Artificial Neural Networks and Fuzzy System

FUZZY LOGIC
VERSUS
NEURAL NETWORK

FUZZY LOGIC	NEURAL NETWORK
Reasoning methodology that resembles the human decision making and deals with vague and imprecise information	A system which is inspired by biological neurons in the human brain that can perform computing tasks faster
Helps to perform pattern recognition and classification tasks	Helps to perform prediction, recognition and classification tasks
Simpler than neural network	Complex than fuzzy logic

Visit www.PEDIAA.com

ANNFS

Artificial neural network	Fuzzy inference system
Difficult to use prior rule knowledge	prior rule base can be incorporated
Based on Learning	Can not learn
Black box	Interpretable(If – Then rules)
Complicated learning algorithms	Simple interpretation and implementation
Difficult to extract knowledge	Knowledge can be available

There are similarities between fuzzy logic and neural networks:

- estimate functions from sample data
- do not require mathematical model
- are dynamic systems
- can be expressed as a graph which is made up of nodes and edges
- convert numerical inputs to numerical outputs
- process inexact information inexactly
- have the same state space
- produce bounded signals
- a set of *n* neurons defines *n*-dimensional fuzzy sets
- learn some unknown probability function
- can act as associative memories
- can model any system provided the number of nodes is sufficient.

The main dissimilarity between fuzzy logic and neural network is that FL uses heuristic knowledge to form rules and tunes these rules using sample data, whereas NN forms "rules" based entirely on data.

AI Vs ANN

- **Artificial Intelligence**
  - Intelligence comes by designing.
  - Response time is consistent.
  - Knowledge is represented in explicit and abstract form.
  - Symbolic representation.
  - Explanation regarding any response or output i.e. it is derived from the given facts or rules.

- **Artificial Neural Network**
  - Intelligence comes by Training.
  - Response time is inconsistent.
  - Knowledge is represented in terms of weight that has no relationship with explicit and abstract form of knowledge.
  - Numeric representation.
  - No explanation for results or output received.

AI Vs ANN

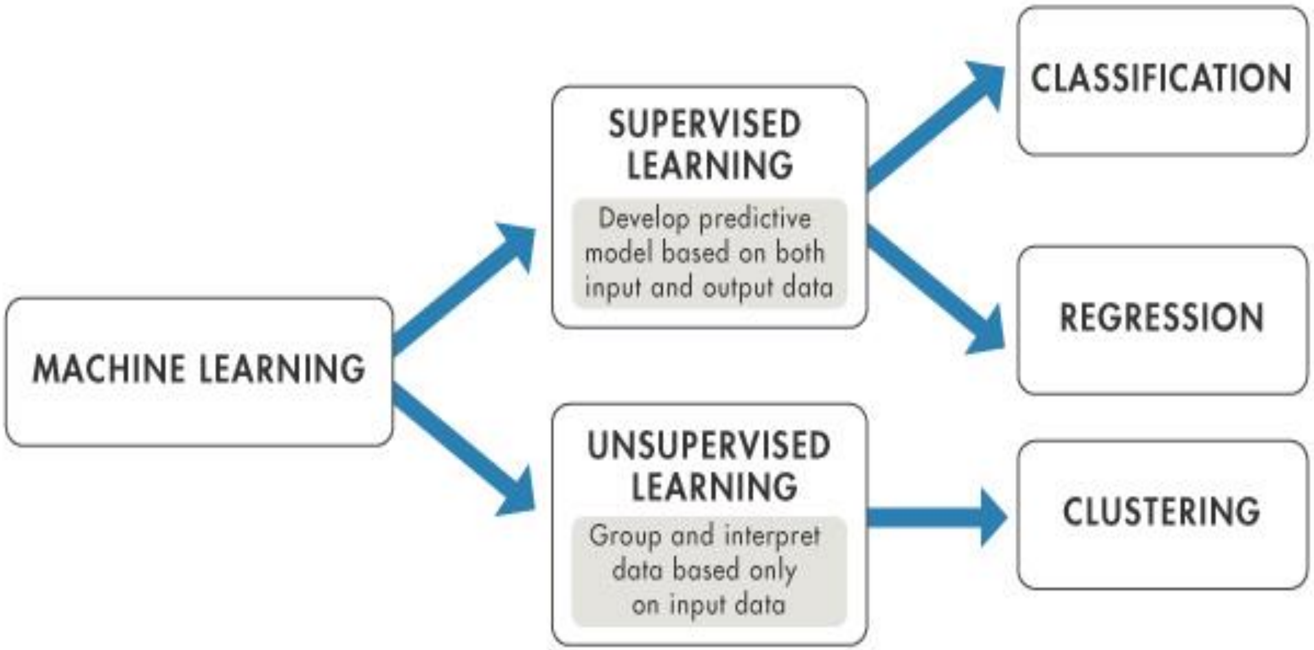
- **Artificial Intelligence**
  - Errors can be explicitly corrected by the modification of design, etc.
  - Sequential Processing
  - It is not a fault tolerant system.
  - Processing speed is slow.

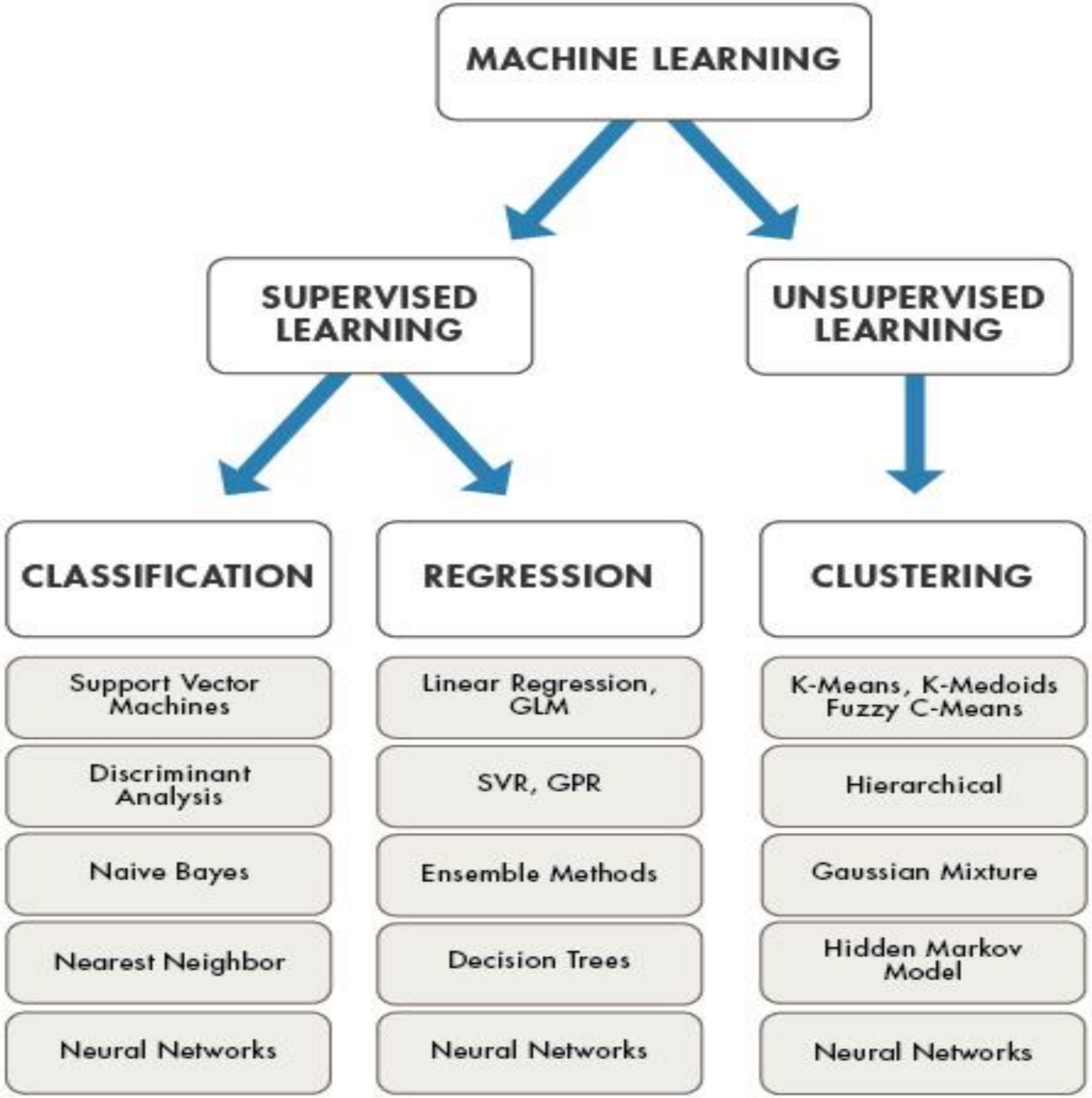
- **Artificial Neural Network**
  - Errors can't be explicitly corrected. Network itself modifies the weights to reduce the errors and to produce the correct output.
  - Distributed Processing.
  - Partially fault tolerant system.
  - Due to dedicated hardware the processing speed is fast.

Module 1.1: Basics of Neural Network

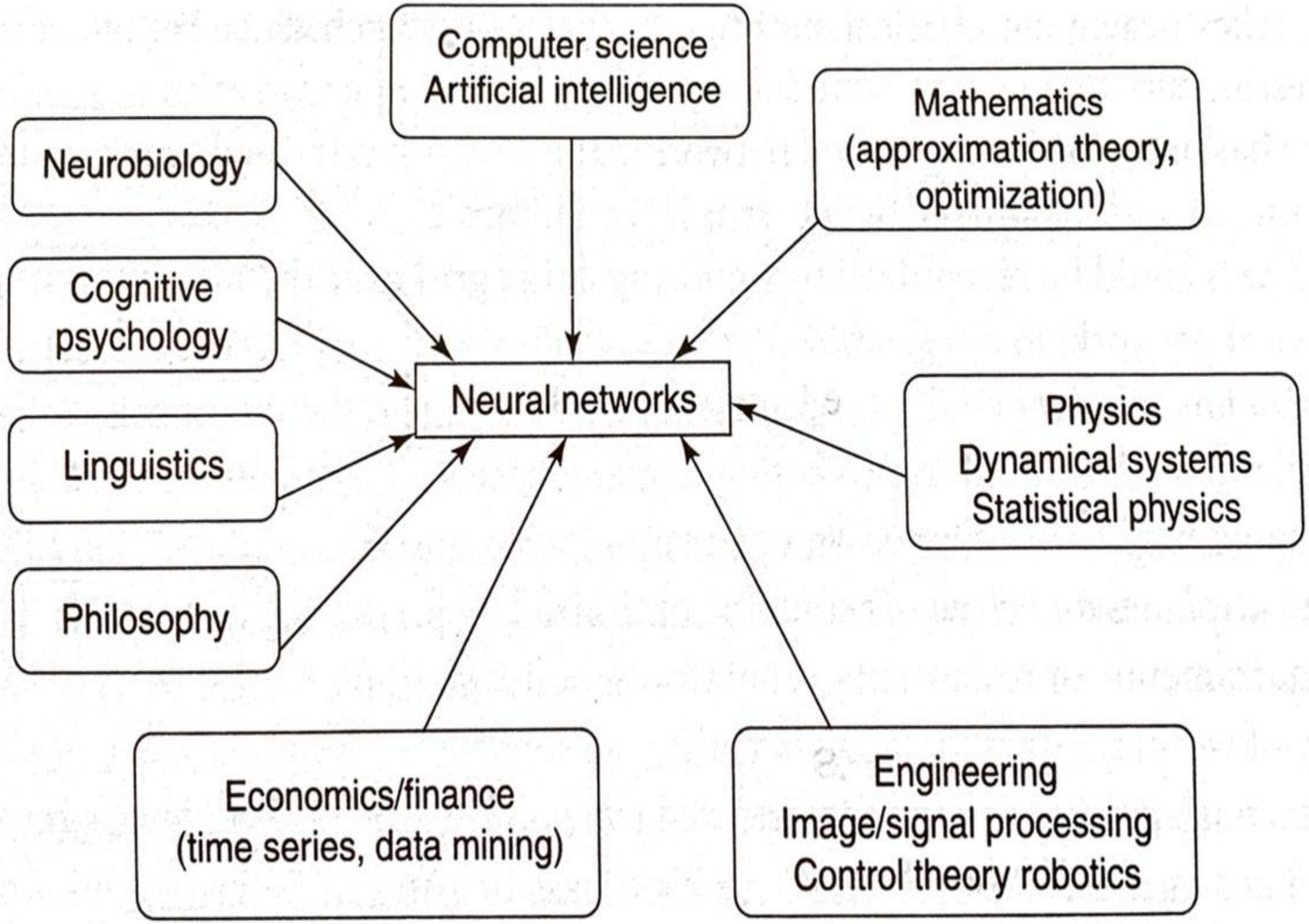
- **Definition, why and how are neural networks being used in solving problems**
- **Human biological neuron**
- **Artificial Neuron**
- **Applications of ANN**
- **Comparison of ANN vs BNN**

Machine Learning




MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Neural Networks
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Hidden Markov Model
Neural Networks

Interdisciplinary

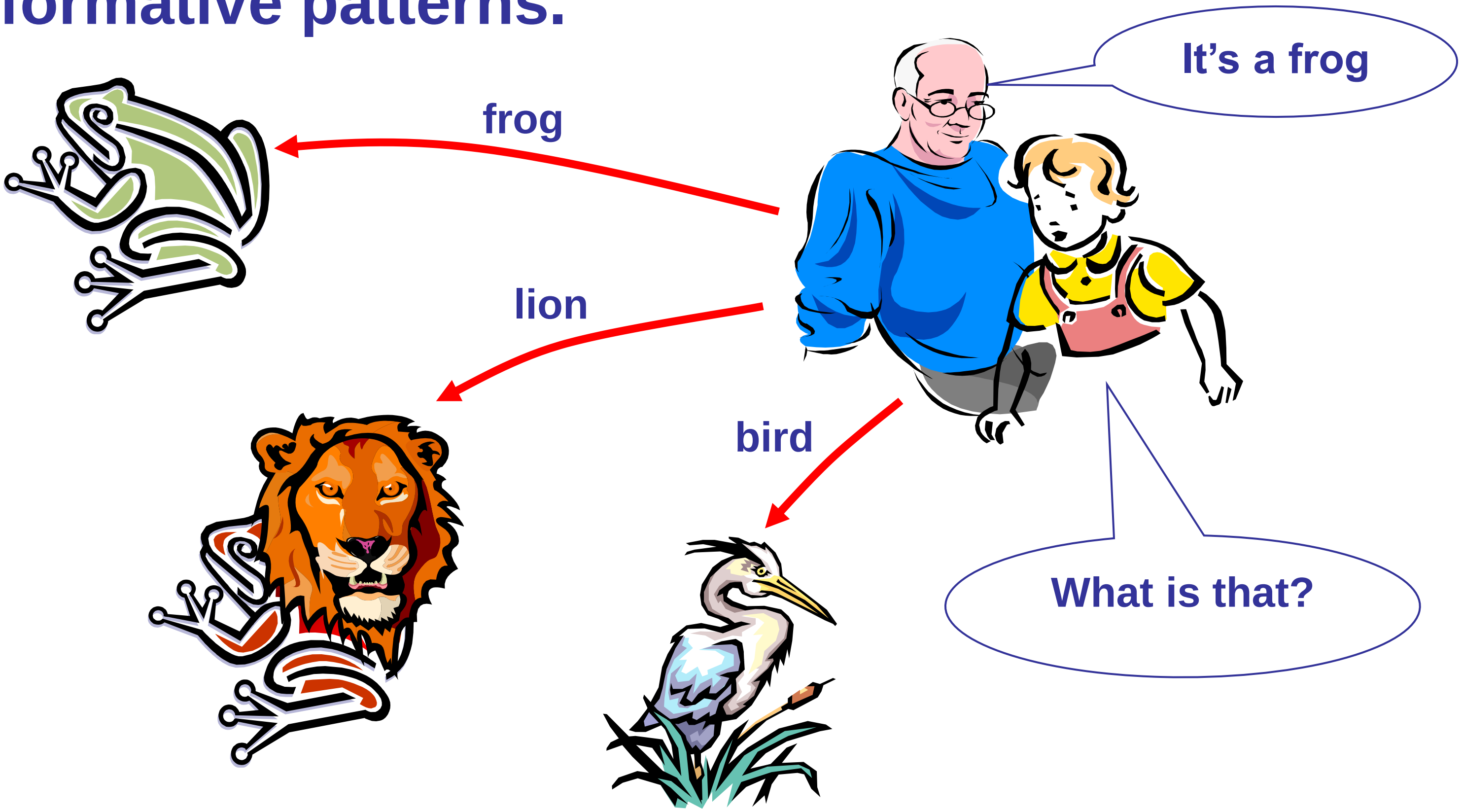


Neural networks to the rescue…

- **Neural network:** *information processing paradigm inspired by biological nervous systems, such as our brain*
- Structure: large number of highly interconnected processing elements (*neurons*) working together
- Like people, they learn *from experience* (by example)

The idea of ANNs..?

- **NNs learn relationship between cause and effect or organize large volumes of data into orderly and informative patterns.**



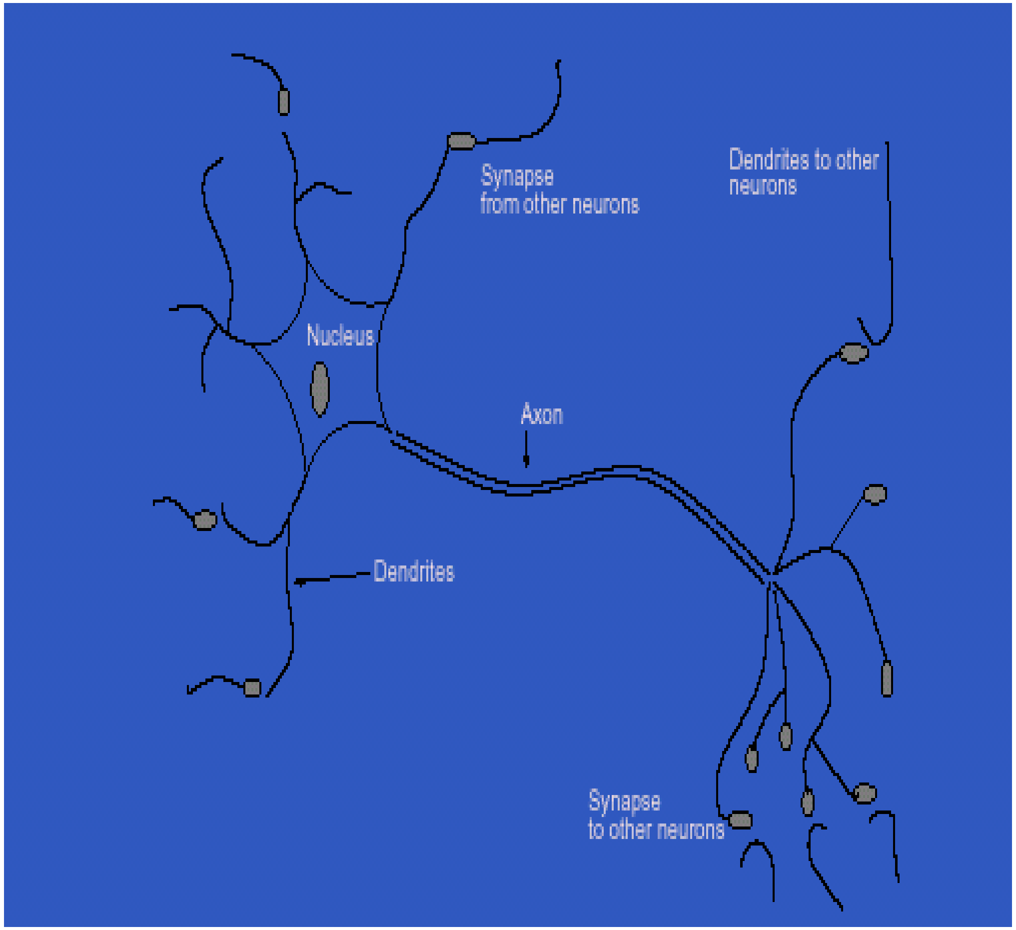
Definition of ANN

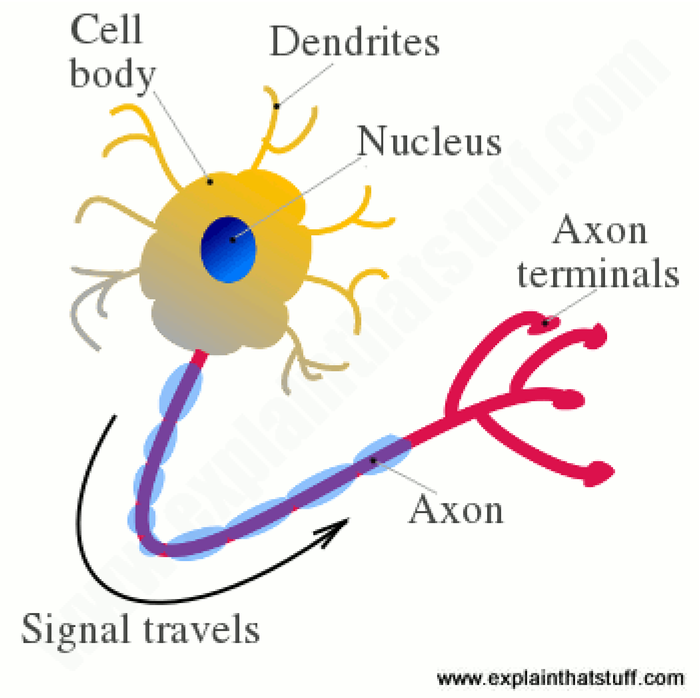
“Data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain”

(Tsoukalas & Uhrig, 1997).

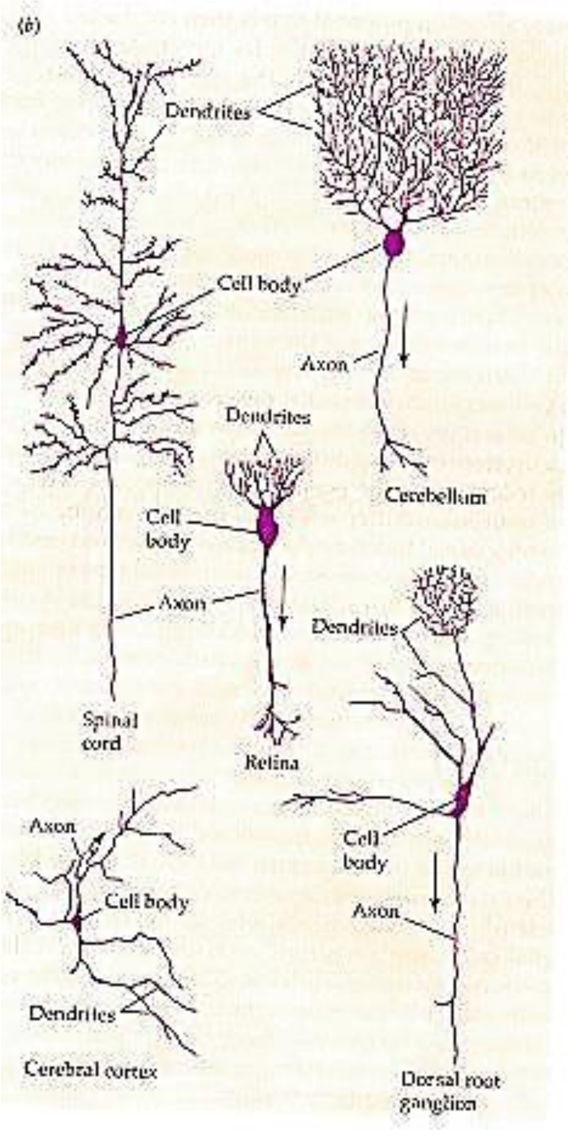
Inspiration from Neurobiology

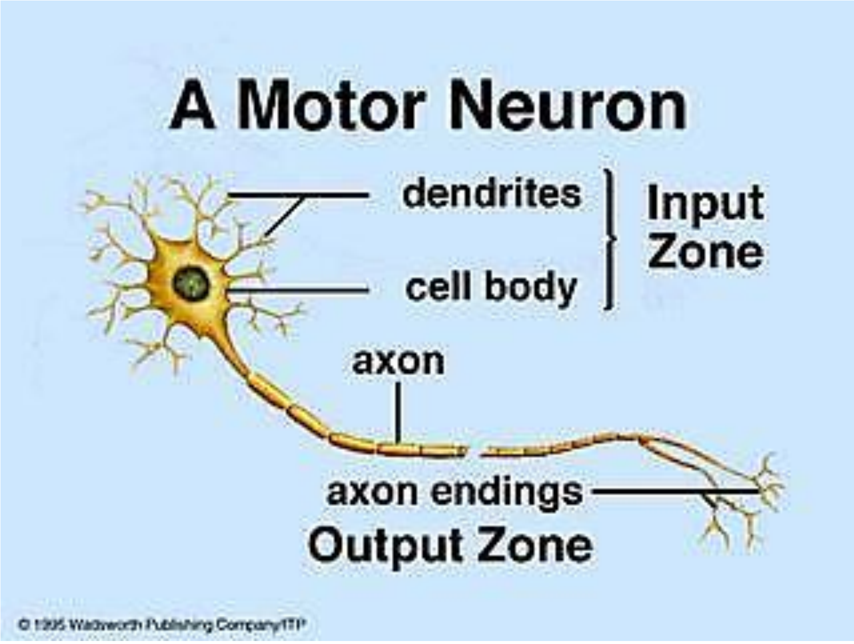
Human Biological Neuron





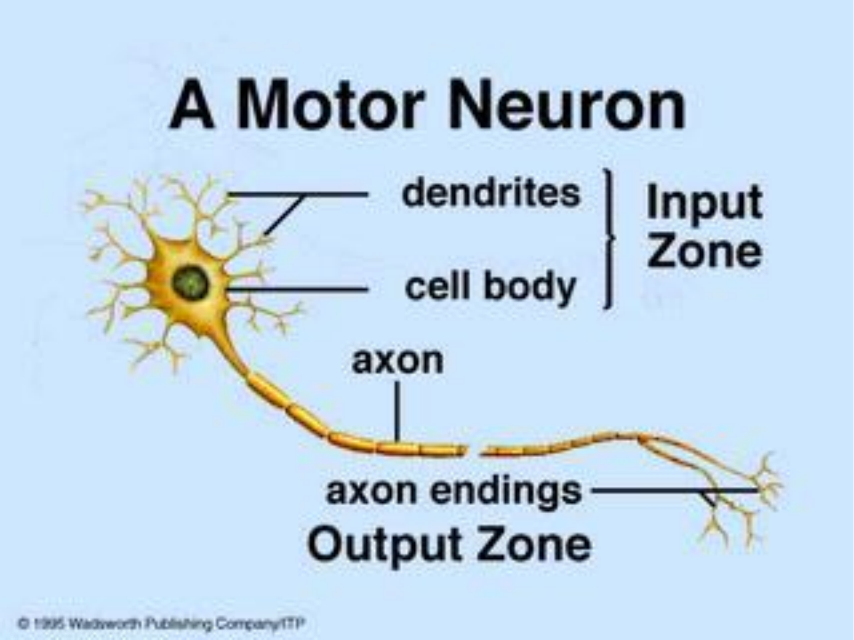
Biological Neural Networks





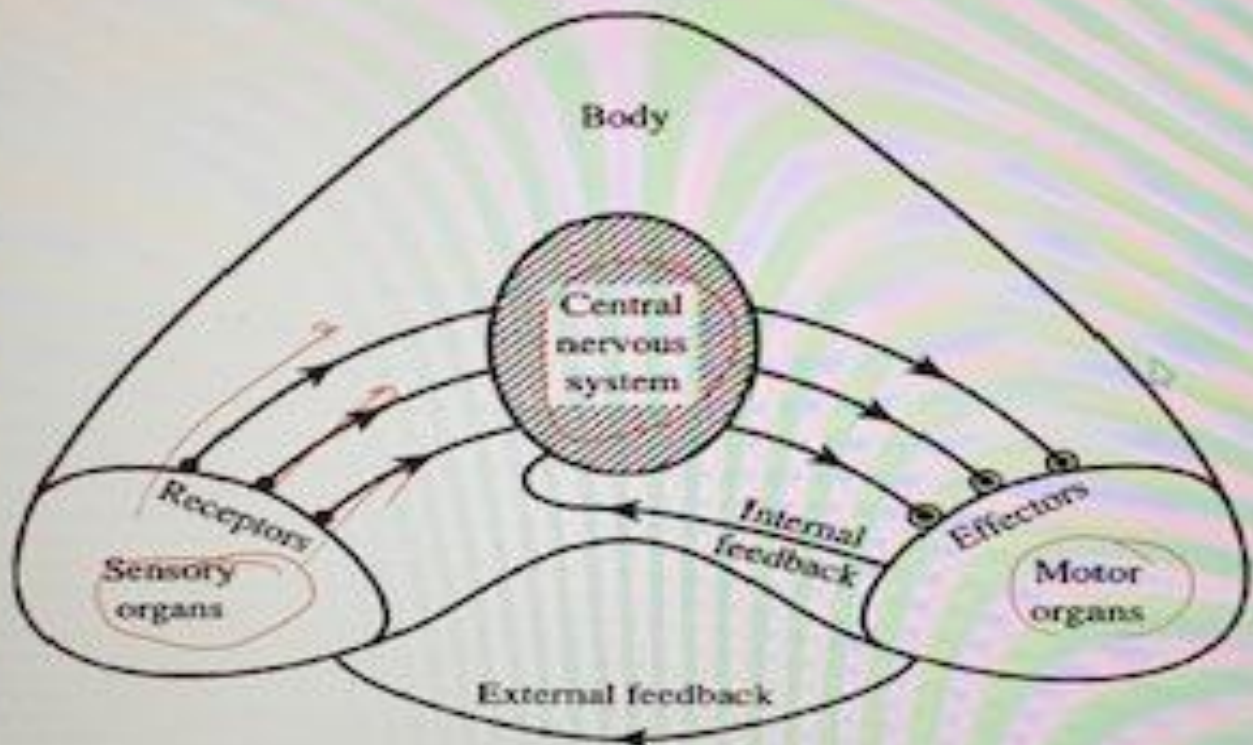
Biological neuron

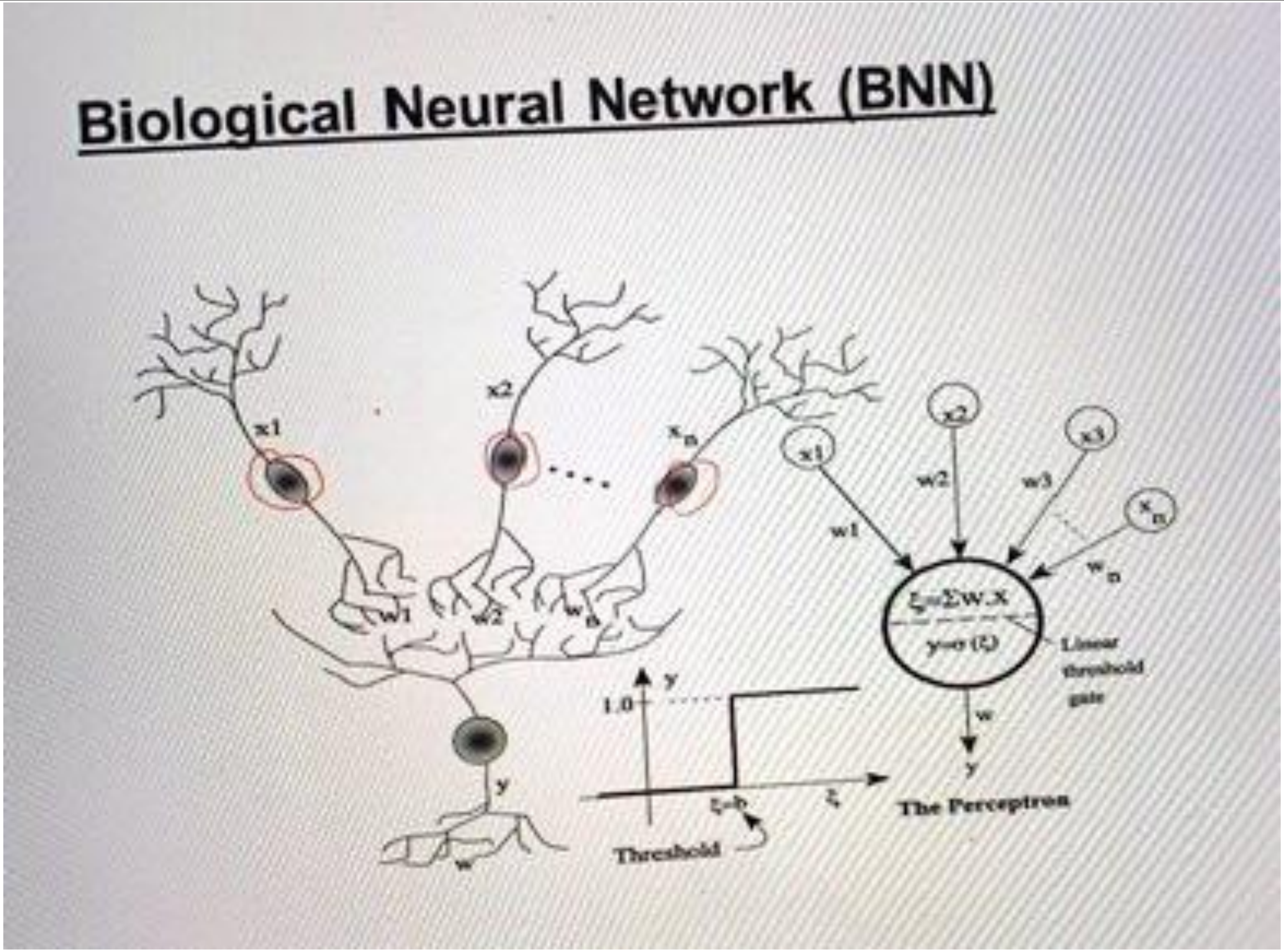
Biological Neural Networks



- A biological neuron has three types of main components; dendrites, soma (or cell body) and axon.
- Dendrites receives signals from other neurons.
- The soma, sums the incoming signals. When sufficient input is received, the cell fires; that is it transmit a signal over its axon to other cells.

Information Processing in nervous system




Biological Neural Network (BNN)
x1
x2
x_n
w1
w2
w_n
y
w
1.0
Threshold
The Perceptron
x1
x2
x3
x_n
w1
w2
w3
w_n
Linear
threshold
gate

BRAIN COMPUTATION

The **human brain** contains about 10 billion nerve cells, or neurons. On average, each neuron is connected to other neurons through approximately 10,000 synapses.

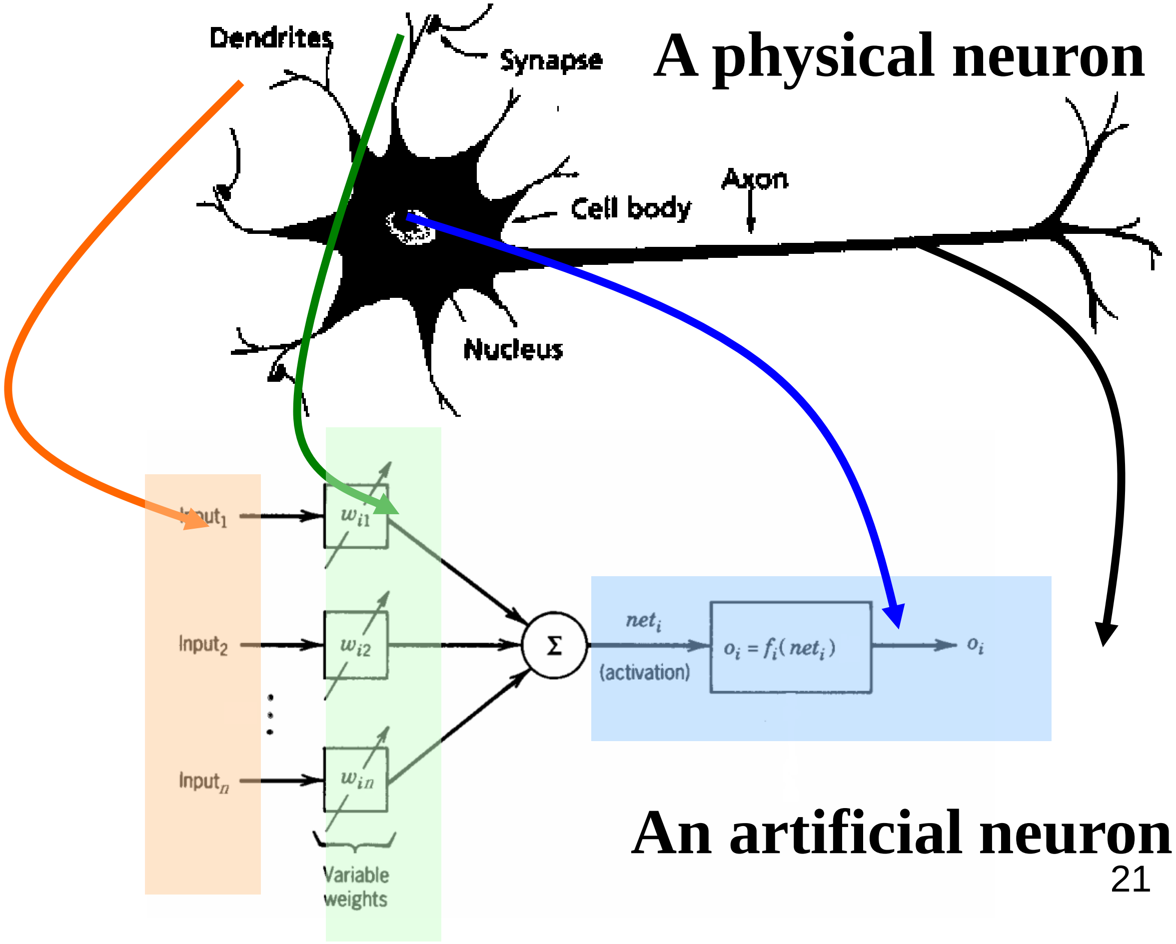
	processing elements	element size	energy use	processing speed	style of computation	fault tolerant	learns	intelligent, conscious
	10^{14} synapses	10^{-6} m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^{8} transistors	10^{-6} m	30 W (CPU)	10^{9} Hz	serial, centralized	no	a little	not (yet)

Artificial Neurons

- ANN is an information processing system that has certain performance characteristics in common with biological nets.
- Several key features of the processing elements of ANN are suggested by the properties of biological neurons:
    1. The processing element receives many signals.
    2. Signals may be modified by a weight at the receiving synapse.
    3. The processing element sums the weighted inputs.
    4. Under appropriate circumstances (sufficient input), the neuron transmits a single output.
    5. The output from a particular neuron may go to many other neurons.

Artificial Neurons

- From experience: examples / training data
- Strength of connection between the neurons is stored as a weight-value for the specific connection.
- Learning the solution to a problem = changing the connection weights

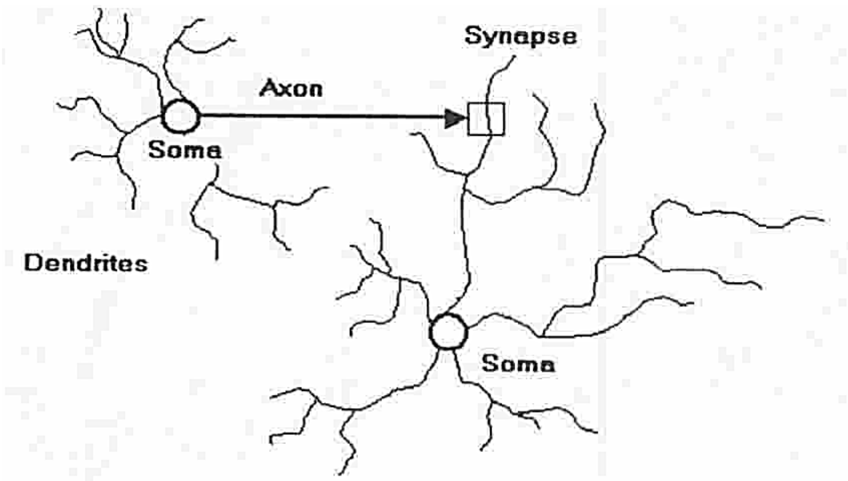
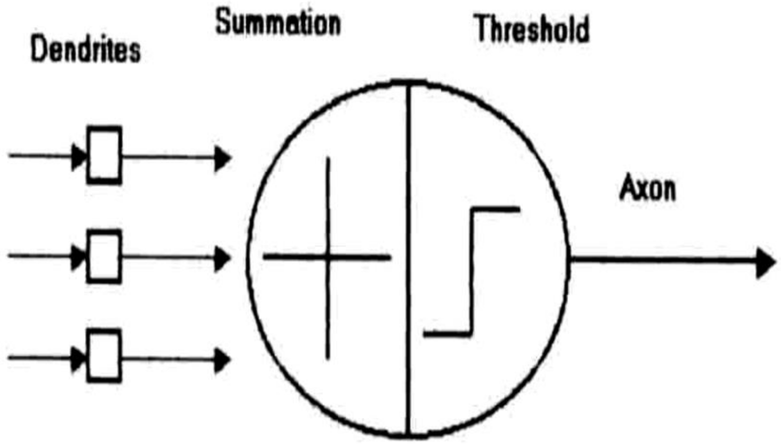


Artificial Neurons

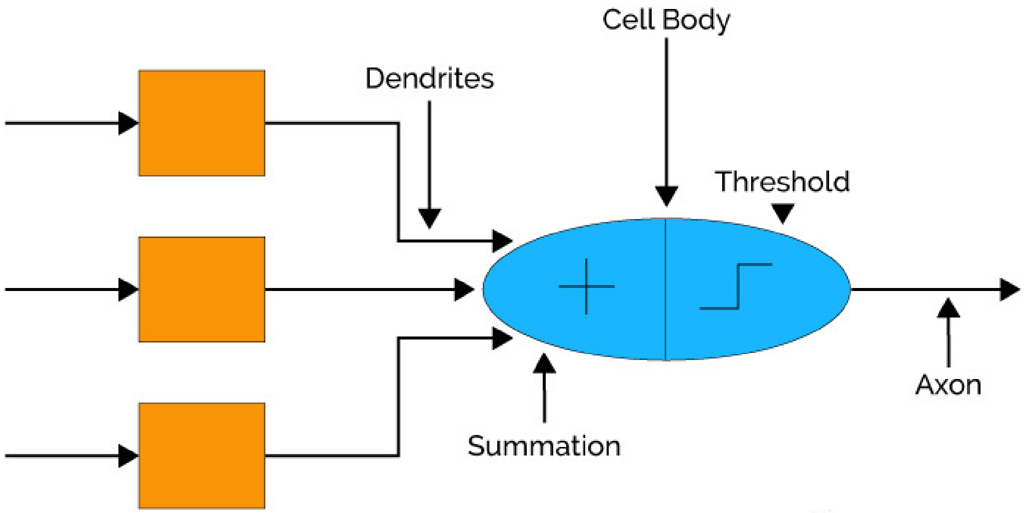
- **ANNs have been developed as generalizations of mathematical models of neural biology, based on the assumptions that:**

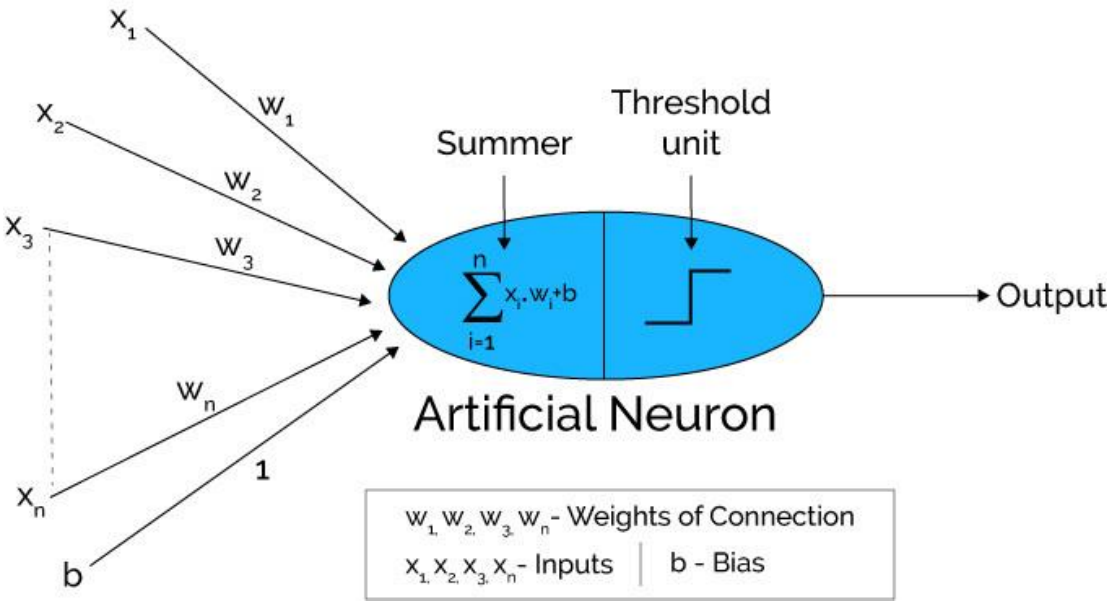
  1. **Information processing occurs at many simple elements called neurons.**
  2. **Signals are passed between neurons over connection links.**
  3. **Each connection link has an associated weight, which, in typical neural net, multiplies the signal transmitted.**
  4. **Each neuron applies an activation function to its net input to determine its output signal.**

Brain Vs Computer

Physical Neuron	Artificial Neuron
	
Cell, Dendrites, Soma, Axon	Neurons, Weights or interconnections, Net input, Output

Brain Vs Computer





Brain Vs Computer

❖ **Terminology**

BNN	**ANN**
Cell	Neuron
Dendrites	Weights or interconnection
Soma	Net input
Axon	Output

❖ **Comparison could be made on the basis of the following criteria**

- **Speed**
- **Processing**
- **Size and complexity**
- **Storage capacity**
- **Tolerance**
- **Control mechanism**

Brain Vs Computer

Comparison could be made on the basis of the following criteria

Criteria	BNN	ANN
Speed (execution time) **Processing** **Size and complexity**	Few millisecond Parallel High (neurons 10^{11} Interconnections 10^{14}	Few nanoseconds Parallel (fast) Based on chosen application & network design
Storage capacity(memory)	Stored in interconnection ie synapes hence new information can be added	Stored in its contiguous memory location, it may overload or destroy older memory
Tolerance	Has fault tolerant capability	No fault tolerant capability
Control mechanism	No, depends on active chemical present	Simple as it has CPU for transfer and control

ANN process the following characteristics

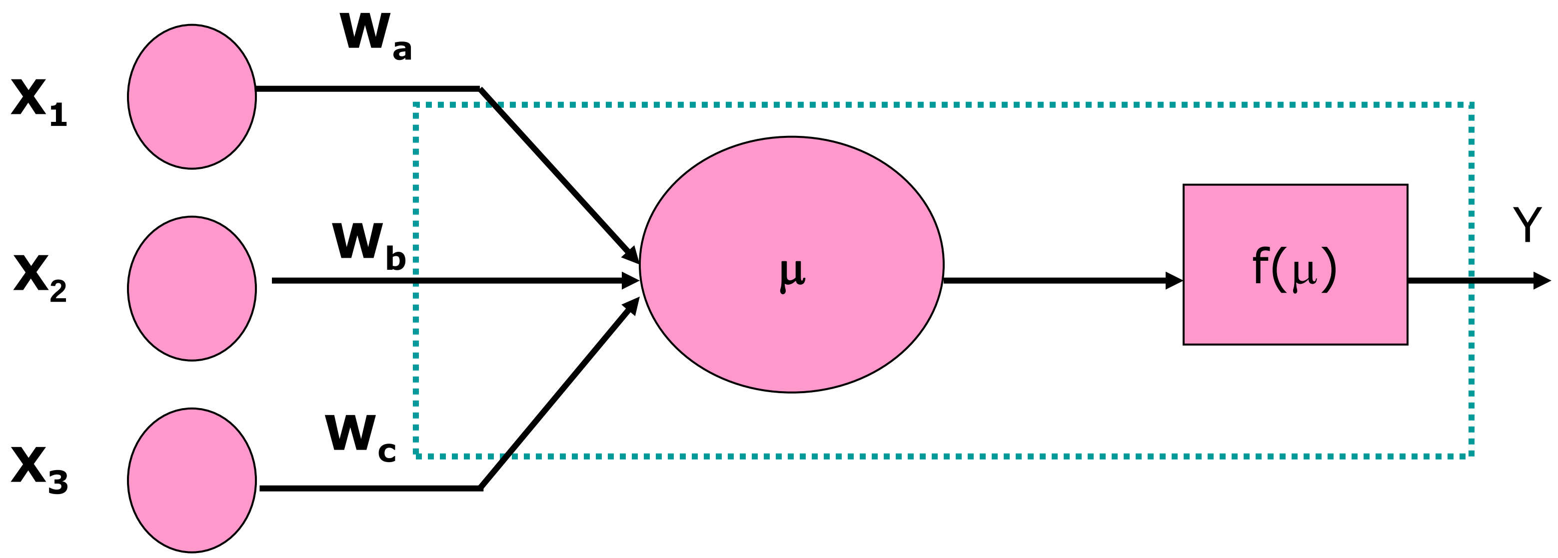
- ➢It’s a Neurally implemented mathematical model
- ➢There exists a large no of highly interconnected processing elements called neurons in an ANN
- ➢The interconnections with their weighted linkages hold the informative knowledge
- ➢The input signals arrive at the processing elements through connections and connecting weights
- ➢The processing elements of the ANN have the ability to learn, recall and generalize from the given data by adjusting the weights
- ➢The computational power can be demonstrated only by the collective behavior of neurons
- ➢ANN is a

•connected model,

•parallel distributed processing models, self organizing systems,

•Neuro computing systems and neuro-morphic systems

Model Of A Neuron



Input units (dendrite)

Connection weights (synapse)

Summing function

computation

(soma)

(axon)

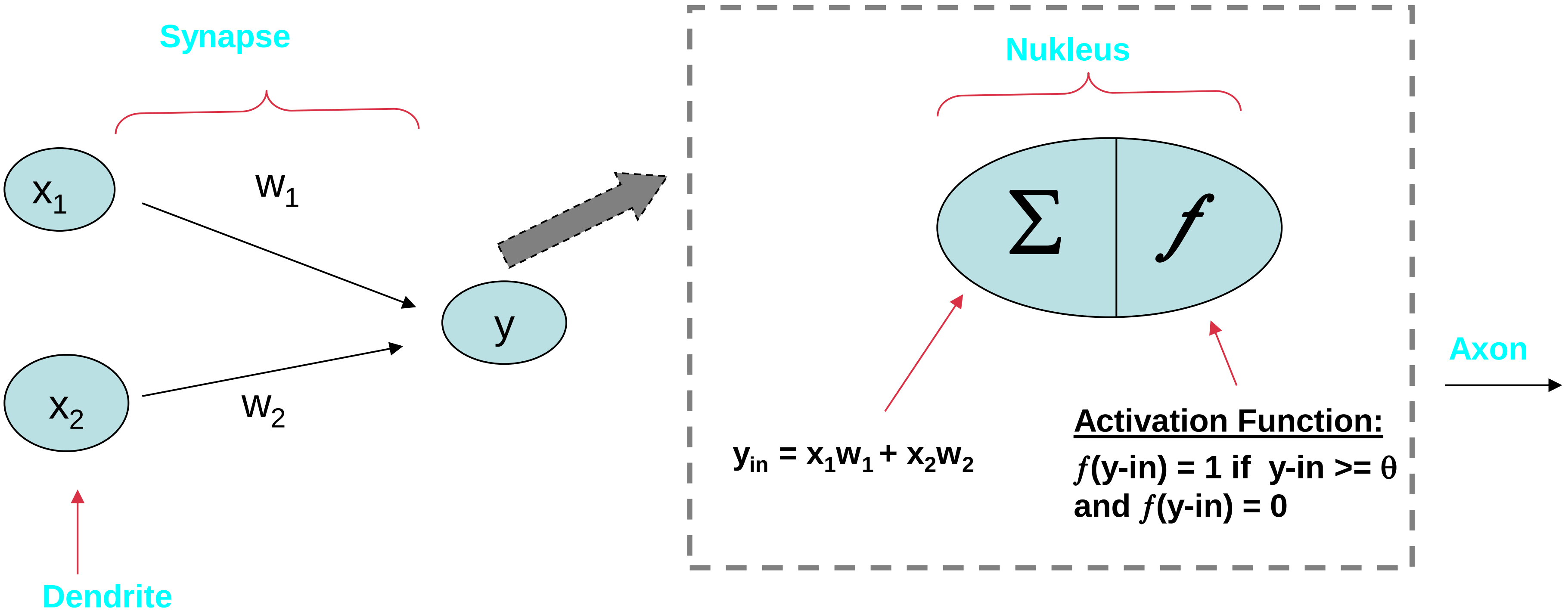
ANN Process the following Characteristics

- A neural net consists of a large number of highly interconnected processing elements called neurons, units, cells or nodes.
- Each neuron is connected to other neurons by means of directed communication links, each with associated weight.
- The interconnections with their weighted linkages hold the informative knowledge.
- The input signals arrive at the processing elements through connection and connecting weights.
- The weight represent information being used by the net to solve a problem.

- Each neuron has an internal state, called its activation or activity level, which is a function of the inputs it has received.
- Typically, a neuron sends its activation as a signal to several other neurons.
- It is important to note that a neuron can send only one signal at a time, although that signal is broadcast to several other neurons.
- Neural networks are configured for a specific application, such as pattern recognition or data classification, through a learning process
- In a biological system, learning involves adjustments to the synaptic connections between neurons

➔ same for artificial neural networks (ANNs)

Artificial Neural Network



-A neuron receives input, determines the strength or the weight of the input, calculates the total weighted input, and compares the total weighted with a value (threshold)

-The value is in the range of 0 and 1

- If the total weighted input greater than or equal the threshold value, the neuron will produce the output, and if the total weighted input less than the threshold value, no output will be produced

History

- 1943 McCulloch-Pitts neurons
- 1949 Hebb's law
- 1958 Perceptron (Rosenblatt)
- 1960 Adaline, better learning rule (Widrow, Huff)
- 1969 Limitations (Minsky, Papert)
- 1972 Kohonen nets, associative memory

- 1977 Brain State in a Box (Anderson)
- 1982 Hopfield net, constraint satisfaction
- 1985 ART (Carpenter, Grossfield)
- 1986 Backpropagation (Rumelhart, Hinton, McClelland)
- 1988 Neocognitron, character recognition (Fukushima)

TABLE 2-2 EVOLUTION OF NEURAL NETWORKS

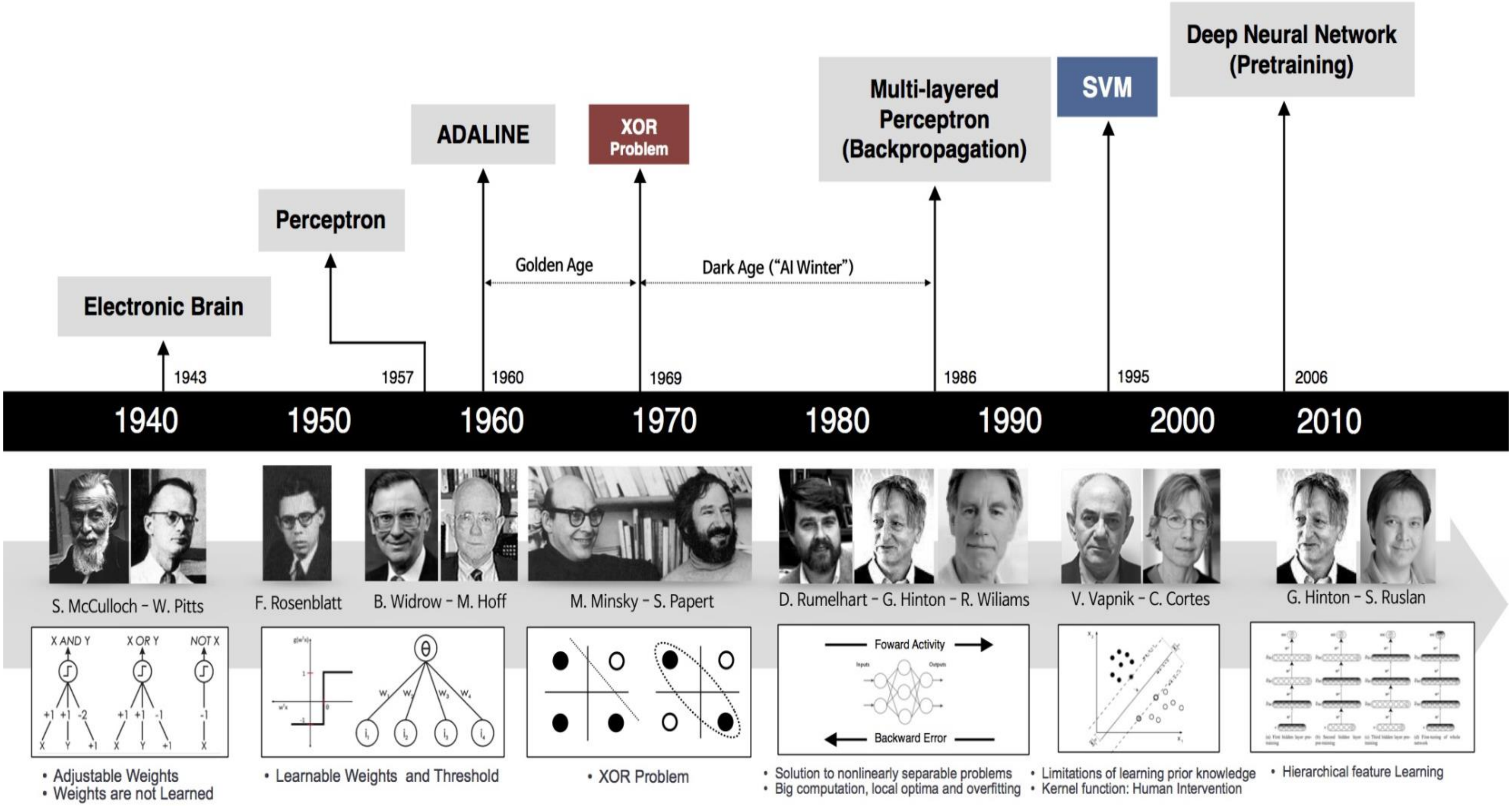
Year	Neural network	Designer	Description
1943	McCulloch and Pitts neuron	McCulloch and Pitts	The arrangement of neurons in this case is a combination of logic functions. Unique feature of this neuron is the concept of threshold.
1949	Hebb network	Hebb	It is based upon the fact that if two neurons are found to be active simultaneously then the strength of the connection between them should be increased.
1958, 1959, 1962, 1988	Perceptron	Frank Rosenblatt, Block, Minsky and Papert	Here the weights on the connection path can be adjusted.
1960	Adaline	Widrow and Hoff	Here the weights are adjusted to reduce the difference between the net input to the output unit and the desired output. The result here is very negligible. Mean squared error is obtained.
1972	Kohonen self-organizing feature map	Kohonen	The concept behind this network is that the inputs are clustered together to obtain a fired output neuron. The clustering is performed by winner-take all policy.
1982, 1984, 1985, 1986, 1987	Hopfield network	John Hopfield and Tank	This neural network is based on fixed weights. These nets can also act as associative memory nets.

(Continued)

TABLE 2-2 EVOLUTION OF NEURAL NETWORKS—CONTINUED

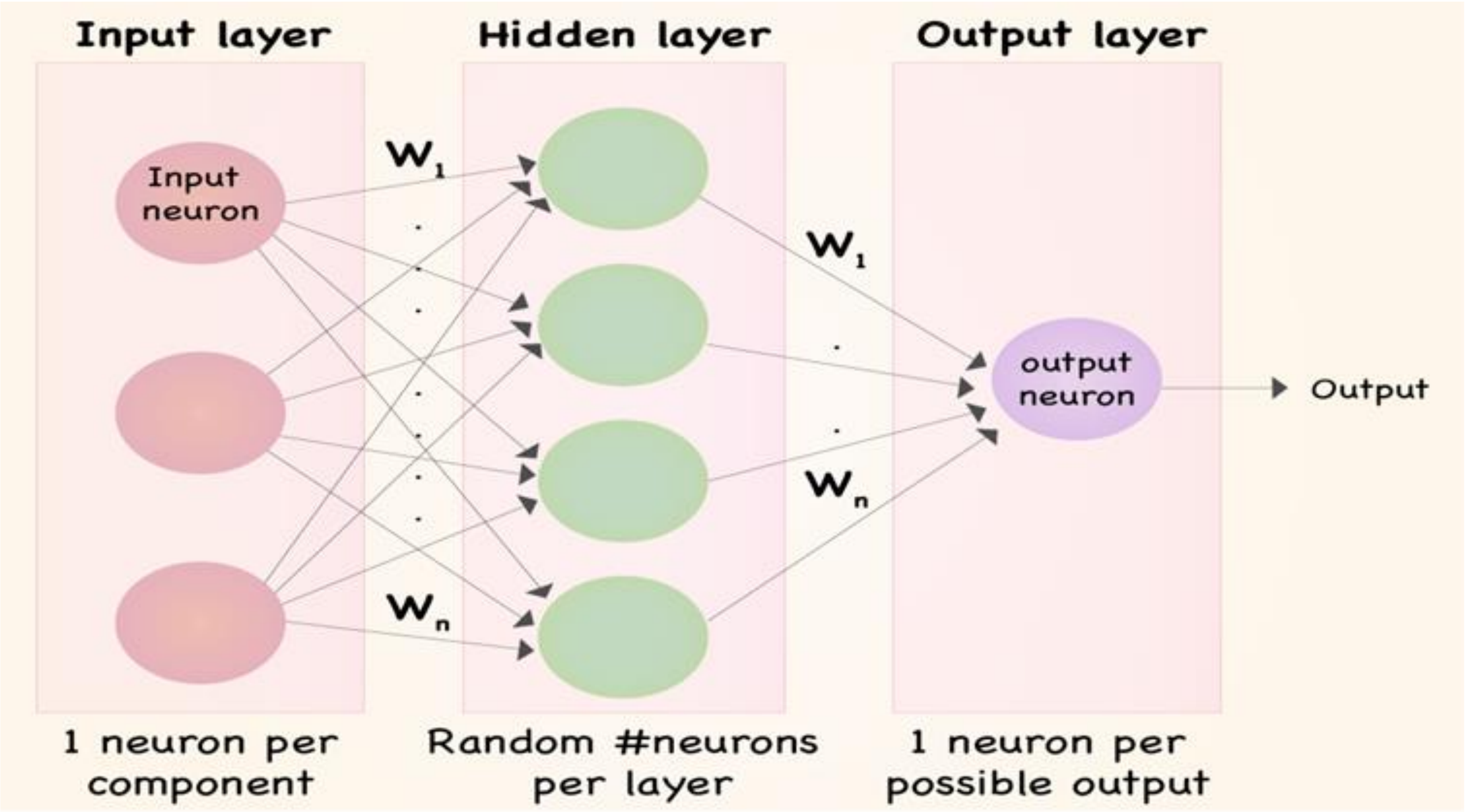
Year	Neural network	Designer	Description
1986	Back-propagation network	Rumelhart, Hinton and Williams	This network is multi-layer with error being propagated backwards from the output units to the hidden units.
1988	Counter-propagation network	Grossberg	This network is similar to the Kohonen network; here the learning occurs for all units in a particular layer, and there exists no competition among these units.
1987–1990	Adaptive Resonance Theory (ART)	Carpenter and Grossberg	The ART network is designed for both binary inputs and analog valued inputs. Here the input patterns can be presented in any order.
1988	Radial basis function network	Broomhead and Lowe	This resembles a back propagation network but the activation function used is a Gaussian function.
1988	Neo cognitron	Fukushima	This network is essential for character recognition. The deficiency occurred in cognitron network (1975) was corrected by this network.

In the later years, the discovery of the neural net resulted in the implementation of optical neural nets, Boltzmann machine, spatiotemporal nets, pulsed neural networks and support vector machines.



Source: http://qingkaikong.blogspot.com/2016/11/machine-learning-3-artificial-neural.html

Basic Structure of ANN



Basic Characteristics of ANN

1. The models synaptic interconnections
   – a pattern of connections between neurons

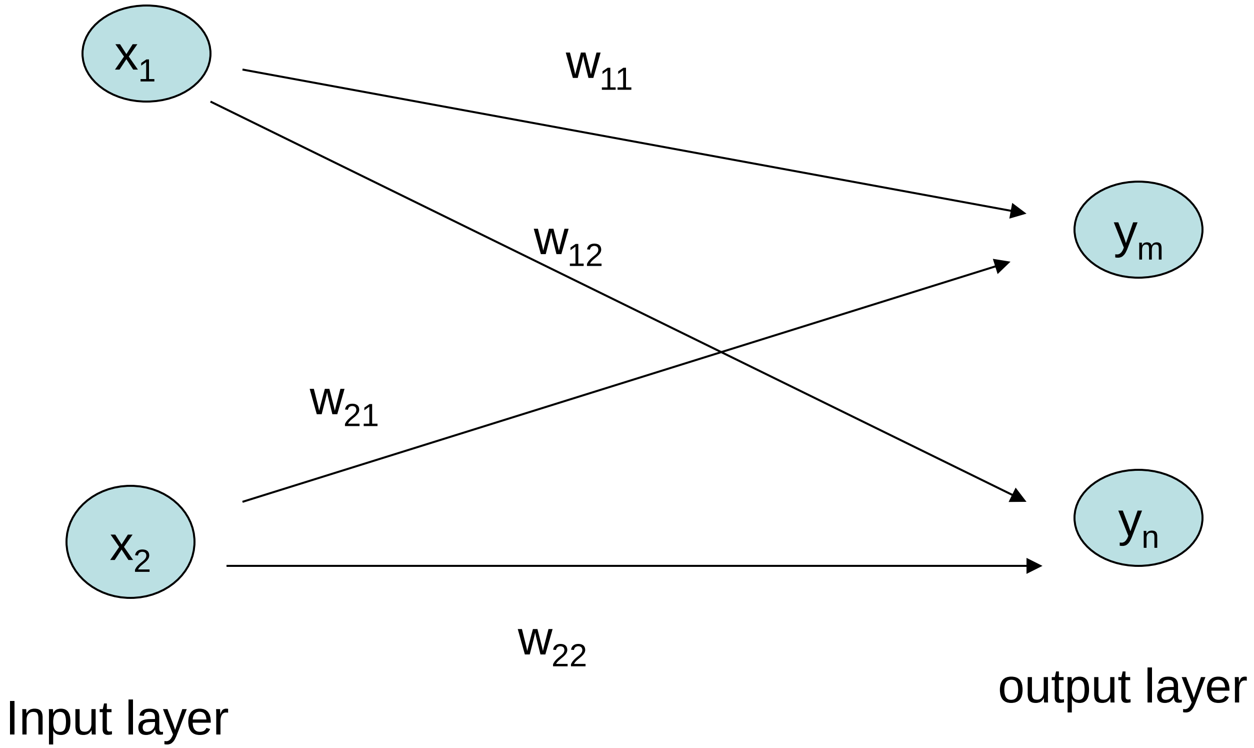
2. The training or learning rules adopted for uploading and adjusting the connection weights
   – a method of determining the connection weights

3. Their Activation Function
   – Function to compute output signal from input signal

Basic characteristics of ANN

1. Architecture
   - a pattern of connections between neurons
     - Single Layer Feedforward
     - Multilayer Feedforward
     - Recurrent
2. Strategy / Learning Algorithm
   - a method of determining the connection weights
     - Supervised
     - Unsupervised
     - Reinforcement
3. Activation Function
   - Function to compute output signal from input signal

1.Architecture: Single Layer Feedforward ANN



Contoh: **ADALINE, AM, Hopfield, LVQ, Perceptron, SOM**

1.Architecture: Single Layer Feedforward ANN

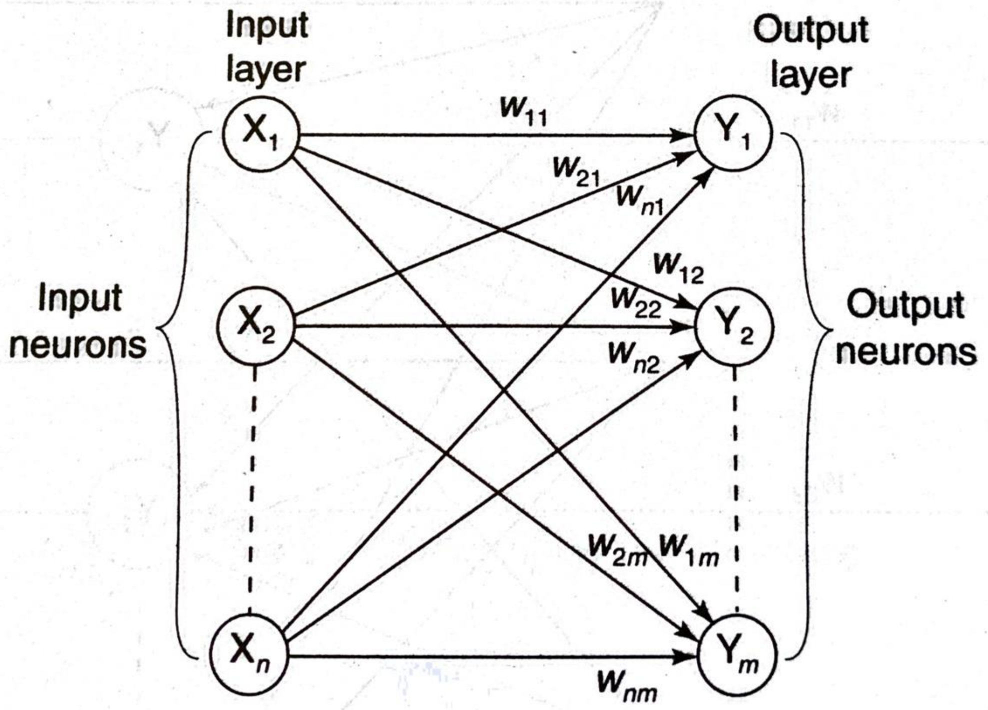


Figure 2-6 Single-layer feed-forward network.

1. Architecture: Multilayer NN Network

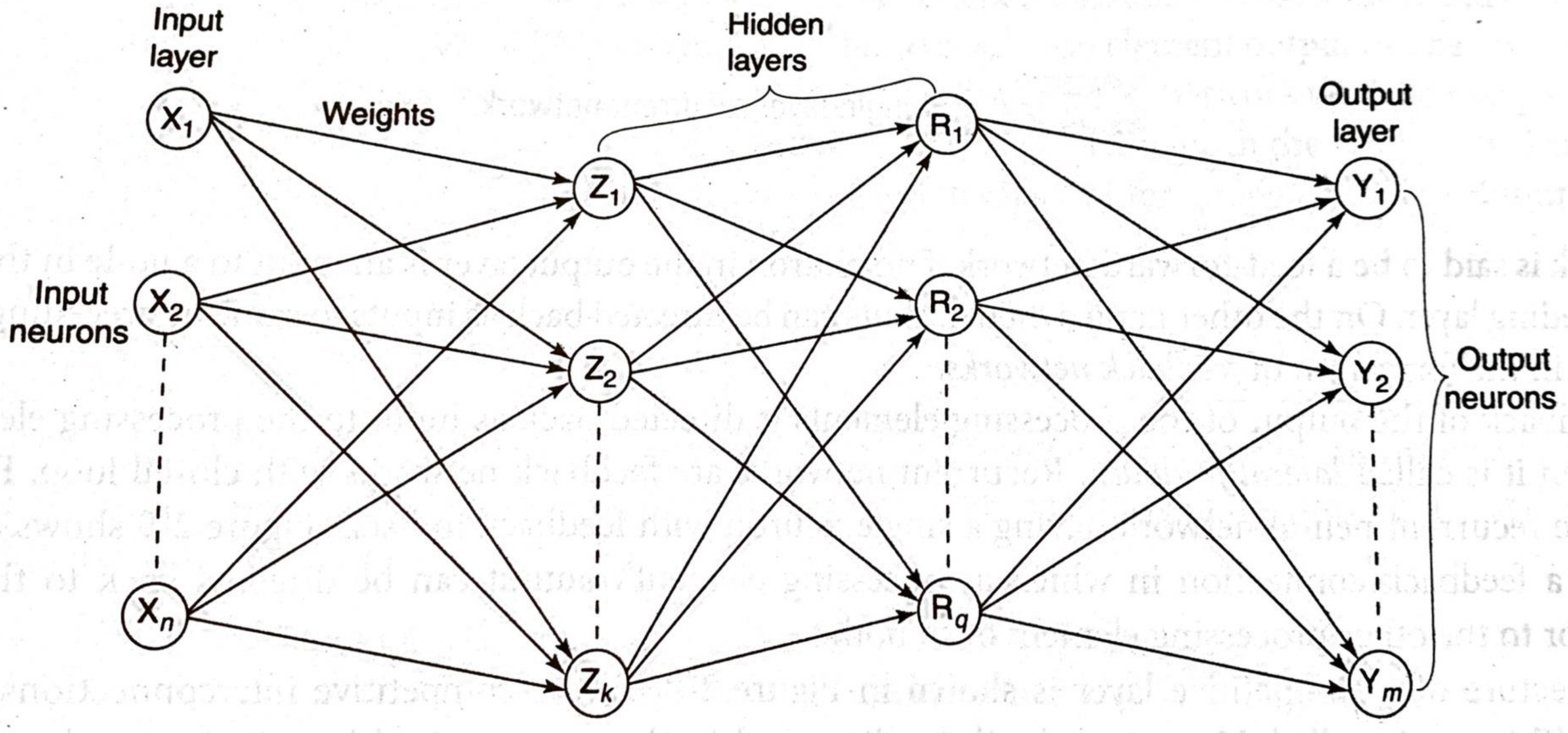
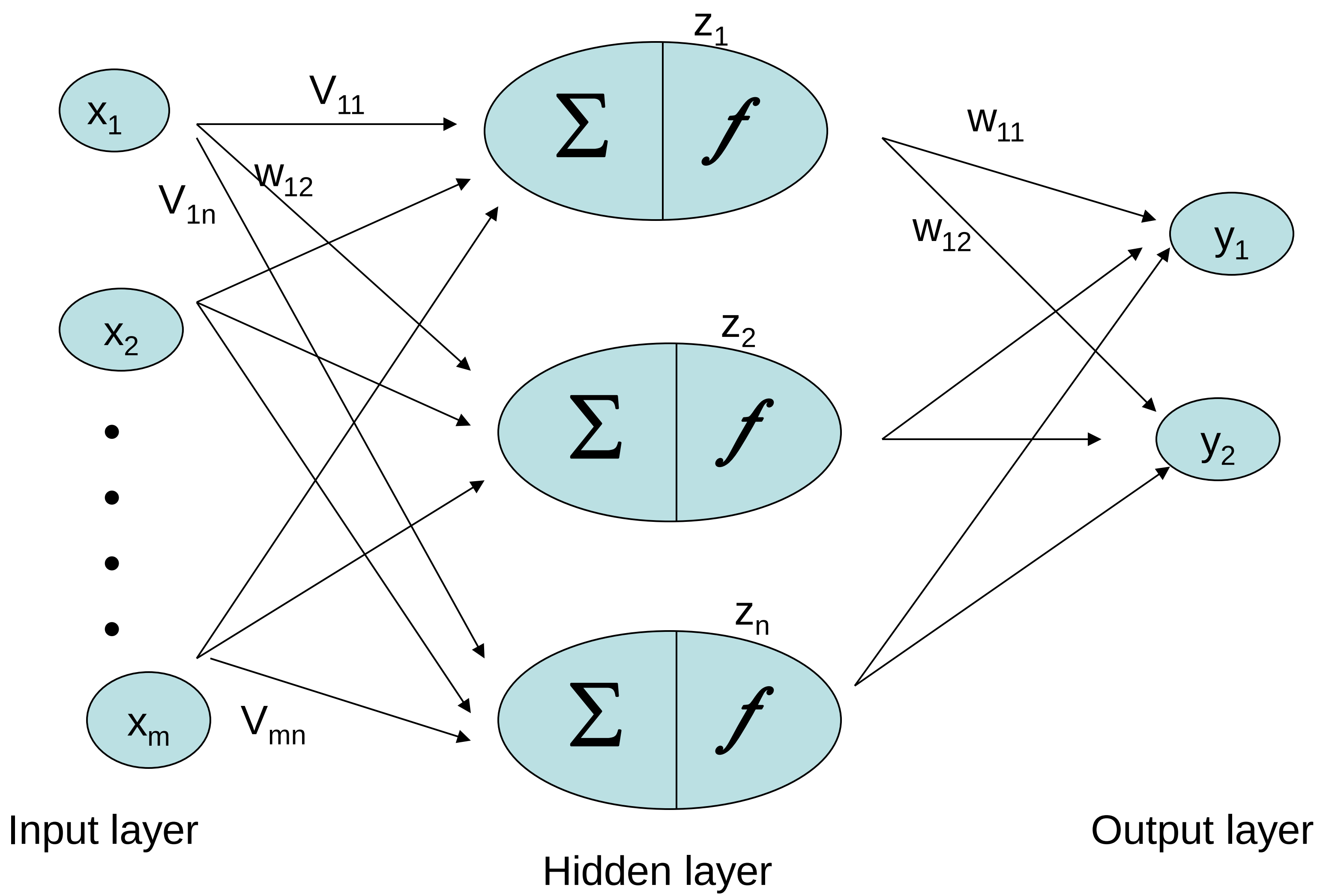


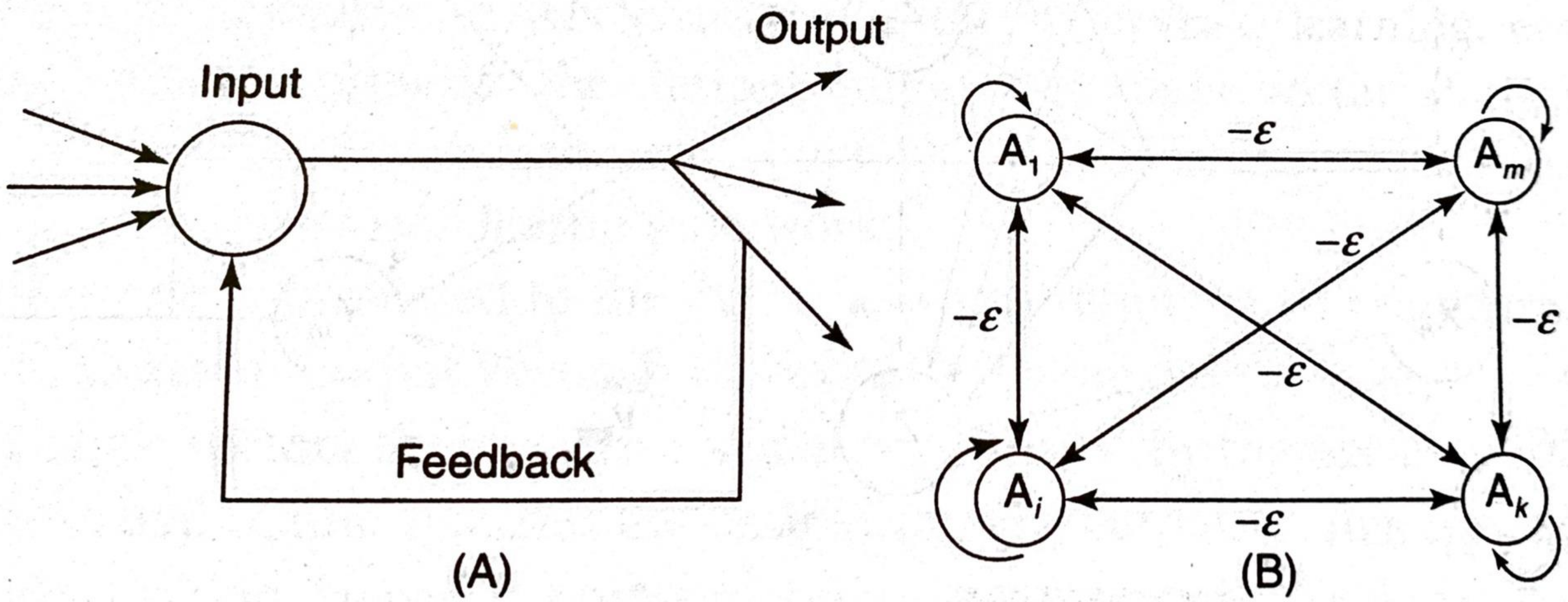
Figure 2-7 Multilayer feed-forward network.

1. Architecture: Multilayer NN Network



Contoh: **CCN, GRNN, MADALINE, MLFF with BP, Neocognitron, RBF, RCE**

1. Architecture: Recurrent NN



gure 2-8 (A) Single node with own feedback. (B) Competitive nets.

1. Architecture: Recurrent NN

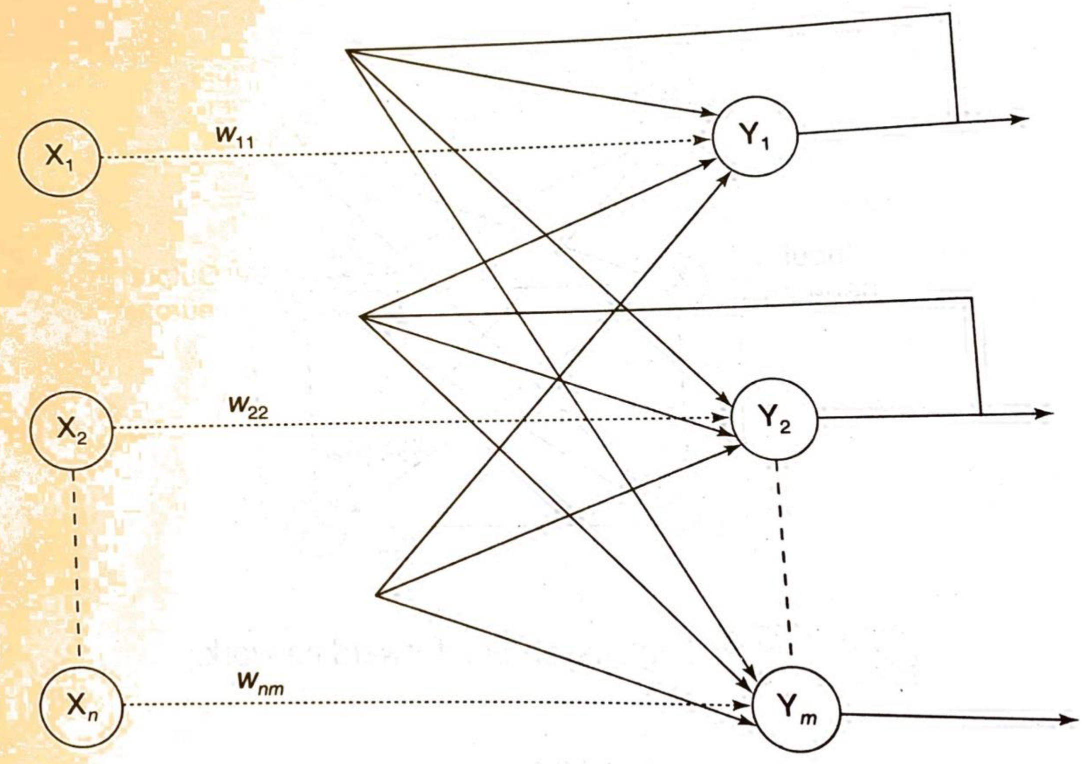


Figure 2-9 Single-layer recurrent network.

1. Architecture: Recurrent NN

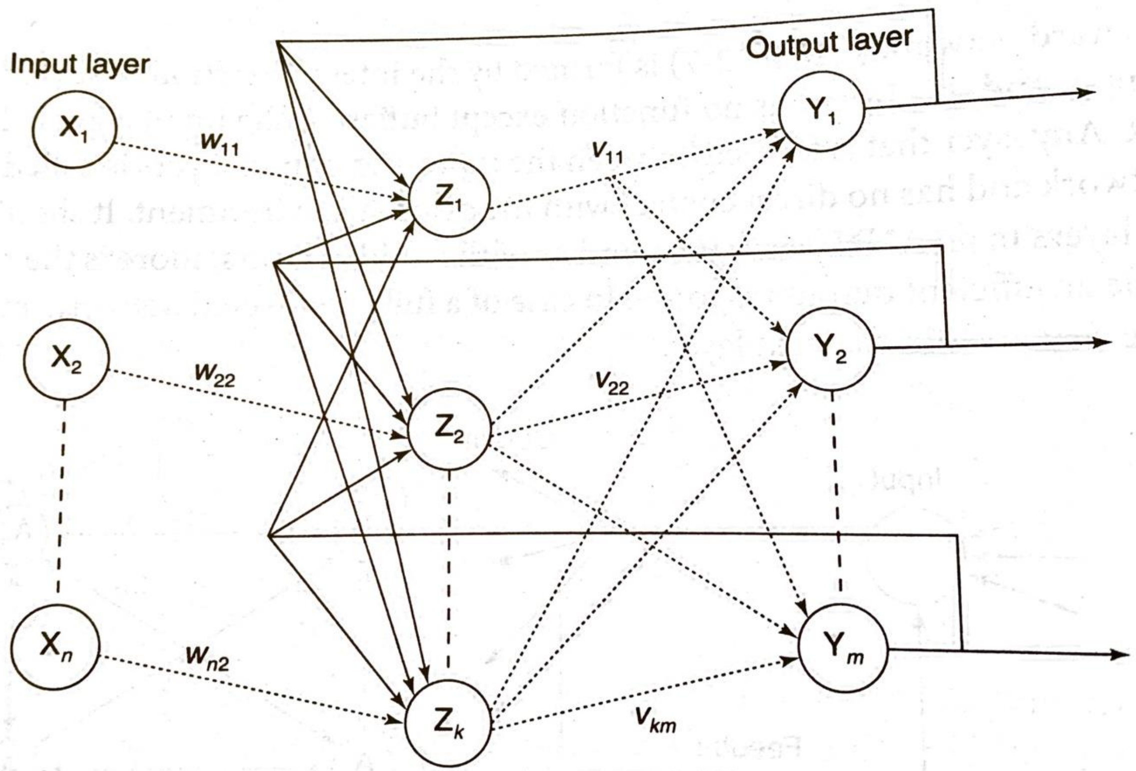
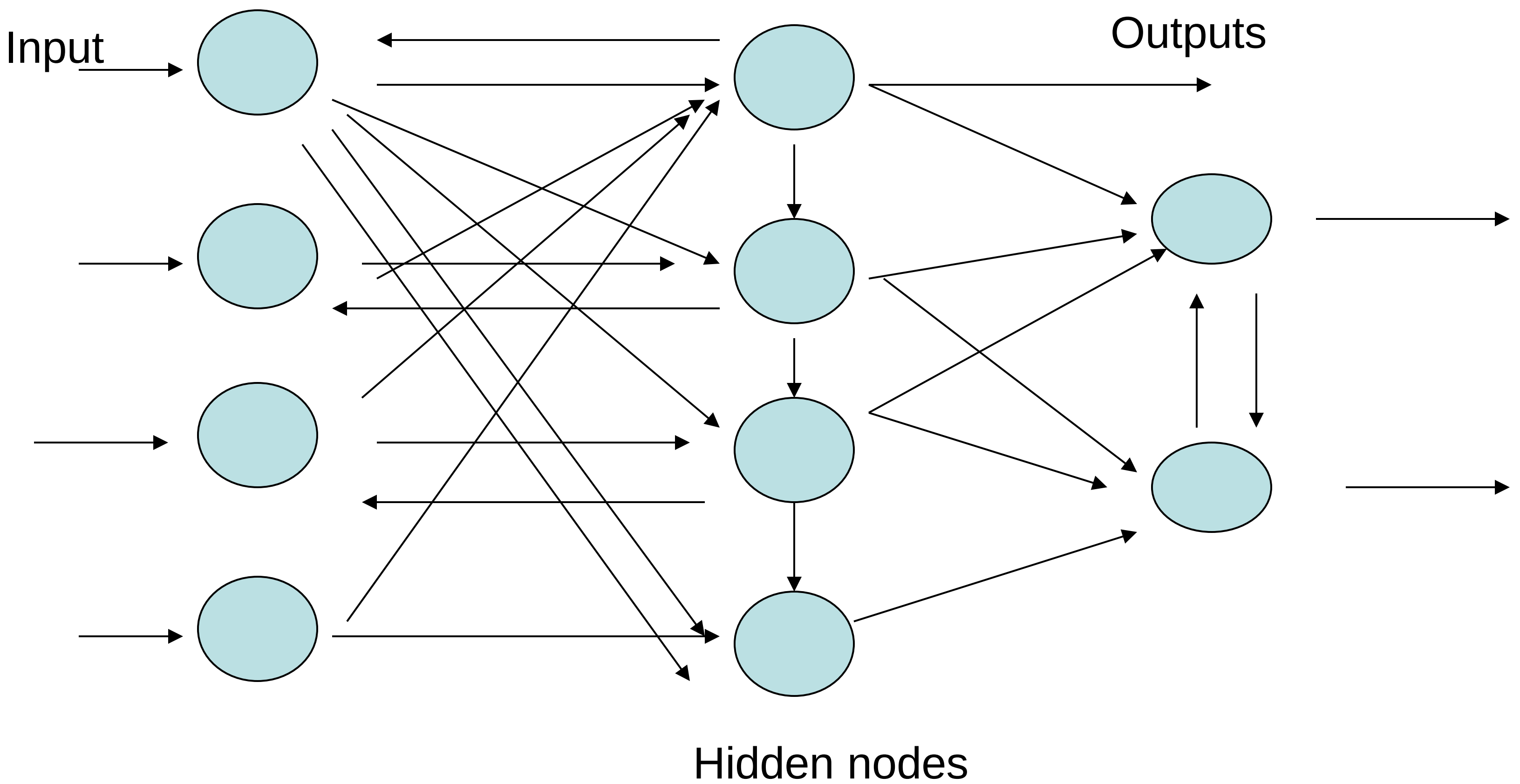


Figure 2-10 Multilayer recurrent network.

1. Architecture: Recurrent NN



Contoh: **ART, BAM, BSB, Boltzman Machine, Cauchy Machine, Hopfield, RNN**

2. Strategy / Learning Algorithm: Supervised

- Learning is performed by presenting pattern with target
- During learning, produced output is compared with the desired output
  - The difference between both output is used to modify learning weights according to the learning algorithm
- Recognizing hand-written digits, pattern recognition and etc.
- Neural Network models**:** **perceptron, feed-forward, radial basis function, support vector machine.**

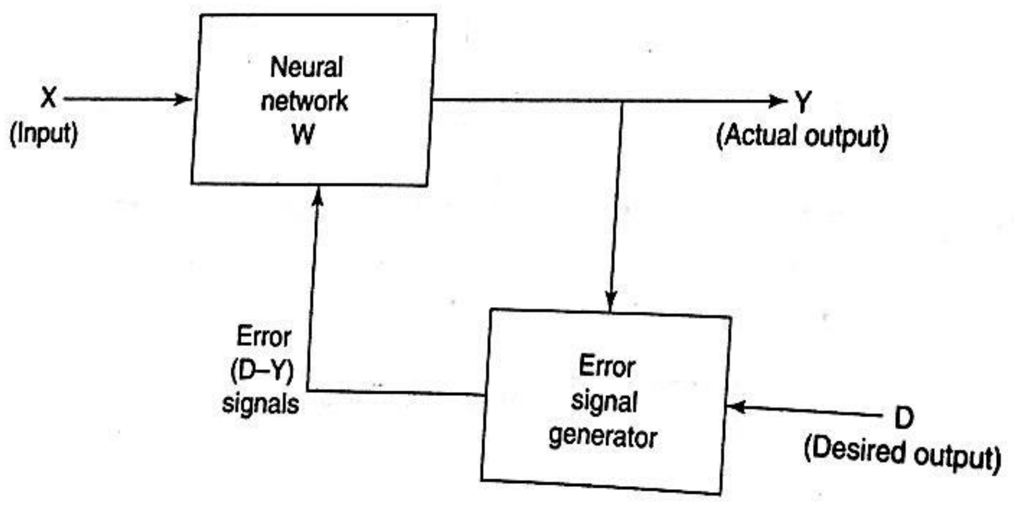


Figure 2-12 Supervised learning.

2. Strategy / Learning Algorithm: Unsupervised

- Targets are not provided
- Appropriate for clustering task
  - Find similar groups of documents in the web, content addressable memory, clustering.
- Neural Network models: Kohonen, self organizing maps, Hopfield networks.

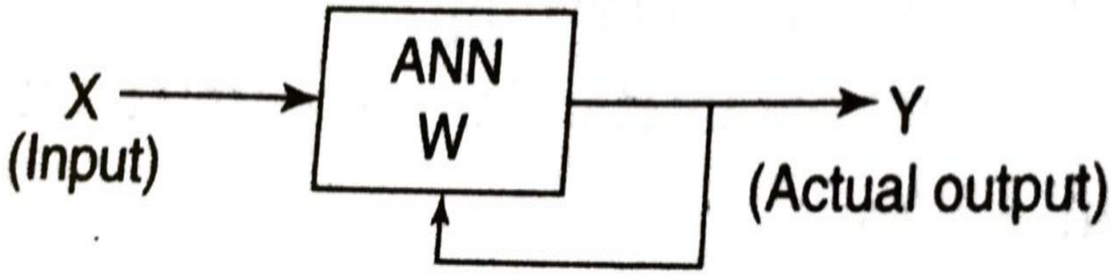


Figure 2-13 Unsupervised learning.

2. Strategy / Learning Algorithm: Reinforcement

- Target is provided, but the desired output is absent.
- The net is only provided with guidance to determine the produced output is correct or vise versa.
- Weights are modified in the units that have errors

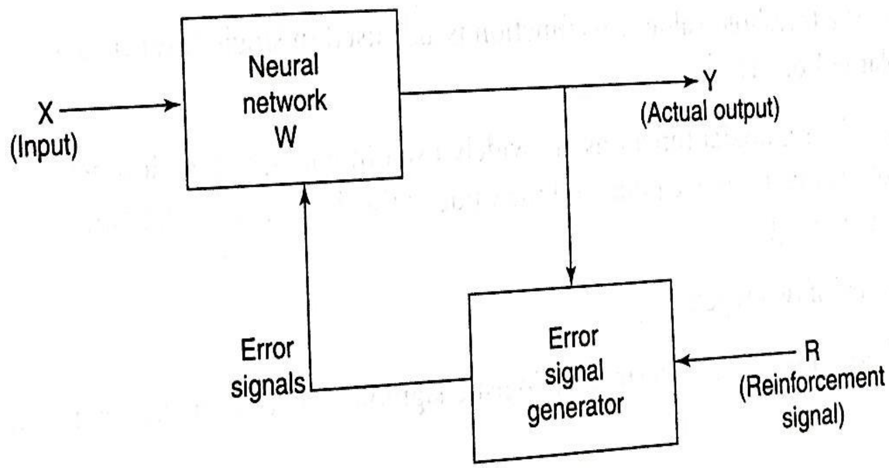


Figure 2-14 Reinforcement learning.

Reinforcement Learning vs. Supervised Learning

Parameters	Reinforcement Learning	Supervised Learning
Decision style	reinforcement learning helps you to take your decisions sequentially.	In this method, a decision is made on the input given at the beginning.
Works on	Works on interacting with the environment.	Works on examples or given sample data.
Dependency on decision	In RL method learning decision is dependent. Therefore, you should give labels to all the dependent decisions.	Supervised learning the decisions which are independent of each other, so labels are given for every decision.
Best suited	Supports and work better in AI, where human interaction is prevalent.	It is mostly operated with an interactive software system or applications.
Example	Chess game	Object recognition

Applications of Reinforcement Learning

Here are applications of Reinforcement Learning:

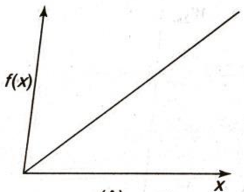
•Robotics for industrial automation.
•Business strategy planning
•Machine learning and data processing
•It helps you to create training systems that provide custom instruction and materials according to the requirement of students.
•Aircraft control and robot motion control

3. Activation Functions

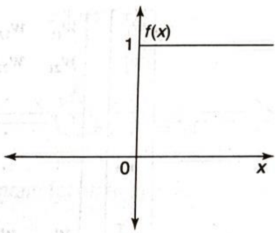
1. *Identity function:* It is a linear function and can be defined as

$$f(x) = x \quad \text{for all } x$$

The output here remains the same as input. The input layer uses the identity activation function.



(A)



(B)

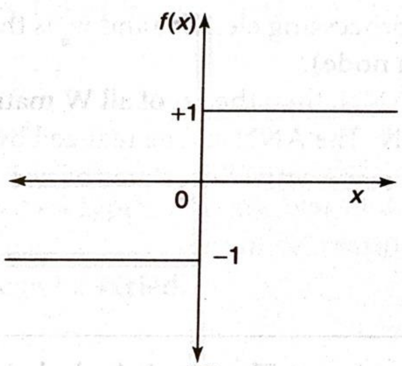
2. *Binary step function:* This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

where θ represents the threshold value. This function is most widely used in single-layer nets to convert the net input to an output that is a binary (1 or 0).

3. *Bipolar step function:* This function can be defined as

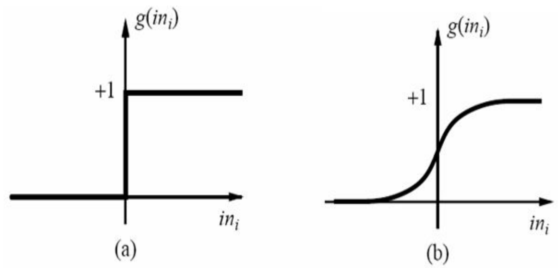
$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$



(C)

3. Activation Functions

Sigmoidal Functions



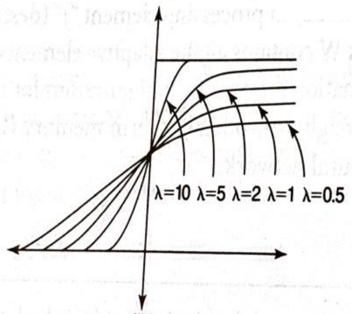
step function　　sigmoid function

- *Bipolar sigmoid function*: This function is defined as

$$f(x)=\frac{2}{1+e^{-\lambda x}}-1=\frac{1-e^{-\lambda x}}{1+e^{-\lambda x}}$$

where λ is the steepness parameter and the sigmoid function range is between −1 and +1. The derivative of th function can be

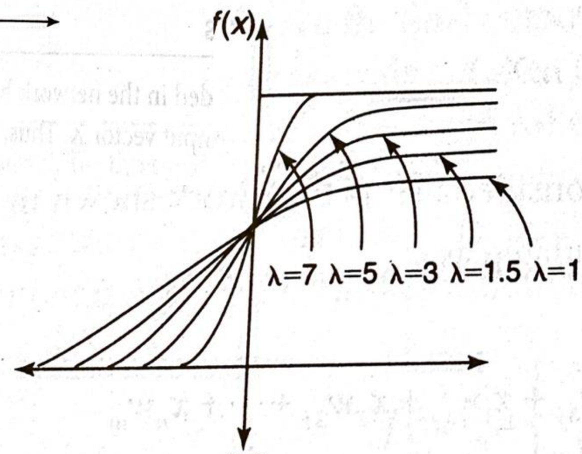
$$f'(x)=\frac{\lambda}{2}[1+f(x)][1-f(x)]$$



(D)

The bipolar sigmoidal function is closely related to hyperbolic tangent function, which is written as

$$h(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=\frac{1-e^{-2x}}{1+e^{-2x}}$$



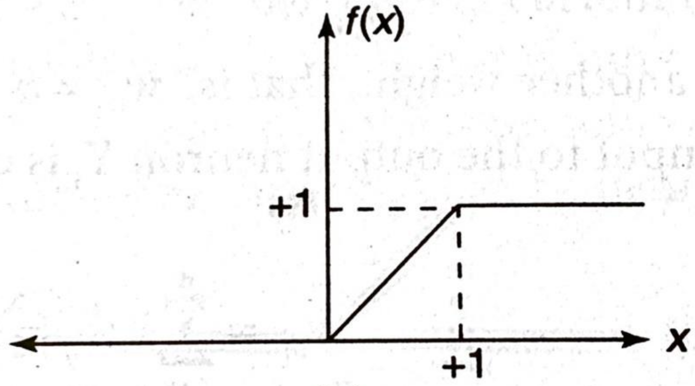
(E)

The derivative of the hyperbolic tangent function is

$$h'(x)=[1+h(x)][1-h(x)]$$

If the network uses a binary data, it is better to convert it to bipolar form and use the bipolar sigmoidal activation function or hyperbolic tangent function.

3. Activation Functions



(F)

5. *Ramp function*: The ramp function is defined as

$$f(x)=\begin{cases}1 & \text{if } x>1\\ x & \text{if } 0\le x\le 1\\ 0 & \text{if } x<0\end{cases}$$

...f all the activation functions are shown in Figure 2-15(A)–(F).

3. Activation Functions

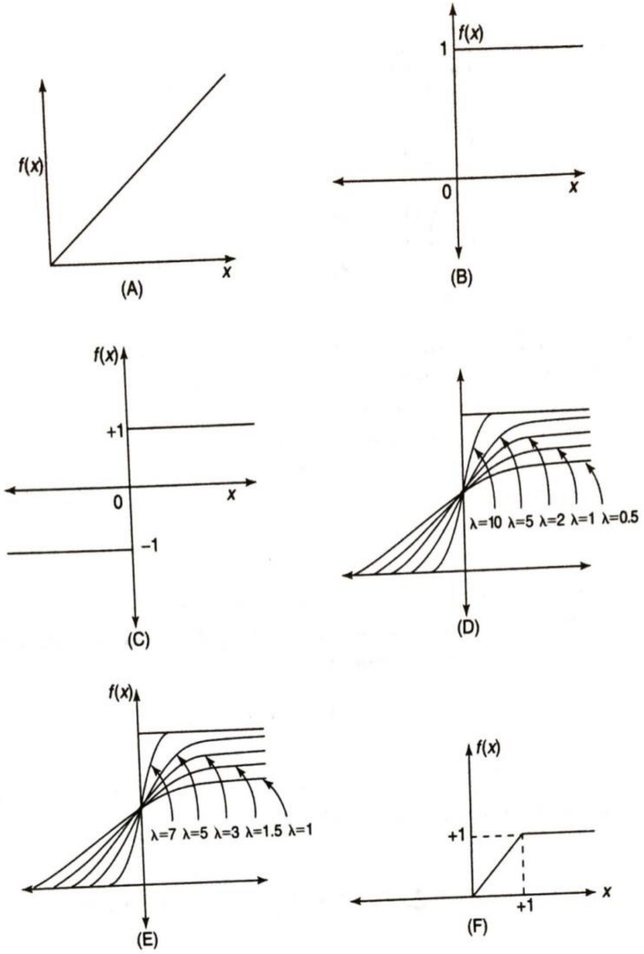
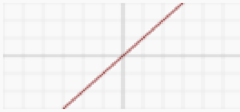
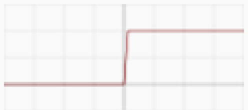
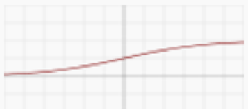
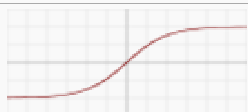
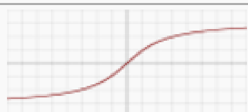
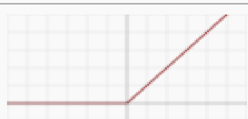
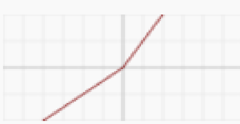
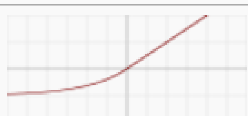
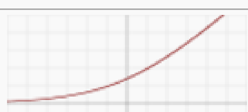


Figure 2-15 Depiction of activation functions: (A) identity function; (B) binary step function; (C) bipolar step function; (D) binary sigmoidal function; (E) bipolar sigmoidal function; (F) ramp function.

- Identity
  $f(x) = x$
- Binary step
  $f(x) = 1$ if $x >= \theta$
  $f(x) = 0$ otherwise
- Binary sigmoid
  $f(x) = 1 / (1 + e^{-\sigma x})$
- Bipolar sigmoid
  $f(x) = -1 + 2 / (1 + e^{-\sigma x})$
- Hyperbolic tangent
  $f(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$

Rectified Linear Unit (ReLU)

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

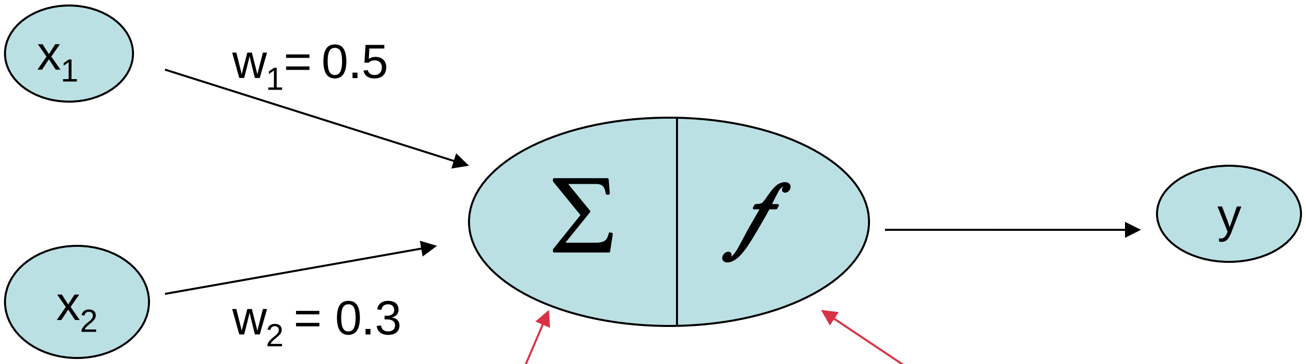
Exercise

- 2 input AND

1	1	1
1	0	0
0	1	0
0	0	0

- 2 input OR

1	1	1
1	0	1
0	1	1
0	0	0



$y_{in} = x_1 w_1 + x_2 w_2$

<u>Activation Function:</u>
Binary Step Function
$\theta = 0.5$,

f(y-in) = 1 if y-in >= θ
dan f(y-in) = 0

Where can neural network systems help…

- when we can't formulate an algorithmic solution.
- when we ***can*** get lots of examples of the behavior we require.

‘learning from experience’

- when we need to pick out the structure from existing data.

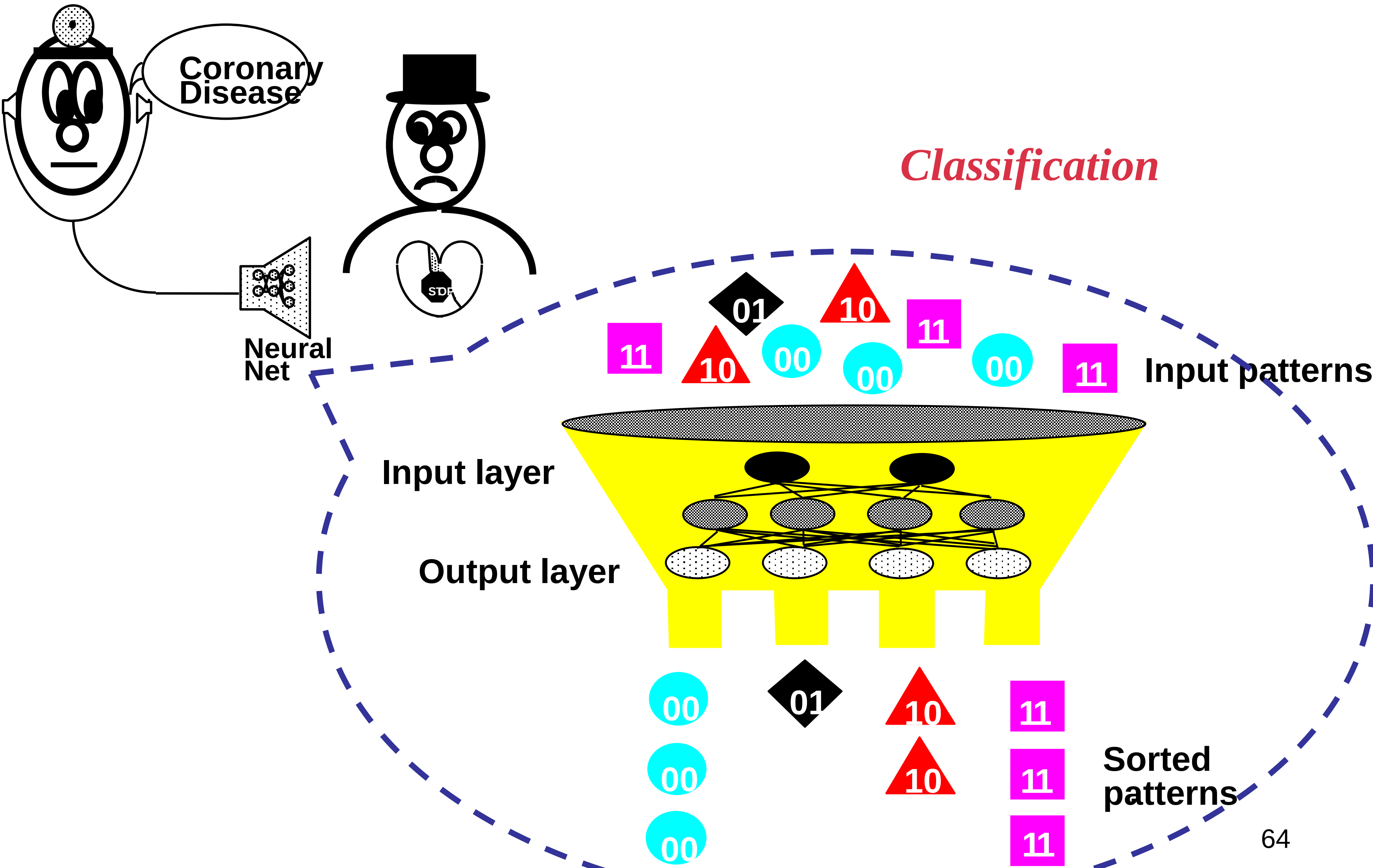
Who is interested?...

- Electrical Engineers – signal processing, control theory
- Computer Engineers – robotics
- Computer Scientists – artificial intelligence, pattern recognition
- Mathematicians – modelling tool when explicit relationships are unknown

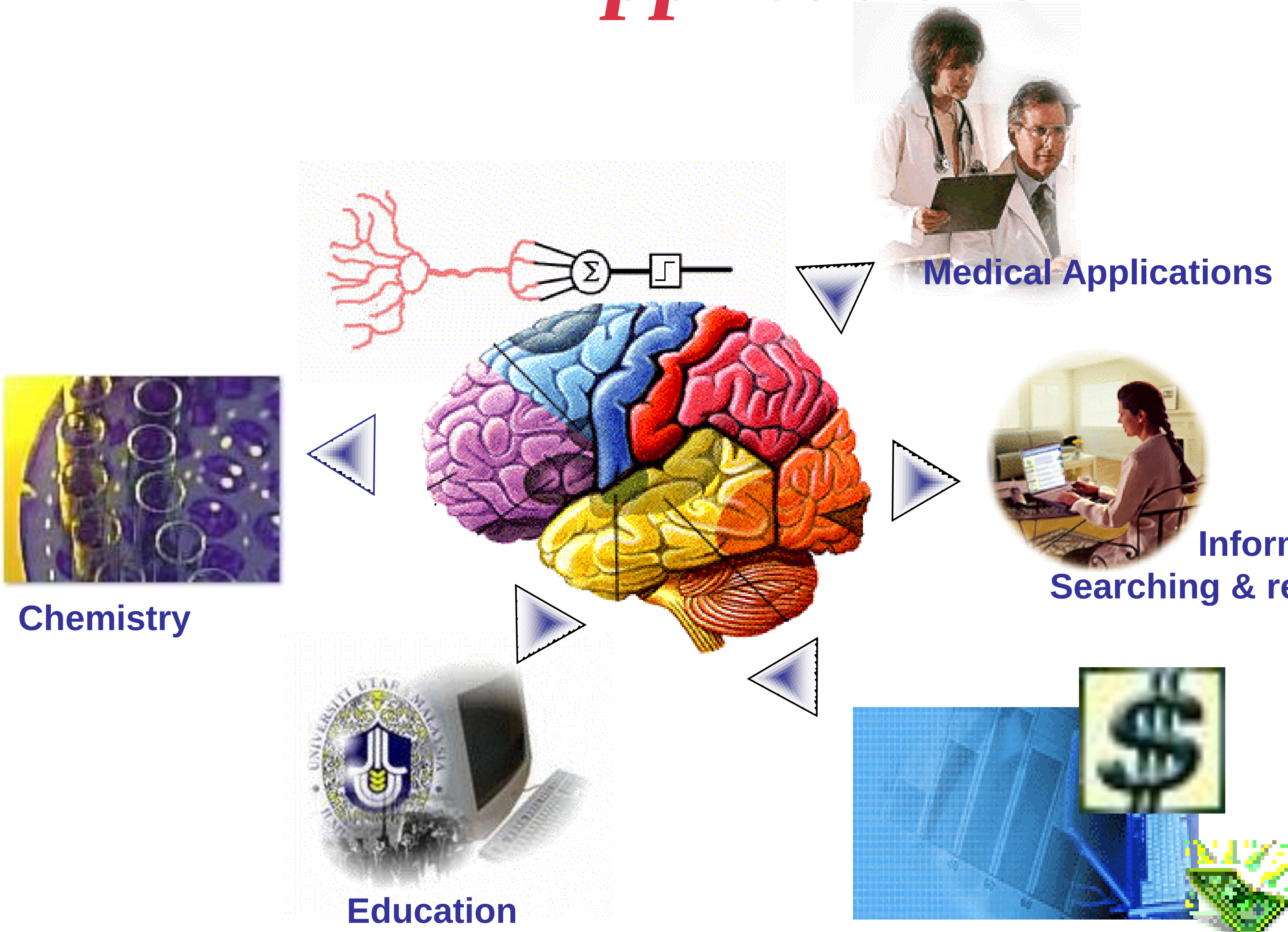
Problem Domains

- Storing and recalling patterns
- Classifying patterns
- Mapping inputs onto outputs
- Grouping similar patterns
- Finding solutions to constrained optimization problems

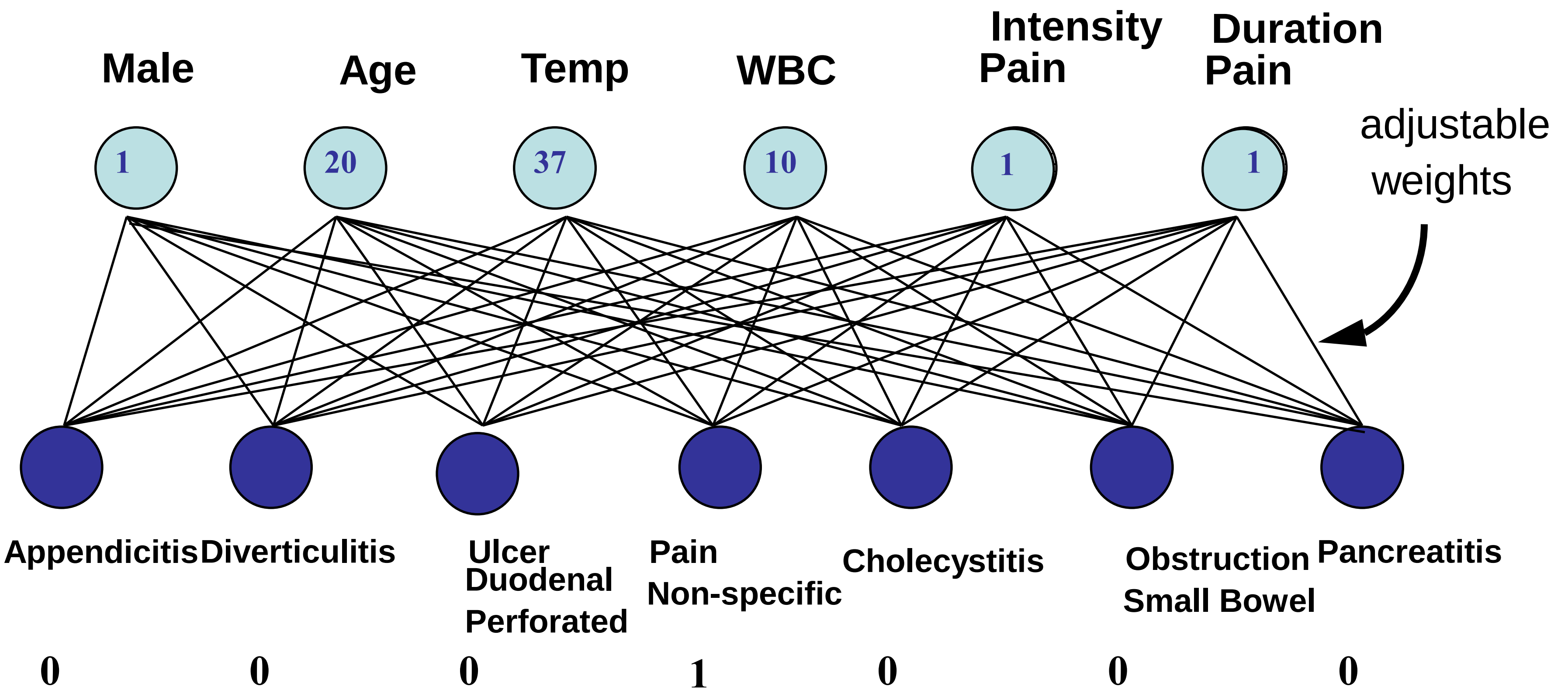
Classification



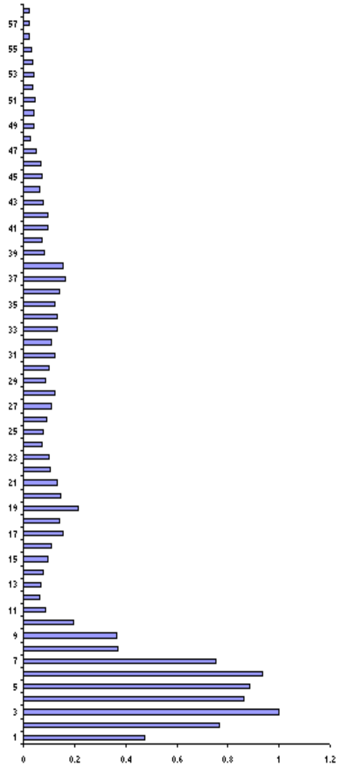
ANN Applications

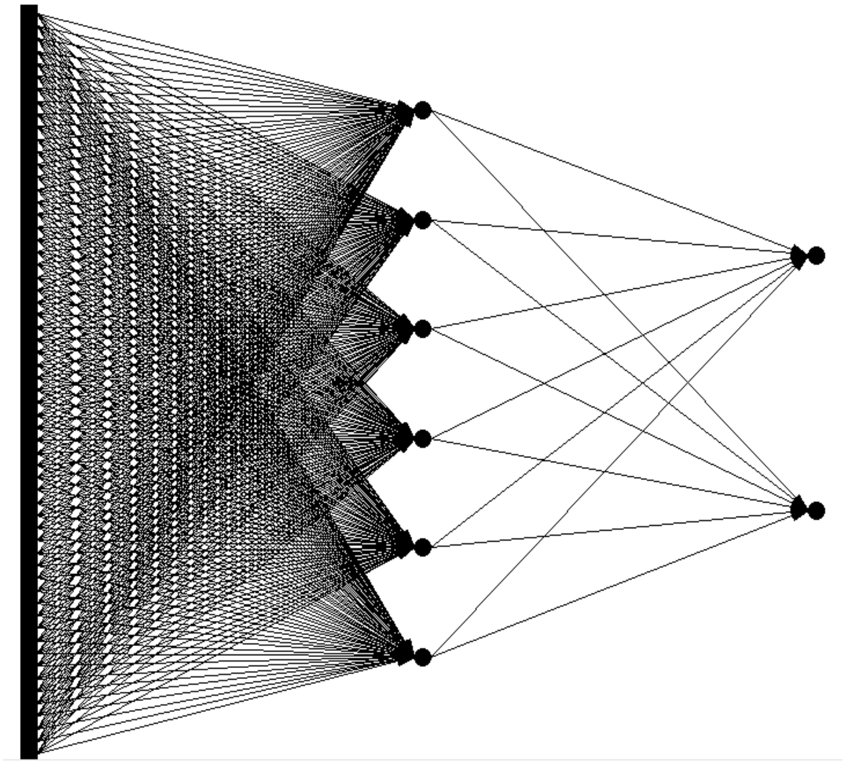


Abdominal Pain Prediction

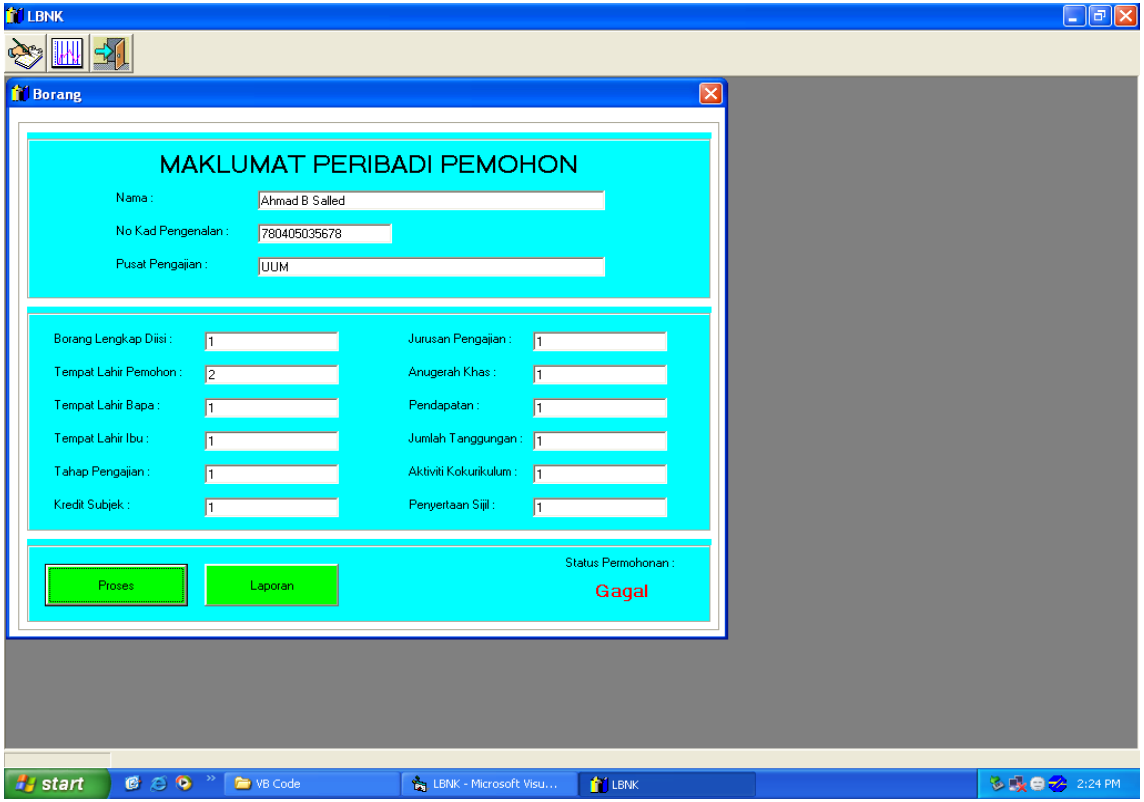


Voice Recognition





Educational Loan Forecasting System



Applications of ANNs

- Signal processing
- Pattern recognition, e.g. handwritten characters or face identification.
- Diagnosis or mapping symptoms to a medical case.
- Speech recognition
- Human Emotion Detection
- Educational Loan Forecasting

Advantages Of NN

NON-LINEARITY

It can model non-linear systems

INPUT-OUTPUT MAPPING

It can derive a relationship between a set of input & output responses

ADAPTIVITY

The ability to learn allows the network to adapt to changes in the surrounding environment

EVIDENTIAL RESPONSE

It can provide a confidence level to a given solution

Advantages Of NN

CONTEXTUAL INFORMATION

Knowledge is presented by the structure of the network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally in the network.

FAULT TOLERANCE

Distributed nature of the NN gives it fault tolerant capabilities

NEUROBIOLOGY ANALOGY

Models the architecture of the brain

Comparison of ANN with conventional AI methods

CHARACTERISTICS	TRADITIONAL COMPUTING (including Expert Systems)	ARTIFICIAL NEURAL NETWORKS
Processing style	Sequential	Parallel
Functions	Logically (left brained) via Rules Concepts Calculations	Gestault (right brained) via Images Pictures Controls
Learning Method	by rules (didactically)	by example (Socratically)
Applications	Accounting, word processing, math, inventory, digital communications	Sensor processing, speech recognition, pattern recognition, text recognition

Module 1.2 Models of ANN

Basic Models of ANN

The models of ANN are specified as

❑ The models synaptic interconnections

- ✓ Single layer feed forward network
- ✓ Multilayer feed-forward network
- ✓ Single node with its own feedback
- ✓ Single layer recurrent network
- ✓ Multi layer recurrent network

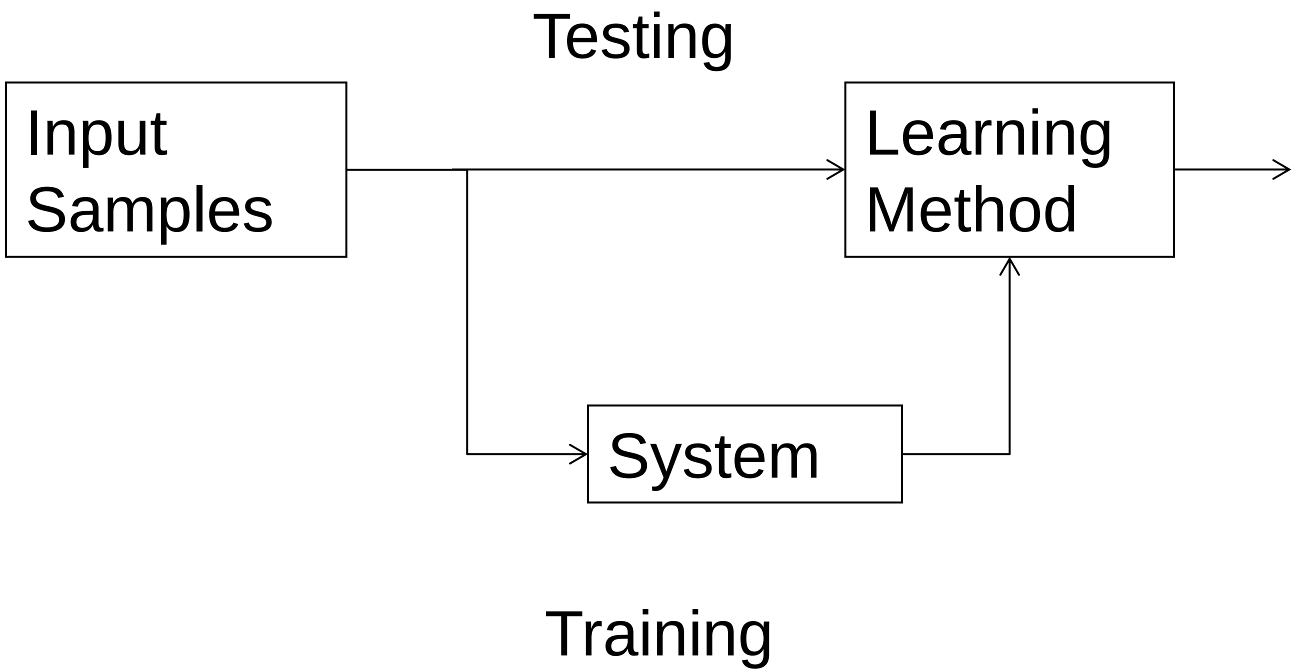
❑ The training or learning rules adopted for updating and adjusting the connection weights

- ✓ Supervised learning
- ✓ Unsupervised learning
- ✓ Reinforcement learning

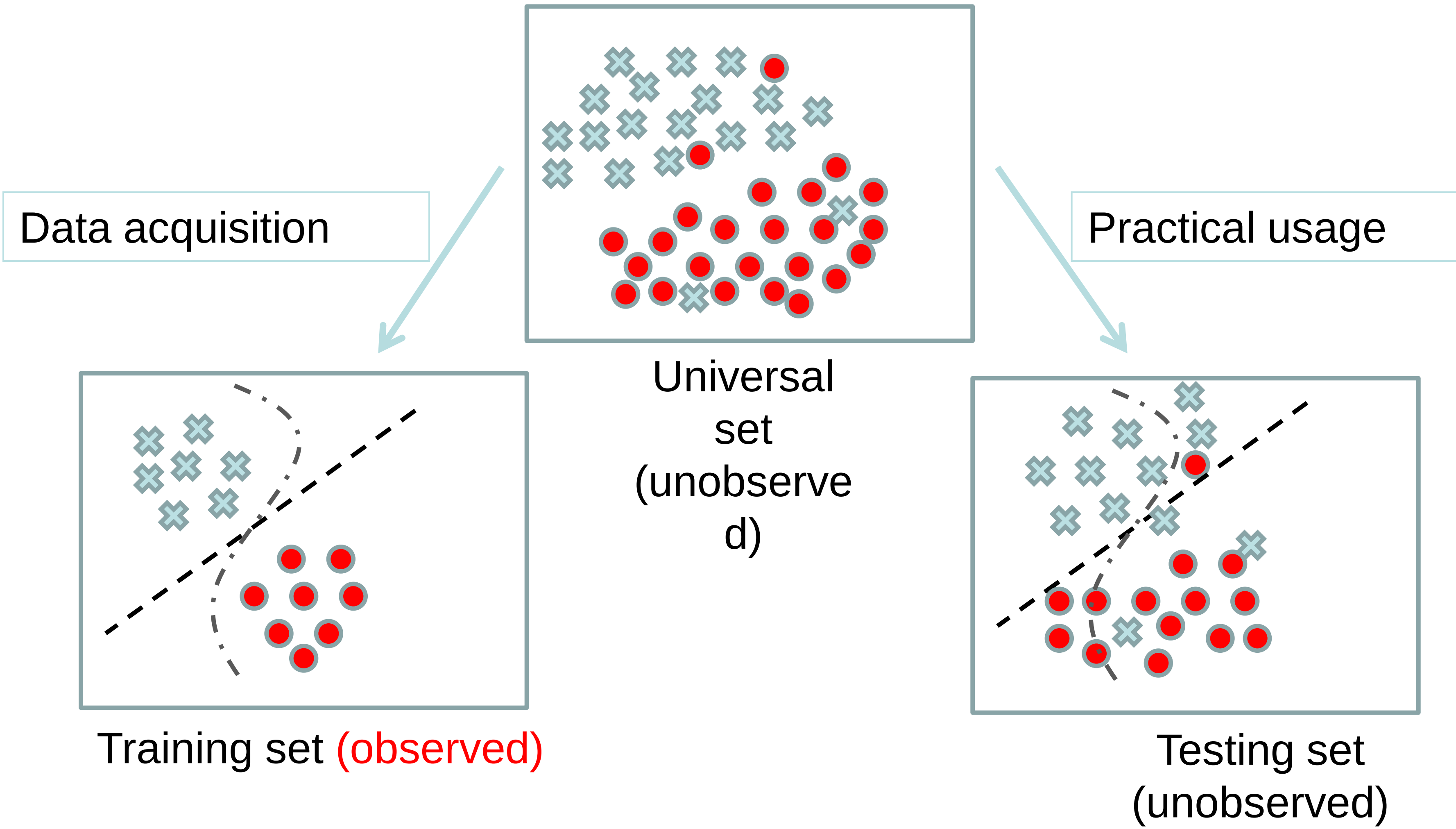
❑ Their activation function

- ✓ Identity function
- ✓ Binary step function
- ✓ Bipolar step function
- ✓ Sigmoidal function
- ✓ Ramp function

Learning system model



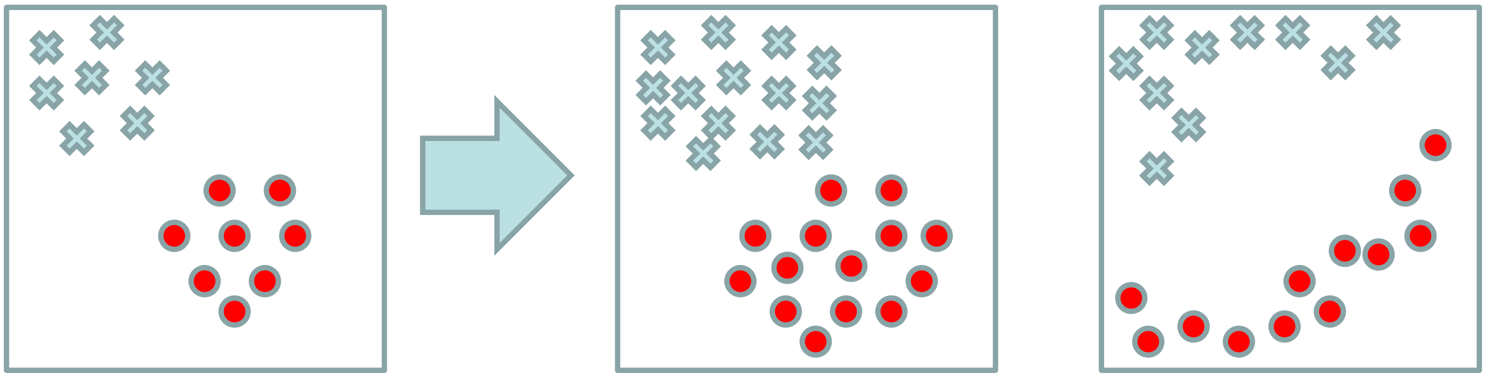
Training and testing



Training and testing

- Training is the process of making the system able to learn.

- No free lunch rule:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Performance

- There are several factors affecting the performance:
  - **Types of training** provided
  - The form and extent of any initial **background knowledge**
  - The **type of feedback** provided
  - The **learning algorithms** used

- Two important factors:
  - Modeling
  - Optimization

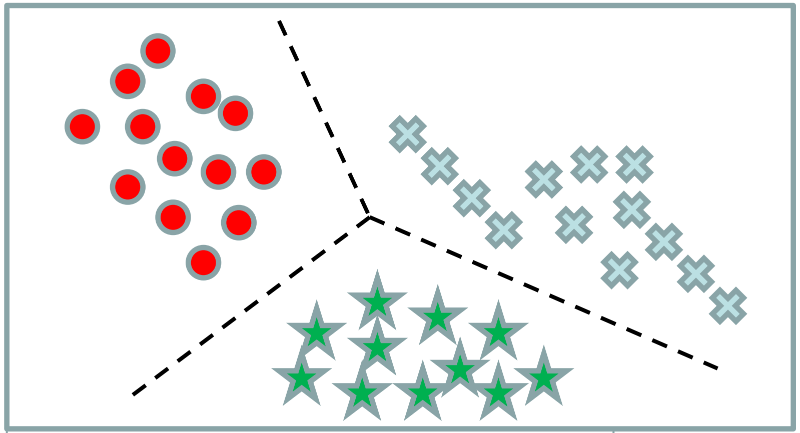
Algorithms

- The success of machine learning system also depends on the algorithms.
- The algorithms control the search to find and build the knowledge structures.
- The learning algorithms should extract useful information from training examples.

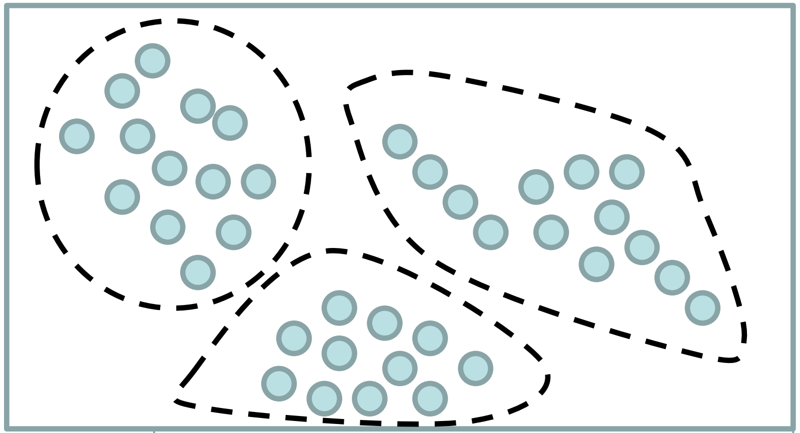
Algorithms

- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)

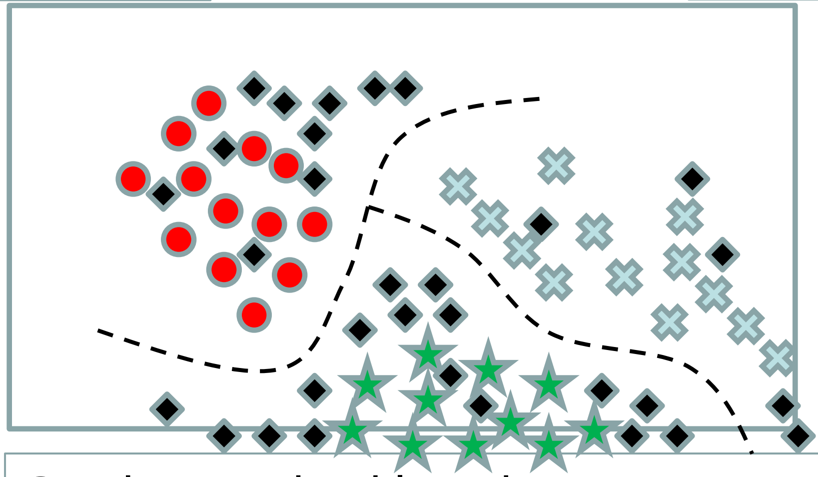
Algorithms



Supervised learning



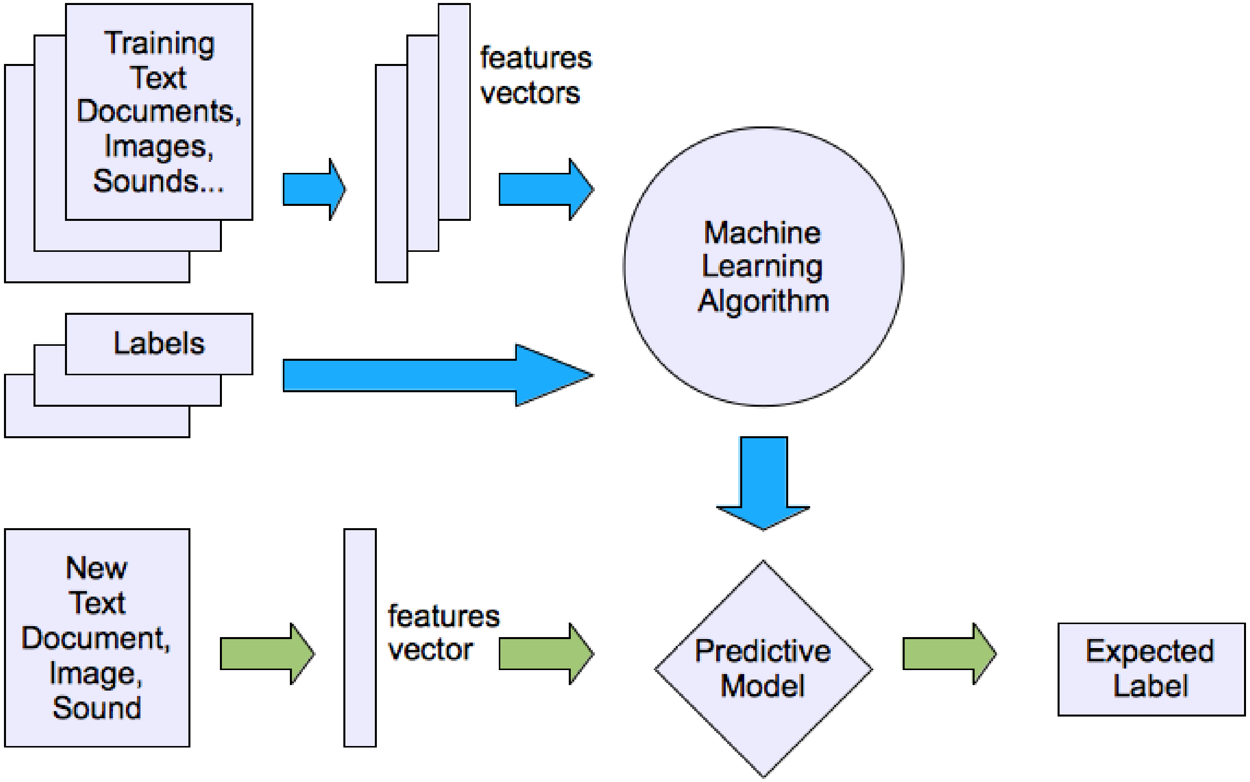
Unsupervised learning



Semi-supervised learning

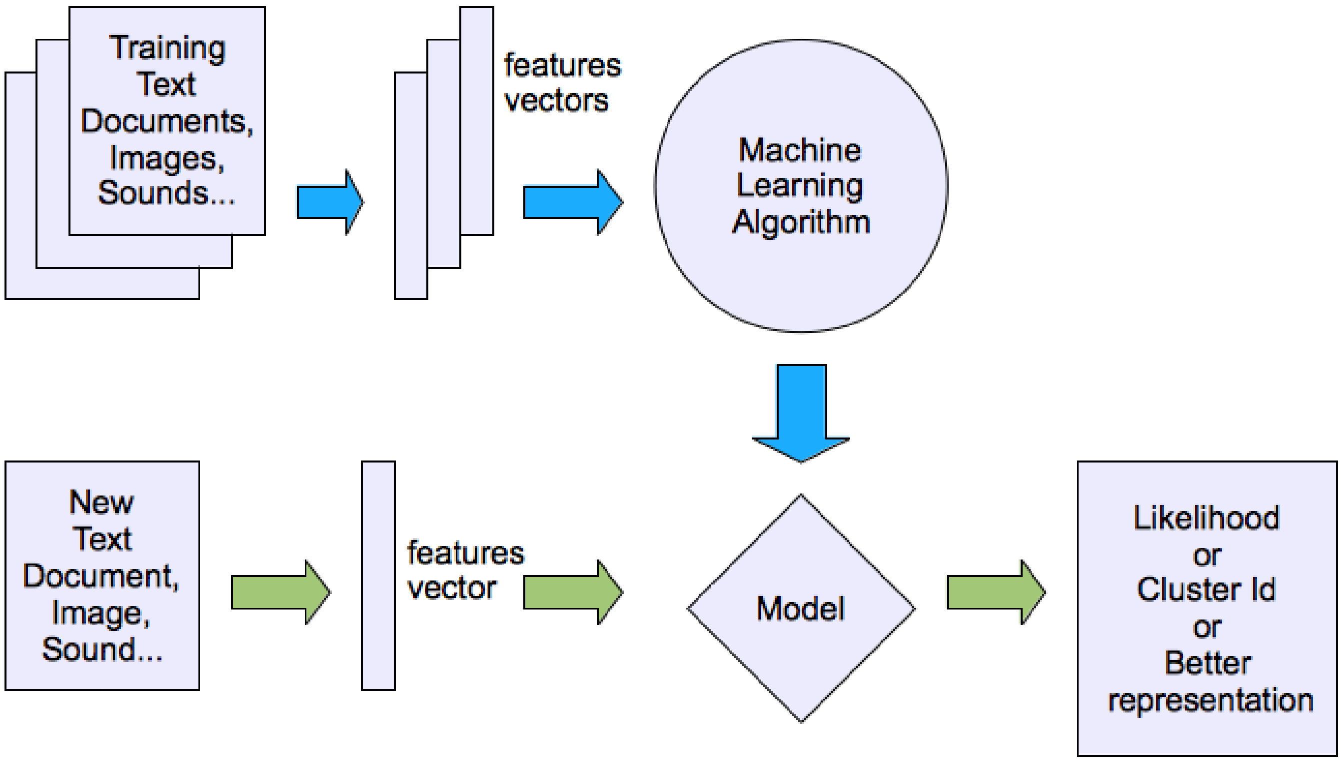
Machine learning structure

- Supervised learning



Machine learning structure

- Unsupervised learning



What are we seeking?

- Supervised: Low E-out or maximize probabilistic terms

$$error = \frac{1}{N}\sum_{n=1}^{N}[y_n \neq g(x_n)]$$

$$Eout(g) \leq Ein(g) \pm O\left(\sqrt{\frac{d_{VC}}{N}\ln N}\right)$$

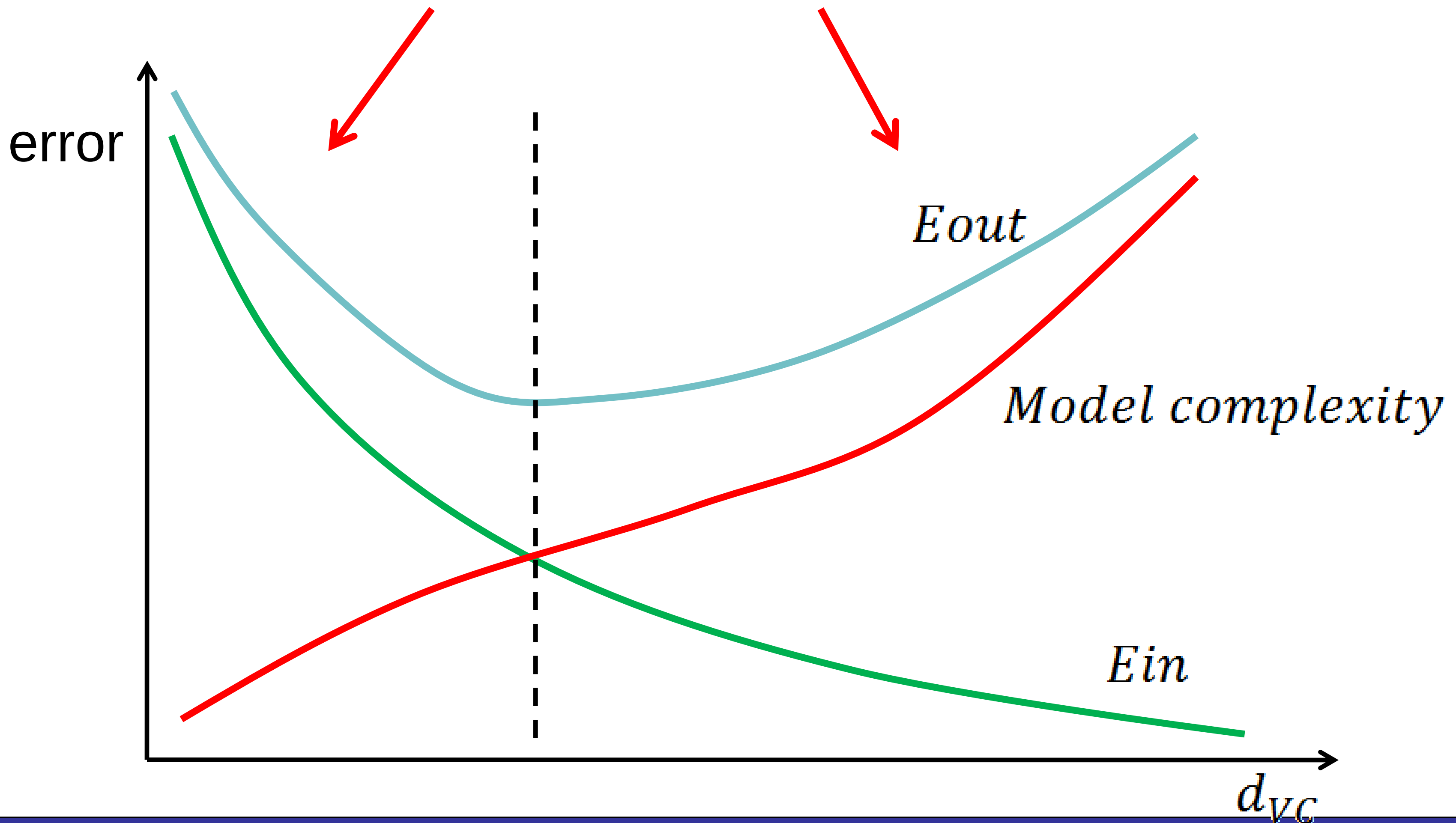
E-in: for training set

E-out: for testing

- Unsupervised: Minimum quantization error, Minimum distance, MAP, MLE(maximum likelihood estimation)

What are we seeking?

Under-fitting VS. Over-fitting (fixed *N*)

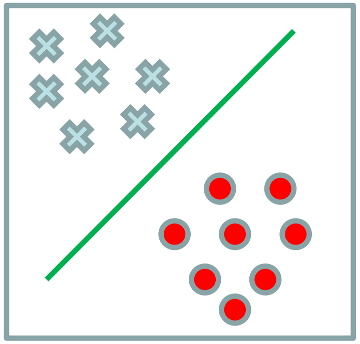


Learning techniques

- Supervised learning categories and techniques
  - **Linear classifier** (numerical functions)
  - **Parametric** (Probabilistic functions)
    - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
  - **Non-parametric** (Instance-based functions)
    - *K*-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
  - **Non-metric** (Symbolic functions)
    - Classification and regression tree (CART), decision tree
  - **Aggregation**
    - Bagging (bootstrap + aggregation), Adaboost, Random forest

Learning techniques

- Linear classifier



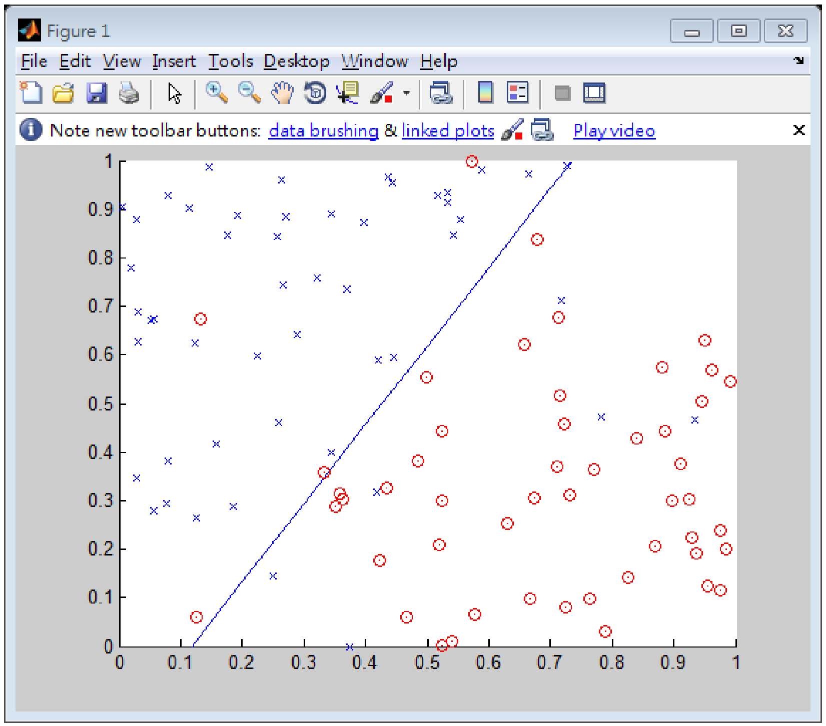
$$g(x_n) = sign(w^T x_n)$$

, where *w* is an *d*-dim vector (learned)

- Techniques:
  - Perceptron
  - Logistic regression
  - Support vector machine (SVM)
  - Ada-line
  - Multi-layer perceptron (MLP)

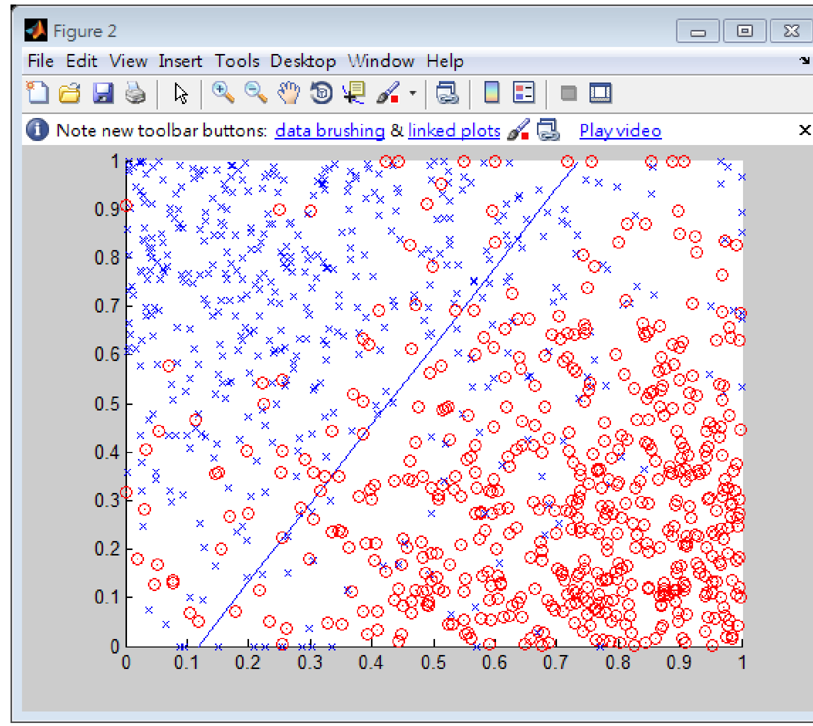
Learning techniques

Using **perceptron learning algorithm**(PLA)



Training

Error rate: 0.10

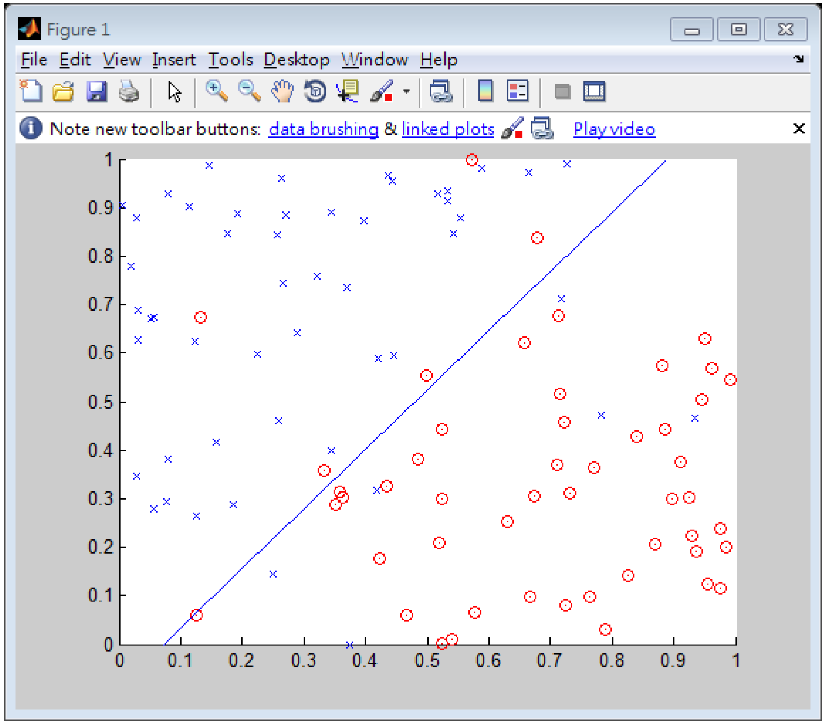


Testing

Error rate: 0.156

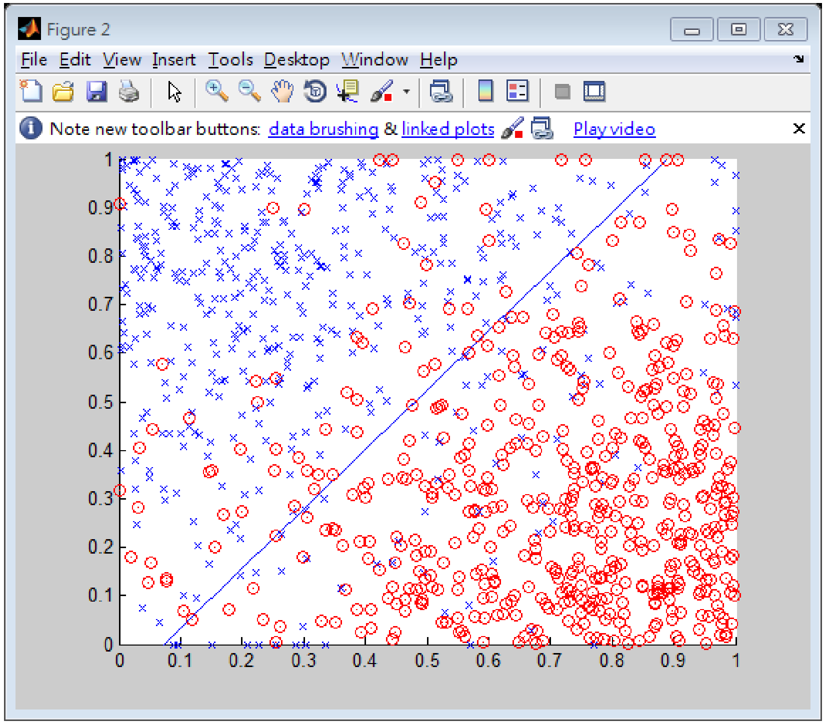
Learning techniques

Using **logistic regression**



Training

Error rate: 0.11

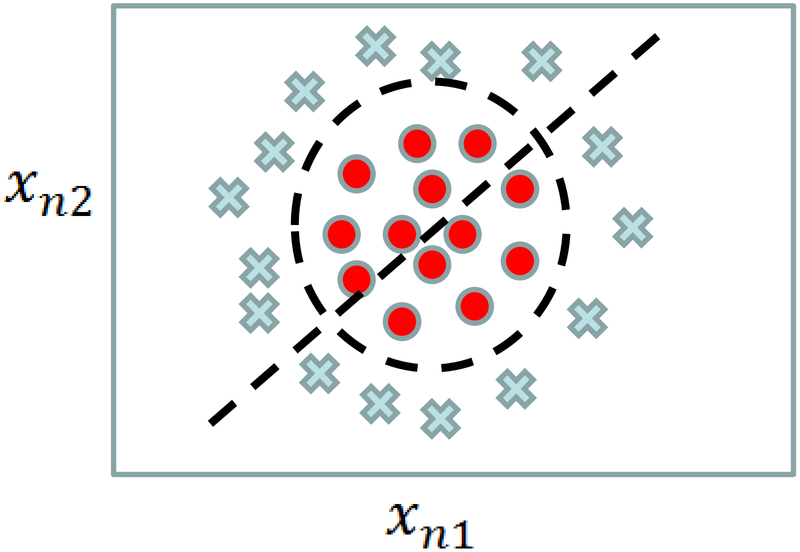


Testing

Error rate: 0.145

Learning techniques

- Non-linear case



$$x_n = [x_{n1}, x_{n2}]$$

$$\Downarrow$$

$$x_n = [x_{n1}, x_{n2}, x_{n1} * x_{n2}, {x_{n1}}^2, {x_{n2}}^2]$$

$$g(x_n) = sign(w^T x_n)$$

- Support vector machine (SVM):
  – Linear to nonlinear: **Feature transform** and **kernel function**

Learning techniques

- Unsupervised learning categories and techniques
  - **Clustering**
    - K-means clustering
    - Spectral clustering
  - **Density Estimation**
    - Gaussian mixture model (GMM)
    - Graphical models
  - **Dimensionality reduction**
    - Principal component analysis (PCA)
    - Factor analysis

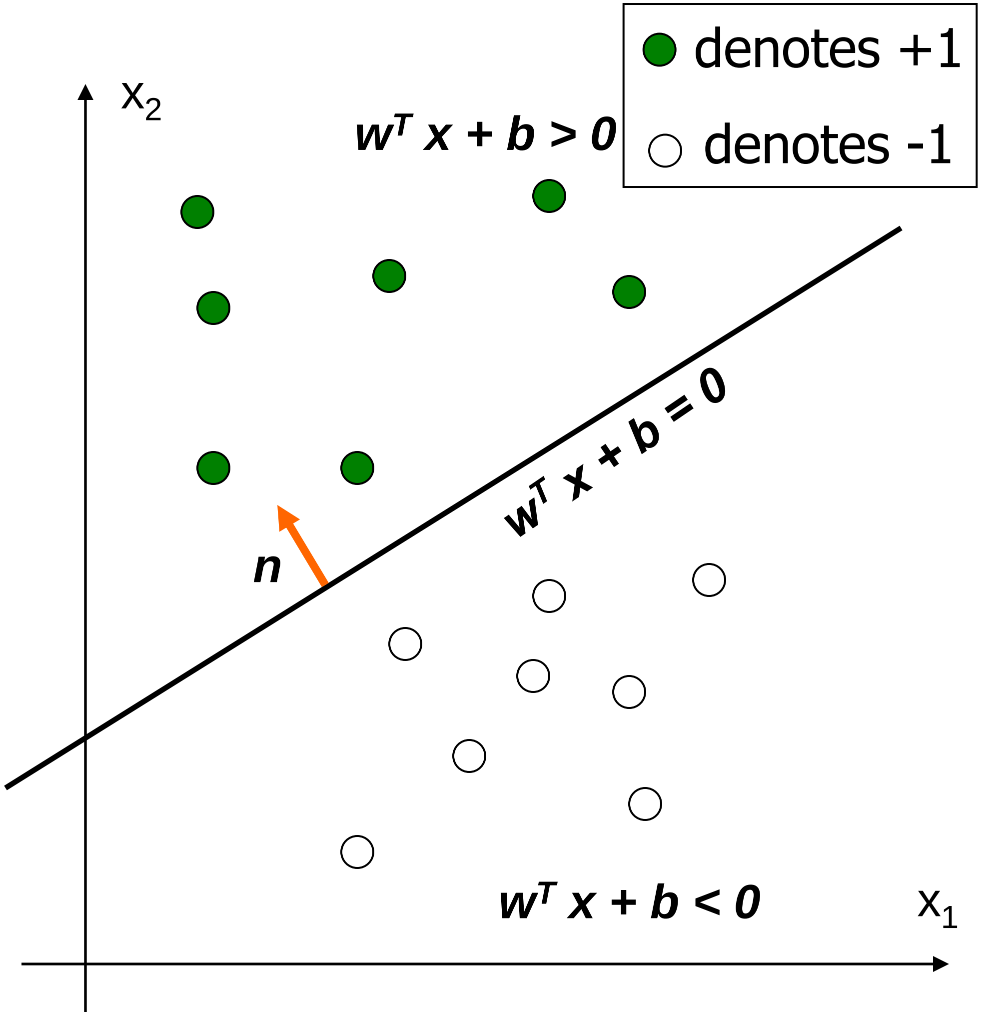
Linear Discriminant Function

- g(x) is a linear function:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- A hyper-plane in the feature space
- (Unit-length) normal vector of the hyper-plane:

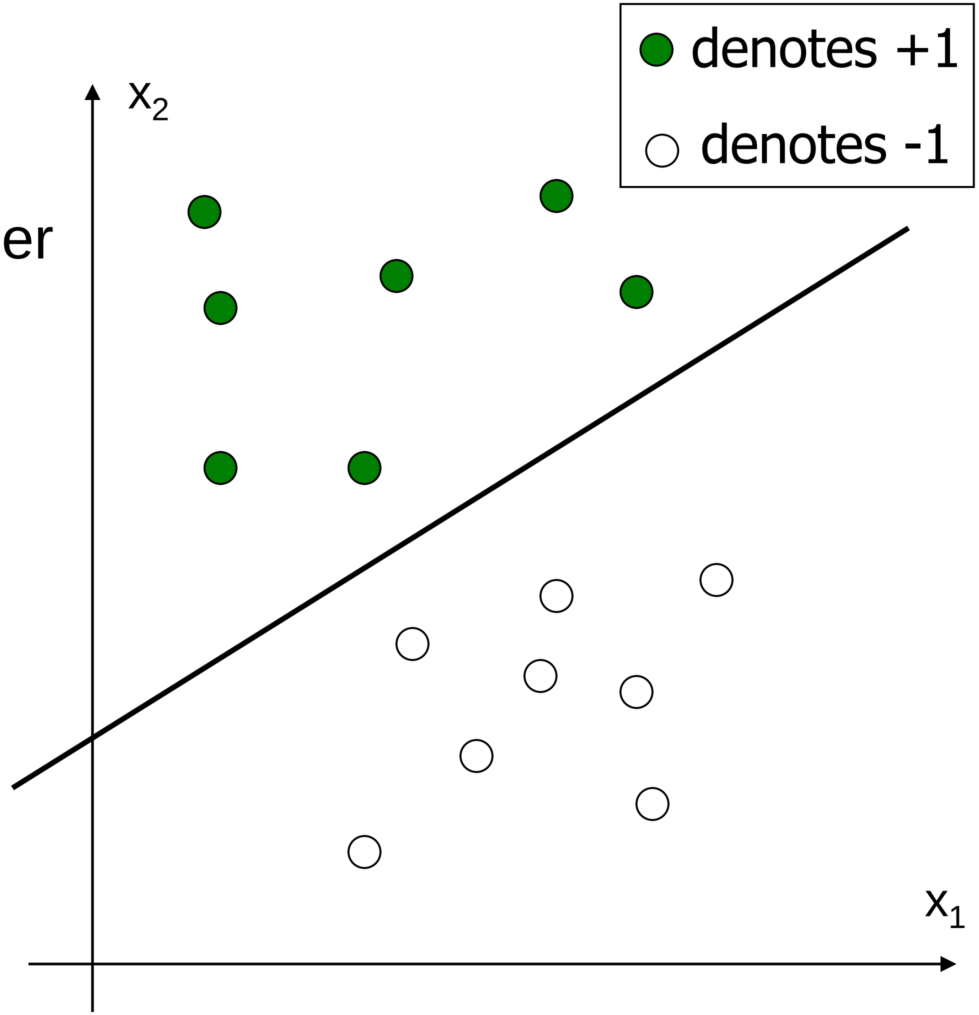
$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$



Linear Discriminant Function

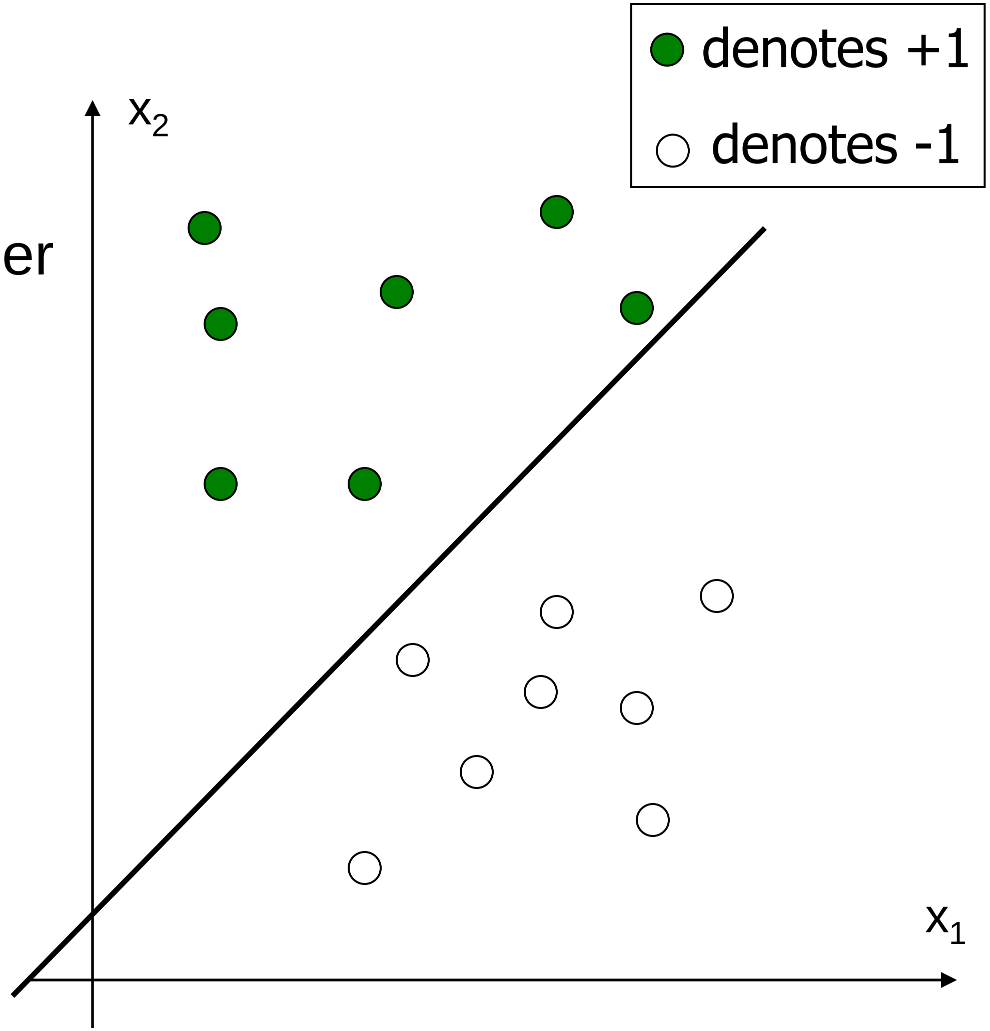
- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers



Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

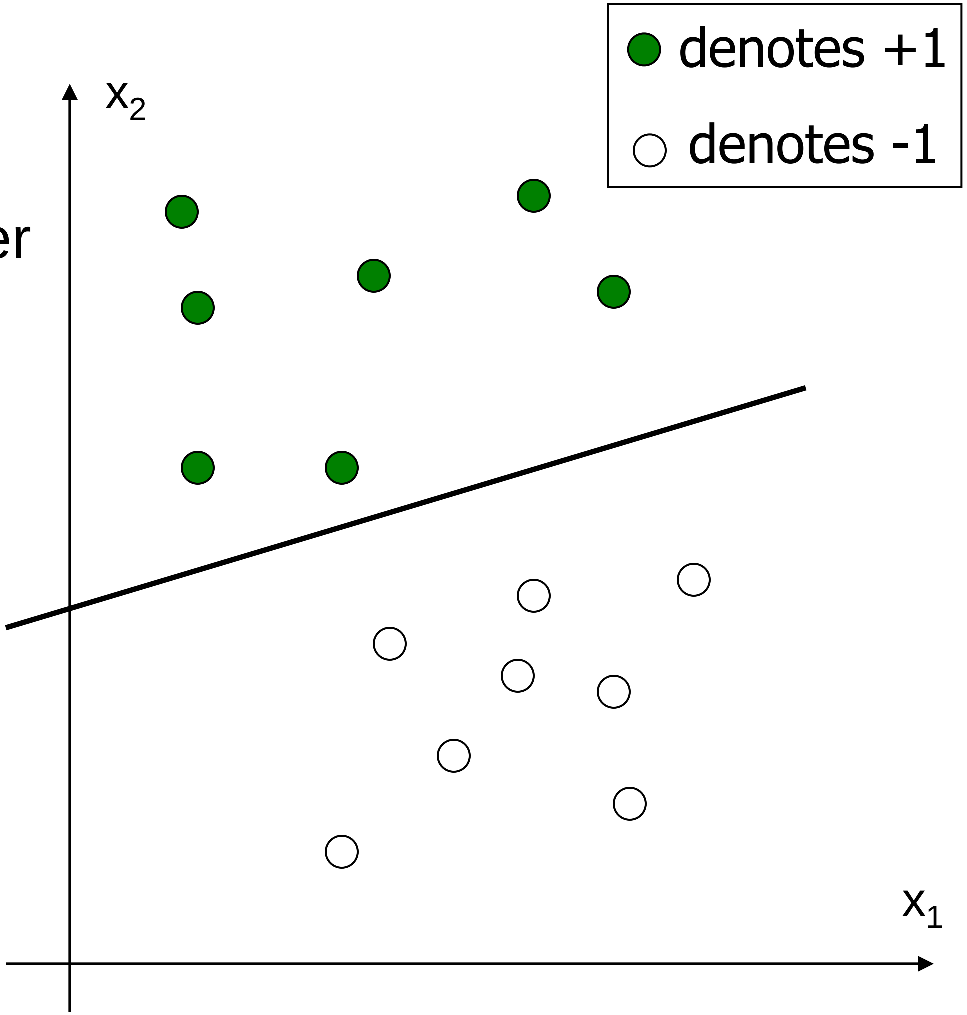


- Infinite number of answers

Linear Discriminant Function

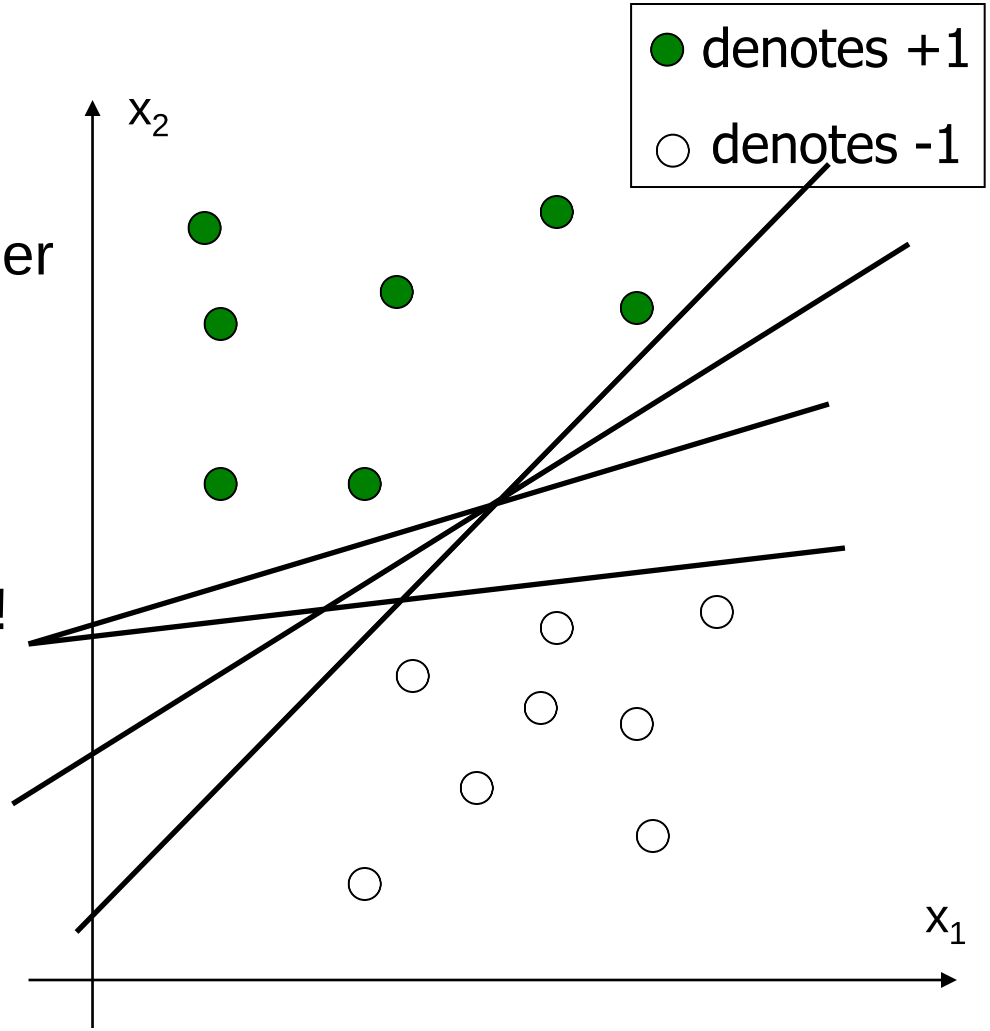
- How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers



Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?

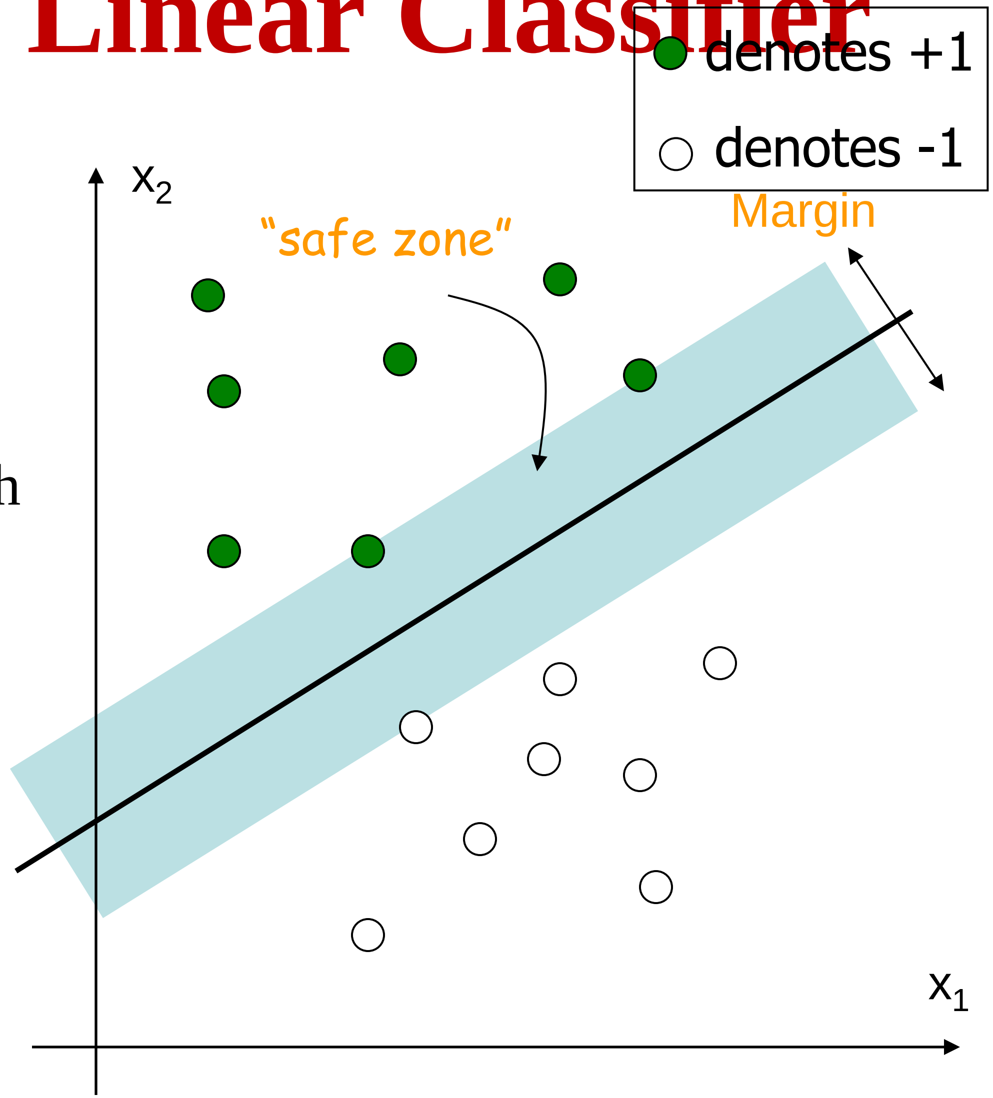


- Infinite number of answers!

- Which one is the best?

Large Margin Linear Classifier

- The linear discriminant function (classifier) with the maximum margin is the best
- Margin is defined as the width that the boundary could be increased by before hitting a data point
- Why it is the best?
  - Robust to outliners and thus strong generalization ability



Large Margin Linear Classifier

- Given a set of data points:

$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n$, where

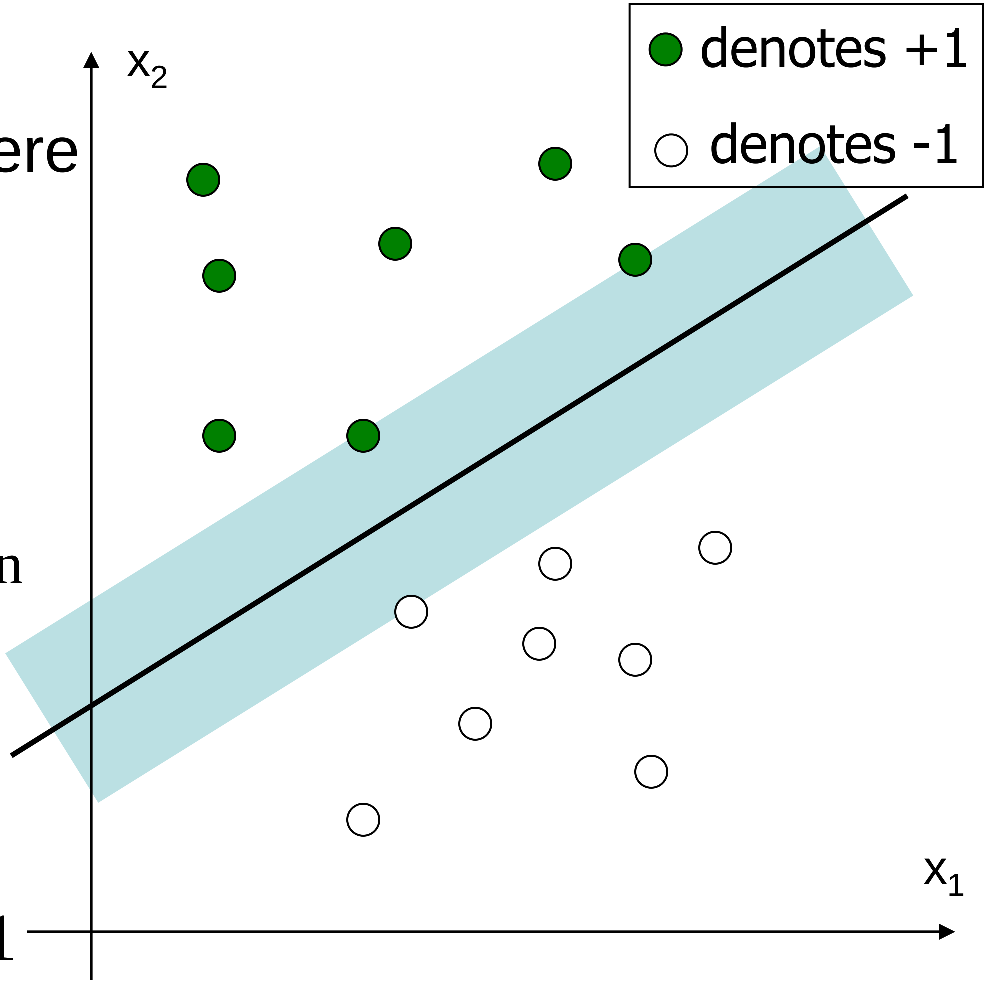
$$\text{For } y_i = +1,\quad \mathbf{w}^T\mathbf{x}_i + b > 0$$

$$\text{For } y_i = -1,\quad \mathbf{w}^T\mathbf{x}_i + b < 0$$

- With a scale transformation on both *w* and *b*, the above is equivalent to

$$\text{For } y_i = +1,\quad \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1,\quad \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

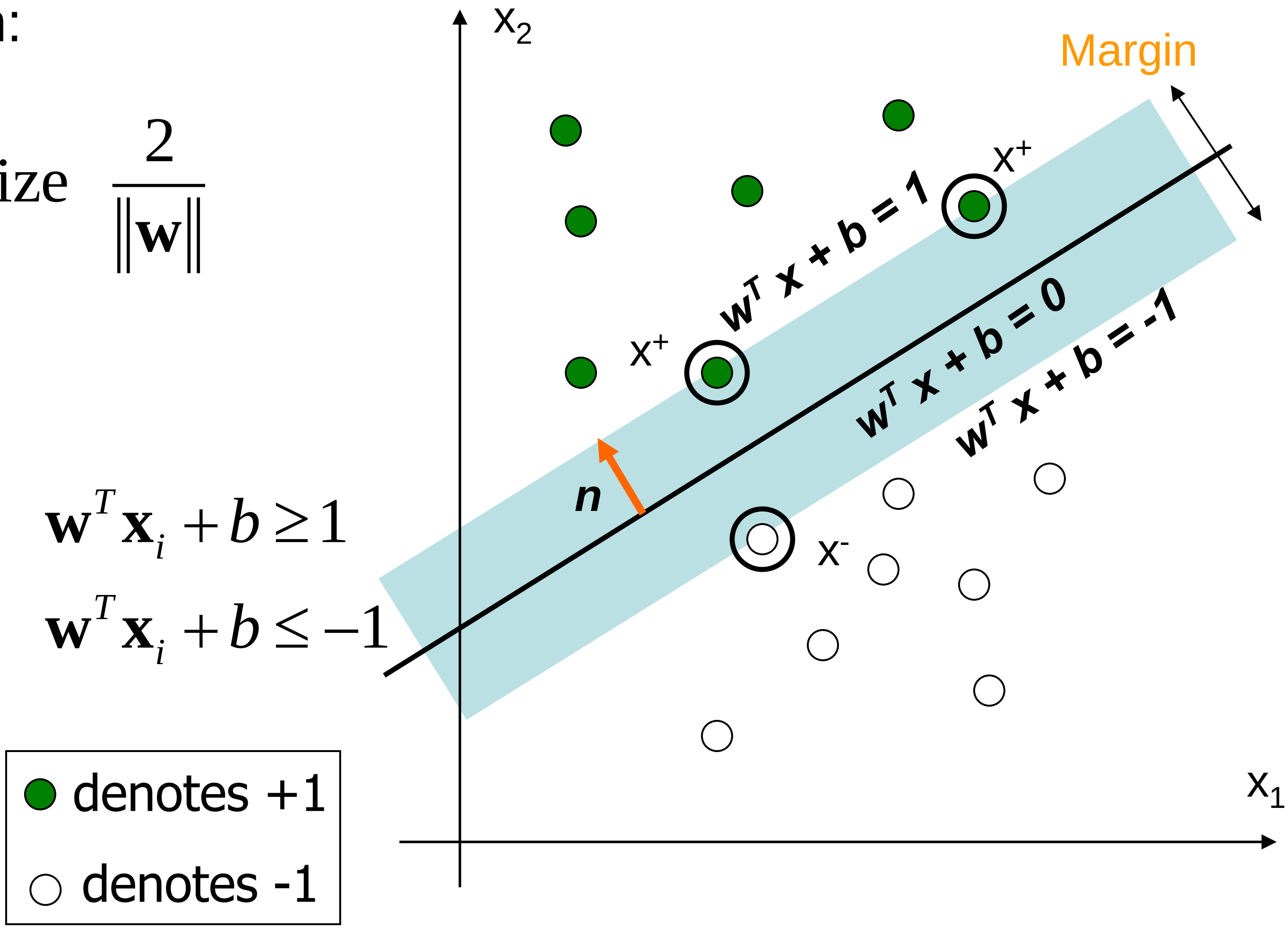
- Formulation:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

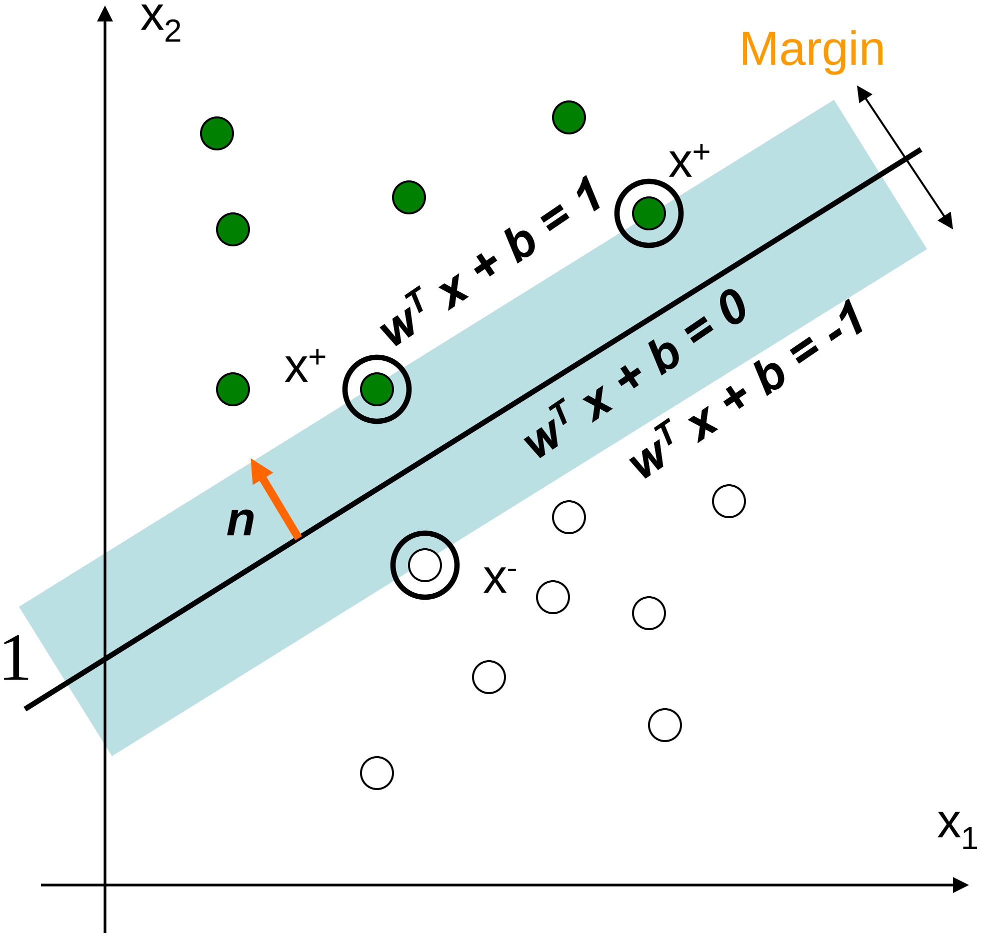
such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$

● denotes +1

○ denotes -1



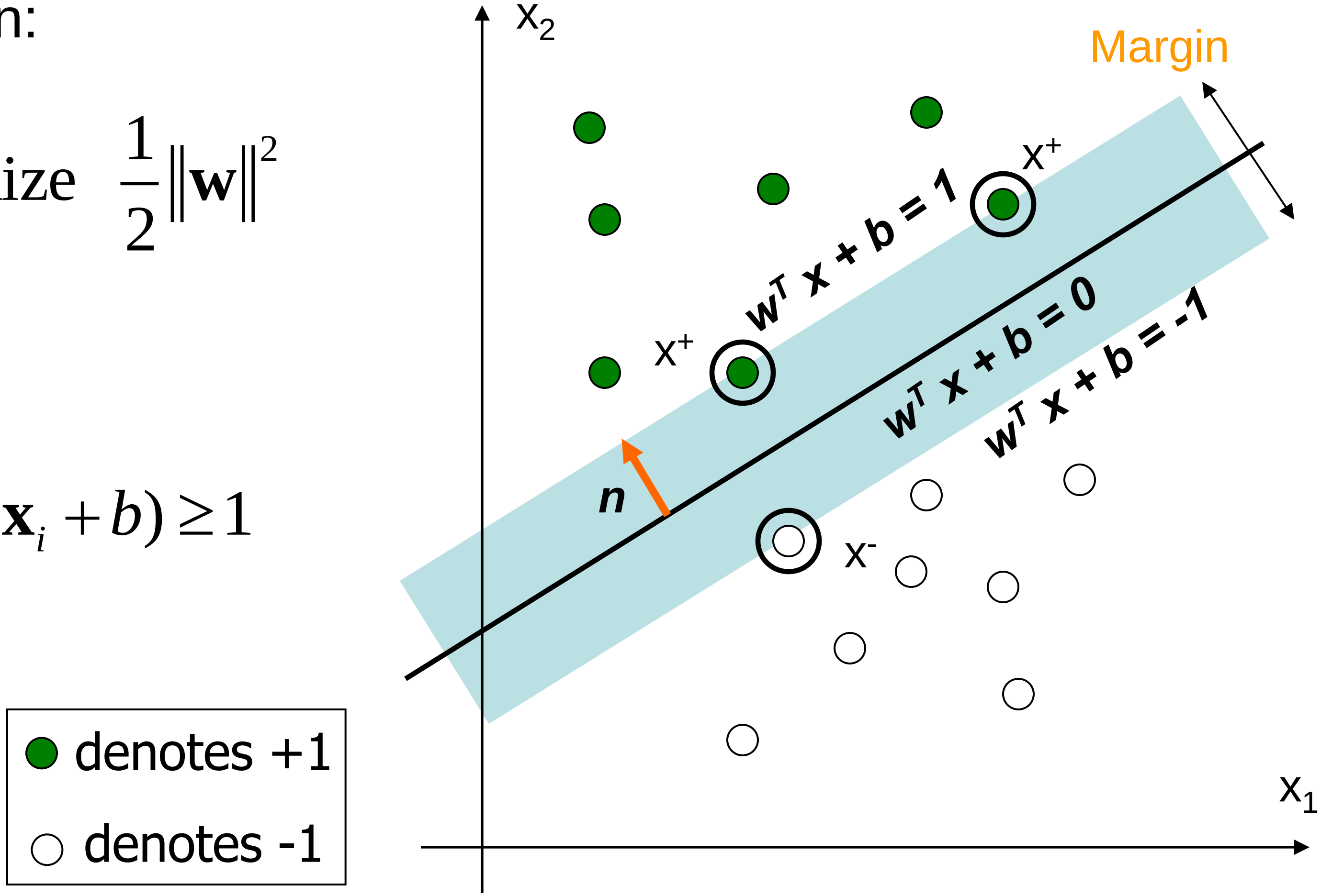
Large Margin Linear Classifier

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

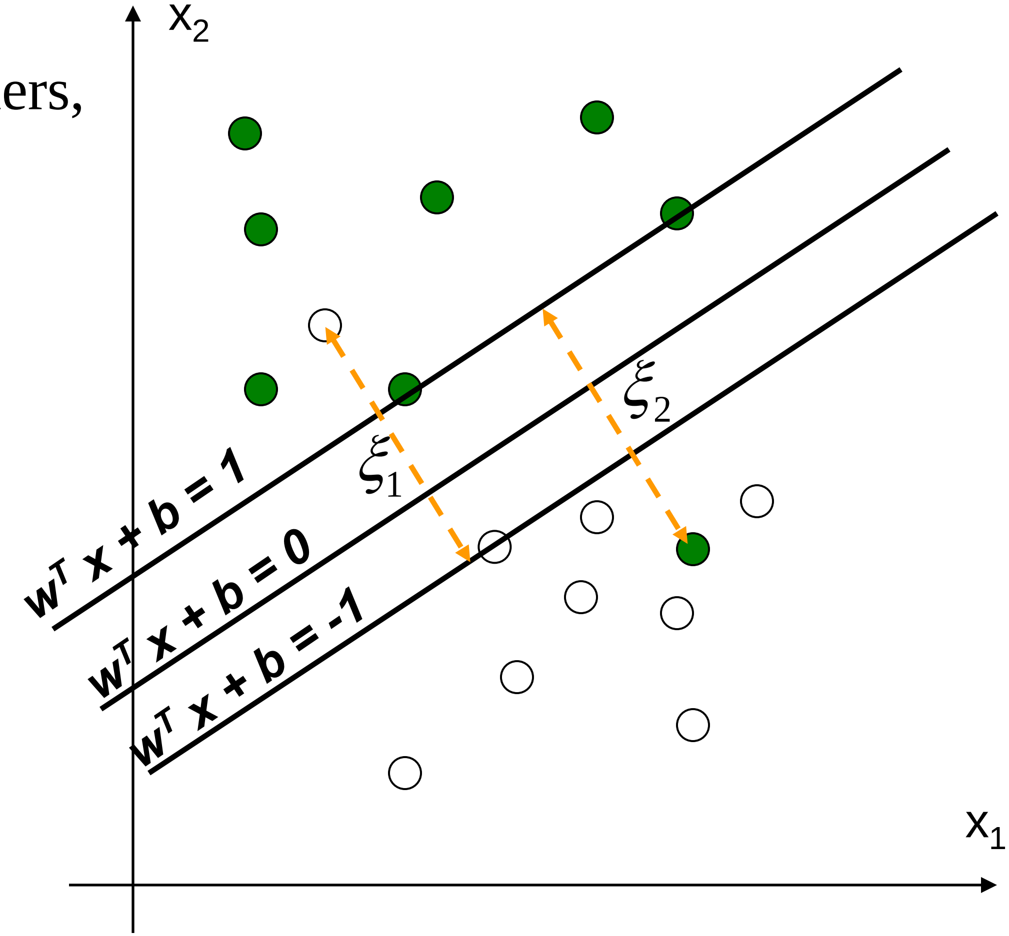
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

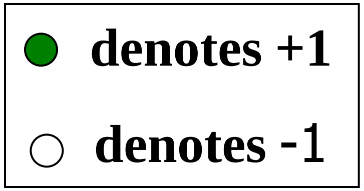


Large Margin Linear Classifier

- What if data is not linear separable? (noisy data, outliers, etc.)

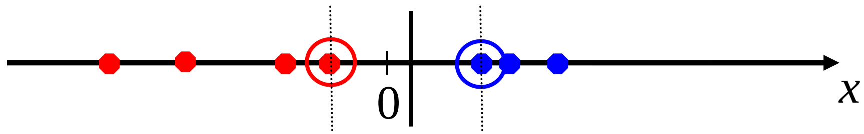
- Slack variables ξ_i can be added to allow mis-classification of difficult or noisy data points



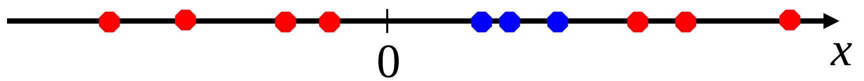


Non-linear SVMs

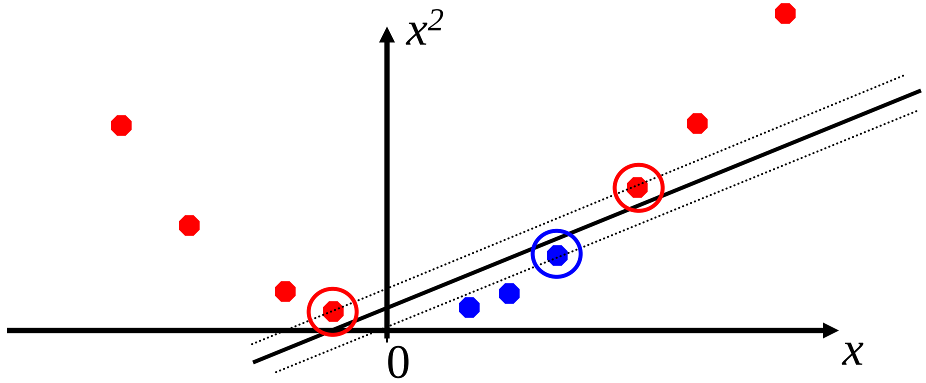
- Datasets that are linearly separable with noise work out great:



- But what are we going to do if the dataset is just too hard?

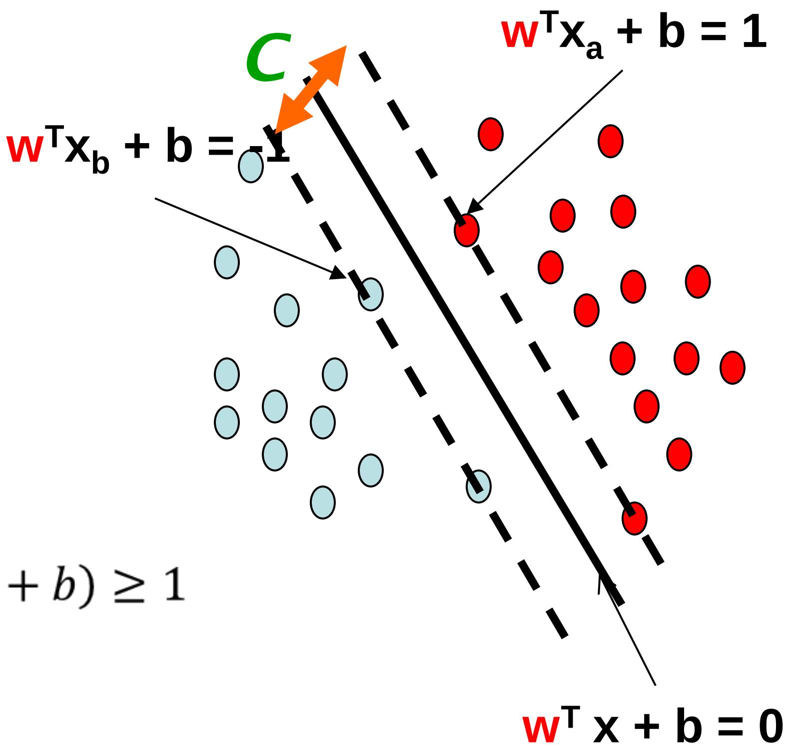


- How about… mapping data to a higher-dimensional space:



Support Vector Machine (SVM)

- Supervised learning model used for classification and regression analysis
  - SVMs maximize the margin around the separating hyperplane.
  - Hyperplane equation
    $$w^T x + b = 0$$
  - Extra scaling hyperplane equation
    $$\min_{i=1,\dots,n} |w^T x_i + b| = 1$$
  - This implies
    $$w^T (x_a - x_b) = 2$$
    $$c = \|x_a - x_b\|_2 = 2/\|w\|_2$$
  - SVM formulation
    $$\min_{w} \frac{1}{2} w^T w \quad Subject\ to\ \ y_i(w^T x_i + b) \geq 1$$



Non-Linear SVM: Feature Mapping

The original input space mapped to some higher-dimensional feature space where the training set is separable:

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

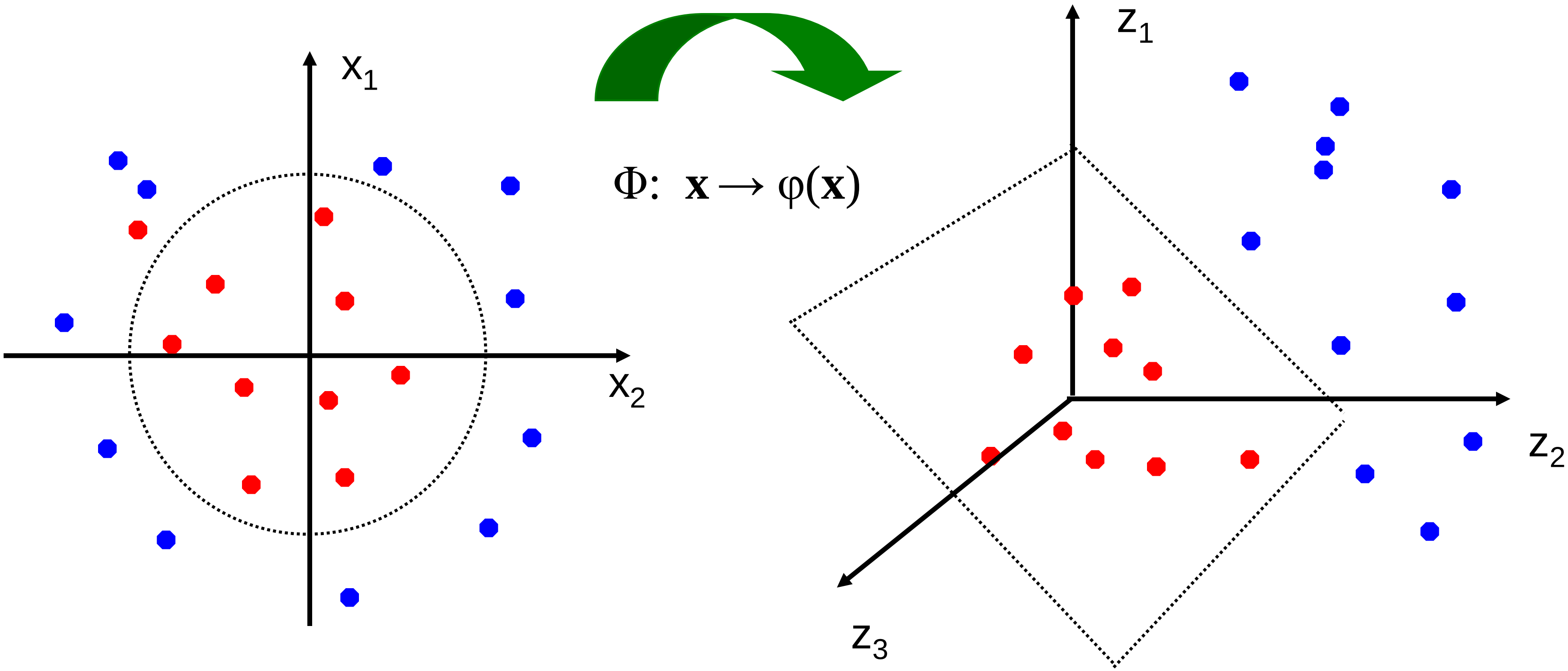
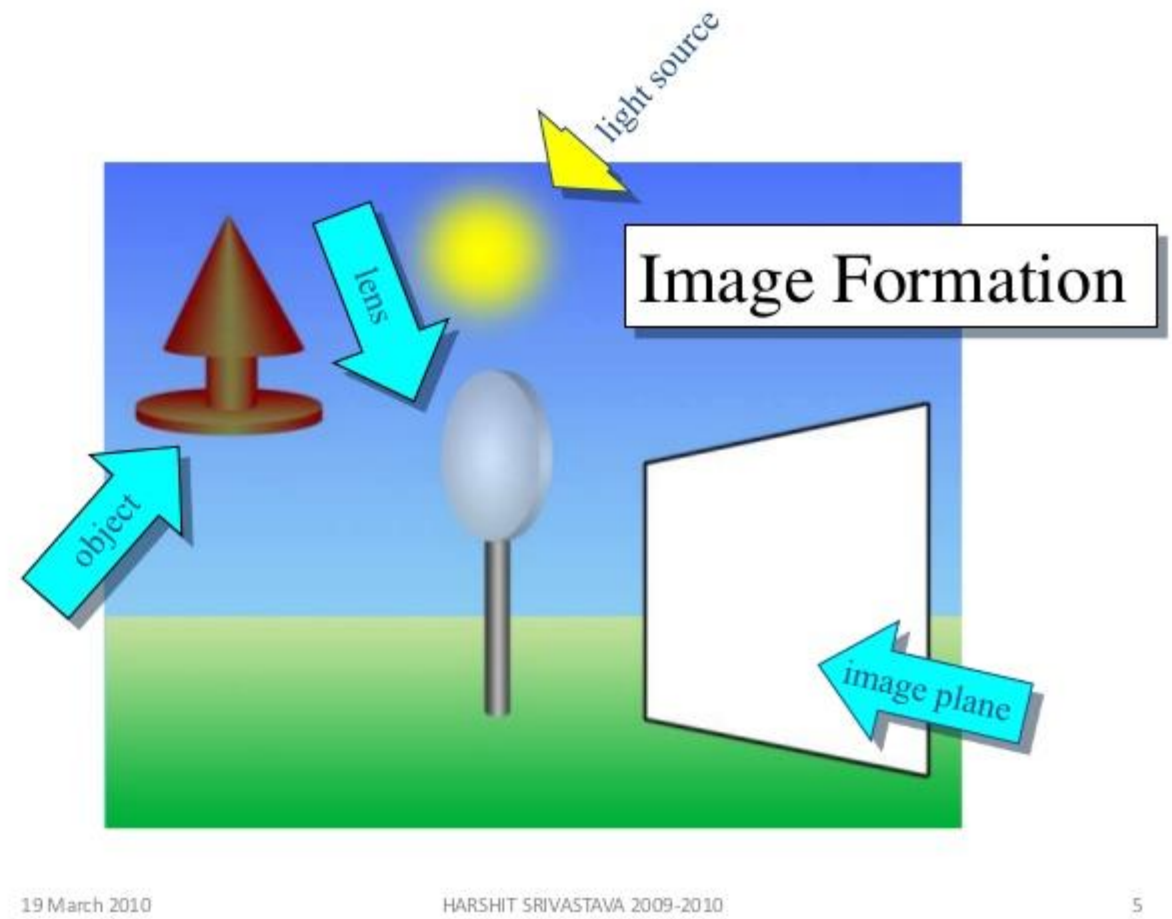


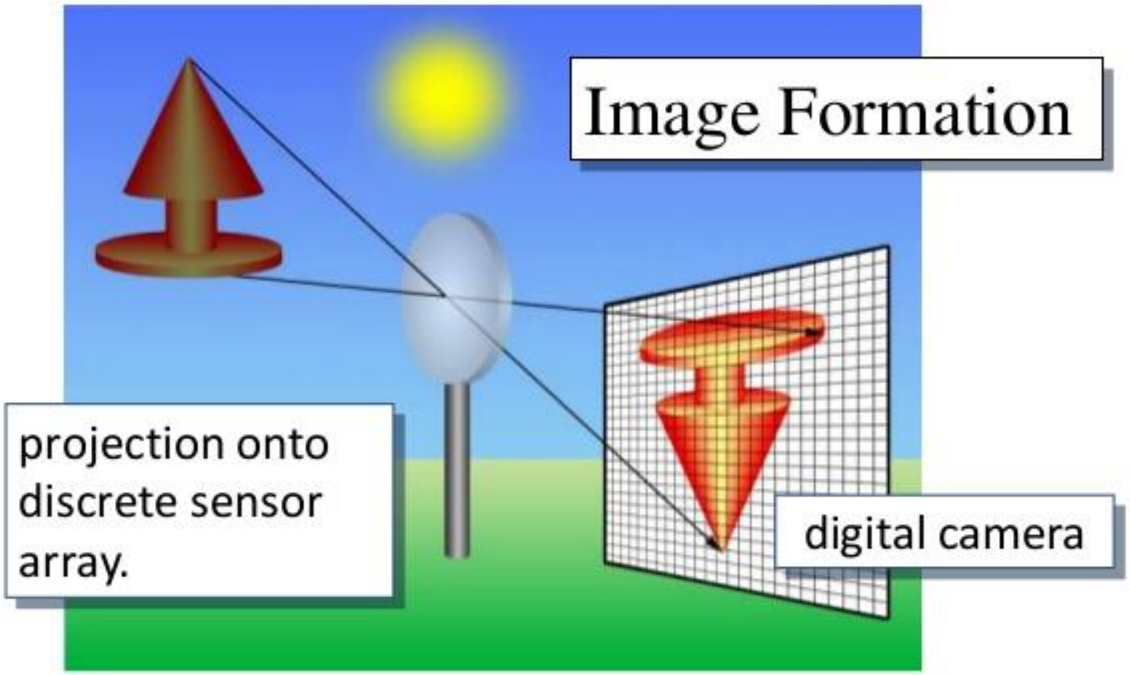
Fig 4.2: Nonlinear SVMs: The Kernel Trick

Applications

- Face detection
- Object detection and recognition
- Image segmentation
- Multimedia event detection
- Economical and commercial usage

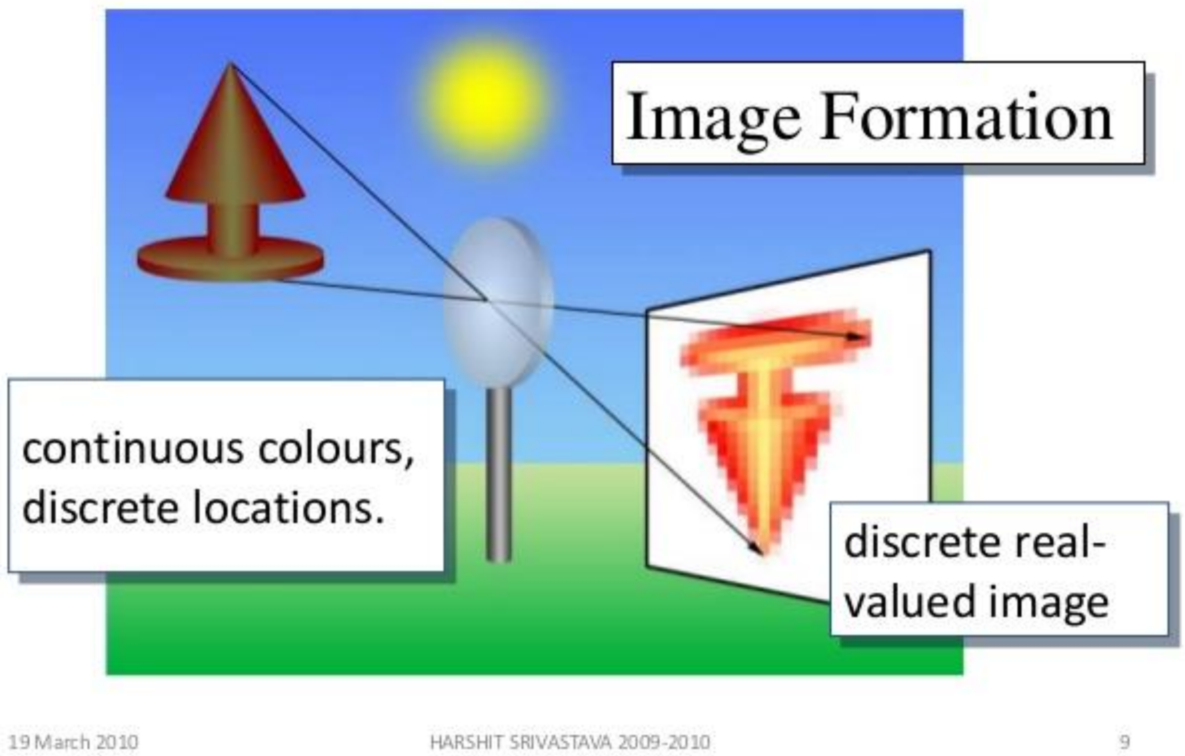

light source
lens
Image Formation
object
image plane
19 March 2010
HARSHIT SRIVASTAVA 2009-2010
5

Basic Models of ANN



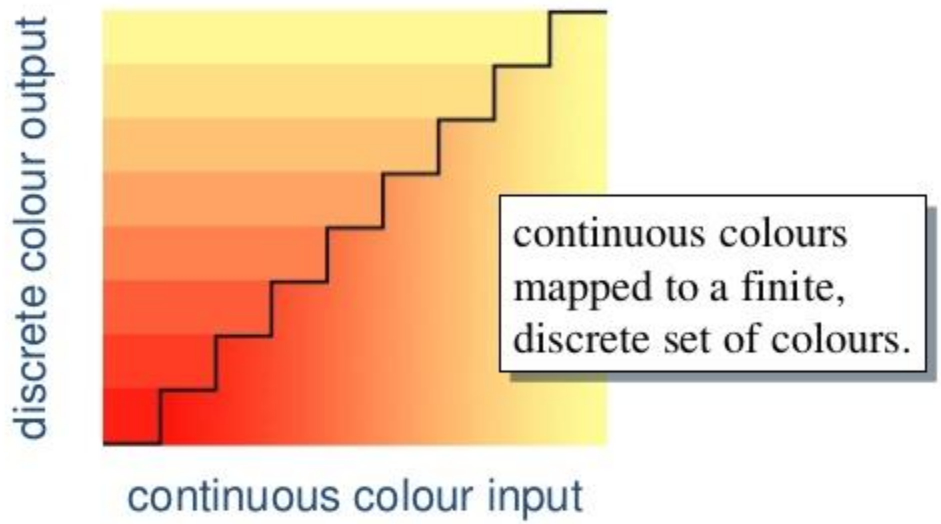
19 March 2010 HARSHIT SRIVASTAVA 2009-2010 7

Basic Models of ANN



Basic Models of ANN

Digital Image Formation: Quantization



19 March 2010 HARSHIT SRIVASTAVA 2009-2010 10

Sampling and Quantization

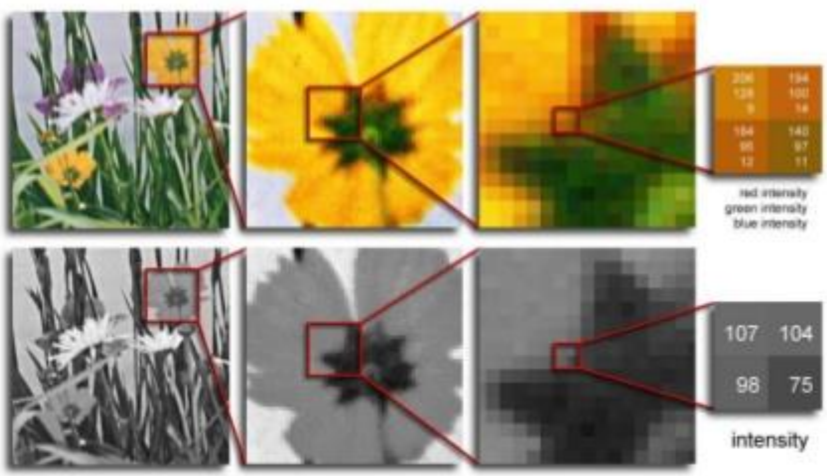
real image sampled quantized sampled & quantized

Digital Image

Colour images have 3 values per pixel; monochrome images have 1 value per pixel.

a grid of squares, each of which contains a single colour

each square is called a pixel (for *picture element*)



Point Processing

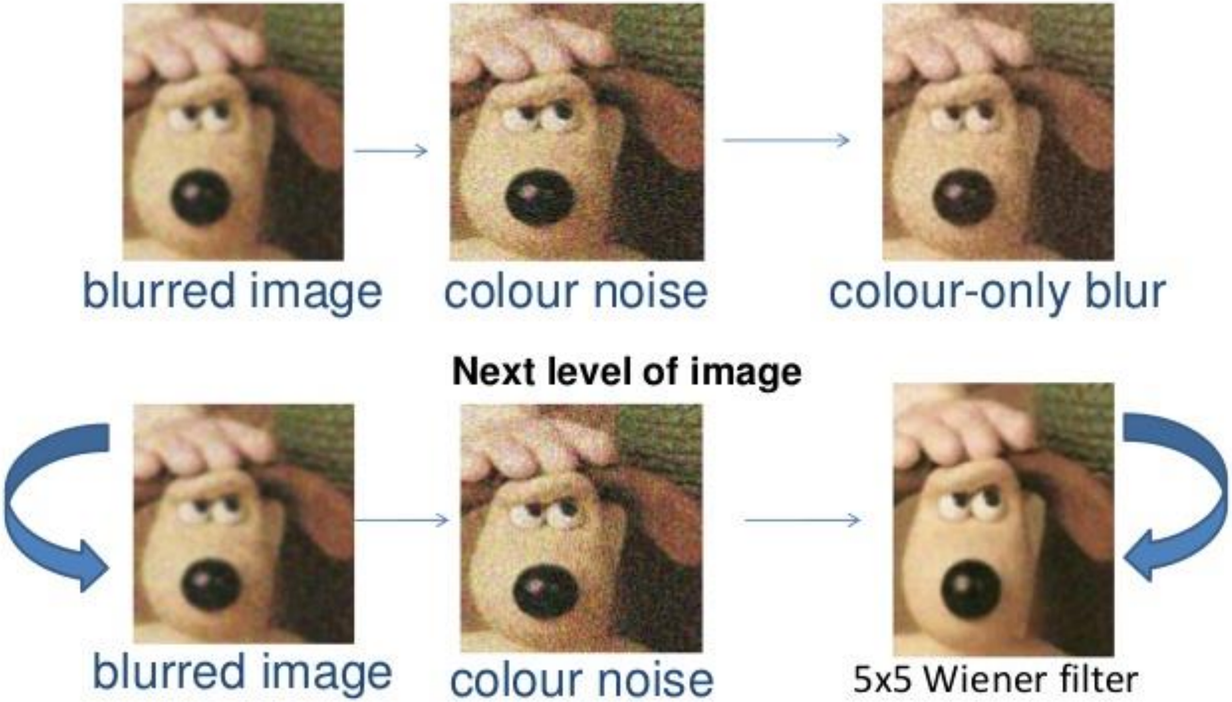
19 March 2010 HARSHIT SRIVASTAVA 2009-2010 17

Colour Balance and Saturation

Uniform changes in colour components result in change of tint.

E.g., if all G pixel values are multiplied by $\alpha > 1$ then the image takes a green cast.

Noise Reduction



Morphology

Nonlinear Processing: Binary Reconstruction

- Used after opening to *grow back* pieces of the original image that are connected to the opening.
- Permits the removal of small regions that are disjoint from larger objects without distorting the small features of the large objects.

original

opened

reconstructed