

Day 66

# Keras Introduction

## Keras 的介紹與應用



出題教練

陳宇春

# 本日知識點目標

- 了解 Keras
- 安裝並設定使用 Keras

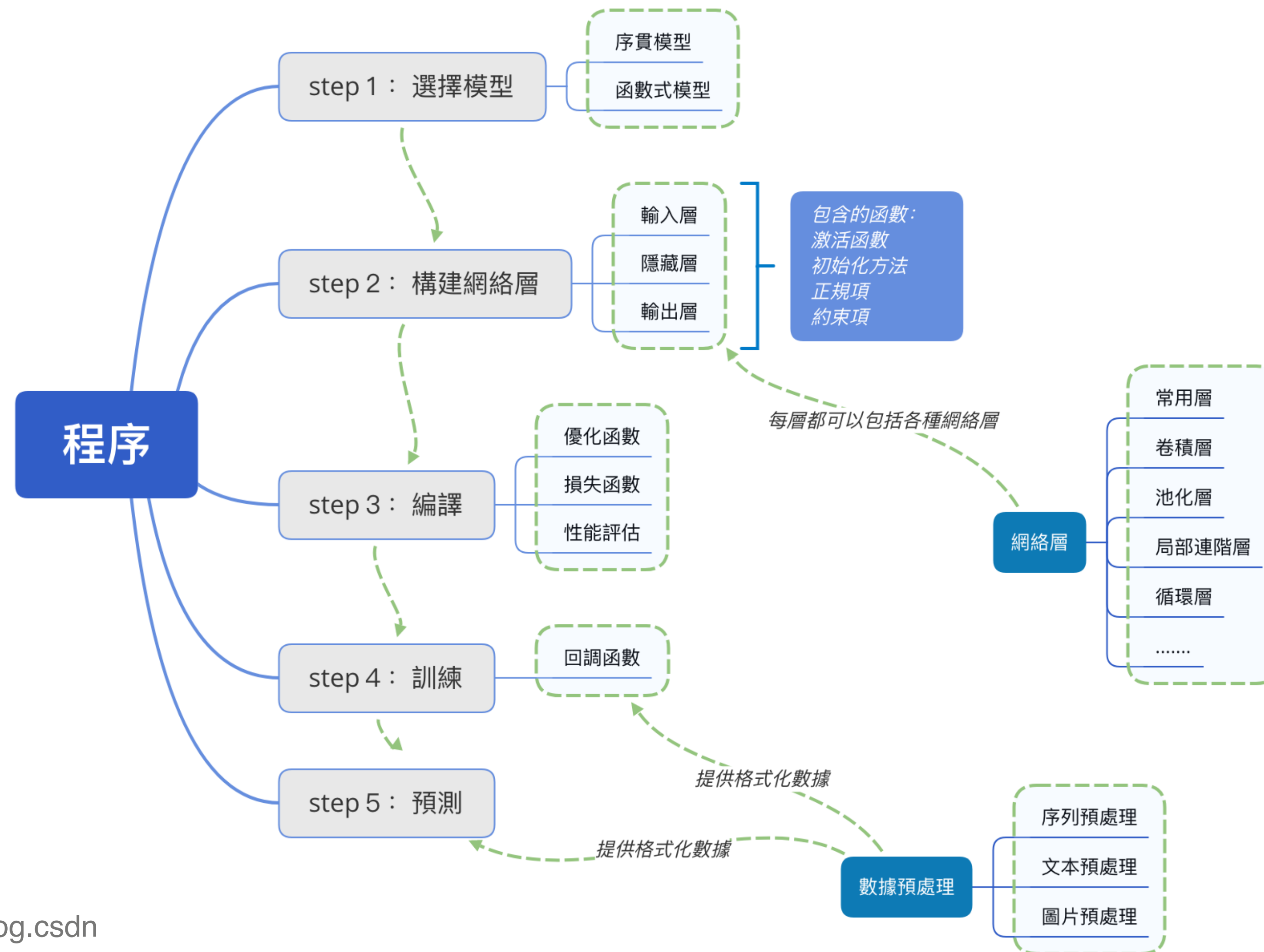


# WHY Keras



- Keras 是一個高層神經網絡 API，Keras 由純 Python 編寫而成並基於 [Tensorflow](#)、[Theano](#) 以及 [CNTK](#) 後端。
- 簡易和快速的原型設計（keras 具有高度模塊化，極簡，和可擴充特性）
- 支持 CNN 和 RNN，或二者的結合
- 無縫 CPU 和 GPU 切換
- Python 協作：Keras 沒有單獨的模型配置文件類型（作為對比，caffe 有），模型由 python 代碼描述，使其更緊湊和更易 debug，並提供了擴展的便利性

# Keras架構圖說明



# Keras安裝 (Linux)



在 終端 中輸入：

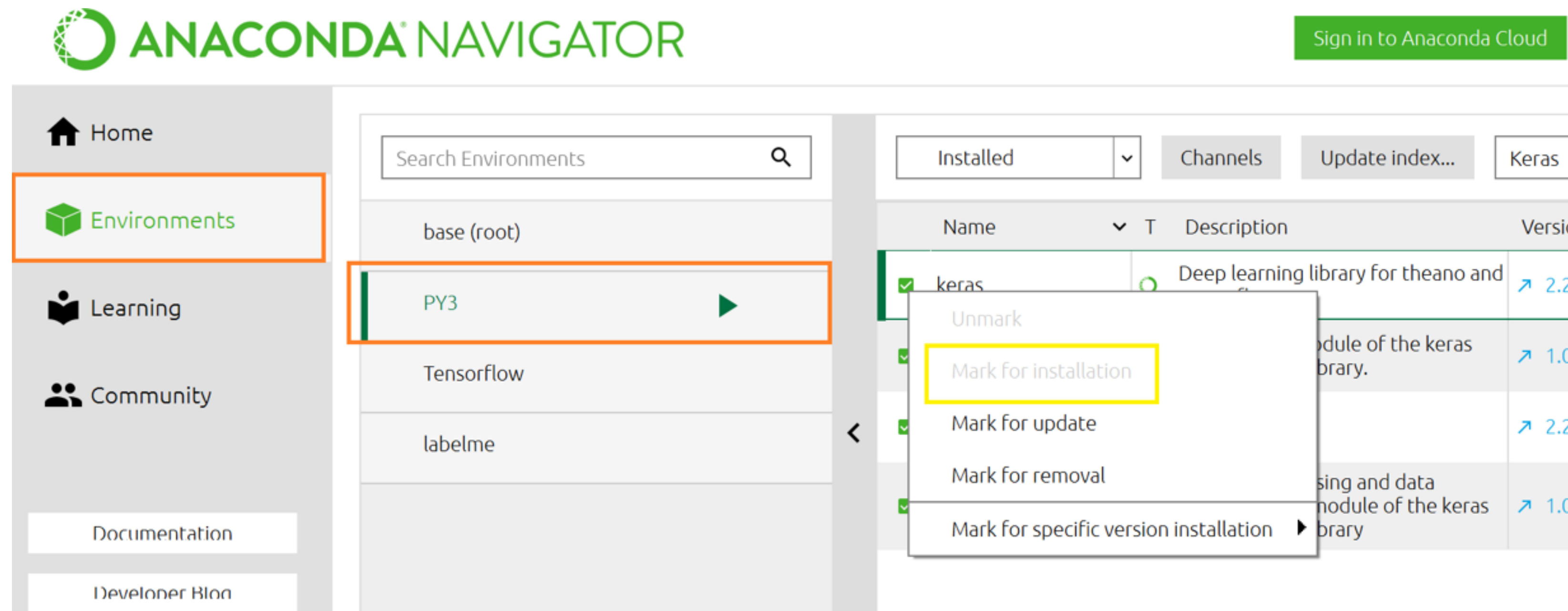
```
>>> sudo pip install -U --pre pip setuptools wheel
>>> sudo pip install -U --pre numpy scipy matplotlib scikit-learn scikit-image
>>> sudo pip install -U --pre tensorflow-gpu
# >>> sudo pip install -U --pre tensorflow ## CPU版本
>>> sudo pip install -U --pre keras
```

安裝完畢後，輸入 `python`，然後輸入：

```
>>> import tensorflow
>>> import keras
```

無錯輸出即可

# Keras安裝 (Anaconda)



# Keras框架搭建 (1) – CuDA, GPU only



下載地址：<https://developer.nvidia.com/cuda-downloads>

### Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture ⓘ	x86_64	ppc64le				
Distribution	Fedora	OpenSUSE	RHEL	CentOS	SLES	Ubuntu
Version	16.04	14.04				
Installer Type ⓘ	runfile [local]	deb [local]	deb [network]	cluster [local]		

### Related Links

- [CUDA Quick Start Guide](#)
- [Release Notes](#)
- [EULA](#)
- [Online Documentation](#)
- [CUDA Toolkit Overview](#)
- [Installer Checksums](#)
- [Open Source Packages](#)
- [Legacy CUDA Toolkits](#)

### Download Installer for Linux Ubuntu 16.04 x86\_64

The base installer is available for download below.

> Base Installer

Download (1.9 GB) ⬇️

Installation Instructions:

1. ``sudo dpkg -i cuda-repo-ubuntu1604-8-0-local_8.0.44-1_amd64.deb``
2. ``sudo apt-get update``
3. ``sudo apt-get install cuda``

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).  
The checksums for the installer and patches can be found in [Installer Checksums](#).  
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).



# Keras框架搭建 (2) – CuDA, GPU only

- Type the following command in Terminal

```
>>> sudo dpkg -i cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64.deb  
>>> sudo apt update  
>>> sudo apt -y install cuda
```

- 將CUDA路徑添加至環境變量在終端輸入：

```
>>> sudo gedit /etc/profile
```

- 在profile文件中添加：

```
export CUDA_HOME=/usr/local/cuda-8.0  
export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}  
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

- 之後source /etc/profile即可



# Keras 的 Backend (一)

- 利用內置層的 Keras 模型都可以在所有這些後端中移植：您可以使用一個後端訓練模型，然後將其加載到另一個後端（例如，用於部署）。可用的後端包括：
  - TensorFlow 後端（來自谷歌）
  - CNTK 後端（來自微軟）
  - Theano 後端
  - 亞馬遜目前正在為 Keras 開發 MXNet 後端。
- 因此，您的 Keras 模型可以在 CPU 之外的許多不同硬件平台上進行培訓：
  - [NVIDIA GPU](#)
  - [谷歌TPU](#)，通過 TensorFlow 後端和谷歌云
  - 支持 OpenCL 的 GPU，例如來自 AMD 的 GPU，通過 [PlaidML Keras](#) 後端

# Keras 的 Backend (二)

- the Keras configuration file at:  
`$HOME/.keras/keras.json`
- NOTE for Windows Users: Please replace `$HOME` with `%USERPROFILE%`.
- The default configuration file looks like this:

```
{  
"image_data_format": "channels_last",  
"epsilon": 1e-07,  
"floatx": "float32",  
"backend": "tensorflow"  
}
```

只需要將 backend 字段改為：`"theano"`、`"tensorflow"`或者`"cntk"`就可以了

# Keras 的 Backend (三)

```
{  
"image_data_format": "channels_last",  
"epsilon": 1e-07,  
"floatx": "float32",  
"backend": "tensorflow"  
}
```

- image\_data\_format  
(1) theano 是 channel 是 channel\_first，TensorFlow 是 channel\_last。  
(2) 2D 數據：  
channel\_last 就是 (rows, cols, channels)，channel\_first 就是 (channels, rows, cols)。

- epsilon：一個浮點數，用來避免某些情況下的除0操作
- floatX："float16"、"float32" 或者 "float64"，默認的浮點數類型。
- backend："tensorflow"，"theano"

# Keras 的 Backend (四)

---



另外一種方法是在環境變量中添加:

```
1 KERAS_BACKEND=tensorflow python -c "from keras import backend"
2 Using TensorFlow backend
```



# Keras 的 Backend (五)

---

- Using the abstract Keras backend to write new code  
the Keras modules you write to be compatible with both Theano (**th**) and TensorFlow (**tf**), you have to write them via the abstract Keras backend API
- `from keras import backend as K`

# 前述流程 / python程式 對照

---

- python 程式 (請參閱今日範例)
- Anaconda 安裝 Keras 與相關套件
- 導入Keras 與判定版本

```
import keras
```

```
print(keras.__version__)
```

# 重要知識點複習

---

- Anaconda 安裝 Keras 與相關套件
- 導入Keras 與判定版本

# 解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

