

Artificial Neural Network : Architectures

Credits:

Soft Computing Applications, Dr. Debasis Samanta, IIT Kharagpur

RC Chakraborty, www.myreaders.info

“Principles of Soft Computing” by S.N. Sivanandam & SN Deepa

Neural Networks, Fuzzy Logic and Genetic Algorithms by S. Rajasekaran and GAV Pai

Neural networks- a comprehensive foundation by Simon S Haykin

Neural network architectures

There are three fundamental classes of ANN architectures:

- Single layer feed forward architecture
- Multilayer feed forward architecture
- Recurrent networks architecture

Before going to discuss all these architectures, we first discuss the mathematical details of a neuron at a single level. To do this, let us first consider the AND problem and its possible solution with neural network.

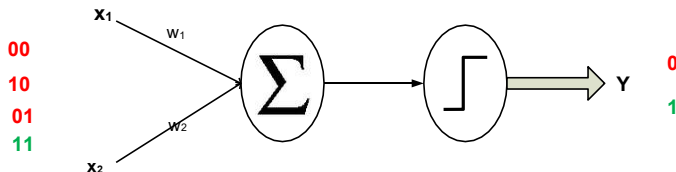
The AND problem and its Neural network

- The simple Boolean AND operation with two input variables x_1 and x_2 is shown in the truth table.
- Here, we have four input patterns: 00, 01, 10 and 11.
- For the first three patterns output is 0 and for the last pattern output is 1.

Inputs x_1 x_2		Output (y)
0	0	0
0	1	0
1	0	0
1	1	1

The AND problem and its neural network

- Alternatively, the AND problem can be thought as a perception problem where we have to receive four different patterns as input and perceive the results as 0 or 1.

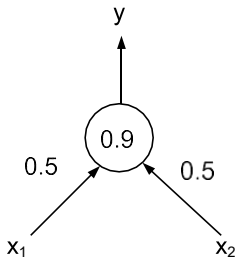


The AND problem and its neural network

- A possible neuron specification to solve the AND problem is given in the following. In this solution, when the input is 11, the weight sum exceeds the threshold ($\theta = 0.9$) leading to the output 1 else it gives the output 0.

Inputs		Output (y)
x_1	x_2	
0	0	0
0	1	0
1	0	0
1	1	1

The AND Logic

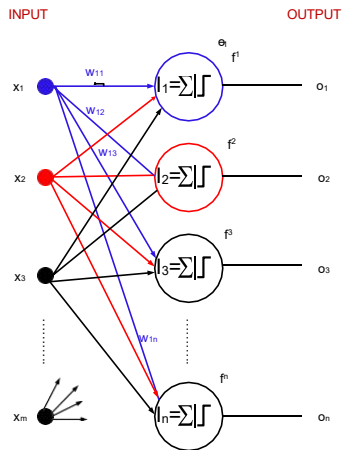


A single neuron

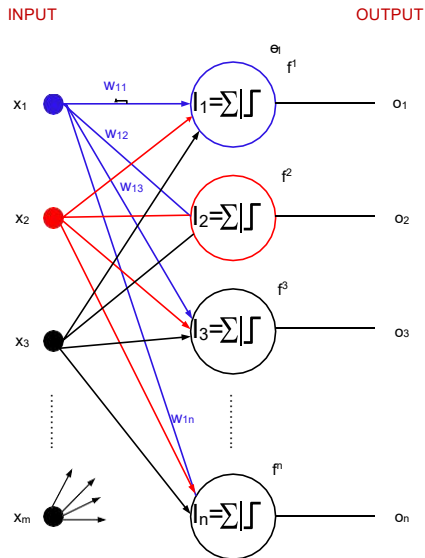
$$w_1 = 0.5, w_2 = 0.5 \text{ and } \theta = 0.9$$

Single layer feed forward neural network

- The concept of the AND problem and its solution with a single neuron can be extended to multiple neurons.



Single layer feed forward neural network



Single layer feed forward neural network

- We see, a layer of n neurons constitutes a single layer feed forward neural network.
- This is so called because, it contains a single layer of artificial neurons.
- Note that the input layer and output layer, which receive input signals and transmit output signals are although called layers, they are actually boundary of the architecture and hence truly not layers.
- The only layer in the architecture is the synaptic links carrying the weights connect every input to the output neurons.

Modeling SLFFNN

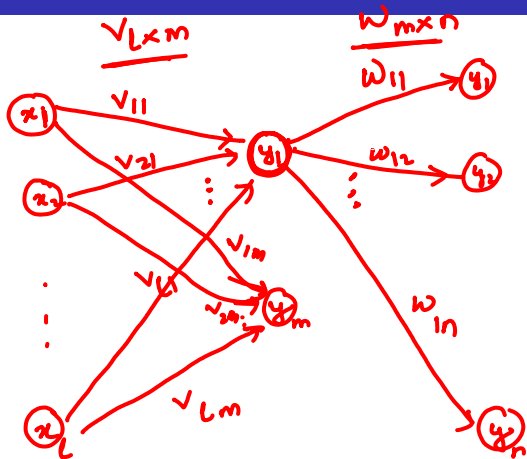
In a single layer neural network, the inputs x_1, x_2, \dots, x_m are connected to the layers of neurons through the weight matrix W . The weight matrix $W_{m \times n}$ can be represented as follows.

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \cdots & W_{1n} \\ W_{21} & W_{22} & W_{23} & \cdots & W_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{m1} & W_{m2} & W_{m3} & \cdots & W_{mn} \end{bmatrix} \quad (1)$$

Multilayer feed forward neural networks

- This network, as its name indicates is made up of multiple layers.
- Thus architectures of this class besides processing an input and an output layer also have one or more intermediary layers called **hidden layers**.
- The hidden layer(s) aid in performing useful intermediary computation before directing the input to the output layer.
- A multilayer feed forward network with l input neurons (number of neuron at the first layer), m_1, m_2, \dots, m_p number of neurons at i -th hidden layer ($i = 1, 2, \dots, p$) and n neurons at the last layer (it is the output neurons) is written as $l - m_1 - m_2 - \dots - m_p - n$ MLFFNN.

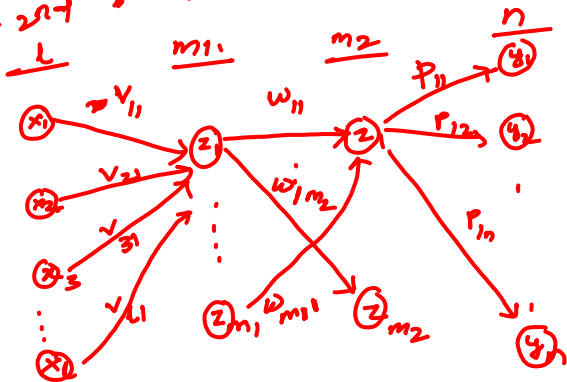
0



$l-m-n$

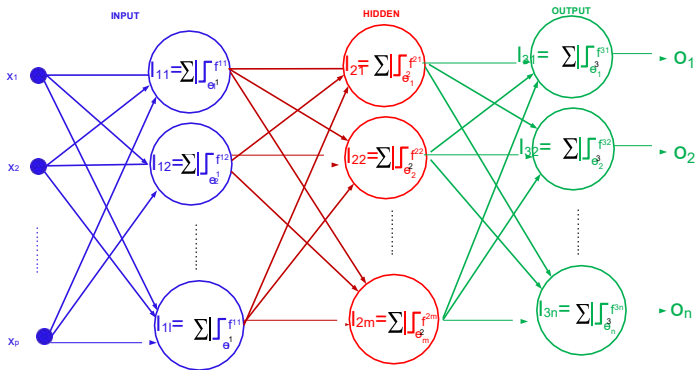
l Input m Hidden n Output

$l - m_1 - m_2 - n$
 \uparrow $\underbrace{\hspace{1cm}}$ \nwarrow
 l input 2 - hidden n output
 neurons layers
 m_1 - first hidden layer
 m_2 - 2nd " "

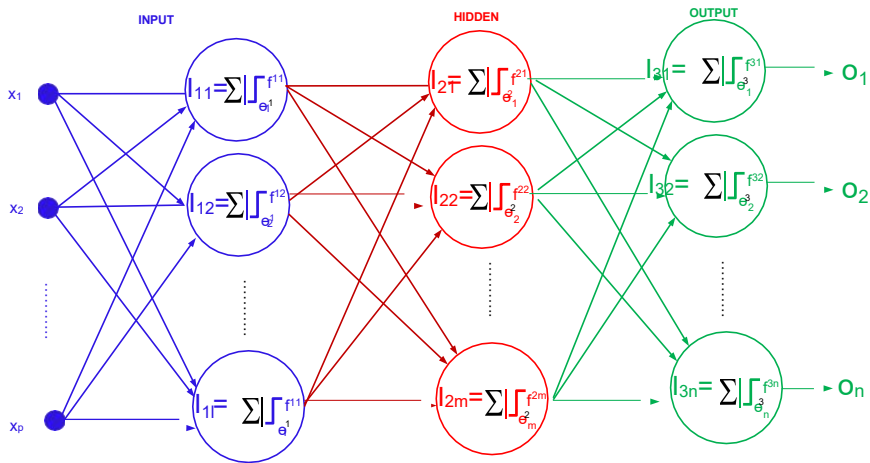


Multilayer feed forward neural networks

Figure shows a schematic diagram of multilayer feed forward neural network with a configuration of $l - m - n$.



Multilayer feed forward neural networks



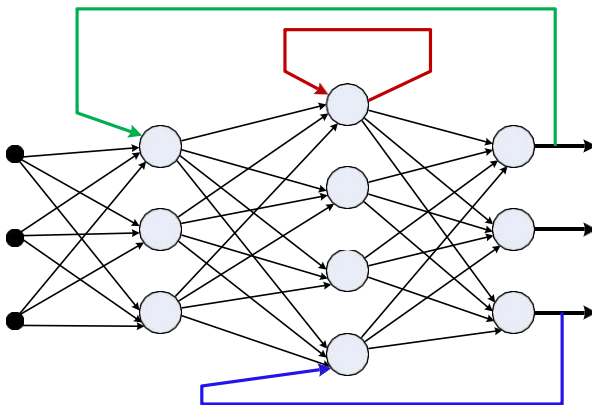
Multilayer feed forward neural networks

- In $l - m - n$ MLFFNN, the input first layer contains l numbers neurons, the only hidden layer contains m number of neurons and the last (output) layer contains n number of neurons.
- The inputs x_1, x_2, \dots, x_p are fed to the first layer and the weight matrices between input and the first layer, the first layer and the hidden layer and those between hidden and the last (output) layer are denoted as W^1 , W^2 , and W^3 , respectively.
- Further, consider that f^1 , f^2 , and f^3 are the transfer functions of neurons lying on the first, hidden and the last layers, respectively.
- Likewise, the threshold values of any i -th neuron in j -th layer is denoted by θ_j^i .

Recurrent neural network architecture

- The networks differ from feedforward network architectures in the sense that there is at least one "feedback loop".
- Thus, in these networks, there could exist one layer with feedback connection.
- There could also be neurons with self-feedback links, that is, the output of a neuron is fed back into itself as input.

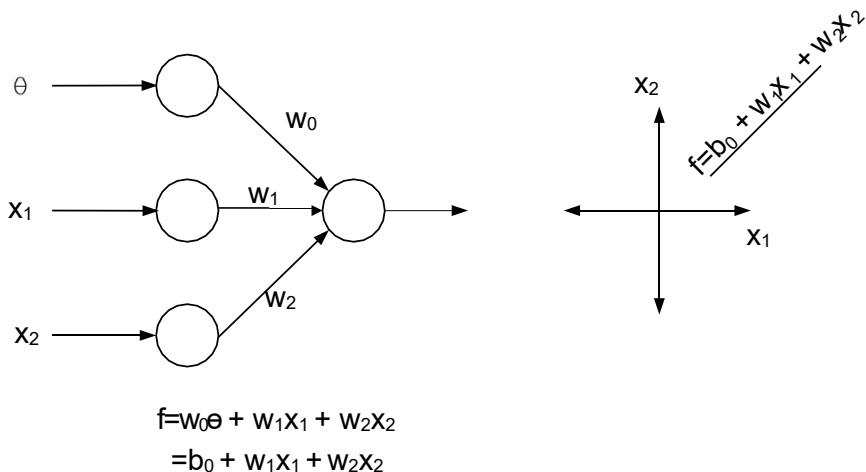
Recurrent neural network architecture



Depending on different type of feedback loops, several recurrent neural networks are known such as Hopfield network, Boltzmann machine network etc.

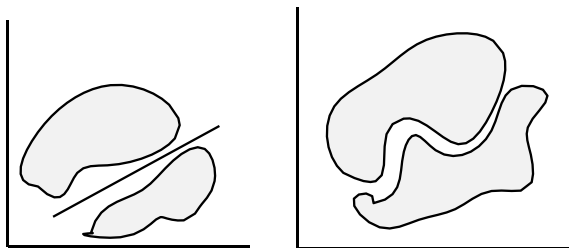
Why different type of neural network architecture?

To give the answer to this question, let us first consider the case of a single neural network with two inputs as shown below.



Revisit of a single neural network

- Note that $f = b_0 + w_1x_1 + w_2x_2$ denotes a straight line in the plane of x_1 - x_2 .
- Now, depending on the values of w_1 and w_2 , we have a set of points for different values of x_1 and x_2 .
- We then say that these points are linearly separable, if the straight line f separates these points into two classes.
- Linearly separable and non-separable points are further illustrated in Figure.





No Bias

The net input (weighted sum)

$$y_{inj} = x_1 * w_{1j} + x_2 * w_{2j}$$

$$+ \vdots x_n * w_{nj}$$

$$= \sum_{i=1}^n x_i w_{ij}$$

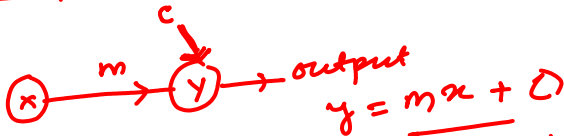
$$[1, x_1, x_2, \dots, x_n]$$

With Bias

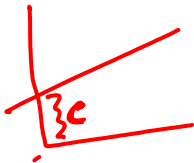
$$y_{inj} = x_0 w_{0j} + \sum_{i=1}^n x_i w_{ij}$$

$$= \textcircled{b_j} + \sum_{i=1}^n x_i w_{ij}$$

$$y_{in j} = b_j + \sum_{i \in I} x_i \cdot w_{ij}$$



straight line equation



AND and XOR problems

To illustrate the concept of linearly separable and non separable tasks to be accomplished by a neural network, let us consider the case of AND problem and XOR problem.

Inputs x_1 x_2		Output (y)
0	0	0
0	1	0
1	0	0
1	1	1

AND Problem

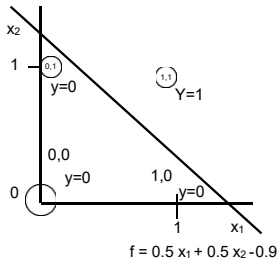
x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem

AND problem is linearly separable

x_1	x_2	Output (y)
0	0	0
0	1	0
1	0	0
1	1	1

The AND Logic

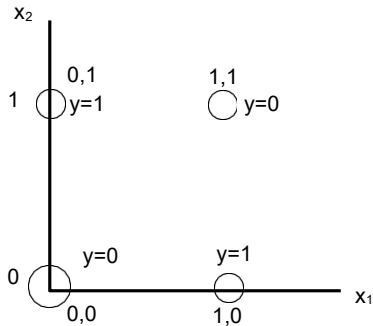


AND-problem is linearly separable

XOR problem is linearly non-separable

x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem



XOR-problem is non-linearly separable

Our observations

From the example discussed, we understand that a straight line is possible in AND-problem to separate two tasks namely the output as 0 or 1 for any input.

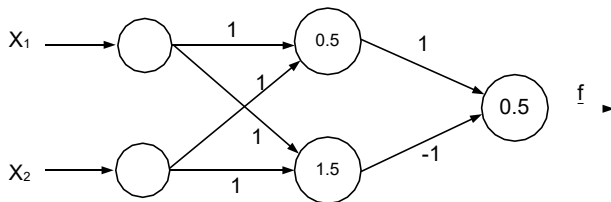
However, in case of XOR problem, such a line is not possible.

Note: horizontal or a vertical line in case of XOR problem is not admissible because in that case it completely ignores one input.

Example

- So, far a 2-classification problem, if there is a straight line, which acts as a decision boundary then we can say such problem as linearly separable; otherwise, it is non-linearly separable.
- The same concept can be extended to n -classification problem.
- Such a problem can be represented by an n -dimensional space and a boundary would be with $n - 1$ dimensions that separates a given sets.
- In fact, any linearly separable problem can be solved with a single layer feed forward neural network. For example, the AND problem.
- On the other hand, if the problem is non-linearly separable, then a single layer neural network can not solves such a problem.
- To solve such a problem, multilayer feed forward neural network is required.

Example: Solving XOR problem

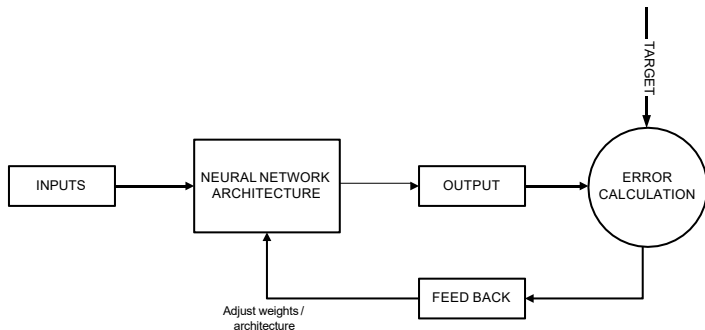


Neural network for XOR-problem

Dynamic neural network

- In some cases, the output needs to be compared with its target values to determine an error, if any.
- Based on this category of applications, a neural network can be static neural network or dynamic neural network.
- In a static neural network, error in prediction is neither calculated nor feedback for updating the neural network.
- On the other hand, in a dynamic neural network, the error is determined and then feed back to the network to modify its weights (or architecture or both).

Dynamic neural network



Framework of dynamic neural network

Dynamic neural network

From the above discussions, we conclude that

- For linearly separable problems, we solve using single layer feed forward neural network.
- For non-linearly separable problem, we solve using multilayer feed forward neural networks.
- For problems, with error calculation, we solve using recurrent neural networks as well as dynamic neural networks.

Thank You.