LEARNING FROM OBSERVATIONS

AI-302T

UNIT –IV

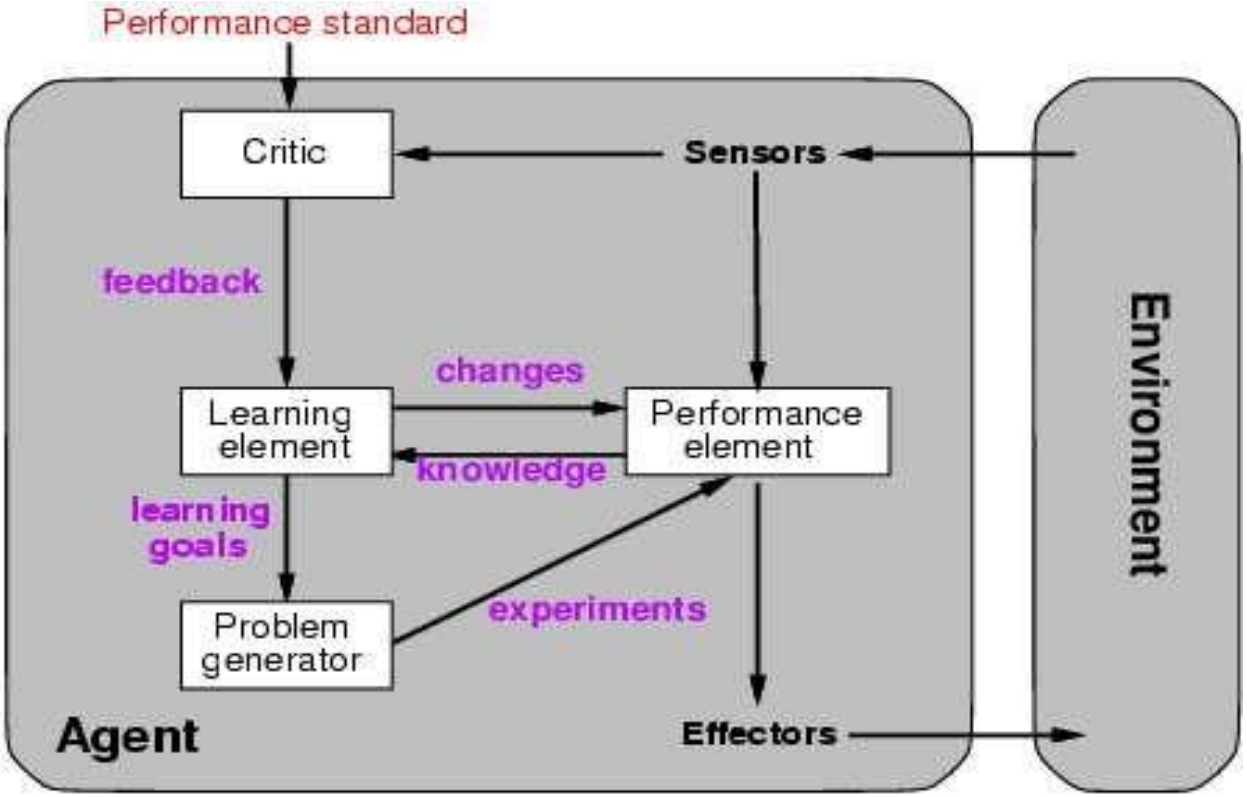
Outline

- Learning agents
- Inductive learning
- Decision tree learning

LEARNING

- Learning is essential for unknown environments,
  - i.e., when designer lacks omniscience
- Learning is useful as a system construction method,
  - i.e., expose the agent to reality rather than trying to write it down
- Learning modifies the agent's decision mechanisms to improve performance

LEARNING AGENTS



LEARNING ELEMENT

- Design of a learning element is affected by
  - Which components of the performance element are to be learned
  - What feedback is available to learn these components
  - What representation is used for the components

- Type of feedback:
  - Supervised learning: correct answers for each example
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards

TYPES OF LEARNING

- Supervised learning: correct answer for each example. Answer can be a numeric variable, categorical variable etc.

M M F F M F

- Unsupervised learning: correct answers not given – just examples (e.g. – the same figures as above , without the labels)
- Reinforcement learning: occasional rewards.

INDUCTIVE LEARNING

Simplest form: learn a function from examples

f is the target function

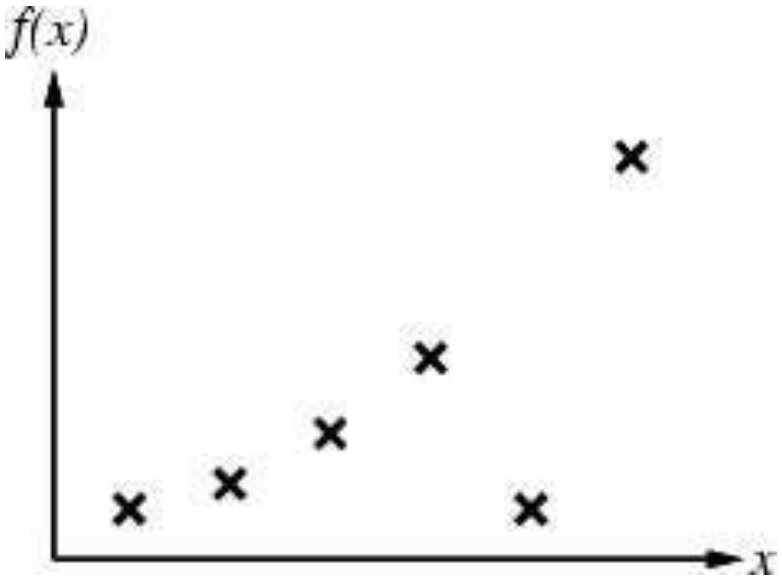
An example is a pair $(x, f(x))$

Problem: find a hypothesis *h*
such that $h \approx f$
given a training set of examples

(This is a highly simplified model of real learning:
- Ignores prior knowledge
- Assumes examples are given)

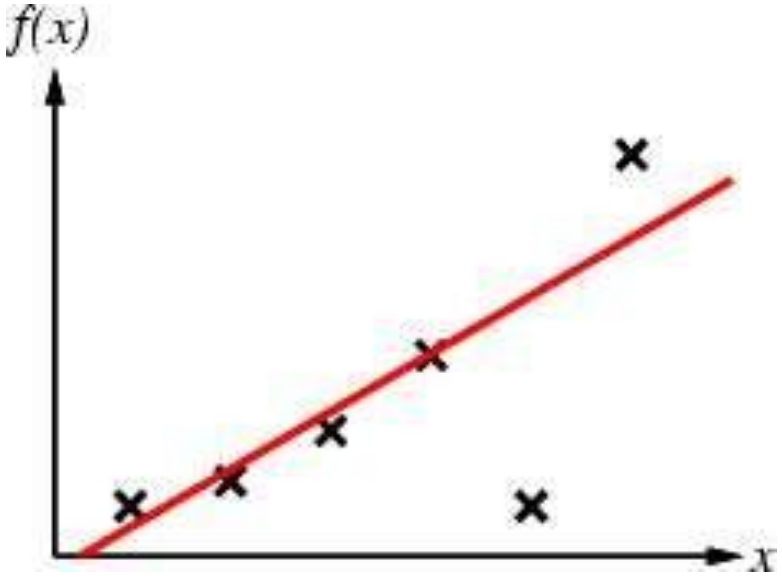
INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



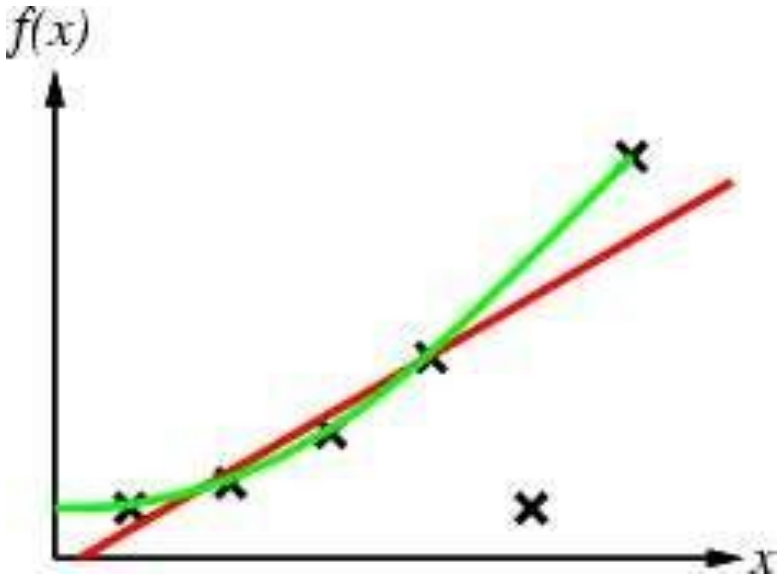
INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



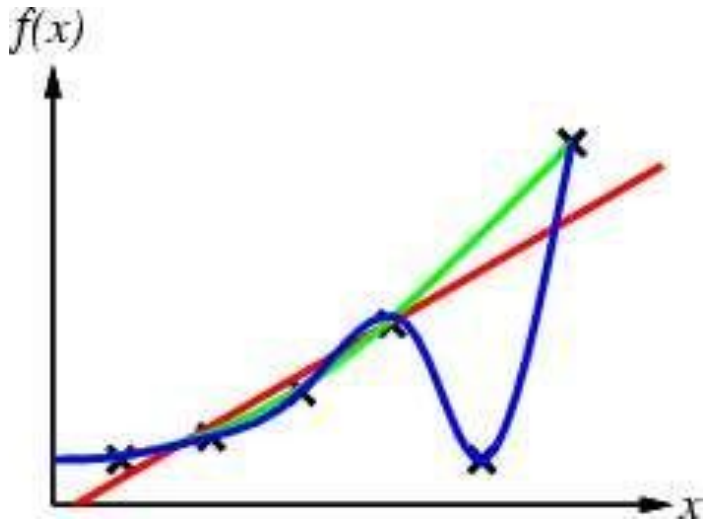
INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



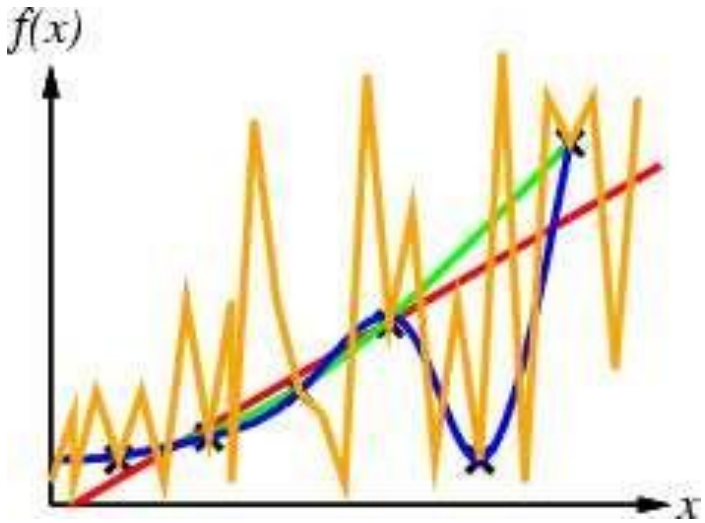
INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



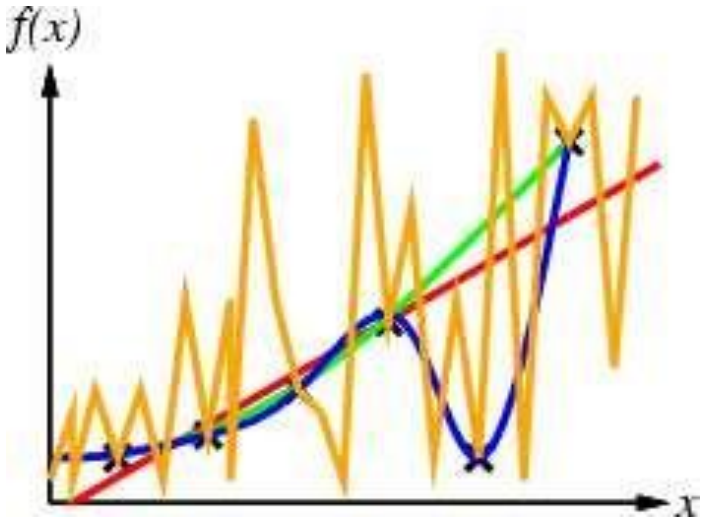
INDUCTIVE LEARNING METHOD

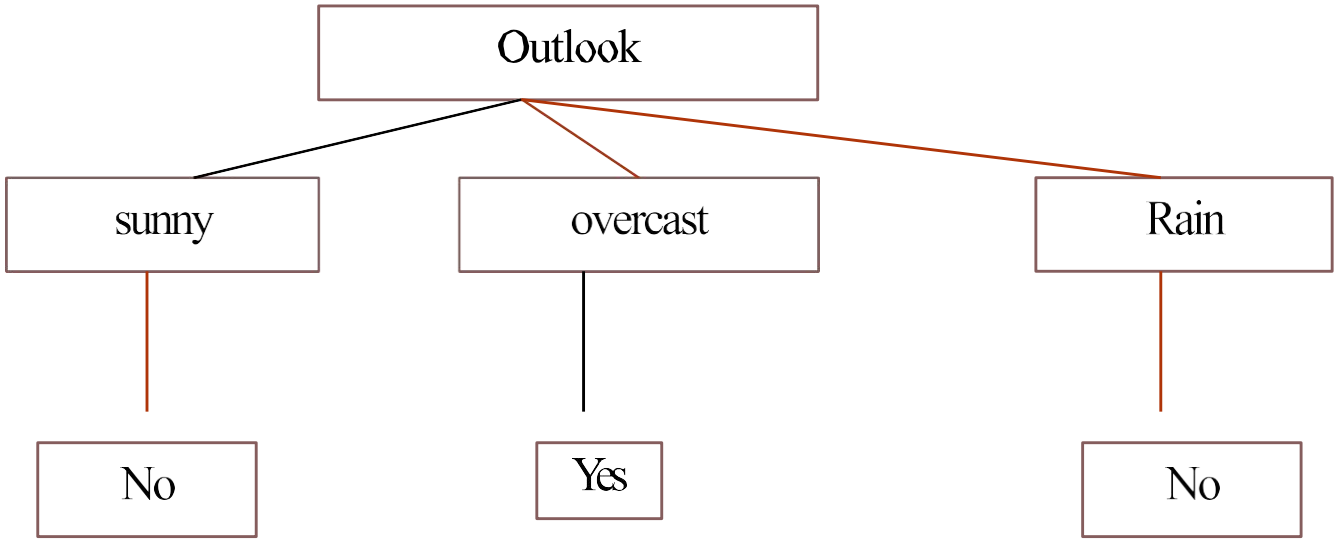
- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



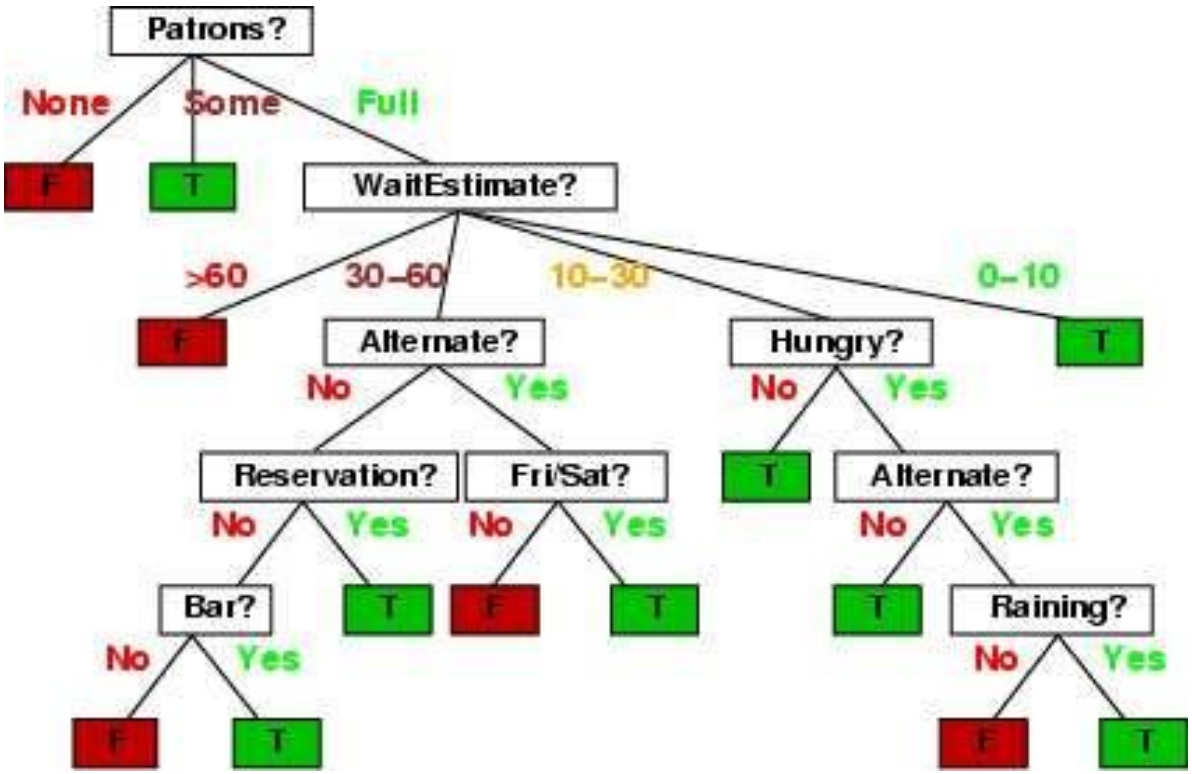

Outlook
sunny
overcast
Rain
No
Yes
No

ID3

- Iterative Dichotomizer-3
- Calculates Entropy and Information Gain

DECISION TREES

- One possible representation for hypotheses
- E.g., here is the "true" tree for deciding whether to wait:



ATTRIBUTE-BASED REPRESENTATIONS

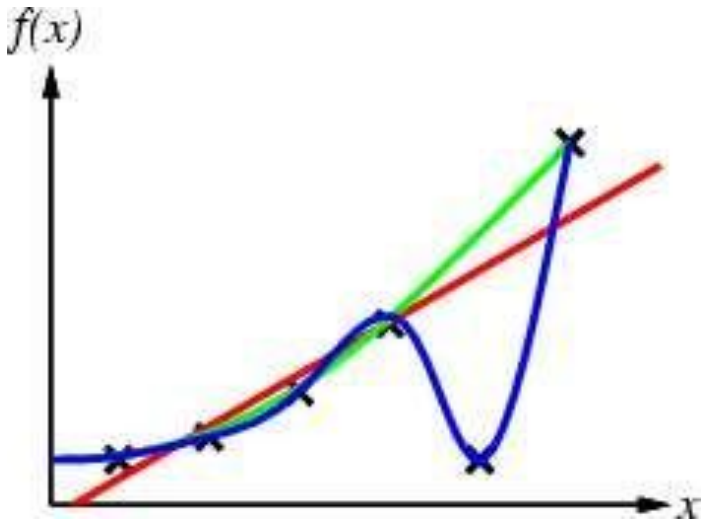
- Examples described by attribute values (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

- Classification of examples is positive (T) or negative (F)
- The set of examples used for learning is called training set.

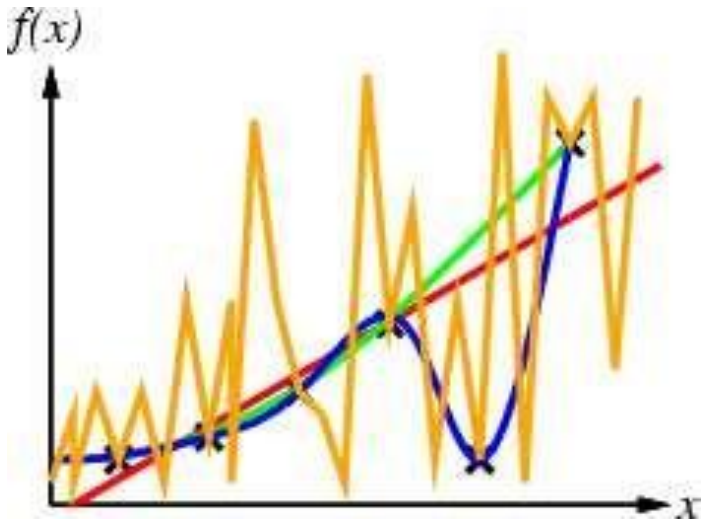
INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



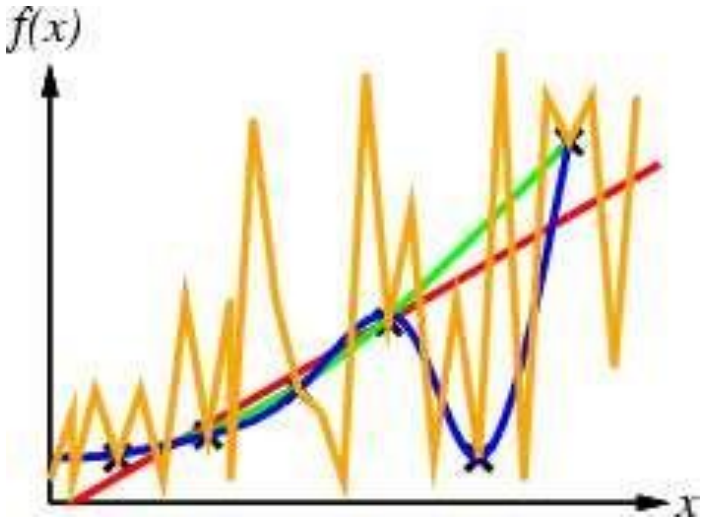
INDUCTIVE LEARNING METHOD

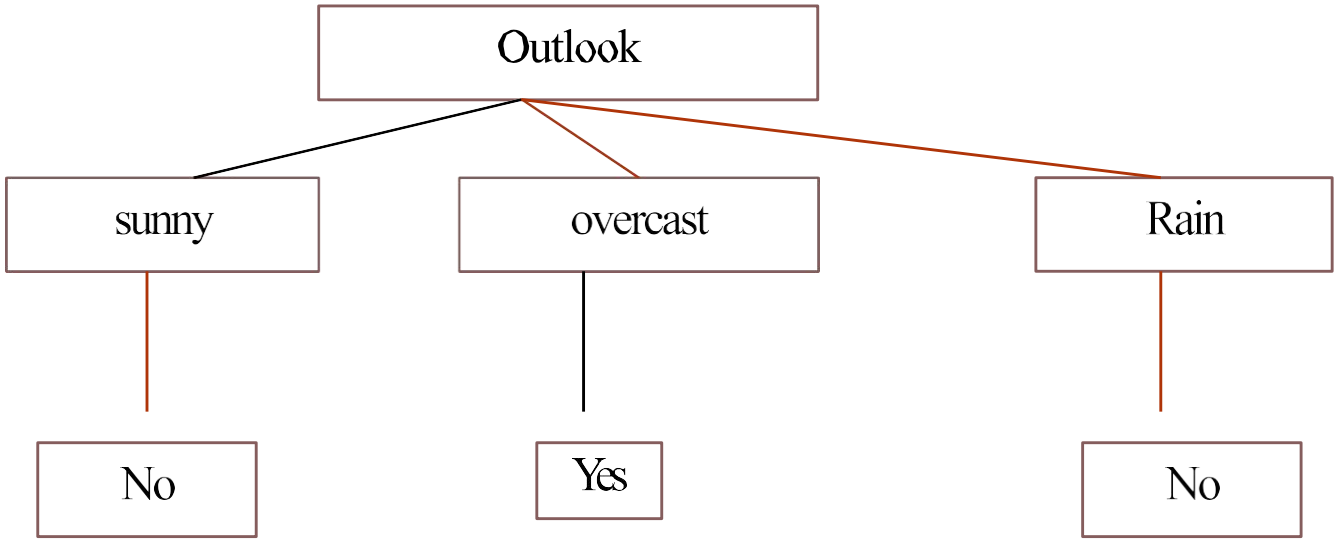
- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



INDUCTIVE LEARNING METHOD

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



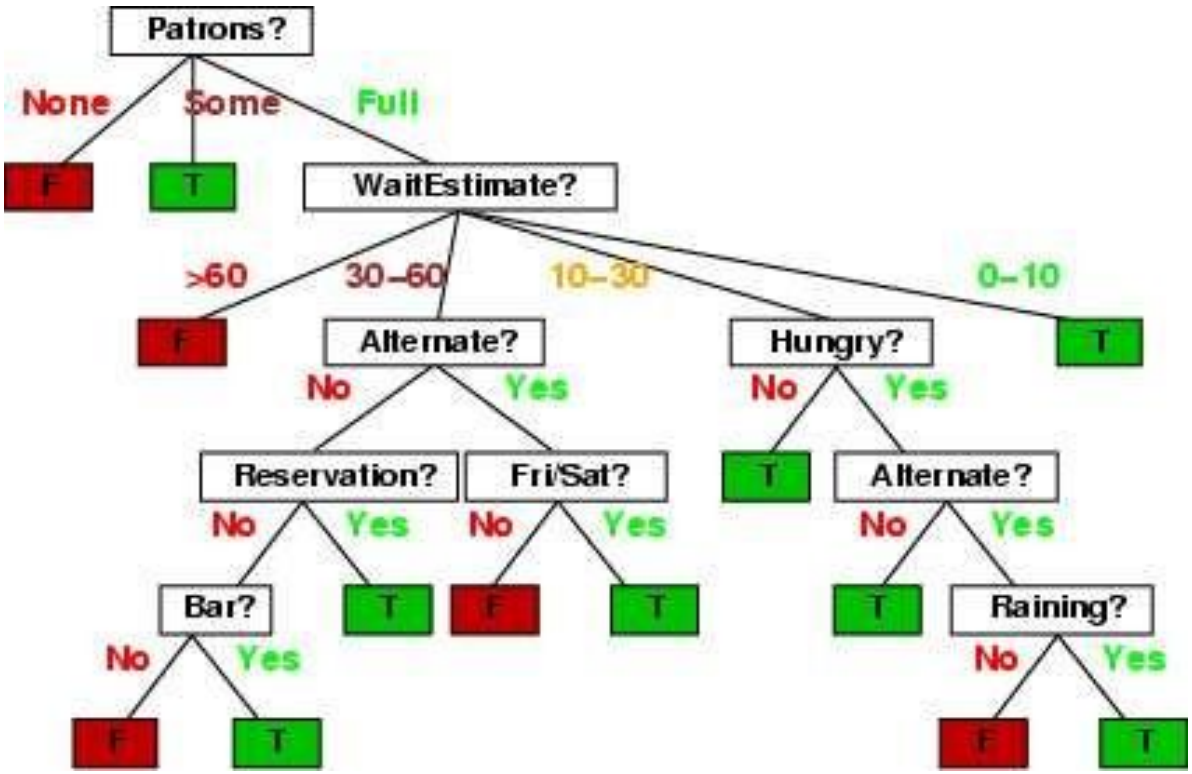

Outlook
sunny
overcast
Rain
No
Yes
No

ID3

- Iterative Dichotomizer-3
- Calculates Entropy and Information Gain

DECISION TREES

- One possible representation for hypotheses
- E.g., here is the "true" tree for deciding whether to wait:



ATTRIBUTE-BASED REPRESENTATIONS

- Examples described by attribute values (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

- Classification of examples is positive (T) or negative (F)
- The set of examples used for learning is called training set.

USING INFORMATION THEORY

- To implement **Choose-Attribute** in the DTL algorithm
- Information Content (Entropy):

$$I(P(v_1), \ldots, P(v_n)) = \Sigma_{i=1} -P(v_i) \log_2 P(v_i)$$

- For a training set containing *p* positive examples and *n* negative examples:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

INFORMATION GAIN

- A chosen attribute A divides the training set E into subsets $E_1, \ldots, E_v$ according to their values for A, where A has v distinct values.

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

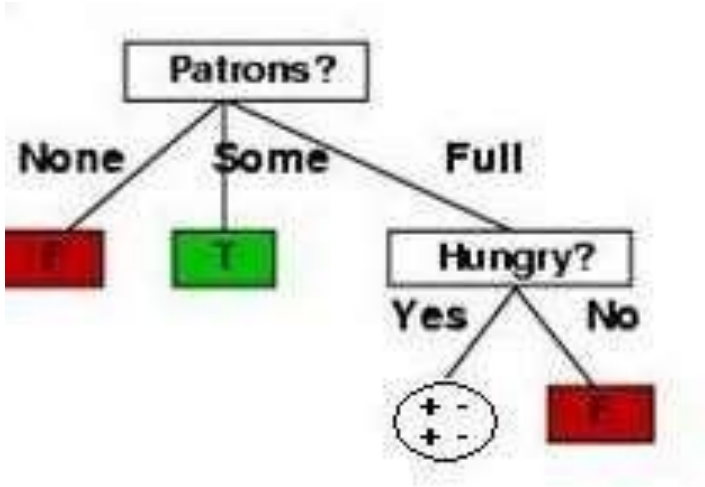
- Information Gain (IG) or reduction in entropy from the attribute test:

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

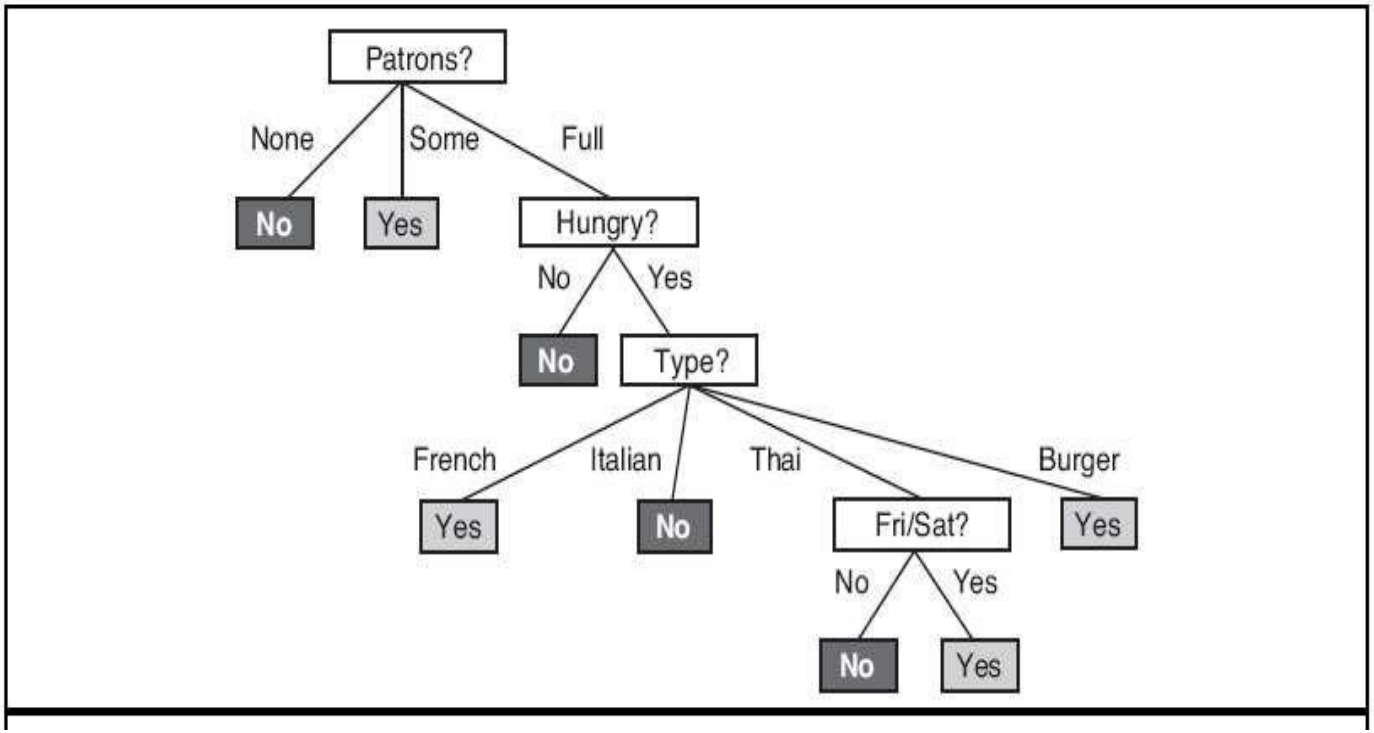
- Choose the attribute with the largest IG

Next step

Given *Patrons* as root node, the next attribute chosen is *Hungry?*, with IG(Hungry?) = I(1/3, 2/3) – (2/3*1 + 1/3*0) = 0.252



FINAL DECISION TREE INDUCED BY 12-EXAMPLE TRAINING SET



BRAODENING THE APPLICABILITY OF DECISION TREES

- Missing data
- Multivalued attributes
- Continuous or integer values attributes
- Continuous output attributes

COMPUTATIONAL LEARNING THEORY – WHY LEARNING WORKS

- PAC learning(Probably Approximately Correct)
- This has been a breakthrough in the theory of machine learning.
- Basic idea: A really bad hypothesis will be easy to identify, With high probability it will err on one of the training examples.
- A **consistent** hypothesis will be probably approximately correct.
- Notice that if there are more training example, then the probability of "approximately correct" becomes higher!

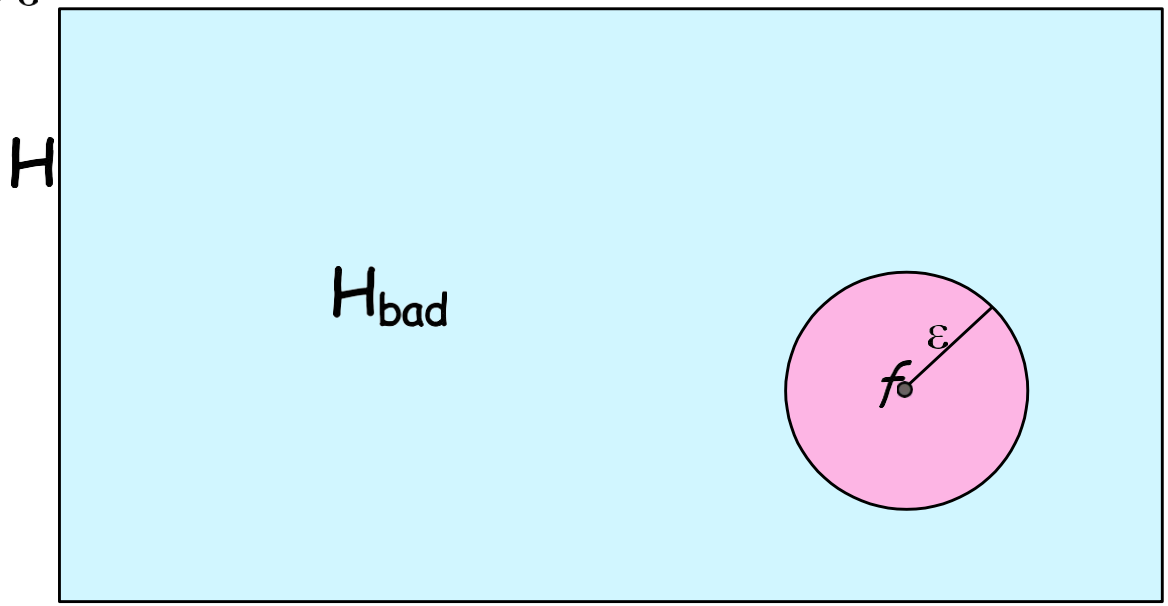
COMPUTATIONAL LEARNING THEORY – HOW MANY EXAMPLES ARE NEEDED

Let:

- X the set of all possible examples
- *D* the distribution from which samples are drawn
- **H** the set of possible hypothesis
- m the number of examples in the training set And assume that f, the true function is in H.

ERROR

- Error(h) = P(h(x) $\neq$ f(x) | x drawn from *D*)
- An **approximately correct** hypothesis h, is a hypothesis that satisfies Error(h) < ε



- $m \geq 1/\varepsilon (\ln|H| - \ln \delta)$, where δ is the probability that H_{bad} contains a hypothesis consistent with all examples

EXAMPLES

- Learning a Boolean function;
- Learning a conjunction of n literals
- Learning decision lists

BOOLEAN FUNCTION

- A general boolean function on n attributes can be represented by its truth table
- Size of truth table 2^n

A	B	C	F(A,B,C)
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	T

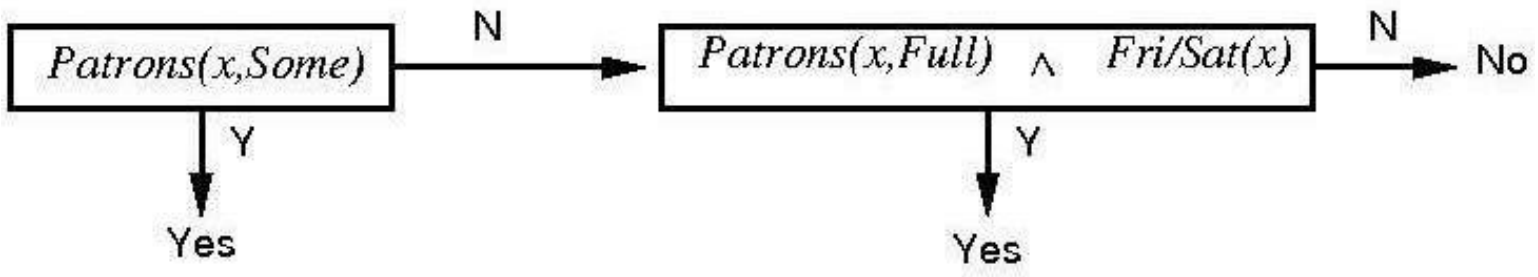
Arbitrary boolean function on 3 attributes

CONJUNTION OF LITERALS

- A literal is a variable or its negation
- $A \wedge \neg B \wedge C$ is an example of conjunction of literals

LEARNING DECISION LISTS

- A decision list consists of a series of tests, each of which is a conjunction of literals. If the tests succeeds, the decision list specifies the value to be returned. Otherwise, the processing continues with the next test in the list.
- Decision lists can represent any boolean function hence are not learnable.



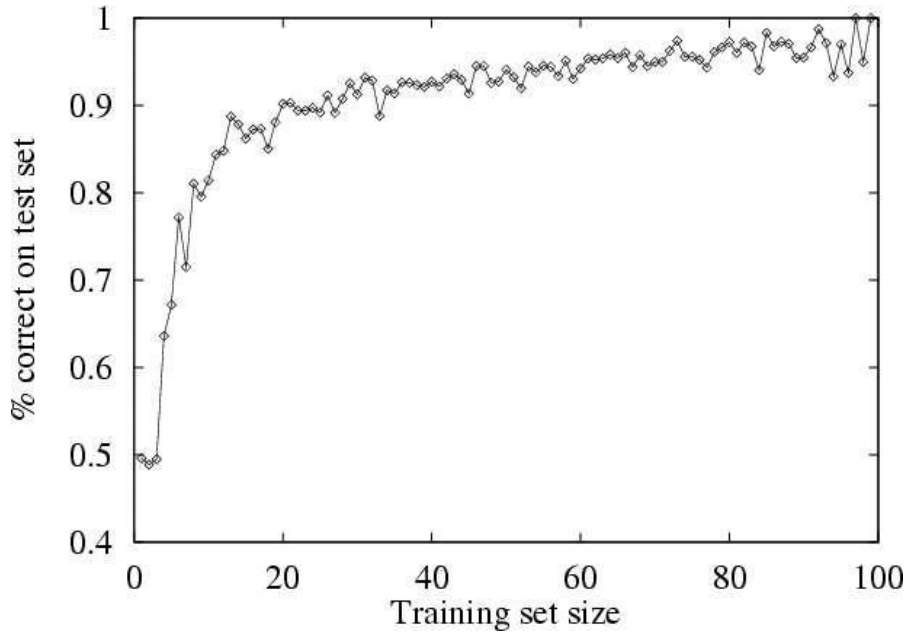
K- DL

- A k-DL is a decision list where each test is restricted to at most k literals.
- K- Dl is learnable!

PERFORMANCE MEASUREMENT

- How do we know that $h \approx f$?
  1. Use theorems of computational/statistical learning theory
  2. Try h on a new test set of examples
     (use same distribution over example space as training set)

Learning curve = % correct on test set as a function of training set size



Learning curve for learning decision trees – restaurant problem

REGRESSION AND CLASSIFICATION WITH LINEAR MODELS

- Univariate linear regression
- Multivariate linear regression
- Linear classifiers with a hard threshold
- Linear classifiers with a logistic regression

ARTIFICIAL NEURAL NETWORDS

- NN structures
- Single layer networks - perceptrons
- Multilayer neural networks

NON PARAMETRIC MODELS

- Nearest Neighbor models

SUMMARY

- Learning needed for unknown environments, lazy designers
- Learning agent = performance element + learning element
- For supervised learning, the aim is to find a simple hypothesis approximately consistent with training examples
- Decision tree learning using information gain
- Learning performance = prediction accuracy measured on test set.

Introduction

Why "Learn"?

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to "learn" to calculate payroll
- Learning is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech recognition)
  - Solution changes in time (routing on a computer network)
  - Solution needs to be adapted to particular cases (user biometrics)

What We Talk About When We Talk About"Learning"

- Learning general models from a data of particular examples
- Data is cheap and abundant (data warehouses, data marts); knowledge is expensive and scarce.
- Example in retail: Customer transactions to consumer behavior: *People who bought "DaVinci Code" also bought "The Five PeopleYou Meet in Heaven" (www.amazon.com)*
- Build a model that is *a good and useful approximation* to the data.

Data Mining/KDD

Definition := *"KDD is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* (Fayyad)

Applications:

- Retail: Market basket analysis, Customer relationship management (CRM)
- Finance: Credit scoring, fraud detection
- Manufacturing: Optimization, troubleshooting
- Medicine: Medical diagnosis
- Telecommunications: Quality of service optimization
- Bioinformatics: Motifs, alignment
- Web mining: Search engines …

What is Machine Learning?

- Machine Learning
  - Study of algorithms that
  - improve their performance
  - at some task
  - with experience
- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

Growth of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment
  - It turns out to be difficult to extract knowledge from human experts → *failure of expert systems in the 1980's.*

Applications

- Association Analysis
- Supervised Learning
  - Classification
  - Regression/Prediction
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

- Basket analysis:

  $P(Y \mid X)$ probability that somebody who buys X also buys Y where X and Y are products/services.

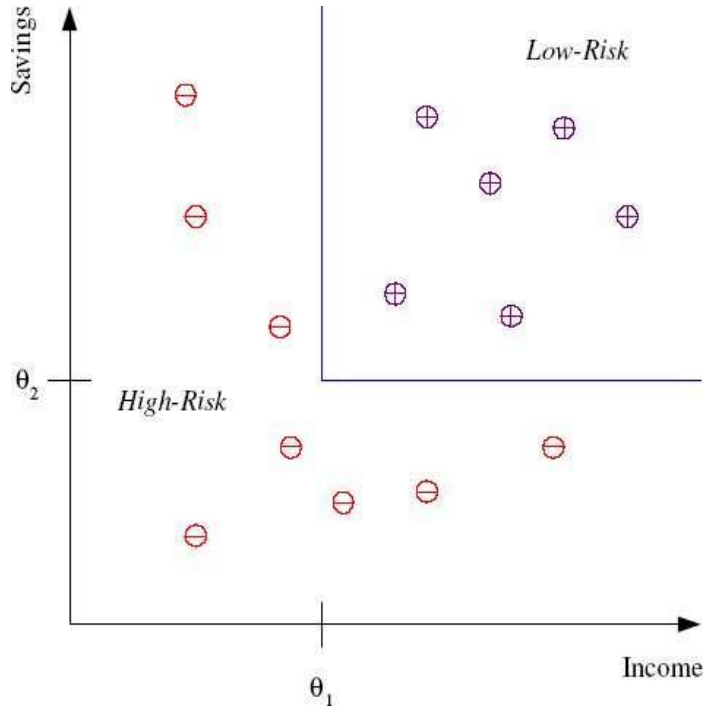
  Example: $P(\text{bread} \mid \text{butter}) = 0.7$

Market-Basket transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF $income > \theta_1$ AND $savings > \theta_2$
THEN low-risk ELSE high-risk

Model

Classification: Applications

- Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- Web Advertizing: Predict if a user clicks on an ad on the Internet.

Face Recognition

Training examples of a person

Test images

AT&T Laboratories, Cambridge UK
http://www.uk.research.att.com/facedatabase.html

Prediction: Regression

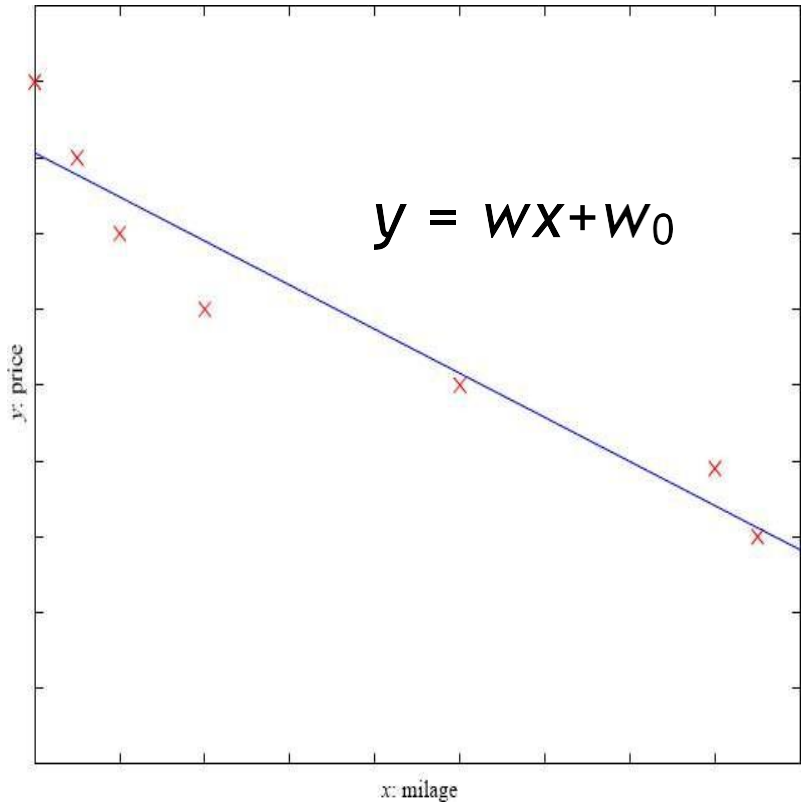
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g(x \mid \theta)$$

  $g(\ )$ model,

  θ parameters



Supervised Learning: Uses

Example: decision trees tools that create rules

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction:The rule is easy to understand
- Compression:The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Unsupervised Learning

- Learning "what normally happens"
- No output
- Clustering: Grouping similar instances
- Other applications: Summarization, Association Analysis
- Example applications
  - Customer segmentation in CRM
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Reinforcement Learning

- Topics:
  - Policies: what actions should an agent take in a particular situation
  - Utility estimation: how good is a state (→used by policy)
- No supervised output but delayed reward
- Credit assignment problem (what was responsible for the outcome)
- Applications:
  - Game playing
  - Robot in a maze
  - Multiple agents, partial observability …

Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/

GENETIC ALGORITHM

INTRODUCTION

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve.

INTRODUCTION TO OPTIMIZATION

Optimization is the process of making something better.

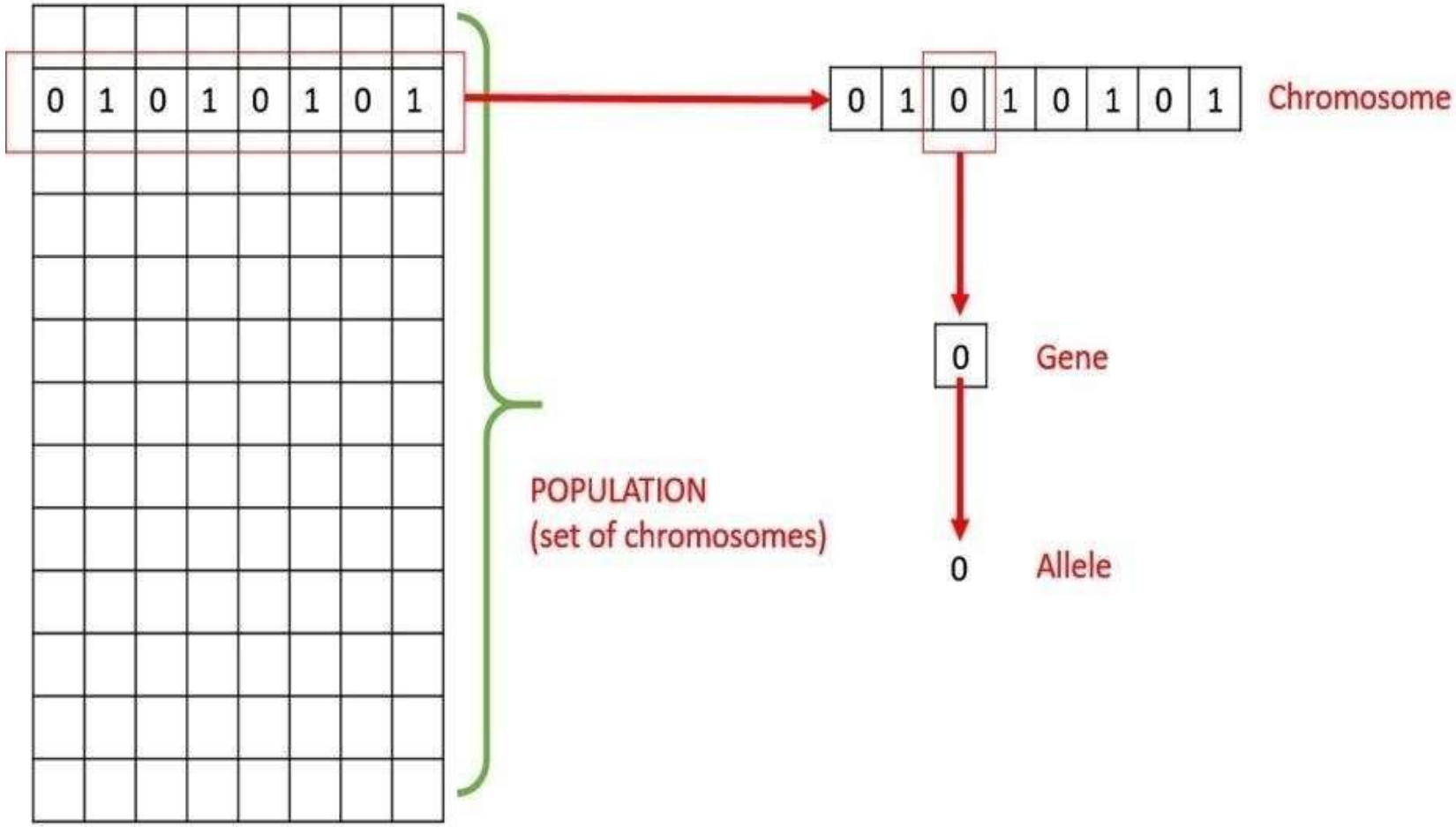
What are Genetic Algorithms?

- Nature has always been a great source of inspiration to all mankind. Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics.
- GAs are a subset of a much larger branch of computation known as Evolutionary Computation.

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.

- Individual - Any possible solution
- Population - Group of all individuals
- Fitness – Target function that we are optimizing (each individual has a fitness)
- Trait - Possible aspect (features) of an individual
- Genome - Collection of all chromosomes (traits) for an
- individual.

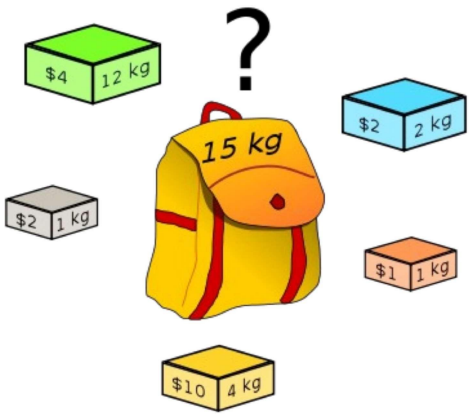

0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1
Chromosome
0
Gene
POPULATION
(set of chromosomes)
0
Allele

- **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** –Decoding is a process of transforming a solution from the genotype to the phenotype space,
- Encoding is a process of transforming from the phenotype to genotype space.

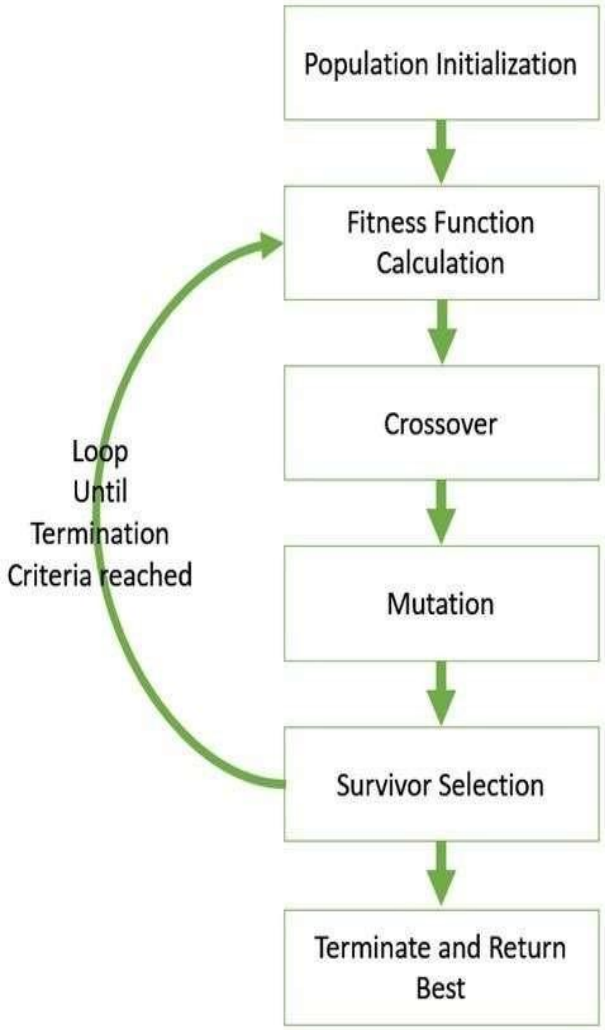
Fitness Function – A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. **Genetic Operators** – These alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

Knapsack Problem

The knapsack problem or rucksack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items.

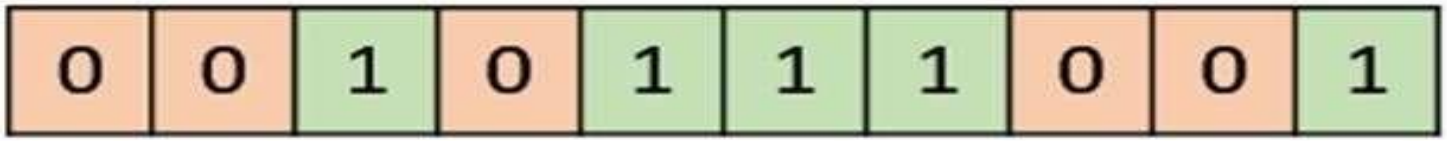


BASIC STRUCTURE OF GENETIC ALGORITHM



GENOTYPE REPRESENTATION

- BINARY REPRESENTATION



- REAL VALUED REPRESENTATION

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

- INTEGER REPRESENTATION

1	2	3	4	3	2	4	1	2	1

- PERMUTATION REPRESENTATION

1	5	9	8	7	4	2	3	6	0

GA-POPULATION

- Population is a subset of solutions in the current generation
- Set of chromosomes

- POPULATION INITIALIZATION
  1. RANDOM INITIALIZATION
  2. HEURISTIC INITIALIZATION

- **POPULATION MODEL**
  1. STEADY STATE
  2. GENERATIONAL

FITNESS FUNCTION

- Takes a candidate solution to the problem as input and produces as output
- The objective is to either maximize or minimize the given objective function

GA- TERMINATION CONDITION

- When there has been no improvement in the population for X iterations.
- When we reach an absolute number of generations.
- When the objective function value has reached a certain pre-defined value

INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS

INTRODUCTION

- "Neural" is an adjective for neuron, and "network" denotes a graph like structure.
- Artificial Neural Networks are also referred to as "neural nets" , "artificial neural systems", "parallel distributed processing systems", "connectionist systems".
- For a computing systems to be called by these pretty names, it is necessary for the system to have a labeled directed graph structure where nodes performs some simple computations.
- "Directed Graph" consists of set of "nodes"(vertices) and a set of "connections"(edges/links/arcs) connecting pair of nodes.
- A graph is said to be "labeled graph" if each connection is associated with a label to identify some property of the connection

CONTD…

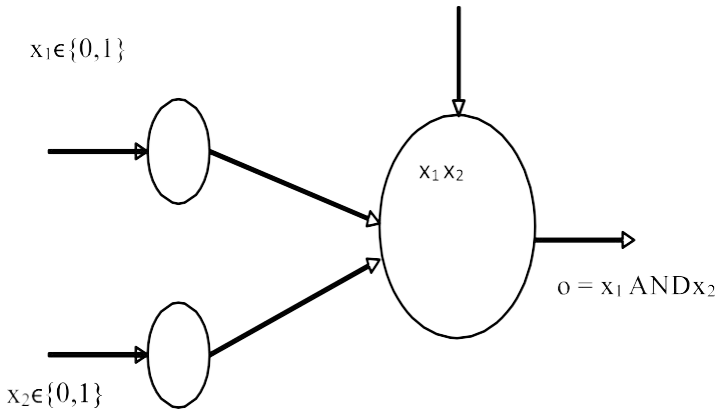


Fig 1: AND gate graph

This graph cannot be considered a neural network since the connections between the nodes are fixed and appear to play no other role than carrying the inputs to the node that computed their conjunction.

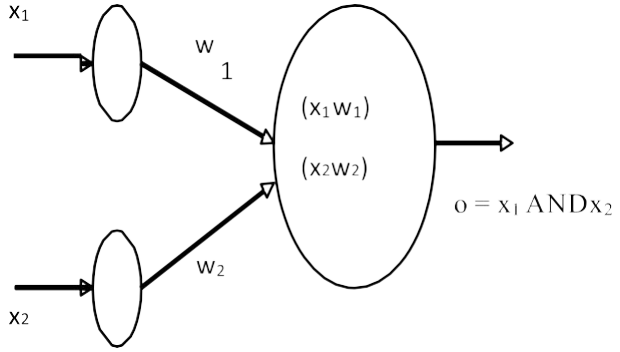


Fig 2: AND gate network

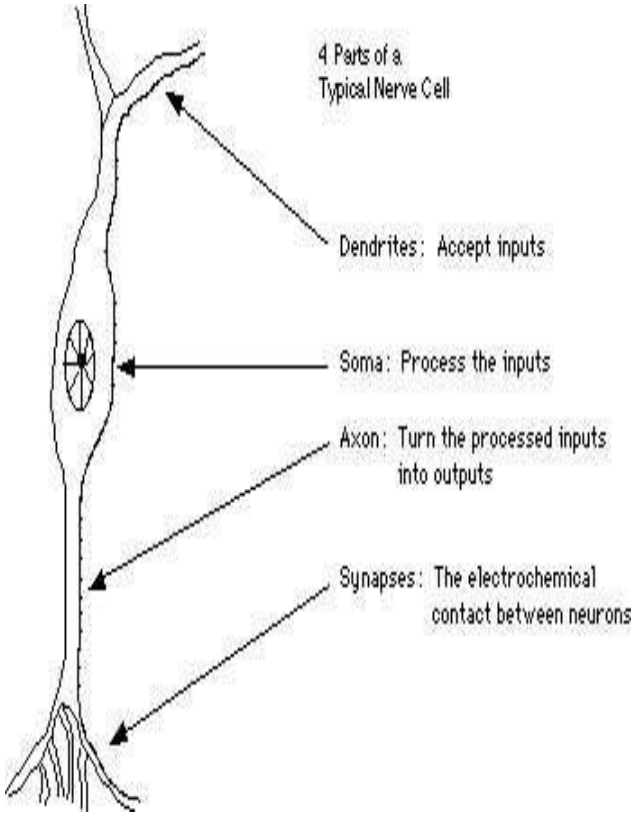
The graph structure which connects the weights modifiable using a learning algorithm, qualifies the computing system to be called an artificial neural networks.

- **The Hebb'ian neural network was pioneered by BERNARD WIDROW of Stanford University in 1950's.**

BIOLOGICAL NEURON MODEL

Four parts of a typical nerve cell : -

- DENDRITES: Accepts the inputs
- SOMA : Process the inputs
- AXON : Turns the processed inputs into outputs.
- SYNAPSES :The electrochemical contact between the neurons.

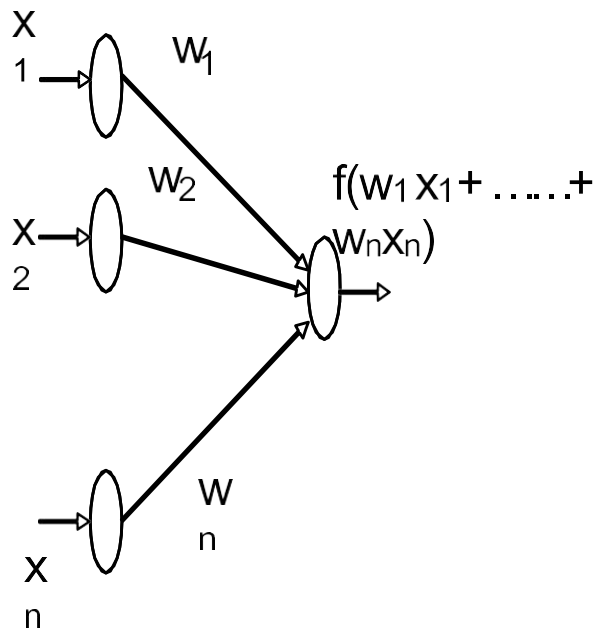


ARTIFICIAL NEURON MODEL

- Inputs to the network are represented by the mathematical symbol, x_n
- Each of these inputs are multiplied by a connection weight, w_n

$sum = w_1 x_1 + \ldots\ldots + w_n x_n$

- These products are simply summed, fed through the transfer function, f( ) to generate a result and then output.



TERMINOLOGY

Biological Terminology	**Artificial Neural Network Terminology**
Neuron	Node/Unit/Cell/Neurode
Synapse	Connection/Edge/Link
Synaptic Efficiency	Connection Strength/Weight
Firing frequency	Node output

ARTIFICIAL NEURAL NETWORK

- **Artificial Neural Network (ANNs)** are programs designed to solve any problem by trying to mimic the structure and the function of our nervous system.
- Neural networks are based on simulated neurons, Which are joined together in a variety of ways to form networks.
- Neural network resembles the human brain in the following two ways: -

* A neural network acquires knowledge through learning.

*A neural network's knowledge is stored within the interconnection strengths known as synaptic weight.

ARTIFICIAL NEURAL NETWORK MODEL

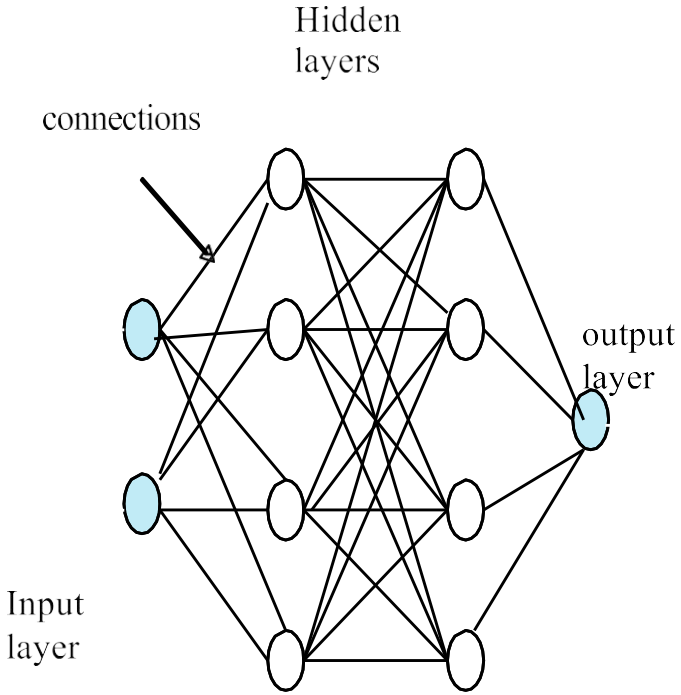


Fig 1: artificial neural network model

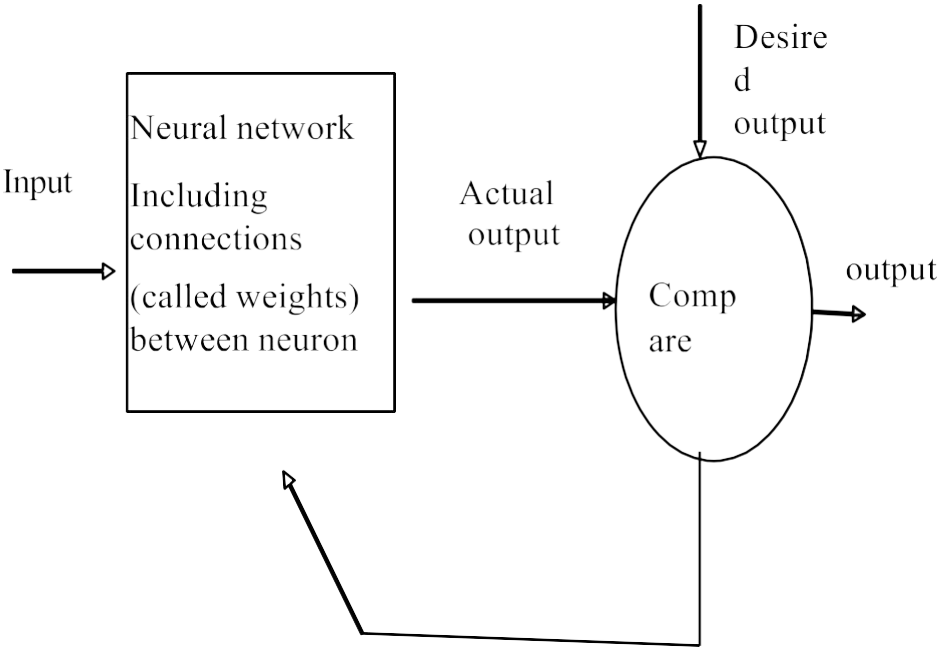
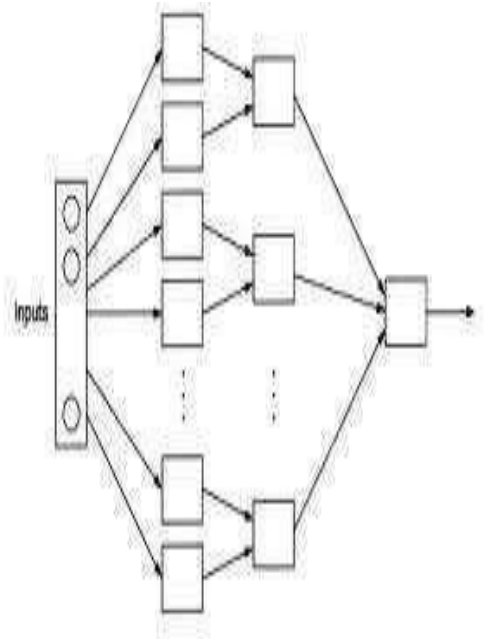


Figure showing adjust of neural network

Fig : Modular neural network



Many problems are best solved using neural networks whose architecture consists of several modules, with sparse interconnections between them. Modules can be organized in several different ways as Hierarchial organization, Successive refinement, Input modularity

LEARNING

- Neurons in an animal's brain are "hard wired". It is equally obvious that animals, especially higher order animals, learn as they grow.
- How does this learning occur?
- What are possible mathematical models of learning?
- In artificial neural networks, learning refers to the method of modifying the weights of connections between the nodes of a specified network.
- The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training.

THE BACKPROPAGATION ALGORITHM

- The backpropagation algorithm (Rumelhart and McClelland, 1986) is used in layered feed-forward Artificial Neural Networks.
- Back propagation is a multi-layer feed forward, supervised learning network based on gradient descent learning rule.
- we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated.
- The idea of the backpropagation algorithm is to reduce this error, until the Artificial Neural Network *learns* the training data.

- The activation function of the artificial neurons in ANNs implementing the backpropagation algorithm is a weighted sum (the sum of the inputs x_i multiplied by their respective weights w_{ji})

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji}$$

- The most common output function is the sigmoidal function:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A(\bar{x}, \bar{w})}}$$

- Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. We can define the error function for the output of each neuron:

$$E_j(\bar{x}, \bar{w}, d) = \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2$$

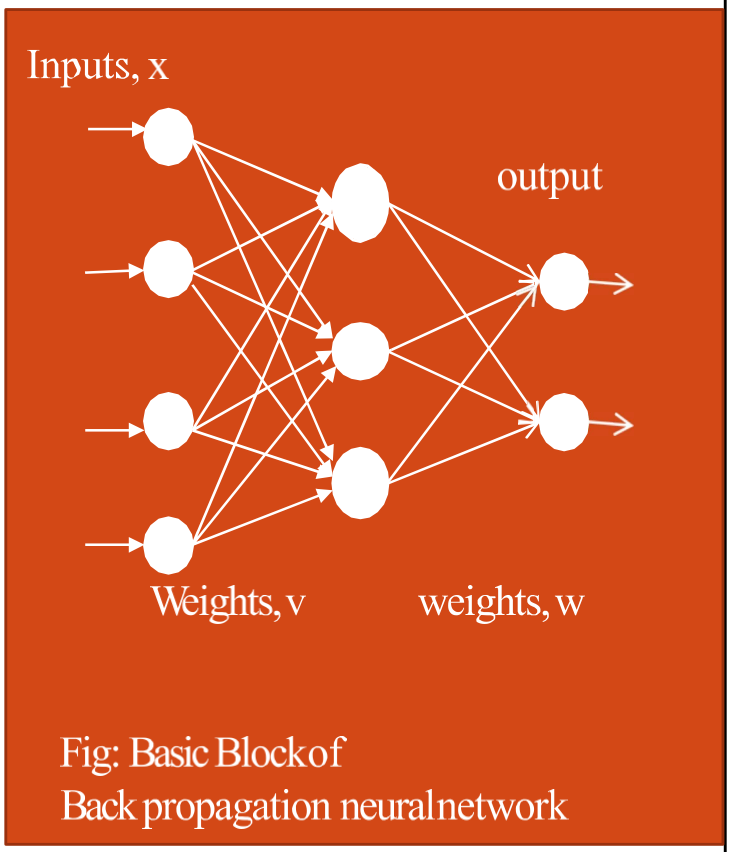


Fig: Basic Block of Back propagation neural network

CONTD…

•The backpropagation algorithm now calculates how the error depends on the output, inputs, and weights.

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}$$

the adjustment of each weight (Δwji) will be the negative of a constant eta (η) multiplied by the dependance of the "wji" previous weight on the error of the network.

•First, we need to calculate how much the error depends on the output

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j)$$

•Next, how much the output depends on the activation, which in turn depends on the weights

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j}\frac{\partial A_j}{\partial w_{ji}} = O_j(1-O_j)x_i$$

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1-O_j)x_i$$

ADVANTAGES

- Itinvolves human like thinking.
- They handle noisy or missing data.
- They can work with large number of variables or parameters.
- They provide general solutions with good predictive accuracy.
- System has got property of continuous learning.
- They deal with the non-linearity in the world in which we live.