

# Radial Basis Function (RBF) Networks

# Introduction to Radial Basis Function Networks

- Radial basis function network is a type of artificial neural network used in soft computing.
- Mainly applied for function approximation, classification, and pattern recognition.

**Note:** It is a simple feedforward network with **faster training** compared to multilayer perceptrons.

# Structure of Radial Basis Function Networks

An RBFN has 3 main layers:

1. **Input Layer** – passes input features to hidden layer.
2. **Hidden Layer** – contains neurons with **radial basis (Gaussian) activation**.
3. **Output Layer** – performs a **weighted linear combination** of these activations to produce the final network output.

## Work of the Output Layer

- The output layer takes the **activations** from the hidden layer (the values of  $\phi(x)$  from each radial basis neuron).
- It then performs a **weighted linear combination** of these activations to produce the final network output.

Mathematically:

$$y(x) = \sum_{i=1}^M w_i \phi_i(x) + b$$

Where:

- $M$  = number of hidden neurons
- $\phi_i(x)$  = output of the i-th radial basis function
- $w_i$  = weight connecting the i-th hidden neuron to the output
- $b$  = bias term
- $y(x)$  = final output of the network

# Radial Basis Function (Gaussian Function)

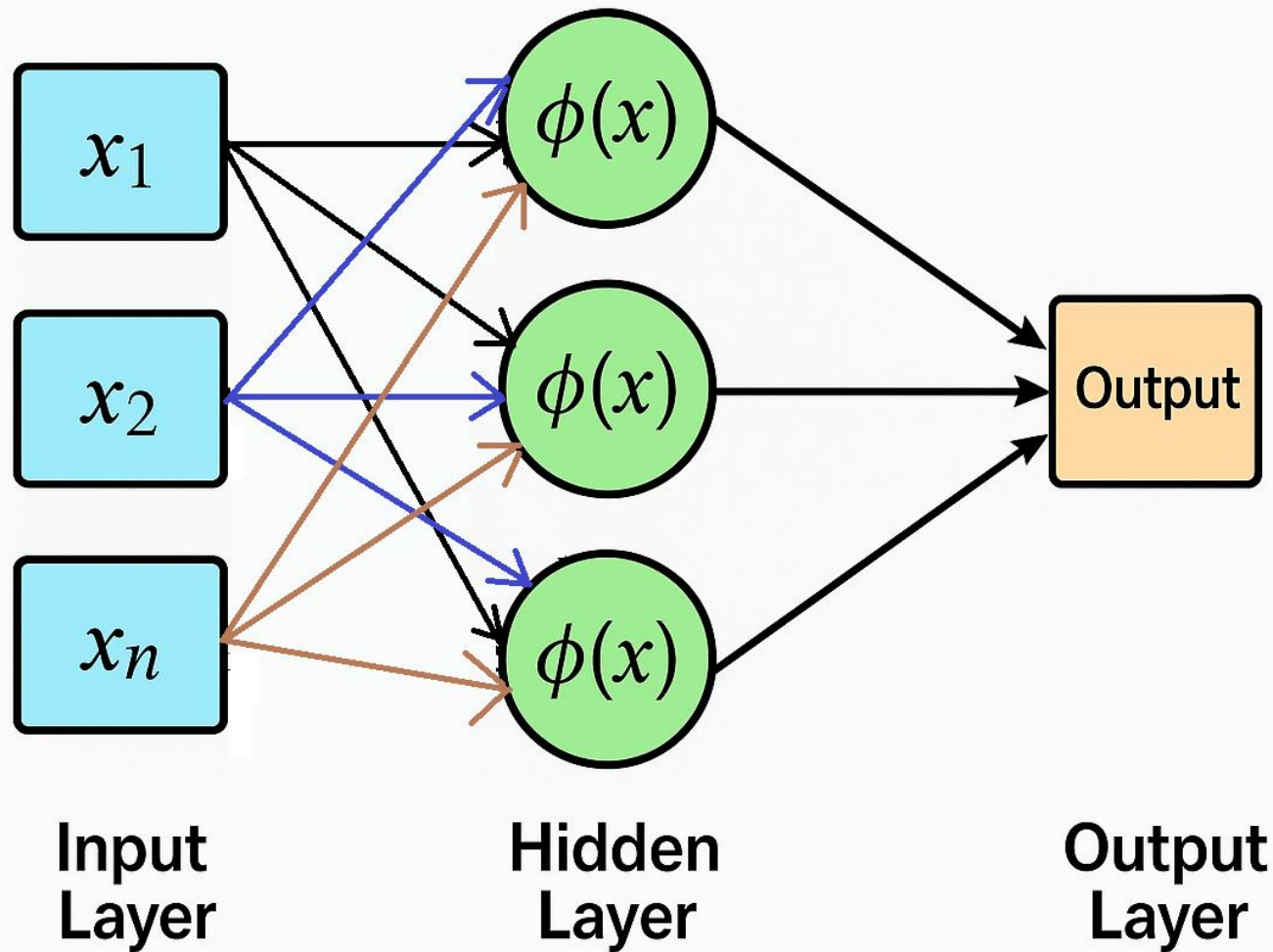
The hidden layer uses Gaussian radial basis functions. The  $\phi$  (phi) symbol refers to the **Radial Basis Function** itself — the **activation function** used in the hidden layer of an RBF Network. :

$$\phi(x) = \exp( - ||x - c||^2 / (2\sigma^2) )$$

Where:

- $x$  = input vector
- $\exp()$  = exponential function
- $c$  =  $n$  dimensional vector (center)
- $\sigma$  = spread (width)

# RADIAL BASIS FUNCTION NETWORKS



## Breaking it down:

- $x$  = input vector (e.g.,  $[x_1, x_2, \dots, x_n]$ )
- $c$  = center of the RBF (also an  $n$ -dimensional vector)
- $x - c$  = the difference vector (how far each component of  $x$  is from  $c$ )
- $\|x - c\|$  = the **Euclidean distance** between  $x$  and  $c$ :

$$\|x - c\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2}$$

- Then,  $\|x - c\|^2$  just means the **squared Euclidean distance** (no square root):

$$\|x - c\|^2 = (x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2$$

---

## Intuition

- If  $x$  is **very close** to  $c$ , then  $\|x - c\|^2$  is small  $\rightarrow$  the Gaussian value is near 1 (high activation).
- If  $x$  is **far away** from  $c$ , then  $\|x - c\|^2$  is large  $\rightarrow$  the Gaussian value is near 0 (low activation).

So the double bars are just a shorthand for "distance between vectors."

# Key Features of RBF Networks

- **Localized Activation**: neurons respond only to nearby inputs.
- **Two-Stage Training**:
  - **Step 1**: Find centers & spreads (unsupervised).
  - **Step 2**: Learn weights (supervised).
- **Fast Training**: due to linear output layer.
- **Universal Approximator**: can approximate any continuous function.