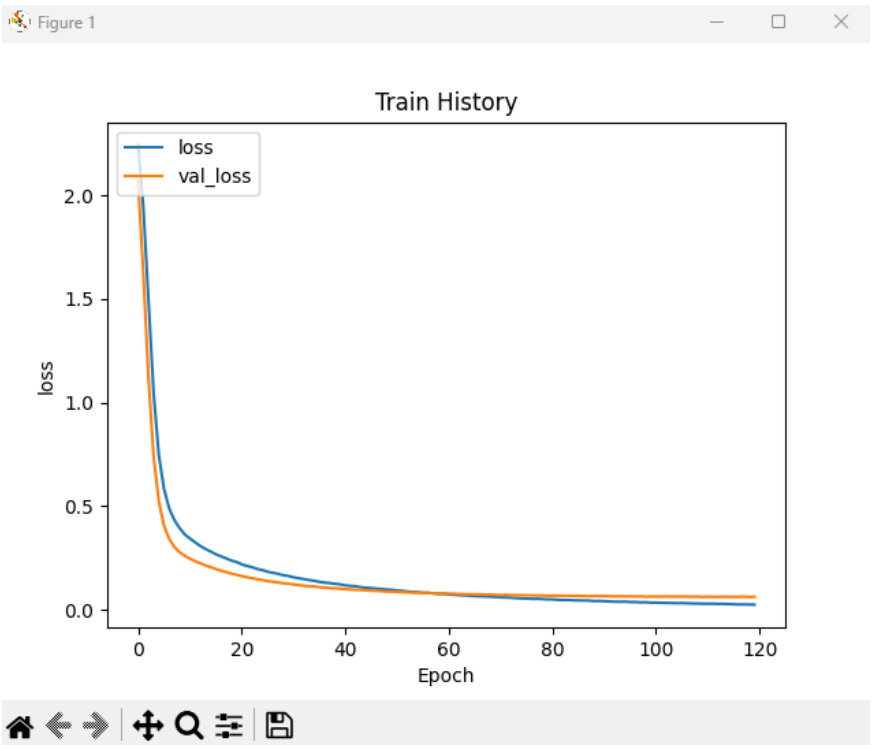
作業三 實作 keras MNIST 手寫數字辨識問題

程式碼:

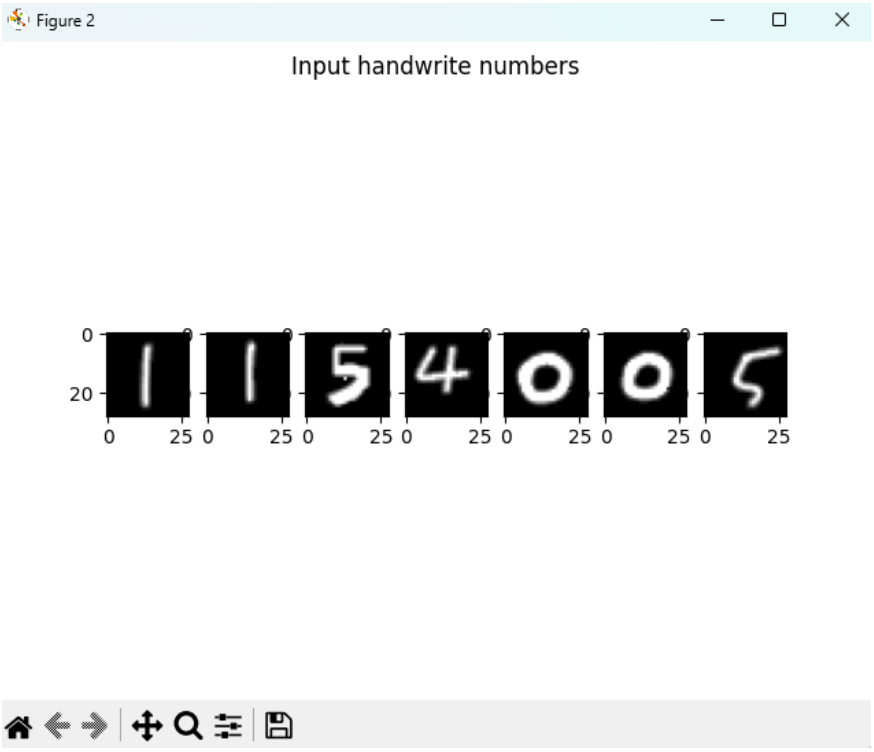
```
from re import split
import keras.utils
import numpy as np
import keras
from keras.models import Sequential
from keras.datasets import mnist
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.utils import np_utils
from keras.preprocessing import image
from matplotlib import pyplot as plt
from PIL import Image
import logging
import os

LOGGING_FORMAT ='%(asctime)s [%(levelname)s]:%(message)s'
logging.basicConfig(level=logging.INFO, format=LOGGING_FORMAT, filename='Log.log', filemode='a')

#獲取資料集
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()

#建立並增加模型層
model = Sequential()
model.add(Dense(units=256, input_dim=784, kernel_initializer='normal', activation='relu'))
model.add(Dropout(rate = 0.2))
model.add(Dense(units=64, kernel_initializer='normal', activation='relu'))
model.add(Dense(units=10, kernel_initializer='normal', activation='softmax'))

# 編譯: 選擇損失函數、優化方法及成效衡量方式
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

#資料處理、正規化
Y_trainOneHot = keras.utils.to_categorical(Y_train)
Y_testOneHot = keras.utils.to_categorical(Y_test)

X_train_2D = X_train.reshape(X_train.shape[0], 28*28).astype('float32')
X_test_2D = X_test.reshape(X_test.shape[0], 28*28).astype('float32')

X_train_norm = X_train_2D/255
X_test_norm = X_test_2D/255

#train
train_history = model.fit(x=X_train_norm, y=Y_trainOneHot, validation_split=0.1,epochs=120, batch_size=10000, verbose=2)

scores = model.evaluate(X_test_norm, Y_testOneHot)
```

```
45.
46. print("\t[Info] Accuracy of testing data = {:2.1f}%".format(scores[1]*100.0))
47. #logging.info("Accuracy of testing data = {:2.1f}%".format(scores[1]*100.0))
48.
49. #繪製 loss 分析圖
50. plt.figure(1)
51. plt.plot(train_history.history['loss'])
52. plt.plot(train_history.history['val_loss'])
53. plt.title('Train History')
54. plt.ylabel('loss')
55. plt.xlabel('Epoch')
56. plt.legend(['loss', 'val_loss'], loc='upper left')
57.
58. #預測學號
59. img = keras.utils.load_img('1154005.bmp', color_mode='grayscale')
60. img_number = []
61.
62. #處理讀取進來的學號圖片
63. for i in range(0, 7):
64. x = i*28
65. out = os.path.join(os.getcwd(), f'{i*28}_{x}.png')
66. splitted_img = img.crop((x, 0, x+28, 28))
67. img_number.append(np.array(splitted_img).flatten()/255)
68. splitted_img.save(f'{i*28}_{x+28}.png')
69.
70. #show 手寫圖片
71. plt.figure(2)
72. for i in range(len(img_number)):
73. img = np.reshape(img_number[i], (28, 28))
74. plt.subplot(1, 7, i+1)
75. plt.imshow(img, cmap='gray')
76. plt.suptitle("Input handwrite numbers")
77. plt.show()
78.
79. #預測手寫學號數字
80. img_number = np.array(img_number)
81. predictions = model.predict(img_number)
82. ans = np.argmax(predictions, axis=1)
83.
84. #輸出預測結果
85. id = str.join("", map(str,ans))
86. print(f"你的手寫學號經過預測是:S{id}")
```

Train History 圖表化



將輸入預測的手寫數字 show 出來



最終顯示預測結果

```
6/6 - 0s - loss: 0.0264 - accuracy: 0.9926 - val_loss: 0.0629 - val_accuracy: 0.9835 - 112ms/epoch - 19ms/step
Epoch 118/120
6/6 - 0s - loss: 0.0263 - accuracy: 0.9928 - val_loss: 0.0638 - val_accuracy: 0.9837 - 96ms/epoch - 16ms/step
Epoch 119/120
6/6 - 0s - loss: 0.0259 - accuracy: 0.9927 - val_loss: 0.0622 - val_accuracy: 0.9830 - 88ms/epoch - 15ms/step
Epoch 120/120
6/6 - 0s - loss: 0.0253 - accuracy: 0.9930 - val_loss: 0.0635 - val_accuracy: 0.9832 - 86ms/epoch - 14ms/step
313/313 [==============================] - 0s 1ms/step - loss: 0.0668 - accuracy: 0.9800
       [Info] Accuracy of testing data = 98.0%
1/1 [==============================] - 0s 52ms/step
你的手寫學號經過預測是:S1154005
```