Introduction
To
Neural Network

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling.

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

Machine Learning

Machine learning involves adaptive mechanisms that enable computers to learn from experience, learn by example and learn by analogy. Learning capabilities can improve the performance of an intelligent system over time. The most popular approaches to machine learning are artificial neural network.

Biological Inspiration

Animals are able to react adaptively to changes in their external and internal environment, and they use their nervous system to perform these behaviours.

An appropriate model/simulation of the nervous system should be able to produce similar responses and behaviours in artificial systems.

The nervous system is build by relatively simple units, the neurons, so copying their behaviour and functionality should be the solution.

Introduction

A neural network can be defined as a model of reasoning based on the human brain. The brain consists of a densely interconnected set of nerve cells, or basic information-processing units, called neurons.

The human brain incorporates nearly 10 billion neurons and 60 trillion connections, synapses, between them. By using multiple neurons simultaneously, the brain can perform its functions much faster than the fastest computers in existence today.

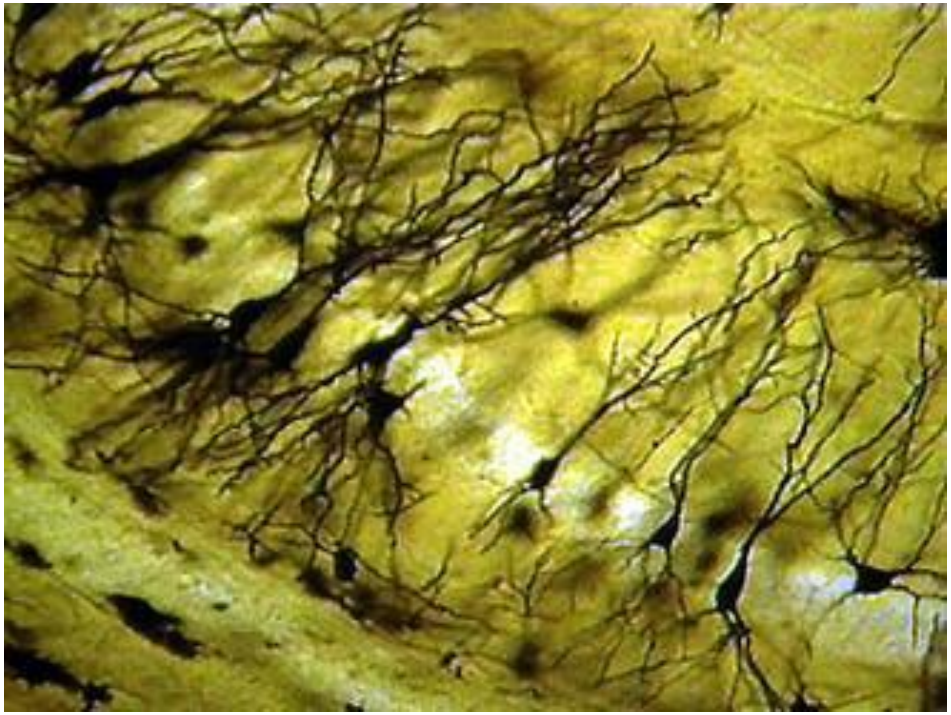
Introduction

Each neuron has a very simple structure, but an army of such elements constitutes a tremendous processing power.

A neuron consists of a cell body, soma, a number of fibers called dendrites, and a single long fiber called the axon.

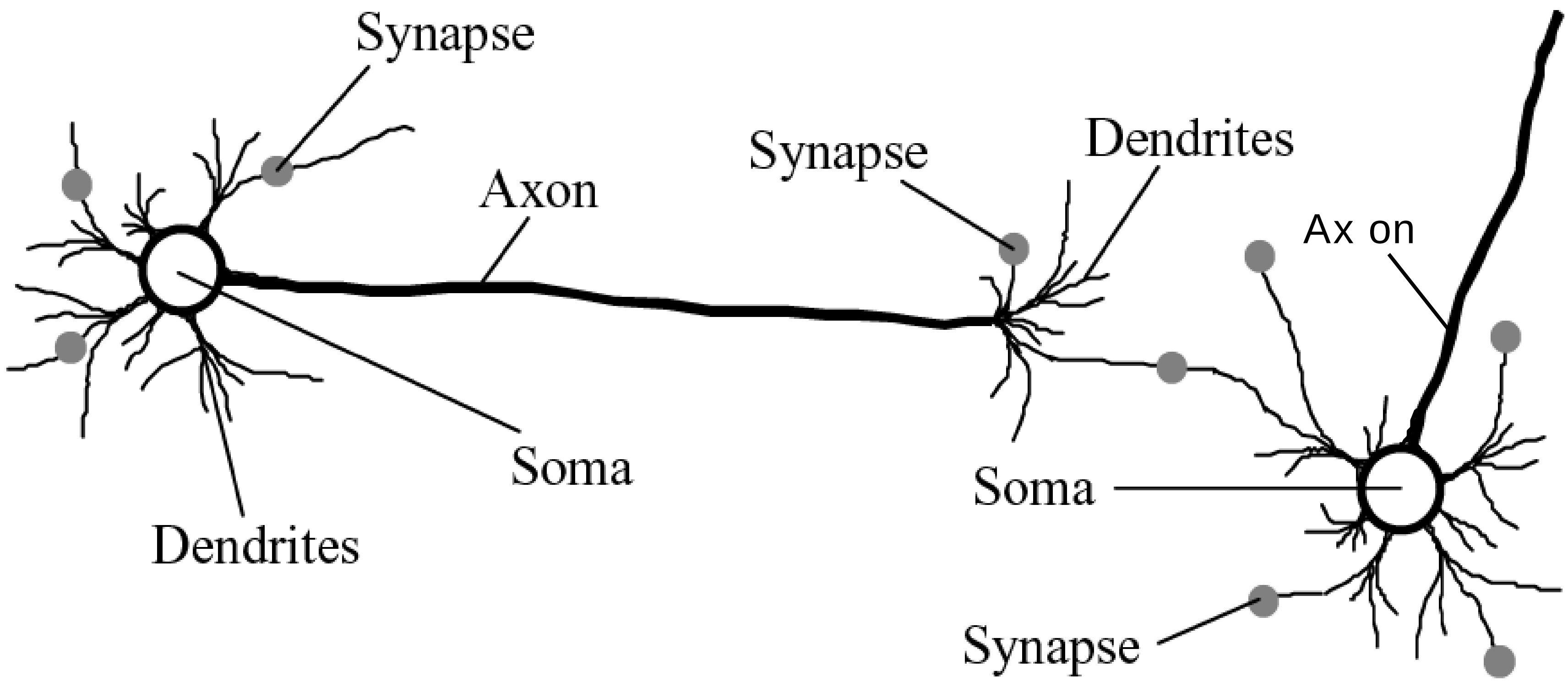
Introduction

Biological Neuron:



Introduction

Biological Neuron architecture:



Introduction

- The spikes travelling along the axon of the pre-synaptic neuron trigger the release of neurotransmitter substances at the synapse.
- The neurotransmitters cause excitation or inhibition in the dendrite of the post-synaptic neuron.
- The integration of the excitatory and inhibitory signals may produce spikes in the post-synaptic neuron.
- The contribution of the signals depends on the strength of the synaptic connection.

Artificial Neural Network

An artificial neural network consists of a number of very simple processors, also called neurons, which are analogous to the biological neurons in the brain.

The neurons are connected by weighted links passing signals from one neuron to another.

The output signal is transmitted through the neuron’s outgoing connection. The outgoing connection splits into a number of branches that transmit the same signal. The outgoing branches terminate at the incoming connections of other neurons in the network.

Artificial Neural Network

Comparison between brain and artificial neural networks:

	Brain	ANN
Speed	Few ms.	Few nano sec. massive ||el processing
Size and complexity	10^{11} neurons & 10^{15} interconnections	Depends on designer
Storage capacity	Stores information in its interconnection or in synapse. No Loss of memory	Contiguous memory locations loss of memory may happen sometimes.
Tolerance	Has fault tolerance	No fault tolerance Inf gets disrupted when interconnections are disconnected
Control mechanism	Complicated involves chemicals in biological neuron	Simpler in ANN

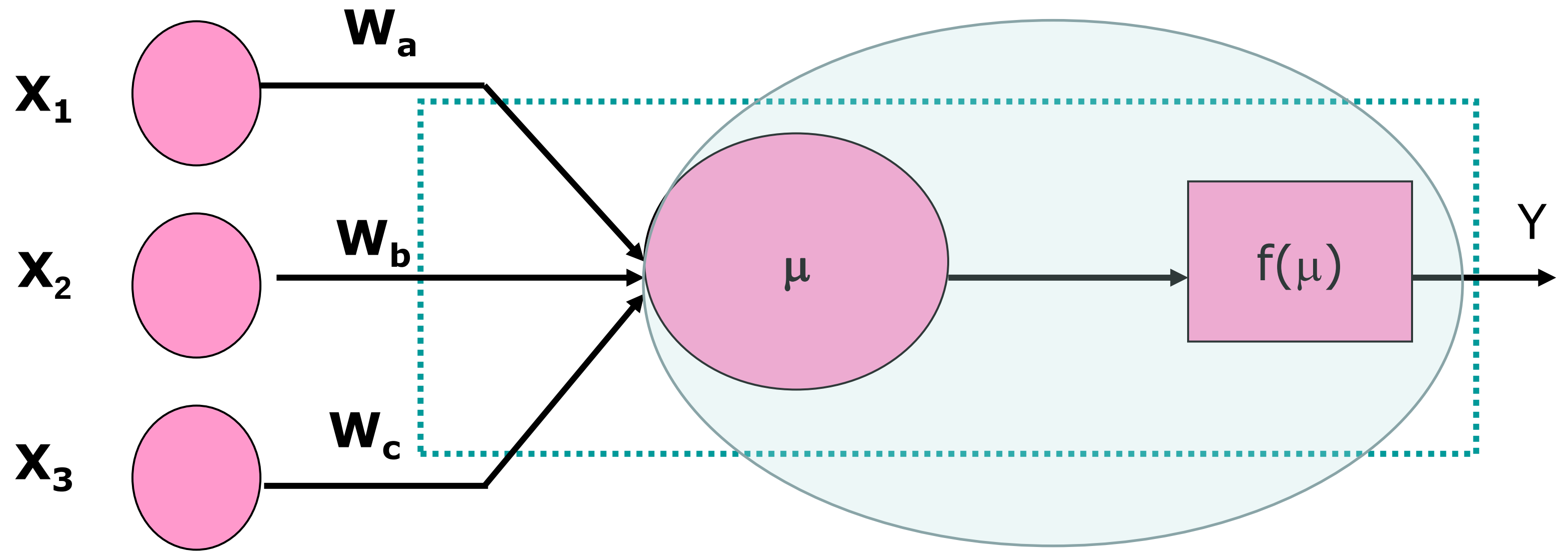
Artificial Neural Network

Analogy between biological and artificial neural networks:

Biological Neural Network	***Artificial Neural Network***
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

Artificial Neural Network

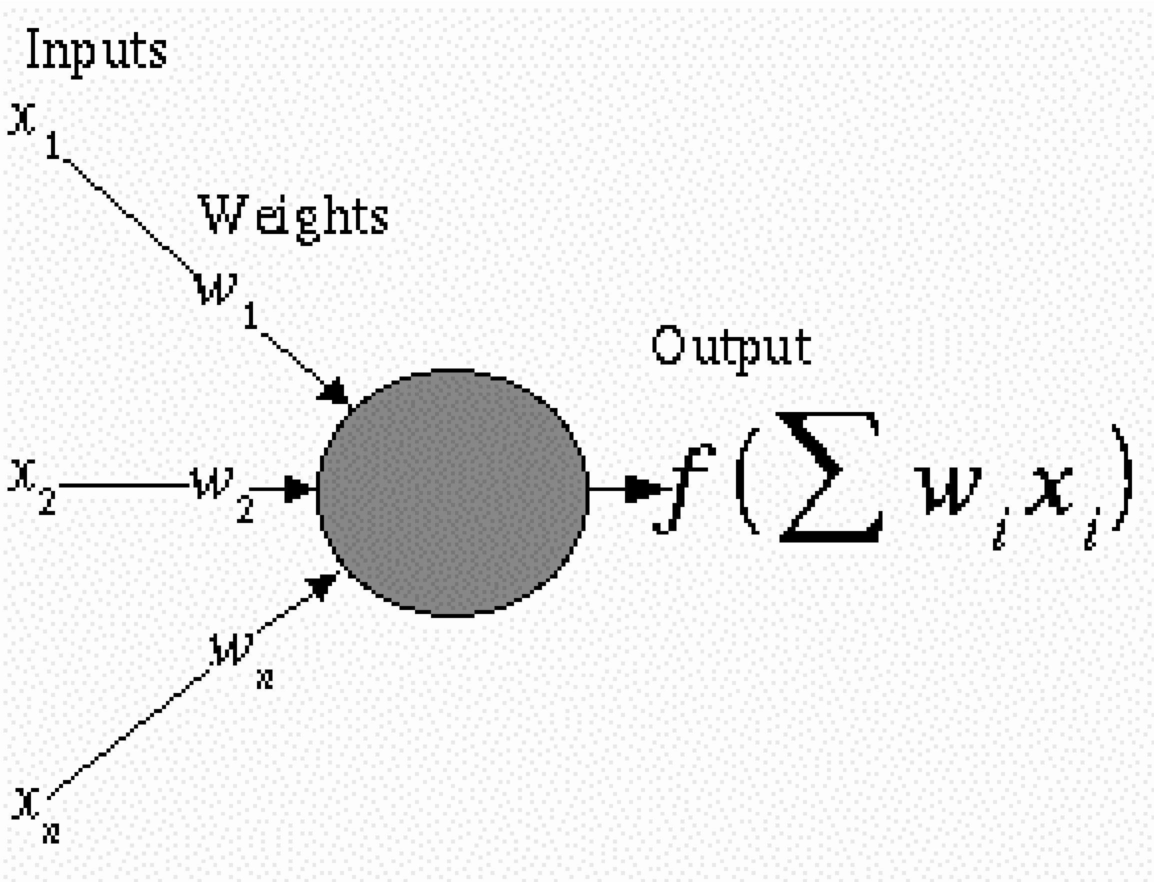
ANN vs Biological terms



Input units	Connection weights	Summing function	computation	
(dendrite)	(synapse)	(soma)		(axon)

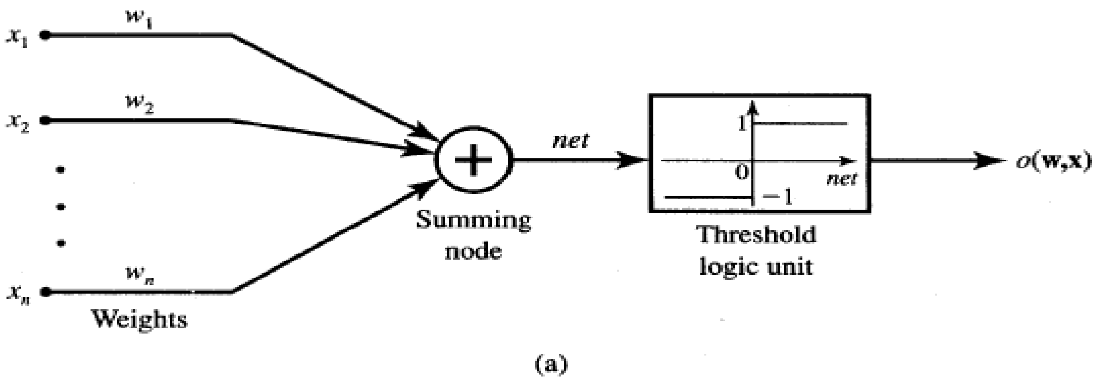
Artificial Neural Network

An ANN neuron architecture (perceptron):

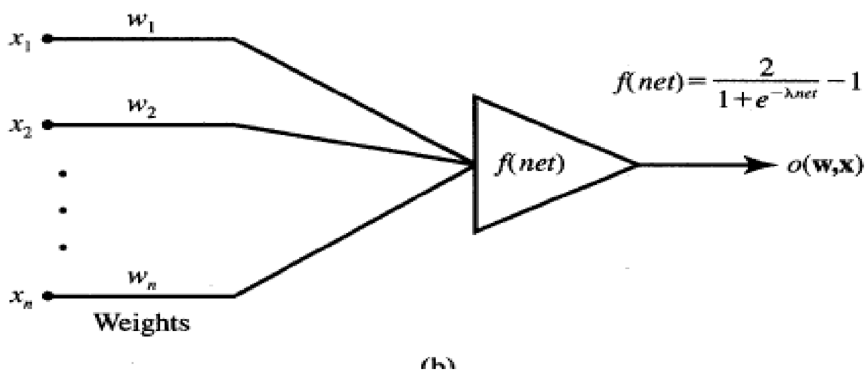


Artificial Neural Network

Common models of neurons:



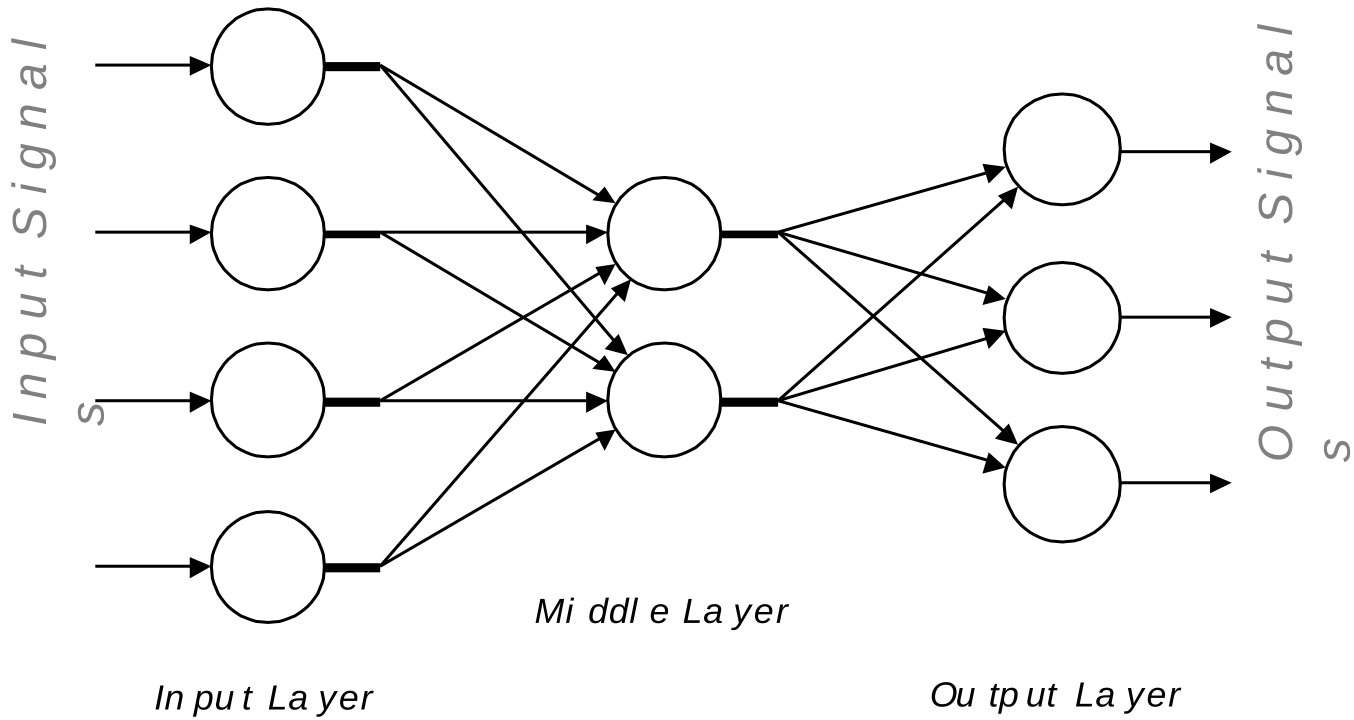
Binary perceptron's



Continuous perceptron's

Artificial Neural Network

Typical ANN architecture:

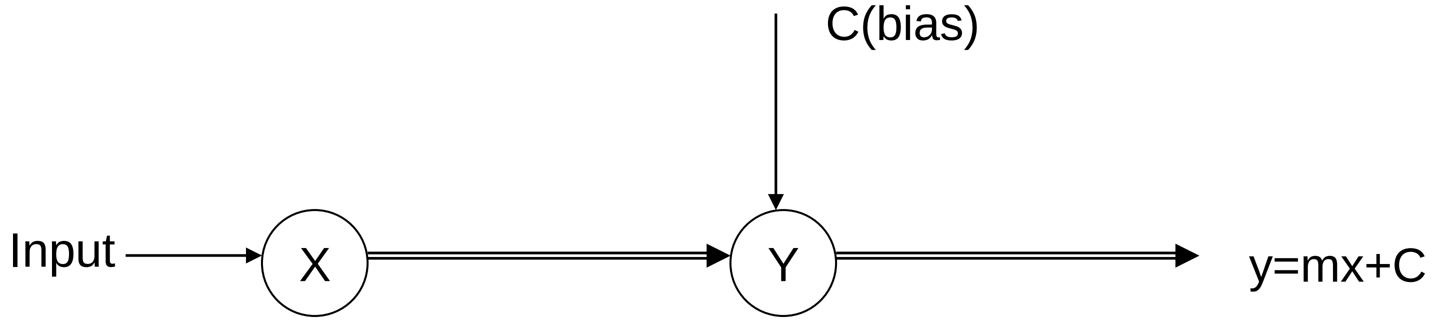


Some Important Terminologies in ANN

- Weights
  - Each neuron is connected to every other neuron by means of directed links
  - Links are associated with weights
  - The weight represent information being used by the net to solve a problem.

Some Important Terminologies in ANN

- Bias
  - Bias is like another weight.
  - Positive bias: increase the net input
  - Negative bias: decrease the net input



Some Important Terminologies in ANN

- Threshold
  - Set value based upon which the final output of the network may be calculated
  - Used in activation function

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

- Activation Function
  - Used to calculate the output response of a neuron.
  - Sum of the weighted input signal is applied with an activation to obtain the response.

Some Important Terminologies in ANN

- Learning rate
  - Denoted by α.
  - Used to control the amount of weight adjustment at each step of training
  - Learning rate ranging from 0 to 1 determines the rate of learning in each time step

- Momentum factor
  - Used for convergence when momentum factor is added to weight updating process.

ANN Components

1. Interconnection
2. Learning Rules
3. Activation Functions

ANN Components

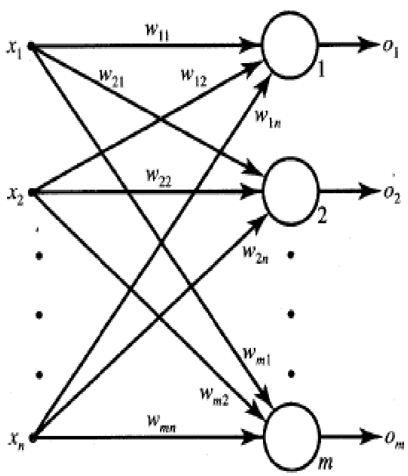
1. Interconnection:
   a) Feed Forward
   b) Feed Back
   c) Recurrent

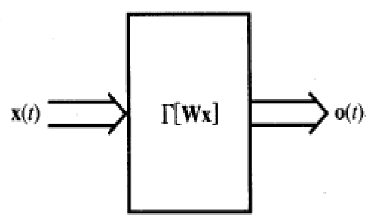
ANN Components

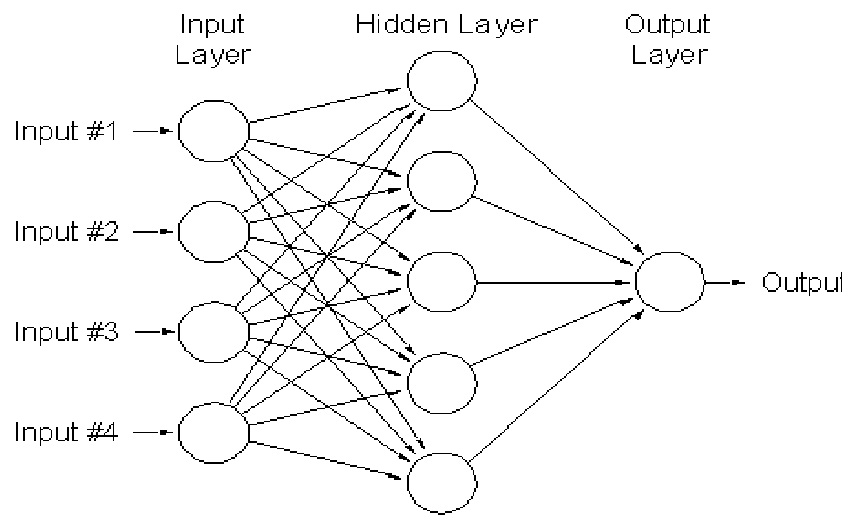
1. Interconnection:
   a) Feed Forward

Multilayer Feed Forward

Single Layer Feed Forward







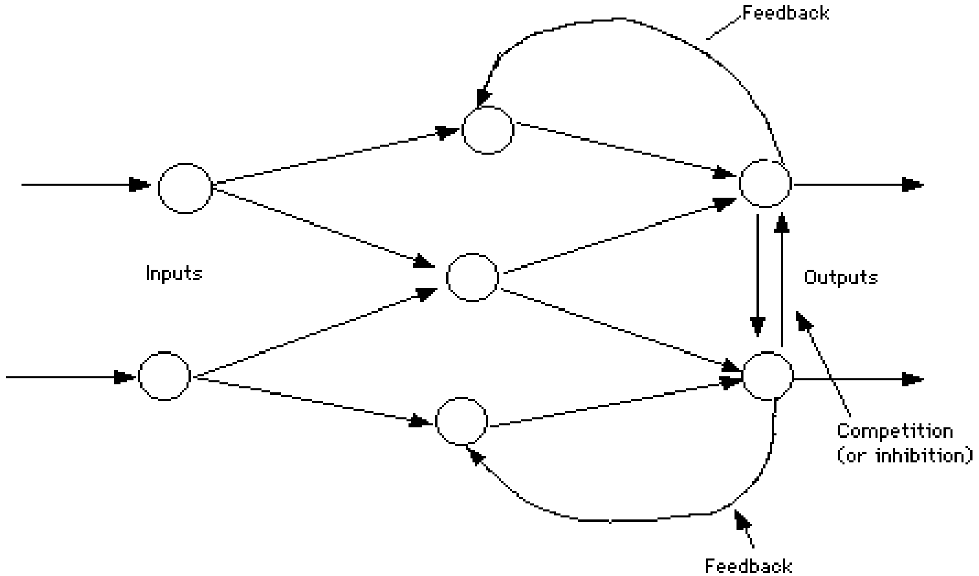
Can be used to solve complicated problems

ANN Components

1. Interconnection:

   b) Feed Back

   When outputs are directed back as inputs to same or preceding layer nodes it results in the formation of feedback networks

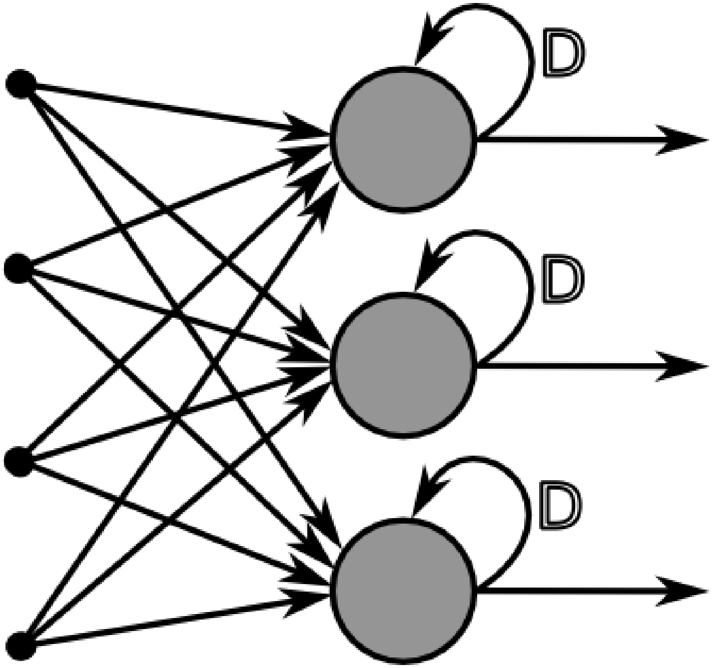


ANN Components

1. Interconnection:

    c) Recurrent

    Feedback networks with closed loop



ANN Components

2. Learning Rules

It’s a process by which a NN adapts itself to a stimulus by making proper parameter adjustments, resulting in the production of desired response

Training in learning

The process of **modifying the weights** in the connections between network layers with the objective of achieving the expected output is called training a network.

ANN Components

2. Learning Rules

Classification of learning:

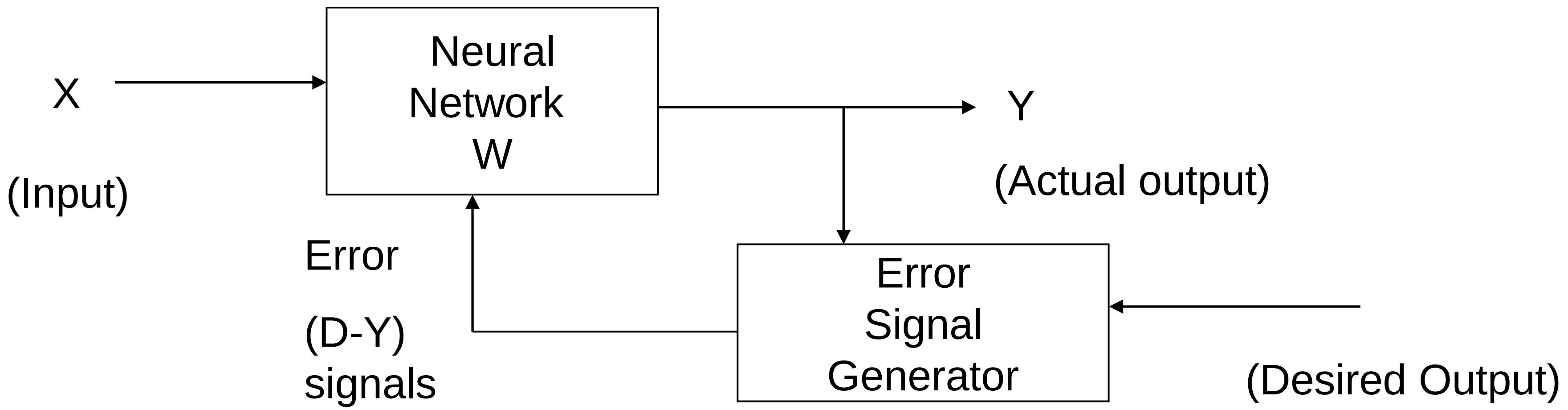
a) Supervised learning
b) Unsupervised learning
c) Semi Supervised learning
d) Reinforcement learning

ANN Components

2. Learning Rules

a) Supervised learning

- Child learns from a teacher
- Each input vector requires a corresponding target vector.
- Training pair=[input vector, target vector]

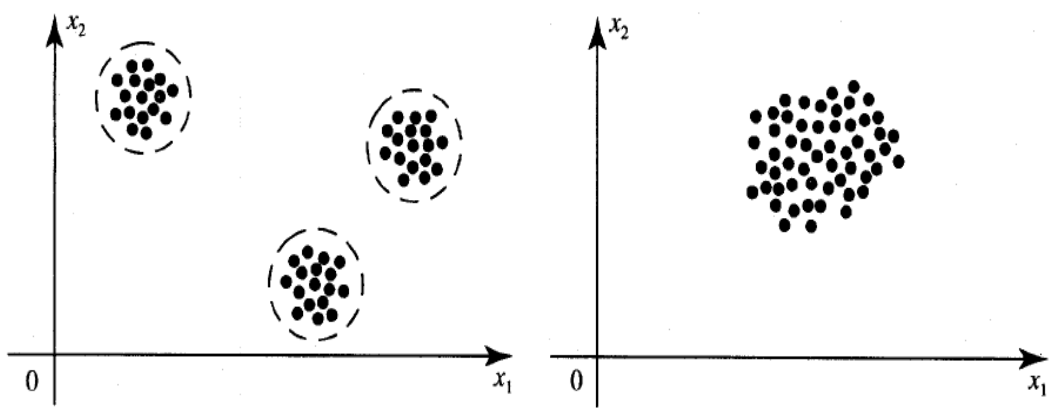


ANN Components

2. Learning Rules

b) Unsupervised learning

- All similar input patterns are grouped together as clusters.
- If a matching input pattern is not found a new cluster is formed
- Self Organizing



ANN Components

2. Learning Rules
c) Semi supervised learning
- Combination of Supervised and Unsupervised learning

ANN Components

2. Learning Rules

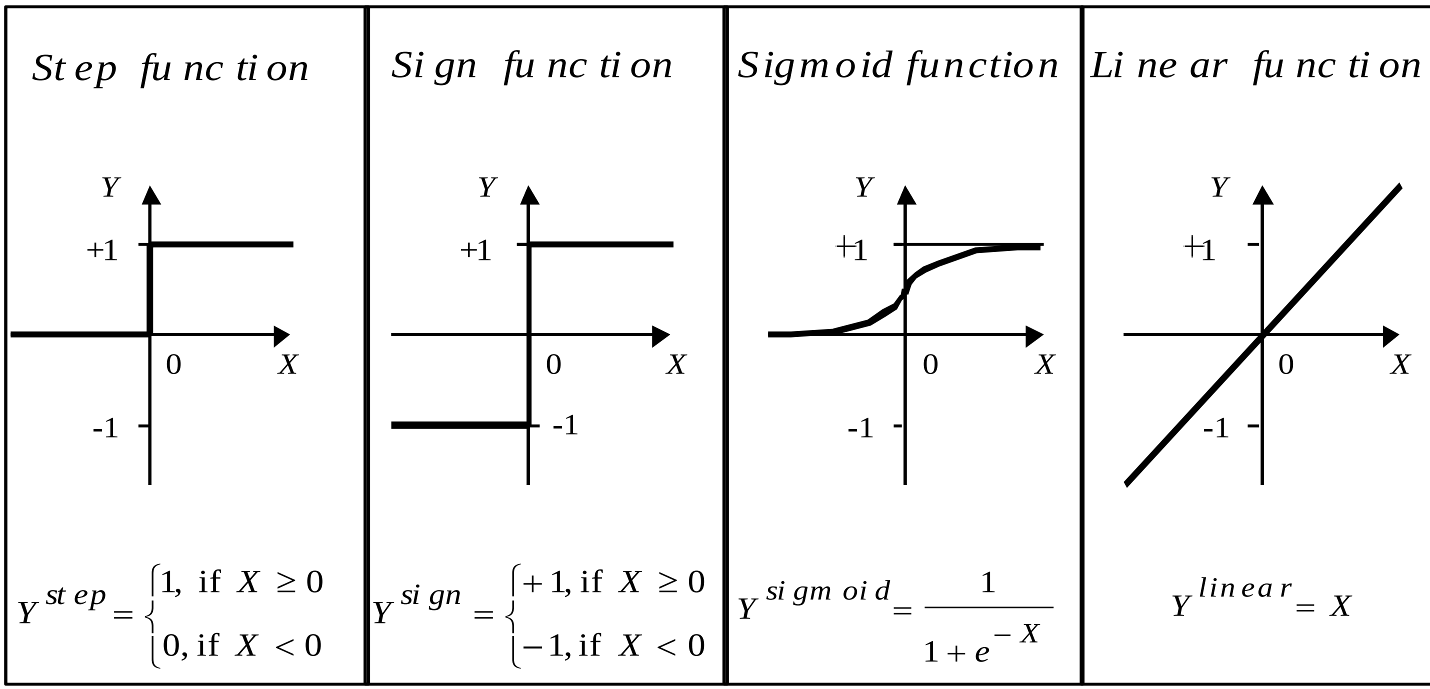
d) Reinforcement learning

- Trial and error, no teacher, but can asses the situations – reinforcement signals.
- If less information is available about the target output values (critic information)
- Learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal

ANN Components

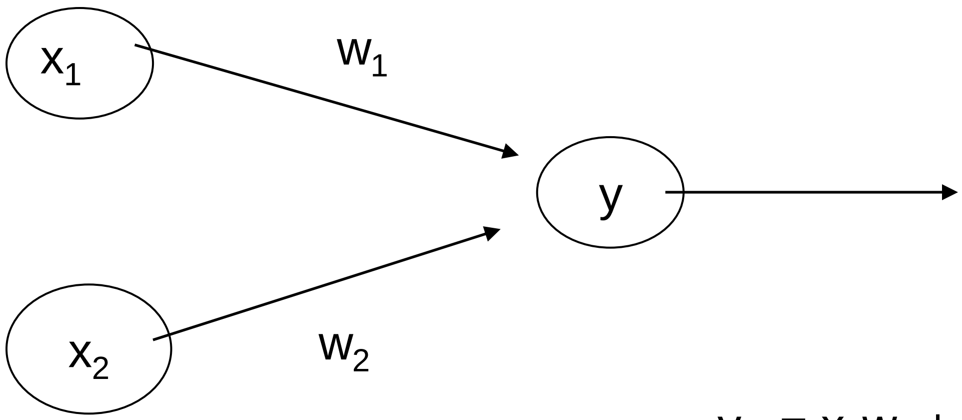
3. Activation Functions

To determine the neuron's output:



ANN Components

3. Activation Functions



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

Activation is $f(y_{in})$

8 Steps ANN Application Development

Step 1: (Data collection) The data to be used for the training and testing of the network are collected. Important considerations are that the particular problem is amenable to neural network solution and that adequate data exist and can be obtained.

Step 2: (Training and testing data separation) Training data must be identified, and a plan must be made for testing the performance of the network. The available data are divided into **training** and **testing** data sets. For a moderately sized data set, normally 80% of the data are randomly selected for training, 10% for testing, and 10% secondary testing (validation).

8 Steps ANN Application Development

Data is divided into two parts

- Train data – to train the network (the network learned from past experience/data).
- Test data – to validate the learned network.

8 Steps ANN Application Development

Step 3: (Network architecture) A network architecture and a learning method are selected. Important considerations are the exact number of perceptron's and the number of layers.

Step 4: (Parameter tuning and weight initialization) There are parameters for tuning the network to the desired learning performance level. Part of this step is initialization of the network weights and parameters, followed by modification of the parameters as training performance feedback is received.

- Often, the initial values are important in determining the effectiveness and length of training.

8 Steps ANN Application Development

Step 5: (Data transformation) Transforms the application data into the type and format required by the ANN.

Step 6: (Training) Training is conducted iteratively by presenting input and desired or known output data to the ANN. The ANN computes the outputs and adjusts the weights until the computed outputs are within an acceptable tolerance of the known outputs for the input cases.

8 Steps ANN Application Development

Step 7: (Testing) Once the training has been completed, it is necessary to test the network.

- The **testing** examines the performance of the network using the derived weights by measuring the ability of the network to classify the testing data correctly. The testing result is the result to be considered and normally to be compared with other conventional methods (statistical etc..). An accuracy of about 80% is usually acceptable for ANN applications. At this level, a system is useful because it automatically identifies problem situations for further analysis by a human expert.

8 Steps ANN Application Development

- **Black-box testing** (comparing test results to historical results) is the primary approach for verifying that inputs produce the appropriate outputs.

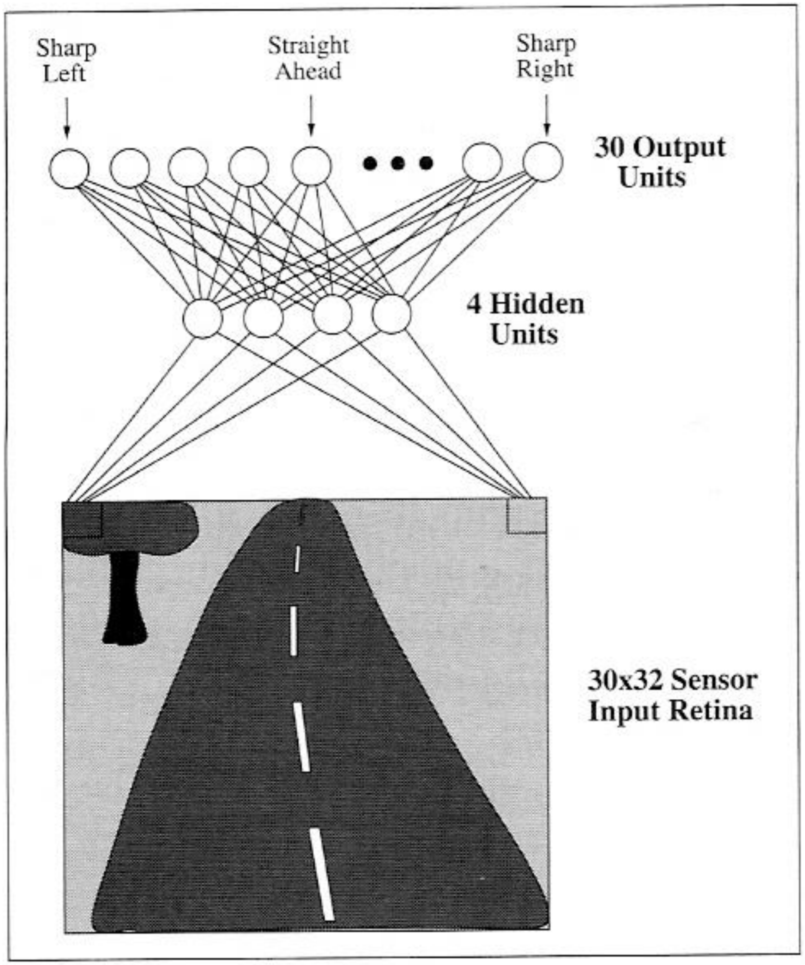
Step 8: (Implementation) Now a stable set of weights are obtained.

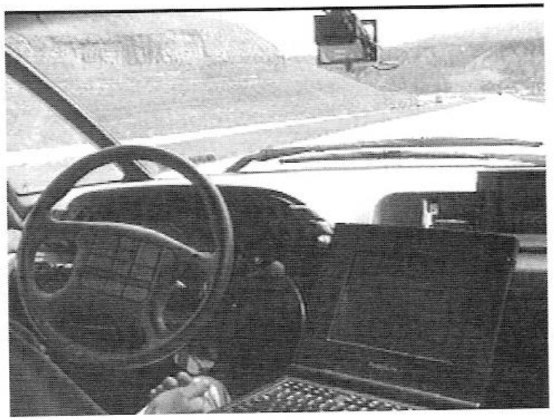
- Now the network can reproduce the desired output given inputs like those in the training set.
- The network is ready to use as a stand-alone system or as part of another software system where new input data will be presented to it and its output will be a recommended decision.

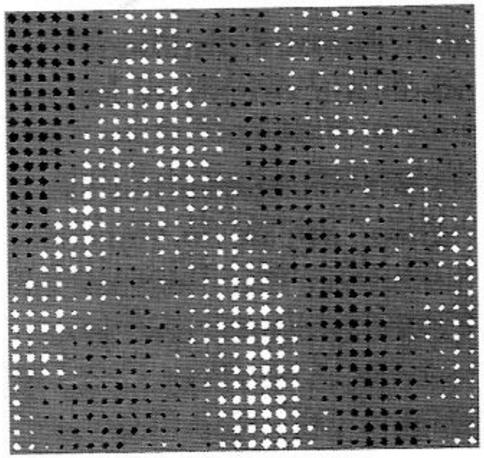
ANN Application

- ALVINN uses a learned ANN to steer an autonomous vehicle driving at normal speeds on public highways
  - Input to network: 30x32 grid of pixel intensities obtained from a forward-pointed camera mounted on the vehicle
  - Output: direction in which the vehicle is steered
  - Trained to mimic observed steering commands of a human driving the vehicle for approximately 5 minutes

ANN Application







ANN Application

ANN learning well-suit to problems which the training data corresponds to noisy, complex data (inputs from cameras or microphones)

Can also be used for problems with symbolic representations

ANN Application

Most appropriate for problems where:

- Input is high-dimensional discrete or real-valued (e.g. raw sensor input)
- Output is discrete or real valued
- Output is a vector of values
- Possibly noisy data
- Long training times accepted
- Fast evaluation of the learned function required.
- Not important for humans to understand the weights

ANN Application

Types of Problems:

- Mathematical Modeling (Function Approximation)

  Modeling – often mathematical relationship between two sets of data unknown analytically

- Classification

  Assignment of objects to specific class

- Clustering

  Grouping together objects similar to one another

ANN Application

- Forecasting

  Prediction of future events based on history

- Vector Quantization

  Object space divided into several connected regions. Objects classified based on proximity to regions

- Pattern Association

  Auto-associative systems useful when incoming data is a corrupted version of actual object e.g. face, handwriting

ANN Application

- Control

  Manufacturing, Robotic and Industrial machines have complex relationships between input and output variables

- Optimization

  Requirement to improve system performance or costs subject to constraints.

ANN Application

Examples:

- Speech phoneme recognition
- Image classification
- Financial prediction
- Tax form processing to identify tax fraud
- Bankruptcy prediction
- Customer credit scoring
- Loan approvals
- Energy forecasting
- Computer access security (intrusion detection and classification of attacks)
- And many more….

ANN Advantages

1. *Usefulness for pattern recognition, classification, generalization, abstraction and interpretation of incomplete and noisy inputs.* (e.g. handwriting recognition, image recognition, voice and speech recognition, weather forecasting).
2. *Providing some human characteristics to problem solving* that are difficult to simulate using the logical, analytical techniques of expert systems and standard software technologies. (e.g. financial applications).
3. *Ability to solve new kinds of problems*. ANNs are particularly effective at solving problems whose solutions are difficult, if not impossible, to define. This opened up a new range of decision support applications formerly either difficult or impossible to computerize.

ANN Advantages

4. *Robustness*. ANNs tend to be more robust than their conventional counterparts. They have the ability to cope with incomplete or fuzzy data. ANNs can be very tolerant of faults if properly implemented.
5. *Fast processing speed*. Because they consist of a large number of massively interconnected processing units, all operating in parallel on the same problem, ANNs can potentially operate at considerable speed (when implemented on parallel processors).
6. *Flexibility and ease of maintenance*. ANNs are very flexible in adapting their behavior to new and changing environments. They are also easier to maintain, with some having the ability to learn from experience to improve their own performance.

ANN Limitations

1. ANNs do not produce an *explicit model* even though new cases can be fed into it and new results obtained.

2. *ANNs lack explanation capabilities*. Justifications for results is difficult to obtain because the connection weights usually do not have obvious interpretations.

ANN Evolution:

1. 1st Generation – McCulloch-Pitts neuron (also known as a perceptron or a threshold-gate)

2. 2nd Generation – Neurons with continuous activation function (Example: MLP-BP)

3. 3rd Generation – Raise the level of biological realism by using individual spikes.