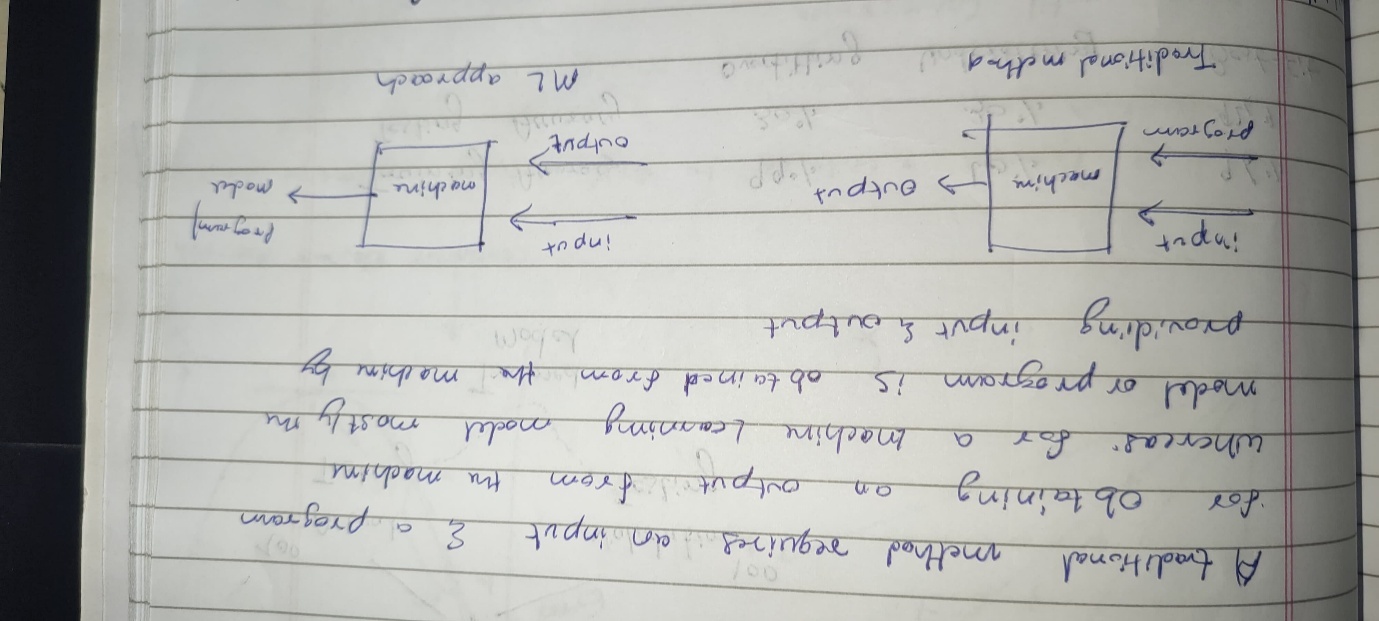
**Deep Learning**

**1.Machine learning:**Machine learning is a “*Field of study that gives computers the ability to learn without being explicitly programmed.*” In other words it is concerned with the question of how to construct computer programs that automatically improve with the experience.– Arthur Samuel

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E – Tom Mitchell

**Difference b/w traditional programming and machine learning .**



**2. Different types of machine learning**

Supervised learning: Machine learning that are designed to learn by examples, i.e., It maps the input to an output based on previous input-output pairs. It is trained with labelled data.

Unsupervised Learning: Machine learning that let the model discover and learn on their own, i.e., It works on its own to discover pattern and information. It is trained with unlabelled data.

Semi-supervised learning is a branch of machine learning that combines supervised and unsupervised learning by using both labeled and unlabeled data to train artificial intelligence (AI) models for classification and regression tasks.

Reinforcement Learning: In some applications, the output of the system is a sequence of actions; the learning in which machine is able to assess the goodness of past approaches or policies and learn from past good action sequences to be able to generate a policy.

**3. Supervised Learning, examples of classification and regression task.**

In machine learning, classification involves predicting a discrete category, while regression predicts a continuous value. Examples of classification tasks include identifying spam emails, classifying images as cats or dogs, or predicting whether a customer will click on an ad. Regression tasks, on the other hand, include predicting house prices, forecasting stock market trends, or estimating a person's salary.

**4. Unsupervised Learning, examples of clustering and association task.**

Clustering involves grouping similar data points, while association tasks find relationships between different data items. Examples of clustering include market segmentation, where customers are grouped by purchasing behaviour. Association analysis is exemplified by "market basket analysis," where items frequently bought together are identified.

**5. Sup vs Unsup vs rein vs semi sup**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Supervised Learning** | **Unsupervised Learning** | **Reinforcement Learning** |
| **Definition** | Learns from labeled data | Identifies patterns in unlabeled data | Learns through interaction with environment |
| **Type of Data** | Labeled data | Unlabeled data | No predefined data; environment-based |
| **Type of Problems** | Classification, Regression | Clustering, Association | Sequential decision-making |
| **Supervision** | Requires external supervision | No supervision | No supervision, learns from feedback |
| **Algorithms** | SVM, Decision Trees, Neural Networks | K-Means, PCA, Autoencoders | Q-learning, DQN, SARSA |
| **Goal** | Predict outcomes accurately | Discover hidden patterns | Optimize actions for maximum rewards |
| **Applications** | Medical diagnosis, fraud detection | Customer segmentation, anomaly detection | Self-driving cars, robotics, gaming |

**7.Role of dataset in ML. Diff types of datasets used in ML.**

The dataset lays the groundwork for machine learning, shaping the model's ability to learn and make accurate predictions. First, a dataset allows you to train your ML model, and, second, it provides a benchmark for measuring the accuracy of the model.

Machine learning models rely on various types of datasets, primarily categorized into numerical, categorical, time series, and text data.

* **Types of Datasets**
  + Structured Datasets
  + Unstructured Datasets
  + Semi-structured Datasets
  + Time-series Datasets
  + Image Datasets

**8. what is meant by dependent and independent variables?**

An independent variable is exactly what it sounds like. It is a variable that stands alone and isn't changed by the other variables you are trying to measure while dependent variable is something that depends on other factors.

(Independent variable) causes a change in (Dependent Variable) and it isn't possible that (Dependent Variable) could cause a change in (Independent Variable).

**9.What is meant by N-dimension pattern represented in 2d plane**

When someone says an "N-dimensional pattern represented in a 2D plane," they usually mean: There is data or a pattern that originally exists in N dimensions (where N could be 3, 5, 100, etc.). But we want to visualize or analyse it in 2 dimensions (the usual X-Y plane). So, some technique is used to project, flatten, or reduce the N-dimensional data into a 2D form while trying to preserve important features like structure, relationships, or patterns.

**10. What is meant by feature selection and feature extraction in ML?**

Feature selection involves selecting asubset of the most relevant features that are actually contributing in prediction while discarding the rest features. This helps improve reducing overfitting and increased accuracy. Common techniques include filter, wrapper and embedded methods.

Feature extraction transforms existing features into a new set of features that captures better underlying patterns in data. It is useful when raw data is in high dimension or complex. Techniques like PCA, LDA and Autoencoders are used for this purpose.

**11. Explain feature reduction with suitable examples.**

Feature reduction, also known as dimensionality reduction, is a technique used to simplify datasets by decreasing the number of features while preserving essential information. This can be achieved through feature selection (choosing relevant features) or feature extraction (creating new features).

Principal Component Analysis (PCA): A linear dimensionality reduction technique that transforms the original features into a set of uncorrelated principal components. These principal components capture the most variance in the data.

Linear Discriminant Analysis (LDA): A supervised dimensionality reduction technique that focuses on maximizing the separation between different classes in the data.

**14. What kind of preprocessing needs to be applied on dataset before passing it to machine learning model?**

Preprocessing steps are crucial to ensure accuracy and efficiency. These include data cleaning, which involves handling missing values, removing duplicates, and correcting inconsistencies; data transformation, which might involve scaling features, encoding categorical data, and handling outliers; and data splitting, which creates training, validation, and testing sets. Additionally, feature selection might be necessary to identify and remove irrelevant or redundant features.

**16. Use of Pandas, Numpy, Matplotlib, Seaborn, Sci-kit learn, sklearn, tensorflow, keras libraries.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Library** | **Main Purpose** | **Typical Usage** | **Detailed Description** |
| **NumPy** | Numerical operations, handling arrays and matrices efficiently. | Fast array calculations, creating numerical datasets, vectorized operations. | A fundamental library for numerical computing, providing support for arrays, matrices, and mathematical functions. It is crucial for efficient data manipulation and forms the base for many other libraries. |
| **Pandas** | Data loading, cleaning, manipulation (tables, CSVs, Excel, etc.). | Reading CSV files, cleaning missing values, dataframes handling, feature selection. | Built upon NumPy, it offers data structures like DataFrames for data manipulation and analysis. It excels in handling structured data, cleaning, and preprocessing. |
| **Matplotlib** | Basic plotting and visualizations (2D graphs, charts). | Line plots, bar charts, scatter plots, customized graphs. | A 2D plotting library for creating static, animated, and interactive visualizations. It is essential for exploring and presenting data insights. |
| **Seaborn** | Advanced visualization built on top of Matplotlib; easier and prettier plots. | Heatmaps, violin plots, pairplots, regression plots (statistical graphs). | Based on Matplotlib, Seaborn provides a high-level interface for creating informative and visually appealing statistical graphics. It simplifies the creation of complex visualizations. |
| **scikit-learn (sklearn)** | Machine learning library for classic ML algorithms and data processing. | Classification, regression, clustering (SVMs, Random Forest, k-NN), model evaluation. | A comprehensive machine learning library with tools for classification, regression, clustering, dimensionality reduction, and model selection. It simplifies the process of building and evaluating machine learning models. |
| **TensorFlow** | Deep learning framework (developed by Google). | Building neural networks, custom deep models, production ML systems. | An open-source framework for deep learning, enabling the development and training of neural networks. It is widely used for tasks like image recognition, natural language processing, and more. |
| **Keras** | High-level API over TensorFlow (easy-to-use for deep learning). | Quickly building, training, and testing neural networks (CNNs, RNNs, etc.). | A high-level API for building and training neural networks, running on top of TensorFlow (and other backends). It simplifies the development of deep learning models with a user-friendly interface. |

**17,18. Explain Regression task, its examples and difference between simple, multiple and polynomial regression.**

Regression tasks in machine learning aim to predict a continuous numerical value based on input data. It's used when you want to predict things like income, temperature, or the price of a house.

Simple linear regression models a linear relationship between one independent and one dependent variable.

Multiple linear regression extends this to include multiple independent variables.

Polynomial regression models non-linear relationships by fitting a curve to the data

**19. Explain performance evaluation parameters in Regression**

**Key Performance Evaluation Parameters:**

**Mean Absolute Error (MAE):** The average of the absolute differences between predicted and actual values. A lower MAE indicates better performance. Good for situations where you want to know the average magnitude of errors, without being overly influenced by extreme values.

**Mean Squared Error (MSE):** The average of the squared differences between predicted and actual values. It penalizes larger errors more heavily than MAE. Useful when you want to penalize larger errors more, as squaring the errors highlights those differences.

**Root Mean Squared Error (RMSE):** The square root of the MSE. It provides an error value in the same units as the target variable, making it easier to interpret.

**R-squared (Coefficient of Determination):** Represents the proportion of variance in the dependent variable that's explained by the model. A higher R-squared value (closer to 1) indicates a better fit. Indicates the goodness of fit of the model, but it doesn't tell you anything about the actual magnitude of the errors.

**20. Explain performance evaluation parameters in Classification task.**

These metrics provide insights into how well a model distinguishes between classes, especially in cases of imbalanced datasets.

Here's a breakdown of some key metrics:

**Accuracy:**

The proportion of correctly classified instances out of all instances.

**Precision:**

The proportion of correctly predicted positive instances out of all instances predicted as positive.

**Recall (Sensitivity or True Positive Rate):**

The proportion of correctly predicted positive instances out of all actual positive instances.

**F1-score:**

The harmonic mean of precision and recall, providing a balanced measure of the model's performance.

**Confusion Matrix:**

A table that visualizes the counts of true positives, true negatives, false positives, and false negatives, providing a detailed breakdown of the model's predictions.

**Area Under the Curve (AUC):**

The area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate against the False Positive Rate at various threshold settings. It represents the model's ability to distinguish between classes across different thresholds.

These metrics help assess the model's performance and identify areas for improvement, such as addressing class imbalance or tuning model hyperparameters.

**21. What is meant by hypothesis? Explain null and alternate hypothesis.**

A hypothesis is a proposed explanation for a phenomenon, a statement that can be tested.

Null Hypothesis (H0): This is the statement that assumes no effect or no relationship between variables being studied. It's often the default assumption, representing the idea that any observed effect is due to chance.

Alternative Hypothesis (Ha or H1): This is the statement that contradicts the null hypothesis. It proposes that there is a significant effect or relationship between the variables being studied.

**22. what is ridge and lasso regression?**

Ridge and Lasso regression are regularization techniques used in linear regression to prevent overfitting and improve model generalization by penalizing the magnitude of coefficients.

Lasso regression, also known as L1 regularization, adds a penalty proportional to the absolute value of the coefficients, potentially driving some coefficients to zero and effectively performing feature selection.

Ridge regression, or L2 regularization, adds a penalty proportional to the squared value of the coefficients, shrinking them towards zero without eliminating them.

**23. Explain classification task and its examples.**

A classification task in machine learning involves assigning instances of data to predefined categories or classes. Essentially, it's about learning to categorize things based on their features. This process is typically supervised, meaning the algorithm learns from labelled examples.

Examples:

* **Spam Detection:** Classifying emails as spam or not spam based on features like email address, content, and sender.
* **Image Recognition:** Classifying images as "cat", "dog", "bird", etc. based on pixel patterns and other visual features.
* **Sentiment Analysis:** Classifying text as having positive, negative, or neutral sentiment.

**24. How linear regression approach used for classification task?**

Normally, Linear Regression is designed for regression problems (predicting continuous values like 25.3, 42.7, etc.).  
However, in some cases, people have tried using it for classification by:

* Use Linear Regression to predict a continuous score.
* Then, set a threshold to convert that score into a class label.

Example:

* Predict a continuous output (say 0.8, 1.3, -0.5) using Linear Regression.
* If predicted value > 0.5, classify it as Class 1.
* If predicted value ≤ 0.5, classify it as Class 0.

**25. Explain parameterised ML models.**

1. Logistic Regression

A simple linear model used for classification (binary or multi-class).

* It calculates a linear combination of input features (like w1x1+w2x2+...+bw\_1x\_1 + w\_2x\_2 + ... + bw1​x1​+w2​x2​+...+b).
* Then it passes the result through a sigmoid function to get output between 0 and 1 (interpreted as probability).

2. Support Vector Machines (SVM)

A powerful supervised learning model mainly used for classification (and regression).

* SVM tries to find the best boundary (hyperplane) that separates different classes.
* The "best" hyperplane is the one with the maximum margin (maximum distance between classes).
* If classes are not linearly separable, SVM uses **kernels** to transform data into a higher dimension where a linear separator can exist.

3. Neural Networks (NN)

A network of neurons (basic computational units) organized in layers.

* Each neuron calculates a weighted sum of its inputs, adds a bias, and passes it through an activation function (like ReLU, sigmoid).
* There can be multiple layers (input, hidden layers, output).
* NNs can model very complex relationships if made deep enough (Deep Neural Networks).

4. Convolutional Neural Networks (CNN)

A specialized type of Neural Network mainly used for image data.

* Instead of connecting every neuron to every neuron, CNNs use convolutional layers that apply small filters/kernels on local regions of the input (like patches of an image).
* Pooling layers reduce the size of the data gradually (e.g., max pooling).
* Fully Connected layers (like a standard NN) come at the end.
* They are good at capturing spatial patterns (e.g., shapes, edges) in images.
* Parameter sharing and local connectivity make them efficient for big images.

Parameterized ML models are powerful because they learn their own best settings (weights/biases) from data instead of being manually programmed.

**26. explain probability-based ml models like naive bayes and their types, multinomial naive bayes.**

In Probability-based models, predictions are made based on the probabilities of outcomes. They use the rules of probability (like Bayes' theorem) to reason about the most likely output. These models don't just fit a curve — they model how likely each class is, given the data

Naive Bayes Classifier: A simple but powerful probabilistic classifier based on Bayes' Theorem and a strong assumption of feature independence.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Type** | **Data type** | **Feature assumption** | **Example Usage** |
| Gaussian Naive Bayes | Continuous | Normal distribution | Predicting age group |
| Multinomial Naive Bayes | Discrete (counts) | Multinomial distribution | Text classification |
| Bernoulli Naive Bayes | Binary | Bernoulli (yes/no) | Spam filter (word present or not) |

**27. Explain how linearly and non-linearly separable patterns are handled in SVM.**

SVM handles both linearly and non-linearly separable patterns by leveraging kernel functions. For linearly separable data, it finds the optimal hyperplane using a linear kernel. For non-linearly separable data, the "kernel trick" maps the data into a higher-dimensional space where it might become linearly separable, allowing the SVM to find a hyperplane in this new space.

**29. Different types of loss functions.**

Loss functions in machine learning quantify the difference between a model's predictions and the actual values, guiding the learning process by indicating how well the model is performing. Regression and classification tasks require different loss functions, as their goals and the nature of their outputs differ.

Regression Loss Functions:

**Mean Squared Error (MSE):** Measures the average of the squared differences between predicted and actual values. It's a commonly used loss function for regression due to its simplicity and smoothness.

**Mean Absolute Error (MAE):** Calculates the average of the absolute differences between predicted and actual values. MAE is less sensitive to outliers than MSE and can be more interpretable in some contexts.

**Log-Cosh Loss:** An approximation of MAE that penalizes large errors less aggressively than MSE but more smoothly than MAE. This can be beneficial when outliers exist but shouldn't dominate the loss function.

Classification Loss Functions:

**Cross-Entropy Loss:** Measures the difference between the predicted probability distribution of a data point belonging to each class and the true class label. It's used for both binary and multi-class classification problems.

**Binary Cross-Entropy Loss (Log Loss):** A specific type of cross-entropy loss used for binary classification problems where the output is a probability between 0 and 1.

**Categorical Cross-Entropy Loss:** Used for multi-class classification problems where the true labels are one-hot encoded vectors.

**Hinge Loss:** Used for training classifiers, especially Support Vector Machines (SVMs), where the goal is to find a decision boundary that maximizes the margin between classes. It penalizes misclassified examples and those that fall on the wrong side of the decision boundary.

**30. What is meat by gradient? Explain gradient descent algorithm with suitable example.**

Gradient Descent is used to iteratively update the weights (coefficients) and bias by computing the gradient of the LOSS FUNCTION with respect to these parameters.

Since LOSS FUNCTION is a convex function gradient descent guarantees convergence to the global minimum if the learning rate is appropriately chosen. For each iteration:

The algorithm computes the gradient of the LOSS FUNCTION with respect to the weights and biases.

It updates the weights (w) and bias (b) using the formula:

* Calculating the gradient of the log-loss with respect to the weights.
* Updating weights and biases iteratively to maximize the likelihood of the correct classification

**31. What is neural network? Explain different types of neural network.**

A neural network is a machine learning model inspired by the human brain, consisting of interconnected nodes called neurons that process data and learn patterns. It's a type of deep learning technology used to solve complex problems like image recognition and natural language processing. Different types of neural networks exist, each suited for specific tasks.

Types of Neural Networks:

**Feedforward Neural Networks (FNNs):** The simplest type where information flows in one direction, from input to output. They are commonly used for basic tasks like classification.

**Multilayer Perceptron (MLP):** A type of feedforward neural network with multiple layers, capable of solving complex tasks.

**Convolutional Neural Networks (CNNs):** Specialized for processing grid-like data, such as images, using convolutional layers to detect spatial hierarchies. They are ideal for computer vision tasks.

**Recurrent Neural Networks (RNNs):** Designed to process sequential data, like text or time series, using loops to retain information over time. RNNs are used in natural language processing and speech recognition.

**Long Short-Term Memory (LSTM) networks:** A type of RNN that addresses the vanishing gradient problem, making them well-suited for processing long sequences.

**Transformers:** A type of neural network that uses self-attention mechanisms, revolutionizing natural language processing for tasks like translation and text generation.

**Generative Adversarial Networks (GANs):** Consist of two networks, a generator and a discriminator, that compete to create realistic data, used for tasks like image generation.

**Self-Organizing Maps (SOM):** A type of neural network used for unsupervised clustering and dimensionality reduction.

**32. Characteristics of neural networks and explain why nn is better than other classifiers.**

Characteristics of Neural Networks:

Interconnected Nodes (Neurons): NNs are composed of interconnected artificial neurons, similar to biological neurons, enabling them to process information and learn from data.

Mimicry of the Human Brain: NNs are designed to mimic the way the human brain works, using interconnected nodes to process information and learn from experience.

Adaptability and Flexibility: NNs can adapt to new situations and learn from data, making them suitable for complex tasks where the relationship between inputs and outputs is not well-defined.

Pattern Recognition: NNs excel at identifying patterns in data, making them effective for tasks like audio and image recognition, natural language processing, and other intricate data patterns.

Parallel Processing: NNs can process multiple tasks concurrently due to their inherent parallel processing capability, which speeds up computation and enhances efficiency.

Non-Linearity: NNs utilize non-linear activation functions in neurons, allowing them to model and understand complex relationships in data that linear models cannot capture.

Why NNs are Better than Other Classifiers:

Handling Complex and High-Dimensional Datasets: NNs are well-suited for handling complex and high-dimensional datasets, making them effective for tasks like image recognition, text classification, and medical diagnosis.

Flexible and Adaptable: NNs are more flexible than classical methods and can adapt to new situations and learn from data.

Learning Complex Features: NNs can learn complex features from raw data, which can be more effective than manually extracting features for classification.

Superior Performance in Certain Applications: NNs often demonstrate superior performance compared to traditional machine learning techniques, especially when dealing with complex data formats like images and text, according to a study on AI in dentistry.

**33. Difference b/w Deep learning and Machine learning.**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Machine Learning (ML)** | **Deep Learning (DL)** |
| **Definition** | Field of AI where algorithms learn from data. | Subfield of ML focused on neural networks with multiple layers. |
| **Data Dependency** | Works well with **small to medium-sized** datasets. | Requires **large amounts** of data for good performance. |
| **Feature Engineering** | **Manual feature extraction** is important. | **Automatic feature extraction** from raw data. |
| **Model Complexity** | Models like decision trees, SVMs are relatively **simple**. | Models like CNNs, RNNs are **highly complex** and layered. |
| **Training Time** | **Faster** training, less computationally intensive. | **Slower** training, needs powerful GPUs/TPUs. |
| **Examples** | Linear Regression, Random Forest, KNN, SVM. | Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN). |
| **Applications** | Spam detection, fraud detection, price prediction. | Image recognition, speech recognition, autonomous driving. |

**34,35. Explain single layer perceptron and MLP with suitable example.**

Single-Layer Perceptron:

**Structure:** Consists of an input layer, an output layer, and a simple decision function (usually a threshold).

**Example:** Imagine classifying emails as "spam" or "not spam". A single-layer perceptron could learn to separate spam emails from legitimate ones based on the presence of certain keywords (e.g., "discount," "urgent").

**Limitations:** Can only learn linearly separable patterns. This means it can only divide the input space into two distinct regions using a straight line (or hyperplane in higher dimensions).

Multi-Layer Perceptron (MLP):

**Structure:** Has an input layer, one or more hidden layers, and an output layer. Each layer consists of multiple neurons that process the data.

**Example:** Recognizing handwritten digits (0-9). An MLP can learn complex relationships between pixel values and the corresponding digit. It can learn to identify features like lines, curves, and loops within the images.

**Benefits:** Capable of learning non-linear relationships, making them suitable for various machine learning tasks, including image recognition, natural language processing, and more.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Single-Layer Perceptron** | **Multi-Layer Perceptron (MLP)** |
| Layers | Input and output layers only | Input, hidden, and output layers |
| Complexity | Simple | Complex |
| Learnability | Linearly Separable Patterns | Non linear relationships |
| Power | Limited | Highly powerful |

**37. What is meant by:**

* 1. **bias and variance.**
  2. **trade-off of bias and variance**
  3. **correlation vs covariance**
  4. **loss function and activation function**
  5. **Regularization techniques in Deep learning**
  6. **optimization and different optimization techniques**

1. Bias and Variance:  
   Bias refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model. High bias can cause a model to miss relevant relations between features and target outputs, leading to underfitting.  
   Variance refers to the error introduced because the model is too sensitive to small fluctuations in the training dataset. High variance can cause overfitting, where the model learns noise and details from the training data that do not generalize to unseen data.
2. Trade-off of Bias and Variance:  
   The bias-variance trade-off is the balance between a model's ability to minimize bias and variance to achieve good generalization performance. A model with high bias pays very little attention to the training data and oversimplifies the model, while a model with high variance pays too much attention to the training data and does not generalize well. The goal is to find a balance where both bias and variance are reasonably low to achieve the best performance on unseen data.
3. Correlation vs Covariance:  
   Covariance measures the degree to which two variables move together. If two variables tend to increase or decrease together, their covariance is positive; if one increases while the other decreases, the covariance is negative. However, covariance is not normalized and depends on the units of the variables.  
   Correlation is a standardized version of covariance that measures both the strength and direction of a linear relationship between two variables and is scaled between -1 and 1. A correlation close to 1 indicates a strong positive relationship, close to -1 indicates a strong negative relationship, and close to 0 indicates no linear relationship.
4. Loss Function and Activation Function:  
   Loss function is a method to evaluate how well the model's predictions match the actual data. It calculates the error or difference between the predicted output and the actual output. The model uses this value to improve its predictions by minimizing the loss through optimization. Examples include Mean Squared Error for regression and Cross-Entropy Loss for classification.  
   Activation function is a function applied at each node or neuron in a neural network to introduce non-linearity into the model. Without activation functions, the neural network would behave like a linear model regardless of its depth. Examples include Sigmoid, ReLU, and Tanh functions.
5. Regularization Techniques in Deep Learning:  
   Regularization techniques are methods used to prevent overfitting in deep learning models. They add a penalty term to the loss function to discourage complex models. Common regularization techniques include:

* L1 Regularization (Lasso): Adds the absolute value of coefficients as a penalty term.
* L2 Regularization (Ridge): Adds the square of coefficients as a penalty term.
* Dropout: Randomly drops neurons during training to prevent reliance on specific neurons.
* Early Stopping: Stops training once the model performance on validation data starts degrading.

1. Optimization and Different Optimization Techniques:  
   Optimization in machine learning refers to the process of adjusting model parameters to minimize or maximize an objective function, such as minimizing the loss function during training.  
   Different optimization techniques include:

* Gradient Descent: Iteratively updates parameters in the opposite direction of the gradient of the loss function.
* Stochastic Gradient Descent (SGD): Updates parameters using a single or few training examples at each step.
* Mini-batch Gradient Descent: Updates parameters based on small batches of training data.
* Adam Optimizer: Combines momentum and adaptive learning rate methods for faster convergence.
* RMSProp: Modifies the learning rate based on the average of recent gradients for each weight.

**39. explain transfer learning and reinforcement learning.**

Transfer learning leverages pre-trained models that have already learned general features from large datasets. These models are then fine-tuned on a new, related task with potentially smaller datasets, saving time and computational resources.

RL involves an agent learning to make a sequence of decisions in an environment to maximize a reward signal. The agent learns by interacting with the environment and receiving feedback in the form of rewards (positive) or penalties (negative) for its actions.

41. Differentiate b/w ML vs DL vs CNN vs RNN vs LSTM vs BERT vs Transformers

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Aspect** | **Machine Learning (ML)** | **Deep Learning (DL)** | **CNN (Convolutional Neural Network)** | **RNN (Recurrent Neural Network)** | **LSTM (Long Short-Term Memory)** | **BERT (Bidirectional Encoder Representations from Transformers)** | **Transformers** |
| Definition | Field of AI where models learn from data | Subset of ML using multi-layered neural networks | Type of DL model for processing grid-like data | Type of DL model for sequential data | Special RNN capable of learning long-term dependencies | Pre-trained language model based on Transformer encoder | Model architecture for sequence modeling |
| Data Handling | Small to medium datasets | Large datasets | Primarily images, videos | Sequential data like time series, text | Sequential data with long-term memory handling | Text data, contextual language understanding | Handles sequential data efficiently |
| Feature Extraction | Manual | Automatic | Automatic via convolution filters | Automatic, but struggles with long dependencies | Automatic, designed to remember longer sequences | Automatic, deep contextual representations | Automatic, attention-based mechanisms |
| Model Complexity | Relatively simple | Highly complex | Complex but efficient for image tasks | Moderate complexity | More complex than RNN due to gates | Very complex with deep attention layers | Complex, but scalable with large data |
| Memory Handling | Not applicable | Some architectures may support | No memory of previous inputs | Remembers previous input in hidden state | Better memory through gates controlling flow of information | No direct memory like RNNs but contextual attention | No direct memory, relies on attention |
| Application Examples | Spam detection, fraud detection | Image recognition, autonomous vehicles | Image classification, object detection | Language modeling, time series prediction | Speech recognition, text generation | Question answering, text classification, translation | Text generation, translation, summarization |
| Architecture Type | Varies (trees, regression, SVMs) | Neural networks with multiple layers | Convolutional layers with pooling layers | Looping architecture processing sequential data | RNN variant with input, output, forget gates | Transformer-based encoder model | Encoder-decoder attention-based model |