Ayush Chanchal

Sap:500097569

Batch: B6

EXPERIMENT: 04

Q). Explain the Hebbian Learning Rule. Implement the same in Python.

: - Hebbian Learning- Hebbian learning is an important idea in both brains and artificial neural networks. It explains how when two neurons are active at the same time, their connection gets stronger. This is like how we remember things in our brains. It's also used in making computer models that learn. So, Hebbian learning helps us understand how our brains learn and remember, and it's also useful for making smart computer programs.

Source code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# dataset = pd.read\_csv('IRIS.csv')

# print(dataset.head())

num\_inputs = int(input("Enter the number of inputs: "))

num\_samples = int(input("Enter the number of training samples: "))

train\_set = [[int(input()) for x in range(num\_inputs)]

for y in range(num\_samples)]

C = float(input("Enter the value of learning constant: "))

weights = [1, -1]

def sign\_function(input\_value):

return (1 if input\_value >= 0 else -1)

for iteration in range(len(train\_set)):

net\_value = 0

for i in range(len(weights)):

net\_value += weights[i] \* train\_set[iteration][i]

signed\_value = sign\_function(net\_value)

delta = [C\*signed\_value\*x for x in train\_set[iteration]]

for j in range(len(weights)):

weights[j] = weights[j] + delta[j]

print(weights)

Screenshot:

