

BARI ANKIT (56)

Exp – 5 : Hebbian Learning

Code :

```
# %%
```

```
# BARI ANKIT (56)
```

```
# %%
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# %%
```

```
learning_rate = 0.01
```

```
epochs = 100
```

```
# %%
```

```
X = np.array([
```

```
    [0, 0],
```

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

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

```
    [1, 1]
```

```
])
```

```
y = np.array([0, 0, 0, 1])
```

```
# %%
```

```
weights = np.random.rand(2)

bias = np.random.rand()

# %%

def activation_function(x):
    return 1 if x >= 0 else 0

# %%

for epoch in range(epochs):
    for i in range(len(X)):
        input_vector = X[i]
        target = y[i]

        weighted_sum = np.dot(weights, input_vector) + bias

        output = activation_function(weighted_sum)

        delta_w = learning_rate * input_vector * output
        delta_b = learning_rate * output

        weights += delta_w
        bias += delta_b

# %%

print("Trained Weights:", weights)
print("Trained Bias:", bias)
```

```
# %%  
  
for i in range(len(X)):  
    input_vector = X[i]  
    weighted_sum = np.dot(weights, input_vector) + bias  
    output = activation_function(weighted_sum)  
    print(f"Input: {input_vector}, Predicted Output: {output}")  
  
# %%
```