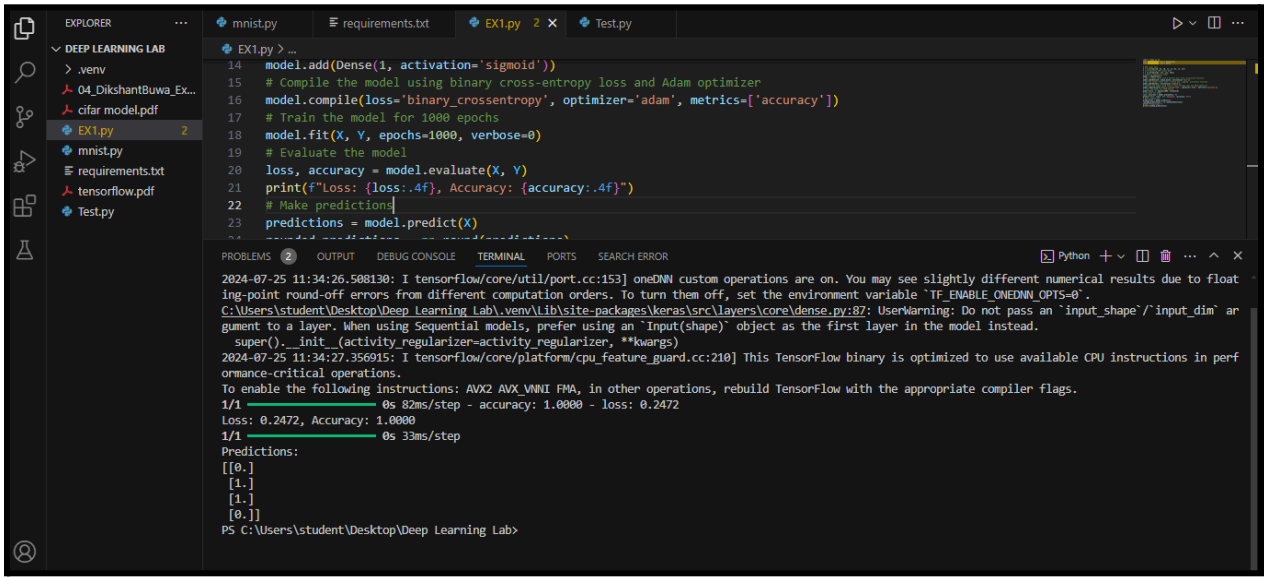
Name: Ramchandra Darade
Roll No: 69
Ex1:XOR implementation using deep learning in python

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 neurons and 'relu' activation function
model.add(Dense(8, input_dim=2, activation='relu'))
# Add the output layer with 1 neuron and 'sigmoid' activation function
model.add(Dense(1, activation='sigmoid'))
# Compile the model using binary cross-entropy loss and Adam optimizer
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# Train the model for 1000 epochs
model.fit(X, Y, epochs=1000, verbose=0)
# Evaluate the model
loss, accuracy = model.evaluate(X, Y)
print(f"Loss: {loss:.4f}, Accuracy: {accuracy:.4f}")
# Make predictions
predictions = model.predict(X)
rounded_predictions = np.round(predictions)
print("Predictions:")
print(rounded_predictions)
```

Output:


EXPLORER
DEEP LEARNING LAB
.venv
04_DikshantBuwa_Ex...
cifar model.pdf
EX1.py 2
mnist.py
requirements.txt
tensorflow.pdf
Test.py
mnist.py
requirements.txt
EX1.py 2
Test.py
EX1.py > ...
14 model.add(Dense(1, activation='sigmoid'))
15 # Compile the model using binary cross-entropy loss and Adam optimizer
16 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
17 # Train the model for 1000 epochs
18 model.fit(X, Y, epochs=1000, verbose=0)
19 # Evaluate the model
20 loss, accuracy = model.evaluate(X, Y)
21 print(f"Loss: {loss:.4f}, Accuracy: {accuracy:.4f}")
22 # Make predictions
23 predictions = model.predict(X)
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SEARCH ERROR
Python
2024-07-25 11:34:26.508130: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
C:\Users\student\Desktop\Deep Learning Lab\.venv\lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2024-07-25 11:34:27.356915: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 0s 82ms/step - accuracy: 1.0000 - loss: 0.2472
Loss: 0.2472, Accuracy: 1.0000
1/1 0s 33ms/step
Predictions:
[[0.]
[1.]
[1.]
[0.]]
PS C:\Users\student\Desktop\Deep Learning Lab>