```
#Akash Adarkar 68
#XOR Implementations


#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)
```

```
<keras.callbacks.History at 0x7cdb6aff4b80>
```

```
# Testing the model on XOR inputs
predictions = model.predict(X)
rounded_predictions = np.round(predictions)
print("Predictions:")
print(rounded_predictions)
```

```
1/1 [==============================] - 0s 150ms/step
Predictions:
[[0.]
 [1.]
 [1.]
 [1.]]
```