Artificial Neural Network - Chapter 4

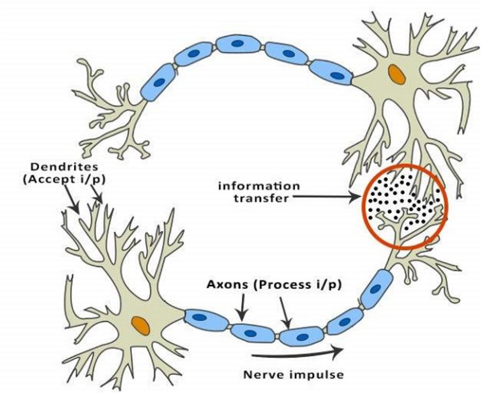
Artificial Neural Network

The building blocks of neural networks are the neurons.
In technical systems, we also refer to them as units or nodes.

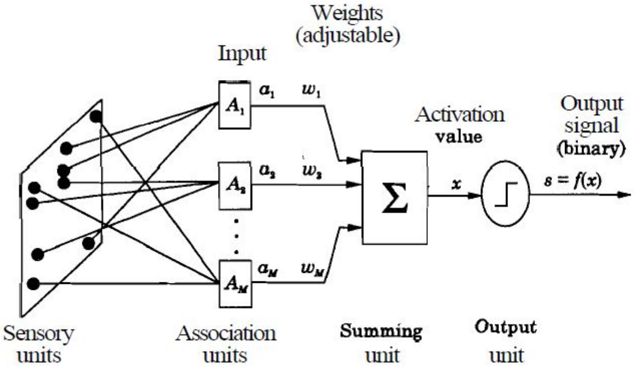
Basically, each neuron

- receives input from many other neurons.
- changes its internal state (activation) based on the current input.
- sends one output signal to many other neurons, possibly including its input neurons (recurrent network).

Biological Neuron



Block diagram of artificial neuron - perceptron



Fundamentals of a neuron

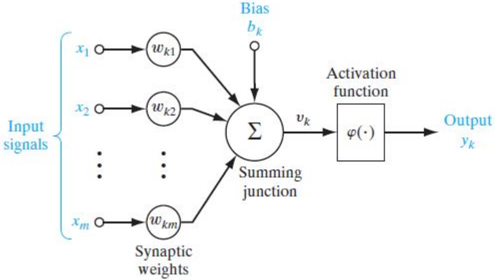
A neuron is an information-processing unit that is fundamental to the operation of a neural network.

The three basic elements of the neural model are:

- A set of synapses, or connecting links, each of which is characterized by a weight or strength of its own.
- An adder for summing the input signals, weighted by the respective synaptic strengths of the neuron; the operations described here constitute a linear combiner.
- An activation function for limiting the amplitude of the output of a neuron.

Neuron

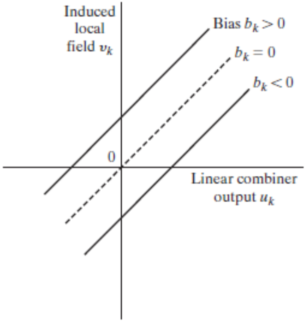
The neural model also includes an externally applied bias, denoted by b_k. The bias b_k has the effect of increasing or lowering the net input of the activation function,depending on whether it is positive or negative, respectively.



The perceptron is the simplest form of a neural network used for the classification of patterns said to be linearly separable. It consists of a single neuron with adjustable synaptic weights and bias.

Effect of adding bias

Affine transformation produced by the presence of a bias; note that $v_k = b_k$ at $u_k = 0$.



$$v_k = u_k + b_k$$

and

$$y_k = \varphi(v_k)$$

Two types of activation function

1. *Threshold Function.*

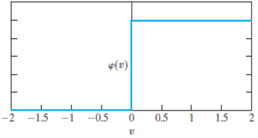
$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

Correspondingly, the output of neuron k employing such a threshold function is expressed as

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases}$$

where v_k is the induced local field of the neuron; that is,

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$$

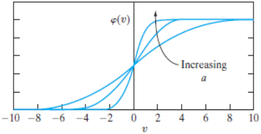


(a) Threshold function.

2. *Sigmoid Function.*

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

where a is the *slope parameter* of the sigmoid function. By varying the parameter a, we obtain sigmoid functions of different slopes.



(b) Sigmoid function for varying slope parameter a.

Mathematical representation of neuron

Mathematically neuron 'k' is represented as:
$u_k = \sum_{j=1}^{m} w_{kj} x_j$
Where:
$x_1, x_2 ... x_m$ are input signals.
$w_{k1}, w_{k2}, .. w_{km}$ are respective synaptic weights of neuron.

$v_k = u_k + b_k$

and output
$y_k = \phi(u_k + b_k)$
Where:
u_k is the linear combiner output due to the input signals
b_k is the bias.
$\phi(.)$ is the activation function
y_k is the output signal of the neuron.

Neural Network

What is a neural Network?

A neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest.

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.

Resemblance with brain

It resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process.
- Inter-neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The procedure used to perform the learning process is called a learning algorithm

Properties and capabilities of neural network

- Nonlinearity: An artificial neuron can be linear or nonlinear.
- InputOutput Mapping: The network learns from the examples by constructing an inputoutput mapping for the problem at hand.
- Adaptivity: Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment.
- Evidential Response: In the context of pattern classification, a neural network can be designed to provide information about the confidence in the decision made.
- Contextual Information: Every neuron in the network is potentially affected by the global activity of all other neurons in the network and hence contextual information is handled.
- Fault Tolerance: A neural network, implemented in hardware form, has the potential to be inherently fault tolerant.

Terminologies

An artificial neural network (ANN) is a highly simplified model of the structure of the biological neural network.

1. Processing unit: The processing unit consists of a summing part followed by an output part.
2. Interconnections: Several processing units are interconnected according to some topology to accomplish a pattern recognition task.
3. Operations: In operation, each unit of an ANN receives inputs from other connected units and/or from an external source.
4. activation value: The weighted sum computed by the summing part.
5. activation state The set of the N activation values of the network
6. output state The set of the N output values of the network

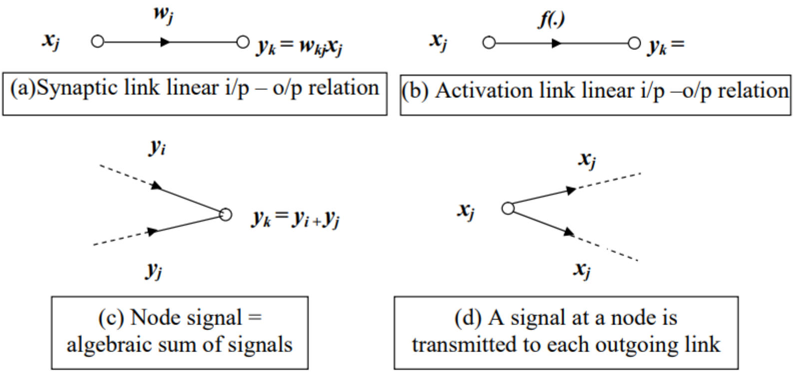
Neural Networks viewed as a Directed Graph

A signal-flow graph is a network of directed links (branches) that are interconnected at certain points called nodes.
Two different types of links may be distinguished.

- Synaptic links, whose behavior is governed by a linear inputoutput relation
- Activation links, whose behavior is governed in general by a nonlinear inputoutput relation.

1. A signal flows along a link only in the direction defined by the arrow on the link.
2. A signal flows along a link only in the direction defined by the arrow on the link.
3. The signal at a node is transmitted to each outgoing link originating from that node, with the transmission being entirely independent of the transfer functions of the outgoing links.

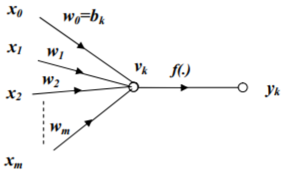
Rules



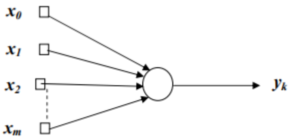
Signal – flow graph component

Rules

Using these rules we may construct the signal – flow diagram

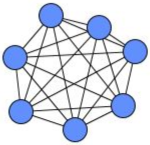


signal – flow graph of a neuron

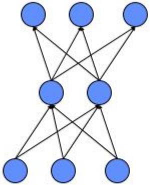


Architectural graph

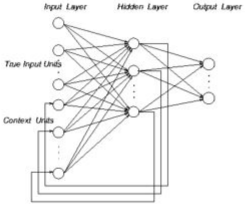
Topologies



completely connected



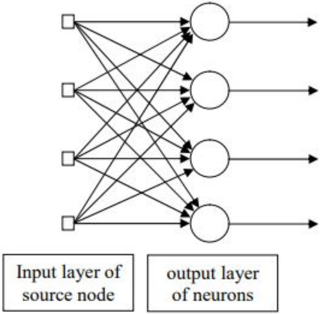
feedforward (directed, a-cyclic)



recurrent (feedback connections)

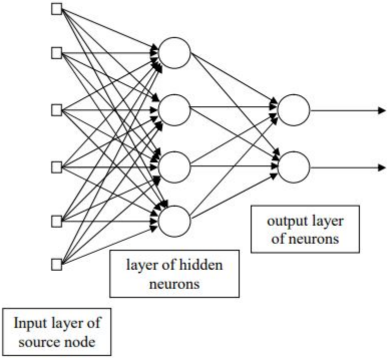
Network Architecture

Single layer feedforward networks: In a layered neural network, the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects directly onto an output layer of neurons, (computation nodes), but not vice versa. In other words, this network of a feedforward type.

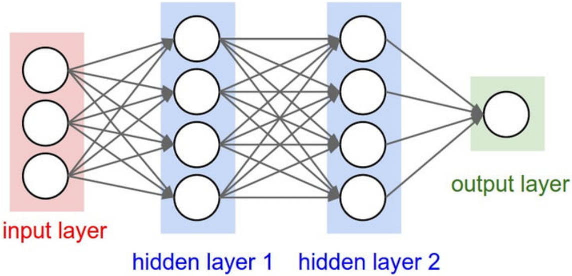


Network Architecture

Multi-layer feedforward networks: The source nodes in the input layer of the network constitute the input signals applied to the neurons in the 2nd. Layer (i.e., the first hidden layer). The output signals of the second layer are used as input to the 3rd. layer, and so on for the rest of the network.

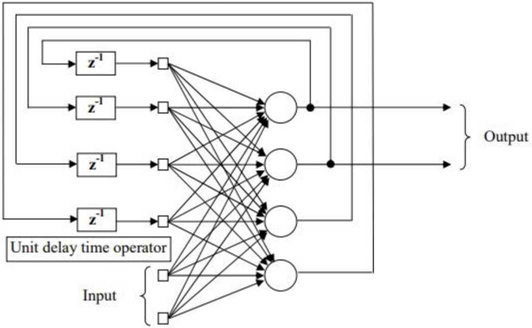


Multi-layer feed-forward network



Network Architecture

Recurrent Networks: A recurrent network distinguishes it self from a feedforward neural network in that it has at least one feedback loop. The feedback connections shown in the figure originate from the hidden neurons as well as from the output

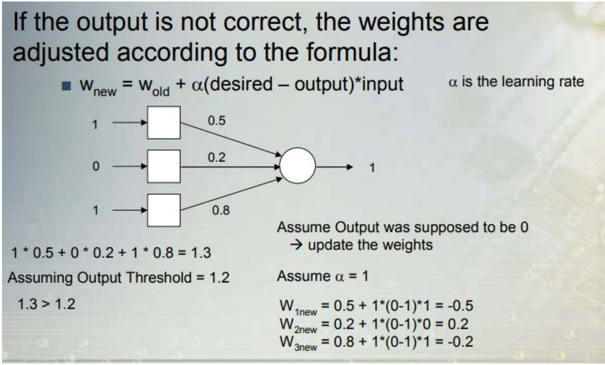


neurons.

Weight Adjustment

After each iteration, weights should be adjusted to minimize the error.

- All possible weights could be adjusted
- Back propagation as shown below



Back propagation

- Back-propagation is an example of supervised learning.
- It is used at each layer to minimize the error between the layers response and the actual data.
- The error at each hidden layer is an average of the evaluated error.
- Hidden layer networks are trained this way

Back propagation Procedure

- N is a neuron.
- N_w is one of N's inputs weights.
- N_{out} is N's output.
- $N_w = N_w + \Delta N_w$
- $\Delta N_w = N_{out} * (1 - N_{out}) * N_{ErrorFactor}$
- $N_{ErrorFactor} = N_{ExpectedOutput} - N_{ActualOutput}$

An example is available at https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Backpropagation Rule - Derivation

In stochastic gradient descent, for each training example 'd' descending the gradient of the error E_d every weight w_{ji} is updated by adding to it $\Delta \mathrm{w}_{ji}$.

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

$$E_d\left(\vec{w}\right) \equiv \frac{1}{2} \sum_{k \in outputs} \left(t_k - o_k\right)^2$$

Here outputs is the set of output units in the network, t_k is the target value of unit k for training example d, and o_k is the output of unit k given training example d.

Representational Power of Feedforward Networks

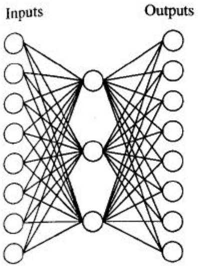
1. Boolean functions: Every boolean function can be represented exactly by some network with two layers of units, although the number of hidden units required grows exponentially in the worst case with the number of network inputs
2. Continuous functions: Every bounded continuous function can be approximated with arbitrarily small error (under a finite norm) by a network with two layers of units
3. Arbitrary functions: Any function can be approximated to arbitrary accuracy by a network with three layers of units

The hypothesis space is the n-dimensional Euclidean space of the n network weights and it is continuous

Hidden Layer Representation

Consider, the network shown below. Here, the eight network inputs are connected to three hidden units, which are in turn connected to the eight output units.

How to train the machine to learn the simple target function f $(\vec{x}) = \vec{x}$, where $\vec{x}$ is a vector containing seven 0's and a single 1?.



Input			Hidden Values			Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.15	.99	.99	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.01	.11	.88	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

Design Choices

1. Input Encoding
2. Output Encoding
3. Network Graph Structure
4. Other learning algorithm parameters