DATA MINING ASSIGNMENT-4

Parampreet Singh

Answer 1

Given,

Wheat Prod.	30	28	32	25	25	25	22	24	35	40
Flour Price	25	30	27	40	42	40	50	45	30	25

There are $n=10$ points in given dataset

Regression line →

$$Y = \hat{\alpha} + \hat{\beta} X$$

where,

$X \rightarrow$ wheat in tons
$Y \rightarrow$ Price of kilo of flour in pesetas
$\hat{\alpha} \rightarrow$ y-intercept
$\hat{\beta} \rightarrow$ Slope

$$\hat{\alpha} = \frac{1}{n} S_y - \hat{\beta} \frac{1}{n} S_x$$

$$\hat{\beta} = \frac{n S_{xy} - S_x S_y}{n S_{xx} - S_x^2}$$

Hand Calculations →

$S_x = \Sigma x_i = 286$

$S_y = \Sigma y_i = 354$

$S_{xx} = \Sigma x_i^2 = 30^2 + 28^2 + \cdots 40^2 = 8468$

$S_{yy} = \Sigma y_i^2 = 25^2 + 30^2 + \cdots + 25^2 = 13268$

$S_{xy} = \Sigma x_i y_i = (30 \times 25) + (28 \times 30) + \cdots + (40 \times 25) = 9734$

$$\hat{\beta} = \frac{(10 \times 9734) - (286 \times 354)}{(10 \times 8468) - (286)^2} = \frac{-3904}{2884}$$

$$\hat{\beta} = -1.3537$$

$$\hat{\alpha} = \frac{354}{10} - (-1.3537) \times \frac{286}{10}$$

$$\hat{\alpha} = 74.1158$$

Regression line →

$$Y = 74.1158 - 1.3537X$$

Answer 2

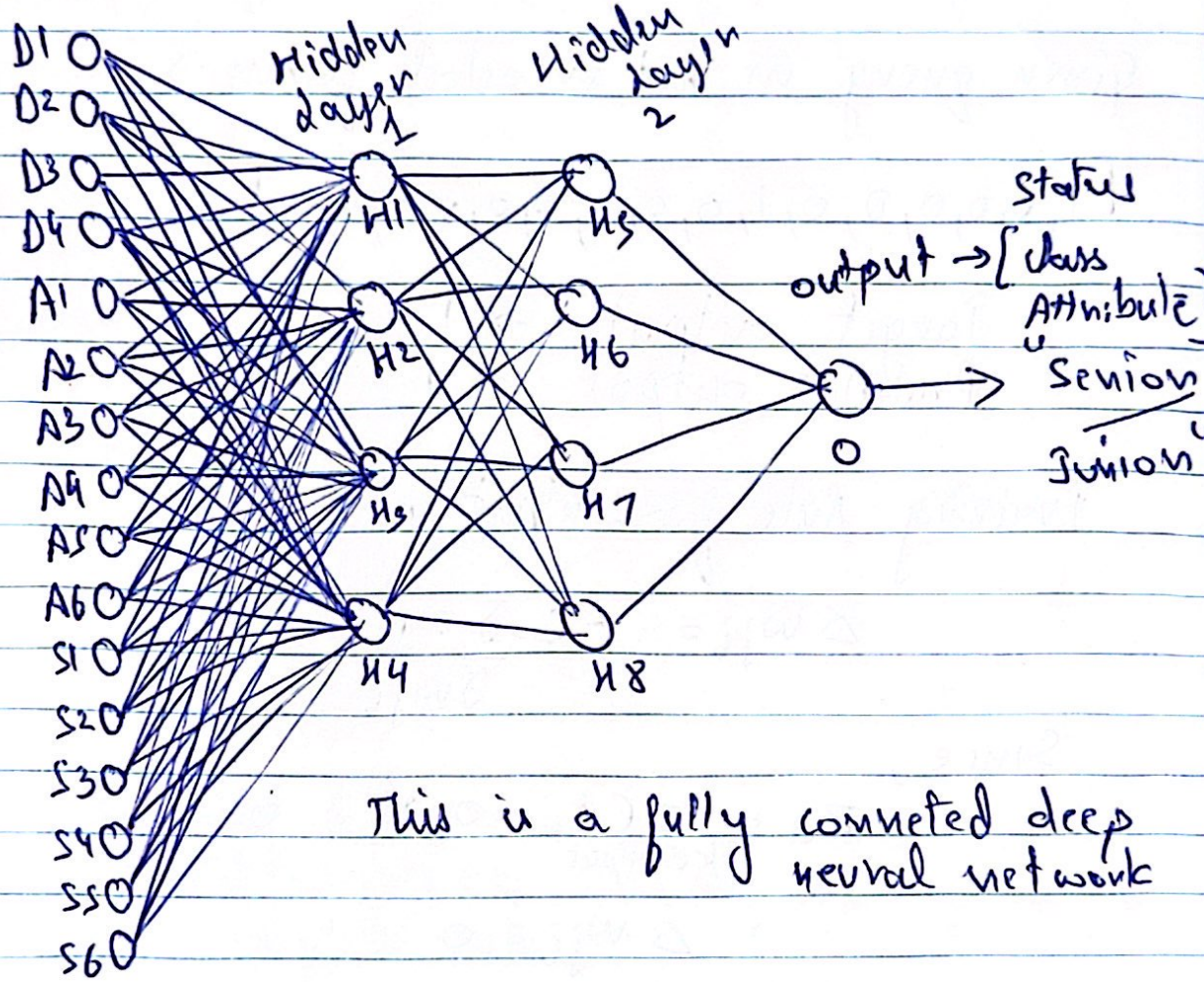
Eager classification → is faster at classification than lazy classification because it constructs a generalization model before receiving any new tuples to classify. weights can be assigned to attributes, which can improve classification accuracy.

Disadvantages of eager classification are that it must commit to a single hypothesis that covers the entire instance space, which can decrease classification, and more time is needed for training.

lazy classification → uses a richer hypothesis space, which can improve classification accuracy. It requires less time for training than eager classification. A disadvantage of lazy classification is that all training tuples need to be stored, which leads to expensive storage costs and requires efficient indexing techniques. Another disadvantage is that it is slower at classification because classifiers are not built until new tuples need to be classified.

Furthermore, attributes are all equally weighted, which can decrease classification accuracy.

Answer3: (a)



This is a fully connected deep neural network

D_1, D_2, D_3, D_4 → Representing 4 departments
$A_1, A_2, A_3, A_4, A_5, A_6$ → Representing 6 Age ranges
$S_1, S_2, S_3, S_4, S_5, S_6$ → Representing 6 salary ranges

All input attributes are 'hot-encoded'

(5) Total weight parameters → $16*4 + 4*4 + 4*1$
$= 84$

→ weights are assigned randomly for initialization

Given query in hot encoded form →

[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]

Target output → 1
Predicted output → 1

Training Rule for output unit

$$\Delta w_{ji} = -C \frac{\partial z_d}{\partial w_{ji}}$$

Since,

$$z_d = \sum_{k \in output} (t_k - o_k)^2 = 0$$

$$\therefore \Delta w_{ji} = 0$$

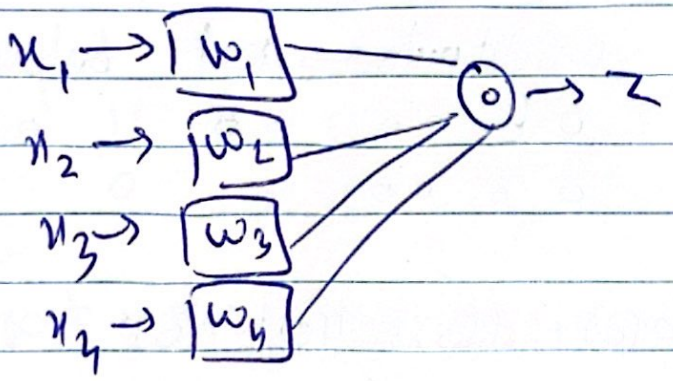
Assumed learning Rate $e = 0.1$
There is no bias for each neuron

Answer:

Perceptron Rule:

Given dataset training is below

Pattern Input				Target values
1	0	0	→	0
0	1	0	→	1
0	0	1	→	1
1	1	0	→	0
1	1	1	→	0
0	0	1	→	0



Learning Rate $c = 1$

initial weights = 0 So, $w_1 = w_2 = w_3 = w_4 = 0$

Back propagation Rule :

$$\Delta w_i = c(t - z)x_i$$

bias = 1

Pattern	Target	weight vector	Net output
1001	0	0 0 0 0	

Net calculation for pattern 1001

$= 1(0) + 0(0) + 0(0) + 0(0)$

$= 0$

output = 0, as Net = 0

rule is output = 1 if net > 0

0 otherwise

Thus here Target = output so there is no error and no changes to Δw, so it will be 0000.

Pattern	Target	weight vector	Net	output	Δw
1001	0	0000	0	0	0000
0111	1	0000			

$Net = 0(0) + 1(0) + 1(0) + 1(0)$

$= 0$

output = 0

$\Delta w = c(t-z)x_i$

$= 1(1-0)x_i - x_i$

$= 0, 1, 1, 1$

$x_1\ x_2\ x_3\ x_4$

$\Delta w = 0, 1, 1, 1$

then add Δw to weight vector

Pattern	Target	weight vector	Net	output	Δw
1001	0	0 0 0 0	0	0	0000
0111	1	0000	0	0	0111
1011	1	0111			

$Net = 1(0) + 0(1) + 1(1) + 1(1) = 2$

output = 1

$\Delta w = 0, 0, 0, 0$

Pattern	Target	weight vector	Net	output	Δw
1001	0	0000	0	0	0000
0111	1	0000	0	0	0111
1011	1	0111	2	1	0000
1101	0	0111			

$Net = 1(0) + 1(1) + 0(1) + 1(1)$

$= 2$

output = 1

As target ≠ output

$\Delta w = c(t-z) x_i$

$= 1(0-1) x_i$

$= -x_i$

$\Delta w = -1, -1, 0, -1$

New weight vector :

$= 0+(-1), 1+(-1), 1+(-1), 1+(-1)$

$= -1, 0, 1, 0$

Pattern	Target	weight vector	Net	output	ΔW
1 0 0 1	0	0 0 0 0	0	0	0 0 0 0
0 1 1 1	1	0 0 0 0	0	0	0 1 1 1
1 0 1 1	1	0 1 1 1	2	1	0 0 0 0
1 1 0 1	0	0 1 1 1	2	1	-1 -1 0 -1
1 1 1 1	0	-1 0 1 0	0		

$$\text{Net} = 1(-1) + 1(0) + 1(1) + 1(0)$$
$$= -1 + 1$$
$$= 0$$

output = 0 ΔW, no change, 0 0 0 0

Pattern	Target	weight vector	Net	output	ΔW
1 0 0 1	0	0 0 0 0	0	0	0 0 0 0
0 1 1 1	1	0 0 0 0	0	0	0 1 1 1
1 0 1 1	1	0 1 1 1	2	1	0 0 0 0
1 1 0 1	0	0 1 1 1	2	1	-1 -1 0 -1
1 1 1 1	0	-1 0 1 0	0	0	0 0 0 0
0 0 1 1	0	-1 0 1 0			

$$\text{Net} = 0(-1) + 0(0) + 1(1) + 0(1)$$
$$= 1$$

output = 1

$$\Delta W = 1(0-1)x_i$$
$$= -x_i$$

$$\Delta W = 0 \quad 0 \quad -1 \quad -1$$

Final table :-

Pattern	Target	weight vector	Net	output	ΔW
1 0 0 1	0	0 0 0 0	0	0	0 0 0 0
0 1 1 1	1	0 0 0 0	0	0	0 1 1 1
1 0 1 1	1	0 1 1 1	2	1	0 0 0 0
1 1 0 1	0	0 1 1 1	2	1	-1 -1 0 -1
1 1 1 1	0	-1 0 1 0	0	0	0 0 0 0
0 0 1 1	0	-1 0 1 0	1	1	0 0 -1 -1

Answer:

To design a CNN to classify human faces into categories of young, middle age, and elderly, I'll use the following layers and functionalities :-

1. Input layer :- The first layer must takes the face images. They should be normalized to the same size and pixel range (0-1).

2. Convolutional layers :- They apply a number of filters to the input to create feature maps. Their purpose is to detect low level features like edges in early layers and complex features like facial parts in deeper layers.

3. Activation layer (ReLU) :- Apply a ReLU activation after each Conv layer to introduce non-linearity, allowing the model to learn more complex patterns.

4. Pooling layer (Max Pooling) :- This layer reduces the spatial dimension of the output from conv layer. It helps reduce computation

and controls overfitting by summarizing the features present in a region of the feature map.

5:- Fully connected layer :- After several conv and pooling layers, the high level reasoning in NN is done via FC layers.

6:- output layer :- The final layer, often another FC layer, has the number of neurons corresponding to the number of classes, in this case, three (youth, middle age, elderly). It is coupled with a softmax activation function that gives the probability distribution over the classes.

The exact architecture, including the no. of layers, the size of filters, and the no. of filters in each conv layer, would need to be determined based on the complexity of the task and amount of the data available. The model would then be trained using a labelled dataset with backpropagation and an appropriate loss function, like cross entropy, to adjust the weights.

Answer6:

MLP Vs CNN

- Structure :- MLP is a basic form of NN where each neuron is fully connected to all the neurons in the next layer. CNN, on the other hand, is designed to recognize visual patterns directly from pixel images with minimal preprocessing. It uses convolutional layer that create a convolutional kernel that is passed over the image to detect features.

- Parameter sharing :- CNNs use parameter sharing, which means the same filter is applied to different parts of the input image, significantly reducing the no. of parameters and computational complexity.

- Feature Detection :- CNNs automatically detect important features without any human supervision required, whereas MLPs require feature extraction to be done manually.

Advantages of CNN over traditional methods:-

CNNs are capable of automatic feature extraction, making them more efficient at image classification tasks. They are also invariant to the location of features in the input image due to the pooling layers.

Answer 7.

The major differences b/w CNN, RNN and GAN are:

- Design Purpose: CNNs are designed primarily for tasks like image recognition and processing where spatial hierarchies of features are important. RNNs are designed to handle sequential data, for tasks where context and temporal sequence is important, such as language modelling and translation. GANs consist of 2 networks, a generator and a discriminator, that are trained together. The generator creates data that is as realistic as possible, and the discriminator evaluates its authenticity.

- Data Handling:- CNNs handle fix size vector inputs, while RNNs can handle sequences of

varying lengths.

- Memory :- RNNs have memory and can remember previous input in its internal state, which is not the case with CNNs.

- Training :- GANs require a different training process where two networks are trained in a game-theoretic approach, which is not applicable to CNNs or RNNs.