

Name: Paritosh

CSE(DS)

Roll No: 72

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:

DL_pract1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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)
```

1/1 [=====] - 0s 127ms/step - loss: -205.9017 - accuracy: 0.2500
Loss: -205.9017, Accuracy: 0.2500
1/1 [=====] - 0s 51ms/step
Predictions:
[[1.]
[1.]
[1.]
[1.]]

3s completed at 10:28 AM

83°F Mostly cloudy 10:42 AM 8/3/2023