

112學年專題 - Feature #47

Feature # 60 (Closed): 深度學習模型

深度學習模型_DNN模型訓練tensorflow

2023-09-25 05:03 - Chifu Chung

狀態:	Closed	開始日期:	2023-09-25
優先權:	Normal	完成日期:	2023-10-31
被分派者:	宏益 廖	完成百分比:	100%
分類:		預估工時:	0:00 小時
版本:		耗用工時:	0:00 小時
概述			
相關的議題清單:			
關聯至	硬體組 - Task #231: AWPPG P type 連續裝置製作	New	2024-09-02

歷史

#1 - 2023-09-25 05:08 - Chifu Chung

- 狀態 從 New 變更為 In Progress

#2 - 2023-09-25 05:12 - Chifu Chung

- 完成日期 設定為 2023-10-31

#3 - 2023-09-25 05:20 - Chifu Chung

- 父議題 設定為 #60

#4 - 2023-10-03 05:54 - 宏益 廖

- 檔案 Training Data_A1.xlsx 已新增

- 檔案 Test Data_A1.xlsx 已新增

#5 - 2023-10-03 06:17 - 宏益 廖

- 檔案 clipboard-202310031413-u33lo.png 已新增

- 檔案 clipboard-202310031414-ng4wq.png 已新增

*建立環境 >> Colab

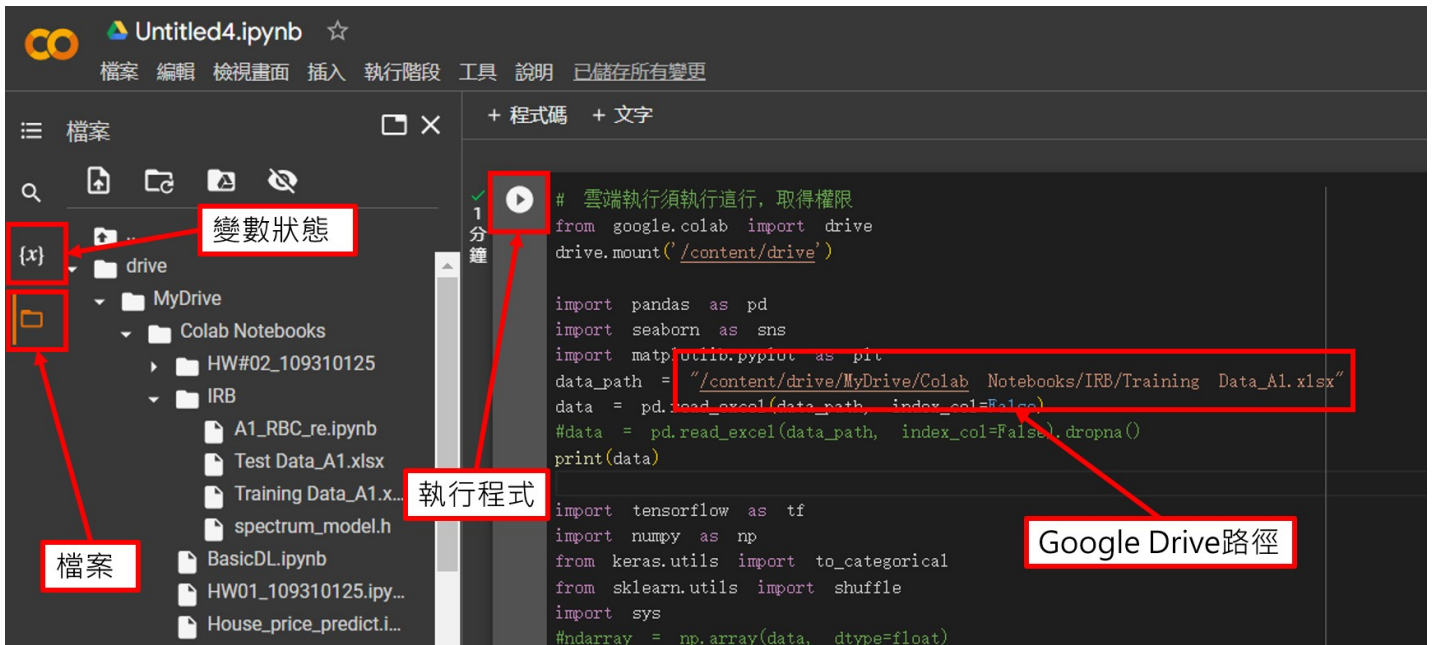
優點:

可連結Google Drive使用的IDE

不需自己建置環境

適合新手使用

(若需更加專業的環境，請google查詢 "Anaconda Jupyter Notebook" 建置虛擬環境)



從上方工具列：執行階段>變更執行階段類型 可更改成不同的硬體(CPU、GPU、TPU)



檔案可從上方下載













Feature #60: 深度學習模型

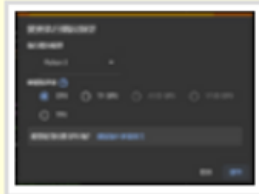
深度學習模型_DNN模型訓練tensorflow

是由 Chifu Chung 於 8 天 前加入, 於 1 分鐘 前更新.

狀態: In Progress
優先權: Normal
被分派者: 宏益 廖

檔案

 Training Data_A1.xlsx (762 KB) 	宏益 廖, 2023-10-03 05:49 
 Test Data_A1.xlsx (943 KB) 	宏益 廖, 2023-10-03 05:53 
 clipboard-202310031413-u33lo.png (895 KB) 	宏益 廖, 2023-10-03 06:13 
 clipboard-202310031414-ng4wq.png (26.9 KB) 	宏益 廖, 2023-10-03 06:14 



*模型訓練

```
# 雲端執行須執行這行, 取得權限
from google.colab import drive
drive.mount('/content/drive')
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data_path = "/content/drive/MyDrive/Colab Notebooks/IRB/Training Data_A1.xlsx"
data = pd.read_excel(data_path, index_col=False)
#data = pd.read_excel(data_path, index_col=False).dropna()
print(data)
```

```
import tensorflow as tf
import numpy as np
from keras.utils import to_categorical
from sklearn.utils import shuffle
import sys
#ndarray = np.array(data, dtype=float)
ndarray = data.values
ndarray = shuffle(ndarray)
print(ndarray)
df_Features = ndarray[:,3:78:3]
print(df_Features)
RBC_Label = ndarray[:,125]
# np.set_printoptions(threshold=sys.maxsize)
# print(data['RBC'])
RBC_Label[RBC_Label==2]=1
print(np.sum(RBC_Label))
```

```

print(RBC_Label)
# one hot encoding
labels = to_categorical(RBC_Label, num_classes=2)
print(labels)

'''

from keras import backend as K
#labelss = np.array([np.array(val) for val in labels])
labelss = K.cast_to_floatx(RBC_Label)
print(labelss)
'''

from tensorflow.keras import layers, Sequential
from keras.layers import Dense, Input
from keras.layers import Dropout, BatchNormalization

#define the model

model=None
model = Sequential()
model.add(Input(shape=(25,)))
#model.add(Dense(1024, activation='relu'))
#model.add(Dense(512, activation='relu'))
#model.add(Dense(256, activation='relu'))
#model.add(Dense(128, activation='relu'))
#model.add(Dense(64, activation='relu'))
#model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(4, activation='relu'))
#model.add(Dense(16, activation=keras.layers.LeakyReLU(alpha=0.01)))
# model.add(BatchNormalization())
# model.add(Dropout(0.2))
model.add(Dense(2, activation='sigmoid'))
#model.add(Dense(2, activation='softmax'))
#model.add(Dense(1))

print(model.summary())

from sklearn.preprocessing import MinMaxScaler, normalize
from tensorflow import keras

# min-max
scalar = MinMaxScaler()
std_df_Features = scalar.fit_transform(df_Features)
# std_df_Features=std_df_Features.T
# print(df_Features)
# print(std_df_Features)
# std_df_Features = normalize(df_Features, norm='l2')
print(std_df_Features)

# Create an optimizer with a specific learning rate
optimizer = keras.optimizers.Adam(learning_rate=0.0005)
#optimizer = keras.optimizers.SGD(0.2, momentum=0.1)

# compile the model
model.compile(loss = 'binary_crossentropy', optimizer = optimizer, metrics=['accuracy'])
#model.compile(loss = 'categorical_crossentropy', optimizer = optimizer, metrics=['accuracy'])

# X = np.asarray(std_df_Features).astype(np.float32)
train_history=None
train_history=model.fit(std_df_Features, labels, validation_split=0.2, epochs=1000, batch_size=128, verbose=2)

def show_train_history(train_history,train,validation):
    plt.plot(train_history.history[train])
    plt.plot(train_history.history[validation])
    plt.title('Train History')
    plt.ylabel(train)
    plt.xlabel('Epoch')
    plt.legend(['train', 'validation'], loc='upper left')
    plt.show()

show_train_history(train_history,'accuracy','val_accuracy')
show_train_history(train_history,'loss','val_loss')

```

```
from tinymlgcn import port
c_code = port(model, optimize=True)
with open('/content/drive/MyDrive/Colab Notebooks/IRB/spectrum_model.h', 'w') as f:
    f.write(c_code)
```

*模型驗證

```
test_data_path = "/content/drive/MyDrive/Colab Notebooks/IRB/Test Data_A1.xlsx"
testdata = pd.read_excel(test_data_path, index_col=False)
```

```
print(testdata)
```

```
test_array = testdata.values
test_Features = test_array[:,3:78:3]
test_RBC_Label = test_array[:,125]
test_RBC_Label[test_RBC_Label==2]=1
print(test_Features.shape)
print(test_RBC_Label)
print(type(test_Features))
```

```
std_test_Features=scaled.fit_transform(test_Features)
x_test=np.asarray(std_test_Features).astype(np.float32)
# one hot encoding
test_labels = to_categorical(test_RBC_Label, num_classes=2)
print(type(x_test))
print(x_test)
```

```
from keras import backend as K
#labelss = np.array([np.array(val) for val in labels])
test_labelss = K.cast_to_floatx(test_RBC_Label)
print(test_labelss)
```

```
scores = model.evaluate(x_test,test_labels)
print(test_labels)
print()
print('Score = ',scores[1])
```

```
prediction=model.predict(x_test)
prediction=np.argmax(prediction, axis=1)
```

```
print(prediction.shape)
print(test_RBC_Label.shape)
pd.crosstab(test_RBC_Label, prediction, colnames=['Predict'], rownames=['Labels'])
```

#6 - 2023-10-03 06:23 - 宏益 廖

- 檔案 clipboard-202310031423-gl0zk.png 已新增

#7 - 2023-10-03 06:24 - 宏益 廖

- 檔案 clipboard-202310031424-xnvic.png 已新增

#8 - 2023-10-03 06:24 - 宏益 廖

- 檔案 已刪除 (clipboard-202310031423-gl0zk.png)

#9 - 2023-10-03 06:26 - 宏益 廖

- 狀態 從 In Progress 變更為 Resolved

#10 - 2023-10-03 09:25 - Chifu Chung

- 完成百分比 從 0 變更為 100

#11 - 2023-12-14 18:45 - Chifu Chung

- 狀態 從 Resolved 變更為 Closed

#12 - 2024-09-04 15:02 - 凱鋒 王

- 關聯至 Task #231: AWPPG P type 連續裝置製作 已新增

檔案

Training Data_A1.xlsx	762 KB	2023-10-03	宏益 廖
Test Data_A1.xlsx	943 KB	2023-10-03	宏益 廖
clipboard-202310031413-u33lo.png	895 KB	2023-10-03	宏益 廖
clipboard-202310031414-ng4wq.png	26.9 KB	2023-10-03	宏益 廖
clipboard-202310031424-xnvic.png	211 KB	2023-10-03	宏益 廖