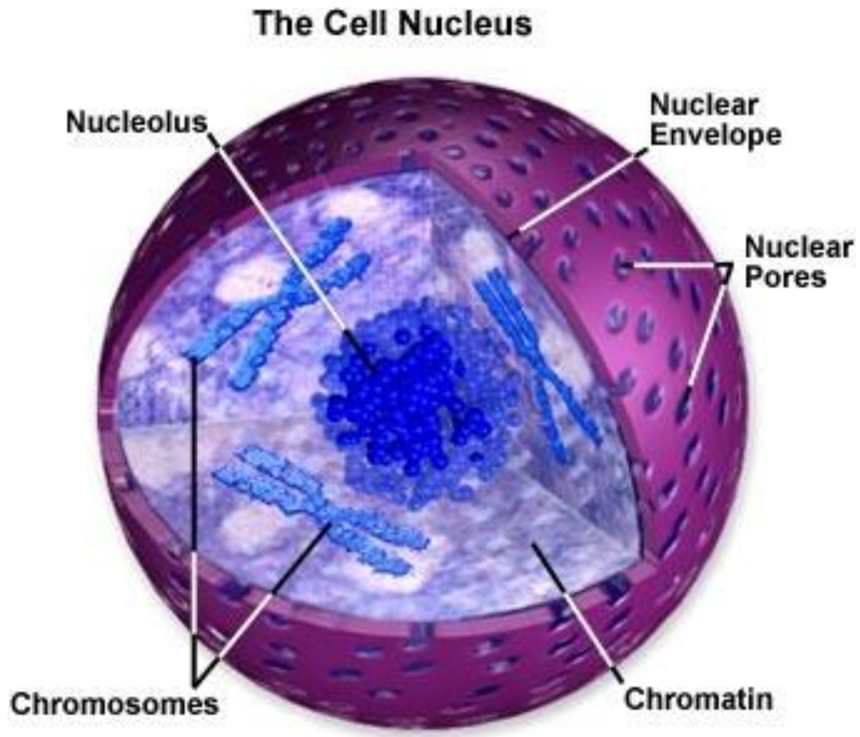
Limitations of GAs

- GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

Biological Background (1) – The Cell

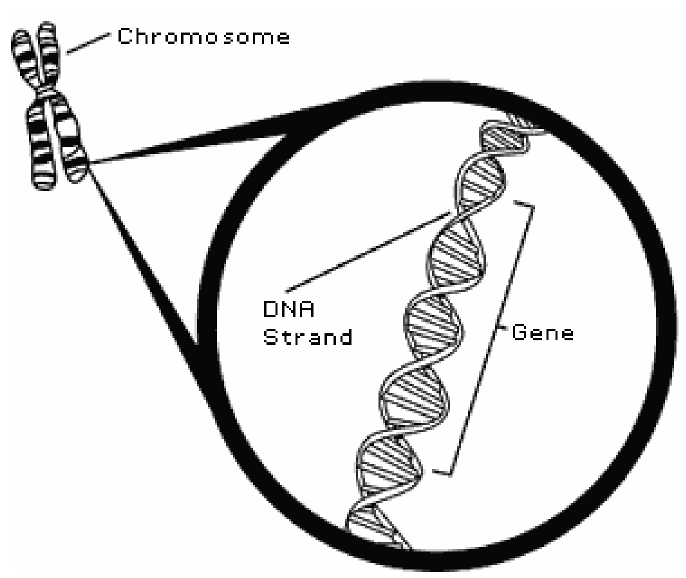
- The center of this all is the cell nucleus.
- The nucleus contains the genetic information.





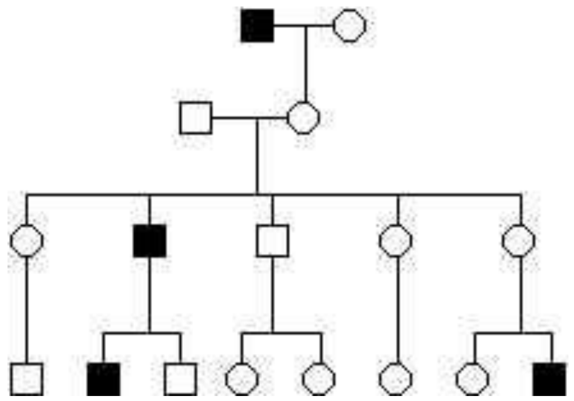
Biological Background (2) – Chromosomes

- Genetic information is stored in the chromosomes.
- Each chromosome is build of DNA (Deoxyribonucleic acid).
- Chromosomes in humans form pairs.
- There are 23 pairs.
- The chromosome is divided in parts: genes.
- Genes code for properties.
- The posibilities of the genes for one property is called: allele.
- Every gene has an unique position on the chromosome: locus.



Biological Background (3) – Genetics

- The entire combination of genes is called genotype.
- A genotype develops into a phenotype.
- Alleles can be either dominant or recessive.
- Dominant alleles will always express from the genotype to the fenotype.
- Recessive alleles can survive in the population for many generations without being expressed.



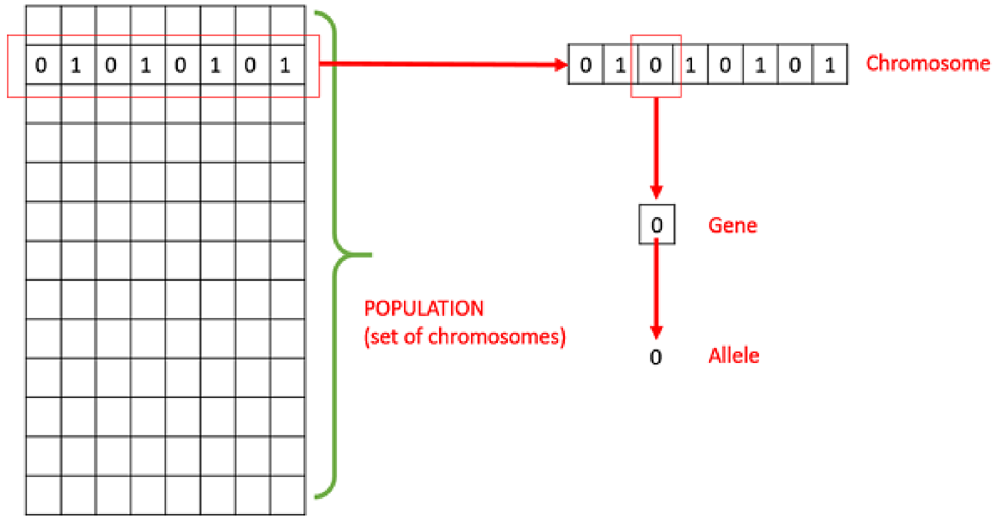
Basic Terminology

Population: It is a subset of all the possible (encoded) solutions to the given problem.

Chromosomes : A chromosome is one such solution to the given problem.

Gene : A gene is one element position of a chromosome.

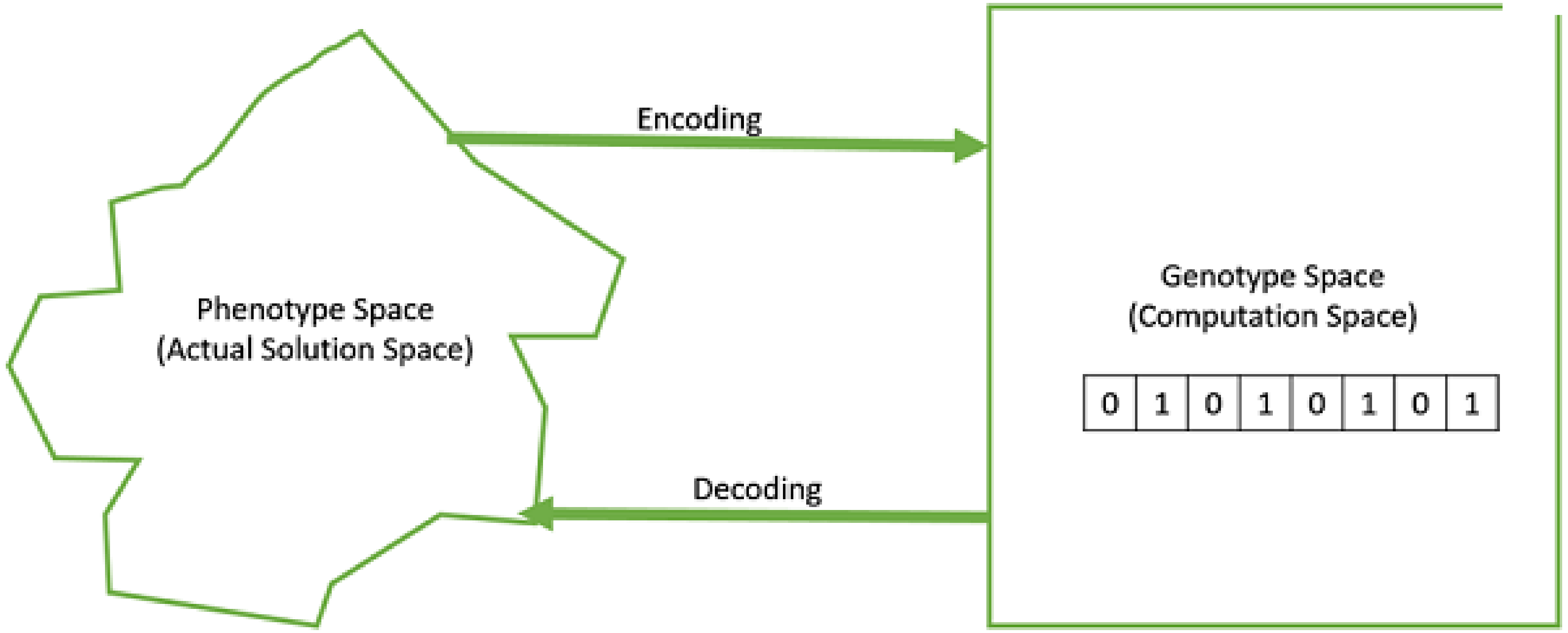
Allele : It is the value a gene takes for a particular chromosome.



Genotype : Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.

Phenotype : Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.

Decoding and Encoding



Fitness Function : A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output.

Genetic Operators : These alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

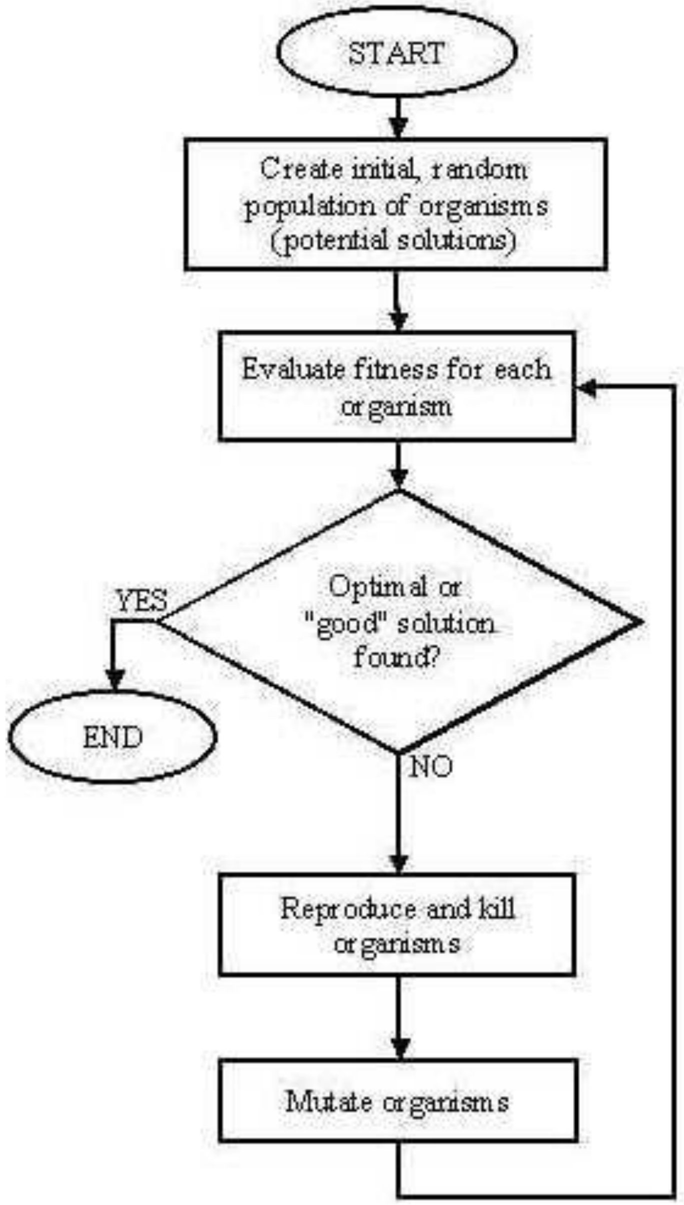
Comparison of Natural and GA Terminology

Natural	Genetic Algorithm
Chromosome	String
Gene	Feature or character
Allele	Feature value
Locus	String position
Genotype	Structure
Phenotype	Parameter set, a decoded structure

Genetic Algorithm (3) – Basic Algorithm

- Outline of the basic algorithm

```
0  START    : Create random population of n chromosomes
1  FITNESS  : Evaluate fitness f(x) of each chromosome in the population
2  NEW POPULATION
        1 REPRODUCTION/SELECTION    : Based on f(x)
        2 CROSS OVER : Cross-over chromosomes
        3 MUTATION      : Mutate chromosomes
3  REPLACE : Replace old with new population: the new generation
4  TEST     : Test problem criterium
5  LOOP     : Continue step 1 – 4 untill criterium is  satisfied
```


START
Create initial, random population of organisms (potential solutions)
Evaluate fitness for each organism
Optimal or "good" solution found?
YES
END
NO
Reproduce and kill organisms
Mutate organisms

Genetic Algorithm – Coding

- Chromosomes are encoded by bitstrings.
- Every bitstring therefore is a solution but not necessarily the best solution.
- The way bitstrings can code differs from problem to problem.

1
0
0
1

Either sequence of on/off or the number 9

GA Operators

- Selection
- Crossover
- Mutation

Selection operator

The process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out.

The primary objective of the selection operator is to emphasize the good solutions and eliminate the bad solutions in a population while keeping the population size constant.

“Selects the best, discards the rest”

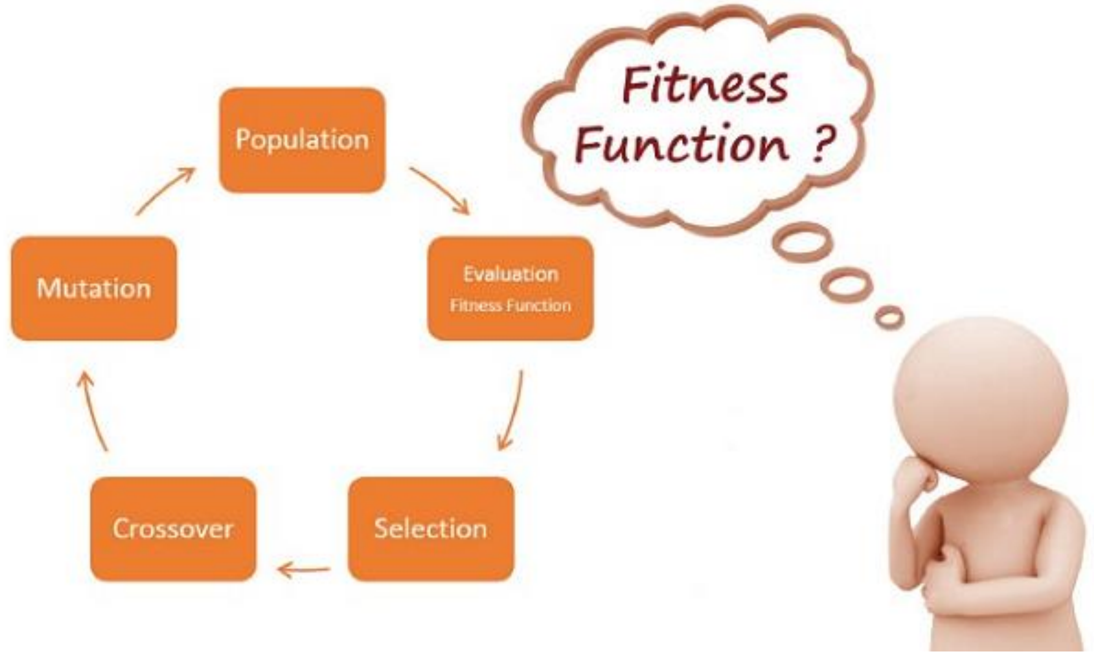
Functions of Selection operator

- Identify the good solutions in a population
- Make multiple copies of the good solutions
- Eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population

Fitness function

- A fitness value can be assigned to evaluate the solutions.

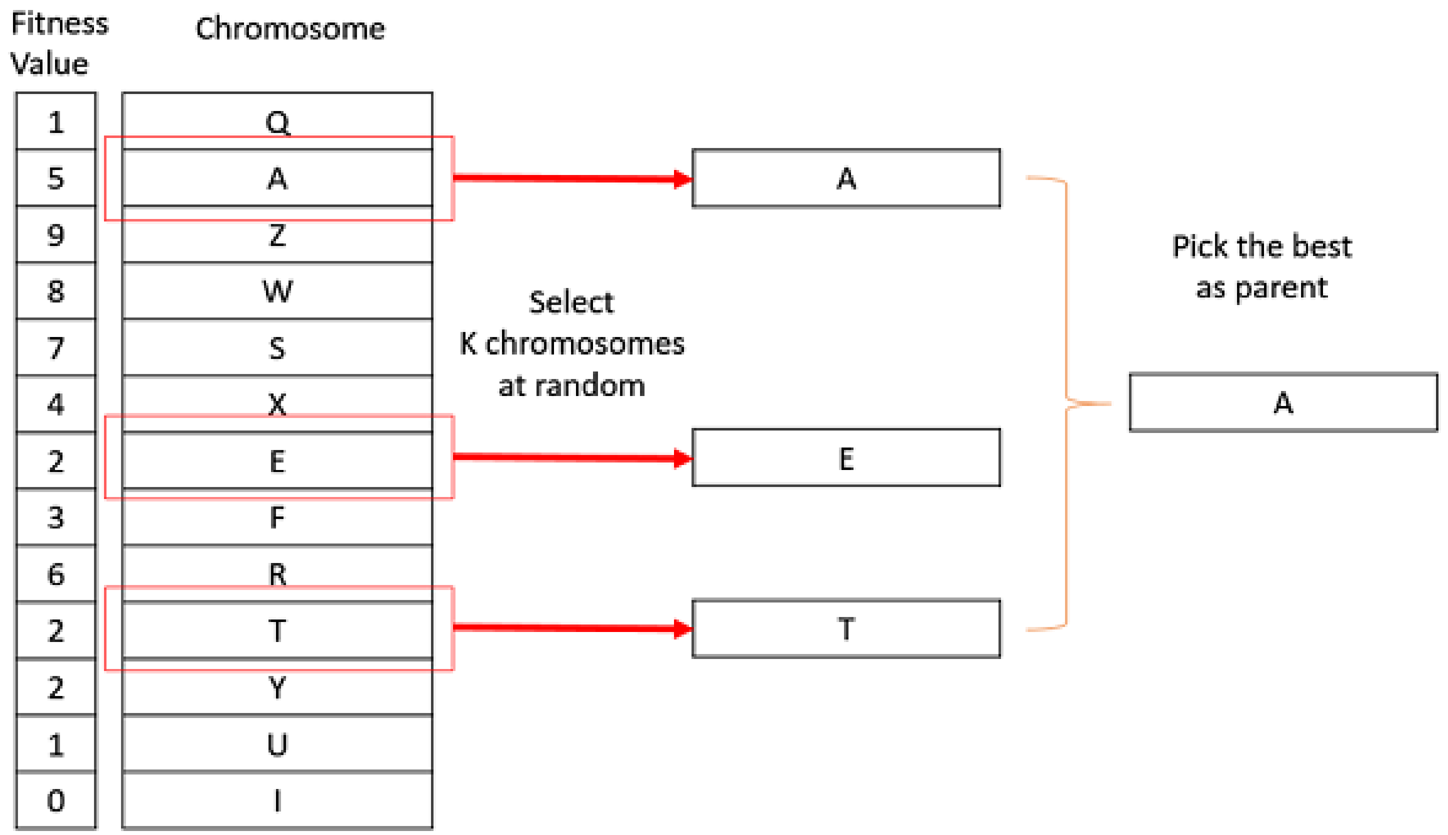
- A fitness function value quantifies the optimality of a solution. The value is used to rank a particular solution against all the other solutions
- A fitness value is assigned to each solution depending on how close it is actually to the optimal solution of the problem
- The fitness function should be implemented efficiently.
  If the fitness function becomes the bottleneck of the algorithm, then the overall efficiency of the genetic algorithm will be reduced.



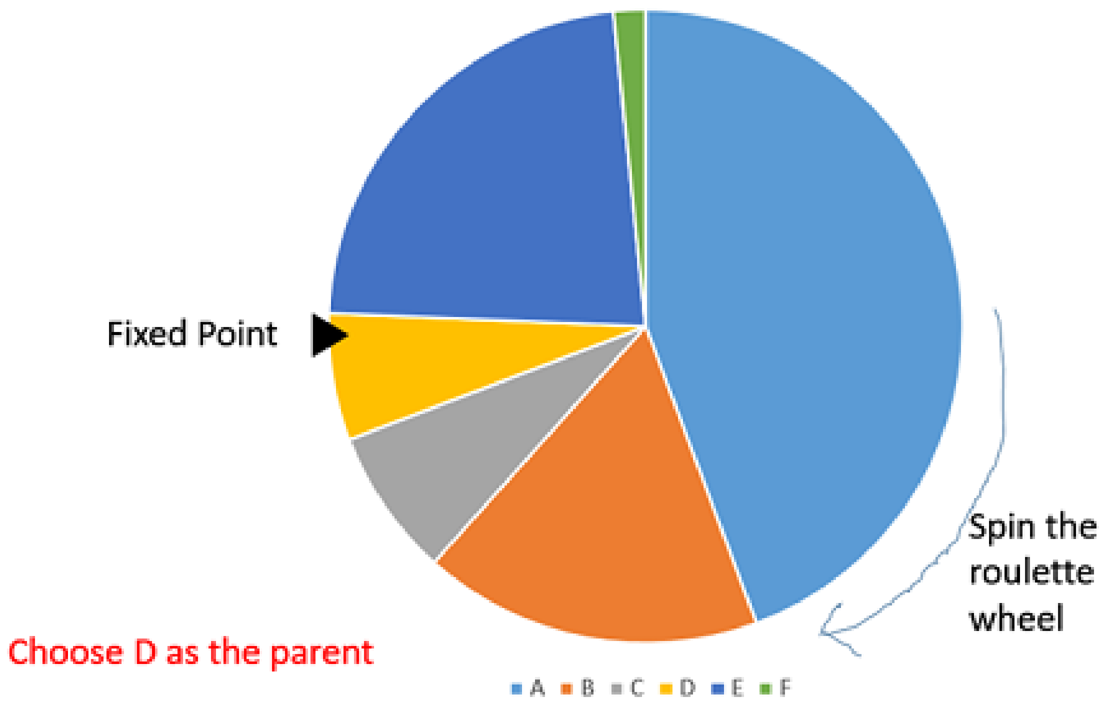
Selection operator

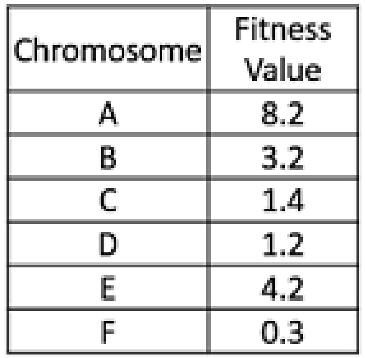
There are different techniques to implement selection in Genetic Algorithms.

- **Tournament selection :** In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.



- **Roulette wheel selection:** In a roulette wheel selection, the circular wheel is divided into n pies. A fixed point is chosen on the wheel circumference and the wheel is rotated.





Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Proportionate selection : Fitness Proportionate Selection is one of the most popular ways of parent selection. In this every individual can become a parent with a probability which is proportional to its fitness. Therefore, fitter individuals have a higher chance of mating and propagating their features to the next generation.

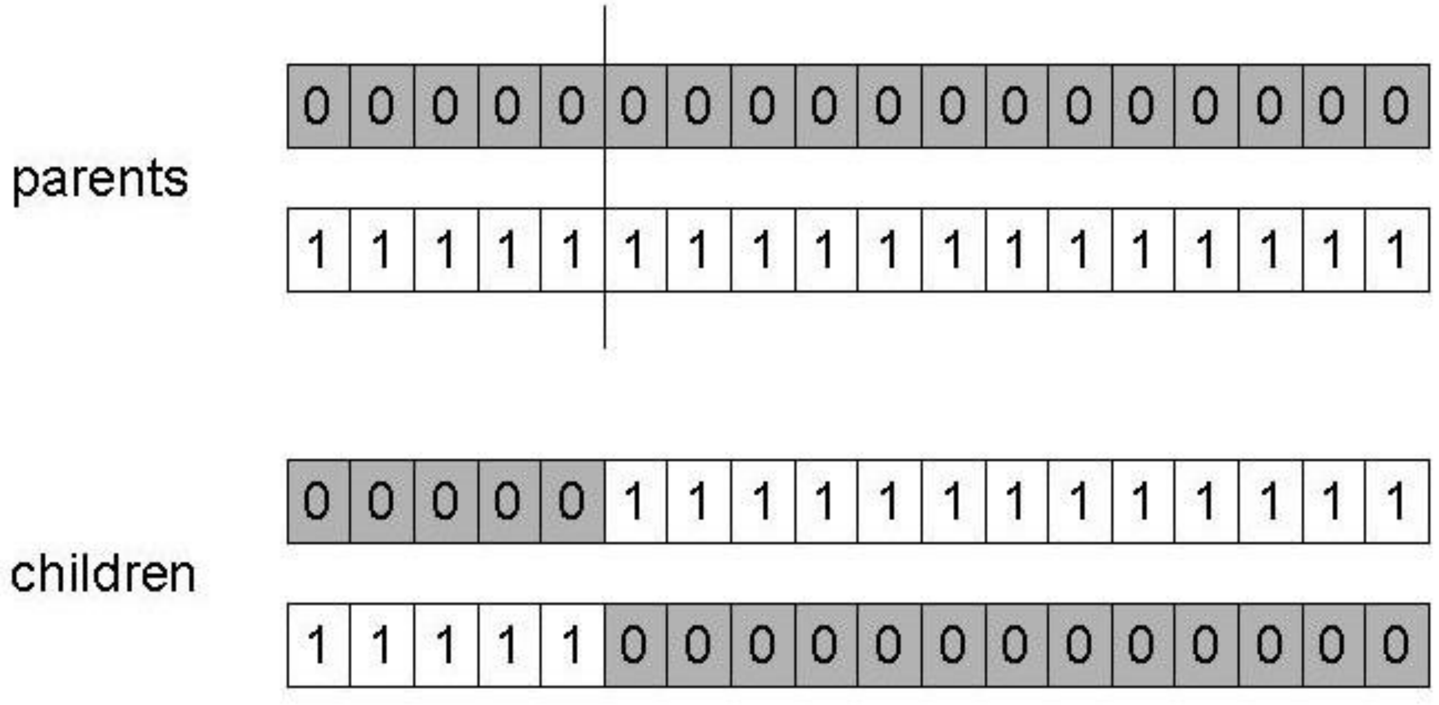
- **Rank selection:** Rank Selection also works with negative fitness values and is mostly used when the individuals in the population have very close fitness values (this happens usually at the end of the run). This leads to each individual having an almost equal share of the pie

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

Random: In this strategy we randomly select parents from the existing population. There is no selection pressure towards fitter individuals

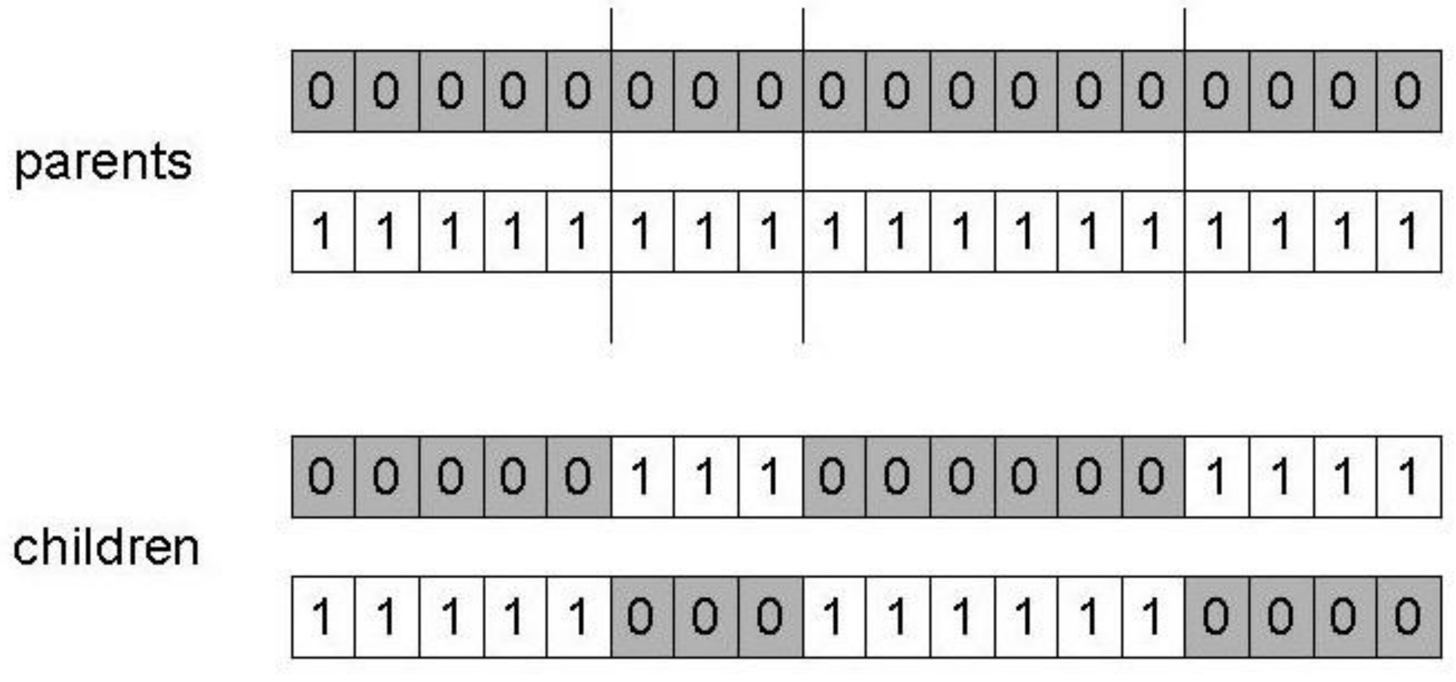
Genetic Algorithm – Crossover (Single Point)

- Choose a random point on the two parents.
- Split parents at this crossover point.
- Create children by exchanging tails.



Genetic Algorithm – Crossover (n Points)

- Choose n random crossover points.
- Split along those points.
- Glue parts, alternating between parents.
- Generalization of 1 point.



Genetic Algorithm – Uniform Crossover

Generate uniformly random number.

X1 = 0 1 1 0 0 0 1 0 1 0

X2 = 1 1 0 0 0 0 0 1 1 1

Uniformly generated = 1 0 0 0 0 0 1 0 0 0

As a result, the new population becomes,

X1 = 1 1 1 0 0 0 0 0 1 0

X2 = 0 1 0 0 0 0 1 1 1 1

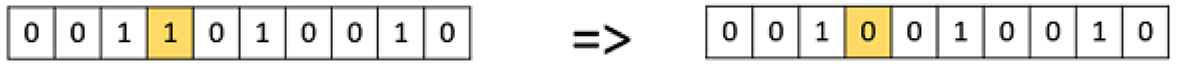
Mutation

- Mutation is the addition of new characteristics into the population. With mutation, we randomly try to inject newer characteristics into the population. If these characteristics are good, they will survive and continue into the next population.

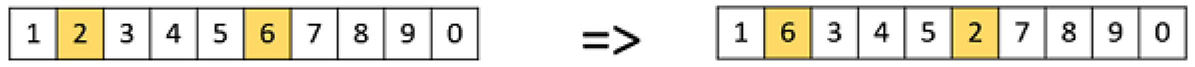
- Mutation is carried out on a very few individuals, because these are random characteristics that can drive the population anywhere.

- The amount of mutation to be carried out is specified by the mutation rate *rm, a number between* 0 and 1. This number is usually kept low to keep a limited mutation in the algorithm. We iterate through each entity in a solution in the population.

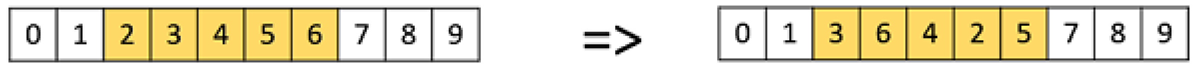
- **Uniform Mutation :** Uniform mutation is the most common mutation operation. In this value at the selected location is simply flipped. Thus if a 0 is present in the location, it changes to 1, and vice versa.



- **Swap Mutation:** In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.



- **Scramble Mutation :** Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.

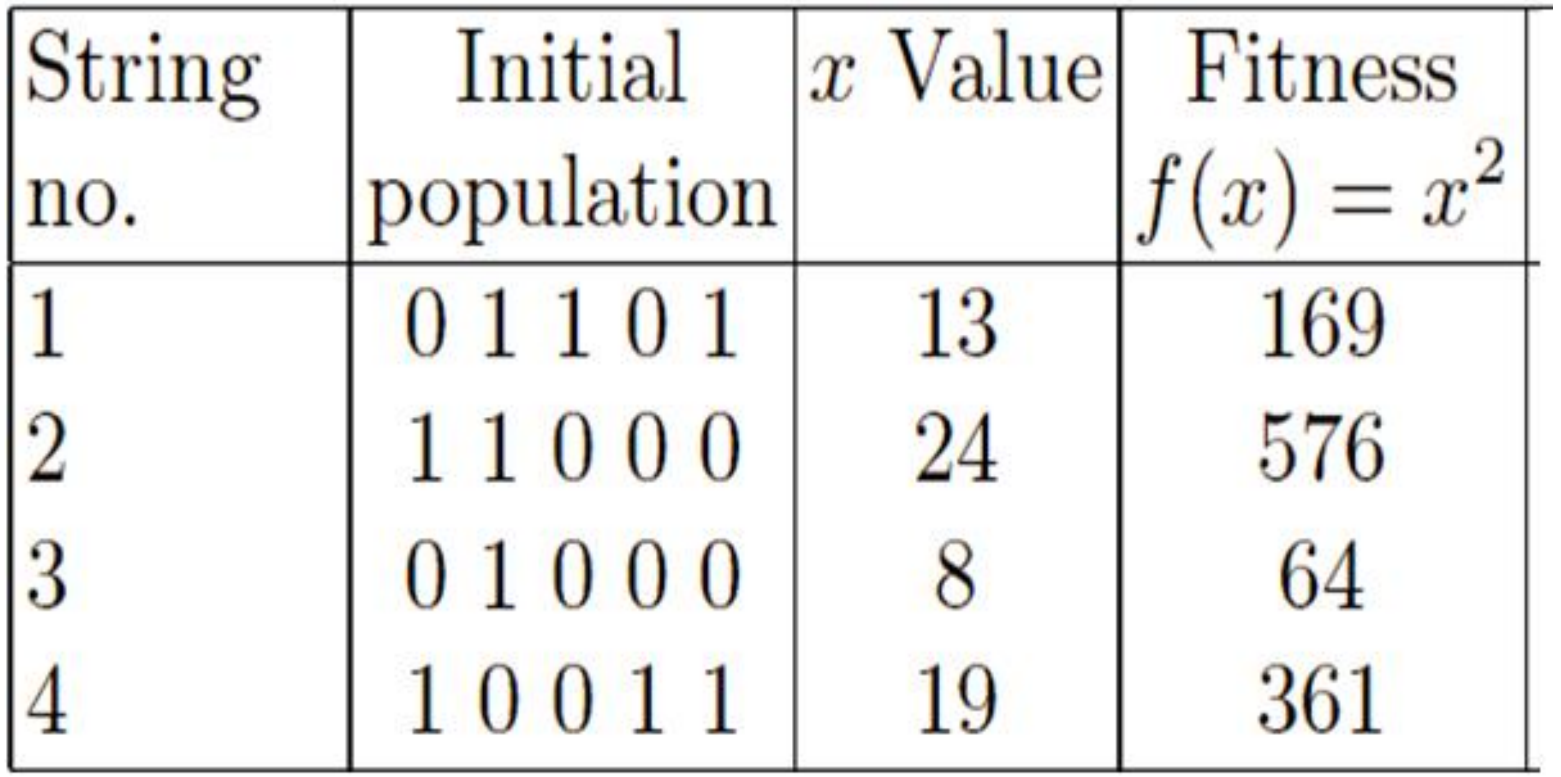


- **Gaussian Mutation:** Gaussian mutation happens in the case of the phenotype representation of the solution, where the solution is represented by simple numerals. In this type of mutation, we add a randomly generated number to the existing value at the selected location.

Genetic Algorithm – An Example

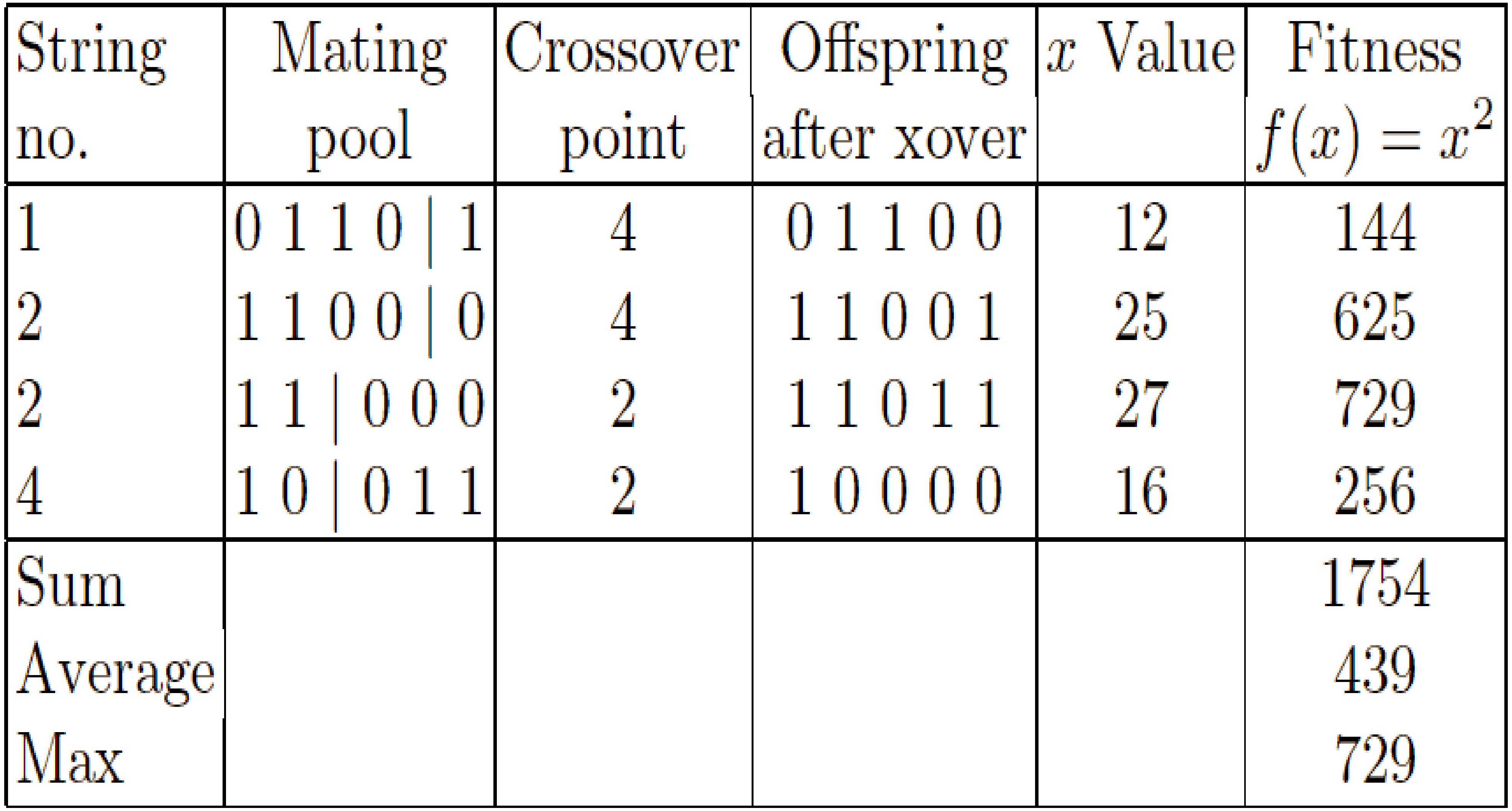
- Simple problem: max x^2 over $\{0, 1, \ldots, 31\}$
- GA approach:
  - Representation: binary code, e.g. 01101 $\leftrightarrow$ 13
  - Population size: 5
  - 1-point xover,
  - bitwise mutation
- One generational cycle performed manually is shown here.

Example : Selection



String no.	Initial population	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 1	13	169
2	1 1 0 0 0	24	576
3	0 1 0 0 0	8	64
4	1 0 0 1 1	19	361

Example : Crossover



String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Example : Mutation

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

Comparison of GA with Traditional Optimization Techniques

- GA works with the coding of solution set and not with the solution itself.

- GA uses population of solutions rather than a single solution for searching.

- GA uses fitness function for evaluation rather the derivatives.

- GA uses probabilistic transition and not deterministic rules.