

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([[1, 0], [4, 7], [8, 1], [2, 5]])
Y = np.array([[1], [4], [2], [6]])
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 + Text All changes saved

RAM
Disk

Q

✓

0s

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

{X}

1/1

✓

0s

[4] X = np.array([[1, 0], [4, 7], [8, 1], [2, 5]])
Y = np.array([[1], [4], [2], [6]])

1/1

✓

0s

[5] model = Sequential()

1/1

✓

7s

[6] 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 155ms/step - loss: -60.6462 - accuracy: 0.2500

1/1

✓

0s

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

Loss: -60.6462, Accuracy: 0.2500

1/1

✓

0s

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

<>

1/1 [=====] - 0s 84ms/step
Predictions:
[[1.]
 [1.]
 [1.]
 [1.]]

0s

completed at 9:58 AM