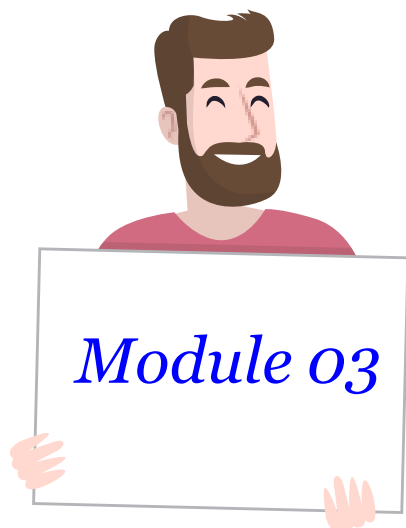




# TensorFlow基礎使用



designed by  freepik

Estimated time:  
**45** min.



資訊工業策進會 Institute for Information Industry

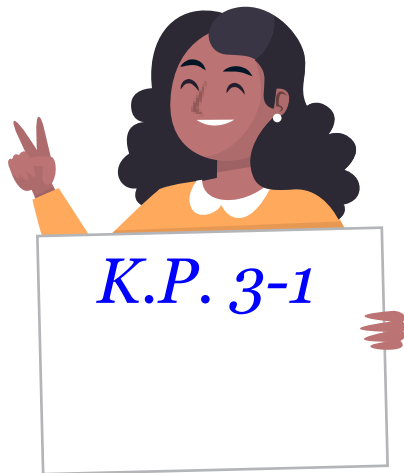
# 學習目標

- 3-1: TensorFlow介紹
- 3-2: TensorFlow基本運算
- 3-3: TensorFlow常見函數



# 3-1: TensorFlow 介紹

- 不同深度學習框架
- TensorFlow 介紹



designed by freepik

# 不同深度學習框架

- 現在要時做深度學習，市面上的框架非常多
  - 目前最熱門的幾個為TensorFlow、Keras、Pytorch
  - 其中比較特別的是，Keras底層其實是TensorFlow，只是它把TensorFlow包得更高層，更容易使用，但其缺點是很難去修改函數內的行為

Caffe

Microsoft  
CNTK

TensorFlow

PyTorch

torch

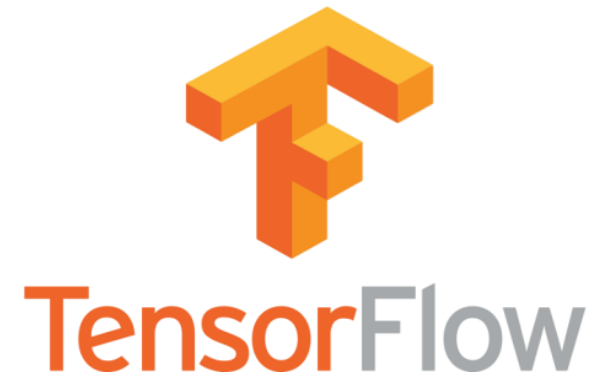
dmlc  
mxnet

Chainer

K

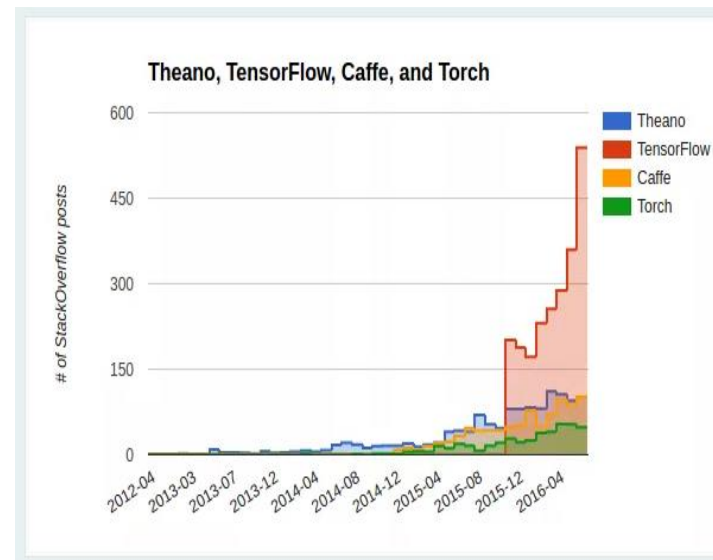
# TensorFlow介紹

- TensorFlow為著名的深度學習開源軟體
  - 由Google Brain團隊於2015年11月釋出
  - <https://github.com/tensorflow/tensorflow>
- 支援許多好用工具
  - 可視化工作Tensorboard
  - 加速讀寫的檔案格式tfrecord
- 支援多GPU以及分散式運算環境



# TensorFlow 熱門程度

- TensorFlow其實自從一推出後，就有非常多的討論度
  - 我們也可以從中推得有不少開發者、社群再討論它，因此本門課程實作會以TensorFlow為主



<https://www.zdnet.com/article/caffe2-deep-learning-wide-ambitions-flexibility-scalability-and-advocacy/>

# 什麼是"Tensor"

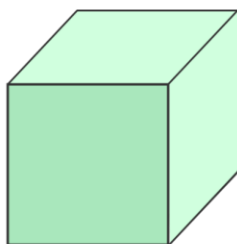
- **Tensor**代表的是一個高維度的陣列
  - **1D**代表是向量、**2D**則是矩陣，這樣的觀念可以拓展到更高維度，這也是**Tensor**存在的意義



1d-tensor



2d-tensor



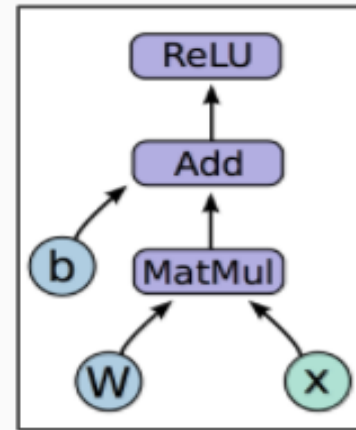
3d-tensor



4d-tensor

# 什麼是”Flow”

- 在TensorFlow裡面，做運算的時候，TensorFlow都會維繫一張名為計算圖的有向圖
  - 在此圖當中，每個節點代表一個運算元
  - 在此圖當中，每個邊代表一個張量結果
- Flow的意思是資料輸入到計算圖中，資料好像在途中流動
  - 因此才會用Flow這個字





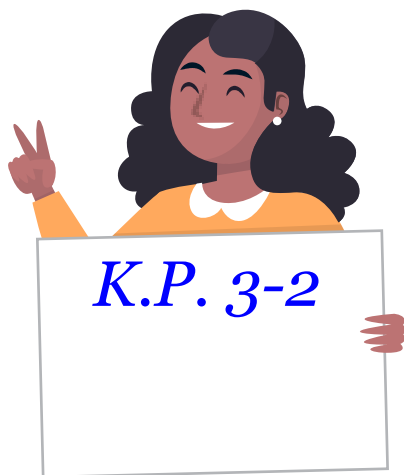
# TensorFlow 特性總攬

- 以下是TensorFlow的總特性，幾個很重要的性質如下
  - 支援Python及C++介面(不建議使用，太少人用了)
  - 有CPU版本以及GPU版本
  - 支援分散是運算
  - 自動微分

Programming Model	<ul style="list-style-type: none"><li>• Dataflow-like model</li></ul>
Language	<ul style="list-style-type: none"><li>• Python</li><li>• C++</li></ul>
Deployment	<ul style="list-style-type: none"><li>• Code once, Run everywhere</li></ul>
Computing Resource	<ul style="list-style-type: none"><li>• CPU</li><li>• GPU</li></ul>
Distribution Process	<ul style="list-style-type: none"><li>• Local Implementation</li><li>• Distributed Implementation</li></ul>
Math Expressions	<ul style="list-style-type: none"><li>• Math Graph Expression</li><li>• Auto Differentiation</li></ul>
Optimization	<ul style="list-style-type: none"><li>• Auto Elimination</li><li>• Kernel Optimization</li><li>• Communication Optimization</li><li>• Support model, data parallelism</li><li>• ...</li></ul>

## 3-2: TensorFlow 基本運算

- 加減乘除基本運算
- 張量型態
- **Constant**、**Variable**、**Placeholder**
- 常見函數使用



designed by freepik

# 加減乘除基本運算

- **TensorFlow**基礎加減乘除語法如下：
  - 可以使用如**tf.add**這種英文式的寫法或是符號式的寫法都行
  - 英文式的寫法會比較推薦，對於大的專案來說後許比較好維護程式碼

加減乘除語法	符號式加減乘除語法	描述
tf.add(x, y)	+	32 bits floating point
tf.subtract(x, y)	-	64 bits floating point
tf.multiply(x, y)	*	8 bits signed integer
tf.div(x,y)	/	16 bits signed integer
tf.mod(x,y)	%	32 bits signed integer
tf.negative(x)	-	64 bits signed integer

# 張量型態

- TensorFlow跟許多程式語言一樣，支援非常多不同的型別

型別	描述
tf.float32	32 bits floating point
tf.float64	64 bits floating point
tf.int8	8 bits signed integer
tf.int16	16 bits signed integer
tf.int32	32 bits signed integer
tf.int64	64 bits signed integer
tf.uint8	8 bits unsigned integer
tf.uint16	16 bits unsigned integer
tf.string	Variable length byte arrays. Each element of a Tensor is a byte array
tf.bool	Boolean

# Constant、Variable、Placeholder

- TensorFlow內建三種常見的變數方法
- Constant
  - 當被指定為constant的時候，未來無法更改數值
- Variable
  - 當被指定為variable的時候，需要給予一個初始值，此數值未來可以被更改
- Placeholder
  - 跟TensorFlow說未來資料會從這裡輸入，先預留一個空間

# 常見函數使用

- **Random Generate Function**是TensorFlow裡常被用來產生隨機變數的函數
  - 其支援不同種類的機率分布來產生隨機變數

函數名稱	機率分布	主要函數參數
tf.random_normal	normal distribution	mean, std, data type
tf.truncated_normal	normal distribution within two std	mean, std, data type
tf.random_uniform	uniform distribution	min value, max value, data type
tf.random_gamma	gamma distribution	alpha, beta, data type

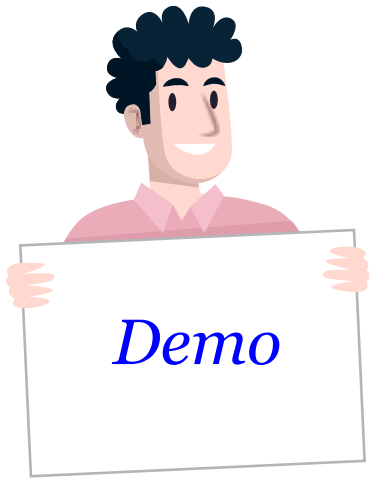
# 常見函數使用

- **Constant Generate Function**是TensorFlow裡常被用來產生constant的方法

函數名稱	功能	範例
tf.zeros	produce all zeros	tf.zeros([2,1], int32)
tf.ones	produce all ones	tf.ones([2,3], int32)
tf.fill	produce specific number	tf.fill([2,1], 9)
tf.constant	produce a given constant	tf.constant([2.0,1.0])

# Demo\_3-2

- 開啟Demo\_3-2.ipynb
- TensorFlow基礎操作
- Constant, Variable, Placeholder使用

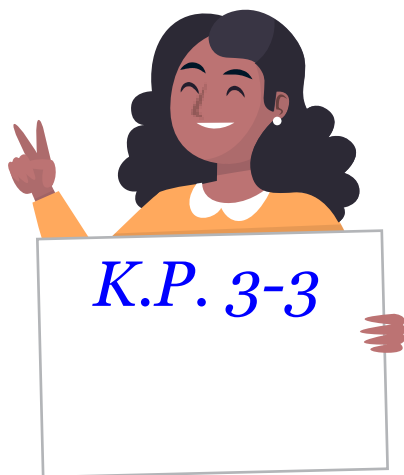


designed by freepik



## 3-3: TensorFlow 常見函數

- 優化器使用
- 模型儲存與部屬



designed by freepik

# 優化器使用

- 優化器在TensorFlow裡非常常被拿來使用
  - 其概念是給定一個函數以及一個起始值，讓TensorFlow自動去幫忙找區域最佳值
  - 優化器的原理後面會跟各位同學做介紹

# 優化器使用

- **TensorFlow**支援非常多不同種類之優化器
  - 目前還沒有任何理論可以說哪一個優化器最好，只能說不同情況會有最適合的優化器

優化器	描述
<code>tf.train.GradientDescentOptimizer</code>	Gradient Descent
<code>tf.train.AdamOptimizer</code>	Adam
<code>tf.train.RMSPropOptimizer</code>	RMSProp

# 模型儲存與部屬

- 在TensorFlow當中，可以使用saver物件來儲存/恢復模型資料
- **Saver**物件儲存模型成功後，會產生下列三個檔案
  - model.ckpt.meta(保存計算圖)
  - model.ckpt(保存計算圖上的參數)
  - checkpoint(紀錄不同時期存入的模型編號)

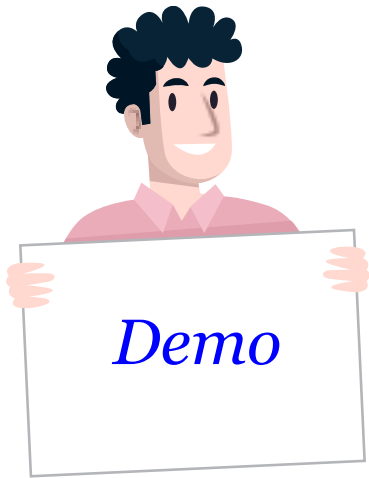
# 模型儲存與部屬

- **TensorFlow saver**語法如下
  - 可以使用這些方法來儲存或是恢復模型
  - 在實務上是非常重要的技巧

Saver物件操作	描述
<code>saver = tf.train.Saver()</code>	創建saver物件
<code>saver.save(sess, "Saved_model/model.ckpt")</code>	儲存計算圖上的參數
<code>saver.restore(sess, "Saved_model/model.ckpt")</code>	恢復計算圖上的參數

# Demo 3-3

- 開啟Demo\_3-3.ipynb
- TensorFlow優化器使用
- TensorFlow模型儲存



designed by freepik

# 線上Corelab

- 題目1：請使用`tf.session`的方式計算A與B四則運算(基礎)
- 題目2：請使用`tf.session`的方式計算A與B四則運算(進階)
- 題目3：請使用`tf.subtract`、`tf.divide`等方式計算  $(10/2)-1$  的答案

# 本章重點精華回顧

- 不同深度學習框架
- TensorFlow介紹
- TensorFlow基本語法使用
- TensorFlow常見函數使用





# Lab: TensorFlow基礎使用

- Lab01: TensorFlow基礎使用
- Lab02: TensorFlow優化器使用
- Lab03: TensorFlow模型儲存

Estimated time:

**20** minutes

