Reliable Neural Network

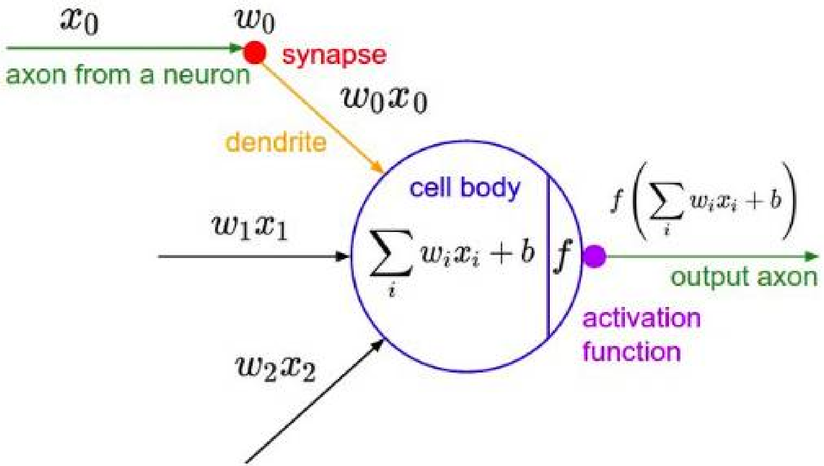
Reliability Theory

Reliability is defined as the probability that a system is still operating correctly.

Failure:

- According to its specification, it was correct at time t=0.
- At time t, the operation of a system may no longer meet its specification

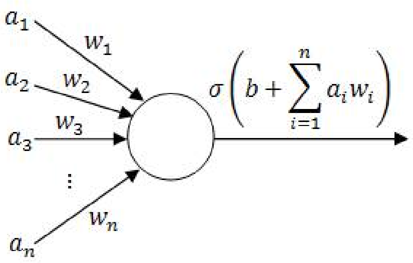
Basics of NN



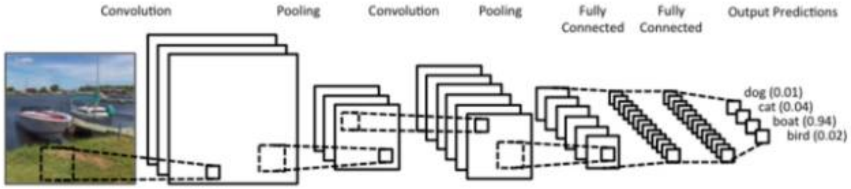
Activation function:

- It takes in the output signal from the previous cell and converts it into some form that can be taken as input to the next cell.
- deciding what is to be fired to the next neuron.
- help the network learn complex patterns in the data.
- Introduces nonlinearity

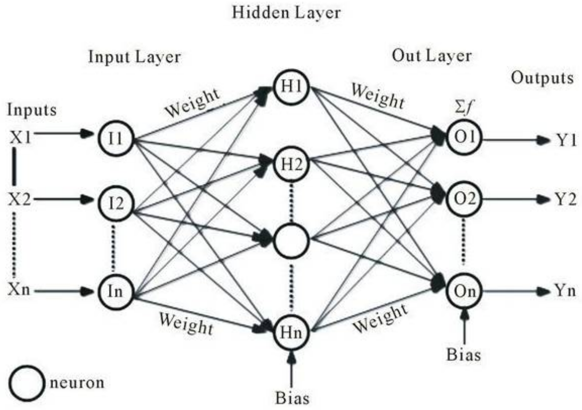
Types of NN



Simple Perceptron



CNN for image classification



MLP

Multi Layer Perceptron

Distribution

- One of the features of neural network's computation which is a major incentive for their application in solving problems is that of distribution.
- Two distinct forms of this property can be considered to exist:
  - Distributed Information Storage
  - Distributed Processing.
- Distributed processing refers to how every unit performs its own function independently of any other units in the neural network.
- However, correctness of its inputs may be dependant on other units.

- The global recall or functional evaluation performed by the entire neural network results from the joint (parallel) operation of all the units.

Drawback

- Faults cannot be located easily.
- In an implementation each component would require extra circuitry to detect and signal the occurrence of a fault.
- The cost and reduction in overall reliability of the system might render this approach unsatisfactory.

Advantage

- However, neural networks also have another important property, they can learn.
- This feature will allow a faulty system, once detected, to be retrained either to remove or to compensate for the faults without requiring them to be located.
- The re-learning process will be relatively fast compared to the original learning time since the n**eural network will only be distorted by the faults, not completely randomised**.

- **distributing information across all units within a neural network is also beneficial if the information load on every unit is approximately equivalent.**
  - **decrease the chance of having critical components which might cause system failure, even if the remainder are free from faults**

Generalisation.

This refers to the ability of a neural network which has been trained using a limited set of training data, to supply a reasonable output to an input which it did not encounter during training.

- As an adaptive system, generalisation in a neural network can be considered to represent the underlying problem rather than just memorising the particular inputs in the training set.
- **Robustness to noisy inputs in classification systems can be a product of generalisation.**

Local vs. Global Generalisation

- Two distinct computational techniques by which a neural network generalises
- identified by considering the nature of the response of internal units to inputs ranging over the input space.
- **Local:**
  - Some neural networks employ units which only activate for inputs in a limited bounded region of input space, e.g. Radial Basis Function networks.
  - An unknown input will only activate those units whose activation regions includes the new input.
- **Global:** This is where the internal units of a neural network respond to all inputs lying anywhere within the input space.

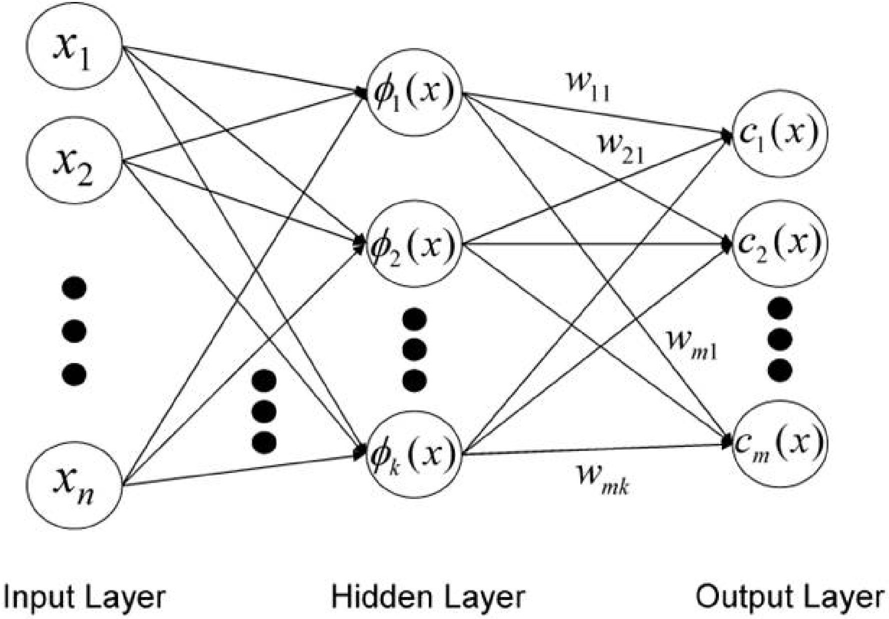
Radial basis function network (RBF)

Radial basis function network is an artificial neural network that uses radial basis functions as activation functions.

The output of the network is a **linear combination of radial basis functions of the inputs and neuron parameters**.

Radial basis function networks have many uses, including function approximation, time series prediction etc.

[Universal approximation theorems imply that neural networks can *represent* a wide variety of interesting functions when given appropriate weights.]



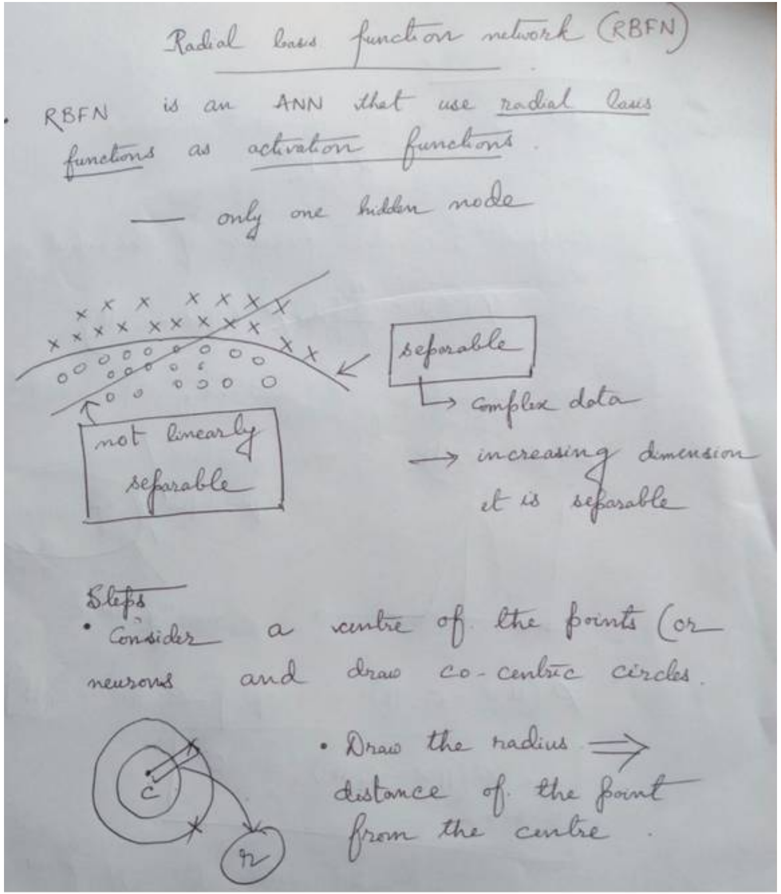
Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer.

The input can be modeled as a vector of real numbers $\mathbf{x} \in \mathbb{R}^n$

The output of the network is then a scalar function of the input vector, $\varphi(\mathbf{x}) = \sum_{i=1}^{N} a_i \rho(||\mathbf{x} - \mathbf{c}_i||)$

Where, $\varphi : \mathbb{R}^n \to \mathbb{R}$

where N is the number of neurons in the hidden layer, $\mathbf{c}_i$ is the center vector for neuron i, and a_i is the weight of neuron i in the linear output neuron.



Radial basis function network (RBFN)

- RBFN is an ANN that use radial basis functions as activation functions.

— only one hidden node

separable → complex data

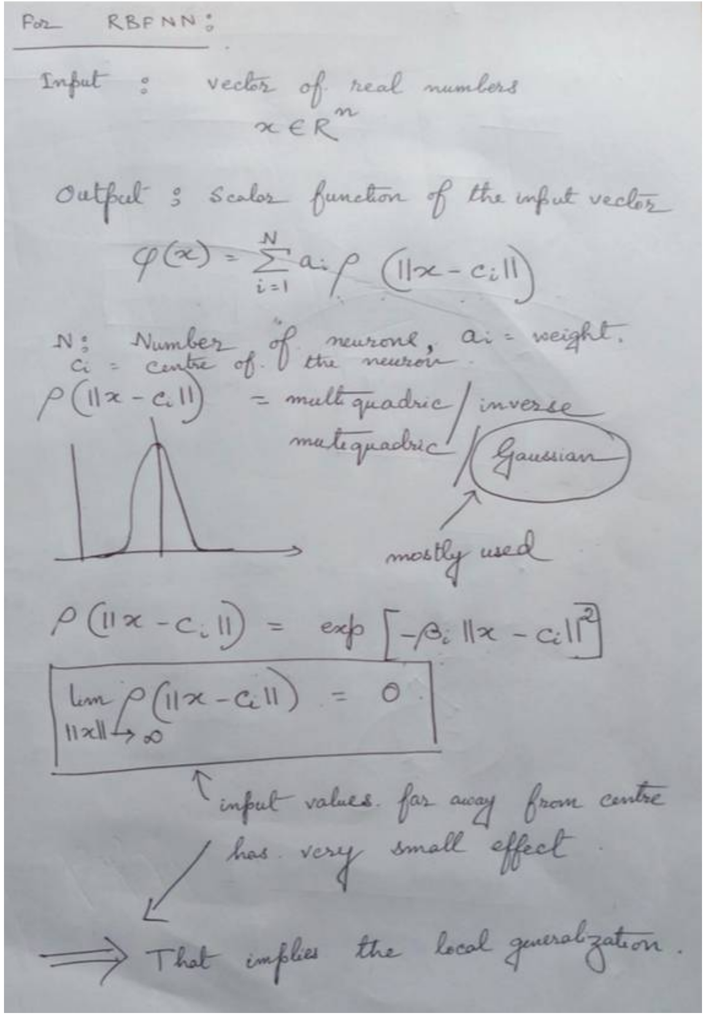
→ increasing dimension it is separable

not linearly separable

Steps

- Consider a centre of the points (or neurons and draw co-centric circles.

- Draw the radius ⟹ distance of the point from the centre.



For RBFNN:

Input : vector of real numbers
$x \in R^n$

Output : Scalar function of the input vector

$$\varphi(x) = \sum_{i=1}^{N} a_i \, \rho\left(\|x - c_i\|\right)$$

N: Number of neurone, a_i = weight.
c_i = centre of the neuron

$\rho(\|x - c_i\|)$ = multiquadric / inverse mutequadric / Gaussian

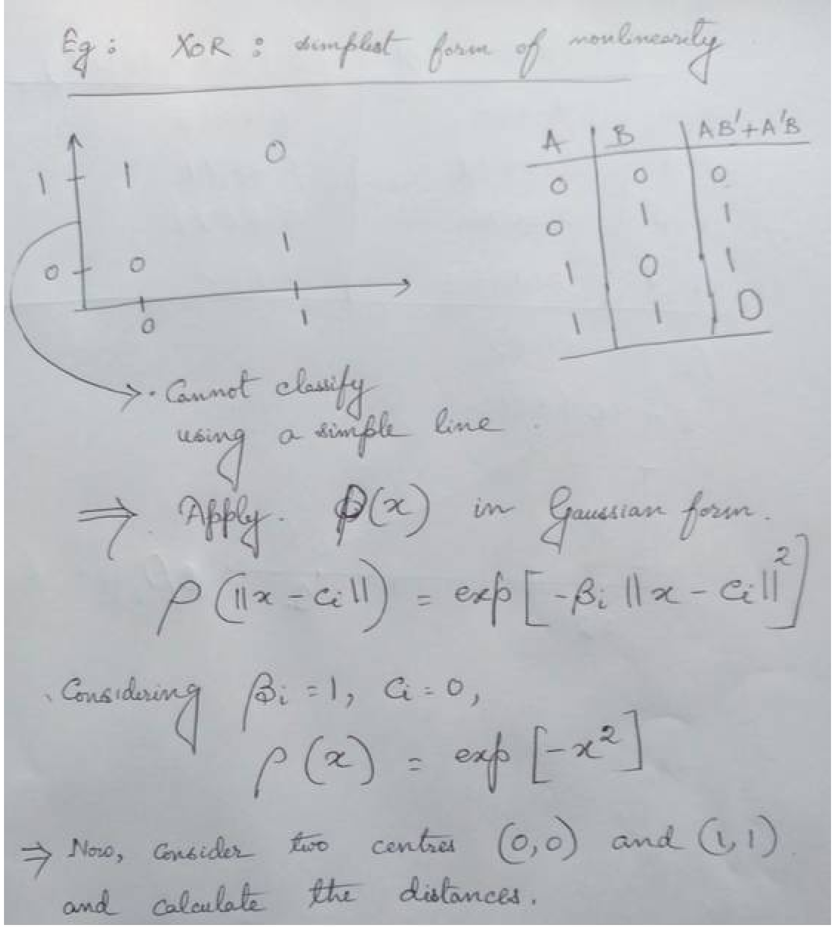
mostly used

$$\rho(\|x - c_i\|) = \exp\left[-\beta_i \|x - c_i\|^2\right]$$

$$\boxed{\lim_{\|x\| \to \infty} \rho(\|x - c_i\|) = 0}$$

input values far away from centre has very small effect

⟹ That implies the local generalization.



Eg: XOR : simplest form of nonlinearity

A	B	AB'+A'B
0	0	0
0	1	1
1	0	1
1	1	0

→ Cannot classify using a simple line.

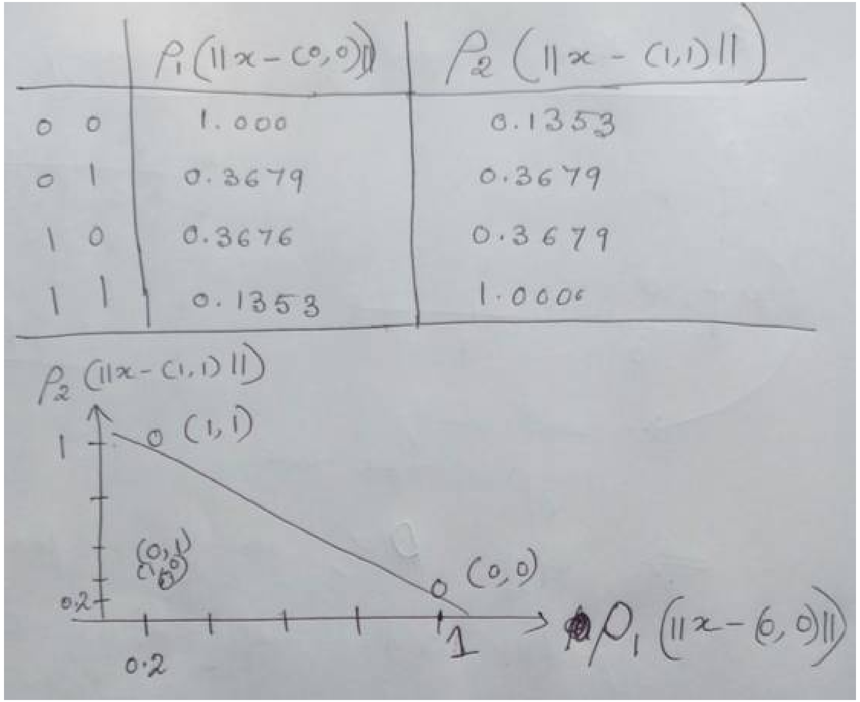
⇒ Apply. $\phi(x)$ in Gaussian form.

$$\rho(\|x - c_i\|) = \exp\left[-\beta_i \|x - c_i\|^2\right]$$

Considering $\beta_i = 1$, $c_i = 0$,

$$\rho(x) = \exp\left[-x^2\right]$$

⇒ Now, consider two centres (0,0) and (1,1) and calculate the distances.

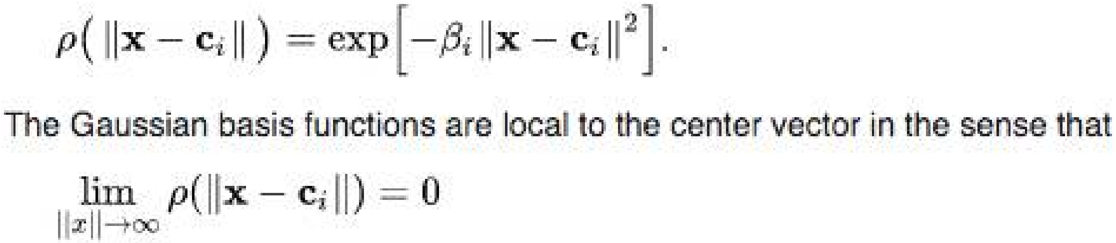


	$\rho_1(\|x-(0,0)\|)$	$\rho_2(\|x-(1,1)\|)$
0 0	1.000	0.1353
0 1	0.3679	0.3679
1 0	0.3676	0.3679
1 1	0.1353	1.000

$\rho_2(\|x-(1,1)\|)$
1
(1,1)
(0,1)
(1,0)
(0,0)
0.2
0.2
1
$\rho_1(\|x-(0,0)\|)$

Why local generalization

- The radial basis function is commonly taken to be Gaussian:



$$\rho\big(\,\|\mathbf{x}-\mathbf{c}_i\|\,\big) = \exp\left[-\beta_i\,\|\mathbf{x}-\mathbf{c}_i\|^2\right].$$

The Gaussian basis functions are local to the center vector in the sense that

$$\lim_{||x||\to\infty} \rho(\|\mathbf{x}-\mathbf{c}_i\|) = 0$$

- Changing parameters of one neuron has only a small effect for input values that are far away from the center of that neuron.

Generalisation and fault tolerance of a neural network

If local :

- Unreliable in limited regions of input space.
- Only when an input falls into a region where the neural network's operation is affected will the effect of faults be apparent and possible failure occur.
    - if a large number of extra units are used in a locally generalising neural network, the degree of overlap between the input space regions of each unit can be increased such that a general improvement in fault tolerance will be achieved.

If Global:

- Global generalisation will cause a small loss of generalisation for any input pattern.