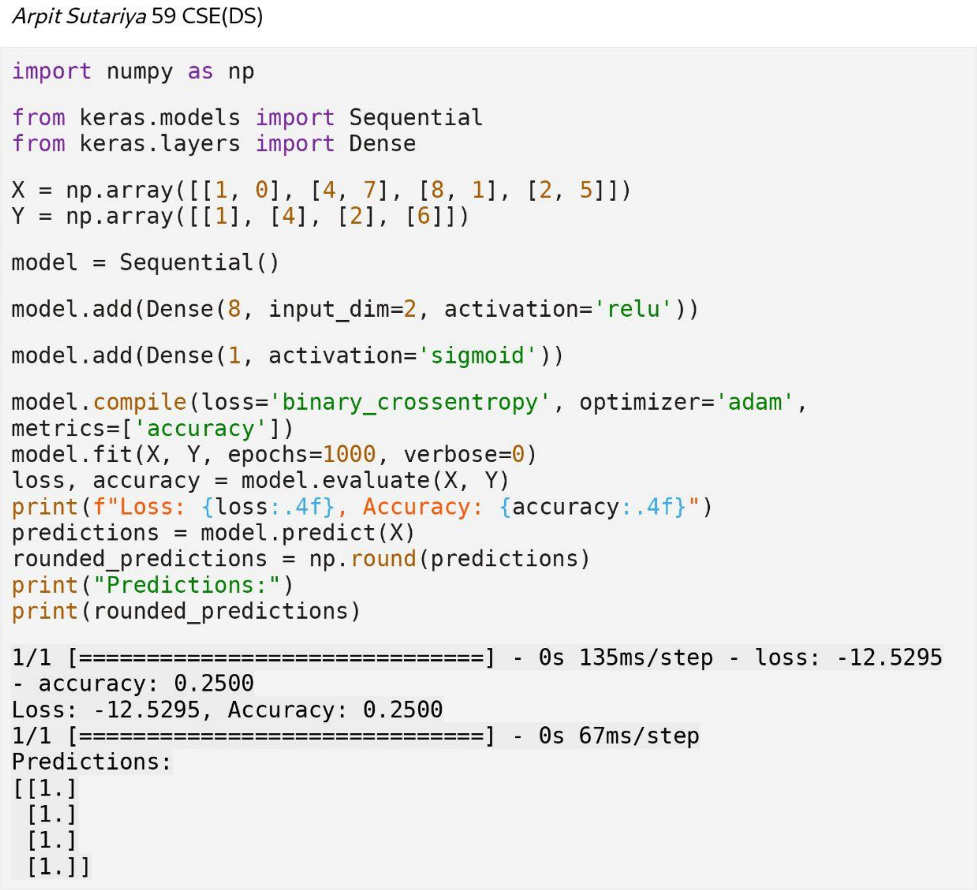
Name: Arpit Sutariya
Roll no.: 59
DL Experiment 2

Implementing XOR in Deep learning using python

Code:

```
#Importing all the libraries
import numpy as np
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
#Input and Output data
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([[0], [1], [1], [0]])
# Creating a sequential model
model = Sequential()
# Adding an input layer with two units
model.add(Dense(2, input_dim=2, activation='sigmoid'))
# Add a hidden layer with two units
model.add(Dense(2, activation='sigmoid'))
# Add the output layer with one unit
model.add(Dense(1, activation='sigmoid'))
# Compile the model
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
# Training the model
model.fit(X, y, epochs=10000, verbose=0)
# Testing the model on XOR inputs
predictions = model.predict(X)
rounded_predictions = np.round(predictions)
print("Predictions:")
print(rounded_predictions)
```

Output:



Arpit Sutariya 59 CSE(DS)

```
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 135ms/step - loss: -12.5295
- accuracy: 0.2500
Loss: -12.5295, Accuracy: 0.2500
1/1 [==============================] - 0s 67ms/step
Predictions:
[[1.]
 [1.]
 [1.]
 [1.]]
```