

餐廳風格預測AI

第二組 資科三 翁愉嫻 資科三 張曉雲 企管三 陳懿禎 企管三 陳品熏

摘要

透過建立預測餐廳風格的AI model，協助使用者看到餐廳名字時，就能判斷出其餐廳風格；開設餐廳者也得以透過此系統了解如何取名來凸顯其餐廳料理風格。

資料前置準備

餐廳名稱資料：

- (1) 刪除字串中特殊與用不到的字元
- (2) 使用CkipTagger斷詞
- (3) 將剩下的文字轉數據，本組利用以下兩種方法：

法一：Tokenization

```
from tensorflow.keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer(num_words=1000)
tokenizer.fit_on_texts(ws_results)
sequences = tokenizer.texts_to_sequences(ws_results)
```

法二：Word2vec

建立Word2Vector的model後，根據Word2Vector字典順序，給餐廳名稱進行編號

```
myWord2Vec =Word2Vec(ws_results, size=25, iter=10, sg=1, min_count=1)

for i in range(len(ws_results)):
    print([myWord2Vec.wv.vocab[sen].index for sen in ws_results[i]])

[136, 137, 59, 7, 138]
[139, 140]
```

把不足25維的vector後面補0,但0目前代表“苗栗”這個詞，會影響判斷，故把所有vocabulary號碼加1,讓“苗栗”的號碼從0變1,0號留給內容全都是0的vector

```
embedding_matrix[0]

array([[ 0.00231358, -0.00605555,  0.00644752,  0.00110873, -0.00133057,
         0.0051888 ,  0.0009851 ,  0.00875798, -0.01713349, -0.02000091,
         0.00245957,  0.01351579, -0.00906018,  0.00358147, -0.00013027,
         0.0012522 ,  0.02003205,  0.01087694, -0.01035121, -0.01232434,
         0.00855276, -0.00827995, -0.01242046, -0.00088804,  0.01958855],
      dtype=float32)

embedding_matrix=np.vstack((np.array(np.zeros(25)),embedding_matrix))

embedding_matrix[0]

array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0.])

embedding_matrix[1]

array([[ 0.00231358, -0.00605555,  0.00644752,  0.00110873, -0.00133057,
         0.0051888 ,  0.0009851 ,  0.00875798, -0.01713349, -0.02000091,
         0.00245957,  0.01351579, -0.00906018,  0.00358147, -0.00013027,
         0.0012522 ,  0.02003205,  0.01087694, -0.01035121, -0.01232434,
```

餐廳風格資料：one_hot_encoding處理

```
from sklearn.preprocessing import OneHotEncoder
onehotencoder = OneHotEncoder(categories="auto")
sequences = onehotencoder.fit_transform(restaurant_style).toarray()
```

資料處理

所有蒐集來的資料，20%作為測試資料，80%作為訓練資料。

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2, random_state=87)
```

Model

由於我們的model必須從多種風格中判斷出某一餐廳是何種風格，故以softmax來做為激活函數；損失函數為分類交叉熵函式，優化器為adam，以正確率作為衡量指標。

```
model.add(Dense(13, activation='softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

訓練

資料格式改為numpy.ndarray，方便輸入

```
x_test = np.asarray(x_test)
y_test = np.asarray(y_test)
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

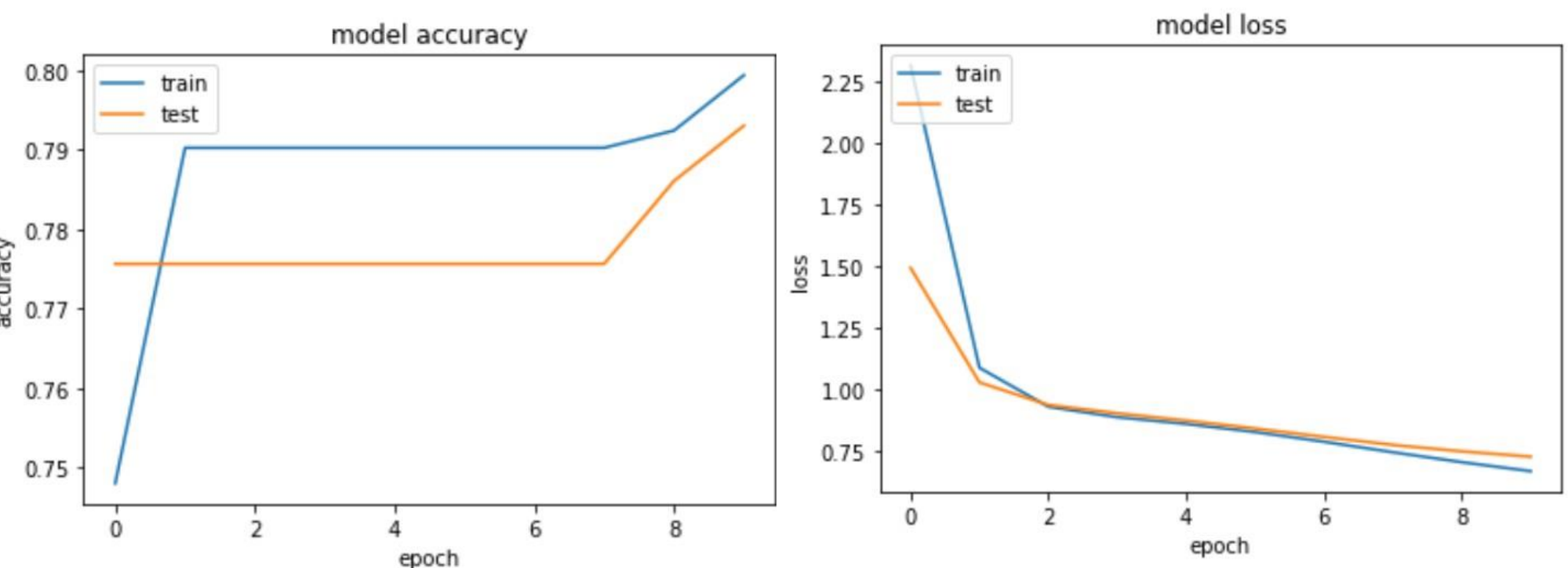
訓練次數改為十次，batch_size改成300，把測試資料也拿來fit

```
history = model.fit(x_train, y_train, batch_size=300, epochs=10, validation_data=(x_test, y_test))
```

結果

1.使用Tokenization方法

正確率為：0.8355 損失率為：0.5428



2.使用Word2vec方法

正確率為：0.7903 損失率為：0.9802

