Definition of Tensor Flow + Keras

深度學習人工智慧實務應用

Speaker: 蔡煒俊

Outline

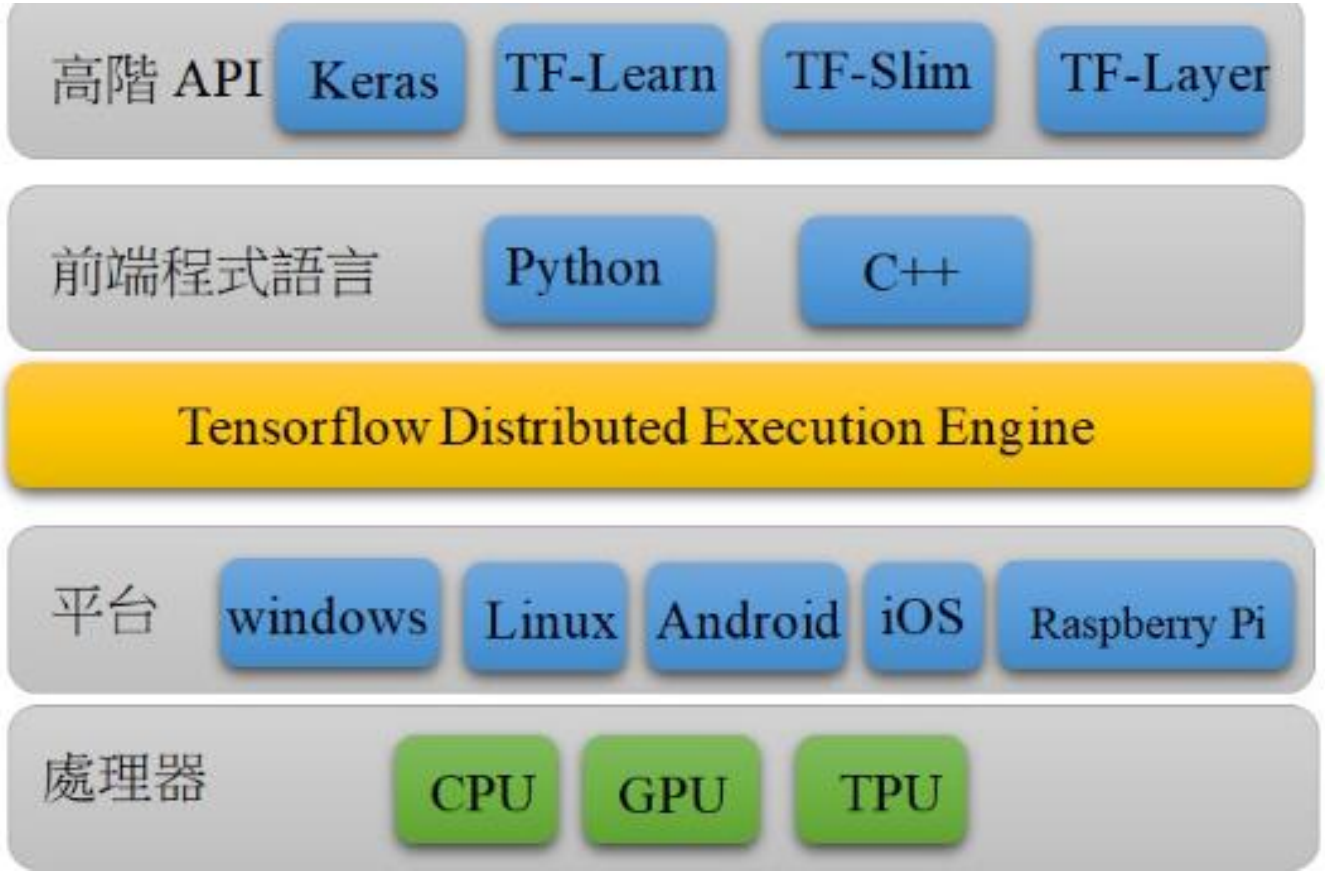
- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

History of TensorFlow

TensorFlow is an open source library provided by Google. Many of Google's products have long used TensorFlow technology to develop deep learning and machine learning aptitudes.
For example Gmail filtering spam, Google voice recognition, Google image recognition, Google translation, etc.

Outline

- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**



Structure of TensorFlow

Outline

- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

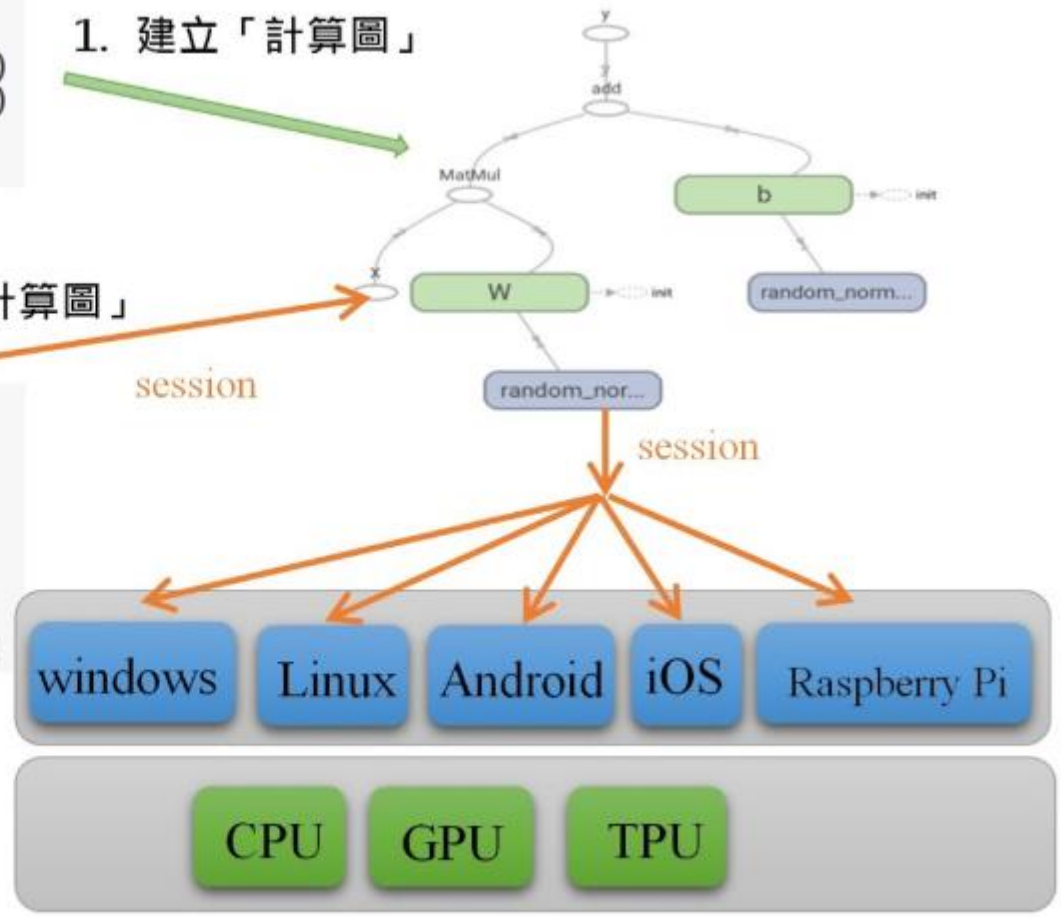
Data flow of *FLOW*

```python
import tensorflow as tf
import numpy as np
W = tf.Variable(tf.random_normal([3, 2]),name='W')
b = tf.Variable(tf.random_normal([1, 2]),name='b')
X = tf.placeholder("float", [None,3],name='X')
y=tf.nn.sigmoid(tf.matmul(X,W)+b,'y')
```

1. 建立「計算圖」

2.執行「計算圖」

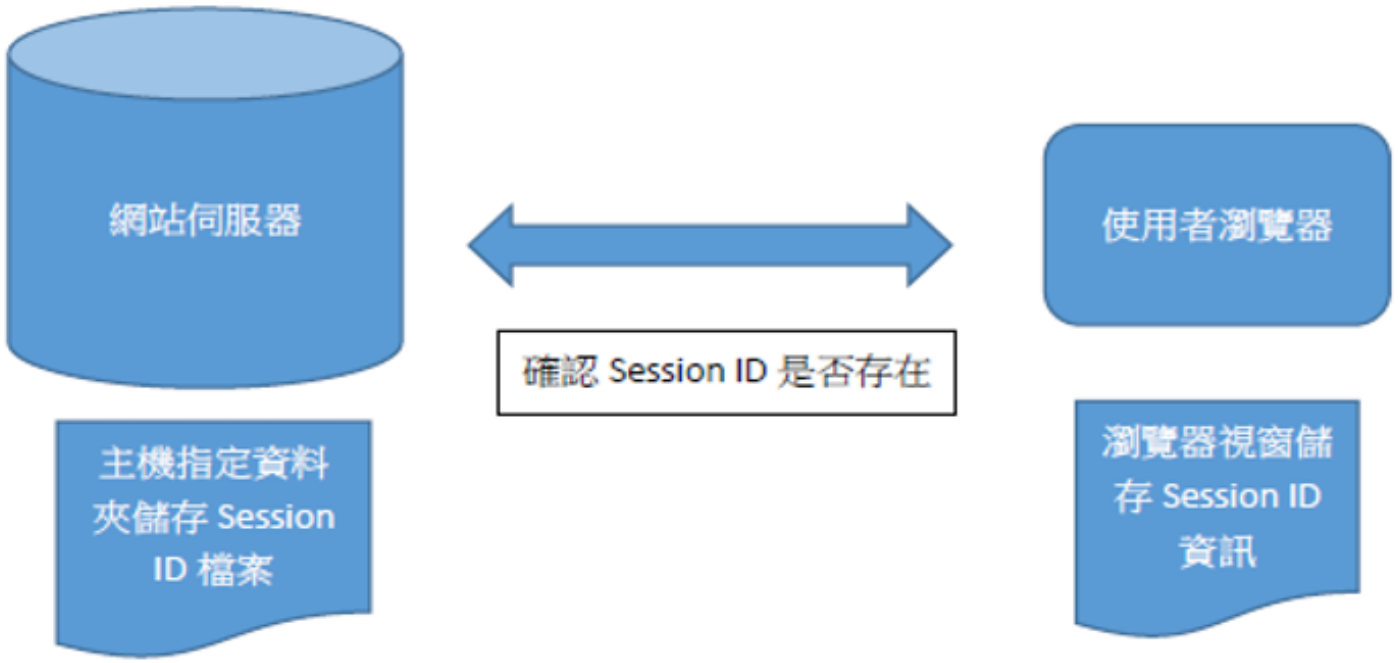
```python
with tf.Session() as sess:
    init = tf.global_variables_initializer()
    sess.run(init)
    X_array = np.array([[0.4,0.2 ,0.4],
                        [0.3,0.4 ,0.5],
                        [0.3,-0.4,0.5]])
    (_b,_W,_X,_y)=sess.run((b,W,X,y),
                           feed_dict={X:X_array})
```



Outline

- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

SESSION


網站伺服器
使用者瀏覽器
確認 Session ID 是否存在
主機指定資料夾儲存 Session ID 檔案
瀏覽器視窗儲存 Session ID 資訊

Outline

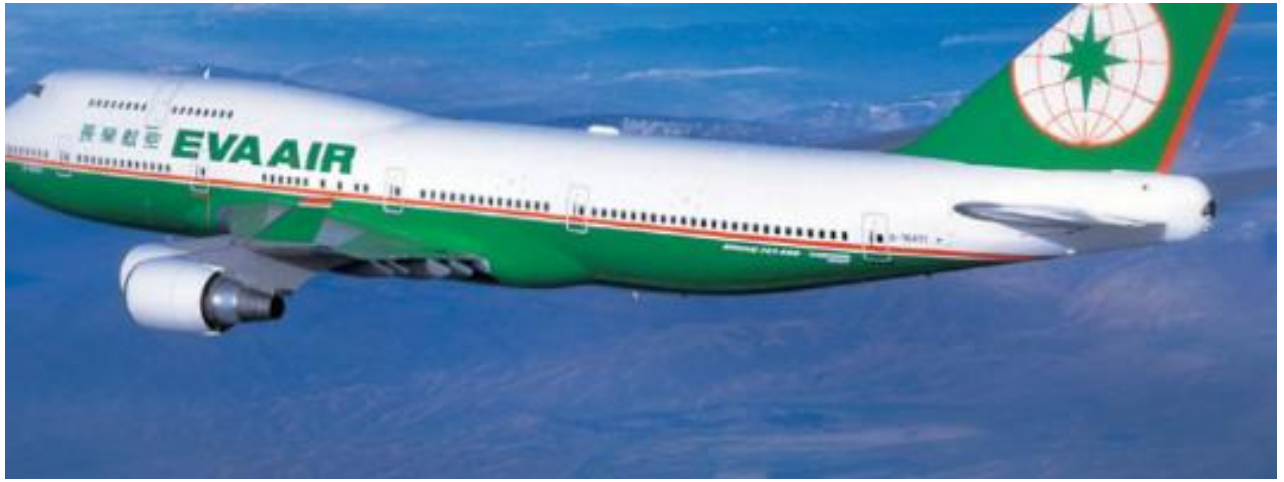
- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

Before we start

Before we start

Before we start

Before we start



Before we start

Before we start

Before we start

Before we start

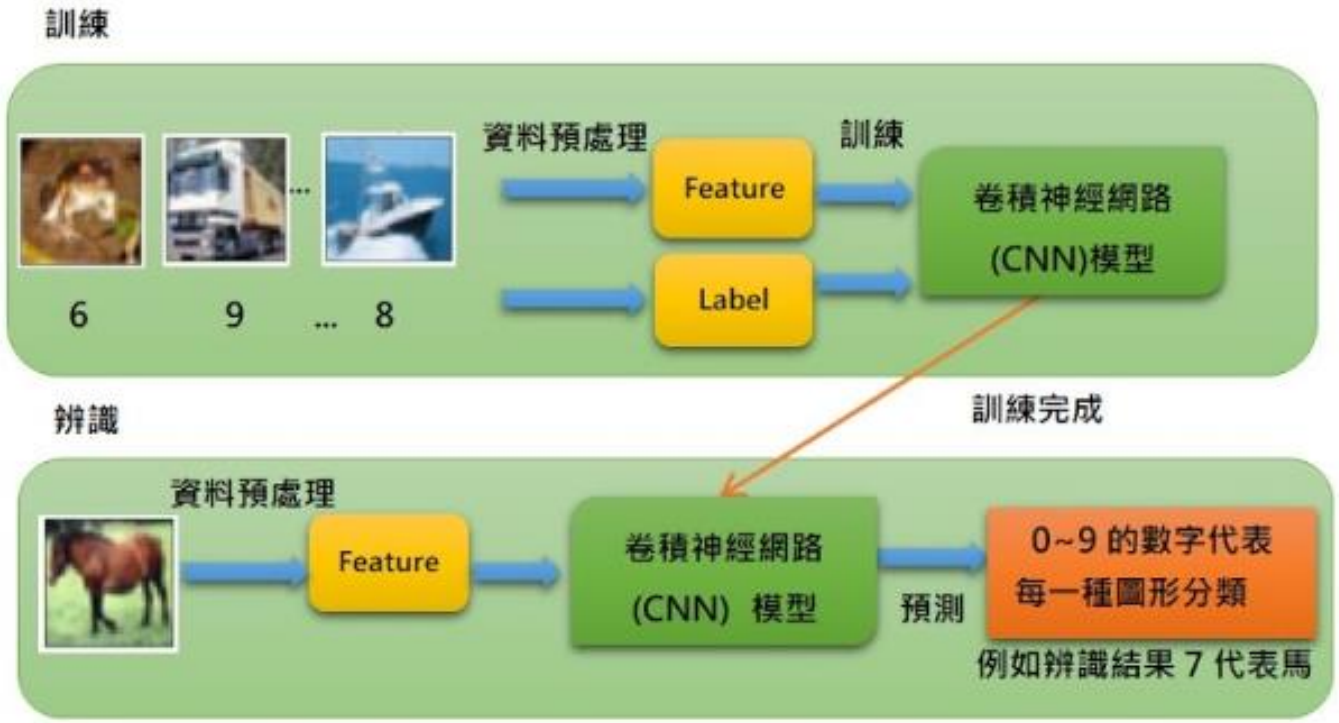
Before we start

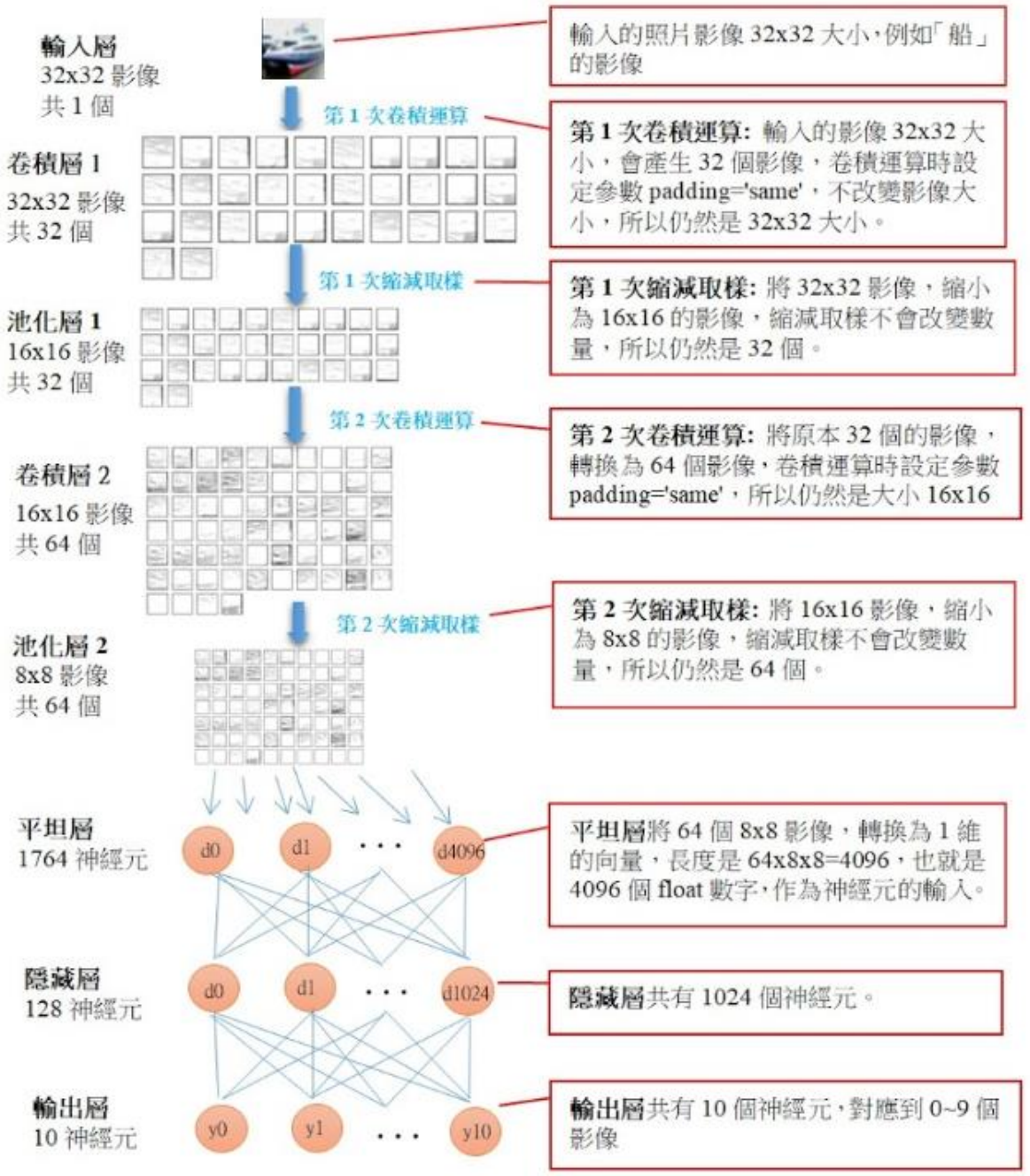
Before we start

Outline

- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

Train and Recognize




輸入層
32x32 影像
共 1 個
輸入的照片影像 32x32 大小，例如「船」的影像
第 1 次卷積運算
第 1 次卷積運算: 輸入的影像 32x32 大小，會產生 32 個影像，卷積運算時設定參數 padding='same'，不改變影像大小，所以仍然是 32x32 大小。
卷積層 1
32x32 影像
共 32 個
第 1 次縮減取樣
第 1 次縮減取樣: 將 32x32 影像，縮小為 16x16 的影像，縮減取樣不會改變數量，所以仍然是 32 個。
池化層 1
16x16 影像
共 32 個
第 2 次卷積運算
第 2 次卷積運算: 將原本 32 個的影像，轉換為 64 個影像，卷積運算時設定參數 padding='same'，所以仍然是大小 16x16
卷積層 2
16x16 影像
共 64 個
第 2 次縮減取樣
第 2 次縮減取樣: 將 16x16 影像，縮小為 8x8 的影像，縮減取樣不會改變數量，所以仍然是 64 個。
池化層 2
8x8 影像
共 64 個
平坦層
1764 神經元
d0
d1
d4096
平坦層將 64 個 8x8 影像，轉換為 1 維的向量，長度是 64x8x8=4096，也就是 4096 個 float 數字，作為神經元的輸入。
隱藏層
128 神經元
d0
d1
d1024
隱藏層共有 1024 個神經元。
輸出層
10 神經元
y0
y1
y10
輸出層共有 10 個神經元，對應到 0~9 個影像

Outline

- **History of TensorFlow**
- **Structure of TensorFlow**
- **Data flow of *FLOW***
- **Session**
- **CNN Identifications Pictures**
- **Train and Recognize**
- **Probability of Recognize**

Probability of Recognize


```
airplane Probability:0.475268602
automobile Probability:0.004711268
bird Probability:0.025450774
cat Probability:0.000778025
deer Probability:0.013557564
dog Probability:0.000000456
frog Probability:0.000022055
horse Probability:0.000008237
ship Probability:0.479947060
truck Probability:0.000255971
```

預測「飛機」的機率次高

預測「船」的機率最高