

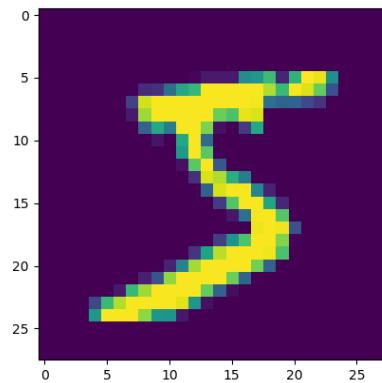
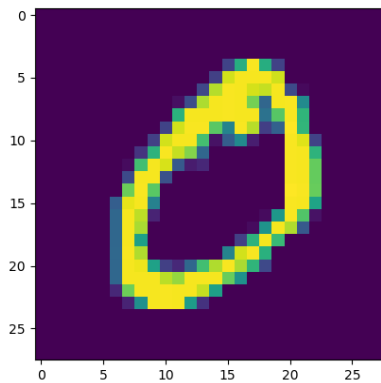
AI 無線通訊系統實驗

Lab 1 Classification DNN

Author: 蕭安紘 助教

實驗目的

建立一個手寫辨識之神經網路，藉此判斷手寫的數字是多少



實驗介紹

How to Build a NN Model

一個完整的網路包含輸入(Input Layer)、輸出(Output Layer)、隱藏層(Hidden Layer):

Input Layer:

```
inputs = keras.Input(shape = 784)
```

Output Layer:

```
outputs = tf.keras.layers.Dense(10, activation=tf.nn.softmax)(d4)
```

Hidden Layer(Dense):

```
Object2 = tf.keras.layers.Dense(units, activation=None)(Object1)
```

Object2: 物件名稱

units: 輸出維度

activation: 挑選想使用的 activation function

Object1: 輸入之物件

範例：

```
d1 = tf.keras.layers.Dense(512, activation=tf.nn.leaky_relu)(inputs)
```

Compile Model

將以上你所建立的 hidden layre 串聯起來，並命名為 Model

```
model = tf.keras.models.Model(inputs=inputs, outputs=outputs)
```

設定model需用到的Loss Function與optimizer

```
model.compile(loss= None, optimizer = None, metrics=['accuracy'])
```

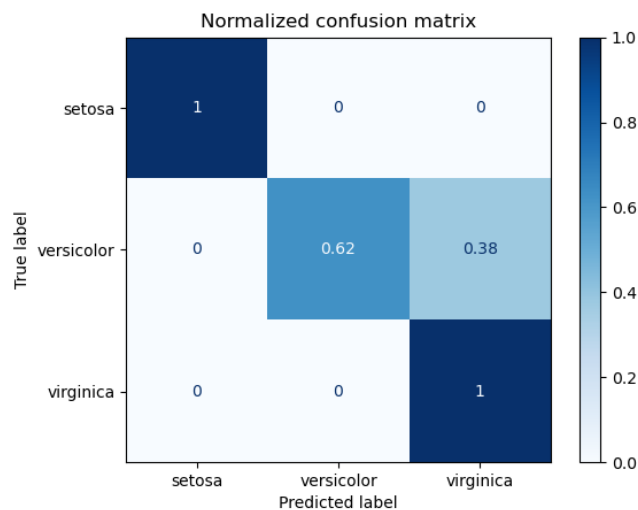
loss: 選擇使用的 loss function，須為字串，如"MSE"

optimizer: 選擇使用的 optimizer，須為字串，如"Adam"

metrics: 在訓練與測試時的模型評估標準

model.summary(): 顯示目前所建立之神經網路的層數與參數

confusion matrix: 常用來觀察分類問題之結果

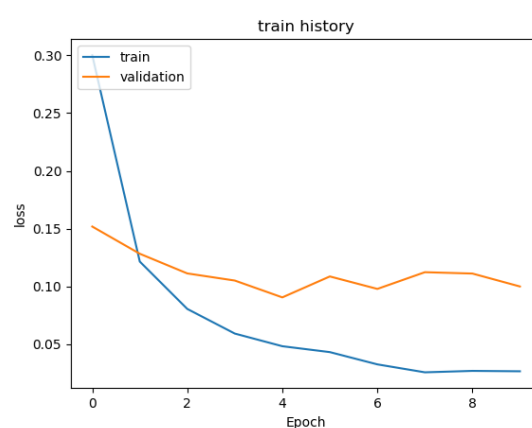
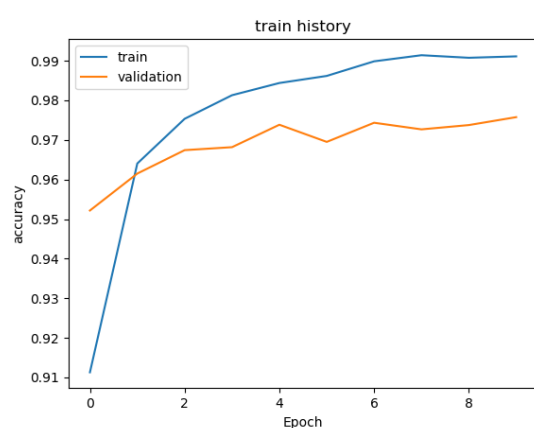


實驗步驟

1. 讀取資料
2. 資料前處理(Reshape, Normalize)
3. 建立包含 Input Layer、4 層 Hidden Layer(Node: 512, 256, 128, 64)、Output(Node: 10) Layer 之神經網路，loss 選擇"category_crossentropy"，optimizer 選擇"Adam"，並利用 `model.summary()` 在 console 裡印出下圖

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 784)]	0
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 10)	650
Total params: 575,050		
Trainable params: 575,050		
Non-trainable params: 0		

4. 進行神經網路模型的訓練 `model.fit()`
5. 畫出訓練過程中的精準度(Accuracy)與損失函數(Loss Function)



6. 印出最後一個 epoch 所產生之模型的測試精準度(Test Accuracy)

```
10000/10000 [=====] - 1s 55us/sample - loss: 0.0873 - accuracy: 0.9777  
test loss, test accuracy= [0.08734572482189906, 0.9777]
```

基礎題

完成以上實驗步驟並找助教 Demo

加分題(先完成基礎題才可以 demo)

1. 畫出 train data 的圖，包含 0~10 的數字各一張

Hint:

可使用 `plt.imshow()` 幫助作圖

2. 畫出判斷的 confusion matrix

Hint:

可用 sklearn 裡的套件 `confusion_matrix()` 幫忙計算

可用畫的或是 `print` 在 console 裡