Feedforward Neural Network (FNN)

→ A feedforward Neural Network (FNN) is the most basic type of artificial neural Network.

→ In this network, the information moves in only one direction - forward - from the input nodes, through the hidden nodes (if any), and to the output nodes.

→ There are no cycles or loops in the network.

→ Feedforward neural network were the first type of artificial neural network invented and are simpler than their counterparts like recurrent neural networks and convolutional neural networks.

Structure of a Feedforward Neural Network

1) Input layer:

The input layer consist of neurons that receive the input data. Each neuron in the input layer represents a feature of the input data.

2) Hidden layers:

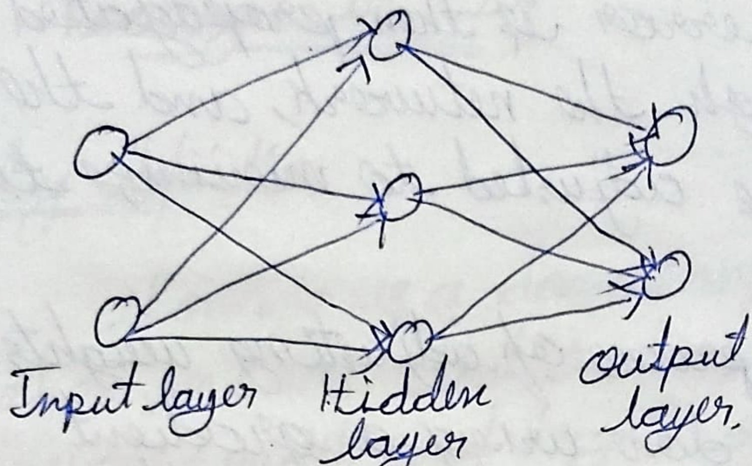
These layers are not exposed to the input or output and can be considered as the computational engine of the neural network.

Each hidden layers neurons take the weighted sum of the outputs from the previous layer, apply an activation function, and pass the result to the next layer. The network can have zero or more hidden layers.

3) Output layers:

→ The final layer that produces the output for the given inputs. The number of neurons in the output layer depends on the number of possible outputs the network is designed to produce.

{ ⁂ Each neuron in one layer is connected to every neuron in the next layer, making this a fully connected network. The strength of the connection between neurons is represented by weights, and learning in a neural network involves updating these weights based on the error of the output.



Working of Feedforward Neural Network.

The working of a feedforward neural network involves two phases:

i) The feedforward phase
ii) The backpropagation phase.

i) Feedforward phase.

In this phase, the input data is fed into the network, and it propagates forward through the network. At each hidden layer, the weighted sum of the inputs is calculated and passed through an activation function, which introduces non-linearity into the model. This process continues until the output layer is reached, and a prediction is made.

ii) Backpropagation phase:

Once a prediction is made, the error is calculated (difference b/w the predicted output & the actual output).

This error is then propagated back through the network, and the weights are adjusted to minimize this error.

The process of adjusting weights is typically done using a gradient descent optimization algorithm.

Applications of feedforward Neural Network:-

→ Image classification
→ Speech Recognition
→ Regression tasks
→ Basic NLP tasks.

Advantages of feedforward Neural Network:

→ Simplicity
→ General-Purpose
→ Deterministic output.

Disadvantages

→ Overfitting
→ Lack of Memory
→ Vanishing Gradients.
→ Difficulty in handling complex data

Pooling in CNN

Pooling definition:-

→ Pooling is a downsampling operation in CNNs that reduces the spatial dimensions (width and height) of the feature maps generated by the convolutional layers.

→ It helps to condense the information and reduce the computational complexity of the network.

Purpose:

→ **Reduces Dimensionality:**
Decreases the number of parameters and computations in the network, making it more efficient.

→ **Controls Overfitting:**
By reducing the spatial size, it also helps prevent overfitting.

→ **Invariance to translation:**
Makes the detection of features more robust to small changes or shifts in the input image.

Types of Pooling

i) Max pooling

→ Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.

→ Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

$\xrightarrow{\text{Max pool}}$ filter-(2x2) stride-(2,2)

9	7
8	6

ii) Average pooling

→ Average pooling computes the averages of the elements present in the region of feature map covered by the filter.

→ Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of feature present in a patch.

$$\begin{bmatrix} 2 & 2 & 7 & 3 \\ 9 & 4 & 6 & 1 \\ 8 & 5 & 2 & 4 \\ 3 & 1 & 2 & 6 \end{bmatrix} \xrightarrow[\substack{\text{filter-}(2\times2) \\ \text{stride-}(2,2).}]{\text{Average Pool.}} \begin{bmatrix} 4.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$

iii) Global pooling:-

→ Global pooling applies pooling over the entire feature map, resulting in a single value per feature map.

→ Thus, an $n_h \times n_w \times n_c$ feature map is reduced to $1 \times 1 \times n_c$ feature map. This is equivalent to using a filter of dimensions $n_h \times n_w$, the dimensions of the feature map.

→ There are two types in global pooling:-

i) Global Max pooling

ii) Global Average pooling.

Pooling parameters:

→ Pool size: The size of the window, (e.g. 2x2, 3x3) that moves across the feature map to perform pooling.

→ stride: The number of pixels by which the window moves. A stride of 2, for example moves the window 2 pixels at a time, reducing the output size by half.

Advantages of pooling:-

→ Reduces Overfitting.

→ Reduces computational load.

→ Invariance

Disadvantages of pooling:-

→ Loss of Information.

→ Pooling window selection.

Pooling is an essential operation in CNNs that balances the network's ability to capture important features while maintaining computational efficiency and reducing overfitting.

Regularization

Definition:-

→ Regularization is a technique used to prevent overfitting in machine learning models by adding a penalty to the loss function for large coefficients or weights.

→ It helps to ensure that the model generalizes well to new, unseen data rather than just memorizing the training data.

Why regularization is needed:-

→ **Overfitting:-** Occurs when a model performs well on the training data but poorly on the test data due to capturing noise or irrelevant patterns.

→ **Complex Models:-** Models with too many parameters (eg: deep neural networks) can easily overfit without regularization.

Types of regularization

(*) → Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding Overfitting.*

→ The commonly used regularization techniques are:-

1) Lasso Regularization - (L1)
2) Ridge Regularization - (L2)
3) Elastic Net Regularization
   ↳ L1 and L2 Regularization.

1) Lasso Regression.

→ A regression model which uses the L1 regularization technique is called LASSO (Least Absolute Shrinkage and Selection Operator) regression.

→ Lasso regression adds the "absolute value of magnitude" of the coefficient as a penalty term to the loss function (L).

$$\boxed{\text{cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}|w_i|}$$

Where,
m - Number of features
n - Number of Examples
y_i = Actual Target value
$\hat{y}_i$ = Predicted target value.

Ridge regression:-

→ A regression model that uses the L2 regularization technique is called ridge regression.

→ Ridge regression adds the "Squared magnitude" of the coefficient as a penalty term to the loss function (L).

$$\boxed{cost = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m} w_i^2}$$

Elastic Net Regression:

→ This model is a combination of L1 and L2 regularization. That implies that we add the absolute norm of the weights as well as the squared measure of the weights, with the help of an extra hyperparameter that controls the ratio of the L1 and L2 regularization.

$$\boxed{cost = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\left((1-\alpha)\sum_{i=1}^{m}|w_i| + \alpha\sum_{i=1}^{m} w_i^2\right)}$$

Advantages of Regularization

14
$7 \overline{)100}$

→ Reduces overfitting
→ Improves Generalization.
→ Feature selection (L1)

Disadvantages of Regularization

→ Bias-Variance tradeoff
→ Hyperparameter tuning.
→ Complexity in Interpretation

Applications of Regularization

→ Linear and logistic Regression.
→ Neural Networks.
→ Support Vector Machines (SVM).

Regularization is a fundamental concept in ML that helps models generalize better by penalizing complexity, thereby preventing overfitting and ensuring that the model performs well on new data.