

Name: Prajjwal Vishwakarma

Branch: CSE(DS)

Roll No: 63

Sub.: Deep Learning

Practical No. 02

Implementing XOR in Deep learning using python

Code:

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
Y = np.array([[0], [1], [1], [0]])
model = Sequential()
model.add(Dense(8, input_dim=2, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, Y, epochs=1000, verbose=0)
loss, accuracy = model.evaluate(X, Y)
print(f"Loss: {loss:.4f}, Accuracy:{accuracy:.4f}")
predictions = model.predict(X)
rounded_predictions = np.round(predictions)
print("Predictions:")
print(rounded_predictions)
```

Output:

+ Code+ TextAll changes saved

RAMDisk

Q

✓

[3] import numpy as np
from keras.models import Sequential
from keras.layers import Dense

{x}

✓

[12] X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
Y = np.array([[0], [1], [1], [0]])

✓

[13] model = Sequential()

✓

[14] model.add(Dense(8, input_dim=2, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, Y, epochs=1000, verbose=0)
loss, accuracy = model.evaluate(X, Y)

1/1 [=====] - 0s 127ms/step - loss: 0.2629 - accuracy: 1.0000

✓

[15] print(f"Loss: {loss:.4f}, Accuracy: {accuracy:.4f}")

Loss: 0.2629, Accuracy: 1.0000

✓

[16] predictions = model.predict(X)
rounded_predictions = np.round(predictions)
print("Predictions:")
print(rounded_predictions)

1/1 [=====] - 0s 52ms/step
Predictions:
[[0.]
 [1.]
 [1.]
 [0.]]

<>

0s

completed at 10:33 AM

● x