EX NO : 08

DATE : 16.05.2022

XOR GATE IMPLEMENTATION

Aim:

To implement multi layer artificial neural network using back propagation algorithm.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner /Google Colab

Related Theory Concept:

Algorithm

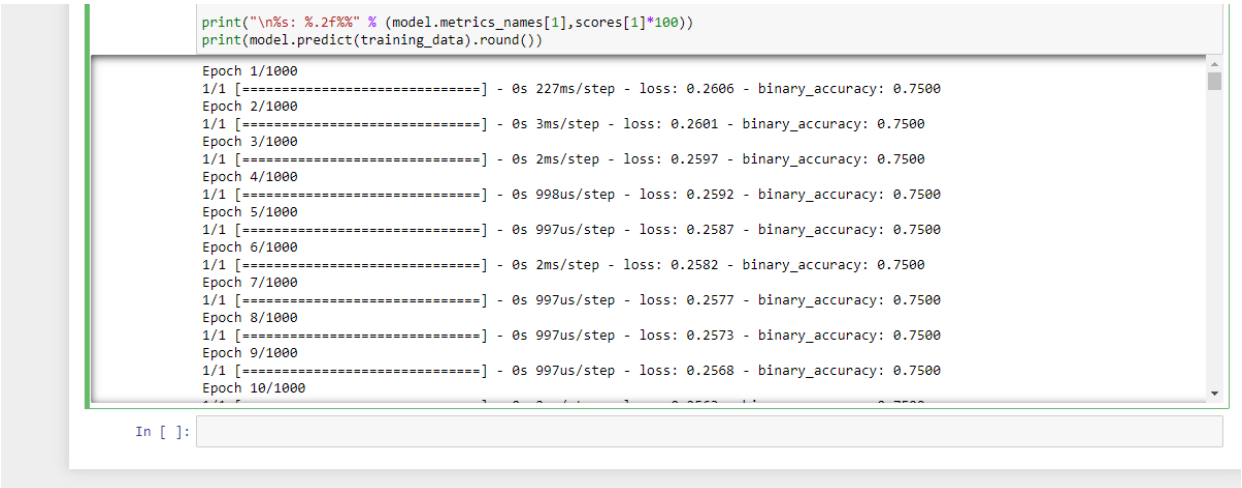
1. Import the required libraries.
2. Create the training dataset.

3. Create the neural network model with one hidden layer.
4. Train the model with training data.
5. Now test the model with testing data.

Program:

```
/*
Program to implement XOR Logic Gate.
Developed by   : Monica M
RegisterNumber :  212219040082
*/

import numpy as np
from keras.models import Sequential
from keras.layers.core import Dense

training_data=np.array([[0,0],[0,1],[1,0],[1,1]],"float32")
target_data=np.array([[0],[1],[1],[0]],"float32")

model=Sequential()
model.add(Dense(16,input_dim=2,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='mean_squared_error',
                    optimizer='adam',
                    metrics=['binary_accuracy'])
model.fit(training_data,target_data,epochs=1000)
scores=model.evaluate(training_data,target_data)

print("\n%s: %.2f%%" % (model.metrics_names[1],scores[1]*100))
print(model.predict(training_data).round())
```

Output:



Result:

Thus the python program successully implemented XOR logic gate.