TensorFlow

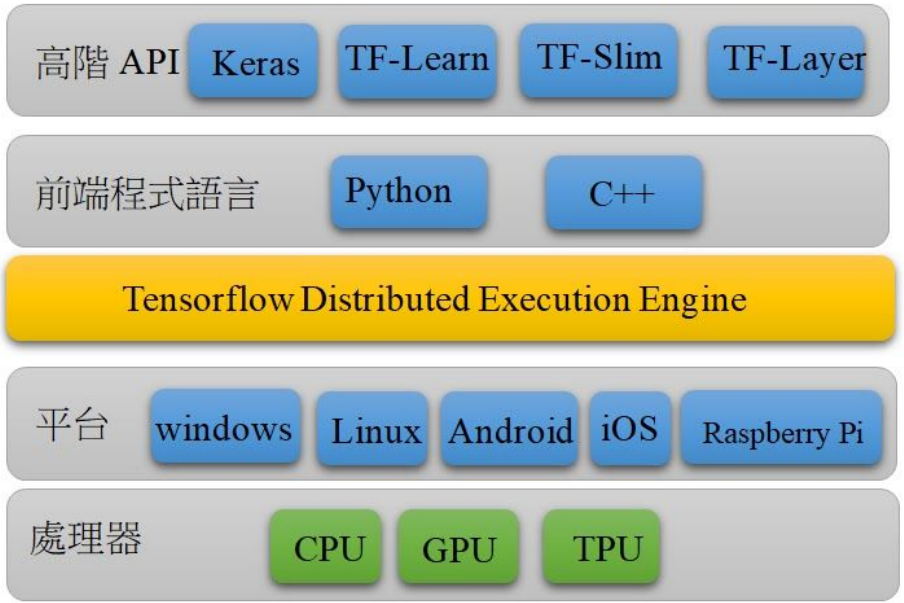
講師:陳聖文

01

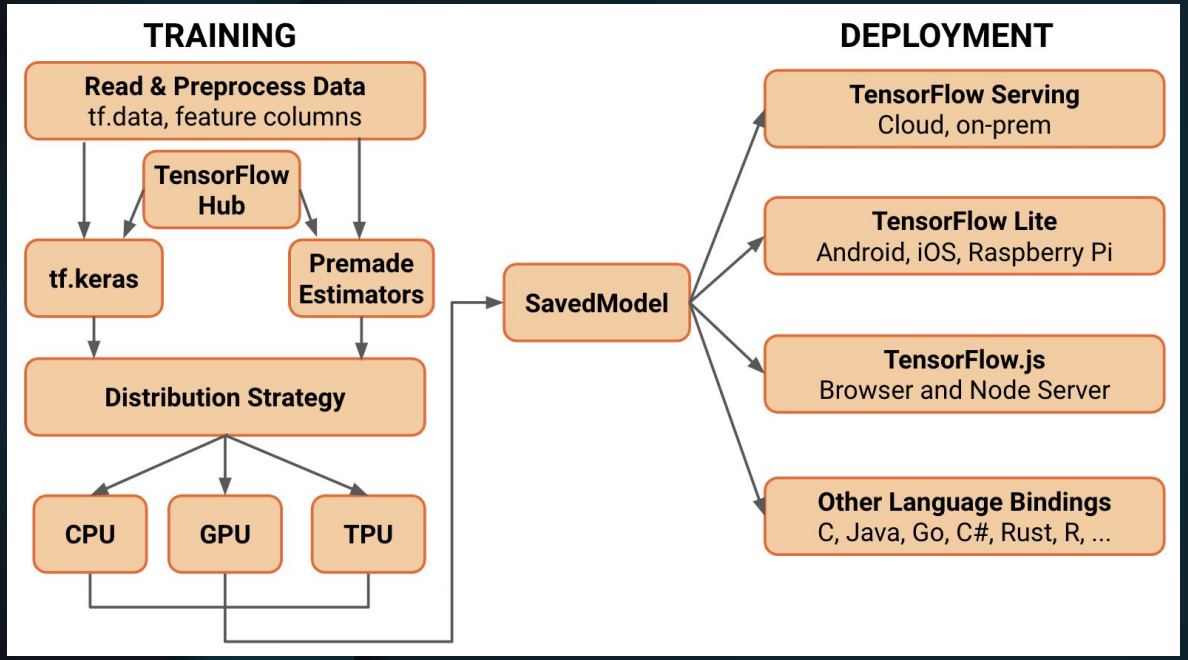
TensorFlow 介紹

什麼是 TensorFlow ?

Tensorflow 架構圖說明:

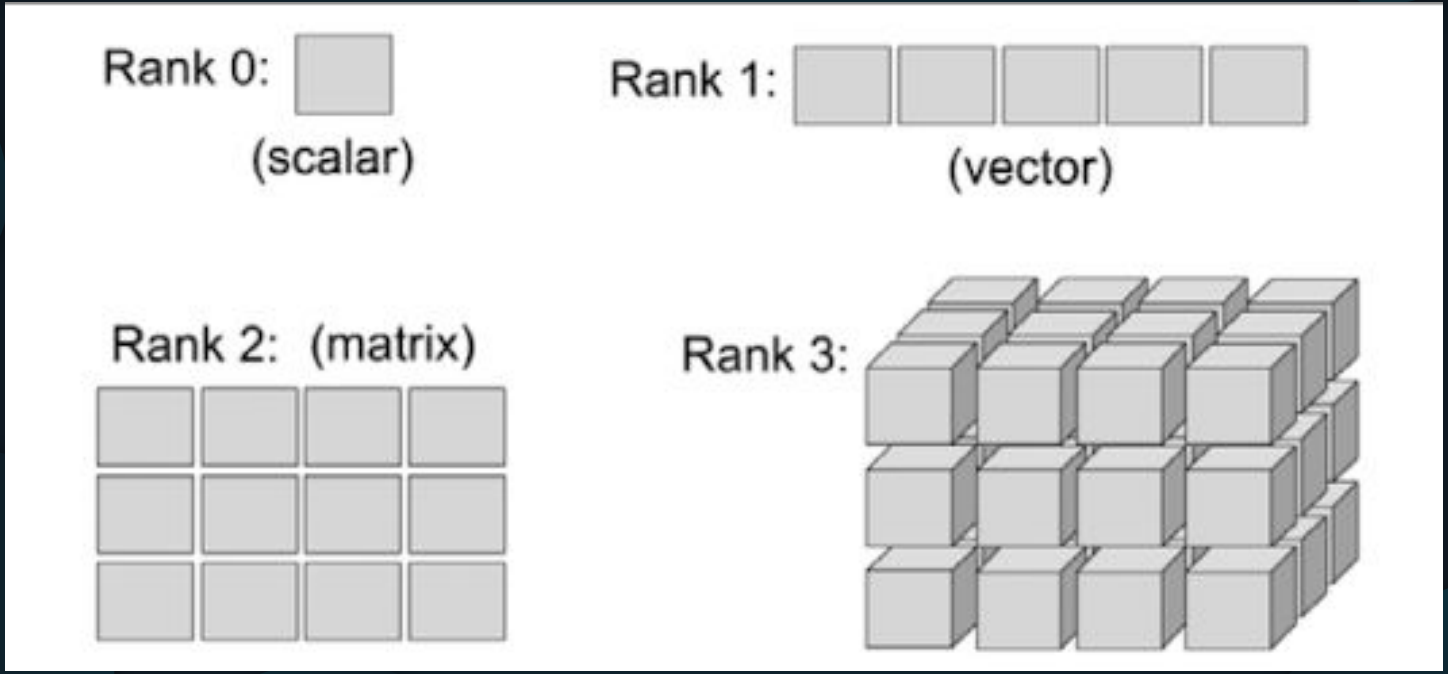


TensorFlow 工作流程



TensorFlow 工作流程

什麼是 Tensor



TensorFlow 基本用法

04. TensorFlow 框架介紹

目錄

基礎用法

匯入套件庫

[1]:
```
# 安裝 TensorFlow 套件
```

[19]:
```
# 匯入 TensorFlow 套件
```

```
TensorFlow 版本: 2.13.0
```

基本用法

創建張量 (Tensor)

創建 rank 0 張量 (純量, Scalar)

[20]:
```

```

```
tf.Tensor(4, shape=(), dtype=int32)
```

創建 rank 1 張量 (向量, Vector)

[21]:
```

```

```
tf.Tensor([2. 3. 4.], shape=(3,), dtype=float32)
```

創建 rank 2 張量 (矩陣, Matrix)

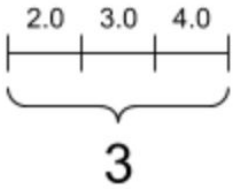
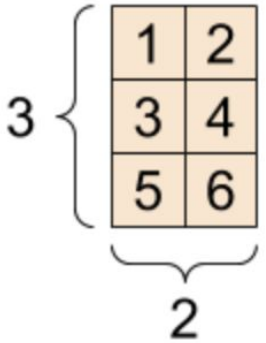
[22]:
```

```

```
tf.Tensor(
[[1. 2.]
 [3. 4.]
 [5. 6.]], shape=(3, 2), dtype=float16)
```

創建更高維度的張量

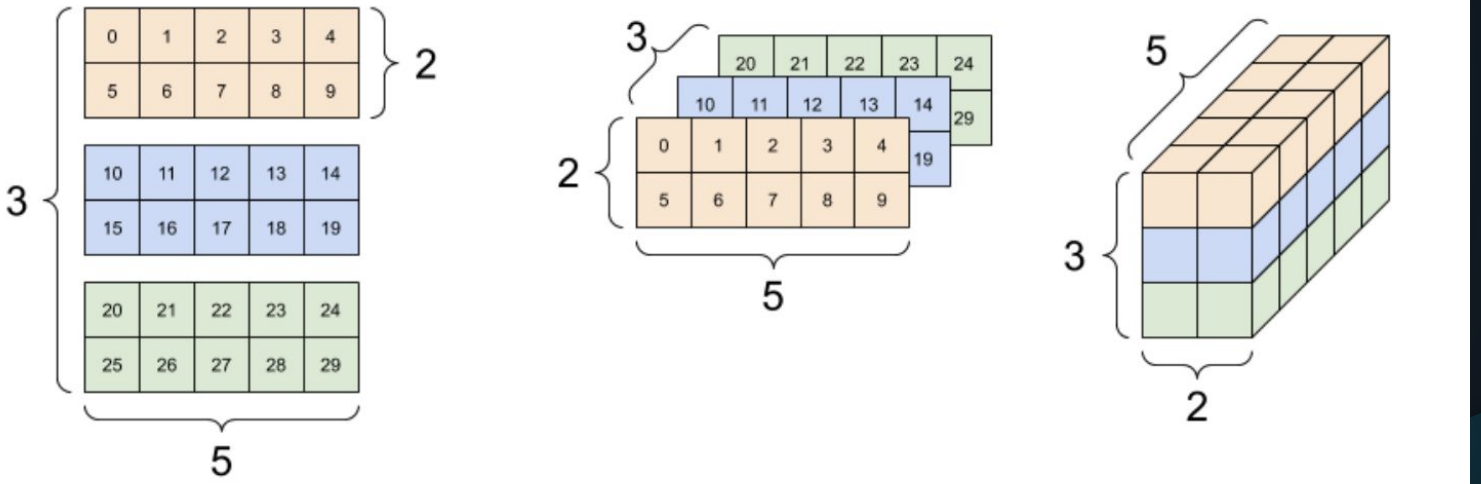
創建 float32 例型的張量

什麼是 Tensor

A scalar, shape: [ ]	A vector, shape: [3]	A matrix, shape: [3, 2]
4		

什麼是 Tensor

A 3-axis tensor, shape: [3, 2, 5]



Tensor 的運算元

TensorFlow 運算方法

定義張量 a 與 b

[30]:

```
tf.Tensor(
[[1 2]
 [3 4]], shape=(2, 2), dtype=int32)
```

[29]:

```
tf.Tensor(
[[1 1]
 [1 1]], shape=(2, 2), dtype=int32)
```

加法運算

[31]:

[31]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[2, 3],
       [4, 5]], dtype=int32)>
```

[34]:

[34]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[2, 3],
       [4, 5]], dtype=int32)>
```

元素乘法

[32]:

[32]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[1, 2],
       [3, 4]], dtype=int32)>
```

[35]:

[35]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[1, 2],
       [3, 4]], dtype=int32)>
```

矩陣乘法

[33]:

[33]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[3, 3],
       [7, 7]], dtype=int32)>
```

[36]:

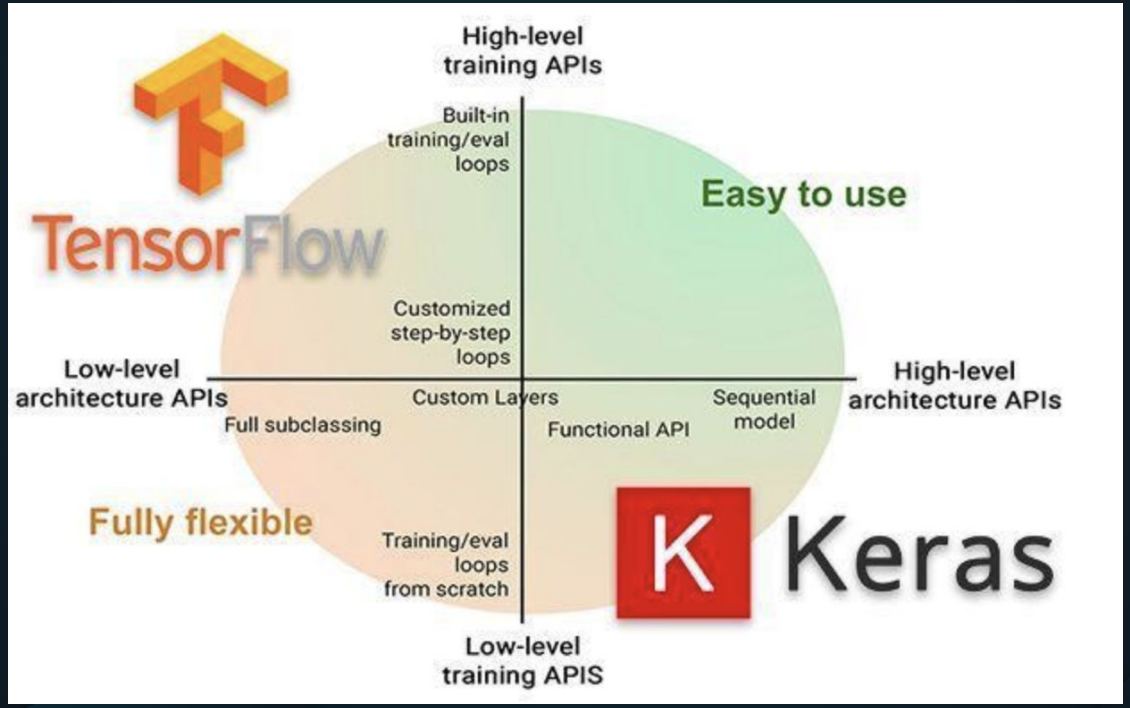
[36]:
```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[3, 3],
       [7, 7]], dtype=int32)>
```

聚合運算

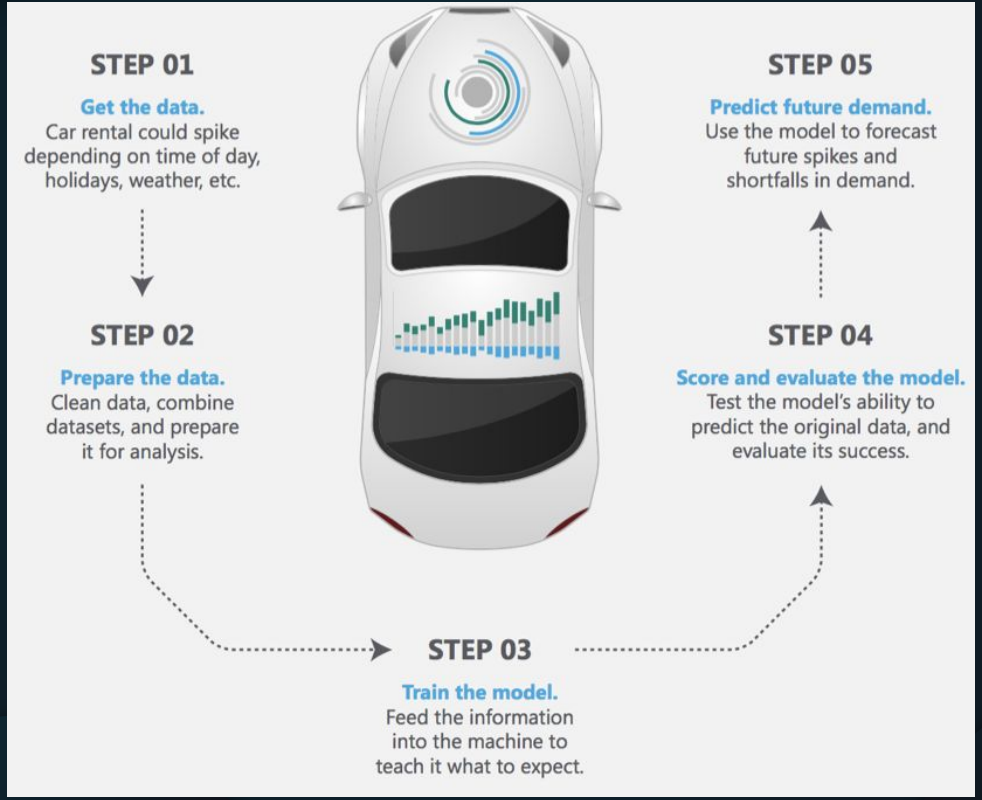
01

Keras 介紹

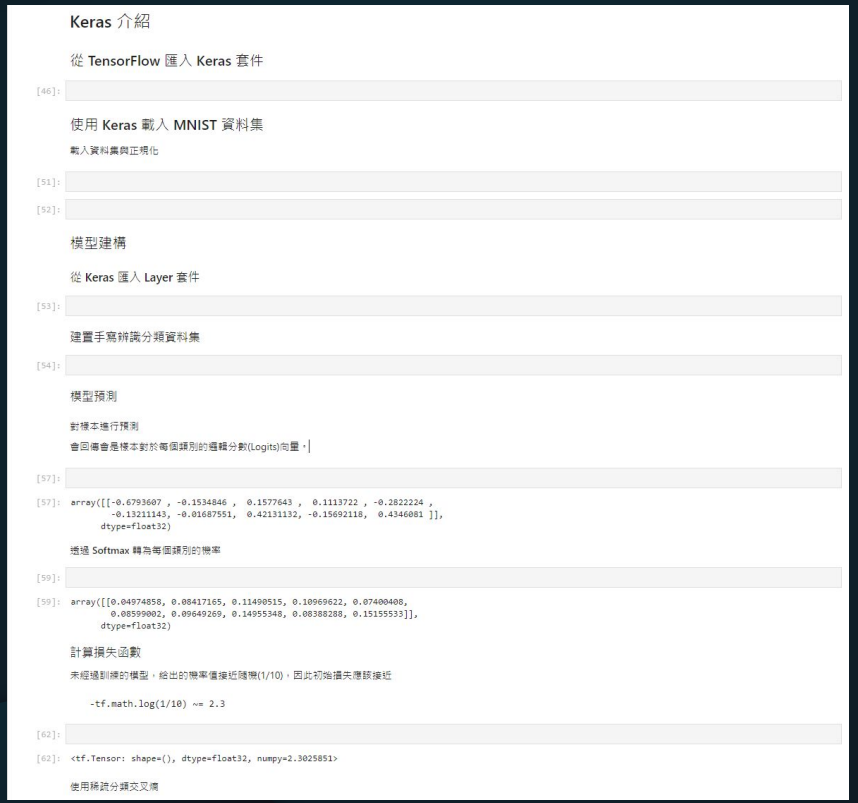
Keras 介紹


High-level
training APIs
Built-in
training/eval
loops
Easy to use
TensorFlow
Customized
step-by-step
loops
Low-level
architecture APIs
High-level
architecture APIs
Custom Layers
Full subclassing
Functional API
Sequential
model
Fully flexible
Training/eval
loops
from scratch
K
Keras
Low-level
training APIS

機器學習基礎流程


STEP 01
Get the data.
Car rental could spike depending on time of day, holidays, weather, etc.
STEP 02
Prepare the data.
Clean data, combine datasets, and prepare it for analysis.
STEP 03
Train the model.
Feed the information into the machine to teach it what to expect.
STEP 04
Score and evaluate the model.
Test the model's ability to predict the original data, and evaluate its success.
STEP 05
Predict future demand.
Use the model to forecast future spikes and shortfalls in demand.

Keras 介紹



Keras 介紹

從 TensorFlow 匯入 Keras 套件

[46]:

使用 Keras 載入 MNIST 資料集

載入資料集與正規化

[51]:

[52]:

模型建構

從 Keras 匯入 Layer 套件

[53]:

建置手寫辨識分類資料集

[54]:

模型預測

對樣本進行預測

會回傳會是樣本對於每個類別的邏輯分數(Logits)向量，

[57]:

[57]:
```
array([[-0.6793607 , -0.1534846 ,  0.1577643 ,  0.1113722 , -0.2822224 ,
        -0.13211143, -0.01687551,  0.42131132, -0.15692118,  0.4346081 ]],
      dtype=float32)
```

通過 Softmax 轉為每個類別的機率

[59]:

[59]:
```
array([[0.04974858, 0.08417165, 0.11490515, 0.10969622, 0.07400408,
        0.08599002, 0.09649269, 0.14955348, 0.08388288, 0.15155533]],
      dtype=float32)
```

計算損失函數

未經過訓練的模型，給出的機率值接近隨機(1/10)，因此初始損失應該接近

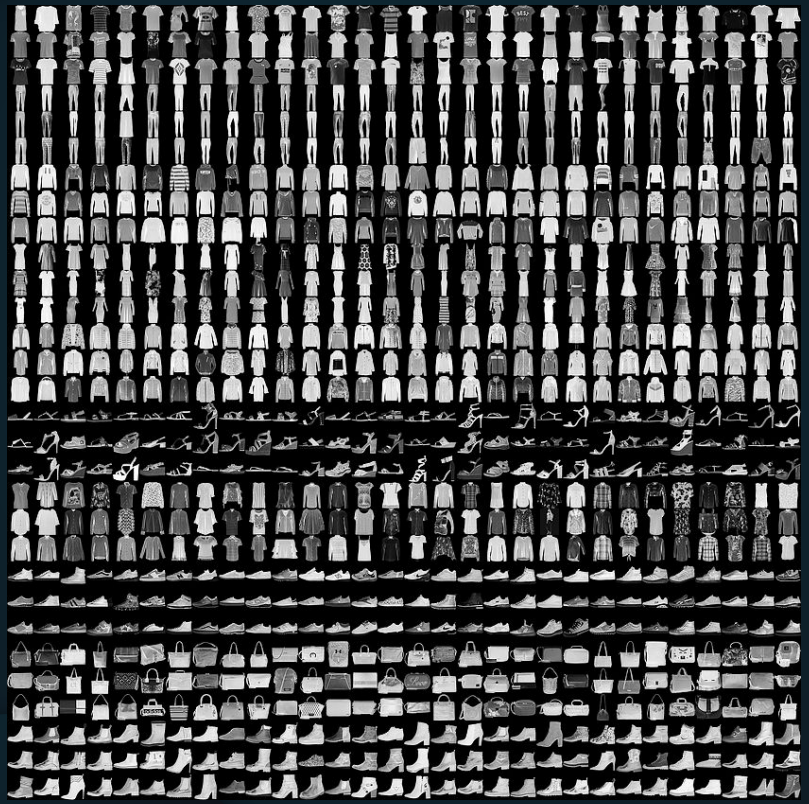
$$-tf.math.log(1/10) \approx 2.3$$

[62]:

[62]:
```
<tf.Tensor: shape=(), dtype=float32, numpy=2.3025851>
```

使用稀疏分類交叉熵

Fashion MNIST 資料集



Fashion MNIST 標籤

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

透過 Keras 實做服裝分類

透過 Keras 實做服裝分類

目錄

前期準備

匯入套件

```
[ ]  1  # 匯入 Tensorflow 與 tf.keras
     2
     3
     4  # 匯入 Numpy 與 Matplotlib
```

匯入 Fashion MNIST 資料集

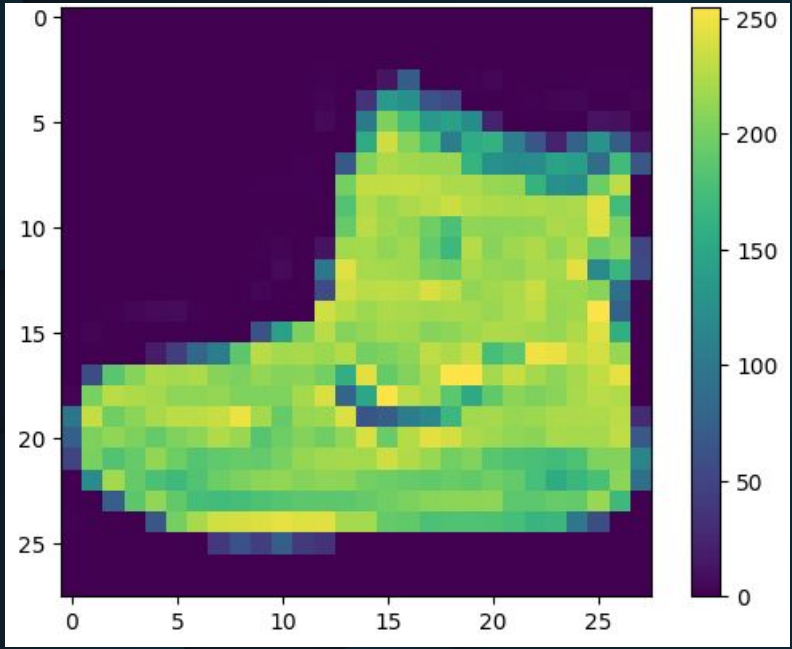
```
[ ]  1  # 從 Keras Datasets 中匯入服裝資料集
     2
     3  # 從資料集中提取資料，分為訓練集與驗證集
```

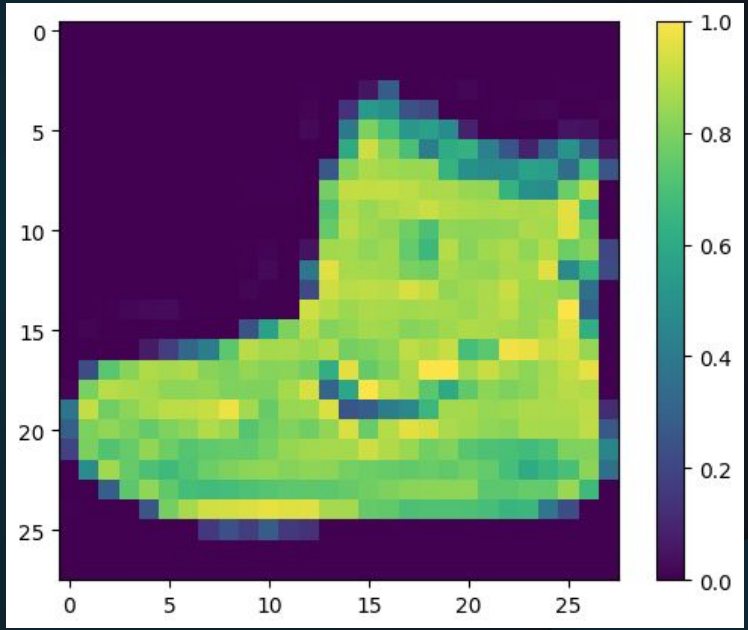
定義服裝類別

```
[ ]  1  # 定義服裝類別
     2  class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
     3                 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
     4
     5  # 縣市服裝類別表
```

	class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt

資料正規化


0
5
10
15
20
25
0
5
10
15
20
25
250
200
150
100
50
0


0
5
10
15
20
25
0
5
10
15
20
25
1.0
0.8
0.6
0.4
0.2
0.0

印出資料集中的一些影像

架構模型

使用 Keras 建構分類模型

設置模型的每一層

```
1 # 使用 Keras 建置序列模型
```

模型總覽

```
[ ] 1
```

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

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 101770 (397.54 KB)
Trainable params: 101770 (397.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

編譯模型

```
[ ] 1 # 對模型進行編譯
```

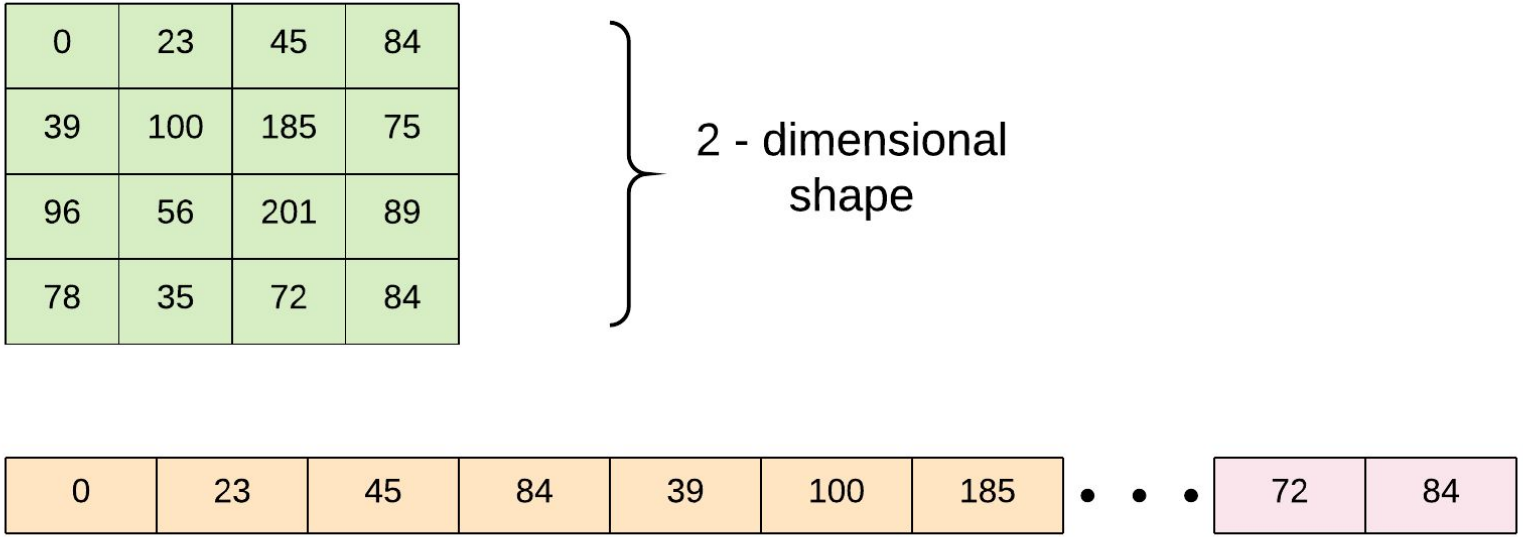
訓練模型

模型擬合

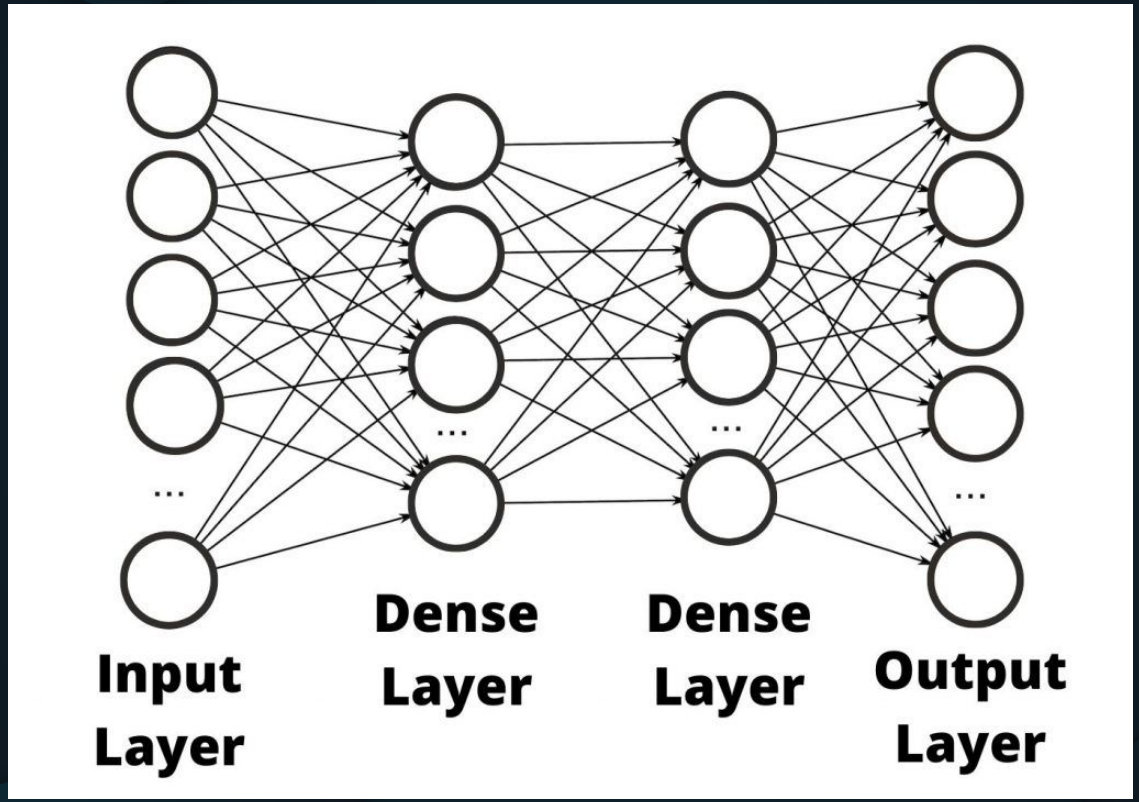
```
[ ] 1 # 對模型進行訓練
```

```
Epoch 1/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.5050 - accuracy: 0.8229
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3811 - accuracy: 0.8615
Epoch 3/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3432 - accuracy: 0.8740
Epoch 4/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3169 - accuracy: 0.8843
Epoch 5/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2983 - accuracy: 0.8901
Epoch 6/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2851 - accuracy: 0.8936
Epoch 7/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2730 - accuracy: 0.8991
Epoch 8/10
```

扁平層 (Flatten)


0 23 45 84
39 100 185 75
96 56 201 89
78 35 72 84
2 - dimensional shape
0 23 45 84 39 100 185 • • • 72 84
Flattened 1 - dimensional shape

全連結層 (Dense)



模型架構

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

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 101770 (397.54 KB)
Trainable params: 101770 (397.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

架構模型

使用 Keras 建構分類模型

設置模型的每一層

```
1 # 使用 Keras 建置序列模型
```

模型總覽

```
[ ] 1
```

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

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 101770 (397.54 KB)
Trainable params: 101770 (397.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

編譯模型

```
[ ] 1 # 對模型進行編譯
```

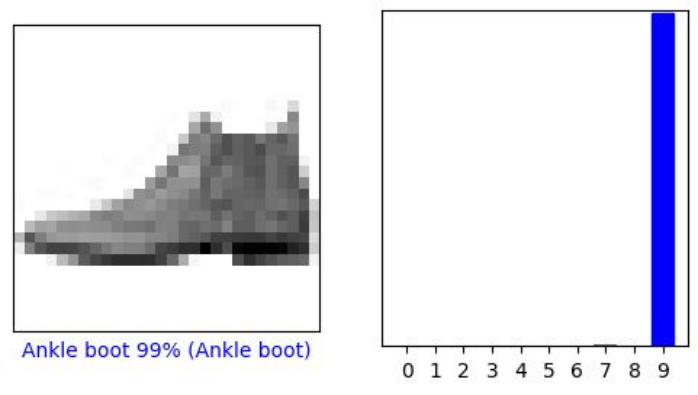
訓練模型

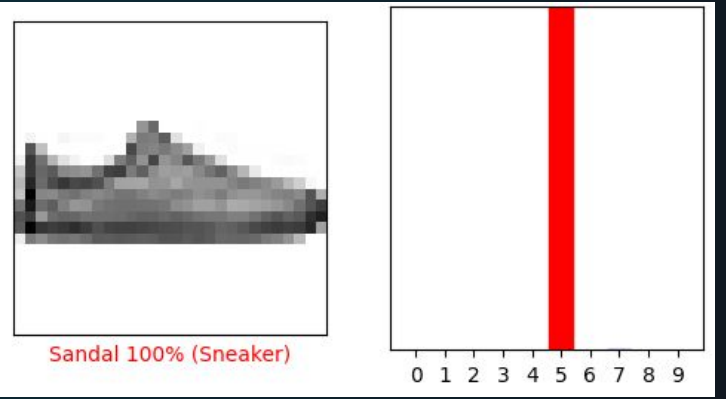
模型擬合

```
[ ] 1 # 對模型進行訓練
```

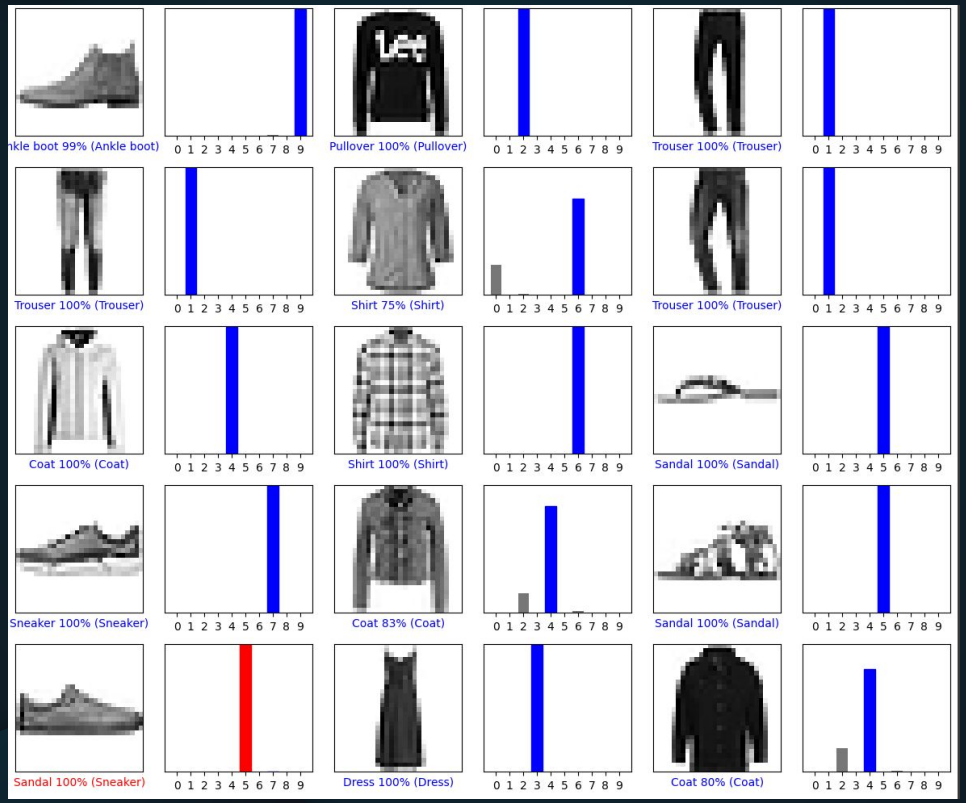
```
Epoch 1/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.5050 - accuracy: 0.8229
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3811 - accuracy: 0.8615
Epoch 3/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3432 - accuracy: 0.8740
Epoch 4/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3169 - accuracy: 0.8843
Epoch 5/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2983 - accuracy: 0.8901
Epoch 6/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2851 - accuracy: 0.8936
Epoch 7/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2730 - accuracy: 0.8991
Epoch 8/10
```

預測正確與預測錯誤

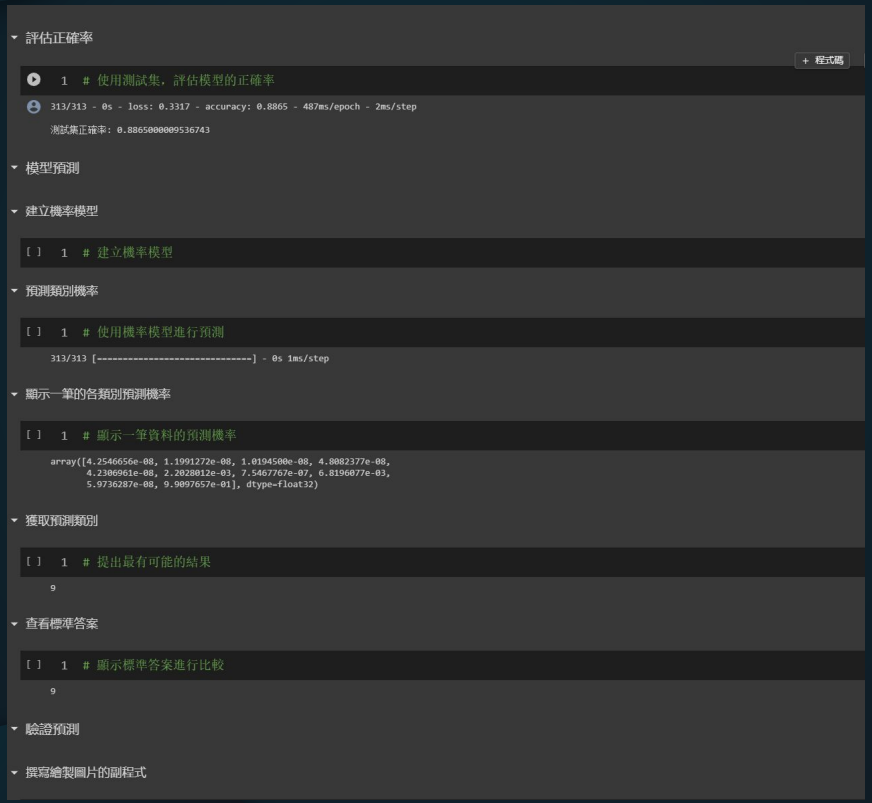

Ankle boot 99% (Ankle boot)
0 1 2 3 4 5 6 7 8 9


Sandal 100% (Sneaker)
0 1 2 3 4 5 6 7 8 9

顯示多筆預測結果


nkle boot 99% (Ankle boot)
Pullover 100% (Pullover)
Trouser 100% (Trouser)
Trouser 100% (Trouser)
Shirt 75% (Shirt)
Trouser 100% (Trouser)
Coat 100% (Coat)
Shirt 100% (Shirt)
Sandal 100% (Sandal)
Sneaker 100% (Sneaker)
Coat 83% (Coat)
Sandal 100% (Sandal)
Sandal 100% (Sneaker)
Dress 100% (Dress)
Coat 80% (Coat)
0 1 2 3 4 5 6 7 8 9

模型預測與評估


評估正確率
+ 程式碼
1 # 使用測試集，評估模型的正確率
313/313 - 0s - loss: 0.3317 - accuracy: 0.8865 - 487ms/epoch - 2ms/step
測試集正確率: 0.8865000009536743
模型預測
建立機率模型
[ ] 1 # 建立機率模型
預測類別機率
[ ] 1 # 使用機率模型進行預測
313/313 [==============================] - 0s 1ms/step
顯示一筆的各類別預測機率
[ ] 1 # 顯示一筆資料的預測機率
array([4.2546656e-08, 1.1991272e-08, 1.0194500e-08, 4.8082377e-08,
4.2306961e-08, 2.2028012e-03, 7.5467767e-07, 6.8196077e-03,
5.9736287e-08, 9.9097657e-01], dtype=float32)
獲取預測類別
[ ] 1 # 提出最有可能的結果
9
查看標準答案
[ ] 1 # 顯示標準答案進行比較
9
驗證預測
撰寫繪製圖片的副程式