

勞動部產業新尖兵計畫

人工智慧金融應用與實務培訓班



課程模組： AI 金融科技課程 - AI 程式設計

2. 基本的 MLP 操作

葉建華 (Yeh, Jian-hua)

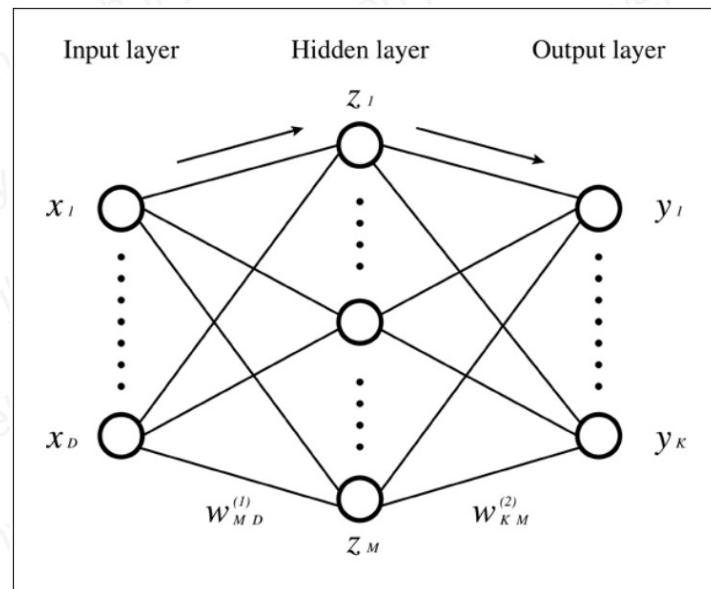
tdi.jhyeh@tdi.edu.tw
au4290@gmail.com

講次內容

- MLP：多層次神經元結構
- Tensorflow 與 Keras 套件介紹
- 基本的 Keras MLP 程式
- 專題：Iris 資料集處理

MLP, Multi-Layer Perceptron

- 由於單一神經元的空間分割能力有限
 - 只能做線性分割
 - 無法解複雜、**非線性**的問題
- 多層次神經元可以解非線性問題
 - 剩下的問題是要「**多複雜**」
 - 越多層造成的誤差倒傳遞問題也會越嚴重



有名的「**梯度消失**」問題！

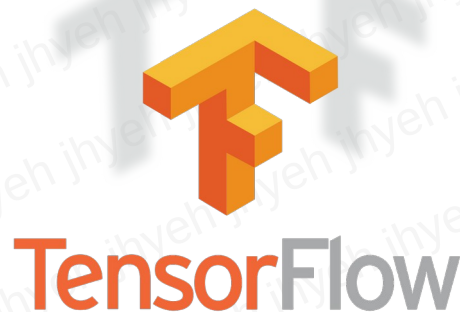
講次內容

- MLP：多層次神經元結構
- Tensorflow 與 Keras 套件介紹
- 基本的 Keras MLP 程式
- 專題：Iris 資料集處理

Tensorflow 與 Keras 介紹

- Tensorflow 是…

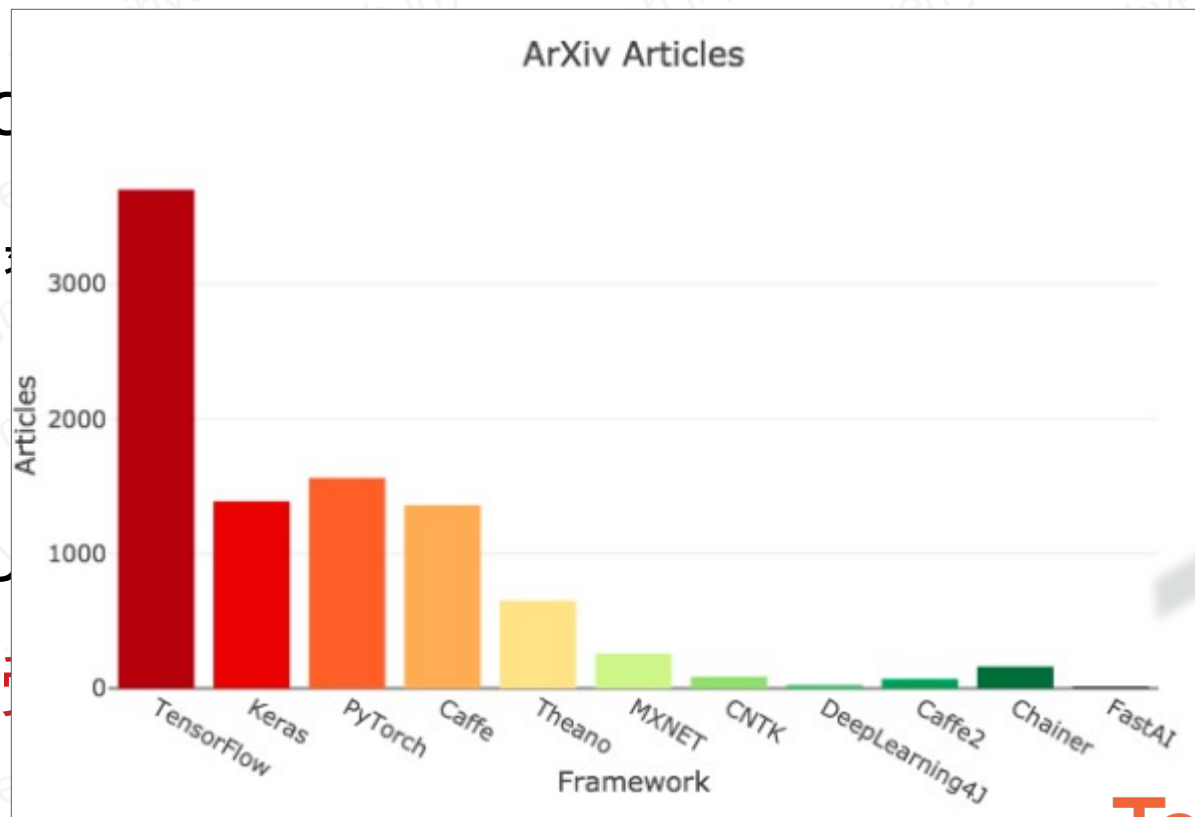
- 機器學習開發平台，**開源**（所以免費）
- 可使用 **Python** 或 C++ 開發
- 有 CPU 與 GPU 兩個版本
- 目前**最受歡迎**平台（研究、應用）



Tensorflow 與 Keras 介紹

- Tensorflow

- 機器學習
- 可使用
- 有 CPU
- 目前最



TensorFlow

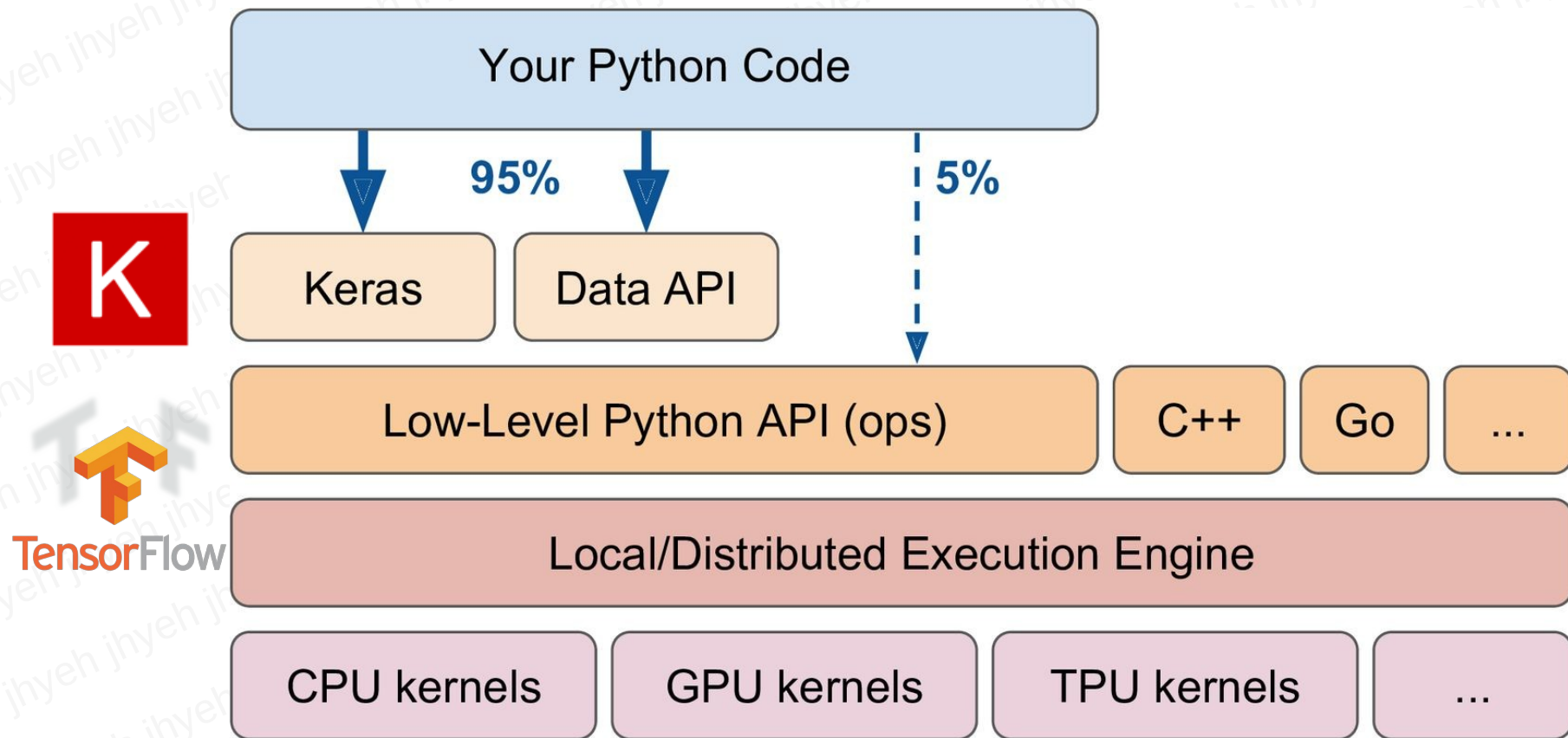
Tensorflow 與 Keras 介紹

- Keras 是…

- Tensorflow 的高階 API，也**開源**
- 以 TensorFlow、Theano 為後端運行 註：Theano 已停止開發
- 以 Python 作為開發語言，容易開發
- 已將模型的各神經層模組化，撰寫方便



Tensorflow 架構與 Keras 角色



安裝 Tensorflow 與 Keras

- 在此介紹 CPU 版本的安裝
 - `pip install tensorflow`



安裝 Tensorflow 與 Keras

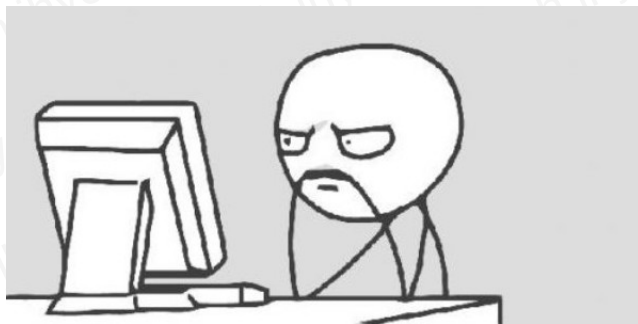
- GPU 版?

- pip install tensorflow-gpu

GPU 版才沒那麼簡單!

1. 裝好 GPU 卡才能裝
2. 裝好 GPU 驅動才能裝
3. 裝好 cuDNN 才能裝

...



本課程以 **CPU 版** 為教學基礎。裝好了？那我們開始吧！

講次內容

- MLP：多層次神經元結構
- Tensorflow 與 Keras 套件介紹
- 基本的 Keras MLP 程式
- 專題：Iris 資料集處理

基本的 Keras MLP 程式

- 如何用 Keras 建立簡單的神經網路
 - 1) 建立 Sequential 模型
 - 2) 建立輸入層、隱藏層
 - 3) 建立輸出層
- 當然還有建模資料的準備 ...

基本的 Keras MLP 程式

1) 建立 Sequential 模型

- `model = Sequential()`

2) 建立輸入層、隱藏層

- `model.add(Dense(256, activation=('sigmoid'), input_shape=(28*28,)))`
- `model.add(Dense(64, activation=('sigmoid')))`

3) 建立輸出層

- `model.add(Dense(10, activation='softmax'))`

真的就差不多這樣而已!



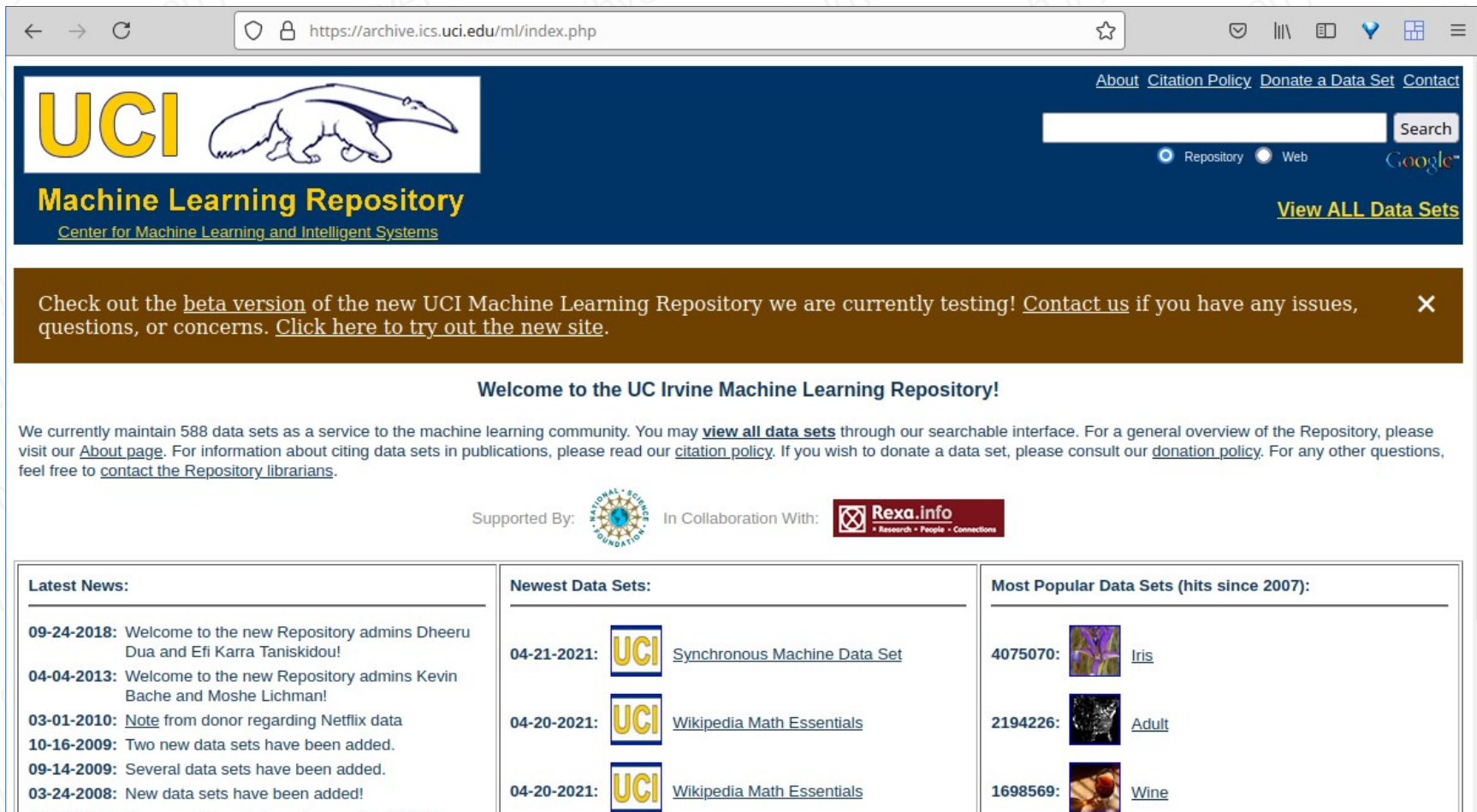
講次內容

- MLP：多層次神經元結構
- Tensorflow 與 Keras 套件介紹
- 基本的 Keras MLP 程式
- 專題：Iris 資料集處理

專題： Iris 資料集處理


- Iris：全世界最有名的機器學習資料集
 - 花朵資料集
 - 英國統計學家 Ronald Fisher 爵士在 1936 年時，對加斯帕半島上的鳶尾屬花朵所量測的花瓣花萼的長寬資料，依照山鳶尾、變色鳶尾、維吉尼亞鳶尾三類進行標示
 - 小型資料集，只有 150 筆資料

Iris 資料集： 400 萬次下載



下載 Iris 資料集

← → ↻ <https://archive.ics.uci.edu/ml/datasets/Iris> 📄 ☆

**UCI**
Machine Learning Repository
Center for Machine Learning and Intelligent Systems


About Citation Policy

☒ Repository

Iris Data Set

Download: [Data Folder](#) [Data Set Description](#)

Abstract: Famous database; from Fisher, 1936



Data Set Characteristics:	Multivariate	Number of Instances:	150	Area:	Life
Attribute Characteristics:	Real	Number of Attributes:	4	Date Donated	1988-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	4075074

Source:

下載 Iris 資料集

The screenshot shows a web browser window with the address bar displaying `https://archive.ics.uci.edu/ml/datasets/Iris`. Below the browser window, the main content of the page is shown, including a list of links and a table of dataset characteristics.

Index of /ml/machine-learning-databases/iris

- [Parent Directory](#)
- [Index](#)
- [bezdekIris.data](#)
- [iris.data](#)
- [iris.names](#) 把 iris.data 抓回來!

Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips SVN/1.7.14 Phusion_Passenger/4.0.53 mod_perl/2.0.11 Perl/v5.16.3 S
443

Attribute Characteristics:	Real	Number of Attributes:	4	Date Donated	1988-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	4075074

Source:

Iris 資料集

Sepal Length	Sepal Width	Petal Length	Petal Width	Label
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
...

可是你看到的是 **CSV** 格式資料！

Iris 資料集

Sepal Length	Sepal Width	Petal Length	Peta
5.1	3.5	1.4	
4.9	3.0	1.4	
7.0	3.2	4.7	
6.4	3.2	4.5	
6.3	3.3	6.0	
5.8	2.7	5.1	
...	

可是你看到的是 **CSV** 格式資料！

File	Edit	Search	View	Document	Help
1	5.1,3.5,1.4,0.2,Iris-setosa				
2	4.9,3.0,1.4,0.2,Iris-setosa				
3	4.7,3.2,1.3,0.2,Iris-setosa				
4	4.6,3.1,1.5,0.2,Iris-setosa				
5	5.0,3.6,1.4,0.2,Iris-setosa				
6	5.4,3.9,1.7,0.4,Iris-setosa				
7	4.6,3.4,1.4,0.3,Iris-setosa				
8	5.0,3.4,1.5,0.2,Iris-setosa				
9	4.4,2.9,1.4,0.2,Iris-setosa				
10	4.9,3.1,1.5,0.1,Iris-setosa				
11	5.4,3.7,1.5,0.2,Iris-setosa				
12	4.8,3.4,1.6,0.2,Iris-setosa				
13	4.8,3.0,1.4,0.1,Iris-setosa				
14	4.3,3.0,1.1,0.1,Iris-setosa				
15	5.8,4.0,1.2,0.2,Iris-setosa				
16	5.7,4.4,1.5,0.4,Iris-setosa				
17	5.4,3.9,1.3,0.4,Iris-setosa				
18	5.1,3.5,1.4,0.3,Iris-setosa				
19	5.7,3.8,1.7,0.3,Iris-setosa				
20	5.1,3.8,1.5,0.3,Iris-setosa				
21	5.4,3.4,1.7,0.2,Iris-setosa				
22	5.1,3.7,1.5,0.4,Iris-setosa				
23	4.6,3.6,1.0,0.2,Iris-setosa				
24	5.1,3.3,1.7,0.5,Iris-setosa				
25	4.8,3.4,1.9,0.2,Iris-setosa				

Iris 使用 MLP 進行學習

- 邏輯流程
 - 讀入資料檔 iris.data
 - 分成 train 和 test 兩部分
 - 資料前處理
 - 根據資料維度，建立 Keras MLP 模型
 - 訓練 train 資料集
 - 用 test 資料集做評估

Iris 使用 MLP 進行學習

- 邏輯流程
 - 讀入資料檔 iris.data (Pandas)
 - 分成 train 和 test 兩部分 (List processing)
 - 資料前處理 (有點 tricky...)
 - 根據資料維度，建立 Keras MLP 模型 (Keras)
 - 訓練 train 資料集 (Keras)
 - 用 test 資料集做評估 (Keras)

Iris 使用 MLP 進行學習

- 讀入資料檔 iris.data (Pandas)

```
import pandas as pd

cols=['Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width', 'Class']
all_df=pd.read_csv('iris.data', names=cols)
print(all_df)
```


Iris 使用 MLP 進行學習

- 讀入資料檔 iris.data (Pandas)

```
import pandas as pd
```

```
cols=['Sepal
```

```
all_df=pd.re
```

```
print(all_df
```

	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

```
[150 rows x 5 columns]
```


Iris 使用 MLP 進行學習

- 分成 train 和 test 兩部分 (List processing)

```
import numpy as np

# 產生一個布林串列，亂數小於0.8為True，反之False
mask=np.random.rand(len(all_df))<0.8
# 用mask串列過濾出train資料集
train_df=all_df[mask]
# 用反向的mask串列過濾出test資料集
test_df=all_df[~mask]
# 故意印出test資料集來看!
print(test_df)
```

Iris 使用 MLP 進行學習

- 分成 train 和 test

```
import numpy as np

# 產生一個布林串列，亂數小於0.8
mask=np.random.rand(len(all))
# 用mask串列過濾出train資料集
train_df=all_df[mask]
# 用反向的mask串列過濾出test資料集
test_df=all_df[~mask]
# 故意印出test資料集來看!
print(test_df)
```

	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	4.7	3.2	1.3	0.2	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
48	5.3	3.7	1.5	0.2	Iris-setosa
49	5.0	3.3	1.4	0.2	Iris-setosa
61	5.9	3.0	4.2	1.5	Iris-versicolor
67	5.8	2.7	4.1	1.0	Iris-versicolor
79	5.7	2.6	3.5	1.0	Iris-versicolor
80	5.5	2.4	3.8	1.1	Iris-versicolor
82	5.8	2.7	3.9	1.2	Iris-versicolor
84	5.4	3.0	4.5	1.5	Iris-versicolor
92	5.8	2.6	4.0	1.2	Iris-versicolor
96	5.7	2.9	4.2	1.3	Iris-versicolor
99	5.7	2.8	4.1	1.3	Iris-versicolor
105	7.6	3.0	6.6	2.1	Iris-virginica
106	4.9	2.5	4.5	1.7	Iris-virginica
107	7.3	2.9	6.3	1.8	Iris-virginica
111	6.4	2.7	5.3	1.9	Iris-virginica
116	6.5	3.0	5.5	1.8	Iris-virginica
117	7.7	3.8	6.7	2.2	Iris-virginica
118	7.7	2.6	6.9	2.3	Iris-virginica
125	7.2	3.2	6.0	1.8	Iris-virginica
129	7.2	3.0	5.8	1.6	Iris-virginica
139	6.9	3.1	5.4	2.1	Iris-virginica
144	6.7	3.3	5.7	2.5	Iris-virginica

隨機的，你不一定看到這樣的結果

Iris 使用 MLP 進行學習

- 資料前處理 (有點 tricky...)
 - 使用 scikit-learn 套件, Anaconda 已裝了
 - 將分類標籤 Iris-setosa, Iris-versicolor, Iris-virginica 轉成數字標籤 (神經網路輸出要用)
 - 用 dict {'Iris-setosa':0, 'Iris-versicolor':1, 'Iris-virginica':2} 代入 DataFrame 的 map() 函數
 - 使用 scikit-learn 的前處理套件 preprocessing 中的 MinMaxScaler 物件
 - 做 DataFrame 內容值的縮放, 對應到 (0, 1) 之間

Iris 使用 MLP 進行學習

- 根據資料維度，建立 Keras MLP 模型 (Keras)
 - 留意 SGD()...

```
from tensorflow.keras.layers import Activation, Dense, Dropout
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import SGD

model = Sequential()
model.add(Dense(128, activation='sigmoid', input_shape=(4, )))
model.add(Dense(32, activation='sigmoid'))
model.add(Dense(3, activation='softmax'))
print(model.summary())
sgd = SGD(lr=0.01, momentum=0.9)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['acc'])
```


Iris 使用 MLP 進行學習

- 根據資料維度，建立 Keras MLP 模型 (Keras)
 - 留意 SGD()...

```
from tensorflow.keras import layers
from tensorflow.keras import models
from tensorflow.keras import optimizers

model = models.Sequential()
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))
print(model.summary())
sgd = optimizers.SGD()
model.compile(loss='categorical_crossentropy', metrics=['acc'])
```

Model: "sequential_27"

Layer (type)	Output Shape	Param #
dense_79 (Dense)	(None, 128)	640
dense_80 (Dense)	(None, 32)	4128
dense_81 (Dense)	(None, 3)	99
Total params: 4,867		
Trainable params: 4,867		
Non-trainable params: 0		
None		

metrics=['acc'])

Iris 使用 MLP 進行學習

- 訓練 train 資料集 (Keras)
 - 出了名的 **model.fit()** 出現了!

```
import matplotlib.pyplot as plt

train_history=model.fit(train_data, train_labels, validation_split=0.2, epochs=500, batch_size=30)

plt.plot(train_history.history['acc'], label='acc')
plt.plot(train_history.history['val_acc'], label='val_acc')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(loc='best')
plt.show()
```

Iris 使用 MLP 進行學習

Epoch 497/500

4/4 [=====] - 0s 11ms/step - loss: 0.0991 - acc: 0.9899 - val_loss: 0.3223 - val_acc: 0.8800

Epoch 498/500

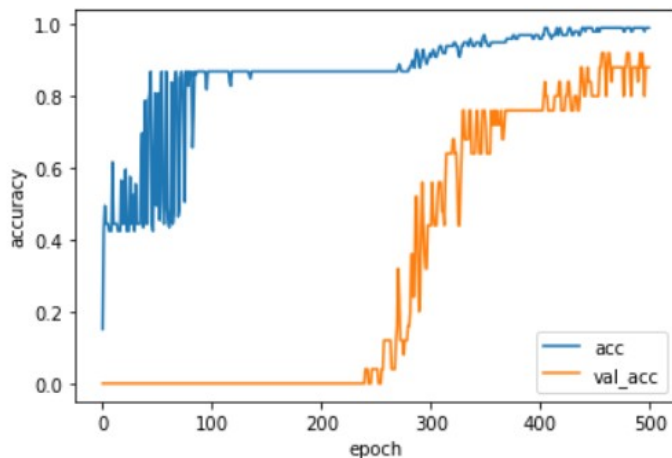
4/4 [=====] - 0s 10ms/step - loss: 0.0978 - acc: 0.9899 - val_loss: 0.3153 - val_acc: 0.8800

Epoch 499/500

4/4 [=====] - 0s 10ms/step - loss: 0.0974 - acc: 0.9899 - val_loss: 0.3157 - val_acc: 0.8800

Epoch 500/500

4/4 [=====] - 0s 11ms/step - loss: 0.0971 - acc: 0.9899 - val_loss: 0.3123 - val_acc: 0.8800



30)

Iris 使用 MLP 進行學習

- 用 test 資料集做評估 (Keras)
 - 使用 model.evaluate()

```
test_loss, test_acc = model.evaluate(test_data, test_labels)
print('loss: {:.3f}'.format(test_loss))
print('accuracy: {:.3f}'.format(test_acc))
```

```
1/1 [=====] - 0s 29ms/step - loss: 0.2341 - acc: 0.8462
loss: 0.234
accuracy: 0.846
```

所以這次的成果就是 **84.6% 正確性!**

xxx_crossentropy

- 評估損失的函數
- 就是分類問題的交叉熵 (Cross Entropy)
 - 說到熵值，很傷腦筋
 - 熱力學借用詞，代表資訊量 ($-\text{plog}(p)$)
 - 交叉熵： $-\text{ylogf}(x)$, y 是答案 $f(x)$ 是預測

Claude Shannon



binary_crossentropy

- 就是兩類分類問題的交叉熵 (Cross Entropy)
 - 交叉熵: $-y \log f(x)$, y 是答案 $f(x)$ 是預測

$$BCE(x)_i = -[y_i \log f_i(x) + (1 - y_i) \log(1 - f_i(x))]$$

$$BCE(x) = \frac{\sum_{i=1}^C BCE(x)_i}{C}$$

categorical_crossentropy

- categorical_crossentropy?
 - 就是多類分類問題的交叉熵 (Cross Entropy)

這個講次中，你應該學到了 ...

- 認識了 Tensorflow 和 Keras
- 了解在 Keras 下基本的神經網路架構
- 做了全世界最有名的 Iris MLP 實驗！