

# TensorFlow + Keras

2019 Ian Fan 范財銘

# 解決問題 Regression / Classification 開發工具 TensorFlow / Keras 神經網路 DNN / CNN



# 「機器學習」

就是讓機器根據一些訓練資料，自動找出有用的函數 (function)，我們可以把找出最佳函數（神經元參數）的過程，想像成是機器在「學習」。

# 「深度學習」

類神經網絡由很多神經元連接而成，人類只需要決定類神經網絡的連結方式，機器可以自己根據訓練資料找出每個神經元的參數。

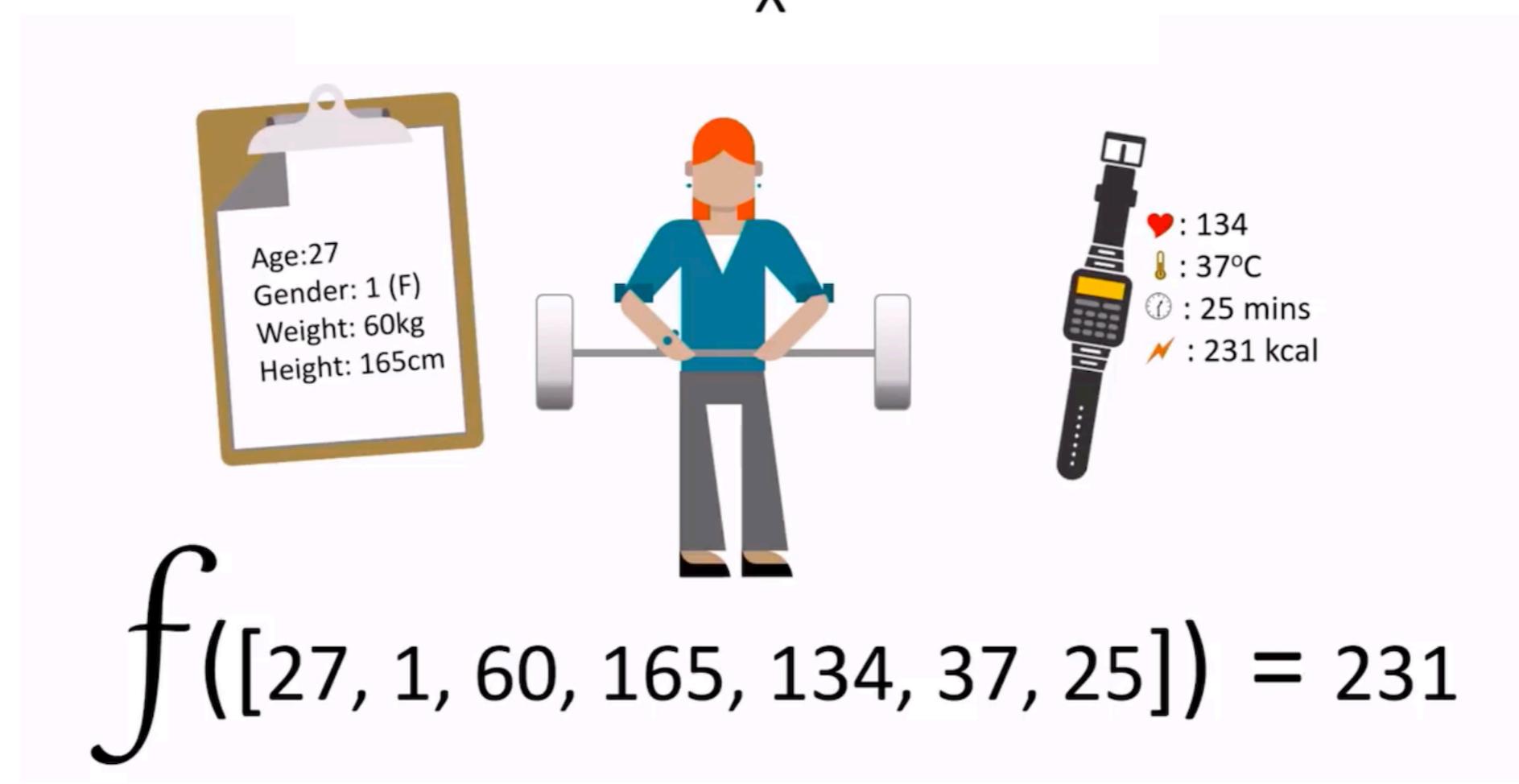
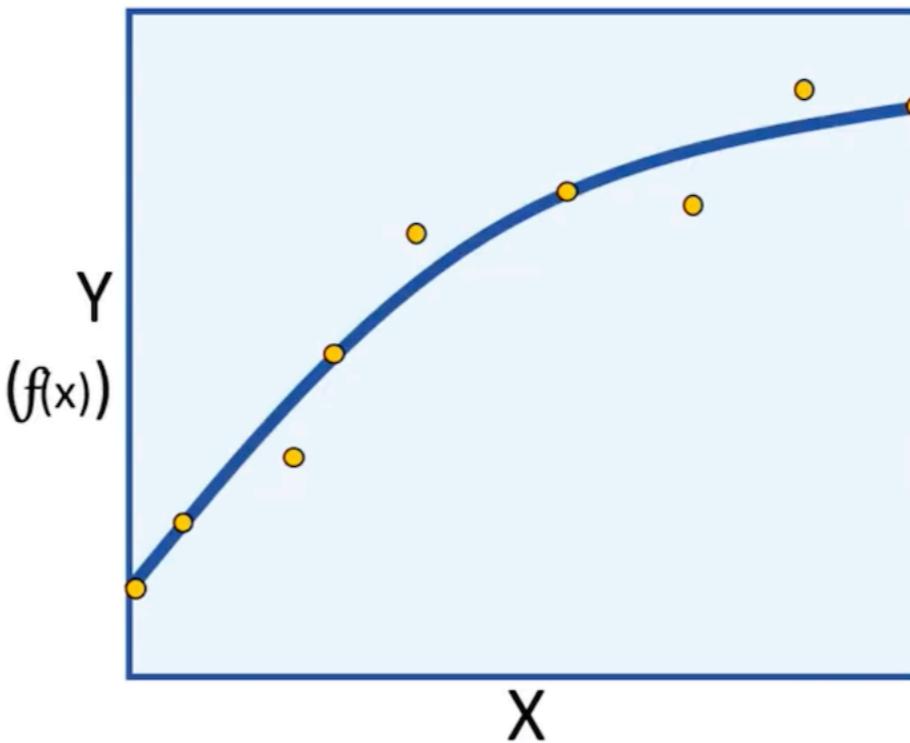
例如：

輸入一段聲音，輸出對應的文字

輸入一張圖片，輸出圖片中物件的名稱

輸入棋盤所有黑子與白子的位置，輸出下一步落子的位置

參考自李宏毅教授



參考自 Microsoft 線上課程

# 機器學習可以解決的問題：

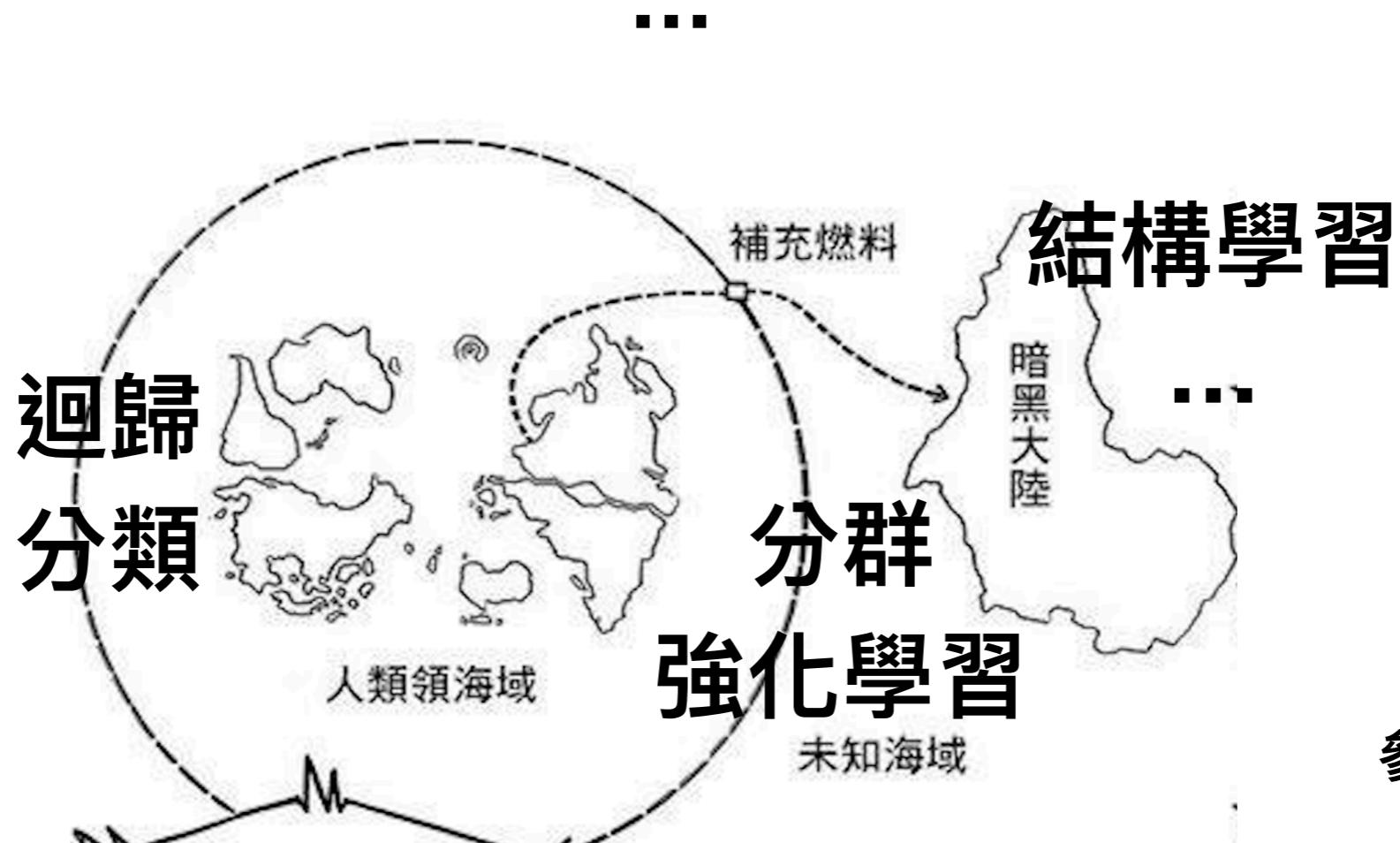
\* Regression 迴歸 \*

\* Classification 分類 \*

Clustering 分群 / 聚類

Reinforcement Learning 強化學習

Structure Learning 結構學習



參考自《獵人》

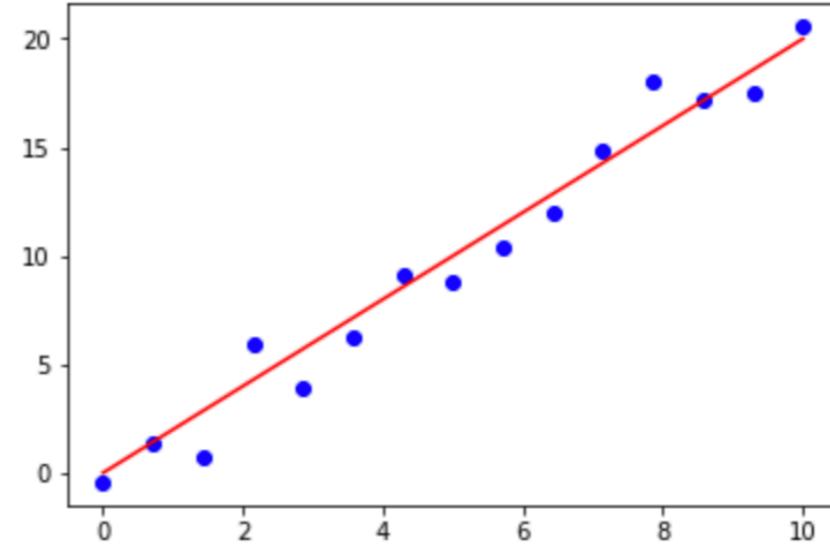
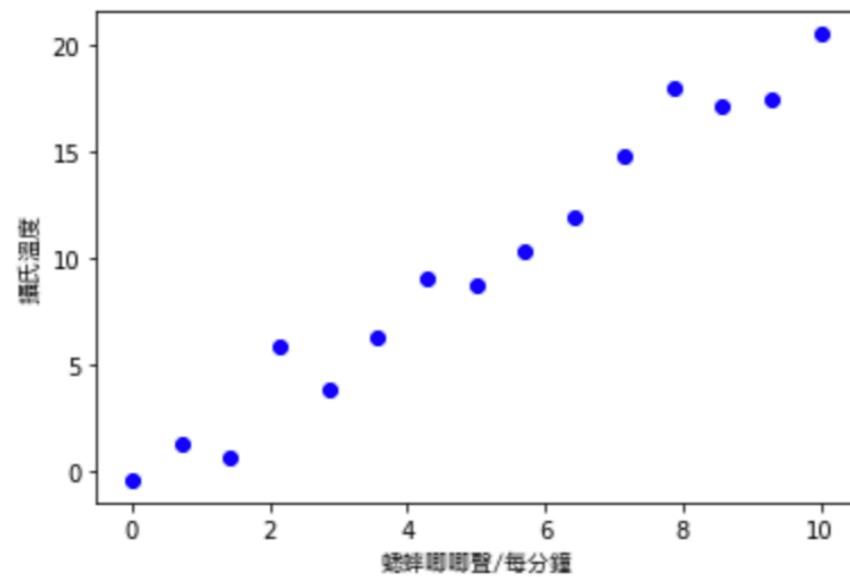
# Regression 迴歸：

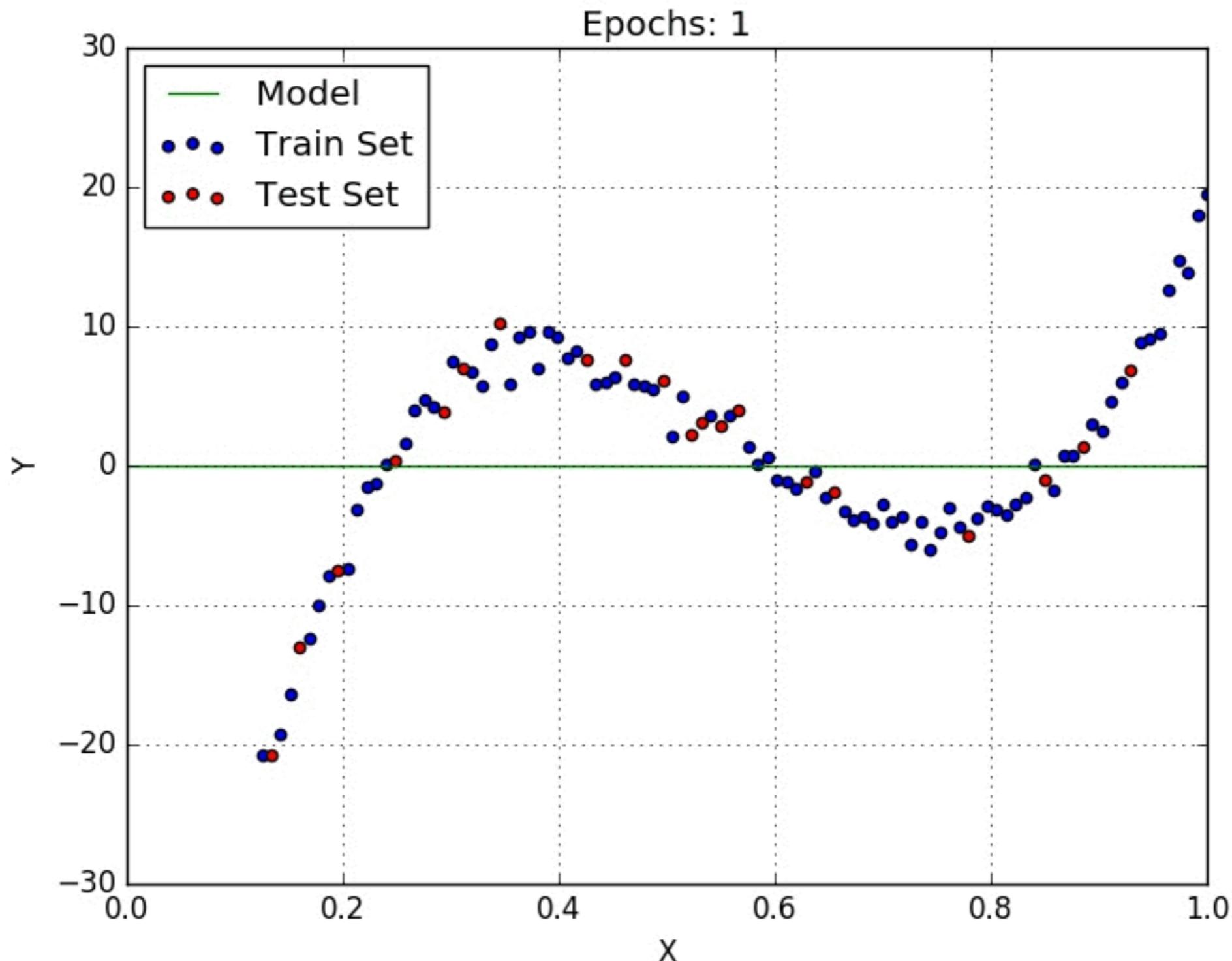
說明：這有多少個？（預測連續數值）

蟋蟀每分鐘唧唧聲，預測溫度多少？

明天的氣溫為何？

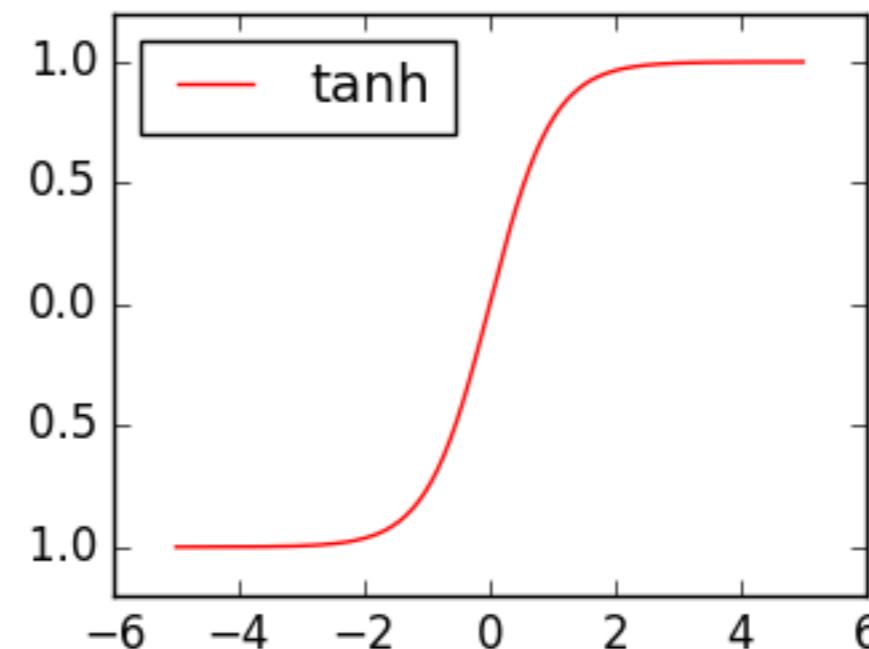
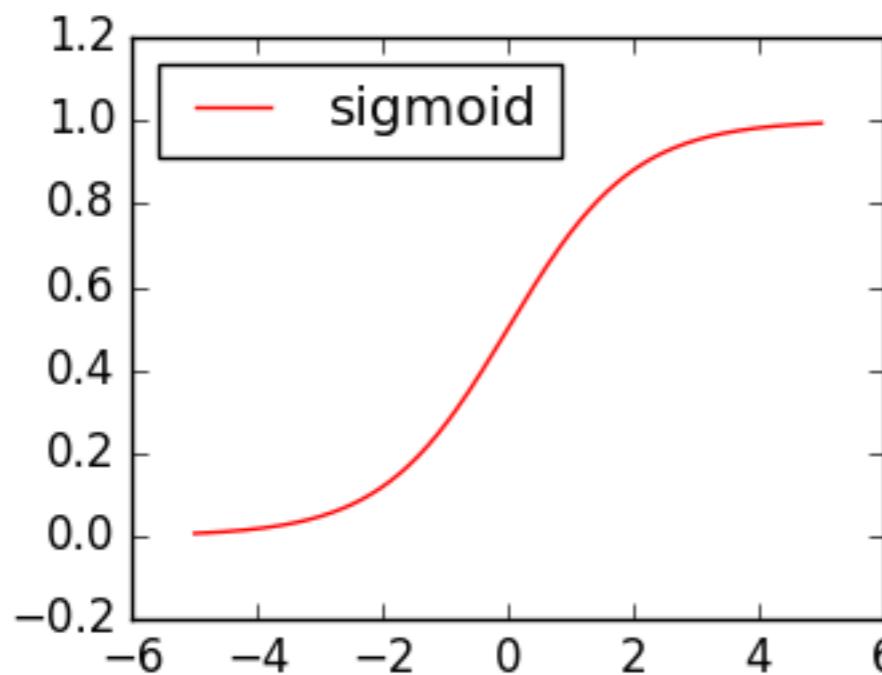
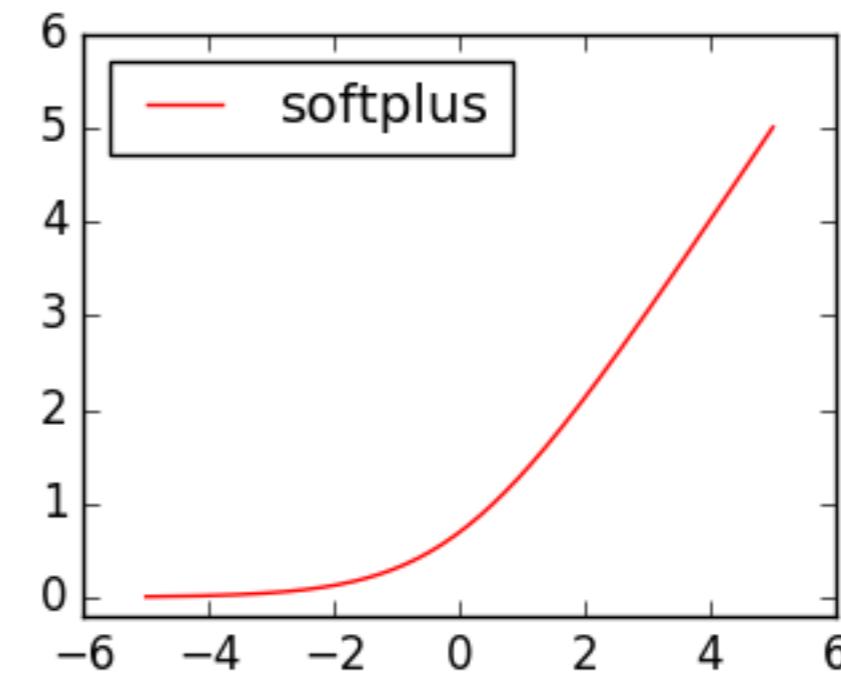
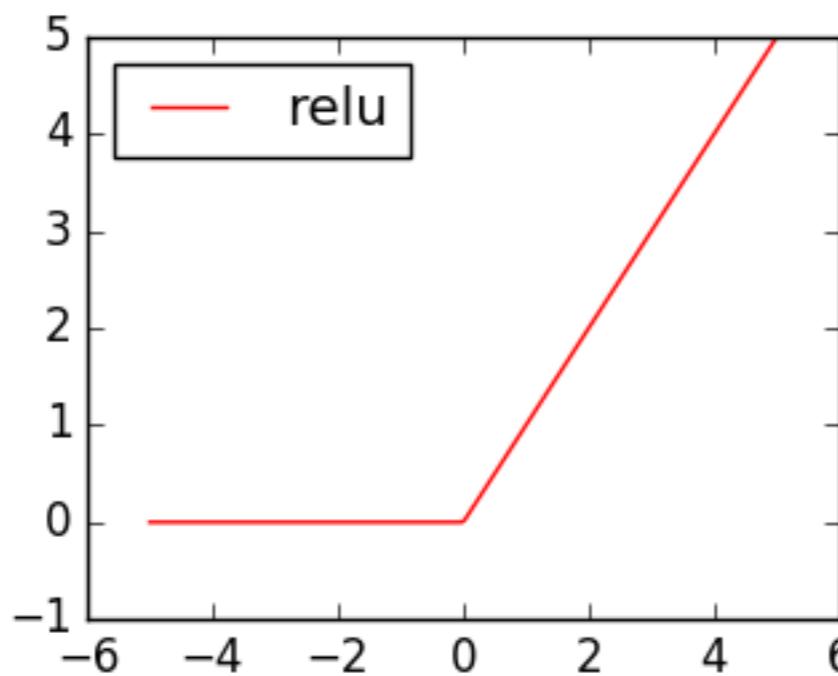
我下週會獲得多少新追蹤者？





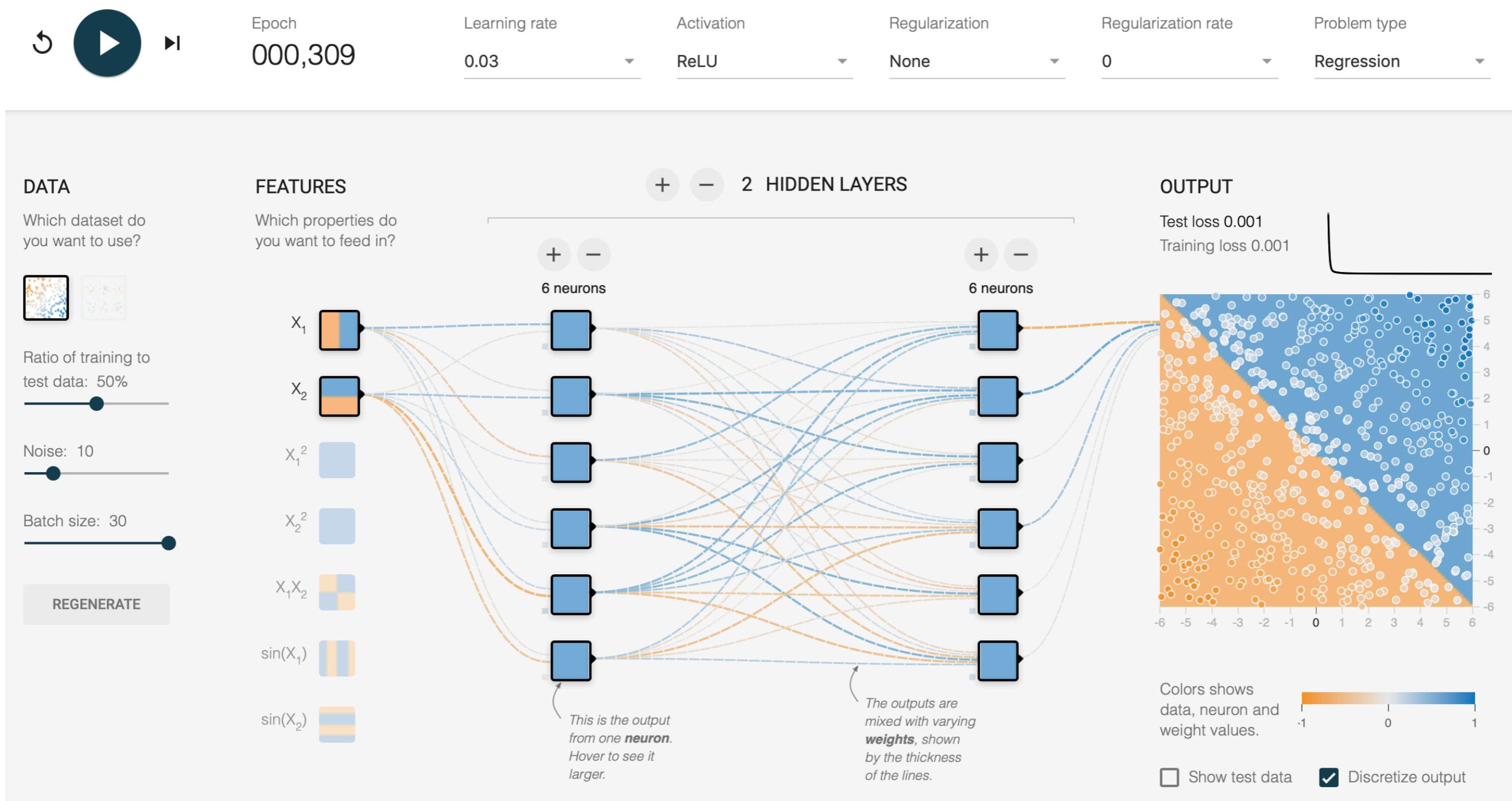
参考自[www.cs.toronto.edu/](http://www.cs.toronto.edu/)  
作者Davi Frossard

# 激勵函數：



# 直觀互動

<https://playground.tensorflow.org/>



# Classification 分類：

說明：這是甲，還是乙？（二元分類）

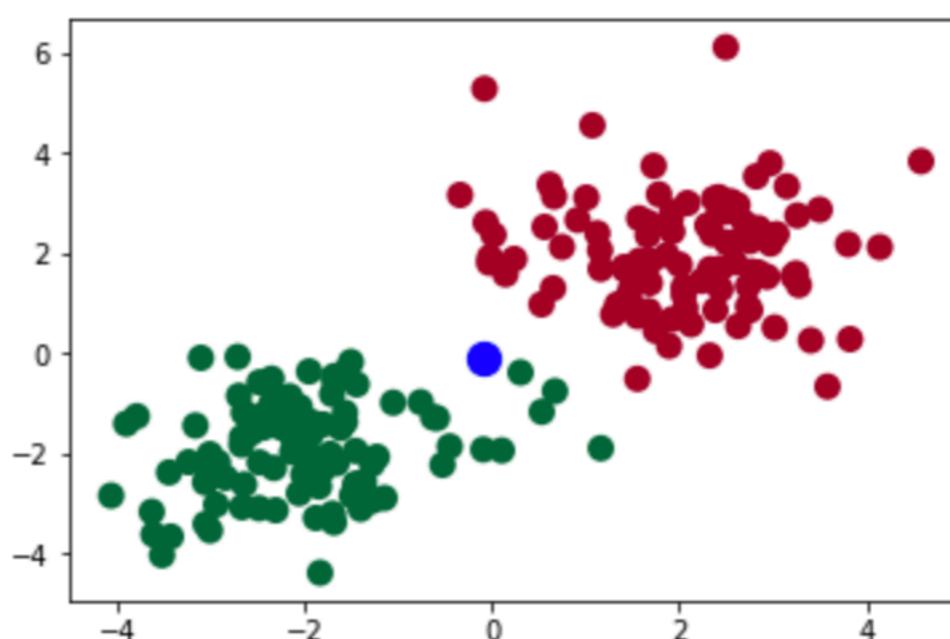
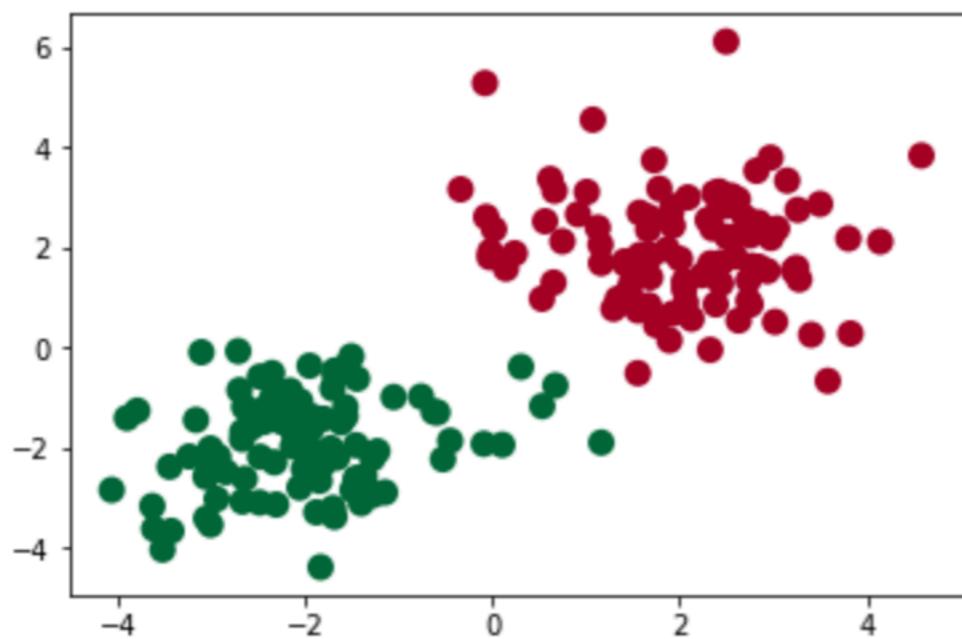
這是一張貓還是狗的圖片？

這位顧客會點還是不會點最上面的連結？

說明：這是甲、乙、丙還是丁？（多元分類）

這是哪種動物的圖片？

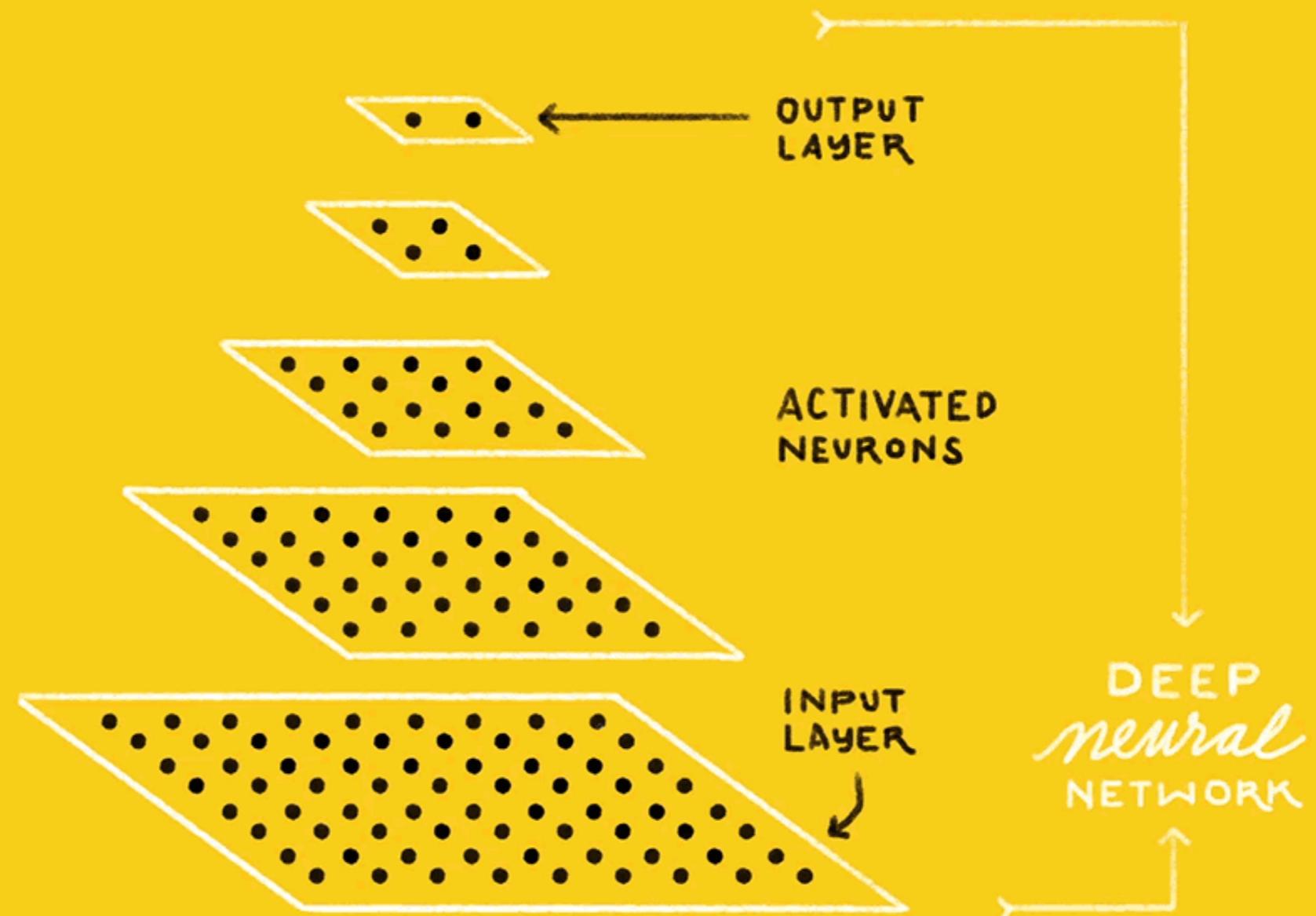
這則錄音裡的講者是誰？



IS THIS A  
**CAT or DOG?**



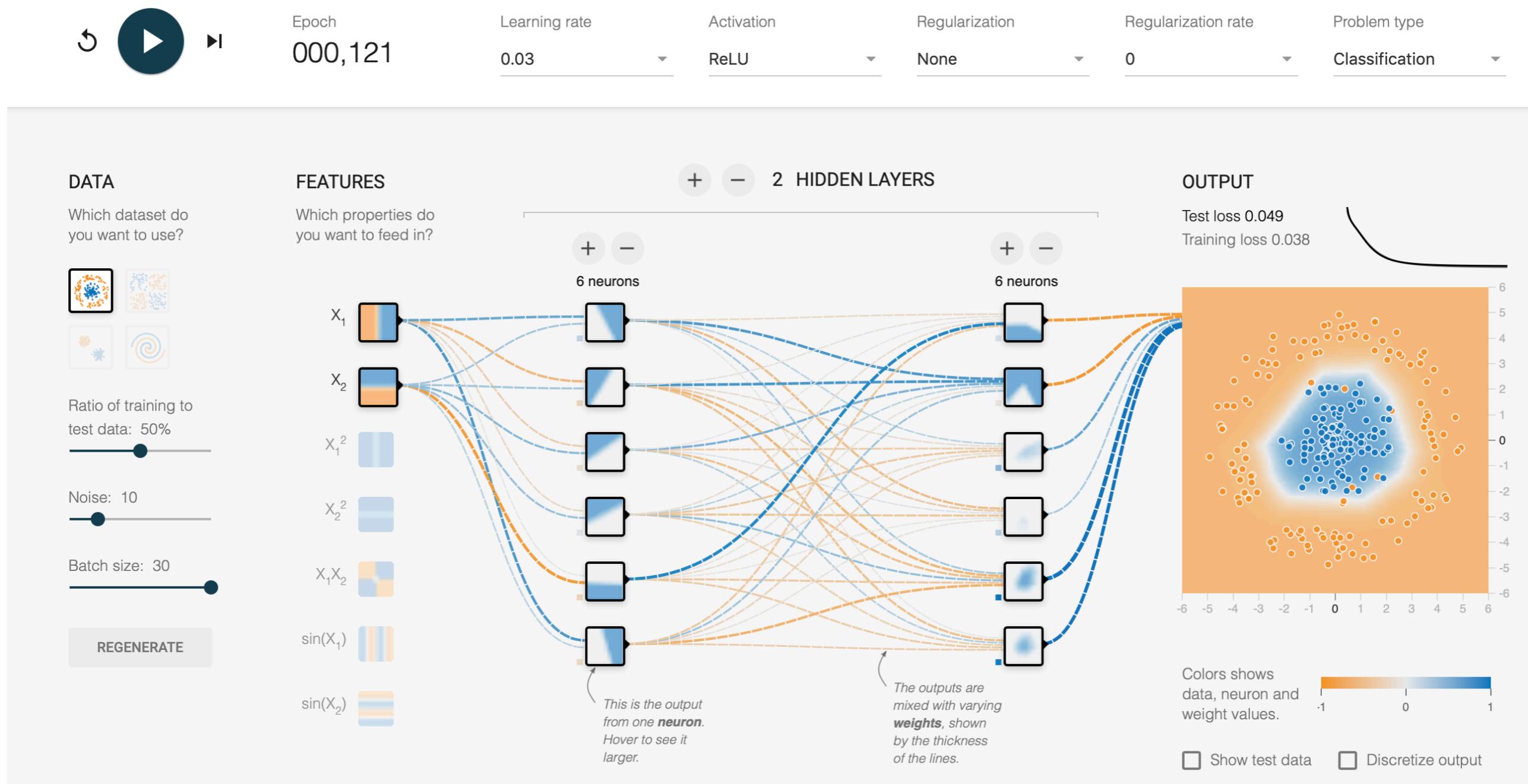
CAT    DOG



參考自[hackernoon.com](https://hackernoon.com)  
作者Sagar Sharma

# 直觀互動

<https://playground.tensorflow.org/>



# Google 圖片標籤辨識（多標籤分類）

<https://cloud.google.com/vision/>

Try the API

Faces Objects Labels Web Properties Safe Search

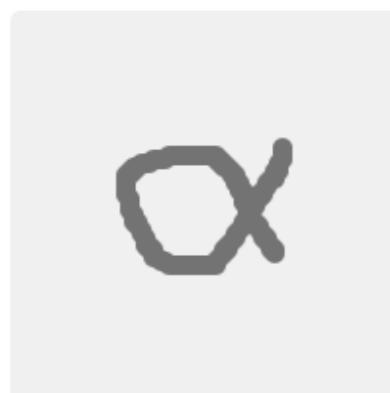
self.jpg

Man	91%
Jacket	64%
Clothing	50%

# 小樣本畫畫分類（多類別分類）

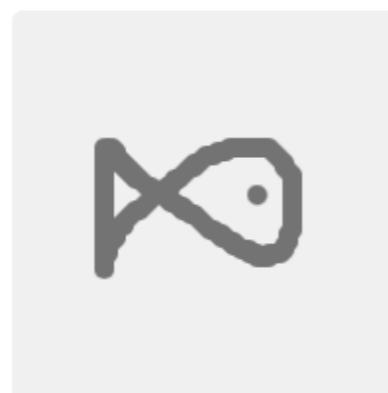
<https://blog.openai.com/reptile>

Training Data



6.4%

ERASE ALL



93.1%



0.5%

Input



ERASE

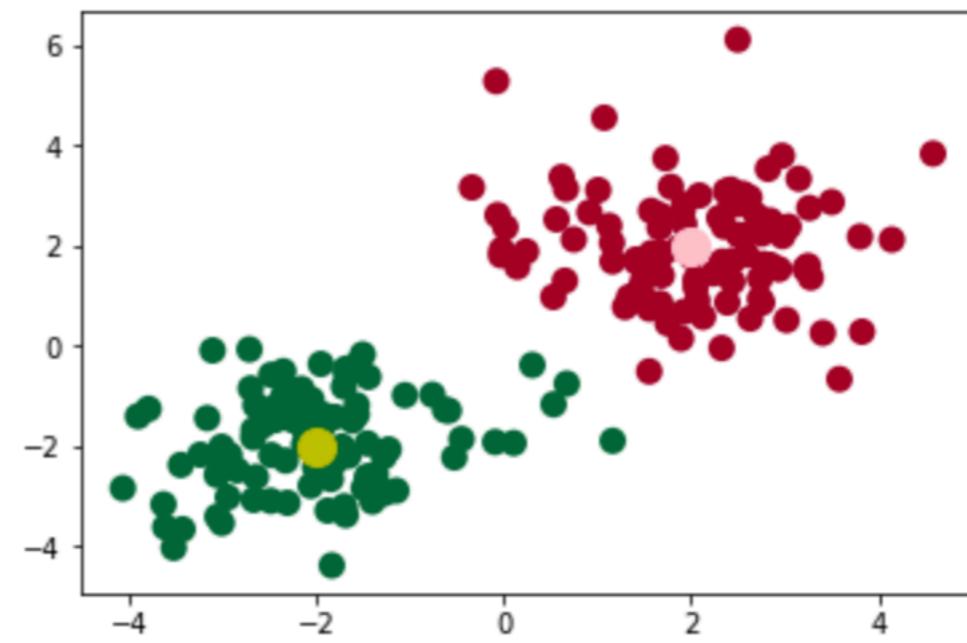
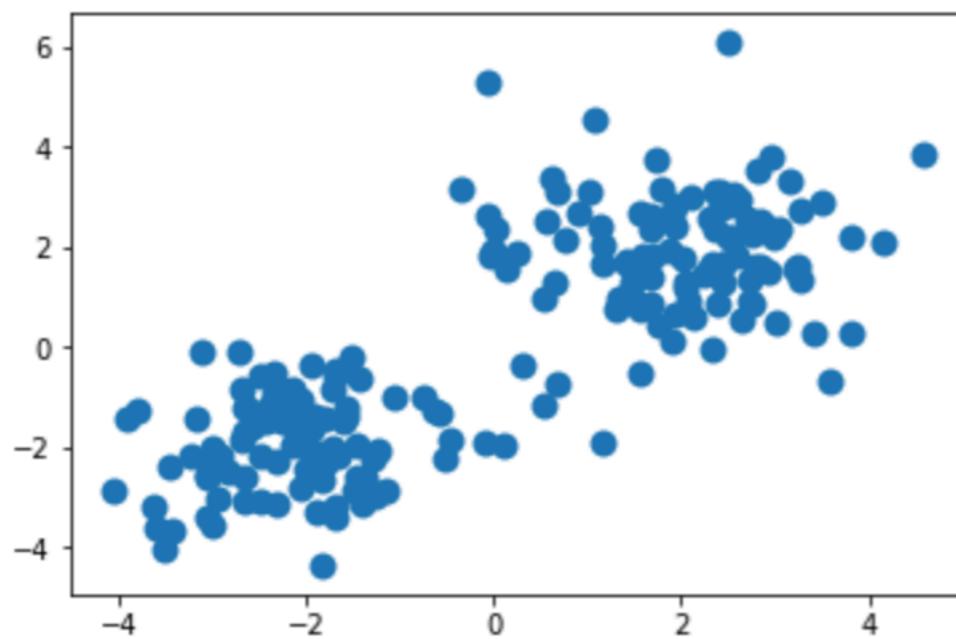
# Clustering 分群 / 聚類 (非監督式學習) :

說明：衡量資料之間的距離或相似度，將相似的樣品分群

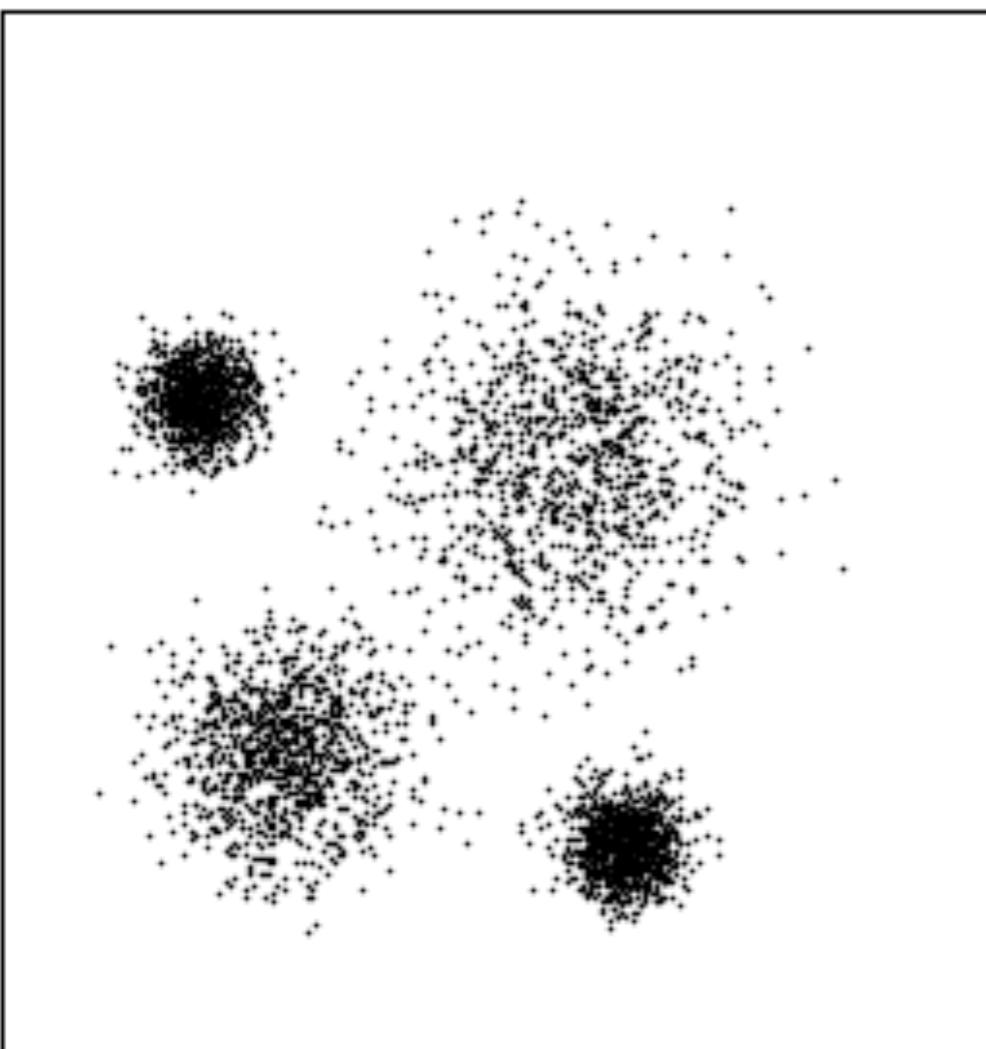
哪些觀眾喜歡同一類電影？

哪些消費者對飲料有相似的品味？

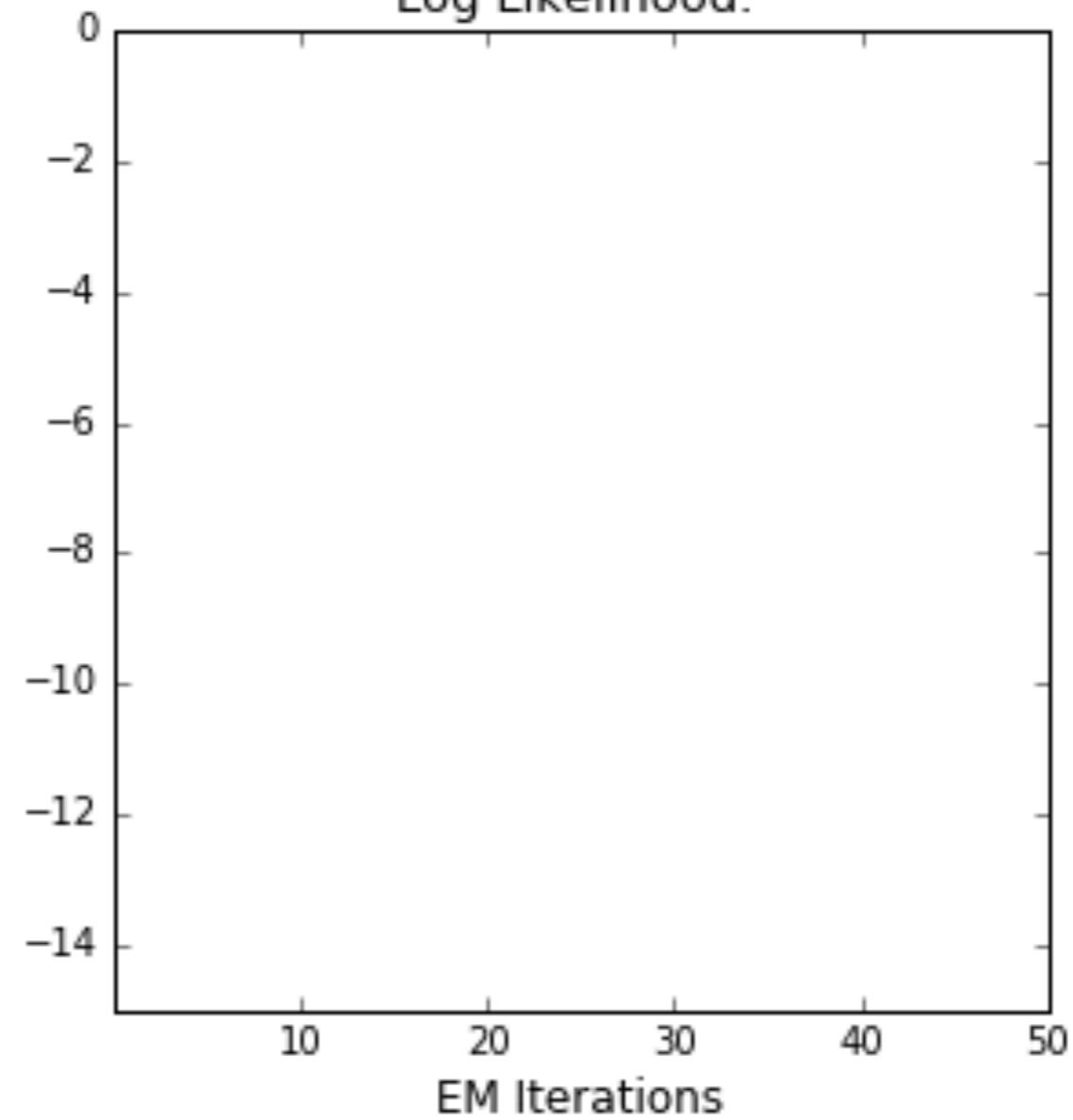
這些文件裡有哪幾組詞彙常常同時出現？



EM Iteration:

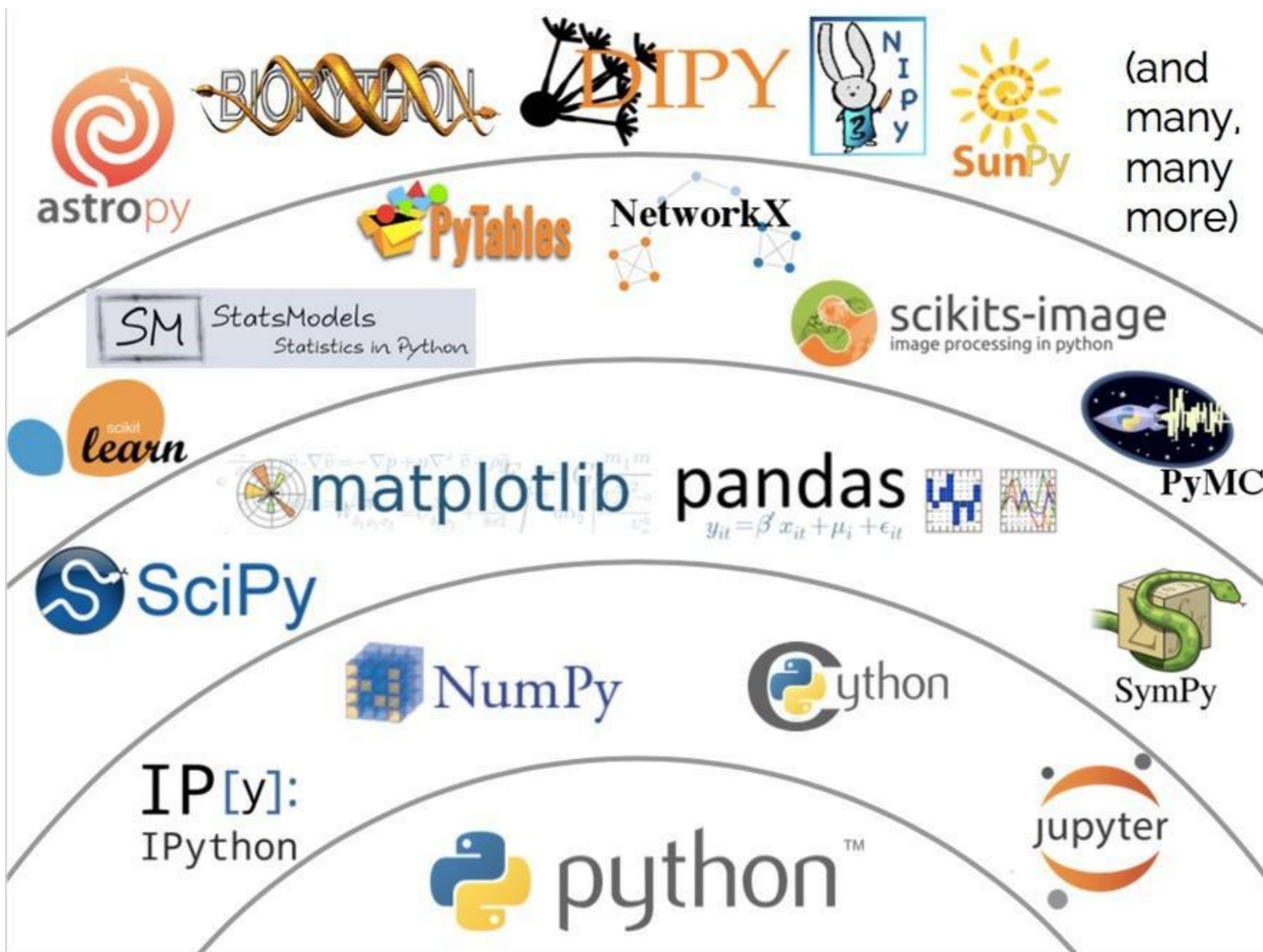


Log Likelihood:

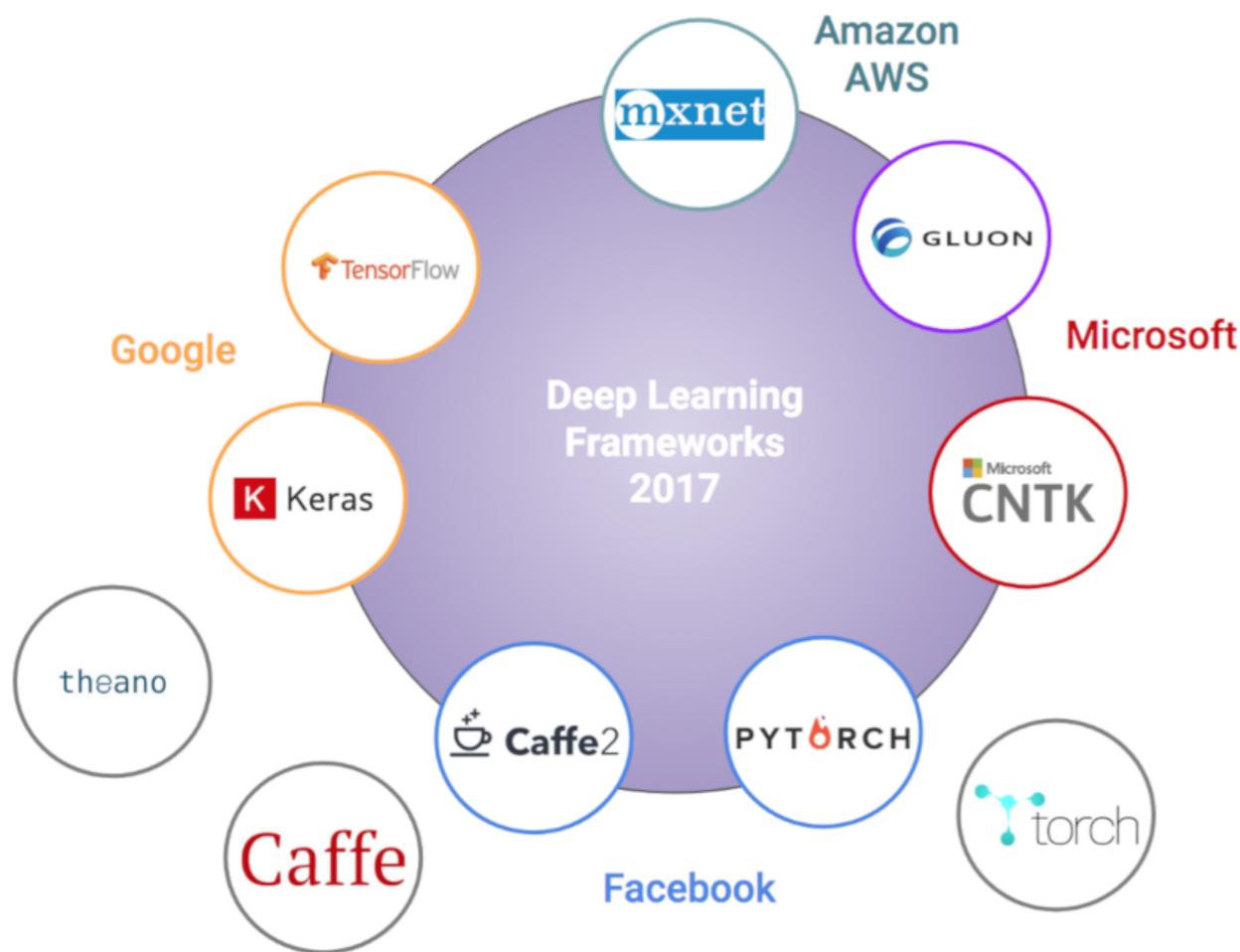


參考自[dashee87.github.io](https://dashee87.github.io)  
作者David Sheehan

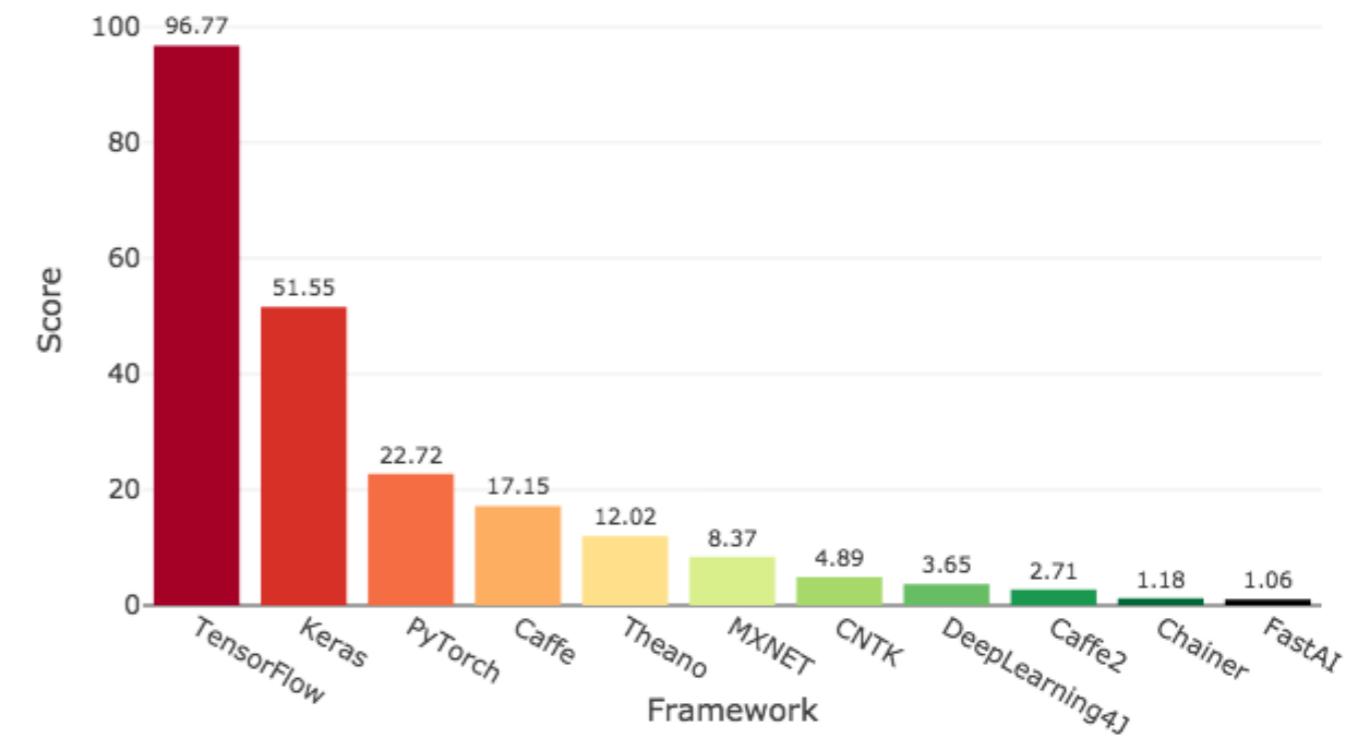
# 數據科學的工具：



# 深度學習的工具：



Deep Learning Framework Power Scores 2018



參考自 [towardsdatascience.com](https://towardsdatascience.com)  
作者 Jeff Hale

# 為什麼使用Keras ?

Keras的後端可以選TensorFlow  
簡單使用  
仍具有一定的彈性



## Sklearn:

```
model = GradientBoostingRegressor()  
model.fit(x_train, y_train)  
y_pred = model.predict(x_test)
```

## Keras:

```
model = Sequential()  
model.add(Dense(input_dim=1, units=10, activation='relu'))  
model.add(Dense(units=20, activation='relu'))  
model.add(Dense(units=10, activation='relu'))  
model.add(Dropout(rate=0.0))  
model.add(Dense(units=1))  
model.compile(loss='mse', optimizer=Adam(lr=0.01))  
model.fit(x_train, y_train, epochs=30, batch_size=64)  
y_pred = model.predict(x_test)
```

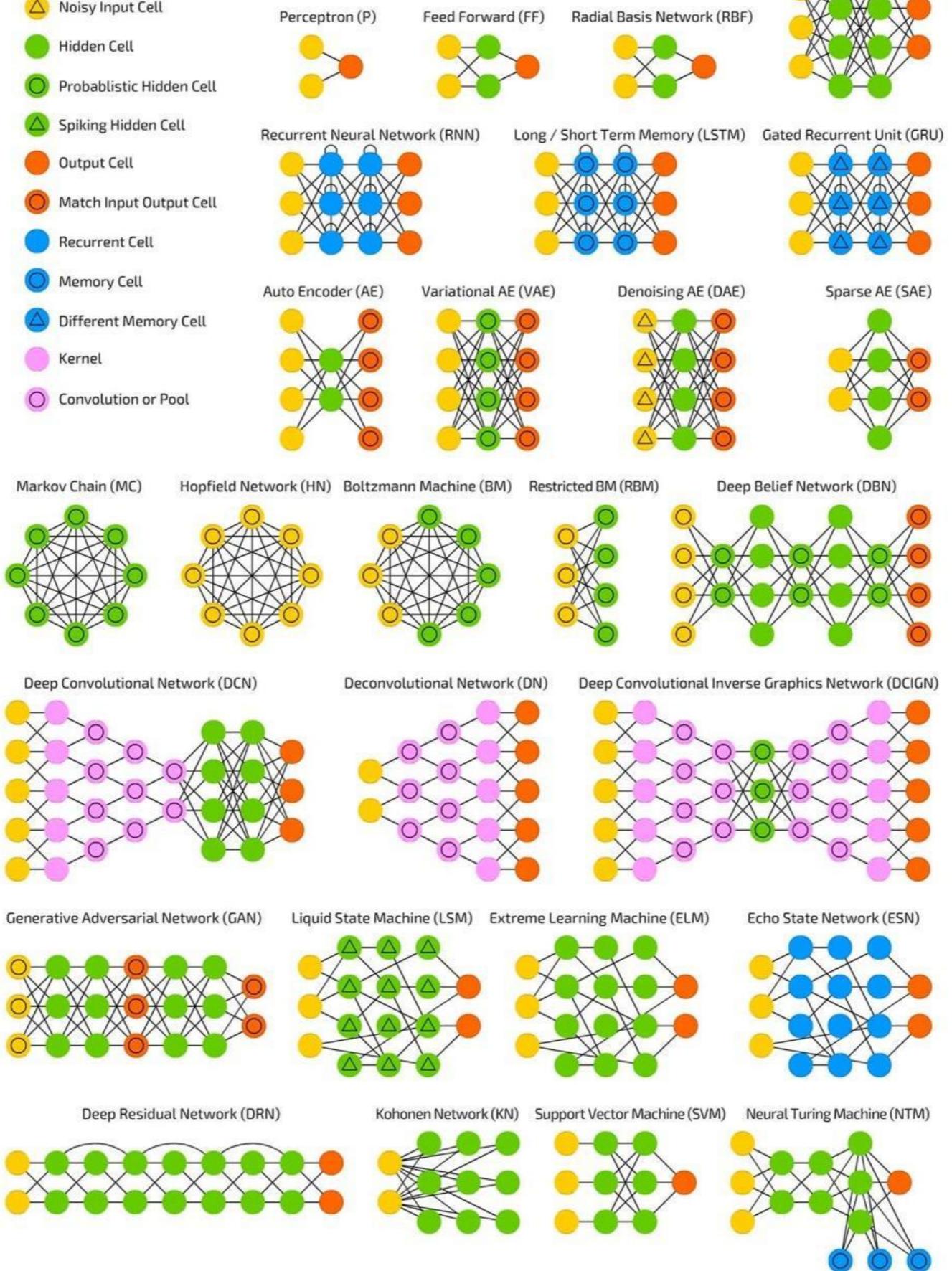
## TensorFlow:

```
x = tf.placeholder(tf.float32, [None, 1])  
y = tf.placeholder(tf.float32, [None, 1])  
layer1 = tf.layers.dense(inputs=x, units=10, activation=tf.nn.relu)  
layer2 = tf.layers.dense(layer1, 20, tf.nn.relu)  
layer3 = tf.layers.dense(layer2, 40, tf.nn.relu)  
predict = tf.layers.dense(layer3, 1)  
loss = tf.losses.mean_squared_error(labels=y, predictions=predict)  
train = tf.train.GradientDescentOptimizer(learning_rate=0.003).minimize(loss)  
sess = tf.Session()  
sess.run(tf.global_variables_initializer())  
for step in range(30000):  
    c_, _ = sess.run([loss, train], feed_dict={x:x_train, y:y_train})  
y_pred = sess.run(predict, feed_dict={x:x_test})
```

A mostly complete chart of  
**Neural Networks**

©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool



# 數據分析流程：

提出問題

收集數據

數據前處理

特徵工程

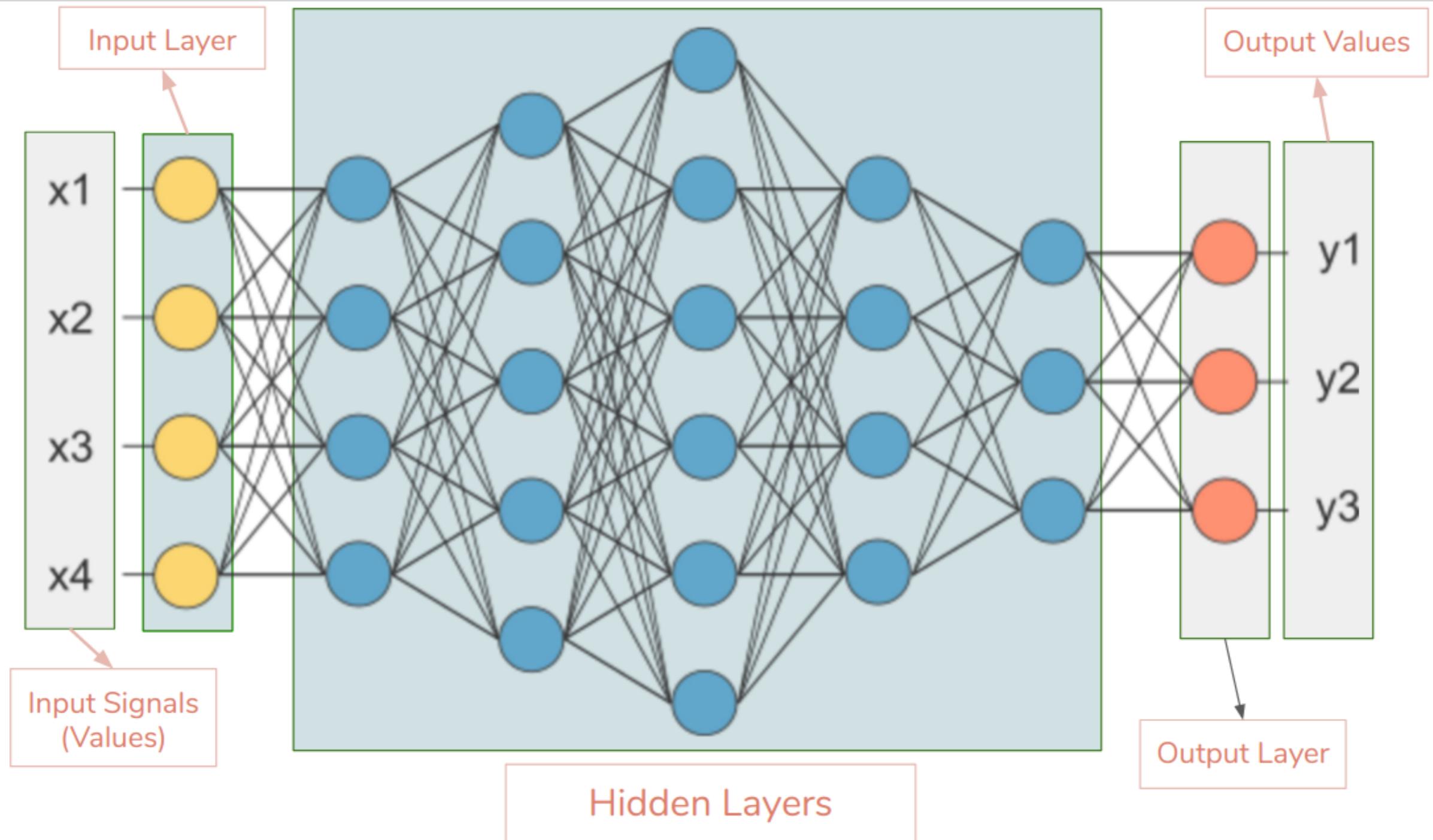
\* 建立模型 \*

\* 調整參數 \*

\* 上線預測 \*

\* 學習新數據 \*

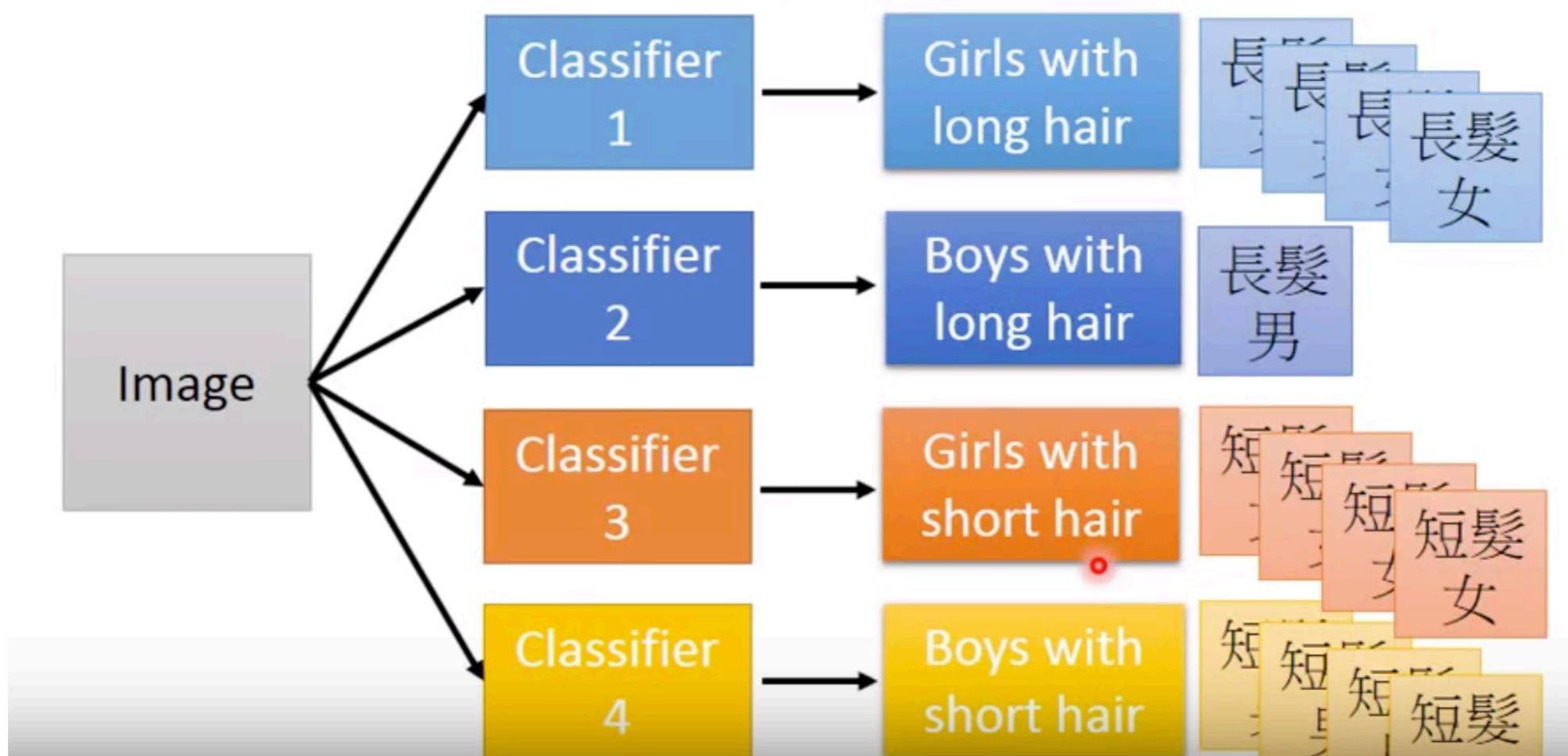
# DNN 深度神經網路：



參考自[hackernoon.com](https://hackernoon.com)  
作者Mate Labs

# 淺的神經網路：

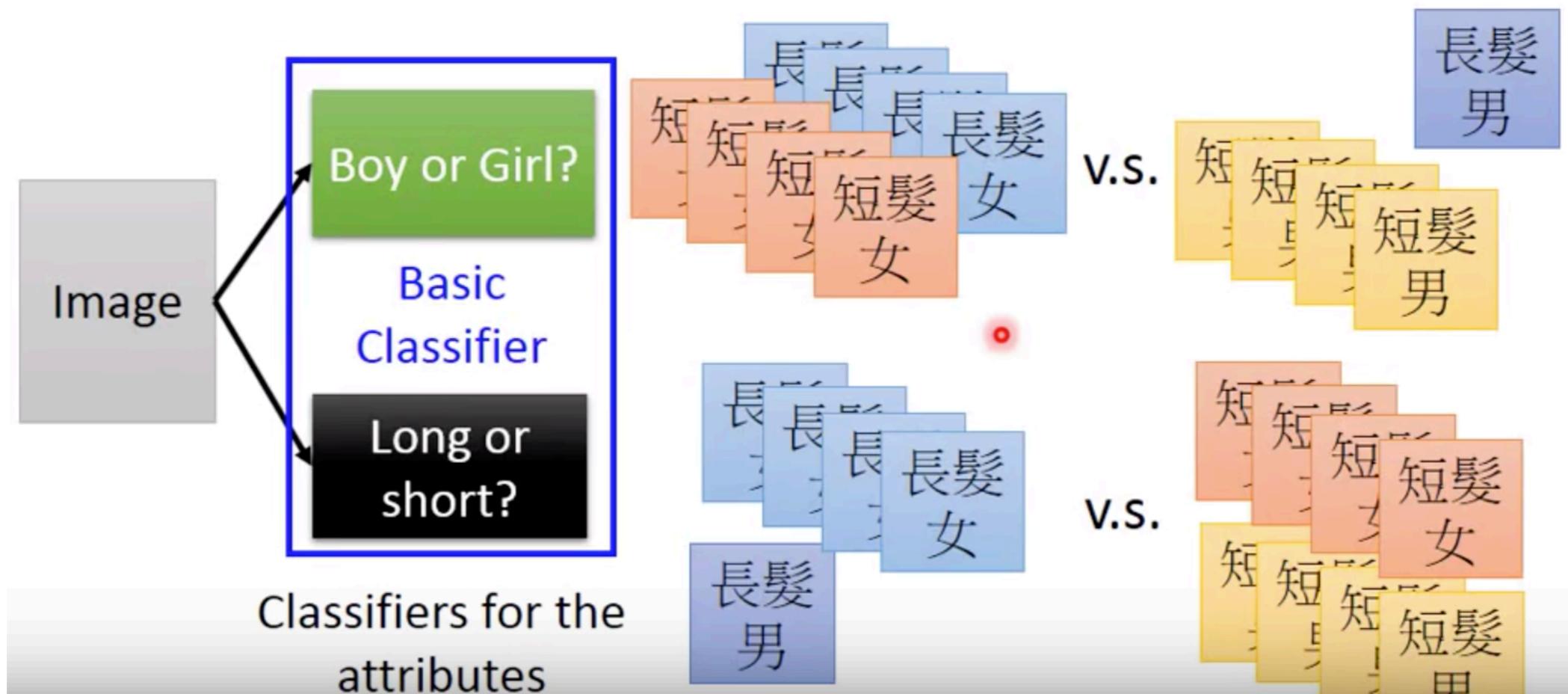
- Deep → Modularization



參考自李宏毅教授

# 深的神經網路：

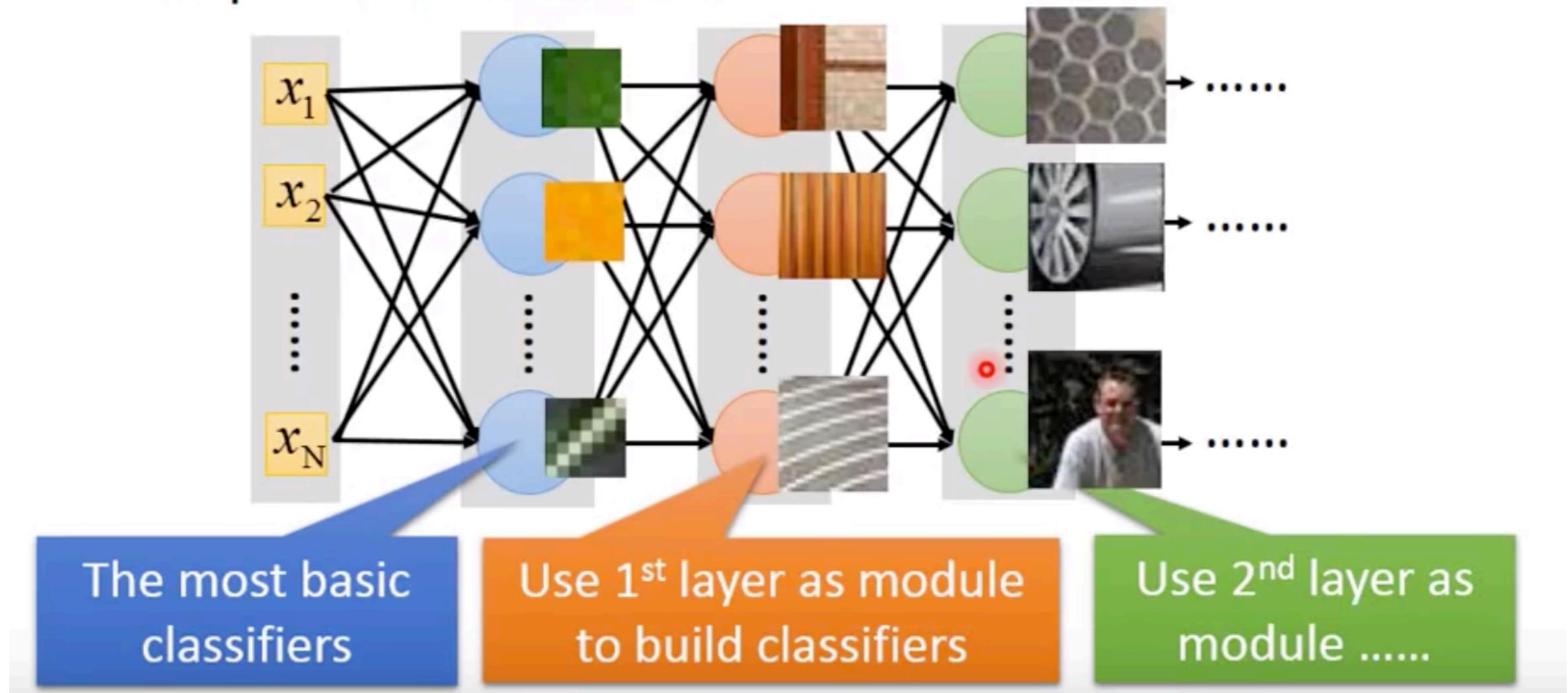
- Deep → Modularization



參考自李宏毅教授

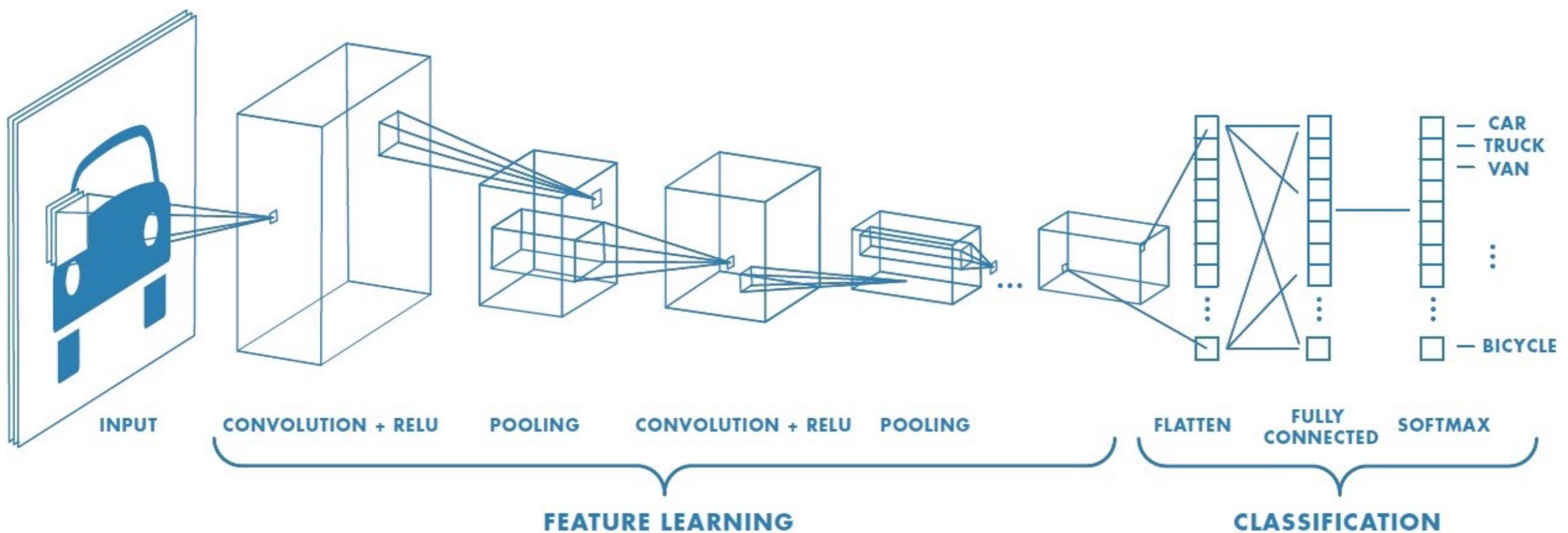
# 選擇深度神經網路：

- Deep → Modularization



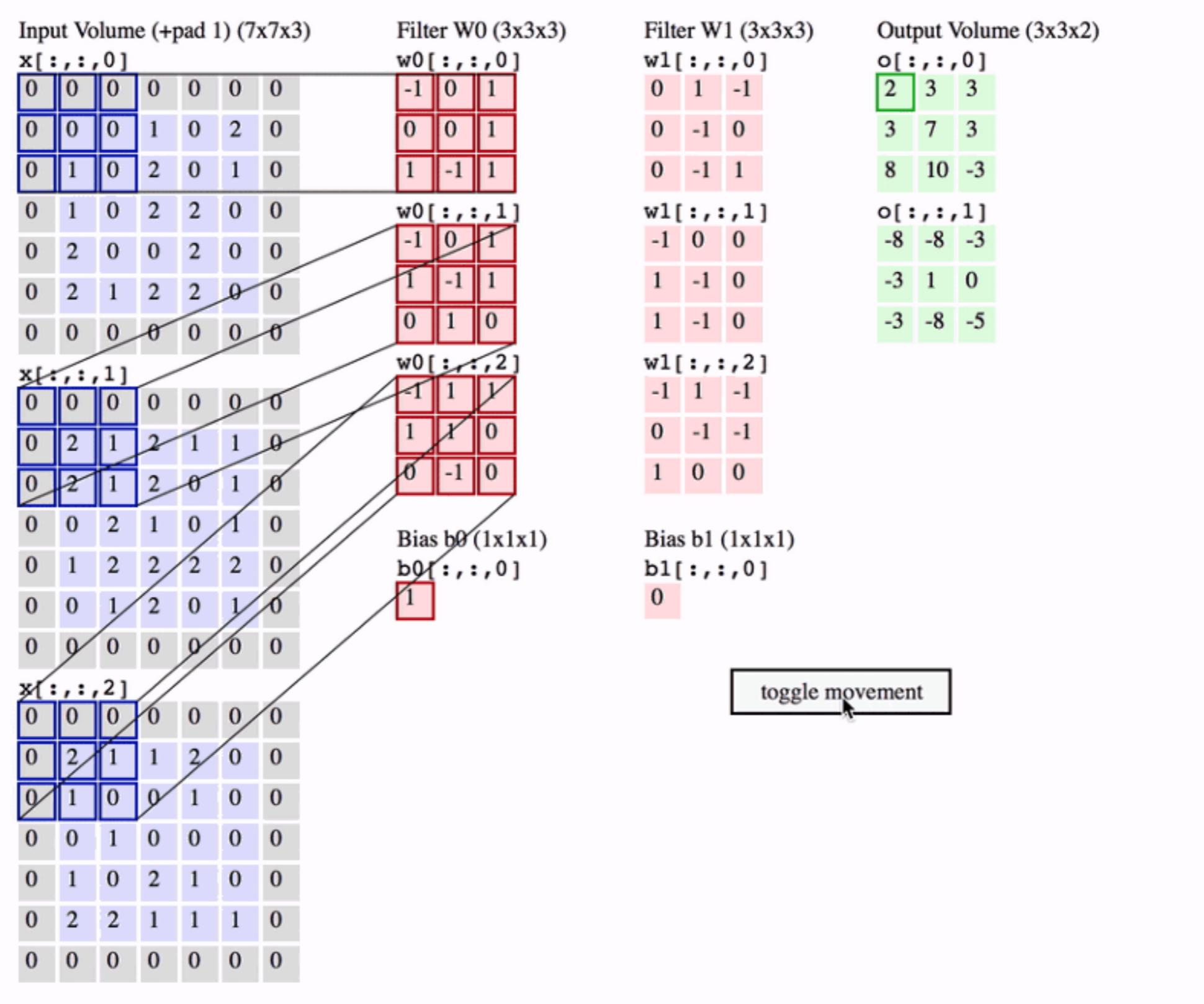
參考自Zeiler, M. D., & Fergus, R. In Computer Vision

# 基本的圖片辨識神經網路架構：



參考自 [towardsdatascience.com](https://towardsdatascience.com)  
作者 Aryan Misra

# CNN 卷積神經網路：



# Max Pooling 池化：

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

Feature map




Pooled  
Feature map

# 下載開發環境

<https://www.anaconda.com>

The screenshot shows the top navigation bar of the Anaconda website. On the left is the Anaconda logo. To its right are five menu items: Products, Why Anaconda?, Solutions, Resources, and Company. A green 'Download' button is positioned on the far right of the bar. A blue rectangular box with a red circle containing the number '1' is overlaid on the 'Download' button area. A magnifying glass icon is located at the far right end of the bar.

**Anaconda and JetBrains  
Join Forces to Launch  
'PyCharm for Anaconda'**

[Learn More](#)



Latest news: *Anaconda named a May 2019 Gartner Peer Insights Customers' Choice for Data Science and Machine Learning Platforms.*

網頁往下滑

A screenshot of a website showing the download options for Anaconda. A large blue arrow points downwards from the top left towards the 'Windows' section. A red circle with the number '1' is positioned above the 'Windows' button. A blue rounded rectangle highlights the 'Windows' and 'macOS' buttons. Below them, the 'Linux' button is shown with its respective icon. The text 'Anaconda 2019.03 for Windows Installer' is displayed prominently. Two sections are visible: 'Python 3.7 version' and 'Python 2.7 version'. The 'Python 3.7 version' section contains a green 'Download' button with a blue border, which has a red circle with the number '2' next to it. Below the button are two download links: '64-Bit Graphical Installer (662 MB)' and '32-Bit Graphical Installer (546 MB)'. The 'Python 2.7 version' section also has a 'Download' button and two download links: '64-Bit Graphical Installer (587 MB)' and '32-Bit Graphical Installer (493 MB)'.

1

Windows | macOS | Linux

Anaconda 2019.03 for Windows Installer

Python 3.7 version

2

Download

64-Bit Graphical Installer (662 MB)

32-Bit Graphical Installer (546 MB)

Python 2.7 version

Download

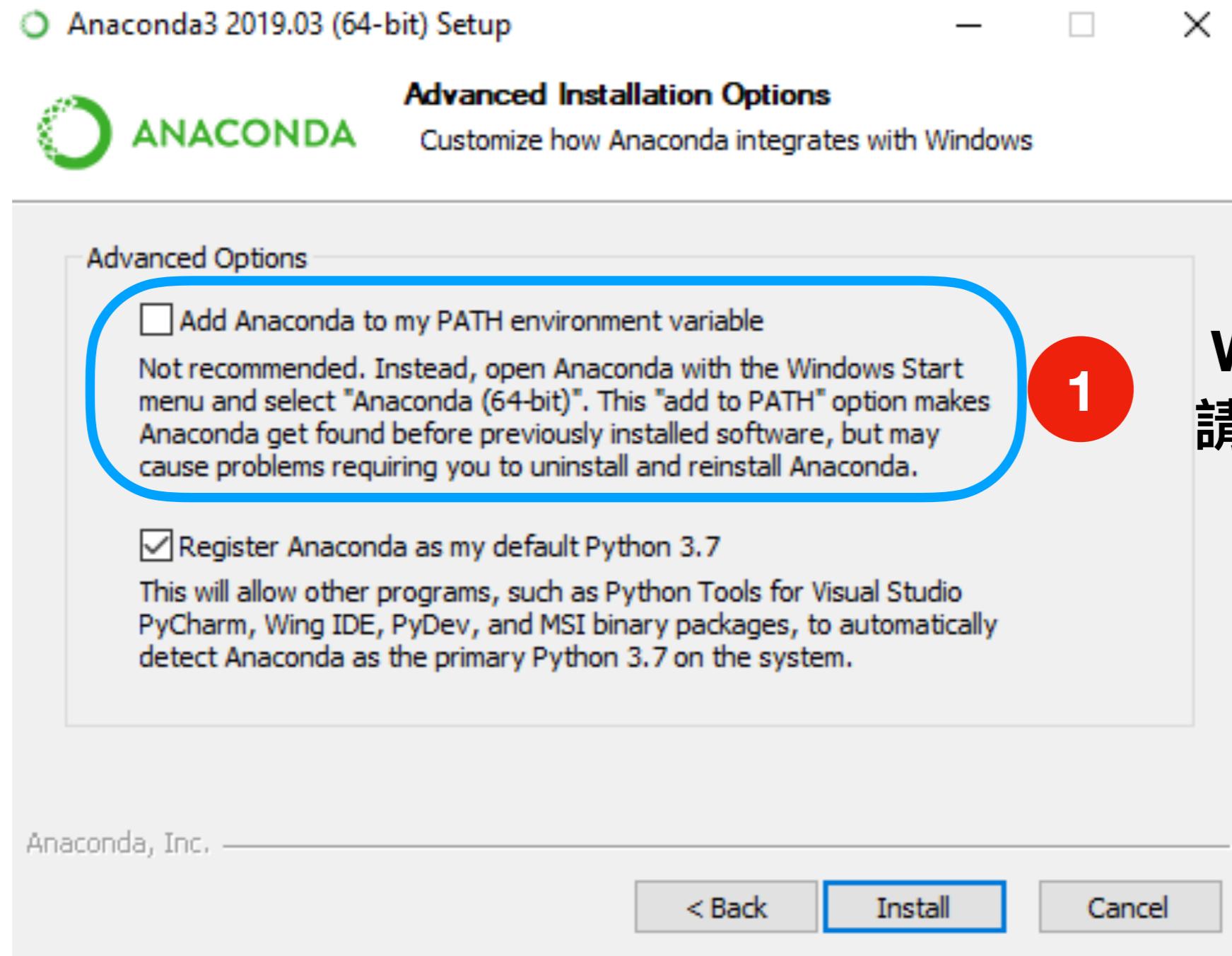
64-Bit Graphical Installer (587 MB)

32-Bit Graphical Installer (493 MB)

# Windows 原則上安裝在C槽，除非空間不夠才選別的硬碟 (D)

## Windows/Mac 都選擇「Install for me only」

(Mac 注意：如果有錯誤提示「You cannot install Anaconda in this location」，請重新選擇「Install for me only」並點下一步)



**Windows  
請都要打勾**

# ANACONDA NAVIGATOR

Home

Environments

Learning

Community

Documentation

Developer Blog



2

Create

Search Environments



Installed

Channels

Update index...

line-bot-sdk



base (root)

python2

spinningup



tensorflow

Name T Description

Create new environment

Name: tensorflow

3

Location: /Users/ianfan/anaconda3/envs/tensorflow

Packages:  Python

3.6

4

R

R

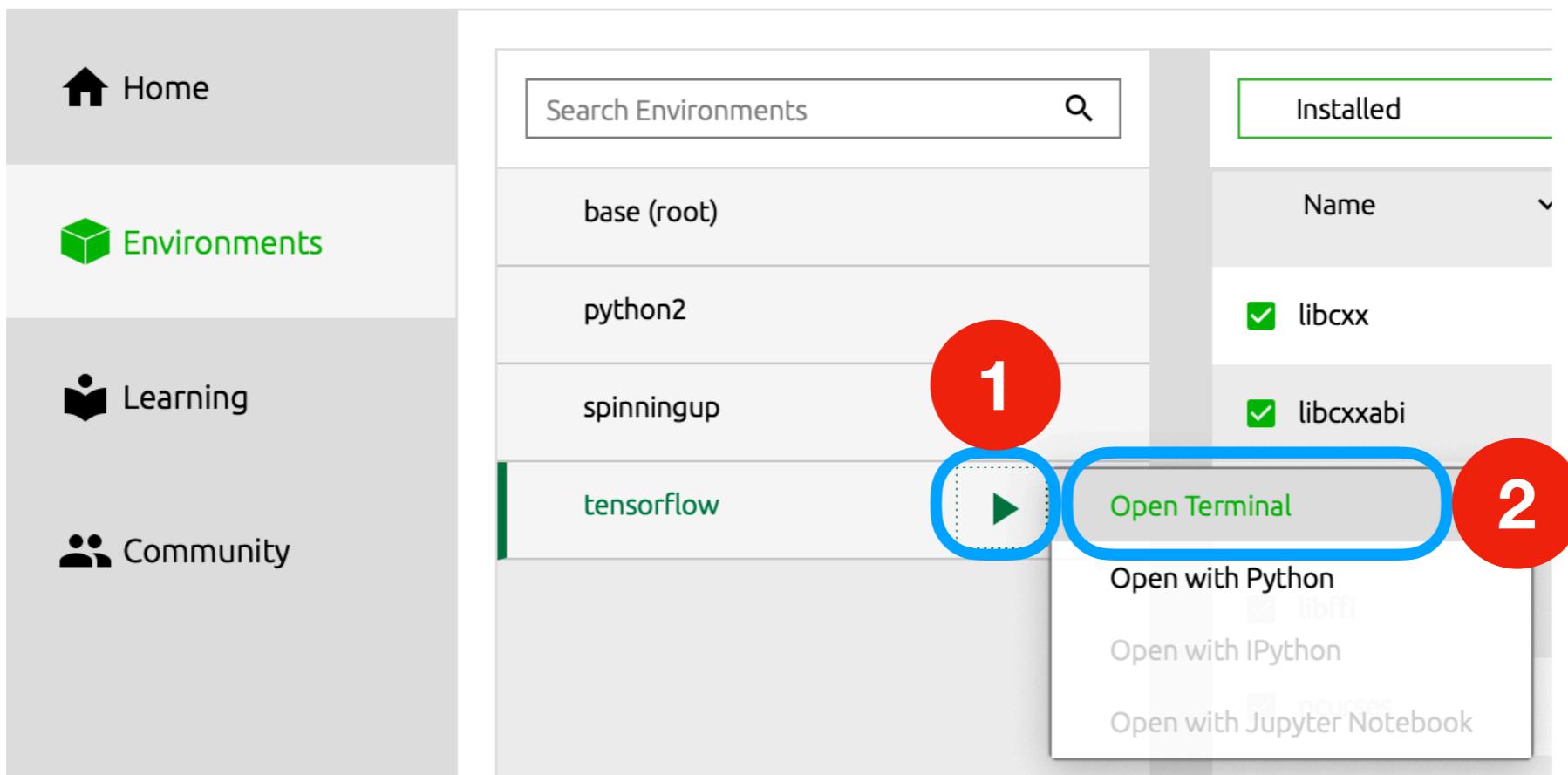
Cancel

Create

5

0 packages available matching "line-bot-sdk"

# ANACONDA NAVIGATOR



ianfan@Ian-Fan-MacBook-Pro-2018 ~ ? source activate tensorflow  
(tensorflow) ianfan@Ian-Fan-MacBook-Pro-2018 ~ ? pip install ipython

有出現(環境名稱)才是正確的

(tensorflow) ianfan@Ian-Fan-MacBook-Pro-2018 ~ ? activate tensorflow

3 Mac

3 Windows

# Mac 安裝方法

<https://www.anaconda.com/download/#macos>

下載「64-Bit Graphical Installer」

選擇「Install for me only」

(注意：如果有錯誤提示「You cannot install Anaconda in this location」，請重新選擇「Install for me only」並點下一步)

最好把bash\_profile跟shrc設置好環境路徑，以後就可以在任意目錄下執行Anaconda的指令

vim ~/.bash\_profile

複製貼上以下三行

```
#added by Anaconda3 5.2.0 installer
export PATH=~/anaconda3/bin:$PATH
export PATH="/anaconda3/bin:$PATH"
```

保存並退出（打:wq）

vim ~/.zshrc

複製貼上以下三行

```
#added by Anaconda3 5.2.0 installer
export PATH=~/anaconda3/bin:$PATH
export PATH="/anaconda3/bin:$PATH"
```

保存並退出（打:wq），再source ~/.zshrc

先把conda更新

conda upgrade conda

conda upgrade --all

安裝 Jupyter

conda install jupyter notebook

產生一個環境，名叫「tensorflow」，python版本3.5以上，建議3.6（會創建在你的anaconda3/envs/，新增移除都方便）

conda create -n tensorflow python=3.6

記得！！每次進terminal終端機，第一件事先進這個叫「tensorflow」的環境（有進入前面會有tensorflow的字串），再安裝工具或執行Jupyter  
source activate tensorflow

注意：若有錯誤提示信息pip版本太低，才要更新pip版本，不然不用upgrade

pip3 install --upgrade pip

## 更新 TensorFlow

```
pip install --ignore-installed --upgrade tensorflow
```

安裝各種工具，不是 pip install，就是 conda install

```
pip install ipython
```

```
pip install line-bot-sdk
```

```
pip install flask
```

```
pip install sklearn
```

```
pip install dataframe
```

```
pip install Pillow
```

```
conda install numpy
```

```
conda install matplotlib
```

```
conda install pandas
```

```
conda install ipython notebook
```

```
conda install Keras
```

```
Proceed ([y]/n)? y|
```

 Home Environments Learning Community

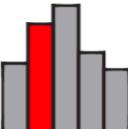
Documentation

Developer Blog

Applications on

tensorflow

Channels



Glueviz  
0.13.3

Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Install](#)



JupyterLab  
0.35.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Install](#)



Jupyter Notebook  
5.7.8

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[1](#) [Install](#) [Launch](#) [2](#) [Install](#)



Orange 3  
3.19.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)



Qt Console  
4.3.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Install](#)



RStudio  
1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

[Install](#)



Spyder  
3.3.4

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Install](#)



VS Code  
1.35.1

Streamlined code editor with support for development operations like debugging, task running and version control.

[Install](#)

記著這個根目錄，之後相關檔案放在方便讀取的地方

The screenshot shows the Jupyter Notebook interface with the following details:

- Header:** "jupyter" logo, "Quit", and "Logout" buttons.
- Navigation:** "Files" (selected), "Running", and "Clusters" tabs.
- Toolbar:** "Select items to perform actions on them.", "Upload", "New", and a refresh icon.
- File Browser:** A table listing directory contents:

	Name	Last Modified	File size
<input type="checkbox"/>	anaconda3	an hour ago	
<input type="checkbox"/>	Applications	2 months ago	
<input type="checkbox"/>	bin	a month ago	
<input type="checkbox"/>	Creative Cloud Files	3 months ago	
<input type="checkbox"/>	Desktop	4 minutes ago	
<input type="checkbox"/>	Documents	7 days ago	
<input type="checkbox"/>	Downloads	15 hours ago	

# Google Drive

## <https://reurl.cc/MxGym>



My Drive > AI > line\_heorku\_lesson ▾

⋮

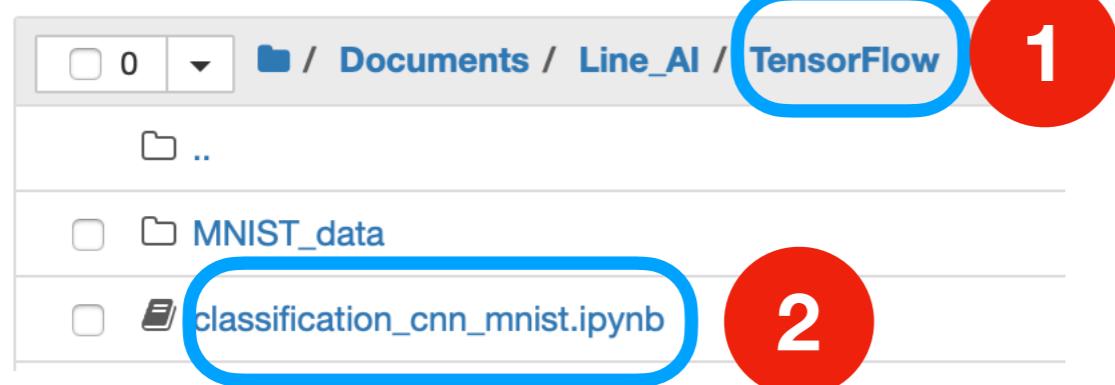
Name	Owner	Last modified	File size
1 data	me	5:38 PM me	—
line_heorku	me	5:22 PM me	—

下載TensorFlow資料夾，放到Jupyter Notebook方便讀取的目錄下



Files    Running    Clusters

Select items to perform actions on them.



從Google Drive下載的檔案夾，  
放在Jupyter好找的地方

The screenshot shows the Jupyter notebook interface. At the top, the title bar reads 'jupyter classification\_cnn\_mnist Last Checkpoint: 02/11/2019 (autosaved)'. The toolbar includes icons for file operations like new, open, save, and run, along with a 'Code' dropdown and a 'Notify' button set to 'Disabled'. A red circle with the number '3' is placed over the 'Run' button in the toolbar.

In the main notebook area, a section titled '載入工具' (Load Tools) is visible. In cell In [1], the following Python code is shown:

```
import os
import pandas as pd
from pandas import DataFrame
from keras.models import Sequential, Model
from keras.layers import Dense, Activation, Convolution2D, MaxPooling2D, Flatten
from keras.optimizers import Adam
from keras.layers import Dropout
from keras.layers import BatchNormalization
from keras.utils import np_utils
import matplotlib.pyplot as plt
#matplotlib.use("TkAgg")
%matplotlib inline
import numpy as np
np.random.seed(1)
from sklearn.metrics import confusion_matrix
from PIL import Image
import random
import math
```

A red circle with the number '3' is also placed over the text '跑完都出現結果，就代表安裝成功' (After running, if results appear, it means the installation was successful), which is displayed prominently in the center of the notebook area.

Using TensorFlow backend.