Important Questions – NNFL

1. Differentiate between ML & Deep Learning

Machine Learning	Deep Learning
Machine Learning is a superset of Deep Learning	Deep Learning is a subset of Machine Learning
The data represented in Machine Learning is quite different compared to Deep Learning as it uses structured data	The data representation used in Deep Learning is quite different as it uses neural networks(ANN).
Machine Learning is an evolution of AI.	Deep Learning is an evolution of Machine Learning. Basically, it is how deep is the machine learning.
Machine learning consists of thousands of data points.	Big Data: Millions of data points.
Outputs: Numerical Value, like classification of the score.	Anything from numerical values to free-form elements, such as free text and sound.
Uses various types of automated algorithms that turn to model functions and predict future action from data.	Uses a neural network that passes data through processing layers to, interpret data features and relations.

Algorithms are detected by data analysts to examine specific variables in data sets.	Algorithms are largely self-depicted on data analysis once they're put into production.
Machine Learning is highly used to stay in the competition and learn new things.	Deep Learning solves complex machine-learning issues.
Training can be performed using the CPU (Central Processing Unit).	A dedicated GPU (Graphics Processing Unit) is required for training.
More human intervention is involved in getting results.	Although more difficult to set up, deep learning requires less intervention once it is running.
Machine learning systems can be swiftly set up and run, but their effectiveness may be constrained.	Although they require additional setup time, deep learning algorithms can produce results immediately (although the quality is likely to improve over time as more data becomes available).
Its model takes less time in training due to its small size.	A huge amount of time is taken because of very big data points.
Humans explicitly do feature engineering.	Feature engineering is not needed because important features are automatically detected by neural networks.

Machine learning applications are simpler compared to deep learning and can be executed on standard computers.	Deep learning systems utilize much more powerful hardware and resources.
The results of an ML model are easy to explain.	The results of deep learning are difficult to explain.
Machine learning models can be used to solve straightforward or a little bit challenging issues.	Deep learning models are appropriate for resolving challenging issues.
Banks, doctor's offices, and mailboxes all employ machine learning already.	Deep learning technology enables increasingly sophisticated and autonomous algorithms, such as self-driving automobiles or surgical robots.
Machine learning involves training algorithms to identify patterns and relationships in data.	Deep learning, on the other hand, uses complex neural networks with multiple layers to analyze more intricate patterns and relationships.
Machine learning algorithms can range from simple linear models to more complex models such as decision trees and random forests.	Deep learning algorithms, on the other hand, are based on artificial neural networks that consist of multiple layers and nodes.
Machine learning algorithms typically require less data than deep learning algorithms, but the quality of the data is more important.	Deep learning algorithms, on the other hand, require large amounts of data to train the neural networks but can learn and improve on their own as they process more data.
Machine learning is used for a wide range of applications, such as regression, classification, and clustering.	Deep learning, on the other hand, is mostly used for complex tasks such as image and speech recognition, natural language processing, and autonomous systems.
Machine learning algorithms for complex tasks, but they can also be more difficult to train and may require more computational resources.	Deep learning algorithms are more accurate than machine learning algorithms.

Example: Predicting house prices using features like the size of the house, number of rooms, and location. In this case, a machine learning algorithm like linear regression is trained to predict the price based on these inputs.	**Example:** In a deep learning model for image recognition, a convolutional neural network (CNN) can automatically learn to identify features like edges, shapes, and objects from raw pixel data to classify the image (e.g., dog vs. cat) without manual feature engineering.

2. Explain the working of Deep Learning Model in brief.

Deep learning is a subset of machine learning and artificial intelligence (AI) that mimics how a human brain functions, and it allows computers to address complex patterns that create new insights and solutions. If you've used technology like a digital assistant on your phone, received a text alerting you of credit card fraud, or ridden in a self-driving car, you've used deep learning.

What is a deep learning model?

Deep learning models are complex networks that learn independently without human intervention. It applies algorithms to immense data sets to find patterns and solutions within the information. Deep learning models typically have three or more layers of neural networks to help process data. These models have the ability to process data that's unstructured or unlabeled, creating their own methods for identifying and understanding the information without a person telling the computer what to look for or solve.

Because deep learning models can identify both higher-level and lower-level information, they can take difficult-to-understand data sets and create simpler, more efficient categories. This ability allows the deep learning model to grow more accurate over time.

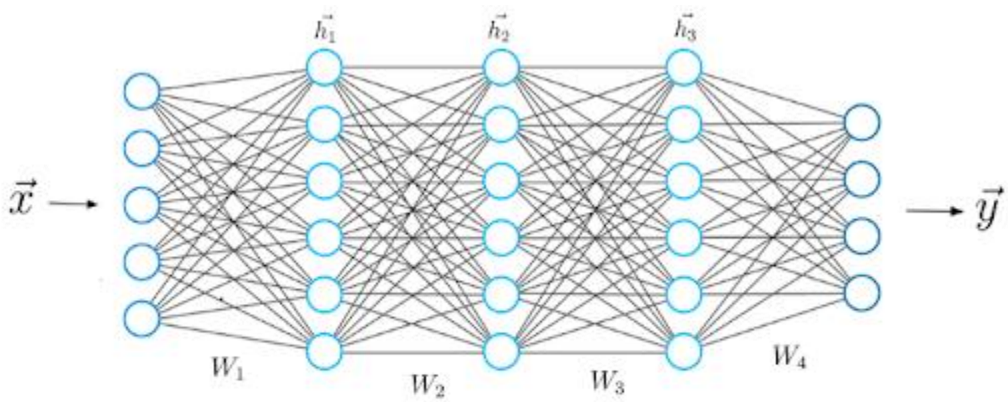
Types of deep learning models

Deep learning models use a variety of constructions and frameworks to achieve specific tasks and goals. Some types of deep learning models include:

- **Convolutional neural networks:** You can use convolutional neural networks for image processing and recognition.
- **Recurrent neural networks:** You can use recurrent neural networks for speech recognition and natural language processing.
- **Long short-term memory networks**: You can use long short-term memory networks for sequential prediction tasks, such as language modeling.

How Does Deep Learning Work?

Deep learning algorithms attempt to draw similar conclusions as humans would by constantly analyzing data with a given logical structure. To achieve this, deep learning uses a multi-layered structure of algorithms called neural networks.



1. **Neural Network Architecture:**
    - **Layers:** A neural network consists of multiple layers of interconnected nodes, often referred to as neurons.
    - **Input Layer:** This layer receives the raw data (e.g., images, text, audio).
    - **Hidden Layers:** These layers process the input data and extract relevant features. The number of hidden layers and neurons in each layer determines the model's complexity.
    - **Output Layer:** This layer produces the final prediction or decision.
2. **Training Phase:**
    - **Data Preparation:** The model is fed a large dataset with labeled examples.
    - **Forward Propagation:** The input data is passed through the network, activating neurons in each layer.
    - **Error Calculation:** The difference between the model's predicted output and the actual target output is calculated.
    - **Backpropagation:** This process adjusts the weights and biases of the neurons to minimize the error. The model learns to identify patterns and correlations in the data.
3. **Testing Phase:**
    - Once trained, the model can be used to make predictions on new, unseen data.
    - The input data is fed through the network, and the output layer produces the prediction.

Key Concepts:

- **Weights and Biases:** These parameters determine the strength of connections between neurons and influence the model's output.
- **Activation Functions:** These functions introduce non-linearity, allowing the model to learn complex patterns.
- **Loss Function:** This function quantifies the model's error during training.
- **Optimization Algorithms:** These algorithms (e.g., gradient descent) are used to update the weights and biases to minimize the loss.

Applications: Deep learning has revolutionized various fields, including:

- **Image recognition:** Identifying objects, facial recognition
- **Natural language processing:** Machine translation, sentiment analysis
- **Speech recognition:** Converting spoken language into text
- **Recommendation systems:** Suggesting products or content
- **Medical diagnosis:** Analyzing medical images and data

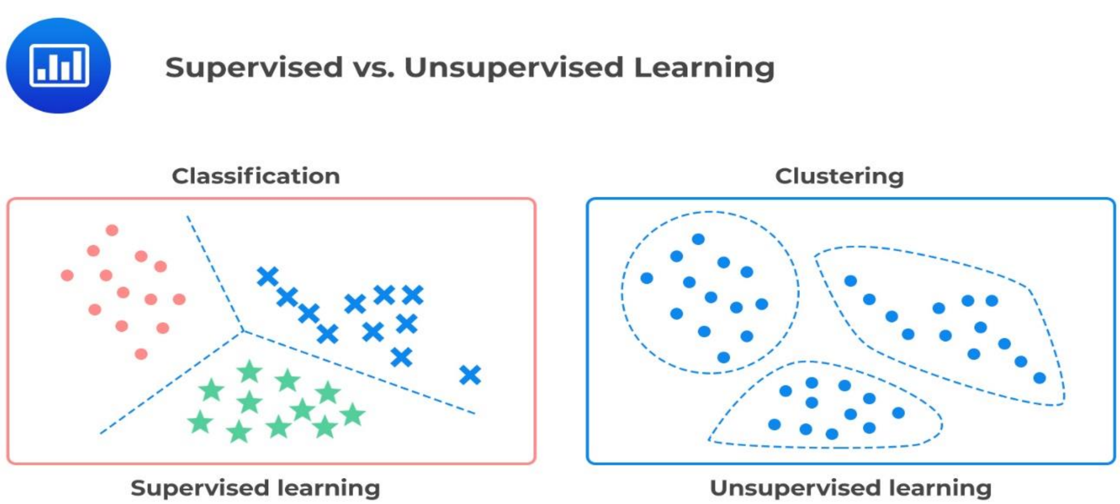
3. Differentiate Classification & Regression with suitable example.

Classification	Regression
In this problem statement, the target variables are discrete.	In this problem statement, the target variables are continuous.
Problems like Spam Email Classification, Disease prediction like problems are solved using Classification Algorithms.	Problems like House Price Prediction, Rainfall Prediction like problems are solved using regression Algorithms.
In this algorithm, we try to find the best possible decision boundary which can separate the two classes with the maximum possible separation.	In this algorithm, we try to find the best-fit line which can represent the overall trend in the data.
Evaluation metrics like Precision, Recall, and F1-Score are used here to evaluate the performance of the classification algorithms.	Evaluation metrics like Mean Squared Error, R2-Score, and MAPE are used here to evaluate the performance of the regression algorithms.
Here we face the problems like binary Classification or Multi-Class Classification problems.	Here we face the problems like Linear Regression models as well as non-linear models.

Input Data are Independent variables and categorical dependent variable.	Input Data are Independent variables and continuous dependent variable.
The classification algorithm's task mapping the input value of x with the discrete output variable of y.	The regression algorithm's task is mapping input value (x) with continuous output variable (y).
Output is Categorical labels.	Output is Continuous numerical values.
Objective is to Predict categorical/class labels.	Objective is to Predicting continuous numerical values.
Example use cases are Spam detection, image recognition, sentiment analysis	Example use cases are Stock price prediction, house price prediction, demand forecasting.
Examples of classification algorithms are: Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, Neural Networks, K-Means Clustering, Multi-layer Perceptron (MLP), etc.	**Examples of regression algorithms are:** Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Decision Trees for Regression, Random Forest Regression, K-Nearest Neighbors (K-NN) Regression, Neural Networks for Regression, etc.

4. Differentiate Supervised & Unsupervised with an example.

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.	Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.



5. Explain Neural Networks in Detail.

A **Neural Network** is a computational model inspired by the way biological neural networks in the human brain function. It is the foundation of many machine learning techniques, especially **Deep Learning**. Neural networks consist of interconnected units called **neurons** (or nodes) that work together to solve complex tasks by learning patterns from data.

Here's a detailed breakdown of neural networks:

1. Components of a Neural Network:

- **Neuron (Node):**
  Each neuron is the basic unit of a neural network. It receives input from other neurons or external sources, processes it, and generates an output. Mathematically, a neuron calculates a weighted sum of its inputs, adds a bias, and applies an activation function to produce the output.

  Mathematical Formulation of a Neuron:

  $$z = (w_1x_1 + w_2x_2 + \ldots + w_nx_n) + b$$

  Where:

- x1,x2,...,xn are inputs.
- w1,w2,...,wn are weights associated with those inputs.
- b is the bias.
- The final output is passed through an **activation function** to introduce non-linearity.

- **Weights (w):**
  Weights are the adjustable parameters that determine the strength of the connection between neurons. During training, these weights are optimized to minimize error and improve the network's predictions.
- **Bias (b):**
  Bias is an additional parameter in the neuron that allows the model to fit data better. It shifts the activation function, helping to adjust the output along with the weighted sum.
- **Activation Function:**
  The activation function determines the output of a neuron based on its input. It introduces non-linearity into the network, enabling it to model complex patterns. Common activation functions include:
  - **Sigmoid:** Produces an output between 0 and 1. Typically used in binary classification.

    $$\text{Sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

- **ReLU (Rectified Linear Unit):** Outputs 0 for negative inputs and the input itself for positive inputs.

  ReLU(z)=max(0,z)

- **Tanh:** Similar to sigmoid but outputs values between -1 and 1.

$$\mathrm{Tanh}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

2. Structure of a Neural Network:

Neural networks are typically organized in layers:

- **Input Layer:**
  This layer receives the input data. Each neuron in the input layer represents one feature of the data. For example, in an image, each pixel might be a feature. There is no computation in the input layer; it just passes the data to the next layer.
- **Hidden Layers:**
  These are the layers between the input and output. Each hidden layer consists of neurons that receive input from the previous layer, perform computations (weighted sums and activation functions), and pass the results to the next layer. The more hidden layers a network has, the deeper it is, allowing it to learn more complex representations of data.
- **Output Layer:**
  The final layer produces the output prediction. For classification tasks, the output might be a probability distribution across multiple classes (using the softmax function). For regression tasks, it might be a continuous value.

3. Working of a Neural Network:

The process of training a neural network involves two key phases: **Forward Propagation** and **Backpropagation**.

a. Forward Propagation:

- In forward propagation, data passes through the network from the input layer to the output layer.
- At each layer, the input data is multiplied by the weights, a bias is added, and an activation function is applied to produce an output, which serves as input to the next layer.
- The final output is compared with the target value using a **loss function** to measure the error.

b. Loss Function:

- The **loss function** measures how well the network's predictions match the actual data.
- Common loss functions include:
    - **Mean Squared Error (MSE):** Used in regression tasks to measure the difference between predicted and actual values.
    - **Cross-Entropy Loss:** Used in classification tasks to quantify the difference between predicted and true class probabilities.

c. Backpropagation and Optimization:

- After calculating the loss, the network updates its weights to minimize this error through a process called **backpropagation**.
- In backpropagation, the error is propagated backward through the network from the output to the input layer. The network calculates the gradient of the loss function with respect to each weight using the **chain rule** of calculus.
- These gradients indicate how the weights should be adjusted to reduce the error.

The weight updates are performed using an optimization algorithm like:

- **Stochastic Gradient Descent (SGD):** Updates the weights by subtracting a fraction of the gradient (called the learning rate).
- **Adam Optimizer:** A more advanced optimizer that adapts the learning rate based on gradient history.

This process is repeated over many iterations (epochs) until the model reaches an optimal level of accuracy.

4. Types of Neural Networks:

There are various types of neural networks, each suited for different kinds of tasks:

a. Feedforward Neural Network (FNN):

- The simplest type of neural network where connections between nodes do not form cycles. Data moves in only one direction, from input to output.
- Used for tasks like image classification or simple regression.

b. Convolutional Neural Networks (CNNs):

- A specialized type of neural network designed for **image processing** tasks. CNNs use **convolutional layers** to detect patterns in images, such as edges, textures, and objects.
- Convolutional layers reduce the dimensionality of images while preserving spatial relationships.

c. Recurrent Neural Networks (RNNs):

- RNNs are designed for **sequential data** like time series or natural language. They have feedback connections that allow them to maintain information from previous inputs.
- However, RNNs suffer from issues like vanishing gradients, so more advanced versions like **Long Short-Term Memory (LSTM)** networks and **Gated Recurrent Units (GRUs)** are often used.

d. Generative Adversarial Networks (GANs):

- GANs consist of two networks: a **generator** that creates new data samples, and a **discriminator** that tries to distinguish between real and generated data.
- These networks are commonly used for generating realistic images, videos, or even art.

e. Autoencoders:

- Autoencoders are used for **unsupervised learning** tasks like data compression and feature extraction. They work by encoding input data into a smaller representation and then reconstructing it from that encoding.

5. Neural Network Training Process:

1. **Initialization:**
   The weights and biases are initialized randomly or using specific techniques like He initialization for deeper networks.
2. **Forward Pass:**
   Data flows through the network to produce predictions.
3. **Loss Calculation:**
   The loss is computed based on the difference between the predicted output and the actual output.
4. **Backpropagation:**
   Gradients of the loss function with respect to weights are calculated.
5. **Weight Update:**
   The weights are updated using an optimizer like SGD or Adam to minimize the loss.
6. **Iteration:**
   The process is repeated for many epochs until the network converges to an optimal solution.

6. Applications of Neural Networks:

Neural networks have diverse applications in various fields, including:

- **Computer Vision:** Image classification, object detection, facial recognition.

- **Natural Language Processing (NLP):** Text translation, sentiment analysis, chatbot development.
- **Healthcare:** Disease prediction, medical image analysis.
- **Finance:** Fraud detection, stock market prediction.
- **Autonomous Systems:** Self-driving cars, robotics.

7. Challenges in Neural Networks:

- **Vanishing/Exploding Gradients:** In deep networks, gradients can become too small (vanishing) or too large (exploding), making training difficult.
- **Overfitting:** If a model is too complex, it can perform well on training data but fail to generalize to new data.
- **Data Requirements:** Neural networks require large amounts of data and computational resources to train effectively.
- **Interpretability:** Neural networks are often considered "black boxes," meaning it is difficult to understand how they make decisions.

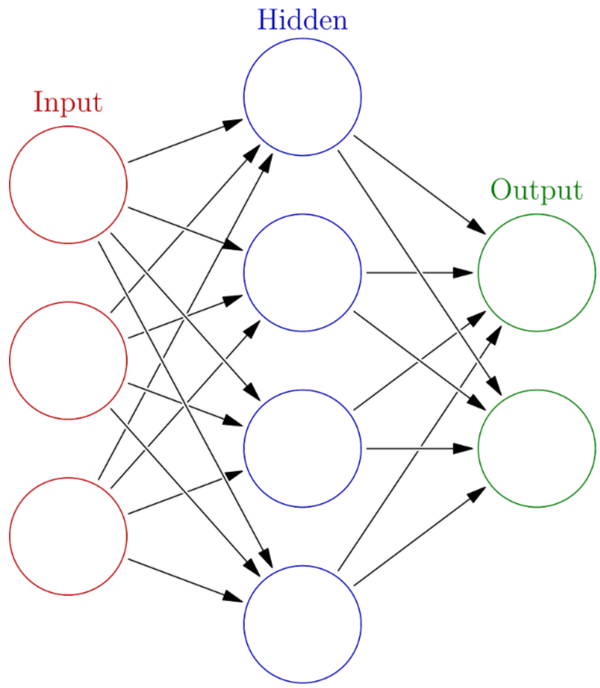
6. Explain the Structure of Neural Network.

A neural network is composed of layers of interconnected nodes (neurons) organized into three primary types of layers: the input layer, hidden layers, and the output layer.

Input Layer: The input layer consists of neurons representing the features of the input data. Each neuron corresponds to a feature, and its value represents the feature’s value.

Hidden Layers: Between the input and output layers, there may be one or more hidden layers. These layers perform complex computations on the input data. Each neuron in a hidden layer receives inputs from all neurons in the previous layer, applies a weighted sum, adds a bias term, and passes the result through an activation function.

Output Layer: The output layer produces the final prediction or result. The number of neurons in this layer depends on the nature of the problem. For example, in a binary classification task, there might be one neuron for each class, outputting probabilities.



Connection Weights and Activation Functions:

- **Connection Weights (W):** These represent the strengths of connections between neurons. Each connection from neuron A to neuron B has a weight associated with it, denoted as WAB. These weights are learned during training and determine the impact of neuron A's output on neuron B.
- **Bias Terms (b):** Each neuron also has a bias term (b) associated with it. The bias allows the neuron to shift its output. Bias terms are also learned during training.
- **Activation Function (σ):** Each neuron applies an activation function to the weighted sum of its inputs plus the bias. Common activation functions include the sigmoid function, ReLU (Rectified Linear Unit), and tanh (hyperbolic tangent). The activation function introduces non-linearity, allowing the network to model complex relationships.

Forward Pass (Inference):

During the forward pass (inference), data is propagated through the network as follows:

1. **Input Propagation:** Input data is assigned to the input layer's neurons.
2. **Hidden Layer Computation:** Each neuron in the hidden layers computes the weighted sum of its inputs and adds its bias term:

    - **Weighted Sum (Z):** $Z = \Sigma(W_{ij} * X_j) + b_i$, where W_{ij} is the weight, X_j is the output of the j-th neuron in the previous layer, and b_i is the bias of the current neuron.
    - **Activation (A):** $A = \sigma(Z)$, where σ is the activation function.

1. **Output Layer Computation:** The output layer neurons perform the same computation as the hidden layers, producing the final predictions or values.

Loss Function:

After the forward pass, the network’s output is compared to the actual target values using a loss function (also called a cost function or objective function). The choice of loss function depends on the task, but common ones include mean squared error (MSE) for regression tasks and cross-entropy for classification tasks.

Backward Pass (Backpropagation):

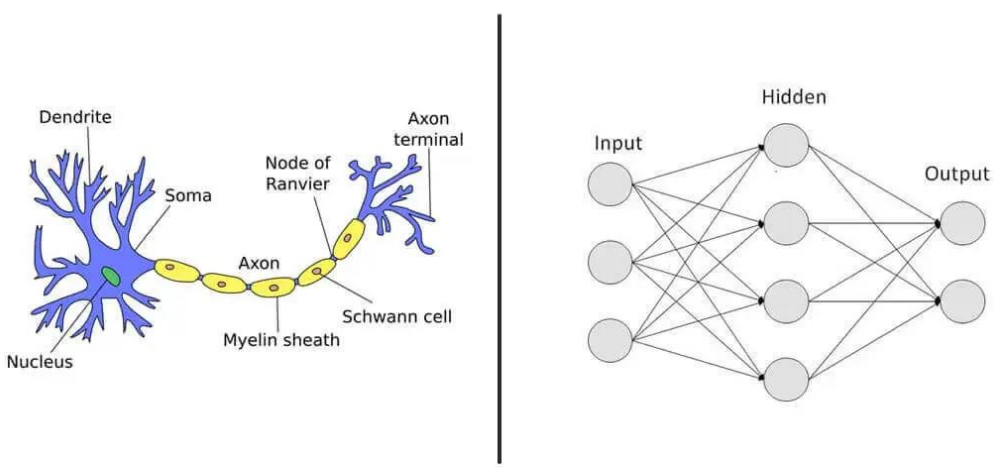
During the backward pass (backpropagation), the network learns from its mistakes and updates its weights to minimize the loss. The steps involved are:

1. **Gradient Computation:** Calculate the gradients of the loss with respect to the weights and biases using the chain rule of calculus. The gradient represents how much a change in a weight will affect the loss.
2. **Weight Updates:** Adjust the weights and biases in the direction that minimizes the loss by subtracting the gradient multiplied by a learning rate (α). This update step follows an optimization algorithm like stochastic gradient descent (SGD).

Optimization:

Optimization algorithms like SGD, Adam, or RMSprop are used to efficiently update weights and biases during training. Learning rate (α) controls the size of weight updates, and other hyperparameters can be fine-tuned to achieve better convergence.

7. Explain Neural Networks and how it is inspired by the Human Brain.



Neural Networks and How They Are Inspired by the Human Brain

Human Brain Overview:

The human brain is an intricate and powerful organ consisting of approximately **86 billion neurons**, interconnected by **trillions of synapses**. This complex network is responsible for everything from basic reflexes to high-level reasoning, learning, and memory. Neurons are the fundamental units of the brain and nervous system that process and transmit information through both **electrical** and **chemical signals**.

Neural Communication in the Brain:

1. **Synapses**: These are the junctions where neurons communicate with each other. A neuron sends signals via its **axon** to the dendrites or cell body of another neuron across the synaptic gap.
2. **Transmission**: Communication occurs through **action potentials** (electrical impulses) that travel along the axon, and **neurotransmitters** (chemical messengers) that cross the synaptic gap.
3. **Example (Reflex Action)**: When you touch a hot object, sensory neurons rapidly transmit signals to your brain and spinal cord. These signals trigger a rapid response, causing you to pull your hand away before you even become consciously aware of the pain.

Key Components of Biological Neurons:

1. **Dendrites**: Branch-like extensions that receive incoming signals from other neurons.
2. **Cell Body (Soma)**: Contains the nucleus and integrates the incoming signals from the dendrites.
3. **Axon**: A long, thin structure that carries electrical signals away from the soma to other neurons or muscles.
4. **Synapse**: The gap between neurons where neurotransmitters carry signals to the next neuron.
5. **Action Potentials**: These are the electrical impulses that travel down the axon, enabling communication between neurons.
6. **Synaptic Transmission**: Neurotransmitters are released from the axon terminals, cross the synaptic gap, and bind to receptors on the dendrites of the receiving neuron.
7. **Example (Learning)**: Repeated activation of specific synapses strengthens them, forming the basis of learning and memory. This is known as **synaptic plasticity**.

Inspiration from Biology:

Artificial Neural Networks (ANNs) are inspired by the basic structure and function of biological neurons. However, they are simplified mathematical models designed to perform tasks like pattern recognition, classification, and prediction. Here’s how they are related:

1. **Learning**: Both biological and artificial neural networks learn from experience or data. In biological systems, learning occurs through synaptic strengthening, while in ANNs, learning occurs by adjusting the weights between artificial neurons.
2. **Connections**: In both systems, information flows through networks of interconnected nodes (neurons in the brain, artificial neurons in ANNs). These connections allow complex computations to emerge from simpler elements.
3. **Adaptability**: Both biological and artificial neural networks can adapt over time. The brain adapts through processes like synaptic plasticity and neurogenesis, while ANNs adapt through techniques like backpropagation and gradient descent.

Key Differences:

1. **Complexity**: Biological neural networks are far more complex, with a vast number of neurons, synapses, and specialized regions performing highly efficient parallel computations. ANNs, while powerful, are much simpler in comparison and have limitations in both capacity and efficiency.
2. **Learning Speed**: Artificial neural networks can be trained rapidly using large datasets and computational power, whereas biological learning involves slower, more intricate processes, including emotional and cognitive factors.
3. **Example (Pattern Recognition)**: Just as the human brain can recognize faces or objects in its environment, ANNs can be trained to identify patterns in data, such as recognizing handwritten digits (e.g., in the MNIST dataset). This similarity allows ANNs to excel in areas like image recognition, natural language processing, and speech recognition.

Biological and Artificial Neural Networks – A Comparison:

Aspect	Biological Neural Networks	Artificial Neural Networks (ANNs)
Neuron Structure	Neurons with dendrites, axons, and synapses	Simplified units with inputs, weights, and activation functions
Learning Mechanism	Synaptic plasticity, neurogenesis	Adjusting weights through training (e.g., backpropagation)
Communication	Electrochemical signals via action potentials and neurotransmitters	Mathematical functions and weighted sums
Processing Power	Vast, parallel processing with efficiency	Dependent on computational resources (often slower but can scale)
Adaptability	Highly adaptable to complex, dynamic environments	Learns from data, but less flexible outside its training scope

In summary, while artificial neural networks are modeled after biological neural networks, they represent only a simplified abstraction of how the brain functions. Despite the differences, the fundamental principles of learning, adaptation, and pattern recognition unite both systems, making ANNs a powerful tool in fields such as artificial intelligence and machine learning.

8. What is the main difference between DFF & RNN.

Aspect	Dense Feedforward Networks (DFF)	Recurrent Neural Networks (RNN)
Definition	A type of neural network where information moves in one direction, from input to output, without loops or cycles.	A type of neural network designed to handle sequential data by allowing loops or cycles, enabling the retention of information over time.
Architecture	Unidirectional: Input flows in one direction (input to output).	Cyclic: Has loops allowing information to persist, creating a feedback mechanism.
Memory	No memory: Cannot retain information from previous inputs.	Has memory: Maintains a hidden state to remember information over time steps.
Data Dependency	Treats each input independently, with no relation to previous inputs.	Processes sequences, where each input is related to previous inputs (time or sequence-based).
Use Cases	Suitable for tasks where input data is independent (e.g., image classification).	Suitable for sequential data tasks like time-series forecasting, speech recognition, and language modeling.
Time Dynamics	Static: No concept of time or sequential order.	Dynamic: Processes data in sequence, handling time dependencies.
Training Complexity	Simpler to train due to lack of dependencies between inputs.	More complex to train due to sequence dependencies and the vanishing gradient problem.
Hidden State	No hidden state: Each layer computes independently.	Has a hidden state that carries information across time steps or layers.
Input Type	Works well with fixed-size inputs (e.g., images, tabular data).	Works well with variable-length sequential data (e.g., text, time-series).
Output	Produces output only after the entire network is processed.	Produces output at each time step or after the entire sequence.
Common Applications	Image classification, object detection, regression tasks.	Language translation, sentiment analysis, video processing, and sequential predictions.