93

Apr-25

Scoring Indicators

P-15

COURSE NAME: MACHINE LEARNING AND NEURAL NETWORKS

COURSE CODE: 4342

QID: 2102250068-B

Q. No.		Scoring Indicators	Split Score	Sub Total	Total Score
PART A					9
I	1	Machine learning is a branch of artificial intelligence that enables algorithms to uncover hidden patterns within datasets.	1	1	
I	2	Agglomerative clustering is a hierarchical clustering method that groups similar items into clusters.	1	1	
I	3	K-Means Clustering is an Unsupervised Machine Learning algorithm which groups the unlabeled dataset into different clusters.	1	1	
I	4	Hierarchical clustering,Centroids-based Clustering (Partitioning methods),Distribution-based Clustering,Density-based Clustering (Model-based methods), Fuzzy Clustering, Constraint-based (Supervised Clustering)	1	1	
I	5	Mean squared error (MSE) is a metric used to measure the average squared difference between the predicted values and the actual values in the dataset.	1	1	
I	6	Reinforcement Learning (RL) is a branch of machine learning focused on making decisions to maximize cumulative rewards in a given situation.	1	1	
I	7	Social media, marketing and sales	1	1	
I	8	An activation function is a mathematical function applied to the output of a neuron.	1	1	
I	9	Batch normalization is a deep learning technique that helps our models learn and adapt quickly.	1	1	
PART B					24
II	1	Automation and Efficiency: Machine learning algorithms can automate repetitive and time-consuming tasks, increasing efficiency and allowing humans to focus on higher value-added tasks. Improved Decision-Making: Machine learning allows companies to make accurate predictions and informed decisions based on specific data and detailed analysis. Personalization: Machine learning-based technologies, such as recommendation systems, allow personalized products and services that better meet individual users' needs and preferences.		3	

II	2	Machine Learning techniques are divided mainly into the following 4 categories: 1. Supervised Learning Supervised learning is applicable when a machine has sample data, i.e., input as well as output data with correct labels. Correct labels are used to check the correctness of the model using some labels and tags. Supervised learning technique helps us to predict future events with the help of past experience and labeled examples. 2. Unsupervised Learning In unsupervised learning, a machine is trained with some input samples or labels only, while output is not known. The training information is neither classified nor labeled; hence, a machine may not always provide correct output compared to supervised learning. 3. Reinforcement Learning Reinforcement Learning is a feedback-based machine learning technique. In such type of learning, agents (computer programs) need to explore the environment, perform actions, and on the basis of their actions, they get rewards as feedback. 4. Semi-supervised Learning Semi-supervised Learning is an intermediate technique of both supervised and unsupervised learning. It performs actions on datasets having few labels as well as unlabeled data.		3	
II	3	Data normalization is a vital pre-processing, mapping, and scaling method that helps forecasting and prediction models become more accurate. The current data range is transformed into a new, standardized range using this method. Normalization is extremely important when it comes to bringing disparate prediction and forecasting techniques into harmony. Data normalization improves the consistency and comparability of different predictive models by standardizing the range of independent variables or features within a dataset, leading to more steady and dependable results. Normalisation, which involves reshaping numerical columns to conform to a standard scale, is essential for datasets with different units or magnitudes across different features. Finding a common scale for the data while maintaining the intrinsic variations in value ranges is the main goal of normalization. This usually entails rescaling the features to a standard range, which is typically between 0 and 1. Alternatively, the features can be adjusted to have a mean of 0 and a standard deviation of 1. Z-Score Normalisation (Standardisation) and Min-Max Scaling are two commonly used normalisation techniques. In order to enable more insightful and precise analyses in a variety of predictive modelling scenarios, these techniques are essential in bringing disparate features to a comparable scale.		3	
II	4	Data normalization is a vital pre-processing, mapping, and scaling method that helps forecasting and prediction models become more accurate. The current data range is transformed into a new, standardized range using this method. Normalization is extremely important when it comes to bringing disparate prediction and forecasting techniques into harmony. Data normalization improves the consistency and comparability of different predictive models by	3	3	

		standardizing the range of independent variables or features within a dataset, leading to more steady and dependable results. Normalisation, which involves reshaping numerical columns to conform to a standard scale, is essential for datasets with different units or magnitudes across different features. Finding a common scale for the data while maintaining the intrinsic variations in value ranges is the main goal of normalization. This usually entails rescaling the features to a standard range, which is typically between 0 and 1. Alternatively, the features can be adjusted to have a mean of 0 and a standard deviation of 1. Z-Score Normalisation (Standardisation) and Min-Max Scaling are two commonly used normalisation techniques. In order to enable more insightful and precise analyses in a variety of predictive modelling scenarios, these techniques are essential in bringing disparate features to a comparable scale.			
II	5	1. Determine the Suitability of Data for Factor Analysis 2. Choose the Extraction Method 3. Factor Extraction 4. Determine the Number of Factors to Retain 5. Factor Rotation 6. Interpret and Label the Factors 7. Compute Factor Scores (if needed) 8. Report and Validate the Results	3	3	
II	6	The main drawbacks of partitional clustering in machine learning are: requiring the pre-defined number of clusters (k), sensitivity to initialization, potential for getting stuck in local optima, difficulty handling non-spherical clusters, and poor performance with unevenly sized clusters; essentially, it struggles when clusters have complex shapes or sizes, and the optimal number of clusters is not readily known, making it less adaptable to diverse data structures. Key points about partitional clustering limitations: Need to specify 'k': Unlike hierarchical clustering, partitional algorithms like K-means require users to specify the desired number of clusters beforehand, which can be challenging to determine accurately. Initialization dependence: The initial placement of cluster centroids can significantly impact the final clustering results, potentially leading to suboptimal solutions if not carefully chosen. Sensitivity to outliers: Outliers can significantly distort the clustering process, as the algorithm tries to minimize distances to these extreme points, affecting the overall cluster quality. Assumption of spherical clusters: Partitional methods typically assume clusters are roughly spherical and evenly distributed, which may not be true for real-world data with complex shapes. Poor performance with uneven sizes: If clusters have vastly different sizes, partitional clustering may struggle to accurately identify and separate them.	3	3	
II	7	Types of Reinforcements in RL 1. Positive Reinforcement	3	3	

		Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior. Advantages: Maximizes performance, helps sustain change over time. Disadvantages: Overuse can lead to excess states that may reduce effectiveness. 2. Negative Reinforcement Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided. Advantages: Increases behavior frequency, ensures a minimum performance standard. Disadvantages: It may only encourage just enough action to avoid penalties.			
II	8	A loss function in machine learning is a mathematical function that measures how well a model's predictions match the actual results. It's used to guide the model's training process. Types of loss functions Mean absolute error (MAE): Calculates the average magnitude of errors in a set of predicted values Mean squared error (MSE): Calculates the square of the difference between predicted and actual value Huber loss: Combines the strengths of both MAE and MSE Cross entropy loss: Used when only two classes are in the dataset, such as spam or not spam Categorical cross-entropy loss: Measures the difference between classes Kullback-Leibler divergence (KL divergence): Measures how one probability distribution differs from a reference probability distribution Hinge loss: Commonly used in support vector machines (SVMs) for classification tasks	3	3	
II	9	1.ANN is the mathematical model which is mainly inspired by the biological neuron system in the human brain. In contrast, the biological neural network is also composed of several processing pieces known as neurons that are linked together via synapses. 2.An artificial neural network's processing was sequential and centralized. In contrast, a biological neural network processes information in parallel and distributive. 3.The artificial neural network is of a much smaller size than the biological neural network. In contrast, the biological neural network is large in size. 4.The biological neural network is fault tolerant. In contrast, the artificial neural network is not fault tolerant. 5.The processing speed of an artificial neural network is in the nanosecond range, which is faster than the biological neural network, where the cycle time associated with a neural event triggered by an external input is in the millisecond range. 6.BNN may perform more difficult issues than artificial neural networks. 7.The operating environment of the artificial neural network is well-defined and well-constrained. In contrast, the operating environment of the biological neural network is poorly defined and unconstrained.	any 3	3	

		8.The reliability of the artificial neural network is very vulnerable. In contrast, the reliability of the biological neural network is robust.			
II	10	<table><tr><th>Comparison Attribute</th><th>Feed-forward Neural Networks</th><th>Recurrent Neural Networks</th></tr><tr><td>Signal flow direction</td><td>Forward only</td><td>Bidirectional</td></tr><tr><td>Delay introduced</td><td>No</td><td>Yes</td></tr><tr><td>Complexity</td><td>Low</td><td>High</td></tr><tr><td>Neuron independence in the same layer</td><td>Yes</td><td>No</td></tr><tr><td>Speed</td><td>High</td><td>slow</td></tr></table>	any 3	3	
		PART A			42
III	III	1. Data Collection Machine learning requires training data, a lot of it. This data can either be labelled meaning Supervised Learning or not labelled meaning Unsupervised Learning. Accuracy of the model depends on the quality and quantity of the data. The outcome of this step is generally a representation of data which will be used for training. Using pre-collected data, by way of datasets from sites like Kaggle, UCI, etc. forms the basis of you Machine learning project. You may also collect data through user-surveys, analysis reports, trends, usage metrics, etc. 2. Data Preparation We cannot work on raw data. Data needs to be processed by normalization, removing duplicates, errors and biases. Visualising data can be helpful in searching for patterns and outliers to check if the data collected is right or if it contains missing values. This can be done using libraries like seaborn, matplotlib, etc. Visualize data to help detect relevant relationships between variables or class imbalances, or perform other exploratory analysis. After performing data wrangling, we need to prepare the data for training. Cleaning of data is done that involves steps like removing duplicates, dealing with missing values, type conversions, correcting errors, normalizing the data, etc. Not all the above steps are needed to be performed as it depends entirely on the data collected. Some datasets may not require data	7	7	

preparation at all while for some data preparation step takes majority of their ML model build time.

We can also Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data. Later we can split the data into training, testing and evaluation sets

3. Choose a Model / Algorithm
The third step consists of selecting the right model. There are many models which can be used for many different purposes. Once the model is selected, it needs to meet the business goal.

We need to have an idea about the preparation the model requires along with its accuracy and scalability. Having a complex model does not mean a better model.

Common machine learning algorithms include Decision Trees, Random Forest, Linear Regression, Support Vector Machines (SVM), Logistic Regression, K-means, Principal Component Analysis (PCA), Naïve Bayes, and Neural Networks. Different algorithms need to be applied to different tasks, you need to choose the correct one for your use case.

4. Training the Model
Training a model forms the basis of machine learning. The goal is to use our training data and improve the predictions of our model.

Every cycle in training a model involves updating the weights and biases in each training step. We can use labelled sample data in case supervised machine learning and unlabelled sample data for unsupervised learning.

The goal of training is to evaluate and further improve our model accuracy and performance. Training happens in the form of iterations which is called a training step.

5. Evaluate the Model
After training the model comes evaluating the model. The larger the number of variables in the real world, the bigger the training and test data should be.

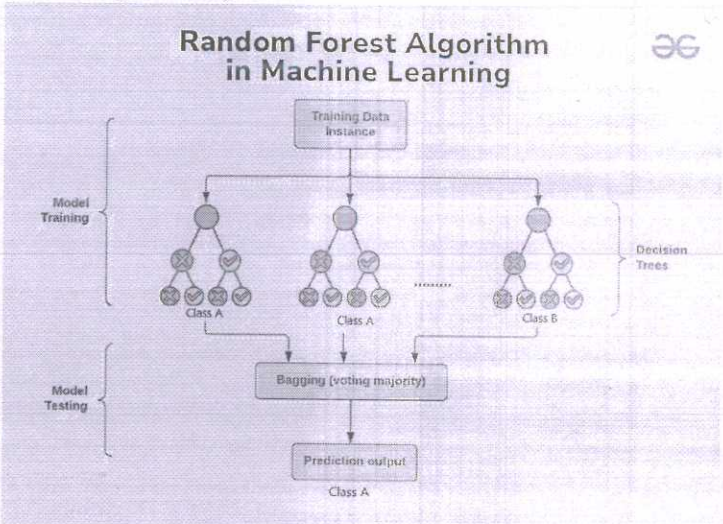
Performance metrics are used to measure the performance of the model. These include precision, recall, accuracy, specificity, etc.

The model is then tested against previously unseen data. The unseen data is meant to act as representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not).

A 70/30 split, or similar, is considered a good train/eval split, which depends on things like data availability, dataset features, domain, etc.

6. Parameter Tuning
The original model parameters need to be tested after evaluating your model. By increasing the training, it can lead to better results.

		Parameter tuning is an experimental process and hence we need to define when to stop parameter tuning otherwise it will continue to tweak the model. Hyperparameter tuning is an art and one that requires patience & experience. Once the model parameters are tuned it can give us better results. Some common hyperparameters include: number of training steps, learning rate, initialization values and distribution, etc. 7. Make Predictions After the processes of collecting data, preparing the data, selecting a machine learning algorithm, training the model and evaluating the model & tuning the parameters, we need to make predictions. Our machine learning model can make predictions ranging from image recognition to predictive analytics to natural language processing. After building the model needs to be tested on a testing set to check how the model performs on unseen data. It helps to further evaluate the model and provides better approximation.			
IV	IV	The integrity of data analysis is highly dependent on the quality of data preprocessing. Data preprocessing determines the usability and interpretability of data, laying the groundwork for accurate machine learning and AI models. Eliminating Errors Cleaning is a pivotal data preprocessing technique. It allows you to eliminate errors, impute missing values, and rectify inconsistencies. For example, a customer dataset with redundant entries due to technical mistakes would undergo cleaning to ensure each customer record is unique and accurately represented. Making Data Uniform Normalization is comparable to establishing a level playing field, where disparate measures are adjusted to a uniform scale, enabling equitable comparisons. For instance, normalization can help you analyze the performance of stocks from different countries despite stock prices being available in various currencies and scales. With normalization techniques such as min-max, you can convert all stock prices into a common currency, for example, USD, and then apply a min-max scaling to compare the relative performance of stocks on a uniform scale. Finding Hidden Patterns Diligent preprocessing can reveal concealed patterns and insights. A marketing team analyzing social media data can identify peak engagement times aligned with spam activity. However, excluding anomalies through data cleaning will allow you to pinpoint genuine peak engagement periods and optimize strategy. Big Data Preprocessing As datasets grow in size and complexity, preprocessing becomes even more critical. Big data has a large volume, is	7	7	

		heterogeneous, and needs to be processed rapidly. Preprocessing transforms raw big data into a cleaner, more structured format, removing noise and making it easier to process. Similarly, advanced techniques such as parallel processing, distributed computing, and automated preprocessing pipelines are indispensable for processing big data effectively.			
V	V	A Random Forest is a collection of decision trees that work together to make predictions. In this article, we'll explain how the Random Forest algorithm works and how to use it. Understanding Intuition for Random Forest Algorithm Random Forest algorithm is a powerful tree learning technique in Machine Learning to make predictions and then we do voting of all the tress to make prediction. They are widely used for classification and regression task. It is a type of classifier that uses many decision trees to make predictions. It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning. Imagine asking a group of friends for advice on where to go for vacation. Each friend gives their recommendation based on their unique perspective and preferences (decision trees trained on different subsets of data). You then make your final decision by considering the majority opinion or averaging their suggestions (ensemble prediction).  Key Features of Random Forest Handles Missing Data: Automatically handles missing values during training, eliminating the need for manual imputation. Algorithm ranks features based on their importance in making predictions offering valuable insights for feature selection and interpretability. Scales Well with Large and Complex Data without significant performance degradation. Algorithm is versatile and can be applied to both classification tasks (e.g., predicting categories) and regression tasks (e.g., predicting continuous values).	7	7	

VI	VI	*Computer vision – Mean-Shift clustering is widely used in computer vision for object tracking, image segmentation, and feature extraction.* *Image processing – Mean-Shift clustering is used for image segmentation, which is the process of dividing an image into multiple segments based on the similarity of the pixels.* *Anomaly detection – Mean-Shift clustering can be used for detecting anomalies in data by identifying the areas with low density.* *Customer segmentation – Mean-Shift clustering can be used for customer segmentation in marketing by identifying groups of customers with similar behavior and preferences.* *Social network analysis – Mean-Shift clustering can be used for clustering users in social networks based on their interests and interactions.*	7	7	
VII	VII	Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique commonly used in machine learning and pattern recognition. In the context of classification it aims to find a linear combination of features that best separates different classes or categories of data. It seeks to reduce the dimensionality of the feature space while preserving as much of the class-separability information as possible. let's walk through a simple example to understand how Linear Discriminant Analysis (LDA) works: Example: Iris Flower Classification Suppose we have a dataset of iris flowers with four features: sepal length, sepal width, petal length, and petal width. We want to classify these flowers into three species: Setosa, Versicolor, and Virginica. Steps: Data Preparation: Let's say we have 150 iris samples with four features each, and the samples are evenly distributed among the three species. Compute Class Statistics: Calculate the mean and covariance matrix for each feature in each class. This gives us three mean vectors and three covariance matrices (one for each class). Compute Between-Class and Within-Class Scatter Matrices: Calculate the between-class scatter matrix by computing the differences between the mean vectors of each class and the overall mean, and then summing these outer products. Calculate the within-class scatter matrix by summing the covariance matrices of each class, weighted by the number of samples in each class. Compute Eigenvectors and Eigenvalues: Solve the generalized eigenvalue problem using the between-class scatter matrix and the within-class scatter matrix. This gives us a set of eigenvectors and their corresponding eigenvalues. Select Discriminant Directions: Sort the eigenvectors by their eigenvalues in descending order. Let's say we want to reduce the dimensionality to 2, so we select the top two eigenvectors. Transform Data: Project the original iris data onto the two selected eigenvectors. This gives us a new two-dimensional representation of the data. Classification: In the reduced-dimensional space, we can use a classifier (e.g., k-nearest neighbors) to classify the iris flowers into one of the three species based on their positions in the reduced space.	7	7	

VIII	VIII	Types of Multidimensional Scaling 1. Classical Multidimensional Scaling Classical Multidimensional Scaling is a technique that takes an input matrix representing dissimilarities between pairs of items and produces a coordinate matrix that minimizes the strain. The steps of a Classical MDS algorithm include setting up the squared proximity matrix, applying double centering to compute matrix B, determining the m largest eigenvalues and corresponding eigenvectors of B, and obtaining the coordinates matrix X. 2. Metric Multidimensional Scaling Metric Multidimensional Scaling generalizes the optimization procedure to various loss functions and input matrices with known distances and weights. It minimizes a cost function called "stress," often minimized using a procedure called stress majorization. 3. Non-metric Multidimensional Scaling Non-metric Multidimensional Scaling finds a non-parametric monotonic relationship between dissimilarities and Euclidean distances between items, along with the location of each item in the low-dimensional space. It defines a "stress" function to optimize, considering a monotonically increasing function f.	7	7	
IX	IX	Key Components of Q-learning **1. Q-Values or Action-Values :** These represent the expected rewards for taking an action in a specific state. The agent updates these values over time using the Temporal Difference (TD) update rule. **2. Rewards and Episodes:** The agent moves through different states by taking actions and receiving rewards. The process continues until the agent reaches a terminal state which ends the episode. **3. Temporal Difference or TD-Update:** The agent updates Q-values using the formula: $Q(S,A) \leftarrow Q(S,A)+\alpha(R+\gamma Q(S',A')-Q(S,A))$ $Q(S,A) \leftarrow Q(S,A)+\alpha(R+\gamma Q(S',A')-Q(S,A))$	7	7	
X	X	1. Accuracy 2. Confusion Matrix 3. Precision, Recall and F1 Score 4. ROC Curve 1. Accuracy Classification accuracy is the simplest evaluation metric. It is defined as the number of correct predictions divided by the total number of predictions multiplied by 100. The accuracy metric works great if the target variable classes in the data are approximately balanced. For example, if 60% of the classes in an animal dataset are dogs and 40% are cats, then we can say that it is a balanced dataset. It calculates the ratio of correctly predicted instances to the total instances. 2. Confusion Matrix The confusion matrix is another way to evaluate the performance of a classifier. Here, it counts the number of times instances of class A	7	7	

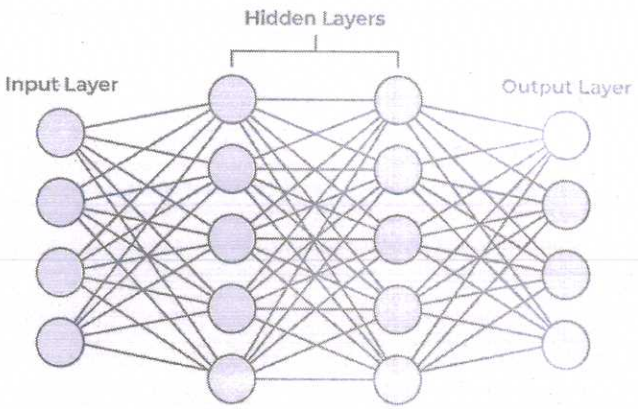
		are classified as class B. For example, the number of times the classifier confused images of 5s with non-5s. This is a table that is often used to describe the performance of a classification model. It presents a summary of the predictions made by the model against the actual class labels. The confusion matrix is a matrix with four different combinations of predicted and actual classes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Let's compute the confusion matrix to evaluate the performance of a classifier. We can make use of MNIST dataset to compute the confusion matrix. 3.1 Precision: Precision provides the accuracy of the positive prediction made by the classifier. The equation is as follows: Precision = True Positive / (True Positive + False Positive) 3.2 Recall: Recall is the ratio of number of true positive predictions (correctly detected by the classifer) to the total number of actual positive instances in the dataset. It measures the completeness of positive predictions. The equation is as follows: Recall = True Positve / (True Positive + False Negative) 3.3 F1 Score: The F1 score is the harmonic mean of precision and recall. It favors classifiers that have similar precision and recall. Here, the classifier will only get a high F1 score if both recall and precision are high. The equation is as follows: F1 = 2 * (Precision * Recall) / (Precision + Recall) 4. ROC Curve The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a classification model at various thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR). The Area Under the ROC Curve (AUC-ROC) is a metric to evaluate the performance of a binary classification model. AUC-ROC value lies between 0 and 1, where a higher value indicates better performance. AUC-ROC is insensitive to class distribution and gives an aggregate measure of performance across all possible classification thresholds. The true positive rate is calculated as: TPR = True Positives / (True Positives + False Negatives)			
XI	XI	Key Concepts of Reinforcement Learning Agent: The learner or decision-maker. Environment: Everything the agent interacts with. State: A specific situation in which the agent finds itself. Action: All possible moves the agent can make. Reward: Feedback from the environment based on the action taken.	7	7	
XII	XII	Artificial Neural Networks and its Applications Last Updated : 07 Aug, 2024 As you read this article, which organ in your body is thinking about it? It's the brain of course! But do you know how the brain works? Well, it has neurons or nerve cells that are the primary units of both the brain and the nervous system. These neurons receive sensory input from the outside world which they process and then provide the output which might act as the input to the next neuron. Each of these neurons is connected to other neurons in complex arrangements at synapses. Now, are you wondering how this is related to Artificial Neural Networks ? Let's check out what they are in detail and how they learn information.	7	7	

Well, Artificial Neural Networks are modeled after the neurons in the human brain. If you want to gain practical skills in Artificial Neural Networks and explore their diverse applications through our interactive live data science course , perfect for aspiring data scientists.

Artificial Neural Networks

Artificial Neural Networks contain artificial neurons which are called units . These units are arranged in a series of layers that together constitute the whole Artificial Neural Network in a system. A layer can have only a dozen units or millions of units as this depends on how the complex neural networks will be required to learn the hidden patterns in the dataset. Commonly, Artificial Neural Network has an input layer, an output layer as well as hidden layers. The input layer receives data from the outside world which the neural network needs to analyze or learn about. Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer. Finally, the output layer provides an output in the form of a response of the Artificial Neural Networks to input data provided.

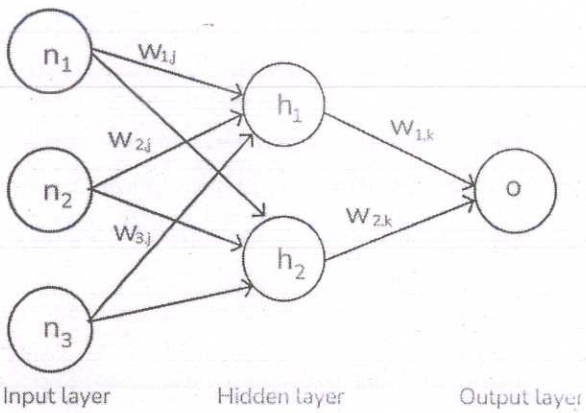
In the majority of neural networks, units are interconnected from one layer to another. Each of these connections has weights that determine the influence of one unit on another unit. As the data transfers from one unit to another, the neural network learns more and more about the data which eventually results in an output from the output layer.



The structures and operations of human neurons serve as the basis for artificial neural networks. It is also known as neural networks or neural nets. The input layer of an artificial neural network is the first layer, and it receives input from external sources and releases it to the hidden layer, which is the second layer. In the hidden layer, each neuron receives input from the previous layer neurons, computes the weighted sum, and sends it to the neurons in the next layer. These connections are weighted means effects of the inputs from the previous layer are optimized more or less by assigning different-different weights to each input and it is adjusted during the training process by optimizing these weights for improved model performance.

Artificial neurons vs Biological neurons

		The concept of artificial neural networks comes from biological neurons found in animal brains So they share a lot of similarities in structure and function wise. Structure : The structure of artificial neural networks is inspired by biological neurons. A biological neuron has a cell body or soma to process the impulses, dendrites to receive them, and an axon that transfers them to other neurons. The input nodes of artificial neural networks receive input signals, the hidden layer nodes compute these input signals, and the output layer nodes compute the final output by processing the hidden layer's results using activation functions. Synapses : Synapses are the links between biological neurons that enable the transmission of impulses from dendrites to the cell body. Synapses are the weights that join the one-layer nodes to the next-layer nodes in artificial neurons. The strength of the links is determined by the weight value. Learning : In biological neurons, learning happens in the cell body nucleus or soma, which has a nucleus that helps to process the impulses. An action potential is produced and travels through the axons if the impulses are powerful enough to reach the threshold. This becomes possible by synaptic plasticity, which represents the ability of synapses to become stronger or weaker over time in reaction to changes in their activity. In artificial neural networks, backpropagation is a technique used for learning, which adjusts the weights between nodes according to the error or differences between predicted and actual outcomes. Biological Neuron Artificial Neuron Synaptic plasticity Backpropagations Activation : In biological neurons, activation is the firing rate of the neuron which happens when the impulses are strong enough to reach the threshold. In artificial neural networks, A mathematical function known as an activation function maps the input to the output, and executes activations.			
XIII	XIII	Basic Components of Perceptron A Perceptron is composed of key components that work together to process information and make predictions. Input Features: The perceptron takes multiple input features, each representing a characteristic of the input data. Weights: Each input feature is assigned a weight that determines its influence on the output. These weights are adjusted during training to find the optimal values. Summation Function: The perceptron calculates the weighted sum of its inputs, combining them with their respective weights. Activation Function: The weighted sum is passed through the Heaviside step function, comparing it to a threshold to produce a binary output (0 or 1). Output: The final output is determined by the activation function, often used for binary classification tasks. Bias: The bias term helps the perceptron make adjustments independent of the input, improving its flexibility in learning.	7	7	

		Learning Algorithm: The perceptron adjusts its weights and bias using a learning algorithm, such as the Perceptron Learning Rule, to minimize prediction errors. These components enable the perceptron to learn from data and make predictions. While a single perceptron can handle simple binary classification, complex tasks require multiple perceptrons organized into layers, forming a neural network.			
XI V	XIV	Backpropagation (short for "Backward Propagation of Errors") is a method used to train artificial neural networks. Its goal is to reduce the difference between the model's predicted output and the actual output by adjusting the weights and biases in the network. Working of Backpropagation Algorithm The Backpropagation algorithm involves two main steps: the **Forward Pass** and the **Backward Pass**. How Does the Forward Pass Work? In the **forward pass**, the input data is fed into the input layer. These inputs, combined with their respective weights, are passed to hidden layers. For example, in a network with two hidden layers (h1 and h2 as shown in Fig. (a)), the output from h1 serves as the input to h2. Before applying an activation function, a bias is added to the weighted inputs. Each hidden layer applies an activation function like **ReLU (Rectified Linear Unit)**, which returns the input if it's positive and zero otherwise. This adds non-linearity, allowing the model to learn complex relationships in the data. Finally, the outputs from the last hidden layer are passed to the output layer, where an activation function, such as **softmax**, converts the weighted outputs into probabilities for classification.  *The forward pass using weights and biases* How Does the Backward Pass Work? In the backward pass, the error (the difference between the predicted and actual output) is propagated back through the	7	7	

network to adjust the weights and biases. One common method for error calculation is the **Mean Squared Error (MSE)**, given by:

MSE=(Predicted Output−Actual Output)2

MSE=(Predicted Output−Actual Output)2

Once the error is calculated, the network adjusts weights using **gradients**, which are computed with the chain rule. These gradients indicate how much each weight and bias should be adjusted to minimize the error in the next iteration. The backward pass continues layer by layer, ensuring that the network learns and improves its performance. The activation function, through its derivative, plays a crucial role in computing these gradients during backpropagation.

Example of Backpropagation in Machine Learning
Let's walk through an example of backpropagation in machine learning. Assume the neurons use the sigmoid activation function for the forward and backward pass. The target output is 0.5, and the learning rate is 1.