手寫數字辨識應用程式

M11259003 吳柏呈

1. 測試平台及資訊

   平台：Colab

   Tensorflow version ：2.9.2

   資料集：mnist。共 60000 張 28*28 的灰階圖片 (60000,28,28)

   訓練集：50000 張 (50000,28,28)

   測試集：10000 張 (10000,28,28)

2. 模型架構

   從圖一看出共有 4 層 hidden layer，參數分別是 784、256、128 以及輸出層的 10，並採用 ReLU 作為 activation function。

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_3 (Flatten)         (None, 784)               0

 dense_12 (Dense)            (None, 784)               615440

 dropout_3 (Dropout)         (None, 784)               0

 dense_13 (Dense)            (None, 256)               200960

 dense_14 (Dense)            (None, 128)               32896

 dense_15 (Dense)            (None, 10)                1290

=================================================================
Total params: 850586 (3.24 MB)
Trainable params: 850586 (3.24 MB)
Non-trainable params: 0 (0.00 Byte)
```

圖 1

3. 超參數及 loss function

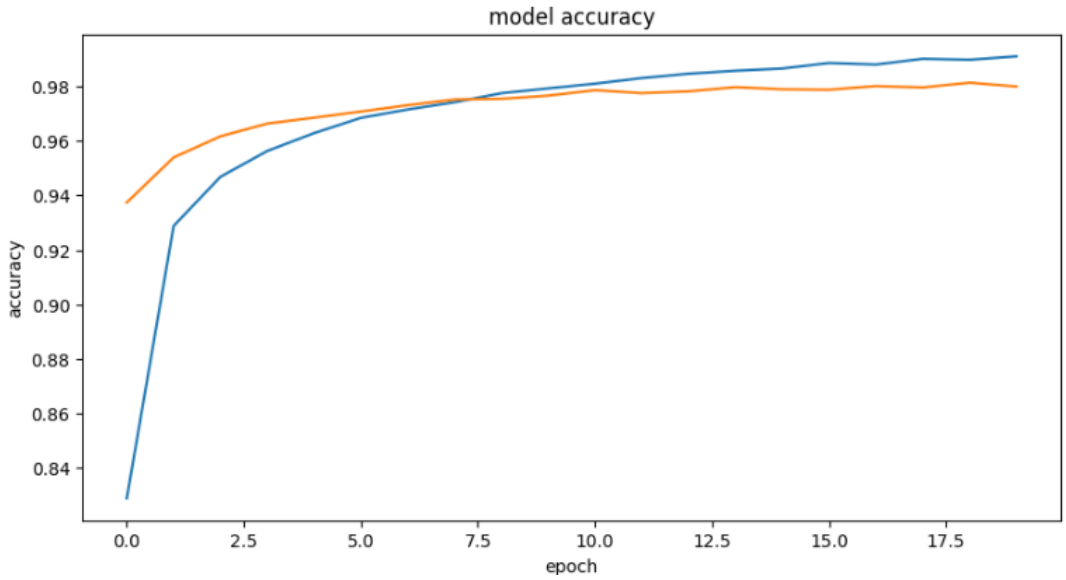
```
loss=tf.keras.losses.categorical_crossentropy
batch_size=64
epochs=20
lr=1e-4
optimizer=tf.keras.optimizers.Adam(learning_rate=lr)
```

圖 2

4. 訓練結果分析

圖 3 是訓練次數(X 軸)與準確率(Y 軸)的視覺化，其中藍線代表訓練的準確率變化，可以看到在訓練 10 次後，準確率已經 95%以上，

另外圖 4 代表訓練完後將測試集進行測試，可以得到約 98%的準確率。



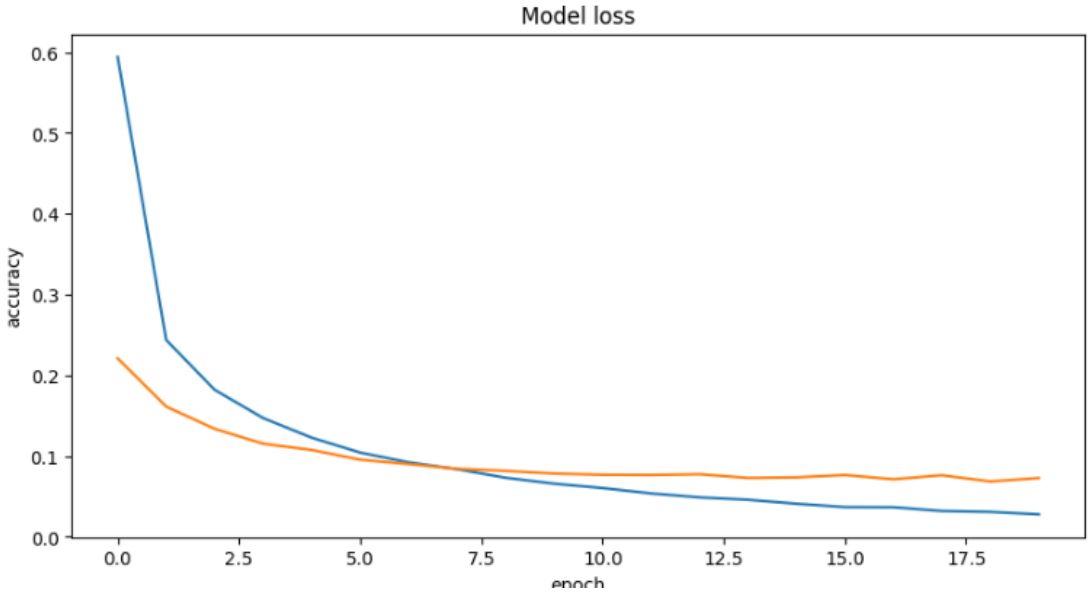
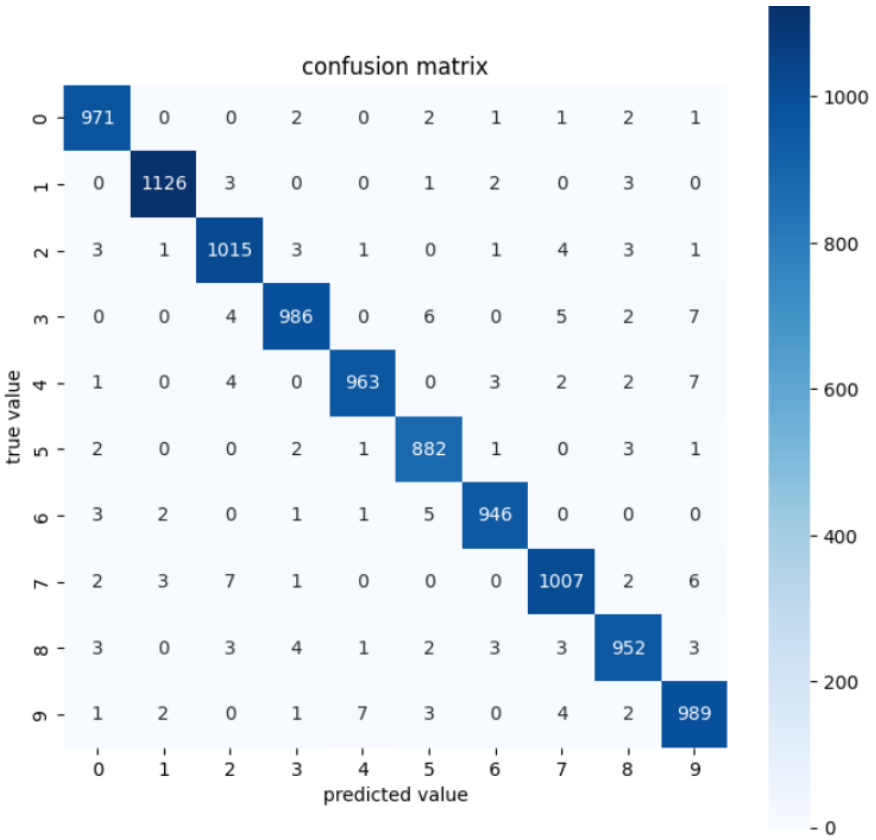


圖 3

```
313/313 [==============================] - 1s 4ms/step - loss: 0.0546 - accuracy: 0.9837
acc: 0.9836999773979187
```

圖 4

5. Confusion matrix



6. Source code

C: > Users > USER > Downloads > mnist.py

```python
# -*- coding: utf-8 -*-
"""mnist.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1nnAj6WfZn38_1q6opF_ohGBGgk_XqmC1
"""

import tensorflow as tf
import tensorflow.keras
import numpy as np
import matplotlib.pyplot as plt
print(tensorflow.__version__)

(x_train,y_train),(x_test,y_test)=tf.keras.datasets.mnist.load_data(
    path='mnist.npz'
)
x_train=x_train/255.
x_test=x_test/255.

y_train=tf.keras.utils.to_categorical(y_train, num_classes=10, dtype='float32')
y_test=tf.keras.utils.to_categorical(y_test, num_classes=10, dtype='float32')

print("training data shape:",x_train.shape)
print("testing data shape:",x_test.shape)

first_train_img = np.reshape(x_train[1, :], (28, 28))
plt.matshow(first_train_img, cmap = plt.get_cmap('gray'))
plt.show()

model=tf.keras.Sequential()
model.add(tf.keras.layers.InputLayer(input_shape=(28, 28, 1)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(784,activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(256,activation='relu'))
model.add(tf.keras.layers.Dense(128,activation='relu'))
model.add(tf.keras.layers.Dense(10,activation='softmax'))

model.summary() #確認每一層內容 繼承的方法不能用.summary

loss=tf.keras.losses.categorical_crossentropy
batch_size=64
epochs=20
lr=1e-4
optimizer=tf.keras.optimizers.Adam(learning_rate=lr)

model.compile(loss=loss, optimizer=optimizer, metrics='accuracy')

history=model.fit(x_train,y_train,batch_size=batch_size,epochs=epochs,shuffle='true')

history.history
#history.history[loss]

his=model.evaluate(x_test,y_test)
loss=his[0]
accuracy=his[1]
print("acc:",accuracy)

import matplotlib.pyplot as plt


plt.figure()
plt.plot(history.history['accuracy'])
plt.plot(history.history['loss'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')

import sklearn
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
import seaborn as sns

y_predict = model.predict(x_test)
y_predict = np.argmax(y_predict, 1)

rounded_ytest=np.argmax(y_test, axis=1)       #轉回整數矩陣 原本one hot無法用在confusion matrix
mat = confusion_matrix(rounded_ytest, y_predict)

plt.figure(figsize=(8,8))
sns.heatmap(mat, square= True, annot=True, cbar= True, fmt='.20g', cmap='Blues')

plt.title('confusion matrix')
plt.xlabel("predicted value")
plt.ylabel("true value")
plt.show()
```