M T W T F S

DATE: 5/4/2023

Assignment #03

Name: Lufyan Ahmad
RollNo: 20198013
Course: DS-306
Section: A
Date: 05-04-2023

Q#04

Solution

Hierarchical Clustering:-

Hierarchical clustering is a clustering technique that groups data points based on their similarity. Using hierarchical structure of nested clusters. There are two types of hierarchical clustering "agglomerative" and "divisive". Agglomerative hierarchical clustering is a "bottom-up" approach where each data point is initially considered as a separate cluster and then the algorithm iteratively merge the

M T W T F S

DATE: 5/4/2023

Closest clusters until all the data points belong to single cluster. "Divisive" hierarchical clustering, on the other hand, is a "top-down" approach, where all the data points are initially considered as a single cluster and then the algorithm iteratively splits the cluster into smaller clusters until each data point belong to its own cluster

Example:-

It is example of agglomerative hierarchial clustering. Suppose we have a dataset of points in a two-dimensional space, and we want to cluster them into three groups. we can start by considering each point as a separate cluster. the algorithm then computes the distance between each pair of cluster and merges the two closest clusters into a new cluster. Until all the points

M T W T F S DATE: 5/4/2023

belong to a single cluster. For instance, suppose we have the following points (1,1), (1,2), (3,3), (4,5), (5,5). We can represent them graphically as shown below.

(4,5) • (3,3) •
(5,5) •• (1,2) •
(5,4) • (1,1) •

We can start by computing the distance between each pair of points and creating a distance matrix.

~~(1,1)~~	(1,1)	(1,2)	(3,3)	(4,5)	(5,4)	(5,5)
(1,1)	0	1.0	2.8	5.0	4.2	4.6
(1,2)		0	2.2	4.2	3.6	3.6
(3,3)			0	2.8	2.0	2.2
(4,5)				0	1.4	1.4
(5,4)					1.4	1.0
(5,5)						0

We can then find the two closest clusters and merge them into a new cluster. In this case the two closest clusters are ((4,5), (5,5)) and (5,4), with

M T W T F S DATE: 5/4/2023

a distance of 1.0. We can merge them into a new cluster and update the distance matrix.

	(1,1)	(1,2)	(3,3)	(4,5,5,4)	(5,5,5,5)
(1,1)	0	1.0	2.8	5.0	5.1
(1,2)		0	2.2	4.2	4.2
(3,3)			0	2.8	2.8
(4,5,5,4)				0	1.4
(5,5,5,5)					0

We can then repeat the process by finding the two closest clusters and merging them into a new cluster, until all the points belong to single cluster. In this case, the next closest clusters are (1,1), (1,2) and (3,3) with a distance.

Q#03

Solution

The K-means algorithm is an

M T W T F S

DATE: 05/04/2023

Unsupervised machine learning algorithm used to cluster data points into K distinct groups based on their similarity. one of the key step in the K-means algorithm is to randomly initialize K cluster centroids, and then iteratively refine their positions to minimize the within-cluster variation which is the sum of the squared distance between each datat point and its closest centroid.

However, the K-means algorithm is susceptible to getting stuck in local optima, which are solution that are optimal within a local region but not necessarily globally optimal. This can happen when the initial cluster centroids are not well placed, or when

M T W T F S DATE: 5/4/2023

the data points are distributed in a way that makes it hard for algorithm to find the true underlying structure.

Example

Consider a dataset of points that form two concentric circles with different radii, and let $K=2$. If the initial cluster centroids happen to be placed near the center of the circles, the algorithm may converge to a suboptimal solution where one cluster includes all the points in the inner circle and the other includes all the points in the outer circle. However, the true optimal solution would be to have one cluster for each circle, which would result in a lower within-cluster variation.

In such case, it may be

M T W T F S

DATE: 5/4/2023

be necessary to run the K-means algorithm multiple times with different initializations. or to use a more sophisticated clustering algorithm that is less (Semi-K)^x sensitive to the initial conditions. such as hierarchial clustering or spectral clustering.

Q#2

Solution

A Multilayer feed-forward neural network is a type of artifical neural network that consists of multiple layers of interconnected nodes, where the output of one layer is used as input to next layer. The first layer is called the input layer, the last layer is called output layer and any layer in between are called hidden

M T W T F S DATE: 5/11/2023

layers.

The basic idea behind a feed-forward neural network is that the input signal is propagated through the network in a forward direction with each neuron in a layer receiving weighted input signals from the previous layer. (Three)[x] applying an activation function to the weighted sum of (Σm)[x] its inputs. and passing the result to the next layer.

The topology of a neural network refers to its structure or architecture, including the number of layers, the number of neurons in each layer and the connections between them. Choosing an appropriate network topology depends on the specific problem being solved the size of the dataset, and

(M)(T)(W)(T)(F)(S) DATE: 5/11/2021

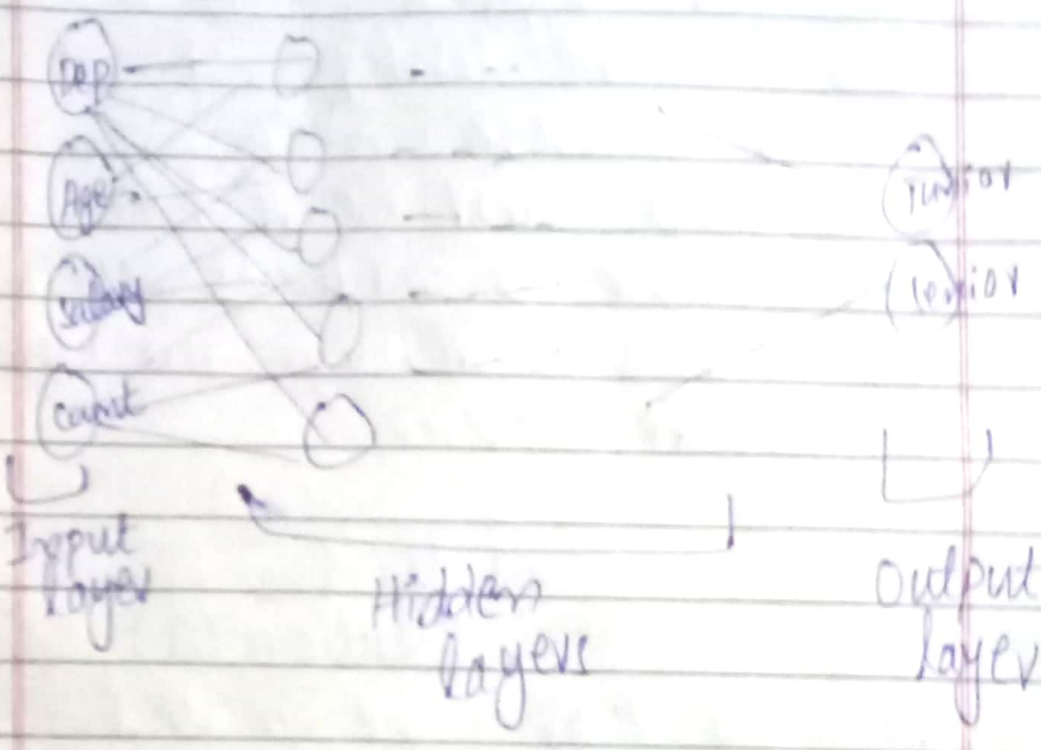
the available computational resources.
There are different approaches to defining the topology of a neural network. including trial and error, expert knowledge and automated methods (such as genetic algorithms and neural architecture search).

Q#01
Solution (a)

The input layer of the neural network will have 4 nodes corresponding to the 4 input attributes "Department", "Age", "Status" "Salary" and "Count". The output layer will have 2 nodes corresponding to the 2 possible values of the class-label attribute "Senior" and "Junior". The hidden layer can have any number of nodes, depending

M T W T F S DATE 05/04/2023

on the complexity of the problem and the amount of training data available.



Q#01

Solution (b)

To show the weight values after one iteration of the backpropagation algorithm. We need to specify the initial values, biases, and learning rate used. Let us assume

M T W T F S DATE: 5/4/2023

the following initial values:
learning rate (α) = 0.1
Bias values $b_1 = 0.5, b_2 = 0.5$
weight values:

$w_1 = [0.1, -0.2, 0.3, 0.4]$
$w_2 = [-0.1, 0.2, -0.3, -0.4]$
$w_3 = [0.2, 0.3, -0.4, -0.1]$
$w_4 = [-0.2, -0.3, 0.4, 0.1]$
$w_5 = [-0.3, 0.1, 0.2, 0.4]$
$w_6 = [0.4, -0.3, 0.2, 0.1]$
$w_7 = [-0.4, 0.3, -0.2, -0.1]$
$w_8 = [0.3, 0.2, -0.1, 0.4]$
$w_9 = [-0.3, -0.2, 0.1, 0.4]$
$w_{10} = [0.1, -0.4, -0.3, 0.2]$

To apply the backpropagation algorithm, we need to first forward propagate the input instance through the network to obtain the output values for each node.
For the input instance "Sales, Senior, ~~Secretary~~ 31 ..." the input layer will be [1, 0, 1, 0]

M T W T F S DATE: 5/4/2023

(corresponding to Sales and 31-35) and the target out values will be [1,0] (corresponding to Senior).

Using the initial weight and bias values, we can calculate the output values for hidden layer and the output layer

Input = [1,0,1,0]

Hidden layer:-

h_1 = Sigmoid(0.5 + (1*0.1) + (0*-0.2)) + (1*0.3) + (0*0.4))

h_2 = Sigmoid(0.5 + (1*-0.1) + (0*0.2) + (1*-0.3)) + (0*0.4)

h_3 = Sigmoid(0.5 + (1*-0.3)) + (0*0.1) + (1*0.2) + (0*-0.4)

~~(Formula)~~

output layer:-

O_1 = Sigmoid(0.5 + (h_1*0.4) + (h_2*-0.3)) + (h_3*0.2)