EE7207 Neural Networks and Deep Learning

Mao Kezhi

School of Electrical and Electronic Engineering
Nanyang Technological University

Email: ekzmao@ntu.edu.sg
Office: S2-B2c-84

Teaching Team

- Mao Kezhi
  - ❑ Course coordinator
  - ❑ Weeks 1-7

- Luo Wuqiong
  - ❑ Weeks 8-10

- Zhang Handuo
  - ❑ Weeks 11-13

Continuous Assessments and Exam

- Continuous Assessments (CA)
  - 40%
  - 2 assignments (released on Week 3, Week 9)
  - 2 quizzes (on Week 7, Week 13),
  - 10% each

- Exam
  - 60%
  - 4 questions (MK 2, LWQ 1, ZHD 1)
  - 25 marks each

References

1. Simon Haykin, Neural Networks and Learning Machines, 3rd Edition, Prentice Hall, 2009.

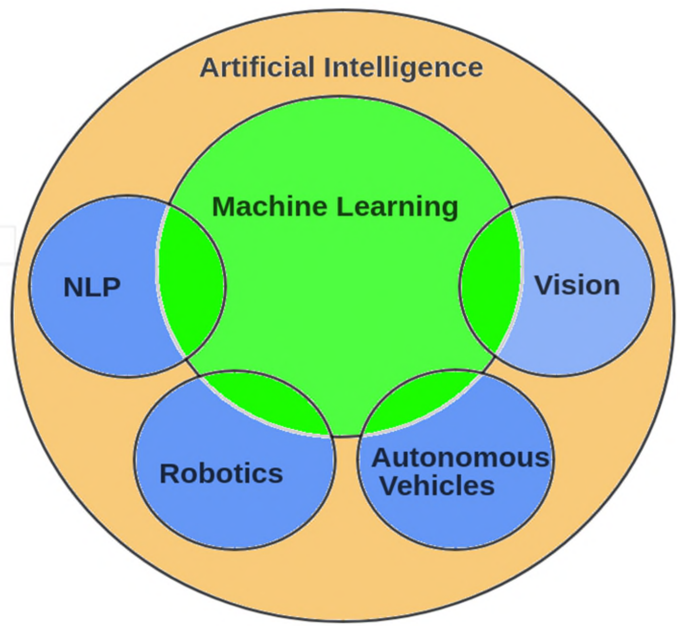
2. Deep Learning, by Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016, MIT Press.

3. Materials on the Internet, in conference proceedings and journals.

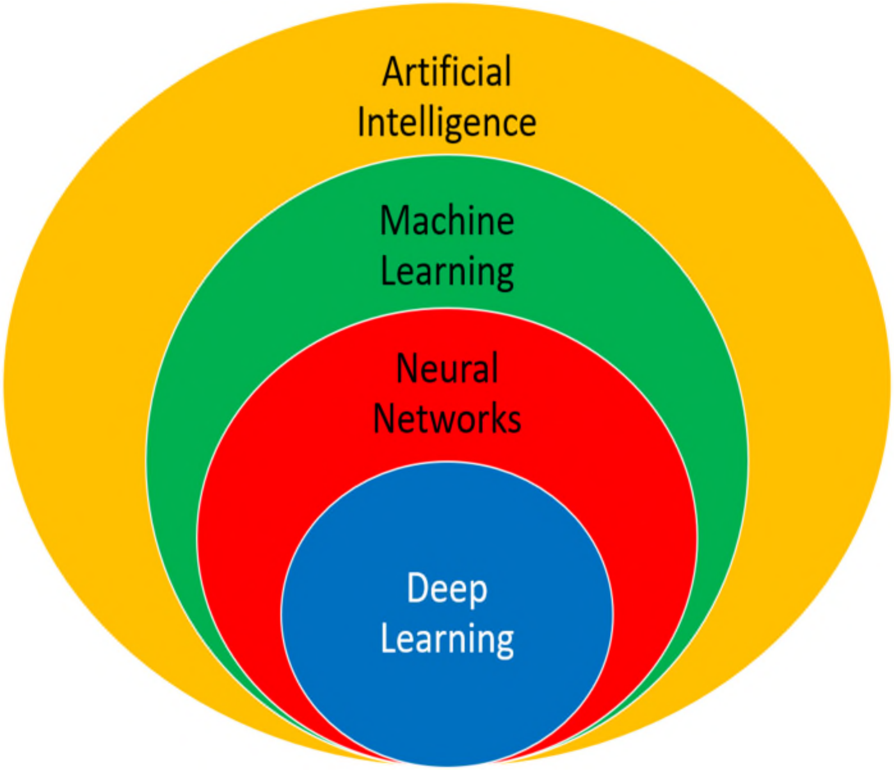
AI, Machine Learning, Neural Networks and Deep Learning: at a Glace

- **AI and Machine Learning**

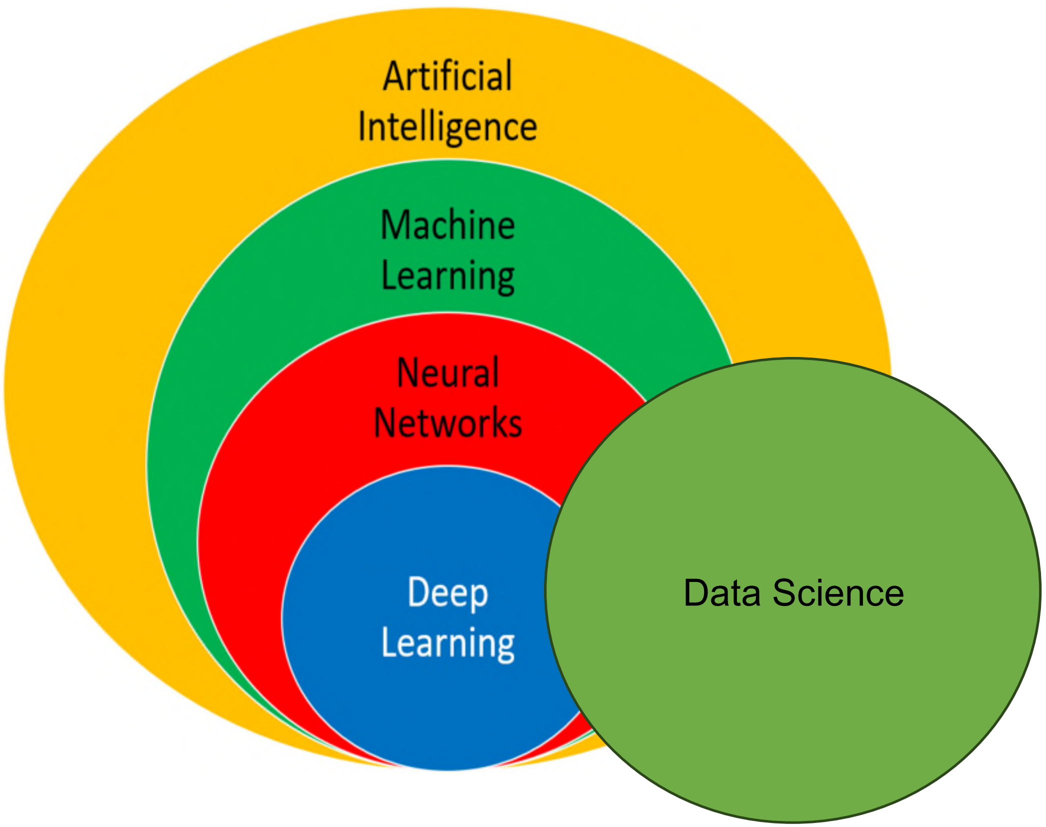
  - ❑ AI is a broad term for techniques which enable machines to mimic human behaviours
  - ❑ Machine learning is a sub-set of AI techniques that enable machines to improve with experience
    - ➢ Data-driven
    - ➢ Statistical algorithms



- **Machine Learning, Neural Networks and Deep Learning**
  - ❑ Neural networks is a sub-field of machine learning
  - ❑ Deep learning is a sub-field of neural networks

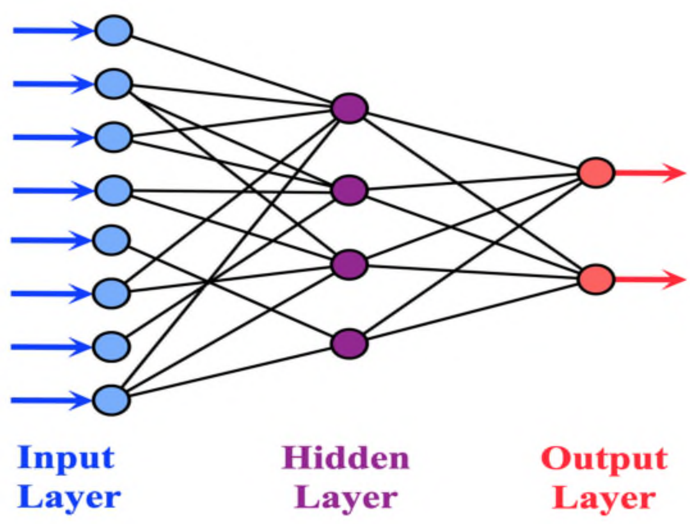


- **Data Science and Machine Learning**
  - ❑ Data science is a broad term covering everything about data
  - ❑ Machine learning focuses on learning algorithms: how to learn from data



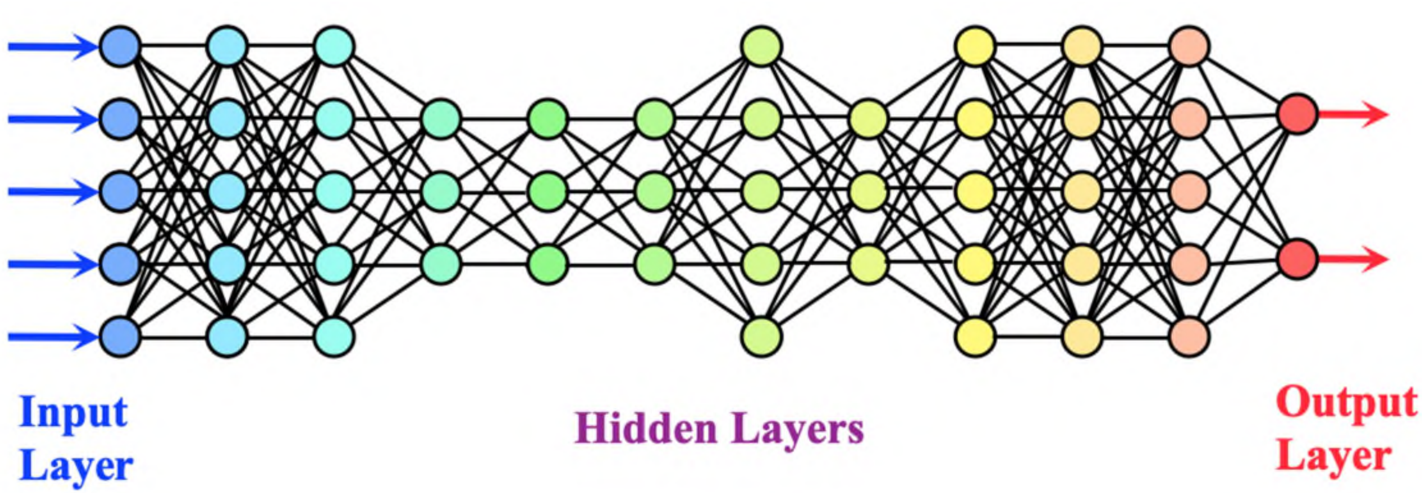
- **Shallow and Deep Neural Networks**

  - ❑ Shallow neural networks
    - ➢ **Structure**: with one or two layers, excluding input layer.
    - ➢ **Strengths**: relatively easy to train; require less computing resource, demand less training data; sufficient for some tasks.
    - ➢ **Limitations**: ability is limited when comes to capturing complex relationships or features in data like images, text, and speech.



❑ Deep neural networks

- **Structure**: with many layers.
- **Strengths**: excel at extracting features and relationships from complex data, leading to breakthroughs in fields like computer vision and natural language processing.
- **Limitations**: training can be computationally expensive; require large amounts of training data; prone to overfitting.



Contents

1. Introduction to Neural Networks

- An Overview of Neural Networks
- Neuron Models and Network Architectures
- Neural Network Learning

1. An Overview of Neural Networks

- **Why Artificial Neural Networks?**

  As modern computers become ever more powerful, scientists continue to be challenged to use machines effectively for some tasks that are relatively simple to humans. A good example is the processing of visual information. A one-year-old baby is much better and faster at recognizing objects, faces and so on than many AI systems running on the fastest computer.

  Computers excel in arithmetic but are less effective than human brains in many fields. Human brains have many other features that would be desirable in AII systems.

- **Features of Human Brain**

  - Human brain is robust and fault tolerant. Nerve cells in the brain die every day without affecting its performance significantly;
  - Human brain is flexible. It can easily adjust to a new environment by "learning"---it does not have to be programmed in any language;
  - Human brain can deal with complex information that is fuzzy, probabilistic, noisy, or inconsistent;
  - Human brain is highly parallel and highly nonlinear.

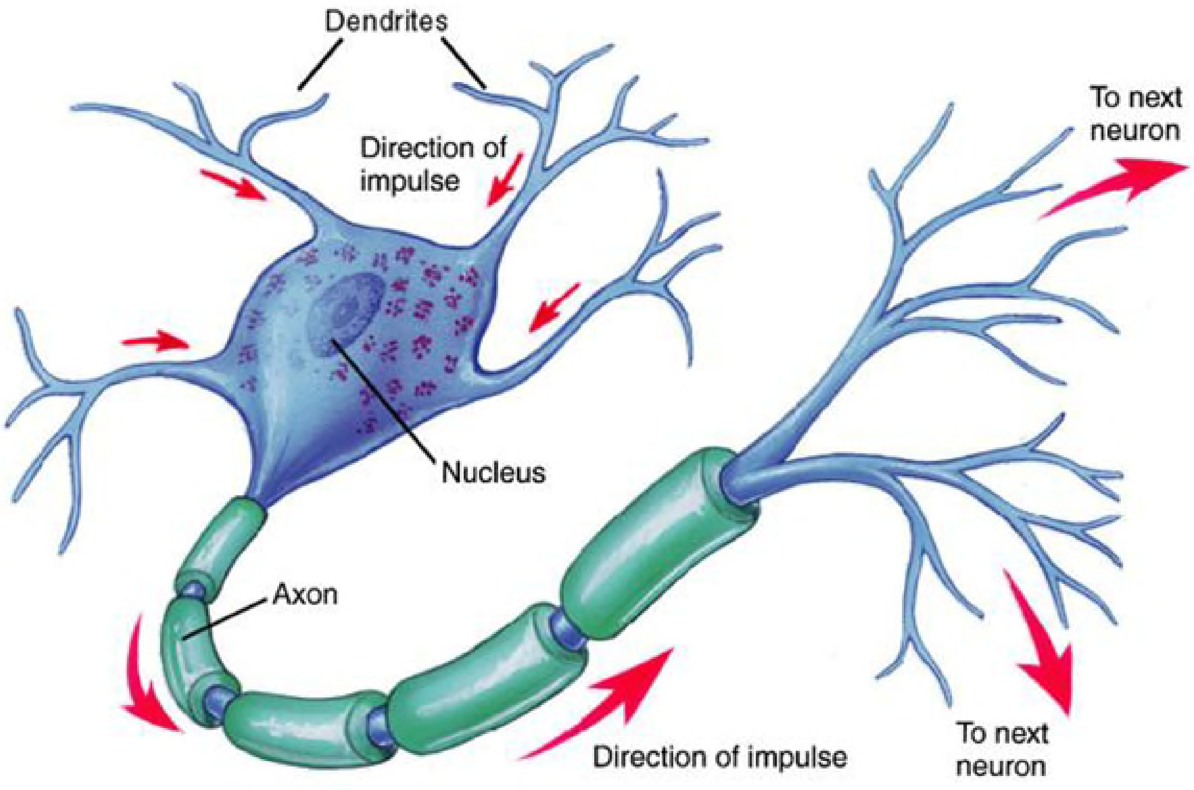
It can be said that human brains outperform computers in many areas. This is the motivation for studying neural computation.

- **Biological Origin of Artificial Neural Networks**

The nervous system consists of two classes of cells: neurons (nerve cells) and glia (glial cells). Neurons are the basic building blocks of biological information processing systems, and the glial cells perform more of a support function. Therefore, neurons are of major concern.

A biological neuron consists of three major portions:

- ❑ *Cell body*: information processing unit;
- ❑ *Axon*: the main conduction mechanism of neurons;
- ❑ *Dendrites*: facilitate excitatory and inhibitory functions in axon signal generation.

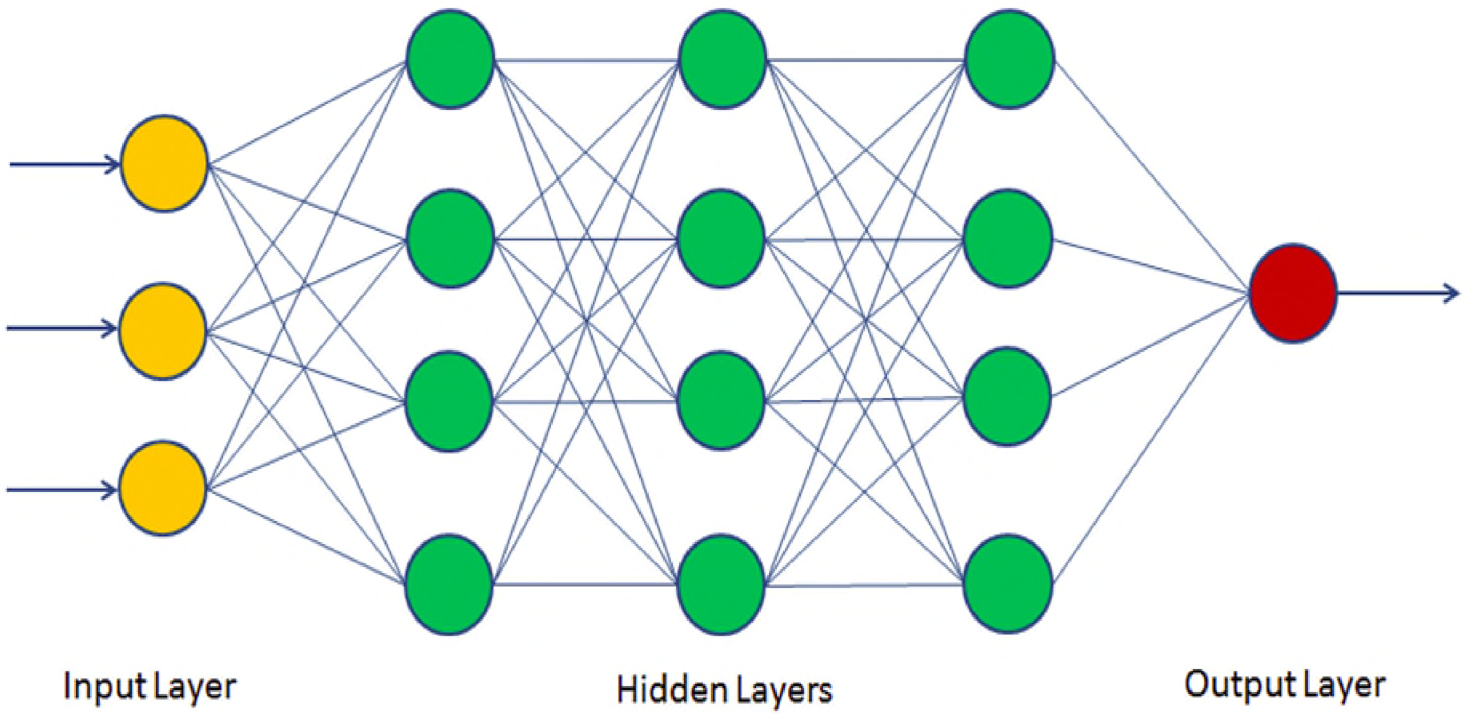


Sketch of a biological neuron

- **What is An Artificial Neural Network?**

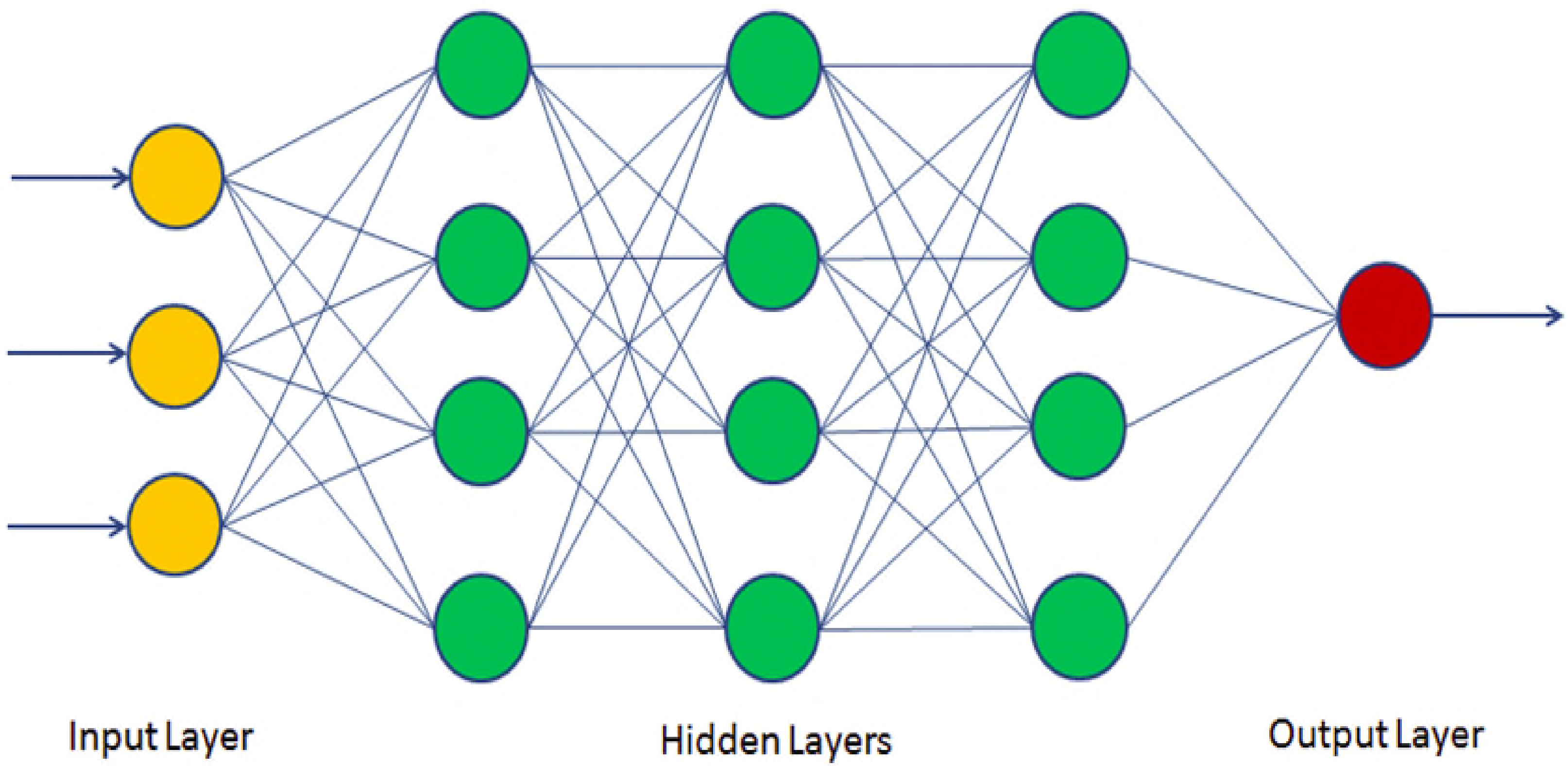
An artificial neural network is an information processing system that has certain performance characteristics in common with biological neuron networks. Artificial neural networks have been developed as generalizations of mathematical models of human neural biology, based on the following assumptions:

- Information processing occurs at a large number of simple elements called neurons;
- Signals are passed between neurons over connection links;
- Each connection link has an associated weight, which multiplies the signal transmitted;
- Each neuron applies an activation function to the input to determine the output signal.



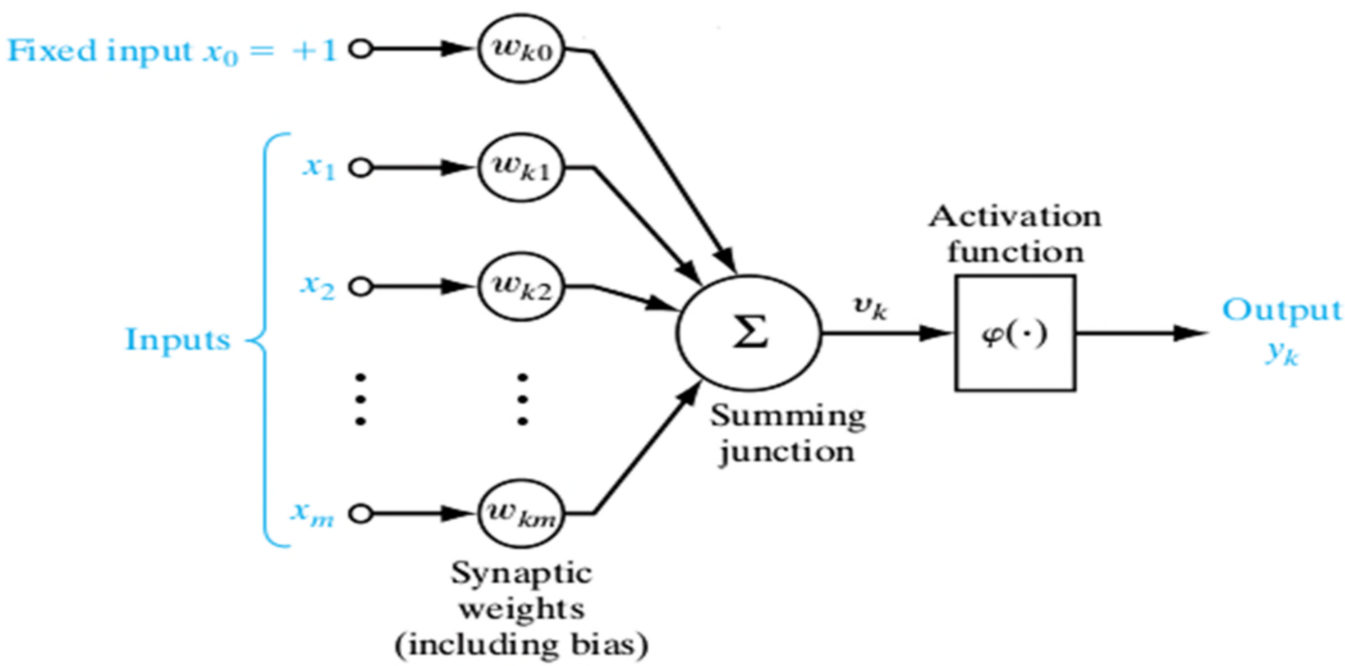
The diagram of a typical artificial neural network

2. Neuron Models and Network Architectures



- **Neuron Models**

A neuron is an information processing unit that is fundamental to the operation of a neural network. The model of a neuron is shown below:



We may identify three basic elements of the neuron model:

- *A set of synapses or connecting links*, each of which is characterized by a weight or strength of its own. A signal x_j at the input of synapse j connected to the neuron k is multiplied by the weight w_{kj};
- *An adder* for summing input signals. The adder actually performs a linear combination or weighted summation of the input signals;
- *An activation function* for nonlinear operation and limiting the amplitude of the neuron output. Typically, the amplitude of the neuron output is in the range of [0,1] or [-1 1], depending on the activation function used in the neuron.

In mathematical terms, we can describe the neuron as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

$$y_k = \varphi(v_k)$$

Where

$x_0, x_1, \dots, x_m$ are the input signals

$w_{k0}, w_{k1}, \dots, w_{km}$ are the weights of neuron k

v_k is the activation signal of neuron k

$\varphi(.)$ is the activation function

y_k is the output of neuron k.

- **Activation Functions**

The activation function defines the output of a neuron in terms of the activation level at its input. We may identify a few types of activation functions:

(1) Binary Function:

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

---uni-polar binary

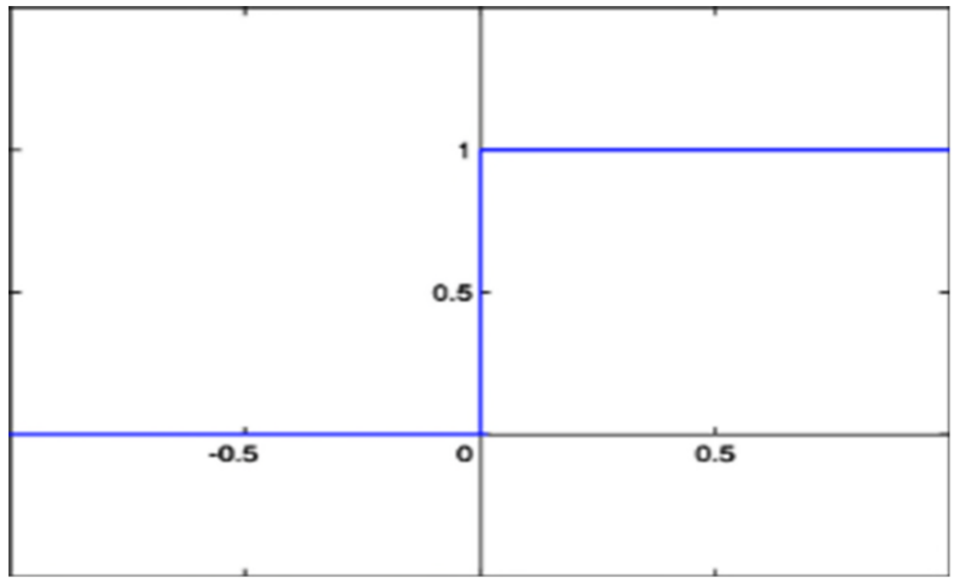
$$f(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

---bi-polar binary

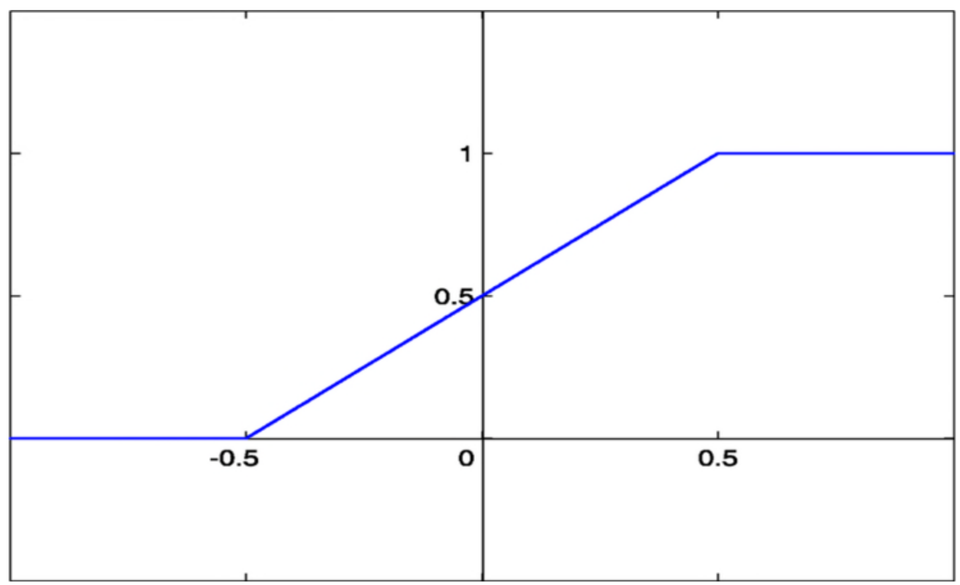
(2) Piecewise-Linear Function

$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ x + 0.5 & -0.5 < x < 0.5 \\ 0 & x \leq -0.5 \end{cases}$$

Where the amplification factor inside the linear region is set to unity.



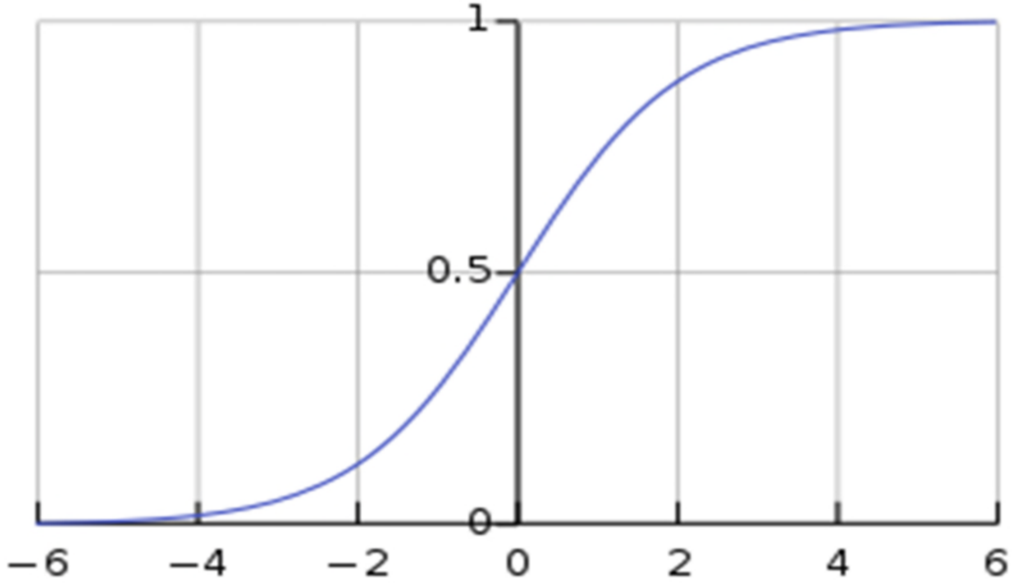
uni-polar binary



piecewise linear

(3) Sigmoid Function

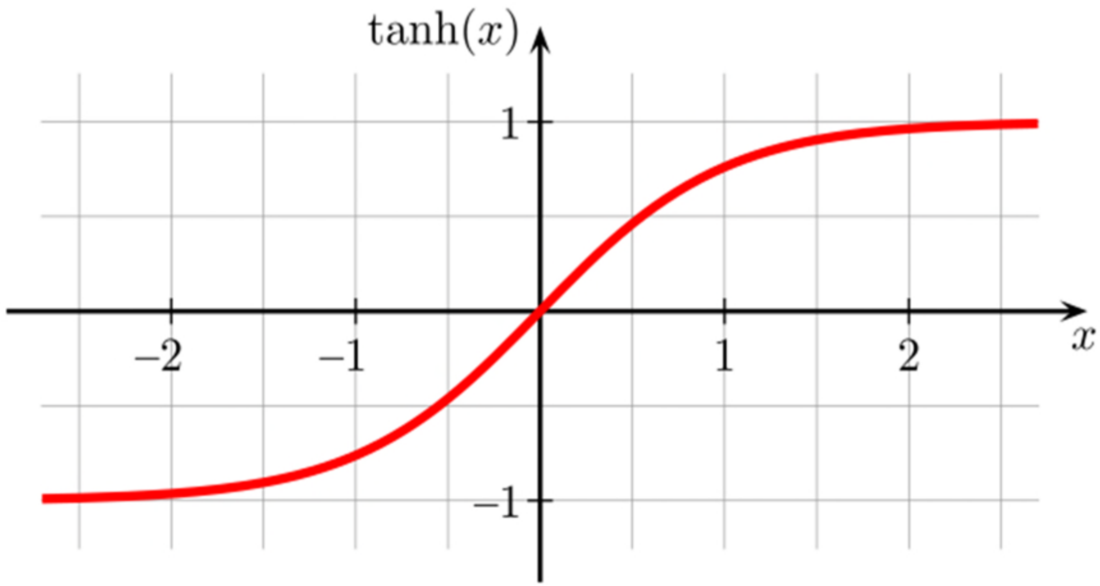
$$f(x) = \frac{1}{1 + \exp(-x)}$$



The output of the neuron is in the range of (0,1).

(4) Hyperbolic Tangent Function

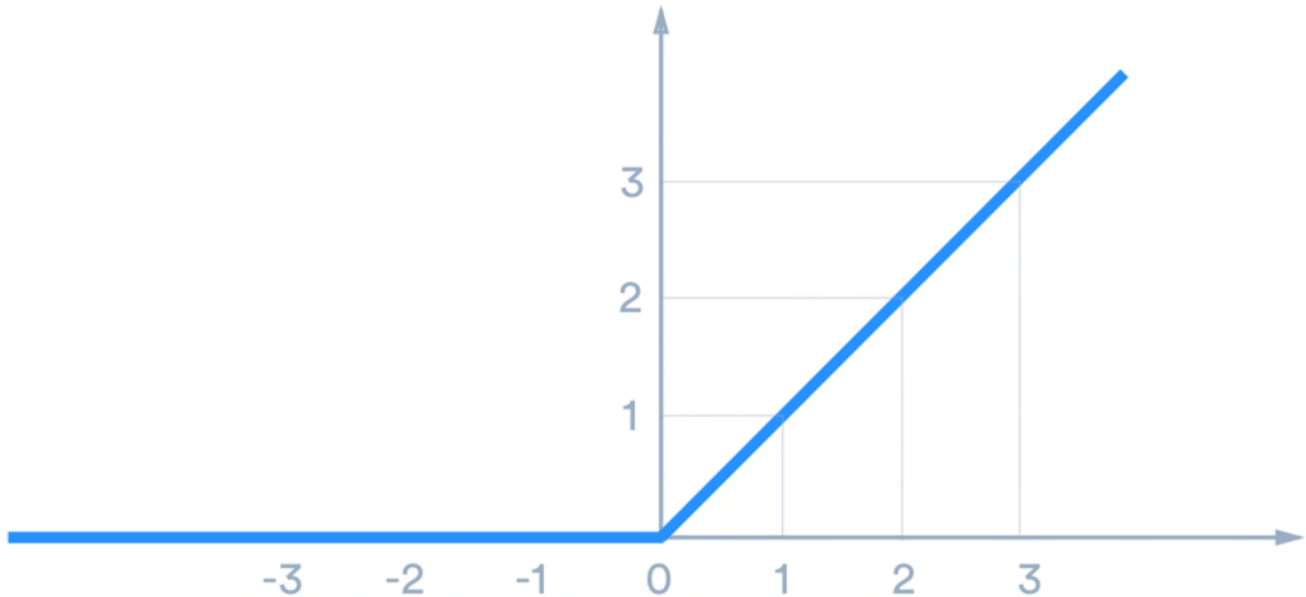
$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$



The output of the neuron is in the range of (-1,1).

(5) Rectified Linear Unit

$$\mathrm{ReLU}(x) = \max(0, x)$$

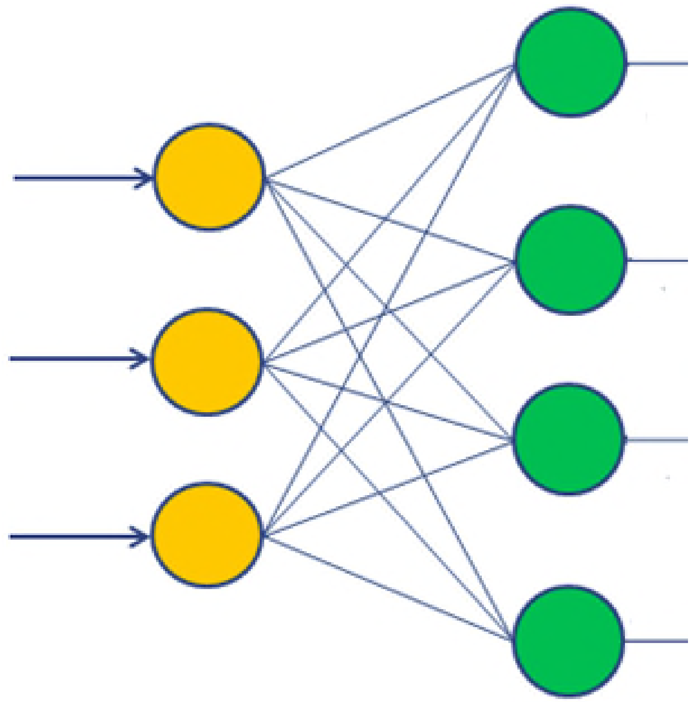


- **Neural Network Architectures**

In general, we may identify two different classes of network architectures.

(1) Feed-Forward Neural Networks

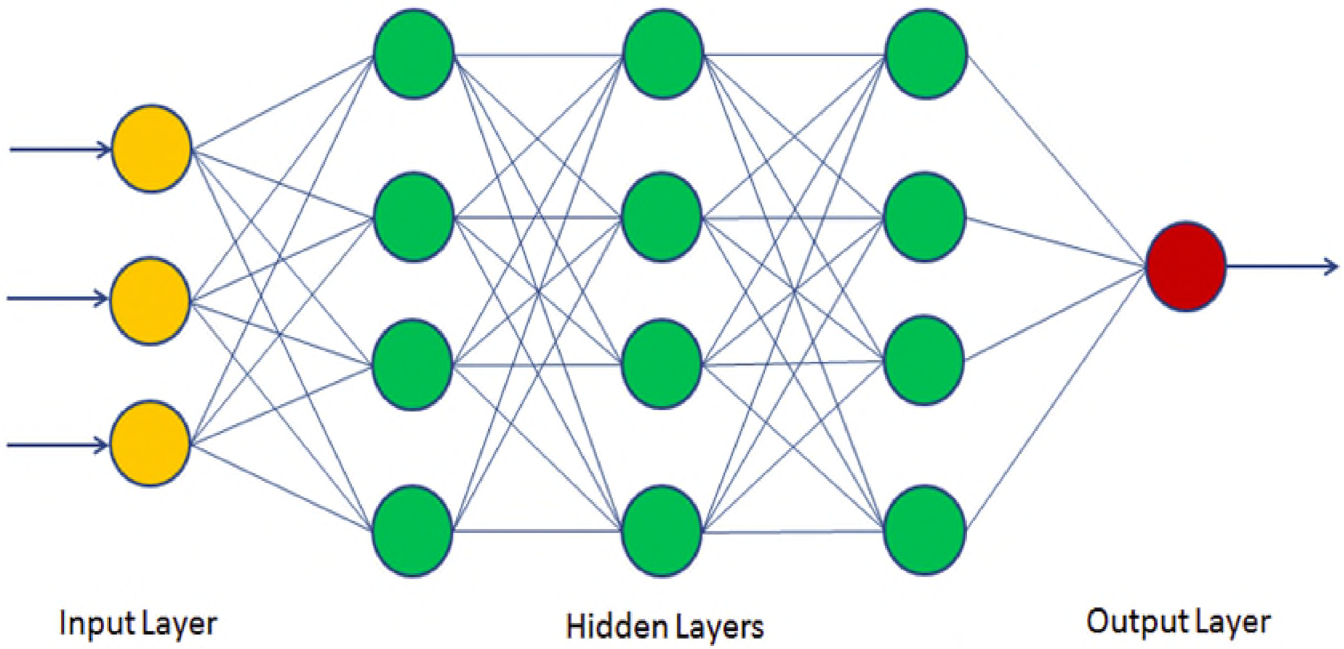
A layered neural network is a network of neurons organized in the form of layers. In the simplest form of a layered network, we just have an input layer of source nodes that projects onto the output layer of neurons. This kind of structure is called single-layer feed-forward network. The term *single-layer* refers to the output layer of the computation neurons. In other words, we do not count the input layer of source neurons because the input layer neurons just distribute the input signals to neurons of the output layer and do not perform any computation.



The single-layer feed-forward network

The second class of the feed-forward network is the multilayer neural network. The multilayer feed-forward neural network distinguishes itself from the single-layer network by the presence of one or more hidden layers. By adding one or more hidden layer, the network is able to approximate severe non-linearity.

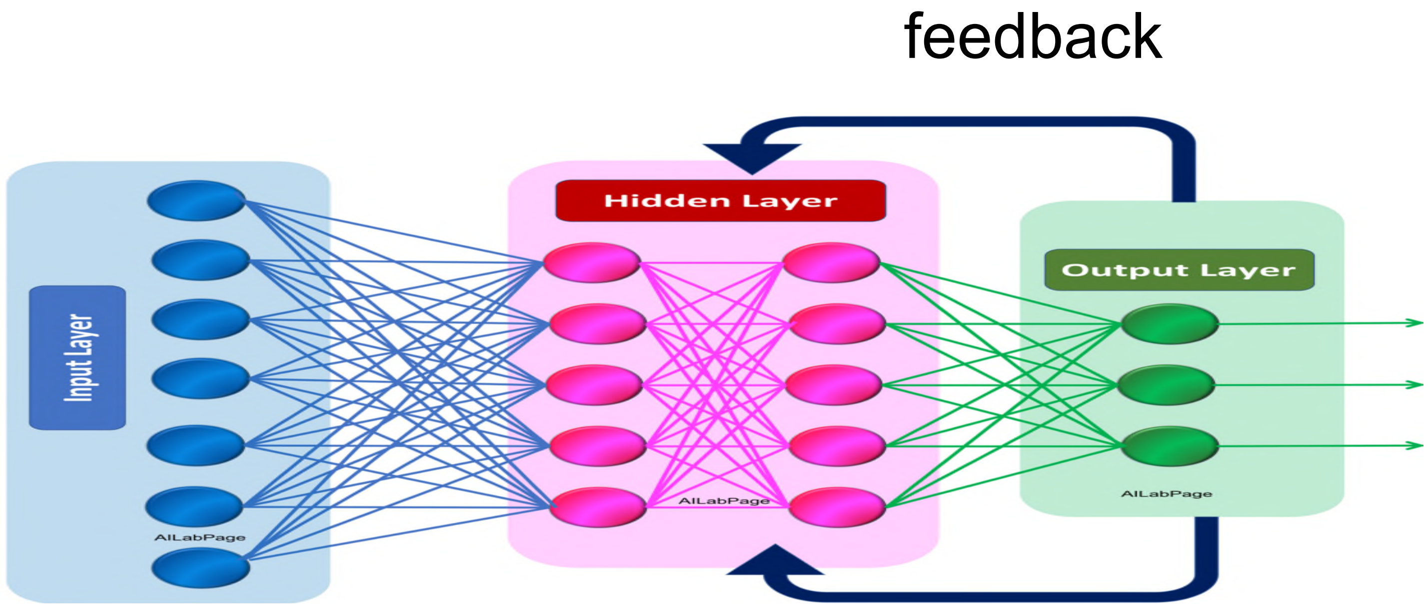
In the following architecture, the hidden layer neuron outputs are used as inputs to the third layer, and so on for the rest of the network. However, it is not necessary to use all external input signals or hidden layer neuron outputs to the input of neurons at the next layer, for example, the dropout in deep neural networks.



Multilayer feed-forward neural network

(2) Recurrent Neural Networks

A recurrent neural network distinguishes itself from the feed-forward network in that it has at least one feedback loop as shown below:



Recurrent neural network

- **How Are Neural Networks Used?**

The use of neural networks comprises two phases:

- **Phase 1: neural network development**

In this phase, an appropriate architecture is selected for the neural network. The weights of all neurons are then determined. The development phase involves *training* of the neural network and *testing* of the neural network trained.

- **Phase 2: neural network deployment**

In this phase, the neural network developed in phase 1 is deployed to newly acquired data that are not seen in the development phase.

- **How to Train and Test Neural Networks?**

Training a neural network is to determine the weights in the neural network based on the known knowledge of the world using suitable learning algorithms.

Knowledge of the world consists of two types of information:

- ❑ The known world state, represented by facts about what is and what has been known. This form of knowledge is often referred to as *prior information*.
- ❑ Observations of the world, obtained by sensors designed to probe the world. These observations are often referred to as *samples*, *examples or data,* which can be labeled or unlabeled.

- **Labelled and unlabelled data**

  (a) In labeled samples, each sample representing an input signal is paired with a target value, which is called class label in pattern classification.

  Example 1: Object Recognition

Class label: Dog

Class label: Cat

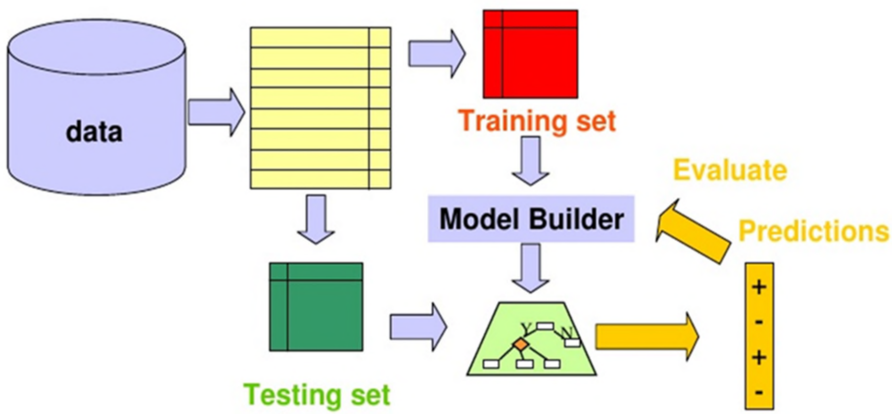
Example 2: Sentiment Analysis

	Sentiment	Tweets
Class label:	**Negative**	@united is the worst. Nonrefundable First class tickets? Oh because when you select Global/FC their system auto selects economy w/upgrade.
		@united I will not be flying you again
Class label:	**Neutral**	@VirginAmerica my drivers license is expired by a little over a month. Can I fly Friday morning using my expired license?
		@VirginAmerica any plans to start flying direct from DAL to LAS?
Class label:	**Positive**	@VirginAmerica done! Thank you for the quick response, apparently faster than sitting on hold ;)
		@united I appreciate your efforts getting me home!

(b) In unlabeled samples, just the input signal is available, the target value is unknown. For example, a basket of fruits, without naming the fruits:

- **Training and Testing Data**

We often divide the data into two subsets. One subset is called *training set*, and the other is called *testing set*.

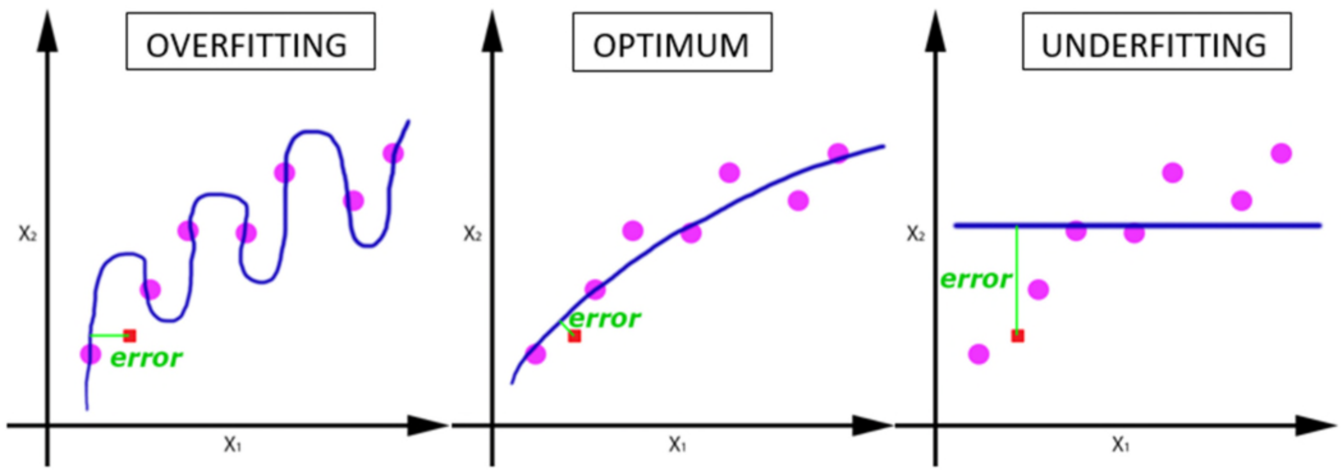


The training set is used to train the neural network, and the testing set is used to evaluate the performance of the neural network trained. Quite often, an additional set, called validation set, is used for determining the hyperparameters of the neural network.

- **Overfitting and Under-fitting**

If a network performs well on the training set but very badly on the testing set, the network might be *over-trained (overfitting)*.

On the other hand, if a network performs badly on the training set, the network might be *under-trained (under-fitting)*. An under-trained network also performs badly on the testing set.



3. Neural Network Learning /Training

In the context of neural networks, *learning or training* is defined as a process by which the free parameters (i.e. the weights) of a neural network are adapted through a continuing process of stimulation by the environment (e.g. data). The type of *learning* is determined by the manner in which the parameter update takes place.

The definition implies that:

- The neural network is stimulated by the environment;
- The neural network is updated as a result of stimulation;
- The neural network responds in a new way after the update.

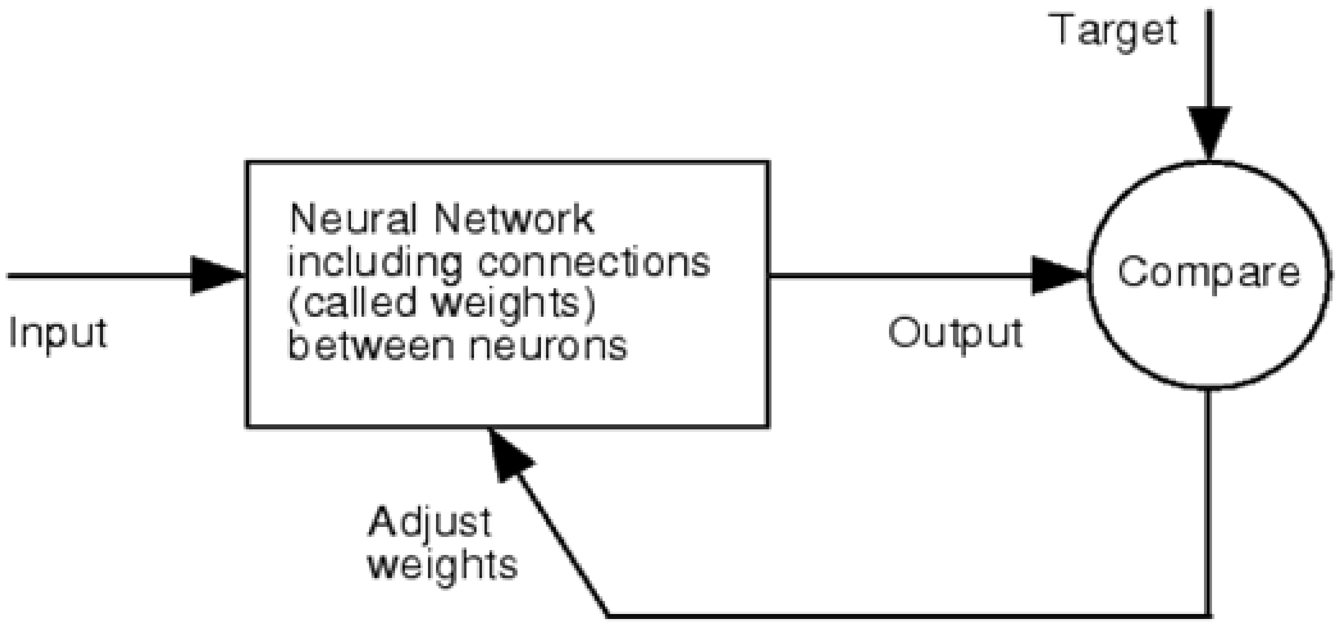
Next, we will introduce three basic *learning rules* and two *learning paradigms*.

- **Three learning rules:**
  - Error-correction learning;
  - Hebbian learning;
  - Competitive learning.

- **Two learning paradigms:**
  - Supervised learning;
  - Unsupervised learning.

(1) Error-correction Learning

The learning of a neural network is the process of determining the weights of the network. In error-correction learning, the weight adjustment is based on the error, which is defined as the difference between the target value and the network output.



One example error-correction learning rule is of the following form:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$$

where η is a positive constant called *learning rate*. $e_k(n)$ is the error signal for neuron k at step n.

In summary, the adjustment of the weight, Δw_{kj}, in error-correction learning is proportional to the product of the error signal e_k and the corresponding input signal x_j.

(2) Hebbian Learning

Hebbian learning rule can be stated as:

- If two neurons on either side of a connection are activated simultaneously, then the strength of that connection (i.e. weight) is selectively increased.

- If two neurons on either side of a connection are activated asynchronously, then the strength of that connection is selectively weakened or eliminated

To formulate Hebbian learning in mathematical terms, consider a weight w_{kj} from input signal x_j to neuron k. The adjustment to the weight at step n is expressed in the general form:

$$\Delta w_{kj}(n) = f[x_j(n), y_k(n)]$$

Where f is a function. The above formula has many forms, the simplest form is as follow:

$$\Delta w_{kj}(n) = \eta x_j(n) y_k(n)$$

Where η is the *learning rate*.

In summary, the adjustment of the weight in Hebbian Learning, Δw_{kj}, is proportional to the product of the input signal x_j and output signal y_k.

(3) Competitive Learning

As its name implies, in competitive learning, the neurons of a neural network compete for being the winning neuron.

There are 3 basic elements in a competitive learning rule:

- *A set of neurons* that are all the same except for the weights, and which therefore respond differently to a given input;
- *A mechanism* that permits the neurons to compete. The neuron that wins the competition is called the *winner* neuron.
- *A mechanism* that allows the winning neuron (and its neighboring neurons) to update weights.

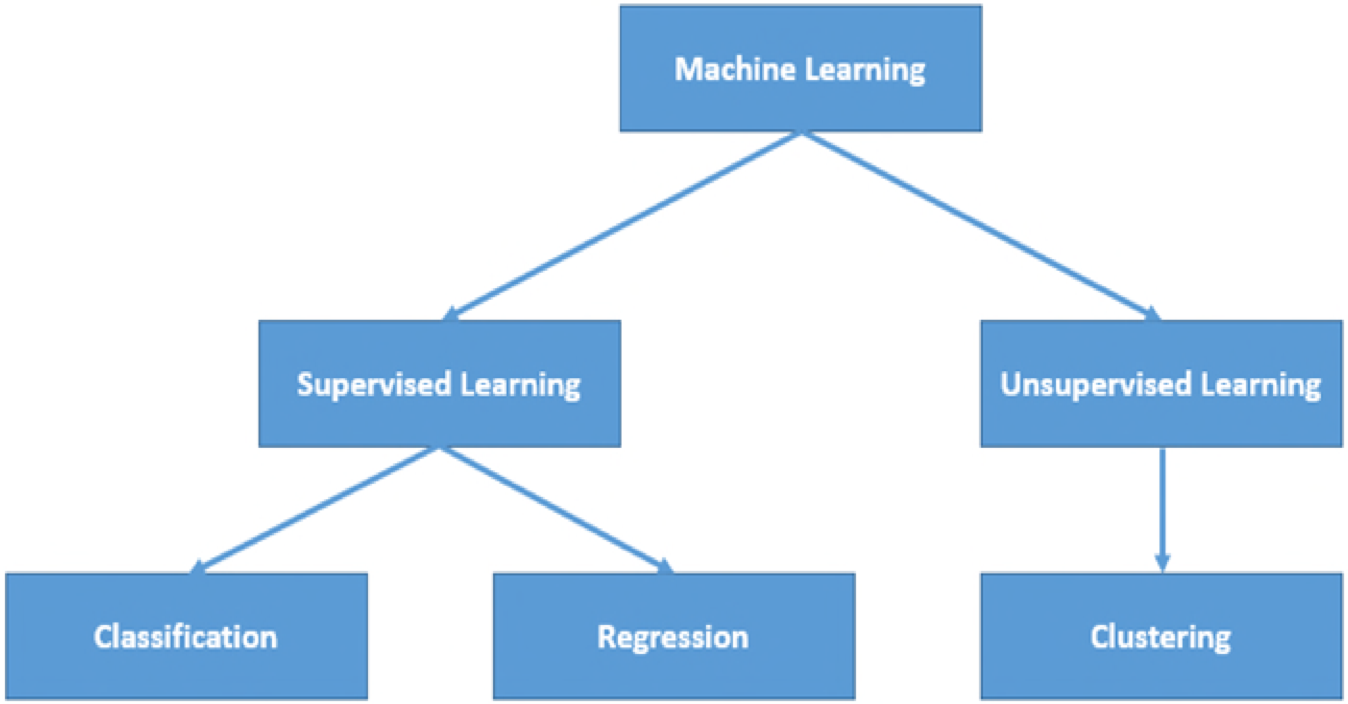
The weights of neuron k could be updated in the following way:

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{for winning neuron} \\ 0 & others \end{cases}$$

Where η is the *learning rate*.

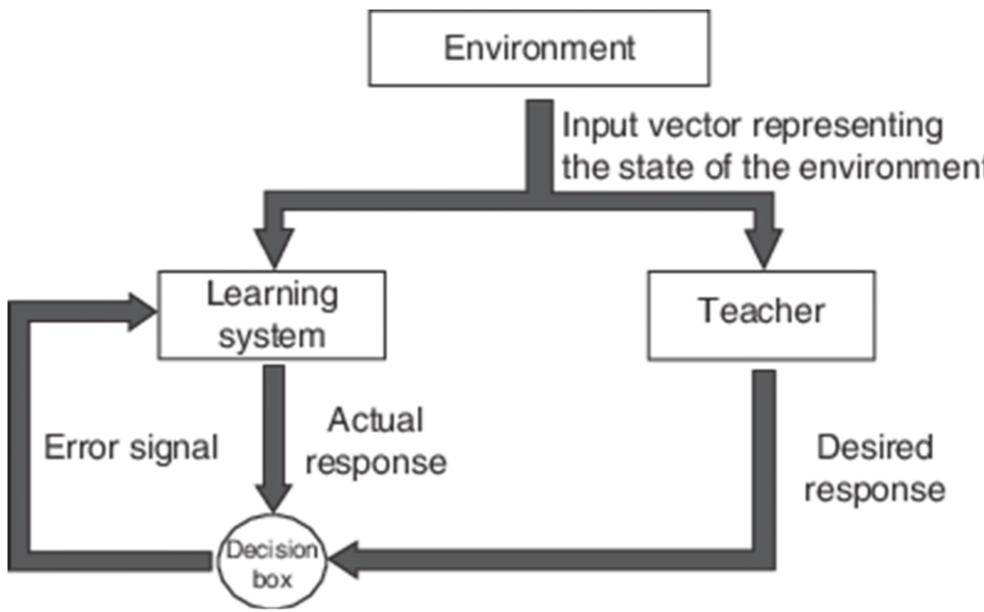
In summary, the adjustment of the weight, Δw_{kj}, in competitive learning is proportional to the difference of the input signal x_j and the weight w_{kj}.

- **Supervised vs Unsupervised Learning**



(1) Supervised Learning

During the supervised learning, an input is applied to the network, and a response is obtained. The response is compared with the desired response. The error signal is then used to compute the adjustments to the network's weights so that the actual response matches the desired response.

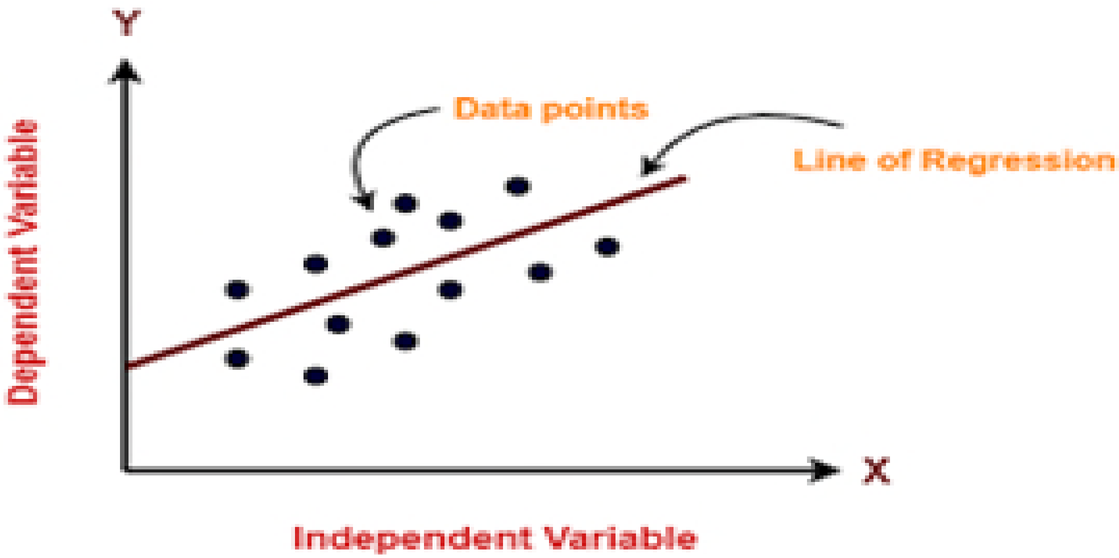


Since the learning process requires a teacher (supervisor) to provide desired response (i.e. target value), this kind of learning is named as *supervised learning*.

The notion of teacher comes from biological observations. For example, when learning a language, we hear the sound of a word from a teacher. The sound is stored in the memory banks of our brain, and we try to reproduce the sound. When we hear our own sound, we mentally compare it (actual response) with the stored sound (desired response) and note the error. If the error is large, we try again and again until it becomes significantly small.

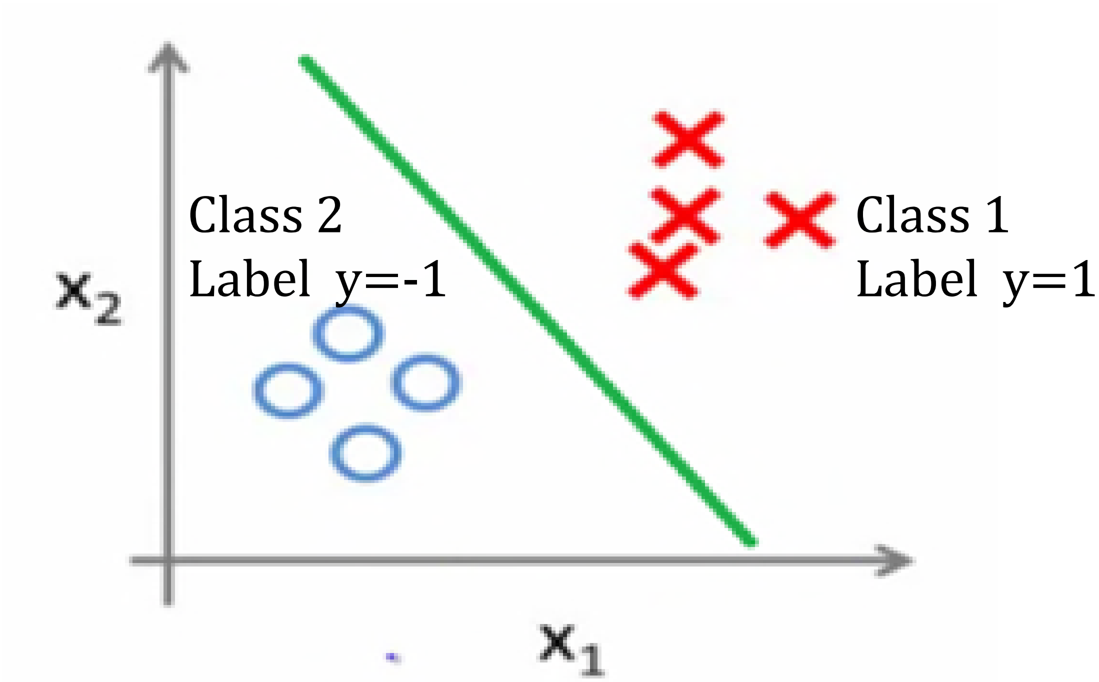
- **Regression vs Classification**
  - ❑ Regression aims to approximate a mapping function from input $\mathbf{x}$ to a continuous output y:

$$y = \mathbf{w}^T\mathbf{x} + b$$



❑ Classification aims to map input **x** to a discrete output y. It can also be interpreted as finding a decision boundary to separate samples in different classes:

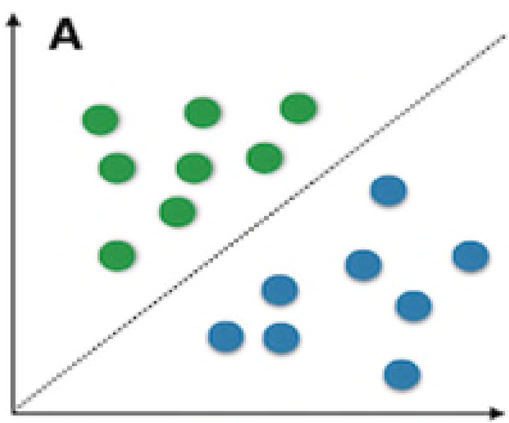
$$y = w_1 x_1 + w_2 x_2 + b = \mathbf{w}^T \mathbf{x} + b$$



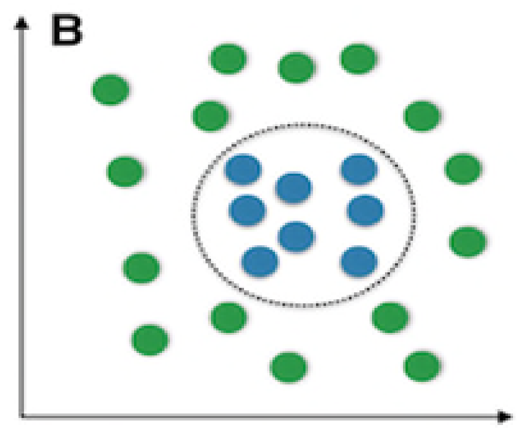
- **Linear vs Nonlinear (e.g. neural network) Classifier**

❑ Linear classifiers can only solve problems where samples from different classes are separable by a hyperplane.

❑ If a hyperplane is unable to separate, then a nonlinear classifier is needed. Neural networks can be used to build nonlinear classifiers.



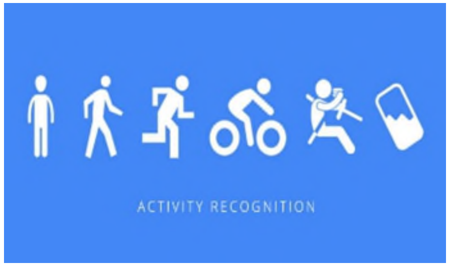
Linear classifier



Nonlinear classifier
(e.g. neural networks)

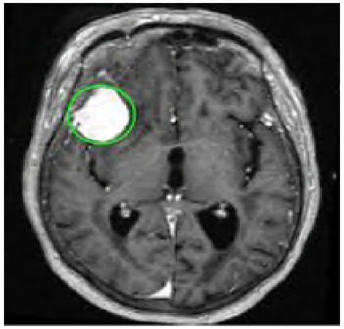
Examples of Classification Tasks

Activity Recognition

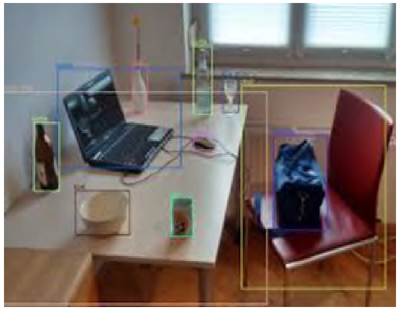


Sentiment Analysis

Brain Tumour Detection



Object Recognition

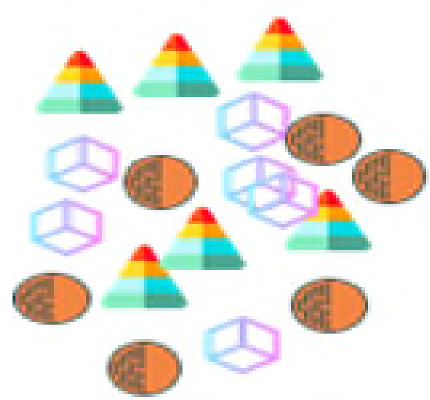


(2) Unsupervised Learning

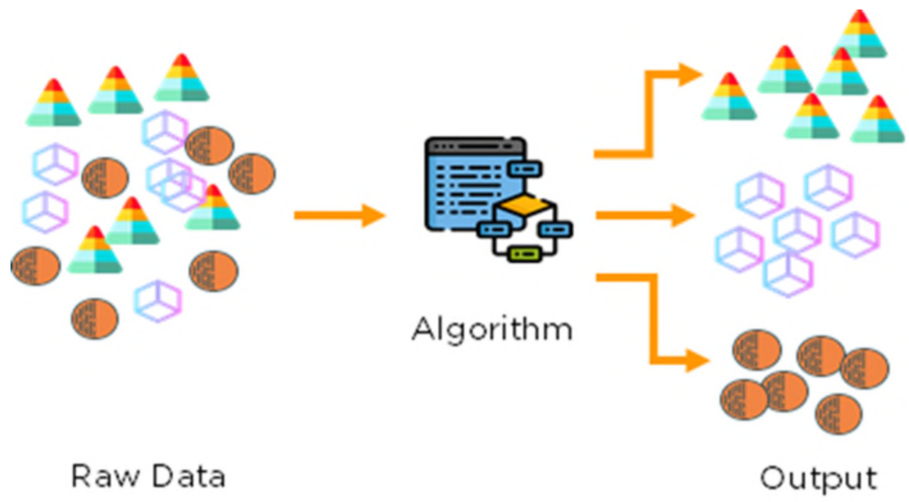
In contrast to supervised learning, *unsupervised learning* does not require a teacher, *i.e.* there is no target value.

The goal of unsupervised learning is to group data based on their intrinsic properties: the samples within each cluster are quite similar to each other, while samples in different clusters are quite different from each other.

For example, a set of objects (without knowing their names):

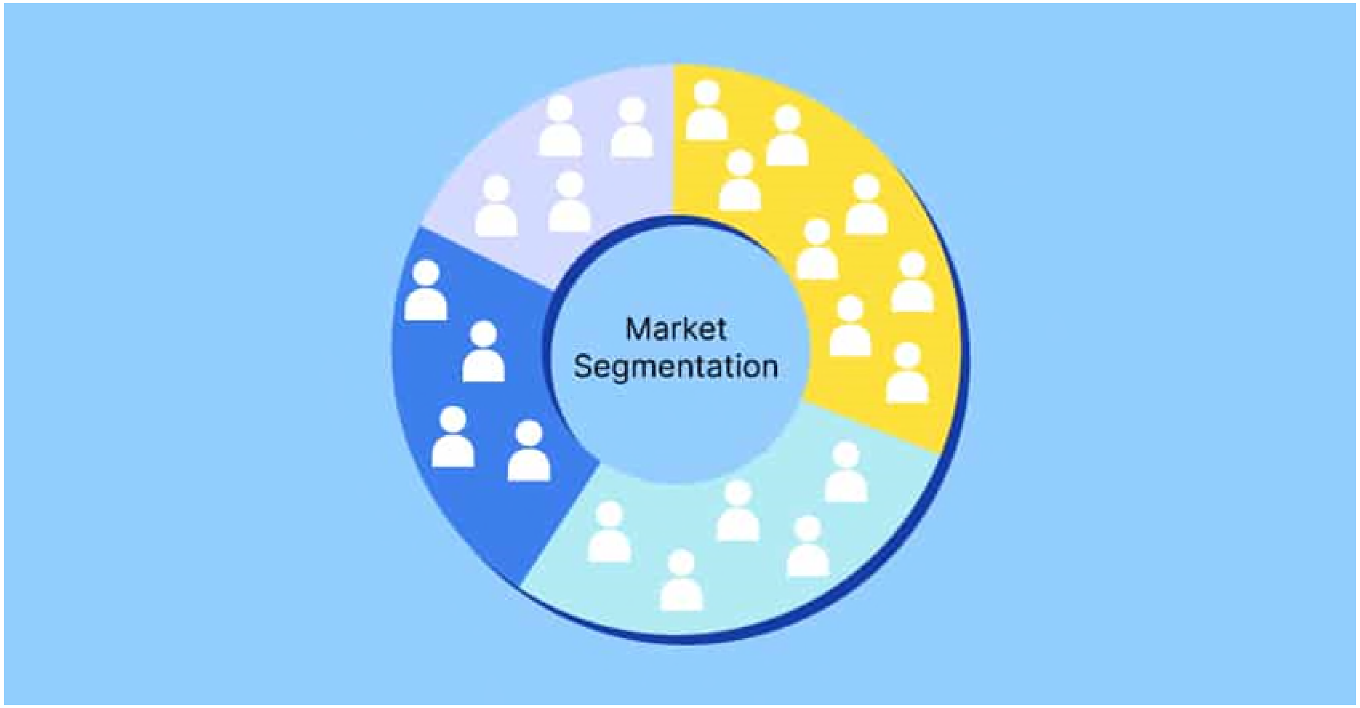


The unsupervised learning algorithm organizes the objects into three groups:



- **Application Examples of Unsupervised Learning**

❑ Market and customer segmentation based on needs, location, interests or demographics etc.



❑ Image compression

❑ Phenotype clustering in health care