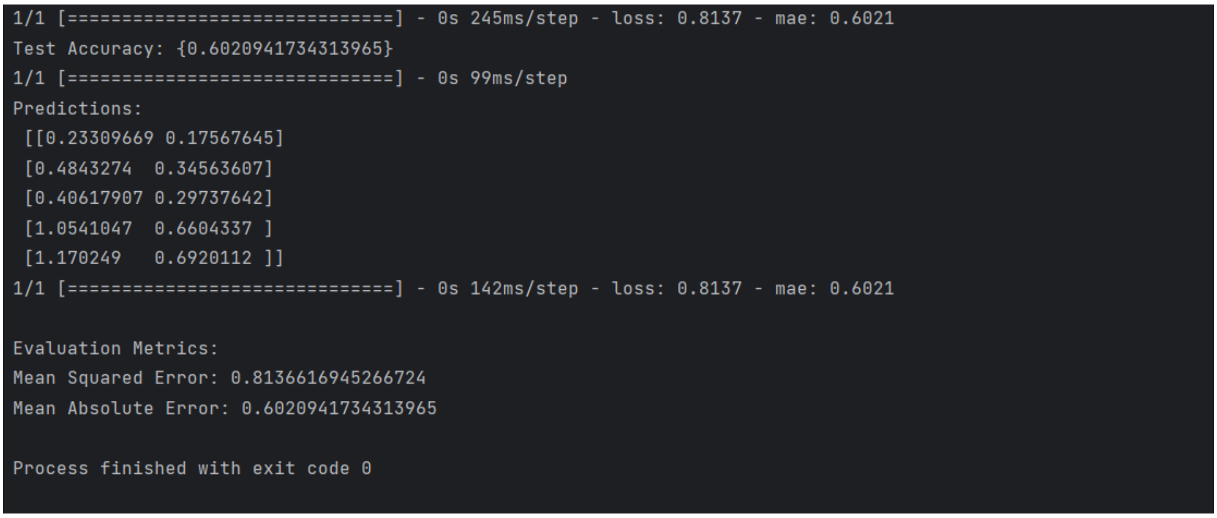
Neural Networks Final Exam Report

Firstly, in order to collect data, me and my friend have gone to the university and prepared a coin dropping environment. We have dropped the coin a total of 105 times. The first being 35 times as heads looking up, 35 times as tails looking up and 35 times as vertical orientation. Then I organized the data into an csv formatted file. I coded Heads as 0 and Tails as 1 and Heads facing up orientation as 0, Tails facing up orientation as 1, vertical orientation as 2. I have chosen the architecture to be a simple feedforward neural network. In the process of coding, I have encountered some challenges. One of them is to create a two output neural network and adjusting the inputs and outputs accordingly. The evaluation metrics can be seen below:



```
1/1 [==============================] - 0s 245ms/step - loss: 0.8137 - mae: 0.6021
Test Accuracy: {0.6020941734313965}
1/1 [==============================] - 0s 99ms/step
Predictions:
 [[0.23309669 0.17567645]
 [0.4843274  0.34563607]
 [0.40617907 0.29737642]
 [1.0541047  0.6604337 ]
 [1.170249   0.6920112 ]]
1/1 [==============================] - 0s 142ms/step - loss: 0.8137 - mae: 0.6021

Evaluation Metrics:
Mean Squared Error: 0.8136616945266724
Mean Absolute Error: 0.6020941734313965

Process finished with exit code 0
```

These are the numerical data on the csv file, ready to be used:

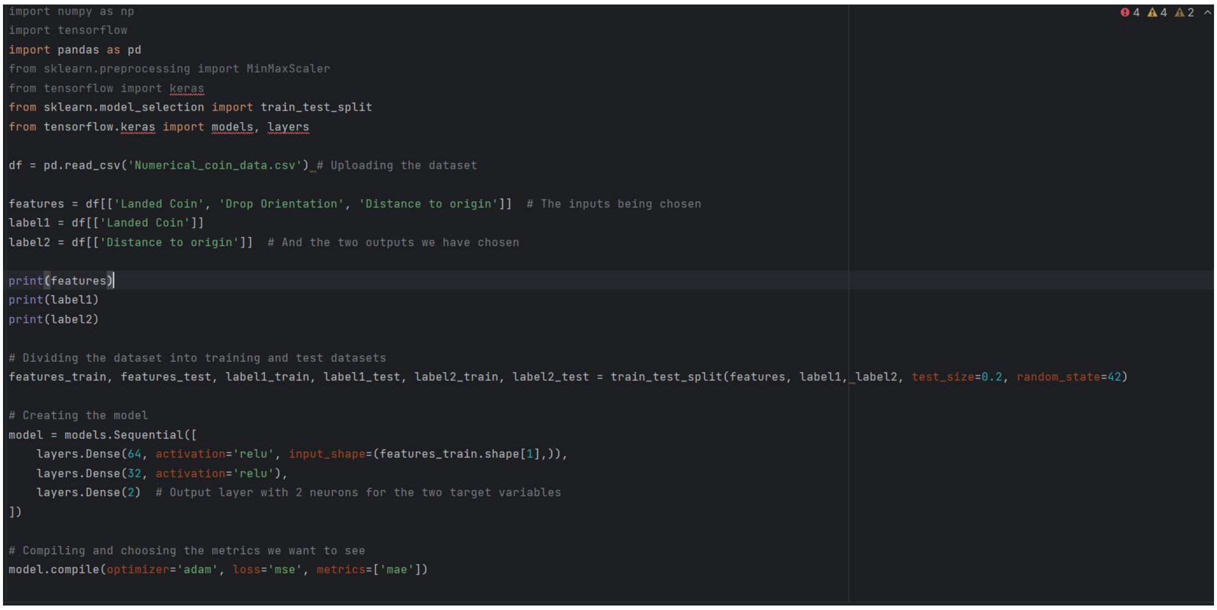
Drop Number,Landed Coin,Drop Orientation,Distance to origin
1,1,0,17
2,1,0,25
3,1,0,13
4,1,0,37
5,0,0,42
6,1,0,11
7,0,0,17
8,1,0,6
9,0,0,8
10,1,0,4
11,0,0,12
12,1,0,9
13,0,0,7
14,1,0,14
15,0,0,63
16,0,0,11
17,0,0,23
18,0,0,7
19,0,0,1
20,0,0,31
21,0,0,36
22,0,0,20
23,1,0,9
24,0,0,7
25,0,0,2
26,1,0,43
27,0,0,3
28,0,0,43
29,0,0,8
30,1,0,35
31,1,0,3
32,1,0,54

33,0,0,60
34,1,0,21
35,1,0,7
36,1,1,26
37,1,1,123
38,1,1,7
39,1,1,13
40,0,1,7
41,1,1,17
42,1,1,24
43,1,1,7
44,0,1,11
45,1,1,12
46,0,1,65
47,0,1,7
48,0,1,19
49,1,1,12
50,0,1,9
51,1,1,4
52,1,1,5
53,1,1,11
54,1,1,50
55,1,1,4
56,1,1,10
57,1,1,23
58,1,1,4
59,1,1,3
60,0,1,1
61,1,1,11
62,1,1,45
63,1,1,8
64,0,1,11
65,1,1,5

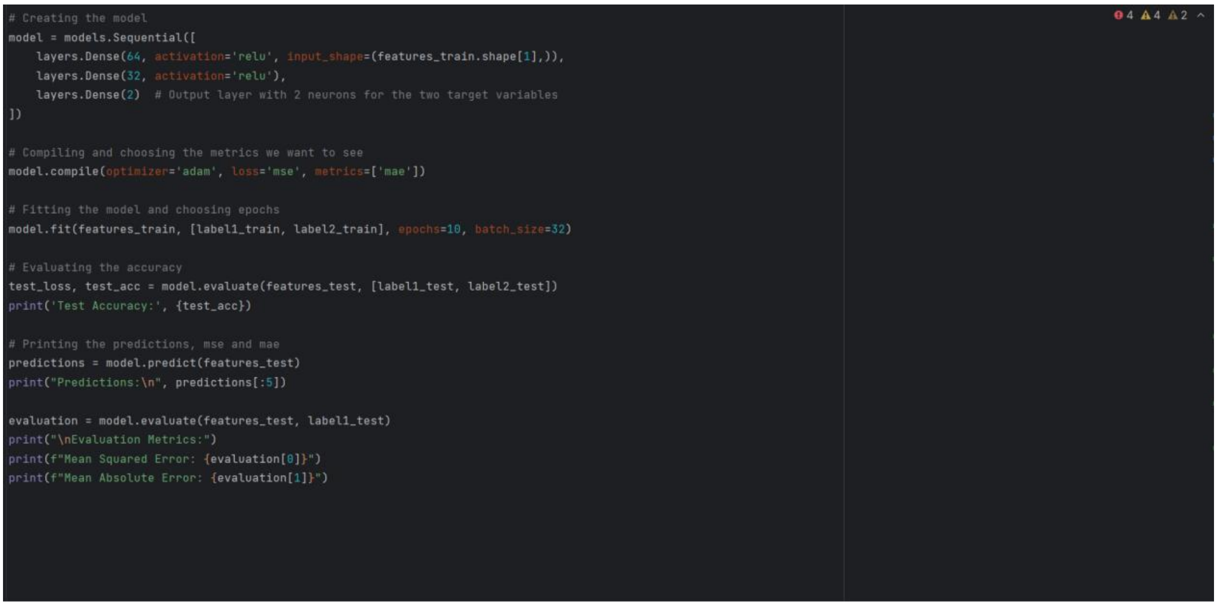
66,1,1,9
67,0,1,3
68,0,1,33
69,1,1,8
70,1,1,10
71,1,2,13
72,0,2,22
73,0,2,40
74,1,2,12
75,1,2,3
76,1,2,54
77,1,2,46
78,1,2,17
79,1,2,31
80,0,2,4
81,1,2,14
82,0,2,27
83,0,2,39
84,1,2,10
85,0,2,59
86,1,2,13
87,0,2,10
88,0,2,58
89,1,2,11
90,0,2,10
91,1,2,27
92,0,2,28
93,0,2,62
94,1,2,63
95,0,2,35
96,1,2,14
97,0,2,271
98,1,2,229

99,1,2,15
100,0,2,28
101,0,2,22
102,0,2,3
103,1,2,6
104,1,2,7
105,0,2,10

In here are the screenshots of the code:



```python
import numpy as np
import tensorflow
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from tensorflow import keras
from sklearn.model_selection import train_test_split
from tensorflow.keras import models, layers

df = pd.read_csv('Numerical_coin_data.csv') # Uploading the dataset

features = df[['Landed Coin', 'Drop Orientation', 'Distance to origin']]  # The inputs being chosen
label1 = df[['Landed Coin']]
label2 = df[['Distance to origin']]  # And the two outputs we have chosen

print(features)
print(label1)
print(label2)

# Dividing the dataset into training and test datasets
features_train, features_test, label1_train, label1_test, label2_train, label2_test = train_test_split(features, label1, label2, test_size=0.2, random_state=42)

# Creating the model
model = models.Sequential([
    layers.Dense(64, activation='relu', input_shape=(features_train.shape[1],)),
    layers.Dense(32, activation='relu'),
    layers.Dense(2)  # Output layer with 2 neurons for the two target variables
])

# Compiling and choosing the metrics we want to see
model.compile(optimizer='adam', loss='mse', metrics=['mae'])
```



```python
# Creating the model
model = models.Sequential([
    layers.Dense(64, activation='relu', input_shape=(features_train.shape[1],)),
    layers.Dense(32, activation='relu'),
    layers.Dense(2)  # Output layer with 2 neurons for the two target variables
])

# Compiling and choosing the metrics we want to see
model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Fitting the model and choosing epochs
model.fit(features_train, [label1_train, label2_train], epochs=10, batch_size=32)

# Evaluating the accuracy
test_loss, test_acc = model.evaluate(features_test, [label1_test, label2_test])
print('Test Accuracy:', {test_acc})

# Printing the predictions, mse and mae
predictions = model.predict(features_test)
print("Predictions:\n", predictions[:5])

evaluation = model.evaluate(features_test, label1_test)
print("\nEvaluation Metrics:")
print(f"Mean Squared Error: {evaluation[0]}")
print(f"Mean Absolute Error: {evaluation[1]}")
```

References

https://www.youtube.com/watch?v=tPYj3fFJGjk&t=6960s

Eren Kuş 040180043