

**ML UNIT – 6 (Introduction To Neural Networks) – END-SEM PYQ Answers****➤ MAY / JUN 2023****Q7) a) What are building blocks of neural network, elaborate? [5]**

A neural network is a computational model inspired by the human brain, made up of interconnected processing units called neurons. These neurons work together in layers to learn patterns from input data and make predictions. The building blocks of a neural network decide how information flows, how learning happens, and how accurate the model becomes.

**Building Blocks:****1. Neurons (Nodes)**

- Basic processing units of a neural network.
- Each neuron receives inputs, multiplies them with weights, adds bias, and applies activation function.
- Neurons help transform input information into meaningful outputs.

**2. Layers (Input, Hidden, Output)**

- Input layer accepts raw data.
- Hidden layers perform computations and learn features.
- Output layer gives final prediction/classification.

**3. Weights and Biases**

- Weights decide the strength of connection between neurons.
- Bias shifts the activation function and improves flexibility.
- During training, weights and biases are updated to reduce errors.

**4. Activation Functions**

- Introduce non-linearity into the network.
- Common functions: ReLU, Sigmoid, Tanh.
- Without activation functions, the network behaves like a simple linear model.
- **Extra:** They help the model learn complex and real-world relationships.

**5. Loss Function & Optimizer**

- Loss function measures the error between actual and predicted values.
- Optimizers (like Gradient Descent, Adam) update weights to reduce this error.
- They guide the network during each training step.

**b) Describe the characteristics of Backpropagation Algorithm. [6 Marks]**

Backpropagation is a supervised learning algorithm used to train neural networks by minimizing the error between predicted and actual outputs. It works by propagating the error backward from the output layer to the input layer and updating weights to improve accuracy. This algorithm is essential for efficiently training multi-layer neural networks.

**Characteristics of Backpropagation Algorithm:****1. Supervised Learning Method**

- It requires labeled data for training.
- The algorithm compares predicted output with the actual output.
- Error is calculated using a loss function.

**2. Gradient-Based Weight Update**

- Uses Gradient Descent to update weights.
- Calculates the gradient of the loss function w.r.t each weight.
- Adjusts weights in the direction that reduces error.

**3. Works Layer-by-Layer (Backward Pass)**

- Error is propagated backward from output layer to hidden layers.
- Each layer updates its weights based on its contribution to the error.
- This allows complex multi-layer networks to learn efficiently.

**4. Minimizes Error Using Loss Function**

- A loss function (e.g., MSE, cross-entropy) measures prediction error.
- Backprop optimizes weights to minimize this loss.
- Lower loss means better learning.

**5. Efficient Computation Using Chain Rule**

- Uses calculus chain rule to compute gradients quickly.
- Saves time by avoiding repeated calculations.
- Suitable for large neural networks.

**6. Requires Differentiable Activation Functions**

- Only works when activation functions are differentiable (e.g., Sigmoid, ReLU, Tanh).
- Necessary for gradient calculation.
- Makes smooth optimization possible.

**c) Write a short note on Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN). [6 Marks]**

**1. Recurrent Neural Network (RNN)**

- Recurrent Neural Network is a type of neural network designed to handle sequential or time-dependent data.
- It has a feedback connection where the output of the previous step is fed back into the network, allowing it to remember past information.
- This makes RNNs suitable for tasks where order and context of data are important.

**Key points:**

- Works on sequential data such as text, speech, and time-series.
- Maintains memory of previous inputs through hidden states.
- Useful in applications like language modeling, sentiment analysis, and machine translation.

**2. Convolutional Neural Network (CNN)**

- Convolutional Neural Network is a deep learning model mainly used for image and spatial data processing.
- It uses convolutional layers to extract features such as edges, textures, and patterns from images.
- CNN reduces the number of parameters using filters and pooling, making it computationally efficient.

**Key points:**

- Extracts spatial features using convolution filters.
- Includes pooling layers to reduce dimensionality.
- Widely used in image classification, object detection, and facial recognition.

**Q8) a) Explain artificial neural network based on perception concept with diagram. [6]**

**1. Introduction**

- An **Artificial Neural Network (ANN)** is a computational model inspired by the structure and functioning of the **human brain**.
- It consists of interconnected units called **neurons**, which process information in parallel.

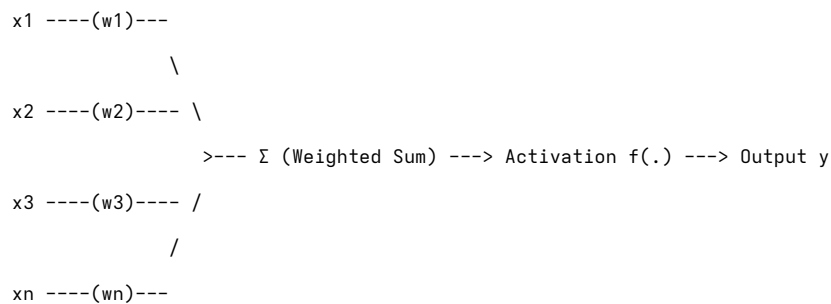
**2. Perceptron Concept**

- The **Perceptron** is the **simplest ANN model**, introduced by Frank Rosenblatt.
- It represents a **single neural unit** that takes multiple inputs and produces a single output.
- It is used mainly for **binary classification**.

### 3. Structure of a Perceptron: A perceptron contains:

- **Input nodes** ( $x_1, x_2, \dots, x_n$ )
- **Weights** ( $w_1, w_2, \dots, w_n$ )
- **Summation unit**
- **Activation (transfer) function**
- **Output (y)**

**Diagram:**



### 4. Working of a Perceptron

- Each input  $x_i$  is multiplied by its weight  $w_i$ .
- A weighted sum is computed:  

$$\text{NET} = w_1x_1 + w_2x_2 + \dots + w_nx_n + \text{bias}$$
- NET value is passed to an **activation function** such as:
  - Step function
  - Sign function
  - Sigmoid (in extended models)

- The output is:

$$y = f(\text{NET})$$

- If  $\text{NET} \geq \text{threshold} \rightarrow \text{output is } 1$
- Else  $\rightarrow \text{output is } 0$

### 5. Learning in Perceptron

- Weights are updated using **Perceptron Learning Rule**:

$$w_i(\text{new}) = w_i(\text{old}) + \eta (d - y) x_i$$

where,  $\eta$  = learning rate,  $d$  = desired output,  $y$  = actual output.

- Weights adjust until the perceptron correctly classifies all training examples (linearly separable).

## 6. Characteristics of ANN Based on Perceptron

- Simple model for **pattern recognition and classification**.
- Works only with **linearly separable data**.
- Foundation of modern neural networks like multi-layer perceptrons (MLPs).
- Supports parallel and distributed information processing.

## 7. Applications

- Basic classification tasks
- Logic gate implementation (AND, OR)
- Simple pattern detection
- Building block for deeper ANN models

**Conclusion:** A perceptron-based ANN is the fundamental unit of neural networks. It mimics the functioning of a biological neuron by applying weighted inputs, summing them, and passing the result through an activation function to generate an output.

### b) Describe multi-layer neural n/w. Explain why back propagation algorithm is required. [6]

#### Multi-Layer Neural Network (MLNN):

- A multi-layer neural network consists of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple neurons, and every neuron in one layer is connected to neurons in the next layer through weighted links.
- MLNN can learn complex and non-linear patterns because hidden layers extract deeper features from the input data.
- MLNN processes input by passing it layer by layer through activation functions, transforming the data at each stage. As the number of layers increases, the network becomes capable of solving advanced tasks like image recognition, speech processing, and classification problems.
- This structure makes MLNN more powerful than a single-layer perceptron.

#### Why Backpropagation is Required:

- In a multi-layer network, errors cannot be corrected easily because hidden layers do not have direct target outputs.
- Backpropagation calculates the error at the output and distributes it backward through the layers.
- It updates the weights of each neuron in every layer to reduce the overall error.
- Without backpropagation, MLNN would not know which weights caused the error or how to adjust them.
- It allows efficient learning in deep networks by using gradients.

Backpropagation is therefore essential for training multi-layer neural networks and enabling them to achieve accurate predictions.

**c) Discuss any two activation functions with example. [5 Marks]**

**1. Sigmoid Activation Function:** The sigmoid function converts the input into a value between 0 and 1. It is commonly used in binary classification problems because its output can be interpreted as probability.

**Formula:** 
$$f(x) = \frac{1}{1 + e^{-x}}$$

**Example:** If the input to a neuron is  $x = 2$ ,  $f(2) = \frac{1}{1 + e^{-2}} \approx 0.88$

This means the neuron activates with 88% strength.

**Usage:** Used in logistic regression, binary output layers, and early neural networks.

**2. ReLU (Rectified Linear Unit) Activation Function:** ReLU outputs zero for negative values and returns the same value for positive inputs. It is widely used because it speeds up training and avoids the vanishing gradient problem.

**Formula:** 
$$f(x) = \max(0, x)$$

**Example:**

If input is  $x = -3$ , output = 0.

If input is  $x = 5$ , output = 5.

**Usage:** Used in hidden layers of deep neural networks, CNNs, and modern deep learning models.

➤ **MAY / JUN 2024**

**Q7) a) What is Multilayer perceptron? Describe with diagram.[6]**

A Multilayer Perceptron (MLP) is an artificial neural network that contains an input layer, one or more hidden layers, and an output layer. It can learn complex and non-linear relationships because multiple layers process data step-by-step.

**1. Layered Architecture**

- Consists of input layer, hidden layers, and output layer.
- Each layer contains multiple neurons.

**2. Fully Connected Neurons**

- Every neuron in one layer connects to all neurons in the next layer.
- Uses weighted links for information flow.

**3. Uses Activation Functions**

- Non-linear functions like Sigmoid, ReLU, Tanh.
- Helps network learn complex patterns.

#### 4. Learns Through Backpropagation

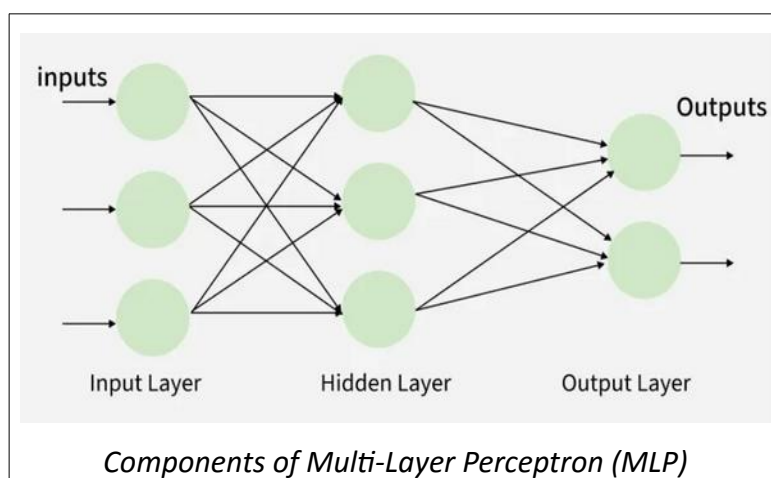
- Weights are adjusted by minimizing error.
- Training becomes efficient for deep networks.

#### 5. Can Solve Complex Problems

- Works for classification, prediction, pattern recognition.
- Performs better than single-layer perceptron.

#### 6. Processes Input Step-by-Step

- Input → Hidden layers → Output.
- Each layer extracts deeper features.



#### b) What are different activation functions used in Neural Networks? [6 Marks]

##### 1. Sigmoid Function

- Output ranges between 0 and 1.
- Good for binary classification.
- Smooth and differentiable.

##### 2. ReLU (Rectified Linear Unit)

- Outputs 0 for negative values, x for positive values.
- Helps avoid vanishing gradient problem.
- Fast and widely used in deep networks.

##### 3. Tanh (Hyperbolic Tangent)

- Output ranges between -1 and +1.
- Centered around zero, better than sigmoid in many cases.
- Used in hidden layers.

#### 4. Softmax Function

- Converts outputs into probabilities.
- Used in multi-class classification.
- Output values sum to 1.

#### 5. Leaky ReLU

- Small slope for negative values instead of 0.
- Solves the “dead ReLU” problem.
- Useful for deep models.

#### 6. Linear Activation

- Output = input (no transformation).
- Used in regression problems.
- Suitable for continuous output prediction.

### c) Explain Convolution Neural Network (CNN) with suitable example. [6 Marks]

A Convolutional Neural Network (CNN) is a deep learning model that automatically learns important features from images using convolution filters. It is mainly used for image classification, object detection, and computer vision tasks.

#### 1. Convolution Layers

- Filters (kernels) slide over the image.
- Detect edges, lines, textures, and shapes.
- Extract important spatial features.

#### 2. Pooling Layers

- Reduce the size of feature maps.
- Make computation faster and reduce overfitting.
- Commonly used: Max Pooling.

#### 3. Activation Functions

- ReLU is widely used after convolution.
- Adds non-linearity to learn complex patterns.
- Helps faster training.

#### 4. Fully Connected Layers

- Used after convolution and pooling.



- Convert extracted features into final outputs.
- Perform classification at the end.

### 5. Spatial Feature Learning

- Learns simple features first (edges, corners).
- Deeper layers learn bigger features (objects, shapes).
- Very effective for visual data.

### 6. Suitable Example

Example: **Image Classification**

- Input: Image of a dog or cat.
- Convolution layers detect ears, eyes, outlines, fur.
- Pooling reduces unnecessary details.
- Fully connected layer predicts → “Dog” or “Cat”.

### Q8) a) Explain building blocks of RBF networks. [6 Marks]

A Radial Basis Function (RBF) Network is a type of neural network that uses radial basis functions as activation functions in the hidden layer. It is mainly used for function approximation, pattern recognition, and interpolation.

#### 1. Input Layer

- Takes the input features and passes them to the hidden layer.
- No processing is done here.
- Simply distributes the input values to all hidden neurons.

#### 2. Hidden Layer with RBF Neurons

- Each neuron uses a radial basis function (usually Gaussian).
- Computes distance between input and neuron’s center.
- Produces high activation only for inputs close to its center.

#### 3. Centers and Spread ( $\sigma$ ) Values

- Each RBF neuron has a center (mean).
- Spread/width controls how wide the RBF curve is.
- Determines sensitivity of the neuron.

#### 4. Distance Calculation

- Calculates Euclidean distance between input and center.

- Distance value goes into the RBF function.
- Helps measure similarity between input and RBF neuron.

## 5. Output Layer

- Takes outputs from RBF neurons.
- Applies linear combination (weights  $\times$  activations).
- Produces the final output of the network.

## 6. Weights and Training

- Weights connect hidden layer to output layer.
- Training adjusts these weights for accurate prediction.
- Centers and spreads can also be learned or predefined.

## b) Personalized Recommendation & Content-Based Recommendation [6 Marks]

### 1. Personalized Recommendation

A personalized recommendation system suggests items based on an individual user's past behaviour, choices, and preferences. It adapts to the taste of each user and provides unique suggestions.

- Uses user history such as searches, watched items, purchases, clicks, and ratings.
- Generates different recommendations for every user.
- Learns from user patterns using machine learning models.
- Continuously improves as the user interacts more with the system.
- Helps users discover new items similar to their interests.
- Enhances user satisfaction and engagement by being highly relevant.

### 2. Content-Based Recommendation

A content-based recommendation system suggests items that are similar to what the user previously liked, based on item features such as keywords, genre, description, or attributes.

- Uses item attributes like type, category, tags, genre, and product features.
- Recommends items similar to the user's liked or rated items.
- Does not depend on the behavior of other users.
- Can work even when user-base is small or new (no cold-start issue).
- Analyses similarity using techniques like cosine similarity or feature vectors.
- Works well for movies, songs, articles, and product recommendations.

### c) Explain Recurrent Neural Networks with as example.

A Recurrent Neural Network (RNN) is a type of neural network designed to work with sequential data where the order of inputs matters. It has feedback connections that allow the network to remember information from previous time steps, making it suitable for time-dependent tasks.

#### 1. Works on sequential/time-series data

- Input is processed step-by-step.
- Order of data is important (e.g., text, speech, sensor data).

#### 2. Has feedback/loop connections

- Output of previous step is fed back as input to the next step.
- Helps the network maintain memory of past information.

#### 3. Uses hidden state to store memory

- Hidden state carries information across time steps.
- Allows the network to learn long-term patterns.

#### 4. Good for context-based tasks

- Remembers what came before in the sequence.
- Useful in applications like language generation and prediction.

#### 5. Trained using Backpropagation Through Time (BPTT)

- Special version of backpropagation used for sequential data.
- Adjusts weights considering time dependencies.

#### 6. Example of RNN

Example: **Next Word Prediction**

- Input: "I am going to the..."
- RNN reads each word one by one and updates its hidden state.
- Predicts next word like "**market**", "**school**", or "**gym**" based on learned patterns.

### ➤ MAY / JUN 2025

#### Q7) a) Write short note on importance of Activation function used in Neural Network. [5]

Activation functions are used in neural networks to introduce non-linearity and decide whether a neuron should be activated or not. They help the network learn complex patterns and relationships from data that cannot be learned using linear models alone.

**1. Introduces non-linearity**

- Enables the network to learn complex and non-linear relationships.
- Without activation functions, the network behaves like a linear model.

**2. Controls neuron output**

- Determines how much input signal should pass to the next layer.
- Helps neurons decide whether to activate or not.

**3. Improves learning capability**

- Allows deep neural networks to learn meaningful features layer by layer.
- Makes the model powerful for real-world problems.

**4. Supports backpropagation**

- Differentiable activation functions help compute gradients.
- Enables effective weight updates during training.

**5. Affects training speed and performance**

- Proper choice (ReLU, Sigmoid, Tanh) improves convergence speed.
- Helps avoid problems like vanishing gradient.

**b) Explain Recurrent Neural Network (RNN) with example. Compare it with simple CNN. [6 Marks]****Recurrent Neural Network (RNN)**

A Recurrent Neural Network (RNN) is a type of neural network used to process sequential or time-dependent data.

It has feedback connections that allow information from previous steps to be remembered and reused, which helps in understanding context.

- Designed for sequential data where order matters.
- Maintains a hidden state that stores past information.
- Output of a previous step is fed back into the network.
- Suitable for text, speech, and time-series data.

**Example:**

- Input sentence: "I am going to school"
- RNN reads words one by one and remembers previous words.
- Uses context to predict the next word or meaning of the sentence.

<b>RNN</b>	<b>Simple CNN</b>
<b>Used for sequential and time-series data</b>	Used for image and spatial data
<b>Has feedback/loop connections</b>	No feedback connections
<b>Remembers past information</b>	Works only on current input
<b>Processes data step-by-step</b>	Processes data in parallel
<b>Used in text, speech, prediction</b>	Used in image classification, detection

### c) Draw & Explain CNN architecture

A Convolutional Neural Network (CNN) is a deep learning model used mainly for image and visual data processing. It learns features automatically from images by passing the input through convolution, pooling, and fully connected layers.

#### 1. Input Layer

- Accepts the input image.
- Image is represented as pixels with height, width, and channels.
- Example: grayscale or color image.

#### 2. Convolution Layer

- Applies filters (kernels) to the input image.
- Extracts features like edges, lines, corners, and textures.
- Produces feature maps.

#### 3. Activation Function

- Usually ReLU is applied after convolution.
- Introduces non-linearity.
- Helps the network learn complex patterns.

#### 4. Pooling Layer

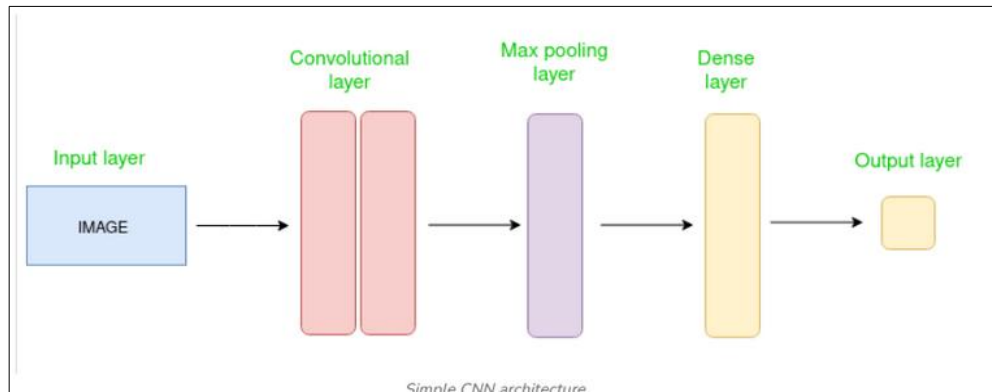
- Reduces the size of feature maps.
- Decreases computation and overfitting.
- Max pooling is commonly used.

#### 5. Fully Connected (Dense) Layer

- Feature maps are flattened into a single vector.
- Dense layer performs classification.
- Combines all extracted features.

## 6. Output Layer

- Produces final prediction or class label.
- Uses Softmax for multi-class problems.
- Example: Cat, Dog, Car, etc.



### Q8) a) Compare Backpropagation Network with Feed Forward Network. [5 Marks]

Aspect	Feed-Forward Neural Network (FFNN)	Backpropagation
<b>Definition</b>	A type of neural network where data flows in one direction from input to output.	A training algorithm that updates weights by minimizing error.
<b>Data Flow</b>	Unidirectional: input → hidden layers → output.	Bidirectional: forward pass for predictions, backward pass for error correction.
<b>Purpose</b>	Produces predictions or classifications.	Trains the network by optimizing weights.
<b>Process</b>	Processes inputs through layers to generate an output.	Calculates error, computes gradients and updates weights iteratively.
<b>Complexity</b>	Conceptually simple—only forward data flow.	More complex—involves calculus (gradients) and optimization.
<b>Usage</b>	Defines the network's structure and prediction mechanism.	Applied during training to improve prediction accuracy.
<b>Role in Learning</b>	Does not involve learning by itself; just forwards inputs.	Essential for learning as it updates weights over time.
<b>Error Handling</b>	Does not handle errors directly.	Directly minimizes errors by propagating them backward.

Aspect	Feed-Forward Neural Network (FFNN)	Backpropagation
<b>Iterations</b>	Single pass of input through the network.	Requires multiple iterations (epochs) for effective training.
<b>Interaction with Layers</b>	Involves input, hidden and output layers for data flow.	Interacts with all layers during weight adjustment.

### b) What are different types of padding used in CNN? [6].

Padding in Convolutional Neural Networks (CNN) is used to add extra pixels around the border of an input image so that important features at the edges are not lost during convolution.

#### Types of Padding used in CNN

##### 1. Valid Padding

- No padding is added to the input image.
- Output size becomes smaller after convolution.
- Border information may be lost.

##### 2. Same Padding

- Zero padding is added around the input image.
- Output size remains same as input size.
- Preserves edge information.

##### 3. Zero Padding

- Input image is padded with zeros.
- Most commonly used type of padding.
- Helps control feature map size.

##### 4. Constant Padding

- Input is padded with a constant value (not necessarily zero).
- Useful when specific border values are required.
- Less commonly used than zero padding.

##### 5. Reflection Padding

- Padding values are reflected from the edge of the image.
- Reduces artificial borders.
- Preserves edge continuity.

##### 6. Replication Padding

- Border pixels are replicated to create padding.
- Maintains edge intensity.
- Useful when edge features are important.

### c) Write short note on layers used in CNN. [6 Marks]

#### i) Convolutional Layer

The convolutional layer is the most important layer in a Convolutional Neural Network. It is responsible for extracting features from the input image. This layer uses small matrices called filters or kernels that slide over the input image and perform convolution operations.

- Each filter detects specific features such as edges, lines, corners, and textures.
- The result of applying these filters is known as a feature map. As we move deeper into the network, convolutional layers learn more complex features like shapes and objects.  
The convolutional layer preserves the spatial relationship among pixels, which helps in better understanding image structure.
- Usually, an activation function like ReLU is applied after convolution to introduce non-linearity and improve learning capability.

#### ii) Pooling Layer

The pooling layer is used to reduce the spatial size of feature maps produced by the convolutional layer.

- It helps in decreasing computation, memory usage, and overfitting.
- Pooling works by selecting a representative value from a small region of the feature map.
- The most commonly used pooling technique is Max Pooling, where the maximum value from a region is selected.
- Another method is Average Pooling, where the average value is taken.  
Pooling makes the network more robust to small variations, shifts, or noise in the input image. By reducing the feature map size, pooling also improves the overall efficiency and speed of the CNN while keeping the most important information.

### ➤ NOV / DEC 2023

#### Q7) a) Write a note on Single Layer Neural Network. [4]

A Single Layer Neural Network is the simplest form of artificial neural network in which there is only one layer of adjustable weights. It consists of an input layer directly connected to the output layer without any hidden layer.

#### Points:

##### 1. Structure

- Has an input layer and an output layer only.
- No hidden layers are present.



## 2. Working Principle

- Inputs are multiplied with weights and summed.
- The result is passed through an activation function to produce output.

## 3. Learning Mechanism

- Weights are updated based on error between actual and predicted output.
- Uses simple learning rules like the perceptron learning rule.

## 4. Capabilities and Limitations

- Can solve only linearly separable problems.
- Easy to design, fast to train, but limited in performance.

5. **Example:** Perceptron used for basic binary classification.

### b) Explain Radial Basis Function networks in detail. [8]

A **Radial Basis Function (RBF) network** is a **feed-forward artificial neural network** that uses **radial basis functions as activation functions** in the hidden layer. It is mainly used for **function approximation, classification, pattern recognition, and time-series prediction**.

An RBF network has **three layers**:

- **Input Layer:** Passes the input features directly to the hidden layer without performing computations.
- **Hidden Layer:** Contains neurons with **radial basis (typically Gaussian) activation functions**. Each neuron measures the distance between the input vector and a **center vector**.
- **Output Layer:** A linear layer that combines weighted outputs of the hidden neurons to produce the final result.

## 2) Radial Basis Function

The most commonly used RBF is the **Gaussian function**:

$$\phi(\|x - c\|) = \exp\left(-\frac{\|x - c\|^2}{2\sigma^2}\right)$$

where

- $x$  = input vector
- $c$  = center of the neuron
- $\sigma$  = spread (width) parameter

The output is **high when the input is close to the center** and decreases as the distance increases.

**3) Training of RBF Network:** Training is usually done in **two stages**:

1. **Hidden layer training:** Centers are selected using methods like **K-means clustering**, and spreads are determined from distances between centers.
2. **Output layer training:** Weights are learned using **supervised learning**, often with **least squares** or simple gradient descent.

This makes training **fast and stable** compared to deep MLPs.

**4) Characteristics**

- Uses **localized activation** (responds strongly to nearby inputs).
- Faster convergence due to simpler training.
- Good approximation capabilities.
- Less prone to local minima.

**5) Applications**

- Function approximation
- Classification problems
- Time-series forecasting
- System modeling and control

**6) Advantages and Limitations****Advantages:**

- Simple structure and fast learning
- Strong performance for interpolation tasks
- Easy mathematical interpretation

**Limitations:**

- Requires careful selection of centers and spreads
- May need a **large number of hidden neurons** for complex problems
- Memory intensive for high-dimensional data

**c) Explain Recurrent Neural Networks and its applications in brief. [5 Marks]****Recurrent Neural Network (RNN):**

A Recurrent Neural Network is a neural network designed to process sequential or time-dependent data. It uses feedback connections so that information from previous steps can be stored and reused while processing new inputs.

**Key Points:**

1. Designed to handle **sequential data** such as text and time series.
2. Contains **feedback (loop) connections** that allow past outputs to influence current processing.
3. Maintains a **hidden state** which acts as memory of previous inputs.
4. Capable of learning **temporal dependencies** in data.
5. Trained using **Backpropagation Through Time (BPTT)**.

**Applications of RNN:**

- Natural language processing and text prediction
- Speech recognition systems
- Time-series forecasting
- Sentiment analysis

**Q8) a) Explain the concept of Back Propagation in ANN with example. [8]**

**Back Propagation** is a **supervised learning algorithm** used to train **multi-layer feedforward neural networks**. It works by computing the error at the output layer and **propagating this error backward** through the network to update the weights so that the overall error is minimized.

**1) Concept of Back Propagation:** The main idea of back propagation is to **reduce the difference between actual output and desired output** by adjusting network weights using **gradient descent**. The algorithm uses the **chain rule of calculus** to compute how changes in weights affect the error.

**2) Steps in Back Propagation Algorithm****Step 1: Forward Propagation**

- Input values are applied to the input layer.
- Signals move forward through hidden layers to the output layer.
- Network produces an output.

**Step 2: Error Calculation**

- Error is calculated using a loss function such as **Mean Squared Error (MSE)**:

$$E = \frac{1}{2}(t - o)^2$$

where  $t$  is target output and  $o$  is actual output.

**Step 3: Backward Propagation of Error**

- Error is propagated backward from output layer to hidden layers.
- Partial derivatives of error with respect to each weight are calculated.

**Step 4: Weight Update**

- Weights are updated using gradient descent:

$$W_{new} = W_{old} - \eta \frac{\partial E}{\partial W}$$

where  $\eta$  is the learning rate.

These steps are repeated until the error becomes minimum or a stopping condition is met.

**3) Example:** Consider a simple ANN for **binary classification** with one hidden layer.

- Input:  $X_1, X_2$
- Output: 0 or 1
- Suppose the target output is 1 but the network predicts 0.6.  
The error is computed and propagated backward, adjusting weights so that the next prediction moves closer to 1. Over multiple iterations, the network learns the correct mapping.

**4) Features of Back Propagation**

- Uses **supervised learning**
- Works with **non-linear activation functions**
- Supports learning in multi-layer networks
- Minimizes training error iteratively

**5) Advantages and Limitations****Advantages:**

- Efficient for complex problems
- Can model non-linear relationships
- Widely used in deep learning

**Limitations:**

- Slow convergence for large networks
- Can get stuck in local minima
- Performance depends on learning rate and initialization

**b) What is Functional Link Artificial Neural Network (FLANN)? Explain its merits over other ANNs.**

A **Functional Link Artificial Neural Network (FLANN)** is a **single-layer feedforward neural network** that improves learning capability by **expanding the input space using functional links**. Unlike multilayer ANNs, FLANN does **not have hidden layers**; instead, nonlinearity is introduced by applying **mathematical functions** to the input features.

## Concept of FLANN

In FLANN, each input is transformed using a set of **basis functions** such as:

- Trigonometric functions ( $\sin$ ,  $\cos$ )
- Polynomial functions
- Exponential functions

The original inputs along with their functionally expanded values are fed directly to the output neuron(s). This allows FLANN to handle **nonlinear problems with a simple structure**.

## Architecture

- **Input Layer:** Original input signals
- **Functional Expansion Block:** Generates nonlinear terms (e.g.,  $x$ ,  $\sin x$ ,  $\cos x$ ,  $x^2$ )
- **Output Layer:** Single-layer neuron with adjustable weights

Training is done using **LMS or gradient descent**, similar to linear adaptive filters.

## Merits of FLANN over Other ANNs

1. **Simple Structure:** No hidden layers, making the network easy to design and understand.
2. **Fast Learning:** Fewer parameters lead to faster convergence compared to multilayer networks.
3. **Lower Computational Cost:** Requires less memory and processing power.
4. **Avoids Local Minima:** Single-layer learning reduces chances of getting trapped in local minima.
5. **Effective Nonlinear Modeling:** Functional expansion enables nonlinear approximation without deep networks.

FLANN provides an efficient alternative to multilayer ANNs by achieving nonlinear learning through functional expansion. Its simplicity, speed, and stability make it suitable for real-time signal processing and adaptive system applications.

## c) What is Activation Function? Explain with a suitable example. [4]

An **activation function** is a mathematical function applied to the **net input of a neuron** in an Artificial Neural Network (ANN) to produce its output. It decides **whether a neuron should be activated or not** and introduces **non-linearity** into the network, enabling it to learn complex patterns.

## Purpose of Activation Function

- Converts weighted sum of inputs into meaningful output
- Introduces non-linearity in ANN
- Helps neural networks learn complex input–output relationships

**Example: Sigmoid Activation Function**

The **sigmoid function** is defined as:  $f(x) = \frac{1}{1 + e^{-x}}$

- Output range: (0, 1)
- Commonly used in binary classification
- Smooth and differentiable, suitable for backpropagation

**Working Example:**

If the net input to a neuron is  $x = 0$ ,

$$f(0) = \frac{1}{1 + e^0} = 0.5$$

This means the neuron is partially activated.

Activation functions play a crucial role in neural networks by controlling neuron output and enabling learning of non-linear decision boundaries. Without activation functions, a neural network would behave like a simple linear model.

➤ **NOV / DEC 2024**

Q7) a) Explain Recurrent Neural Networks with an example. [6] → DONE

b) What are different activation function used in NN? [6] → DONE

c) What is multilayer perceptron? Describe with diagram.[6] → DONE

Q8) a) Explain building blocks of RBF networks. [6] → DONE

b) What is personalized recommendation? What is content based recommendation? [6] → DONE

c) Explain the Convolution Neural Network (CNN) with suitable example.[6] → DONE

➤ **Additional NOV / DEC 2022 Questions:**

Q7) a) Explain ANN with it's Architecture. [5]

**1. Introduction to ANN**

- An **Artificial Neural Network (ANN)** is a computational model inspired by the structure and functioning of the **biological brain**.
- It consists of a large number of interconnected processing elements called **neurons** that work together to solve complex problems.
- ANNs are capable of **learning from data**, **generalizing**, and **recognizing patterns**.

**2. Basic Model of an Artificial Neuron**

- Each neuron receives multiple inputs.
- Each input is multiplied by a **weight**.
- A **weighted sum** is computed and passed through an **activation function** to produce the output.

**3. ANN Architecture (Structure):** The architecture of ANN is typically divided into **three layers**:

**a) Input Layer**

- Receives the raw data or features.
- No computation is performed here.
- Simply passes the input signals to the next layer.

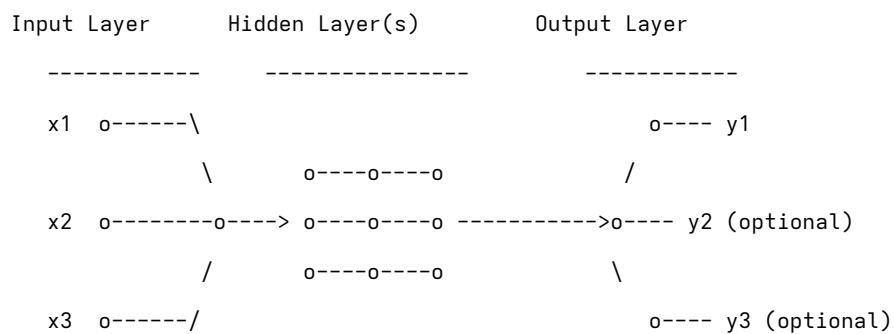
**b) Hidden Layer(s)**

- One or more intermediate layers between input and output.
- Each hidden neuron performs computation using:  
**Weighted Sum + Activation Function**
- Responsible for detecting patterns, relationships, and complex structures in data.
- More hidden layers → deeper network → more learning capability.

**c) Output Layer**

- Produces the final result of the network.
- Number of neurons depends on the type of task:
  - 1 neuron → binary classification
  - Multiple neurons → multi-class or regression output

**4. Diagram of ANN Architecture (Exam-friendly ASCII)**



- Circles represent **neurons**.
- Lines represent **weighted connections**.

**5. Activation Functions in ANN:** Common activation functions include:

- **Step function**
- **Sigmoid / Logistic**
- **Tanh**
- **ReLU**

These functions introduce **non-linearity**, enabling the network to learn complex patterns.

## 6. Learning Process

- ANNs learn by adjusting **weights** using training data.
- Common learning methods:
  - **Supervised learning** (Backpropagation algorithm)
  - **Unsupervised learning** (Hebbian rule, clustering)
- The objective is to minimize error between predicted output and actual output.

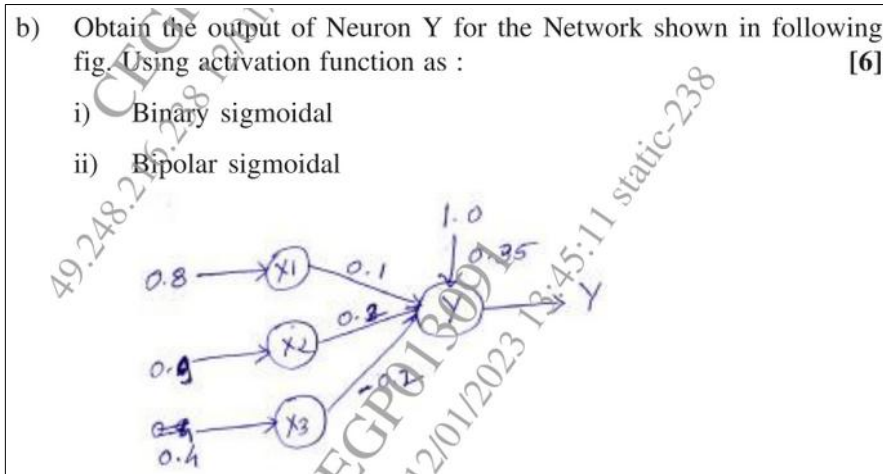
## 7. Applications of ANN

- Pattern recognition
- Image and speech processing
- Prediction and forecasting
- Medical diagnosis
- Robotics and control systems

**Conclusion:** An ANN is a powerful computational model with an architecture consisting of **input, hidden, and output layers**. Its ability to learn and generalize makes it suitable for solving complex real-world problems.

**Q7) b)** b) Obtain the output of Neuron Y for the Network shown in following fig. Using activation function as : [6]

- Binary sigmoidal
- Bipolar sigmoidal



Unverified answer

### 1. Given

- $x_1=0.8, w_1=0.1$
- $x_2=0.9, w_2=0.2$
- $x_3=0.4, w_3=-0.2$
- Bias  $b=1.0, w_b=0.35$



**2. Compute NET input to neuron Y**

$$\begin{aligned}
 NET &= x_1w_1 + x_2w_2 + x_3w_3 + b \cdot w_b \\
 &= (0.8)(0.1) + (0.9)(0.2) + (0.4)(-0.2) + (1)(0.35) \\
 &= 0.08 + 0.18 - 0.08 + 0.35 \\
 &= 0.53
 \end{aligned}$$

**NET = 0.53****3. (i) Binary sigmoid output**

$$\begin{aligned}
 \text{Binary sigmoid: } f(x) &= \frac{1}{1 + e^{-x}} \\
 Y &= \frac{1}{1 + e^{-0.53}} \approx 0.6295 \\
 \text{Binary Sigmoidal Output} &\approx 0.629 \text{ (approx.)}
 \end{aligned}$$

**4. (ii) Bipolar sigmoid output**

$$\begin{aligned}
 \text{Bipolar sigmoid (tanh): } f(x) &= \tanh(x) \\
 Y &= \tanh(0.53) \approx 0.4854 \\
 \text{Bipolar Sigmoidal Output} &\approx 0.485 \text{ (approx.)}
 \end{aligned}$$

**5. Final answers**

- **NET = 0.53**
- **Binary Sigmoid Output  $\approx 0.629$**
- **Bipolar Sigmoid Output  $\approx 0.485$**

**Q8) a) Explain in brief types of ANN based on layers. [5]****Types of ANN Based on Layers**

Artificial Neural Networks can be classified based on the **number and arrangement of layers** in their architecture.

**1. Single-Layer Feedforward Network**

- Consists of **one input layer** and **one output layer** only.
- No hidden layer present.
- Signals move in **one direction** (input  $\rightarrow$  output).
- Suitable for solving **linearly separable problems**.
- Example: **Perceptron**, Single-layer Adaline.

## 2. Multi-Layer Feedforward Network (MLP)

- Contains an **input layer**, **one or more hidden layers**, and an **output layer**.
- Information flows strictly forward (no feedback loops).
- Hidden layers allow learning of **complex, non-linear** patterns.
- Uses **backpropagation** for training.
- Widely used for classification, prediction, and recognition.

## 3. Recurrent Neural Network (RNN)

- Includes **feedback connections** where output of a neuron is sent back into the network.
- Allows the network to have **memory** of previous states.
- Useful for **sequential data** (time series, speech, text).
- Examples: Elman network, Jordan network.

## 4. Feedforward vs Feedback (Recurrent) Networks (General Classification)

- **Feedforward ANN:**  
Information moves from input → hidden → output only.  
No feedback loops.
- **Feedback / Recurrent ANN:**  
Neurons can connect back to previous layers, forming loops.  
Suitable for dynamic and time-dependent systems.

## 5. Modular / Hybrid Layered Networks (Optional mention for full marks)

- Combine multiple layered models (e.g., CNN + MLP).
- Each module handles a specific sub-task.
- Used in real-world applications like image recognition.

### c) Write short note on with reference with CNN. [6]

#### i) Convolution layer

#### ii) Hidden layer

**i) Convolution Layer (With Reference to CNN):** The **Convolution Layer** is the core building block of a **Convolutional Neural Network (CNN)**. It performs feature extraction from the input image using **filters/kernels**.

### 2. Working Principle

- Applies a small-sized **filter (e.g., 3×3, 5×5)** over the input image.
- Performs an operation called **convolution**:

$$\text{mapFeature map} = \text{Input} * \text{Kernel}$$

### 3. Purpose

- Detects **low-level features** such as edges, textures, corners.
- Deeper layers detect **high-level features** such as shapes, objects, and patterns.

#### 4. Key Components

- **Kernel/Filter:** learns pattern weights.
- **Stride:** step size of filter movement.
- **Padding:** keeps output size same (optional).
- **Feature Maps:** output of the convolution operation.

#### 5. Advantages

- Reduces computational cost compared to full connections.
- Automatically learns meaningful features without manual extraction.
- Provides spatial locality (neighboring pixels matter).

**ii) Hidden Layer (With Reference to CNN):** Hidden layers are the internal layers between input and output. In CNNs, hidden layers consist of **Convolution layers**, **Activation layers**, **Pooling layers**, and **Fully connected layers**.

#### 2. Purpose

- Transform extracted features into higher-level representations.
- Learn complex patterns required for classification or detection.

#### 3. Types of Hidden Layers in CNN

- **Convolution Layers:** extract features.
- **Activation Layers (ReLU):** introduce non-linearity.
- **Pooling Layers:** reduce dimensionality and increase robustness.
- **Fully Connected Layers:** perform final classification.

#### 4. Working

- Each hidden layer receives input from the previous layer, processes it, and passes the result forward.
- Hidden layers progressively convert raw pixels → edges → textures → shapes → object category.

#### 5. Importance

- Enable **deep learning** by stacking multiple transformations.
- Improves accuracy and generalization.
- Help the network learn hierarchical feature representation.

**Note: Please verify all answers before referring.**