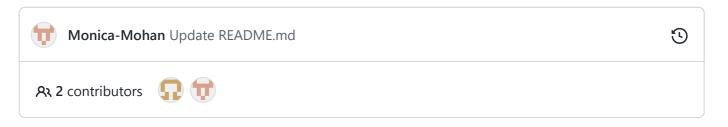
Y Monica-Mohan / XOR-GATE-IMPLEMENTATION Public

forked from HEMALATHA2021/XOR-GATE-IMPLEMENTATION

Code Pull requests Actions Projects Wiki Security Insights Settings



XOR-GATE-IMPLEMENTATION / README.md



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:

```
print("\n%s: %.2f%%" % (model.metrics_names[1],scores[1]*100))
print(model.predict(training_data).round())
        Epoch 1/1000
                          =======] - 0s 3ms/step - loss: 0.2601 - binary accuracy: 0.7500
        Epoch 3/1000
        1/1 [=====
Epoch 4/1000
                                            0s 2ms/step - loss: 0.2597 - binary_accuracy: 0.7500
                                            0s 998us/step - loss: 0.2592 - binary_accuracy: 0.7500
        1/1 [=====
Epoch 6/1000
                                =======] - 0s 997us/step - loss: 0.2587 - binary accuracy: 0.7500
                                  1/1 [======
Epoch 7/1000
        1/1 [=====
Epoch 8/1000
                                  ======] - 0s 997us/step - loss: 0.2577 - binary_accuracy: 0.7500
        1/1 [=====
Epoch 9/1000
                               ======== 1 - 0s 997us/step - loss: 0.2573 - binary accuracy: 0.7500
                            =======] - 0s 997us/step - loss: 0.2568 - binary accuracy: 0.7500
        1/1 [======
Epoch 10/1000
In [ ]:
```

Result:

Thus the python program successully implemented XOR logic gate.