

# Colab 使用簡易說明

---

2018/04/14 Hands-on Deep Learning







# 資料位置

---

- 連結在此: <https://goo.gl/mU4Fi7>

AIA\_mgr

名稱 ↑

-  cnn → CNN 課程程式碼
-  dnn → DNN 課程程式碼
-  Colab\_README
-  pkgo\_city66\_class5\_v1.csv → DNN 課程資料檔

# Prerequisite

---

- 需要個人的 Google 帳號才可以執行 Colab

# 開啟檔案

1. 右鍵點選檔案
2. 選擇開啟工具 → Colaboratory



# 啟用與執行

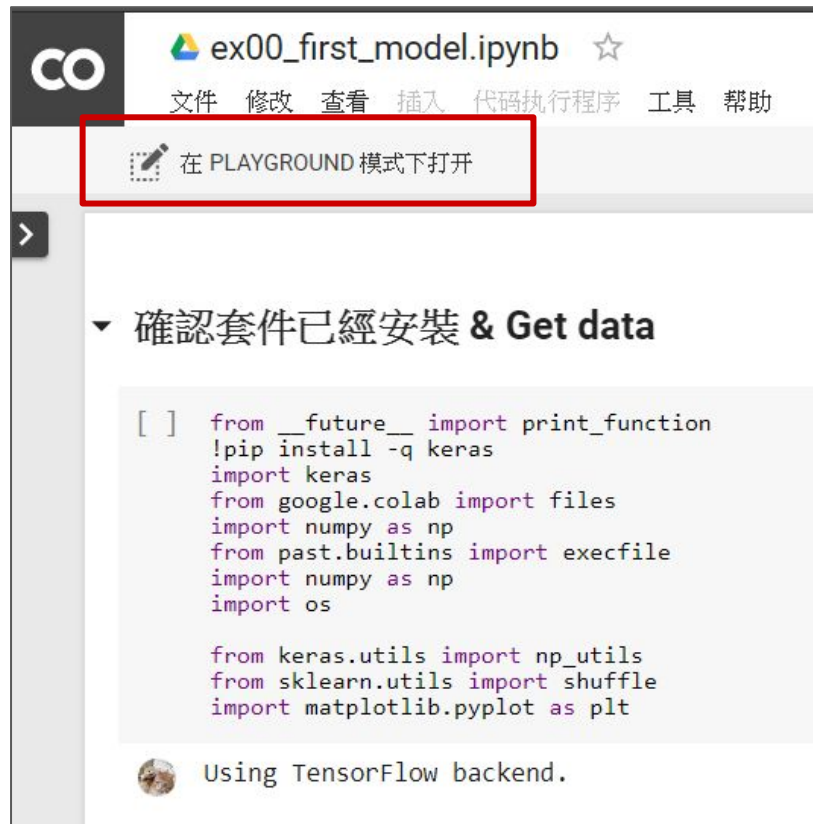
---

要執行本次共享的 Colab notebook 有兩個方式

1. 在 Playground 模式下運行
  - 簡便、快速, 不需佔用個人空間
  - 無法使用 GPU
  - 不能儲存修改後的結果
2. 複製到個人的 Google driver 下執行
  - 可使用 GPU 運行

# 啟用與執行

## 在 Playground 模式下運行



The screenshot shows the Google Colab interface for a notebook named 'ex00\_first\_model.ipynb'. The top navigation bar includes links for '文件' (File), '修改' (Edit), '查看' (View), '插入' (Insert), '代码执行程序' (Code execution), '工具' (Tools), and '帮助' (Help). A red box highlights the button '在 PLAYGROUND 模式下打开' (Open in Playground mode), which is accompanied by a pencil icon. Below this, a section titled '確認套件已經安裝 & Get data' (Confirm packages are installed & Get data) contains a code cell with the following Python code:

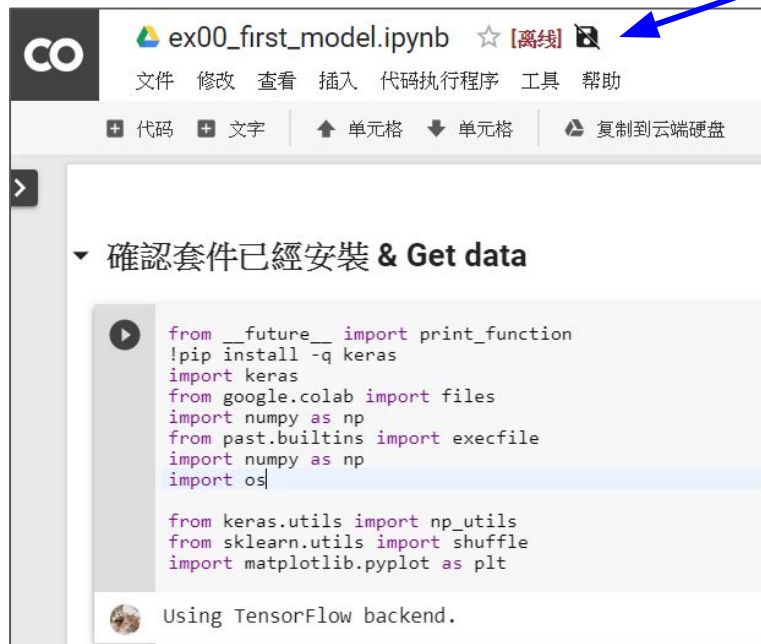
```
[ ] from __future__ import print_function
!pip install -q keras
import keras
from google.colab import files
import numpy as np
from past.builtins import execfile
import numpy as np
import os

from keras.utils import np_utils
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
```

At the bottom of the code cell, it says 'Using TensorFlow backend.'

# 啟用與執行

## 在 Playground 模式下運行

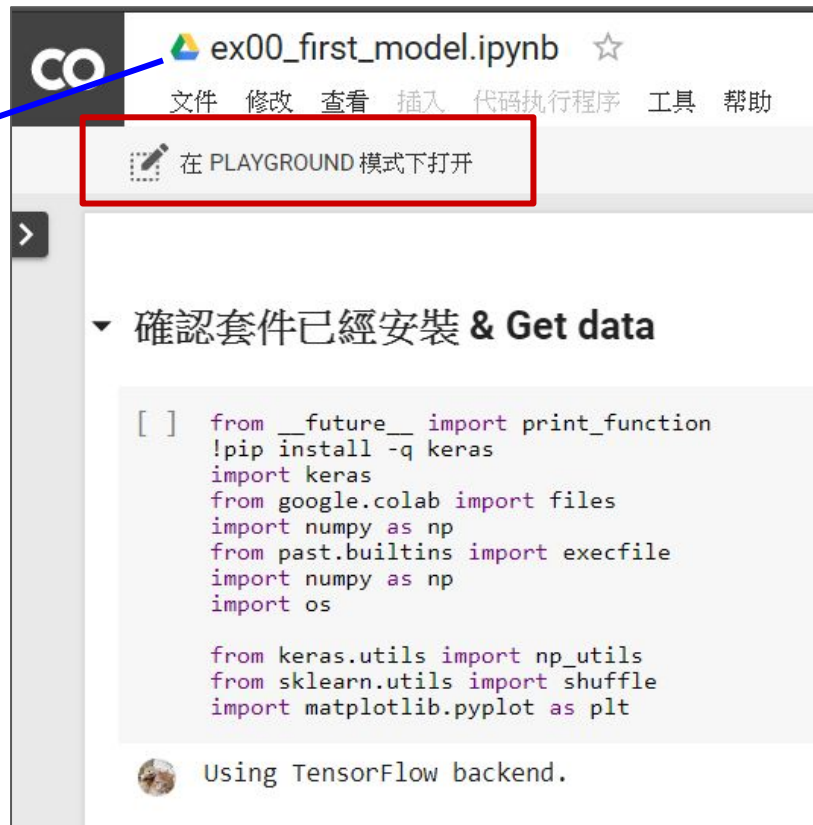


The screenshot shows the Google Colab interface for a notebook named 'ex00\_first\_model.ipynb'. The top bar includes the Colab logo, the notebook name, a star icon, and a status indicator '[离线]' (Offline). Below the top bar is a menu with options: 文件 (File), 修改 (Edit), 查看 (View), 插入 (Insert), 代码执行程序 (Code execution), 工具 (Tools), and 帮助 (Help). A blue arrow points to a small icon in the top right corner of the notebook header, which is used to toggle between different execution modes. Below the menu is a toolbar with icons for adding code cells (+ 代码), text cells (+ 文字), and a dropdown for cell types (单元格). A button '复制到云端硬盘' (Copy to Google Drive) is also visible. The main content area shows a code cell with the following code:

```
from __future__ import print_function
!pip install -q keras
import keras
from google.colab import files
import numpy as np
from past.builtins import execfile
import numpy as np
import os

from keras.utils import np_utils
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
```

At the bottom of the code cell, it says 'Using TensorFlow backend.'



The screenshot shows the Google Colab interface for a notebook named 'ex00\_first\_model.ipynb'. The top bar includes the Colab logo, the notebook name, a star icon, and a status indicator '[离线]' (Offline). Below the top bar is a menu with options: 文件 (File), 修改 (Edit), 查看 (View), 插入 (Insert), 代码执行程序 (Code execution), 工具 (Tools), and 帮助 (Help). A red box highlights a button in the top right corner of the notebook header, which is used to toggle between different execution modes. Below the menu is a toolbar with icons for adding code cells (+ 代码), text cells (+ 文字), and a dropdown for cell types (单元格). A button '复制到云端硬盘' (Copy to Google Drive) is also visible. The main content area shows a code cell with the following code:

```
from __future__ import print_function
!pip install -q keras
import keras
from google.colab import files
import numpy as np
from past.builtins import execfile
import numpy as np
import os

from keras.utils import np_utils
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
```

At the bottom of the code cell, it says 'Using TensorFlow backend.'

# 啟用與執行

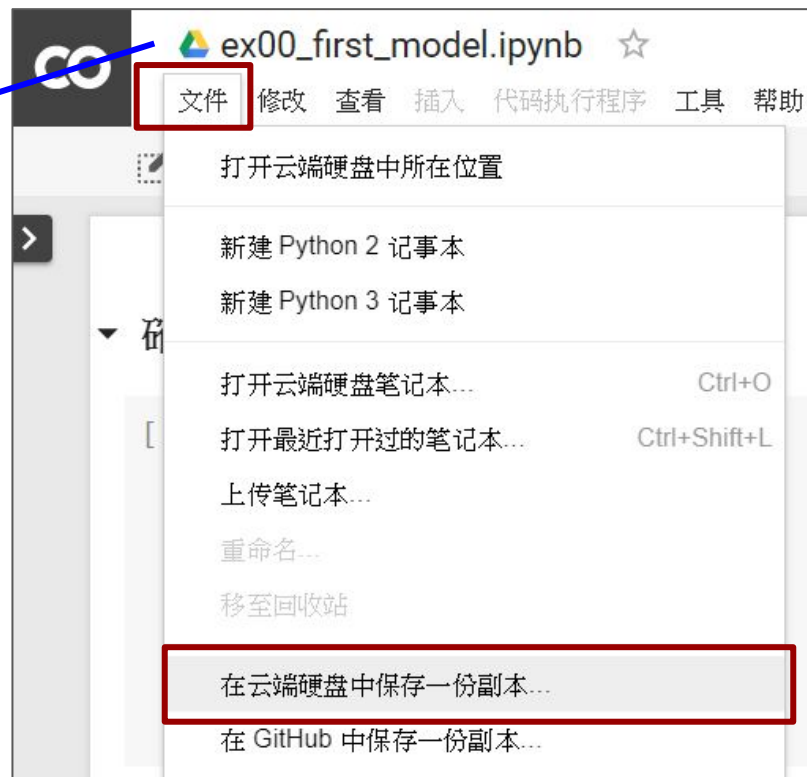
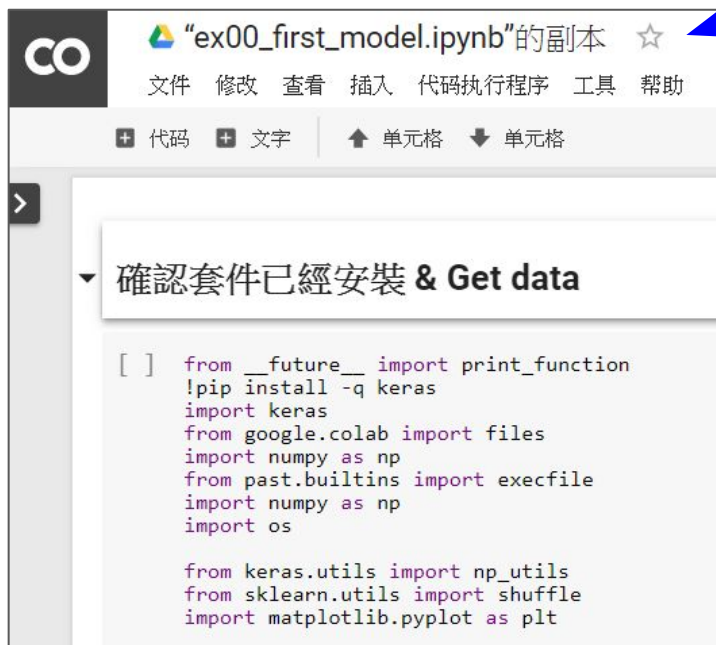
複製到個人的 Google driver 下執行





# 啟用與執行

複製到個人的 Google driver 下執行



# 啟用與執行

---

Method1: 按下每個 Cell 左方的 Run

Method2: 在點選 Cell 後, 按  
Ctrl + Enter 執行

確認套件已經安裝 & Get data



```
from __future__ import print_function
!pip install -q keras
import keras
from google.colab import files
import numpy as np
from past.builtins import execfile
import numpy as np
import os

from keras.utils import np_utils
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
```



Using TensorFlow backend.

# DNN 程式碼中的上傳資料

```
if not os.path.exists("pkgo_city66_class5_v1.csv"):
    uploaded = files.upload()

def readInput(file):
    ''' Read input files '''
    my_data = np.genfromtxt(file, delimiter=',', skip_header=1).

    ''' The first column to the 199th column is used as input features '''
    X_train = my_data[:,0:200]
    X_train = X_train.astype('float32')

    ''' The 200-th column is the answer '''
    y_train = my_data[:,200]
    y_train = y_train.astype('int')

    ''' Convert to one-hot encoding '''
    Y_train = np_utils.to_categorical(y_train,5)

    ''' Shuffle training data '''
    X_train,Y_train = shuffle(X_train,Y_train,random_state=100)
    return X_train, Y_train

# make data #
X_train, Y_train = readInput(file="pkgo_city66_class5_v1.csv")
```

...

選擇檔案 未選擇任何檔案

Cancel upload

- 執行此 Cell 後，會要求上傳檔案
- 將本機端資料上傳
- pkgo\_city66\_class5\_v1.csv

# DNN 程式碼中的上傳資料

```
if not os.path.exists("pkgo_city66_class5_v1.csv"):
    uploaded = files.upload()

def readInput(file):
    ''' Read input files '''
    my_data = np.genfromtxt(file, delimiter=',', skip_header=1)

    ''' The first column to the 199th column is used as input features '''
    X_train = my_data[:,0:200]
    X_train = X_train.astype('float32')

    ''' The 200-th column is the answer '''
    y_train = my_data[:,200]
    y_train = y_train.astype('float32')

    ''' Convert to float32 '''
    Y_train = np.zeros(X_train.shape[0])

    ''' Shuffle '''
    X_train, Y_train = shuffle(X_train, Y_train)

    return X_train, Y_train

# make data #
X_train, Y_train = readInput(file="pkgo_city66_class5_v1.csv")
```

- 執行此 Cell 後，會要求上傳檔案
- 將本機端資料上傳
- pkgo\_city66\_class5\_v1.csv



選擇檔案 未選擇任何檔案

Cancel upload

# DNN 程式碼中的上傳資料

```
if not os.path.exists("pkgo_city66_class5_v1.csv"):
    uploaded = files.upload()

def readInput(file):
    ''' Read input files '''
    my_data = np.genfromtxt(file, delimiter=',', skip_header=1).

    ''' The first column to the 199th column is used as input features '''
    X_train = my_data[:,0:200]
    X_train = X_train.astype('float32')

    ''' The 200-th column is the answer '''
    y_train = my_data[:,200]
    y_train = y_train.astype('int')

    ''' Convert to one-hot encoding '''
    Y_train = np_utils.to_categorical(y_train,5)

    ''' Shuffle training data '''
    X_train,Y_train = shuffle(X_train,Y_train,random_state=100)
    return X_train, Y_train

# make data #
```



選擇檔案 pkgo\_city66\_class5\_v1.csv

- **pkgo\_city66\_class5\_v1.csv**(application/vnd.ms-excel) - 1265155 bytes, last modified: 2017/4/30 - 100% done  
Saving pkgo\_city66\_class5\_v1.csv to pkgo\_city66\_class5\_v1.csv

- 執行此 Cell 後, 會要求上傳檔案
- 將本機端資料上傳
- pkgo\_city66\_class5\_v1.csv

# CNN: From CPU to GPU mode

ex\_CNN\_practice.ipynb ☆

文件 修改 查看 插入 代码执行程序 工具 帮助

+ 代码 + 文字 ↑ 单元

全部运行 Ctrl+F9

运行当前单元格之前的所有单元格 Ctrl+F8

运行光标所在的单元格 Ctrl+Enter

运行所选单元格 Ctrl+Shift+Enter

运行当前单元格以及其后的所有单元格 Ctrl+F10

连接到代码执行程序...

中断执行 Ctrl+M I

重新启动代码执行程序... Ctrl+M .

更改运行时类型

CNN 練習

```
[ ] # Check GPU
import tensorflow as tf
tf.test.gpu_device_name()

[ ] # import package
import keras
```

笔记本设置

运行时类型

Python 3

硬件加速器

None

☐ 保存此笔记本时忽略代码单元格输出项

取消

保存

笔记本设置

运行时类型

Python 3

硬件加速器

GPU

☐ 保存此笔记本时忽略代码单元格输出项

取消

保存

其他程式碼的運行可以請各位先行熟悉  
課堂過程中會再詳細解說!

---