Name:Rameshwar Pawar

RollNo:46

**XOR implementation using deep learning in python Code:**

import

numpy

as

np

from

keras.models

import

Sequential

from

keras.layers

import

Dense

#

XOR

input

data

X

=

np.array([[0,

0]

,

[0

,

1]

,

[1

,

0]

,

[1

,

1]])

#

Corresponding

XOR

output

data

Y

=

np.array([[0],

[1]

,

[1]

,

[0]])

#

Create

a

sequential

model

model

=

Sequential()

#

Add

a

hidden

layer

with

8

neuro

ns

and

'relu'

activation

function

model.add(Dense(8,

input\_dim=2,

activation='relu'))

#

Add

the

output

lay

er

with

1

neur

on

and

'sigmoid'

activation

function

model.add(Dense(1,

activation='si

gmoid'))

#

Compile

the

model

using

binary

cross-entropy

loss

and

Adam

optimizer

model.compile(loss=

'binary\_cross

entropy',

optimizer='adam',

metrics=['accuracy'])

#

Train

the

model

for

1000

epoch

s

model.fit(X,

Y,

epoch

s=1000,

verbose=0)

#

Evaluate

the

mode

l

loss,

accuracy

=

model.evaluate(X,

Y)

print(f"Loss:

{

loss:.4f},

Accuracy:

{

accuracy:.4f}"

)

#

Make

predictions

predictions

=

mod

el.predict(X)

rounded\_predictio

ns

=

np.round(predictions)

print("Predictions:"

)

print(rounded\_predictions)

Output:

