Feedforward Neural Network

What is a Feedforward Neural Network?

A Feedforward Neural Network (FNN) is a type of artificial neural network where connections between the nodes do not form cycles. This characteristic differentiates it from recurrent neural networks (RNNs). The network consists of an input layer, one or more hidden layers, and an output layer. Information flows in one direction—from input to output—hence the name "feedforward."

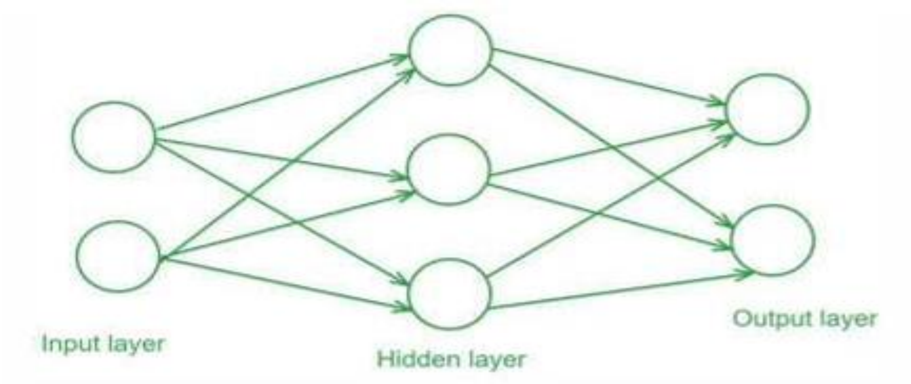
Structure of a Feedforward Neural Network

1. **Input Layer**: The input layer consists of neurons that receive the input data. Each neuron in the input layer represents a feature of the input data.

2. **Hidden Layers**: One or more hidden layers are placed between the input and output layers. These layers are responsible for learning the complex patterns in the data. Each neuron in a hidden layer applies a weighted sum of inputs followed by a non-linear activation function.

3. **Output Layer**: The output layer provides the final output of the network. The number of neurons in this layer corresponds to the number of classes in a classification problem or the number of outputs in a regression problem.

Each connection between neurons in these layers has an associated weight that is adjusted during the training process to minimize the error in predictions.



Feed Forward Neural Network

Applications of Feedforward Neural Networks (FNN):

1. **Image Classification**: Identifying objects or patterns within images.
2. **Medical Diagnosis**: Predicting diseases based on medical data and symptoms.
3. **Fraud Detection**: Recognizing fraudulent transactions in financial systems.
4. **Speech Recognition**: Converting audio input into text.
5. **Predictive Maintenance**: Forecasting equipment failures in industrial settings.

Convolutional Neural Network (CNN)

What is a Convolutional Neural Network?

A Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed for working with structured grid data, such as images. Unlike Feedforward Neural Networks, CNNs are able to learn spatial hierarchies from the data through local connections, shared weights, and pooling layers. CNNs are widely used in applications like image classification, object detection, and face recognition.

Structure of a Convolutional Neural Network

1. **Input Layer**: The input layer represents the raw pixel values of the image or grid data. If the input is a color image, this layer will have three channels (Red, Green, Blue) representing the color information.

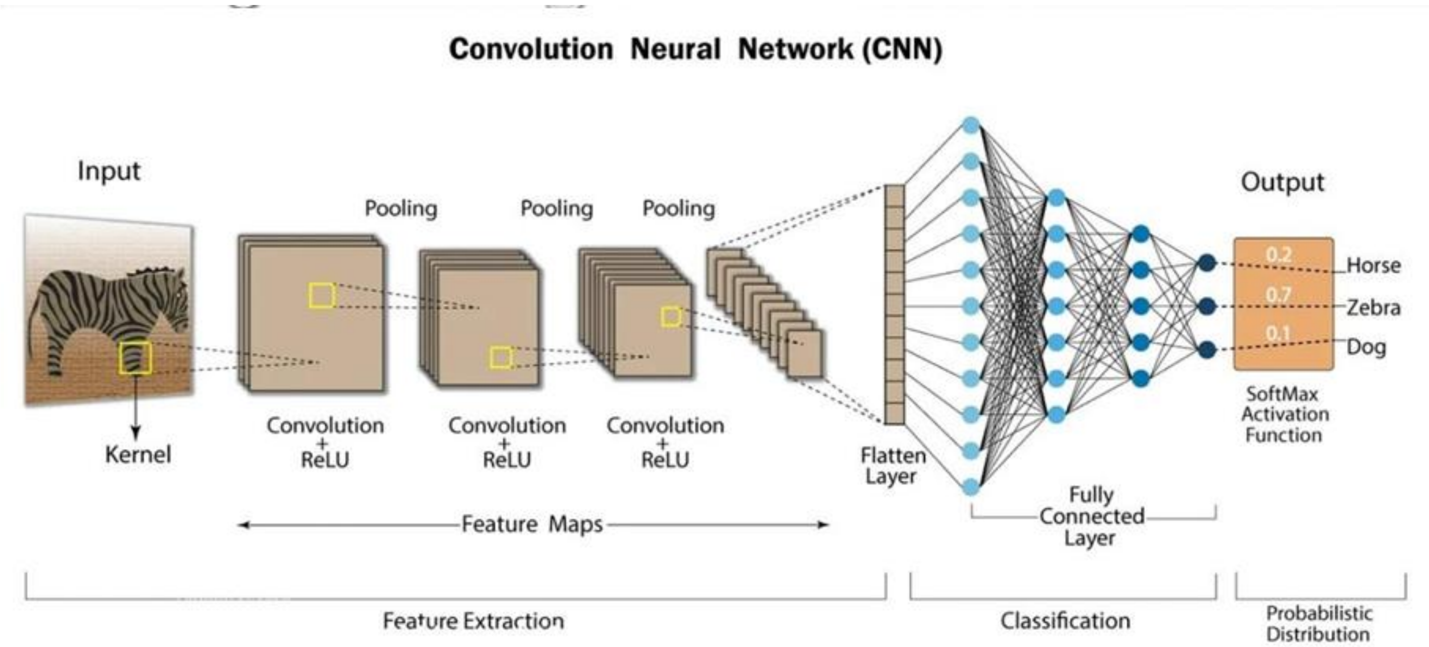
2. **Convolutional Layers**: The convolutional layers apply filters (or kernels) that slide over the input image to create feature maps. Each filter detects different features in the image, such as edges, textures, or shapes. These filters are learned during training and are essential in capturing spatial patterns. the filter kernel are smaller matrices usually of shape 3x3,4x4,5x5

3. **Activation Layers**: After convolution, each feature map typically passes through an activation function, usually a Rectified Linear Unit (ReLU), which introduces non-linearity. ReLU helps the model learn complex patterns by activating only certain neurons based on input values.

4. **Pooling Layers**: Pooling layers reduce the dimensionality of each feature map by down-sampling, often using Max Pooling, which takes the maximum value in a certain area. This process helps reduce the computational load, controls overfitting, and ensures that the CNN is invariant to small translations.

5. **Fully Connected (Dense) Layers**: Towards the end of the network, the feature maps are flattened and passed through one or more fully connected (dense) layers. These layers help combine the features learned by previous layers and produce a final output. The output layer usually has a softmax activation function for multi-class classification.

6. **Output layer:** Produces final predictions/classification. The output from fully connected layer is then fed into a logistic function for the classification taks like sigmoid/softmax which convert output of each class into probability score of each class



Applications of Convolutional Neural Networks

CNNs are widely used in various domains:

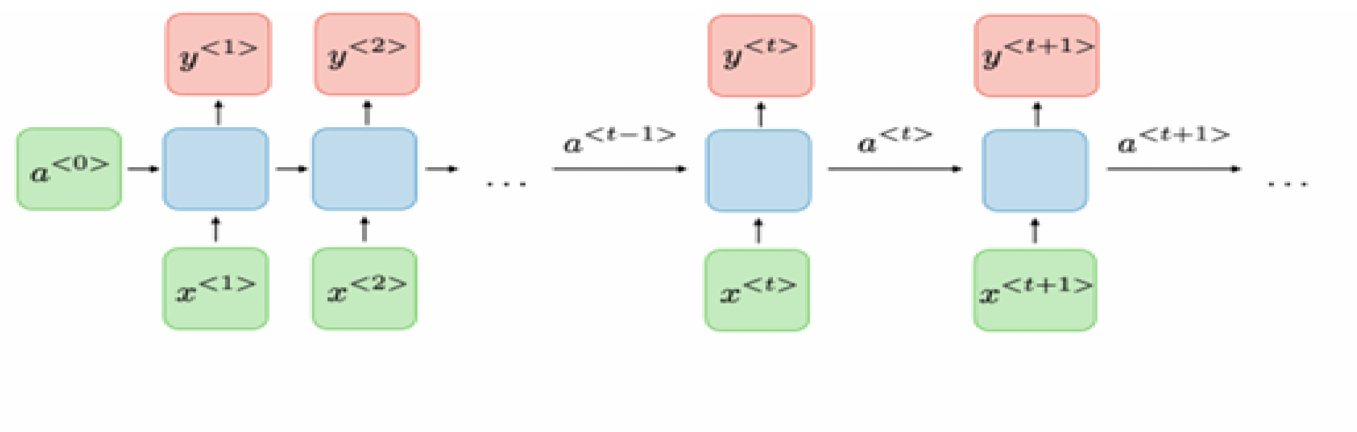
- **Image Classification**: Classifying images into categories, such as identifying animals, objects, or scenes.
- **Object Detection**: Detecting and locating multiple objects within an image.
- **Image Segmentation**: Dividing an image into segments for more granular classification.
- **Medical Imaging**: Assisting in diagnostics by identifying patterns in MRI, CT scans, and X-rays.

Recurrent Neural Network (RNN)

What is a Recurrent Neural Network?

A Recurrent Neural Network (RNN) is a type of artificial neural network designed for processing sequences of data by introducing connections that loop back, allowing information from previous time steps to influence the current step.

The idea was sharing parameters across different parts of a model. Parameter sharing makes it possible to extend and apply the model to examples of different forms (different lengths, here) and generalize across them. RNNs were the standard suggestion for working with sequential data before the advent of attention models. RNNs have a Memory which stores all information about the calculations. It employs the same settings for each input since it produces the same outcome by performing the same task on all inputs or hidden layers.



An RNN processes the sequence one element at a time, in the so-called time steps.

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Structure of a Recurrent Neural Network

1. **Input Layer**: The input layer receives sequential data one element at a time, such as words in a sentence or data points in a time series. Each input at time step t is processed in sequence.

2. **Recurrent (Hidden) Layer**: The hidden layer in an RNN has recurrent connections, meaning each hidden state depends not only on the current input but also on the hidden state from the previous time step. This allows information to flow from one time step to the next, giving RNNs a form of short-term memory. The most common activation function here is the Tanh or ReLU function.

3. **Output Layer**: The output layer generates predictions based on the final hidden state, which incorporates information from the entire sequence. The output may be a single value for the whole sequence (e.g., sentiment classification) or one output per time step (e.g., sequence generation tasks).

Applications of Recurrent Neural Networks

RNNs are widely used in tasks involving sequences:

- **Natural Language Processing**: Language modeling, text generation, and sentiment analysis.
- **Time Series Analysis**: Forecasting stock prices, weather prediction, and anomaly detection.
- **Speech Recognition**: Transcribing spoken words into text.

- **Machine Translation**: Translating text from one language to another by understanding sequence and context.