DEPARTMENT OF MECHATRONICS ENGINEERING

A **Radial Basis Function Network (RBFN)** is a type of **artificial neural network** used for **classification, function approximation, and regression problems**. It is a **three-layered feedforward network** that uses **radial basis functions** as activation functions.

Why RBFN?

Fast learning compared to backpropagation networks (BPN).
Efficient function approximation in high-dimensional spaces.
Handles complex, non-linear problems well.
Good generalization with fewer neurons than MLPs.

Applications of RBFN

✓ Pattern Recognition – Face, speech, and handwriting recognition.

✓ Function Approximation – Time-series prediction, weather forecasting.

✓ Control Systems – Robotics, process control.

✓ Medical Diagnosis – Disease classification.

2. Architecture of Radial Basis Function Network (RBFN)

RBFN consists of **three layers**:

1. Input Layer

- Passes the input **directly** to the hidden layer.
- No computations are performed at this layer.

3. Backpropagation and the Generalized Delta Rule

Forward Pass

1. **Input layer** receives inputs $x_1, x_2, ...x_n$.
2. Neurons in the **hidden layer** compute weighted sums and apply an **activation function** (usually **sigmoid** or **ReLU**).
3. The **output layer** computes the final output o_k.

Backward Pass (Error Propagation)

1. Compute the **error** at the output layer:

$$\delta_k = (t_k - o_k) f'(net_k)$$

   where $f'(net_k)$ is the derivative of the activation function.

3. Output Layer

- Computes a weighted sum of hidden neuron activations:

$$y_k = \sum_i w_{ik} \phi_i(x)$$

where w_{ik} are the weights from the hidden layer to the output layer.

3. Learning Algorithm in RBFN

RBFN uses a **three-step learning process**:

1. Select Centers (Unsupervised Learning)

- Centers c_i are chosen using **clustering algorithms** like **K-means** or randomly from training data.

2. Determine Spread (σ_i)

- The spread **controls how localized each RBF neuron** is.
- It is usually set as:

$$\sigma = \frac{d_{\max}}{\sqrt{2N}}$$

where $d_{\max}$ is the maximum distance between centers, and N is the number of centers.

3. Compute Weights (Supervised Learning)

- Weights w_{ik} are optimized using **Least Squares Estimation (LSE)**:

$$W = \Phi^{+} T$$

where:

- Φ^{+} is the **pseudo-inverse** of the activation matrix.
- T is the target output vector.

Comparison: RBFN vs. Backpropagation Network (BPN)

Feature	RBFN	BPN (MLP with Backpropagation)
Architecture	3 layers (Input, RBF, Output)	Multi-layer with fully connected neurons
Activation Function	Radial Basis (Gaussian)	Sigmoid, Tanh, ReLU
Learning Type	Hybrid (Unsupervised +	Supervised (Backpropagation)

	Supervised)	
Training Speed	Faster (LSE for weights)	Slower (Gradient Descent)
Performance on Noisy Data	Good	Moderate
Generalization	Good with proper spread	Needs large datasets
Convergence	Faster	Slower, risk of local minima

5. Advantages and Disadvantages of RBFN

Advantages

- Fast training due to linear weight optimization.
- Handles non-linearity well using radial basis functions.
- Works well for function approximation and classification.
- Good interpretability with fewer neurons.

Disadvantages

- Needs careful selection of centers (clustering-based).
- Sensitive to spread (σ\sigmaσ) – large spread causes over-generalization, small spread causes poor generalization.
- Not efficient for large datasets (requires many basis functions).