影像辨識”駕”厲害

臺南市和順國中-林信廷

時間規劃

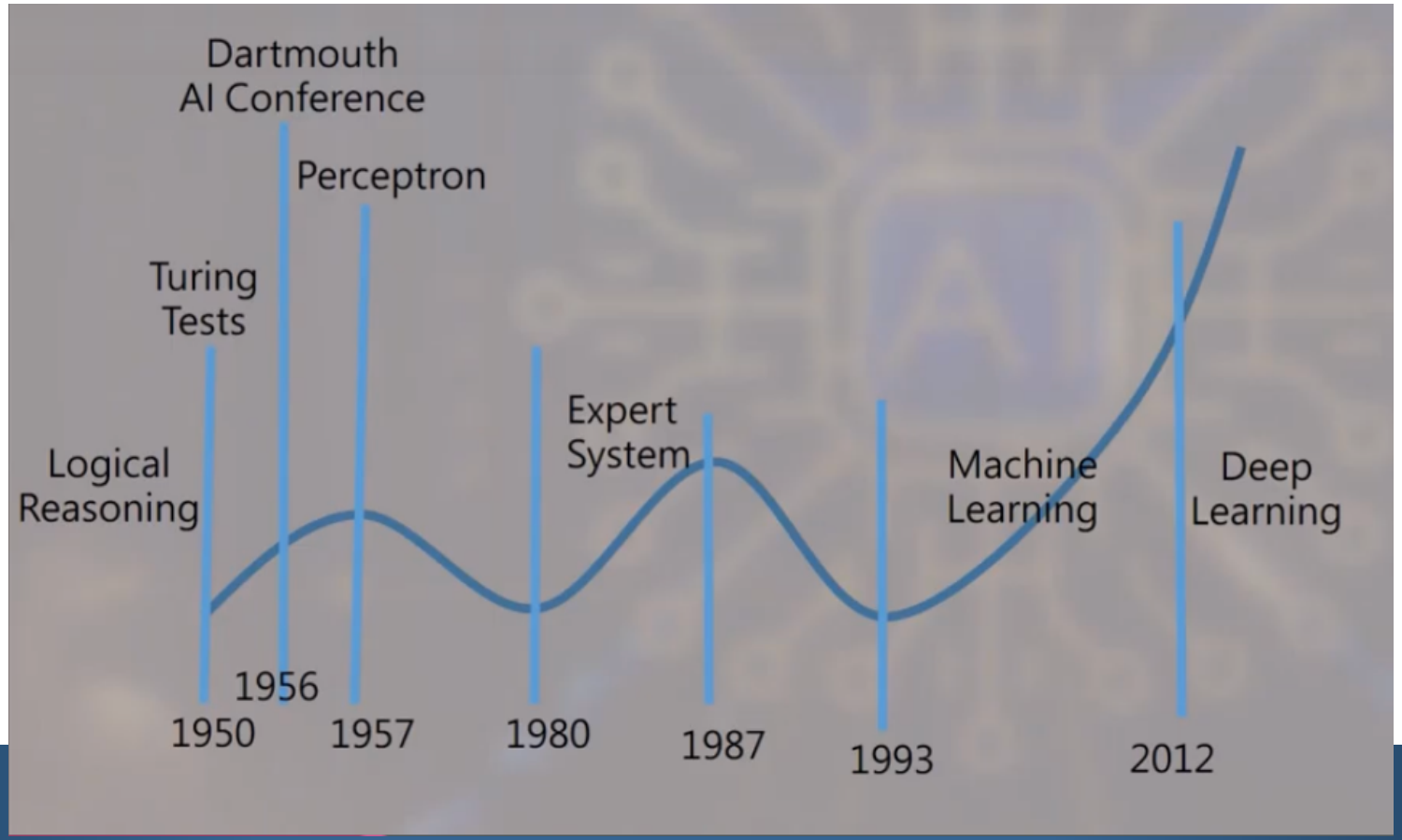
- 人工智慧基本知識：1HR
- 模擬器安裝與體驗：2HR
- 實體自駕車構造說明：1HR
- 實體自駕車體驗：2HR

一、人工智慧發展歷程

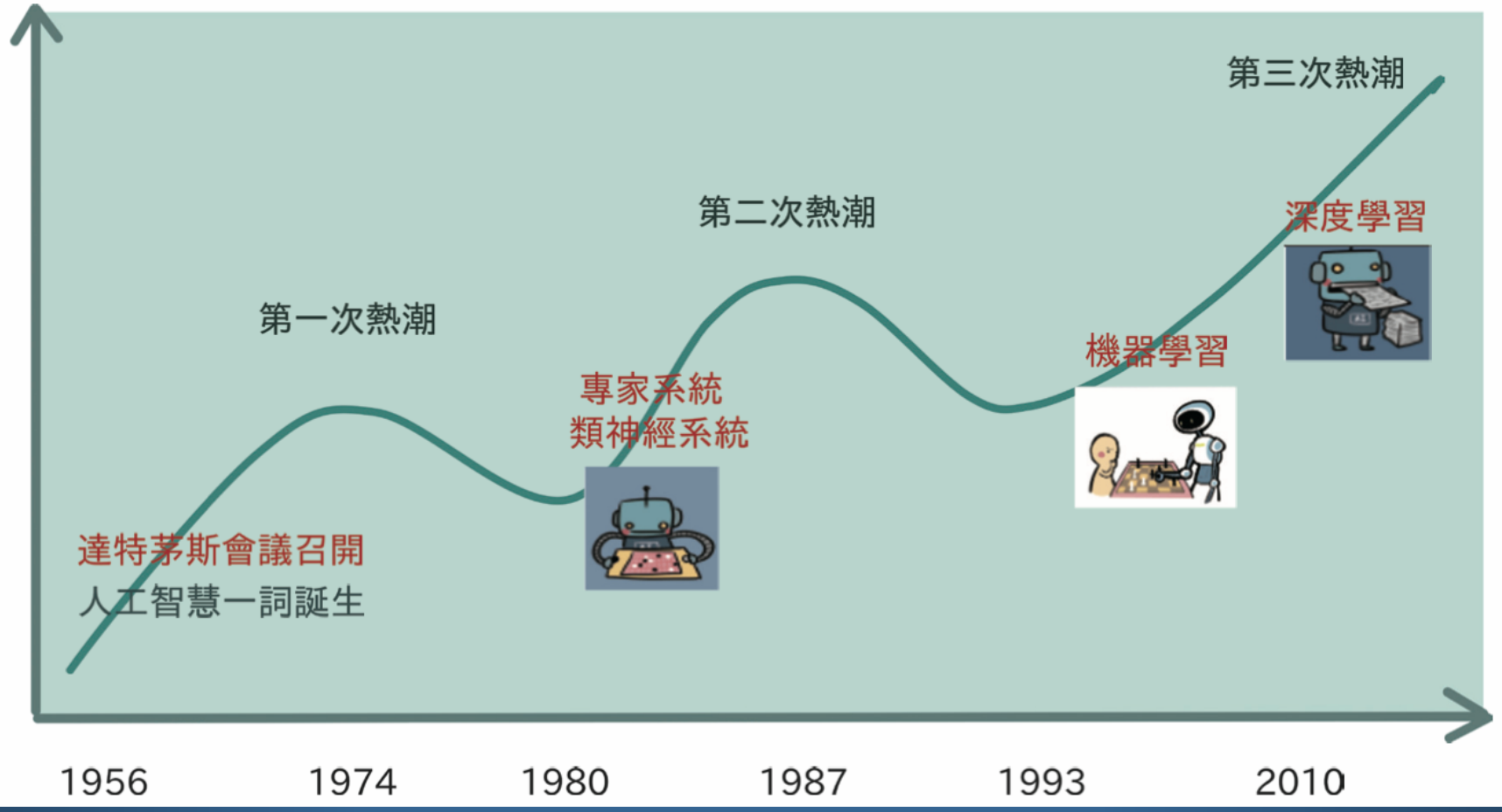
一、人工智慧發展歷程

- 什麼是人工智慧？或者說，什麼是「智慧」？
  - 會思考？→那什麼叫做「會思考」？
  - 你每天都在思考什麼？→要吃什麼？要去哪裡玩？朋友為什麼不理我？
- 看來，智慧就是根據過去的經驗，幫我們做出判斷
- 所以簡單來說，人工智慧就是電腦根據我們提供的資料集，來做判斷

一、人工智慧發展歷程

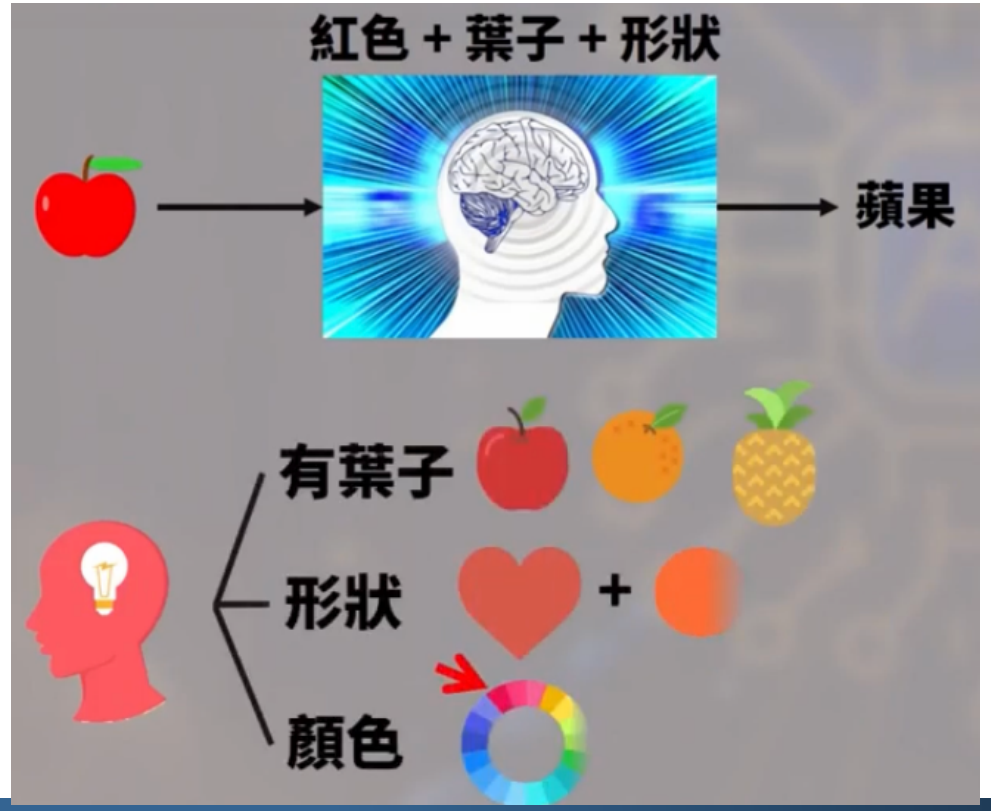

Dartmouth
AI Conference
Perceptron
Turing
Tests
Logical
Reasoning
Expert
System
Machine
Learning
Deep
Learning
1950
1956
1957
1980
1987
1993
2012

一、人工智慧發展歷程


第三次熱潮
第二次熱潮
第一次熱潮
深度學習
機器學習
專家系統
類神經系統
達特茅斯會議召開
人工智慧一詞誕生
1956
1974
1980
1987
1993
2010

一、人工智慧發展歷程

- **LOGICAL REASONING:**
  - 將過去的經驗，用邏輯的推演，判斷看到的物體
  - 將 X 值丟入電腦，用特徵判斷是什麼水果
  - 而僅依照一定的邏輯似乎不夠有智慧

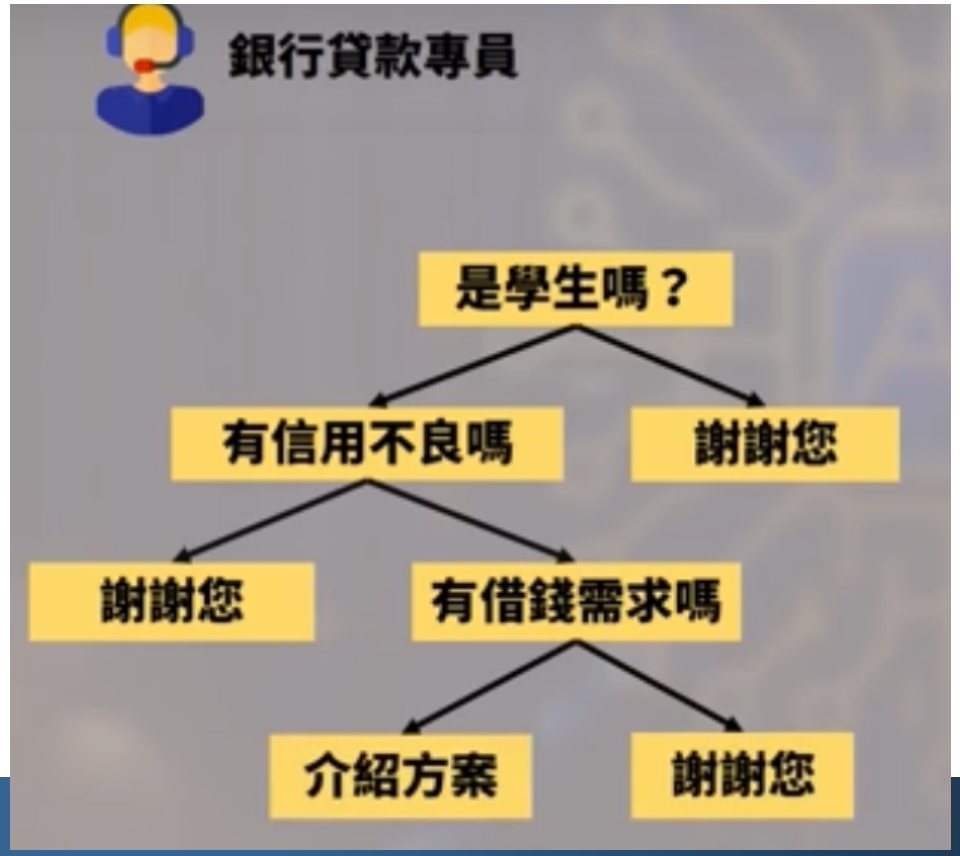


一、人工智慧發展歷程

- **DARTMOUTH AI CONFERENCE:**
  - 會議持續一個月，採討論的方式，最終提出AI的定義
  - 在此之前是用是否通過圖靈測試來定義
- **PERCEPTRON（感知器）：**
  - 想教電腦對事情做出反應模仿人類智慧，衍伸出感知器的技術，是一種線性分類器，但此時量少，無法全面學習，也無法解決非線性問題。

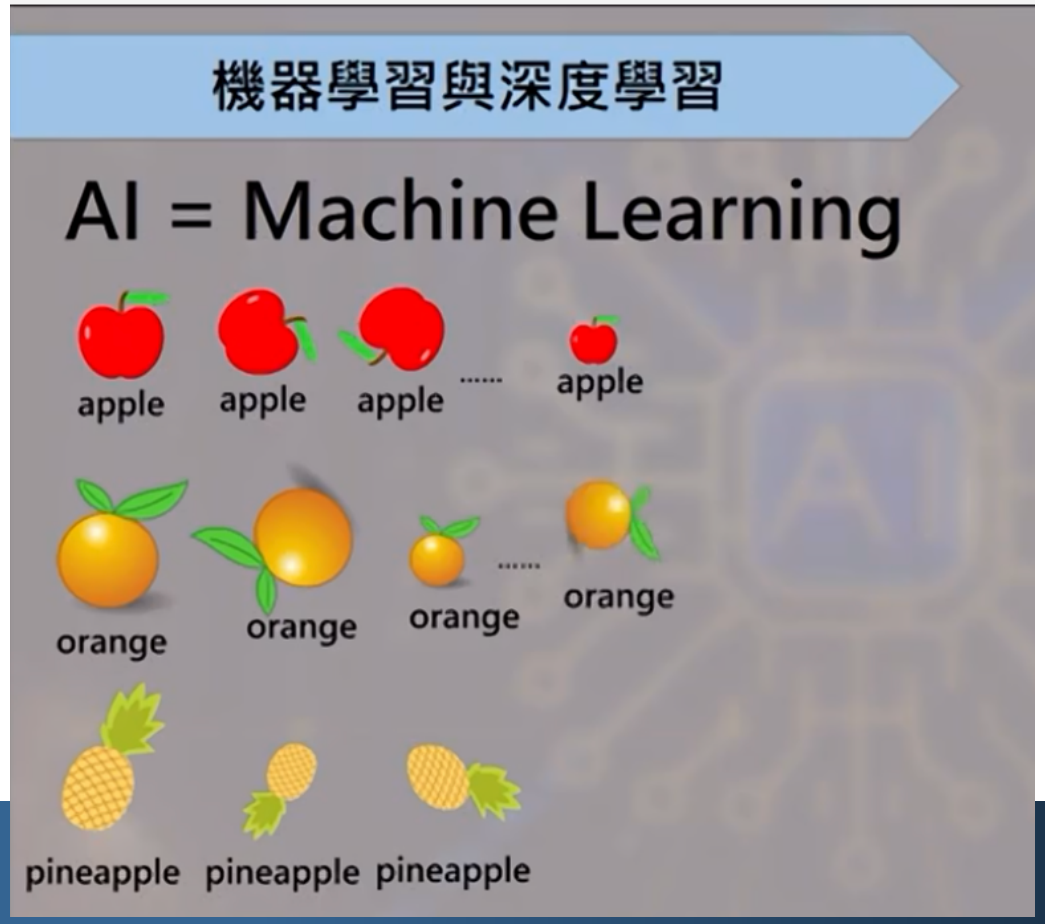
一、人工智慧發展歷程

- **EXPERT SYSTEM**（專家系統）
  - 由於種種限制，科學家嘗試讓電腦在某個領域有智慧就好
  - 例如一個很厲害的銷售員，把推銷商品的技巧一一列出，讓電腦照著做
  - 但老師傅的經驗何其多，很難全部列出
  - 無法超越設計者



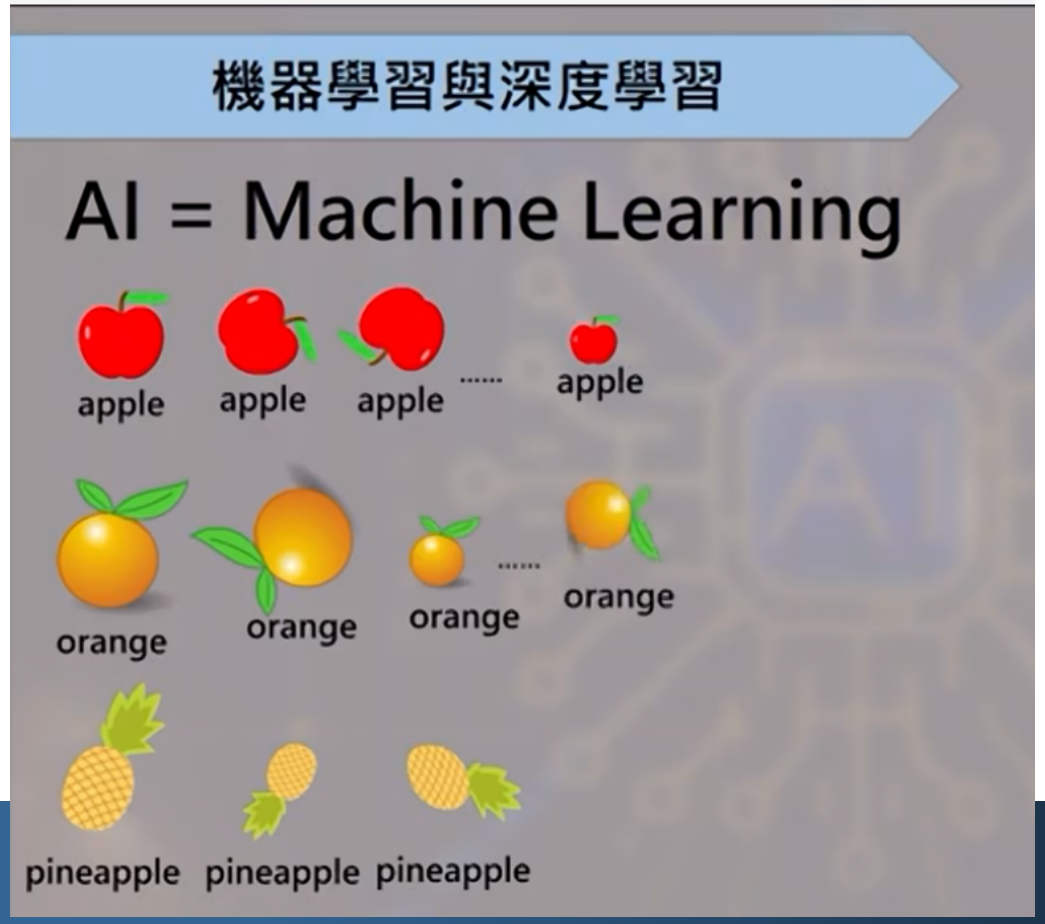
一、人工智慧發展歷程

- **MACHINE LEARNING（機器學習）**
  - 既然經驗教不完，就給電腦很多例子，電腦從資料中自己學習



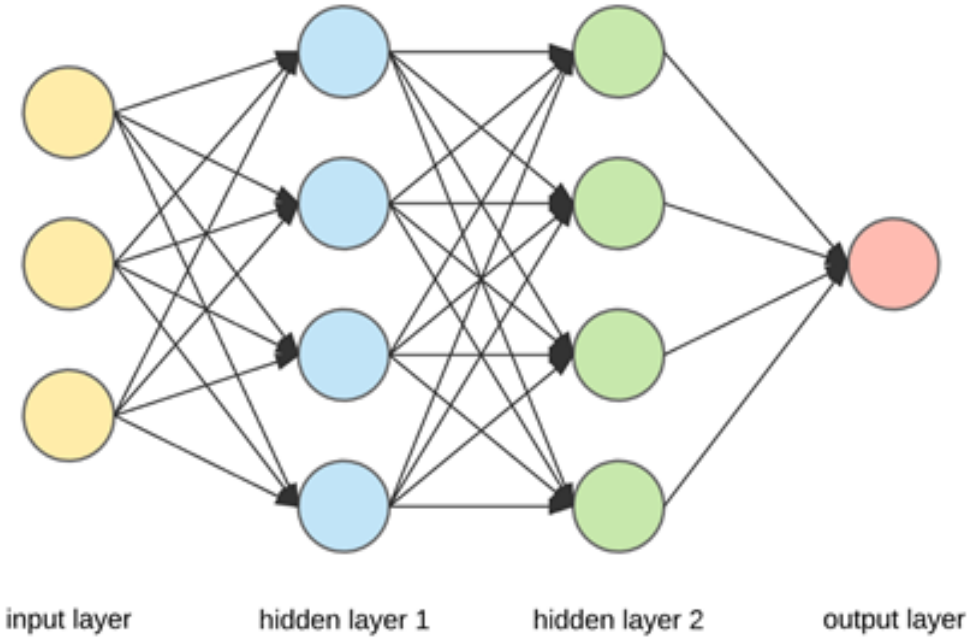
一、人工智慧發展歷程

- **MACHINE LEARNING（機器學習）**
  - 學習什麼？
    - 學習如何分類
      - 找特徵
      - 從資料集比對



一、人工智慧發展歷程

- **MACHINE LEARNING**（機器學習）
  - 單層的感知器無法解決非線性問題
- **DEEP LEARNING**（深度學習）
  - 多層的人工神經網路



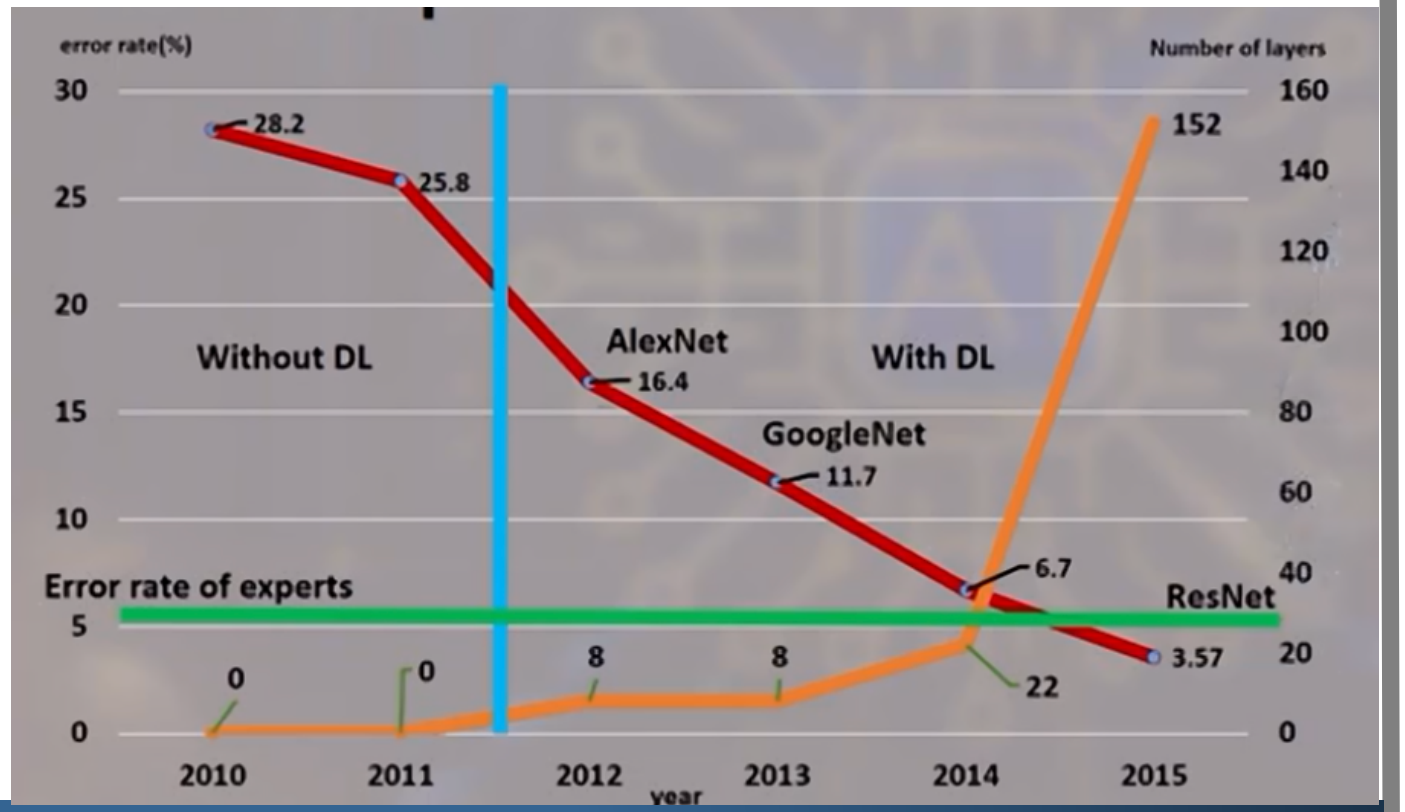
一、人工智慧發展歷程


Deep Learning研究生
朋友覺得我在
我媽覺得我在
大眾覺得我在
指導教授覺得我在
我以為我在
https://blog.csdn.net/qqqinrui
Created with EverCam.

二、人工智慧的應用

二、人工智慧的應用

- 圖片辨識
  - Y軸為辨識圖片錯誤率
  - 綠線為人類專家
  - 紅線為人工智慧
  - 橘線為神經元層數



二、人工智慧的應用

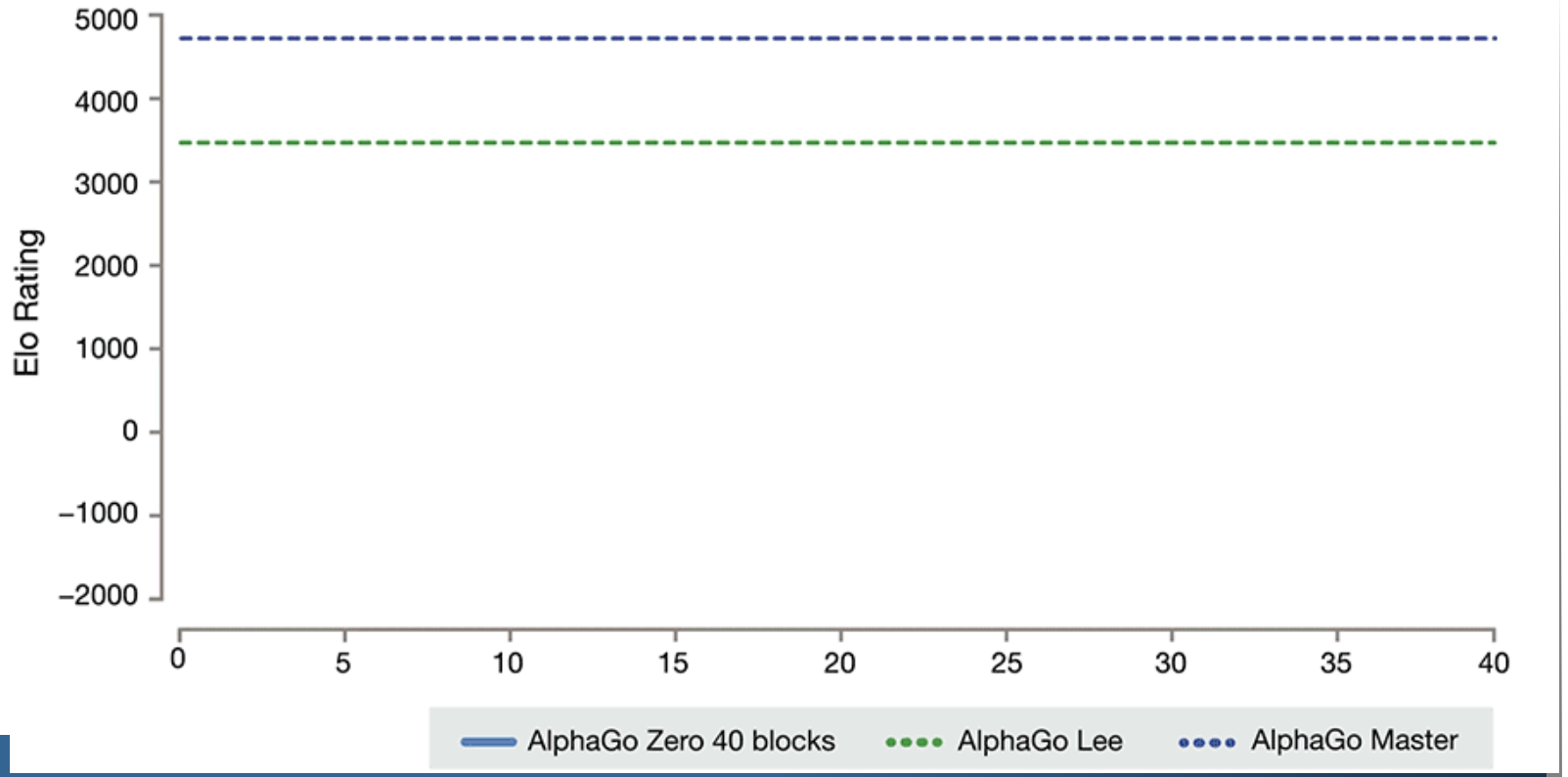
- **ALPHAGO**
  - 使用蒙地卡羅樹搜尋（MONTE CARLO TREE SEARCH），
  - 藉助估值網路（**VALUE NETWORK**）與走棋網路（**POLICY NETWORK**）這兩種深度神經網路

二、人工智慧的應用

版本	使用規則	硬體	Elo等級分的理論峰值	戰績
AlphaGo樊 (v13[64])	中國規則	176個GPU[56]，分散式	3,144[55]	5比0戰勝樊麾
AlphaGo李 (v18[64])		48個TPU[56]，分散式	3,739[55]	4比1戰勝李世乭
AlphaGo Master		4個TPU v2[56]，單機	4,858[55]	網棋60比0戰勝職業棋士； 3比0戰勝柯潔；1比0戰勝人類團隊
AlphaGo Zero	川普–泰勒規則	4個TPU v2[56]，單機	5185[55]	100:0 戰勝AlphaGo李; 與AlphaGo Master 對戰勝率達90%
AlphaZero		4個TPU v2，單機	N/A	60:40 戰勝AlphaGo Zero（3天版本）

二、人工智慧的應用

- **ALPHAGO ZERO:**
  - 只告訴他規則，自己與自己對弈訓練
  - HTTPS://DEEPMIND.COM/BLOG/ARTICLE/ALPHAGO-ZERO-STARTING-SCRATCH



二、人工智慧的應用

- **TESLA**
  -

二、人工智慧的應用

二、人工智慧的應用

二、人工智慧的應用

- 網站
  - GOOGLE搜尋
  - FB
  - YOUTUBE
  - 購物網
  - 垃圾郵件的判斷
  - 等等，好多都有

二、人工智慧的應用

- 影像辨識
  - 醫療
  - 植物
  - 車牌
- 創作
  - 詩集：少女詩人小冰：HTTP://POEM.MSXIAOBING.COM/
  - 畫作

二、人工智慧的應用

- 過去是人類訓練人工智慧
- 現在是人工智慧訓練人類
  - 圍棋AI分級教練
  - 電玩陪打（ALPHA STAR）（200年）
  - HTTPS://TECHNEWS.TW/2020/05/22/AI-THREE-TYPES-OF-DRL/
  - 語音辨識教英文

三、人工智慧相關技術整理

三、人工智慧相關技術整理

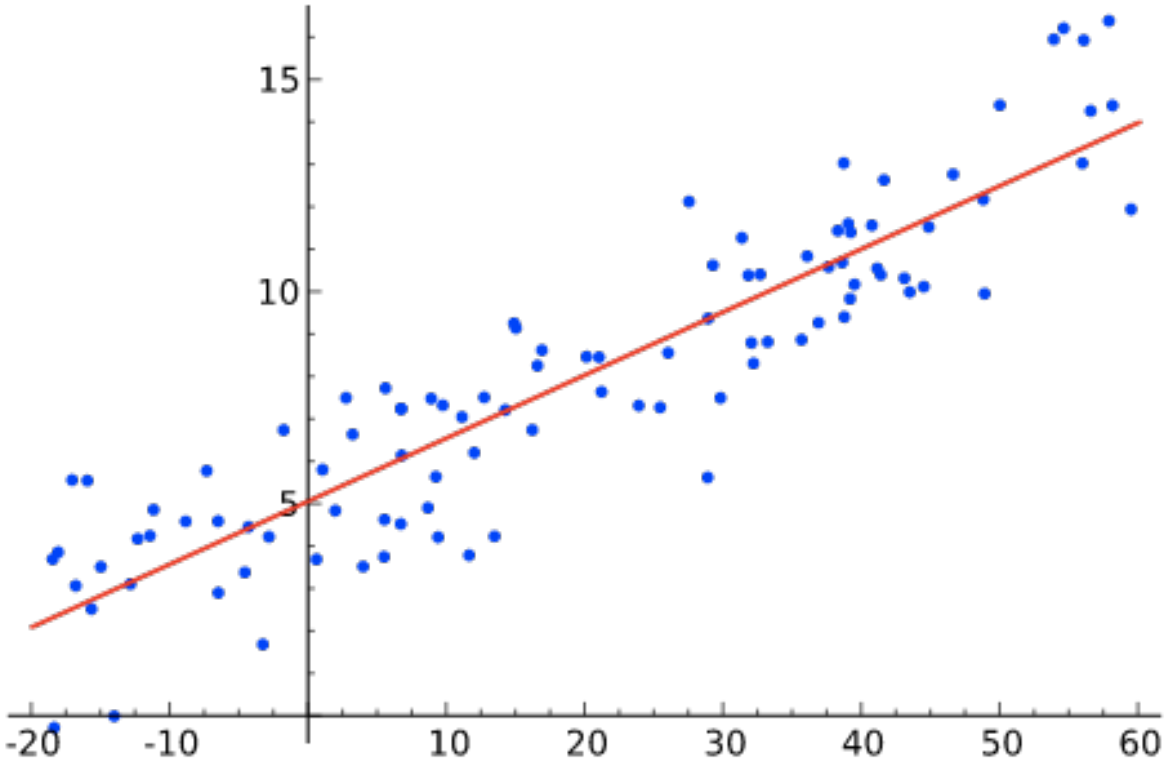
- 人工智慧常用於分類資料，那是因為先分類好了，才能做出判斷
- 專家系統與機器學習的差異是：
  - 專家系統是人類直接告訴機器「特徵」，例如看到水果要先看顏色、葉子、形狀等等特徵來判斷
  - 機器學習是人類給電腦很多水果，電腦自己學著去分類特徵。

三、人工智慧相關技術整理

- 線性迴歸
  - 將許多的(X,Y)資料，用數學算出一條可以代表這些資料的線，稱為迴歸線
  - 迴歸線可以幫我預測在某個Ｘ值時，對應的Ｙ值

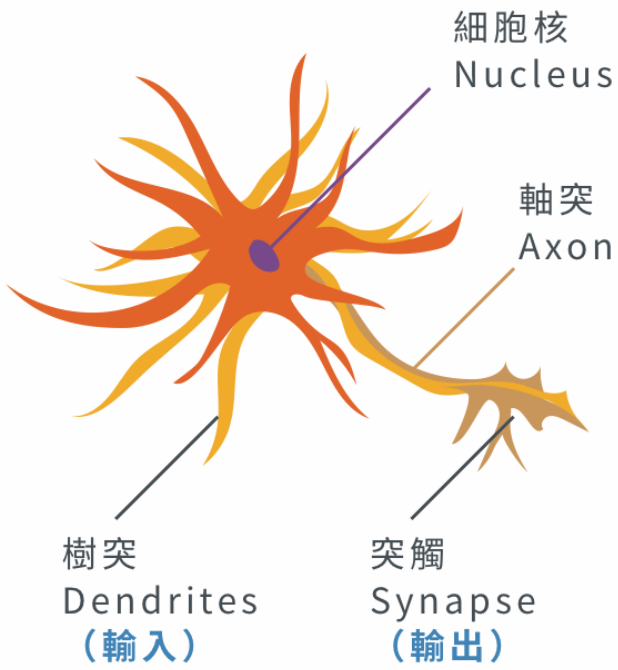
三、人工智慧相關技術整理

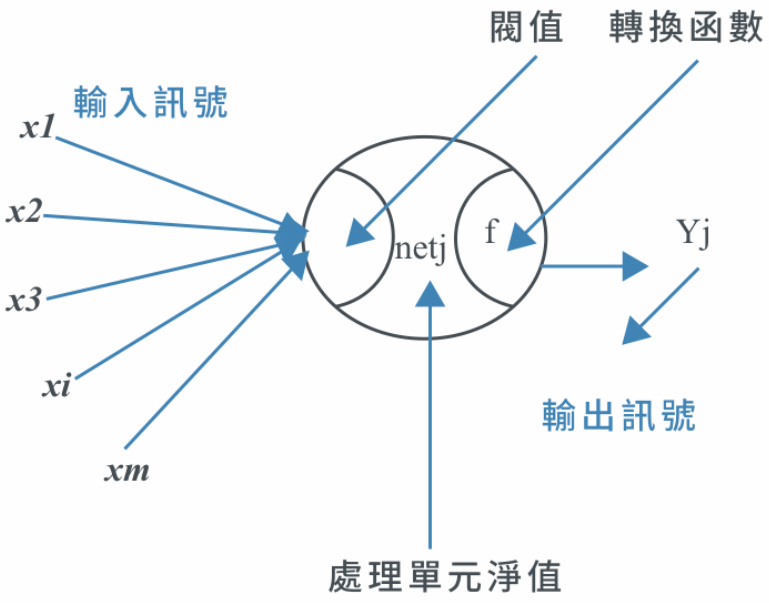
- 線性迴歸



三、人工智慧相關技術整理

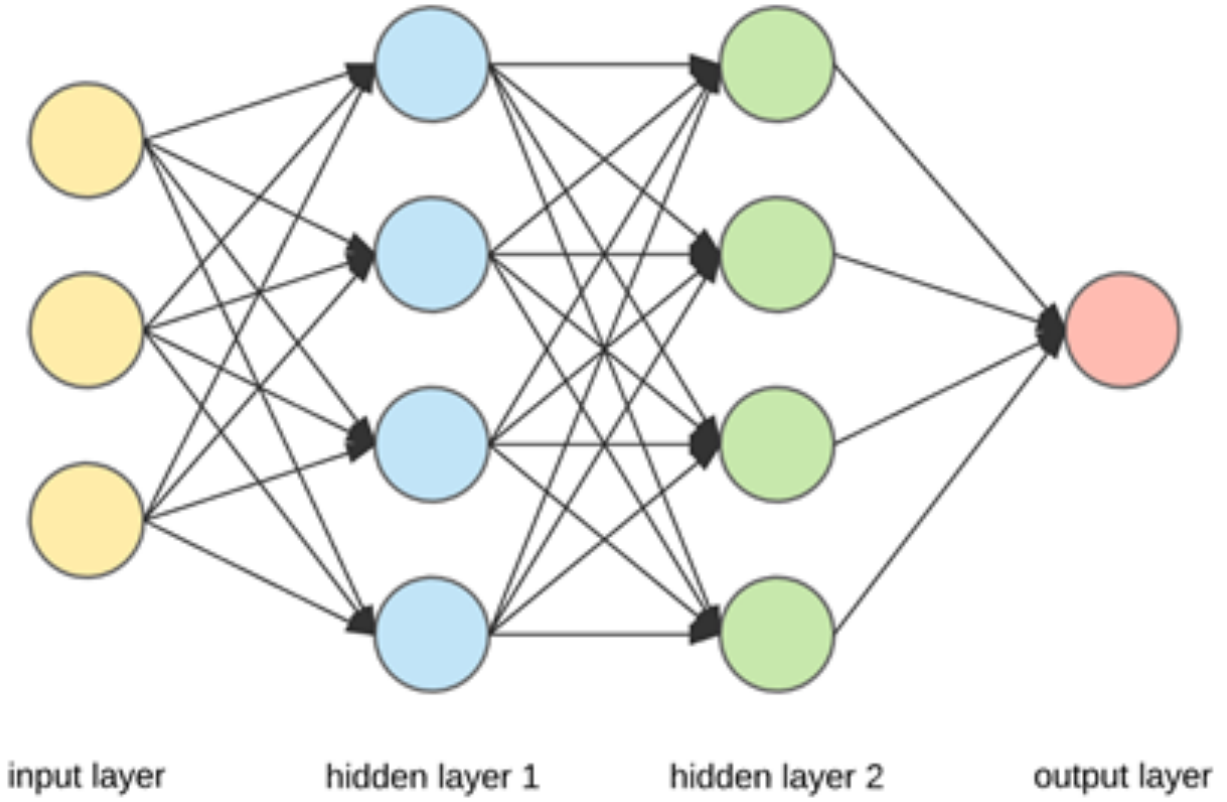
- **ARTIFICIAL NEURAL** ：人工神經
  - 其實就是以前的感知器





三、人工智慧相關技術整理

- **ARTIFICIAL NEURAL NETWORK：**
- 人工神經網路



- 資料來源：一人多工工作室
- 人腦的重量大約一公斤多，預估具有860億個神經元，以及超過100兆條的神經相連

三、人工智慧相關技術整理

- **DATA MINING**：資料探勘
  - 從資料中找到有用的知識
- 用於解決
  - **ASSOCIATION RULE MINING**：關聯規則學習
  - **CLASSIFICATION**：分類
  - **CLUSTERING**：分群

三、人工智慧相關技術整理

- **MACHINE LEARNING**：機器學習
  - 監督式
  - 非監督式
  - 增強式
  - 半監督式
  - 互動式
  - 轉換

三、人工智慧相關技術整理

- **MACHINE LEARNING**：機器學習
  - 監督式：給電腦的資料是有標籤的
  - 非監督式：給電腦的資料是沒有標籤的
  - 增強式：電腦藉由跟環境的互動來學習

三、人工智慧相關技術整理

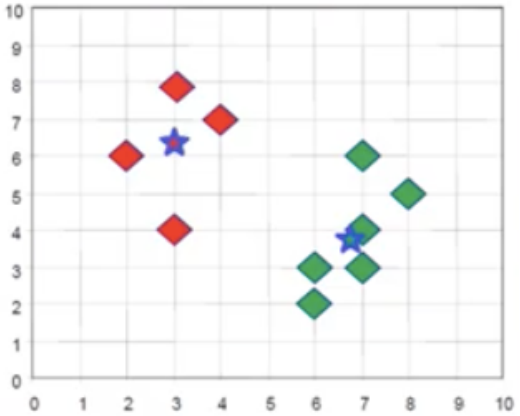
- 監督式學習
  - 最可視為「有老師指導」的學習方法，因為老師擁有正確答案（或稱標記），所以在訓練階段可以提供每個訓練樣本所對應的標記資訊，根據訓練樣本的對應標記資訊所設計的機器學習方法就叫做監督式學習
  - 回想一下：最有效的學習方式—從錯誤中學習！

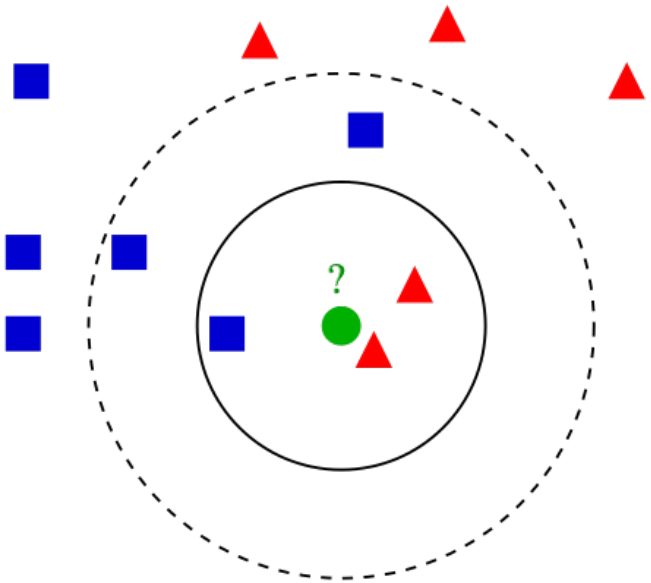
三、人工智慧相關技術整理

- 監督式學習
  - 監督式學習就是透過目前的結果距離目標的誤差回頭修正系統參數的方法（目標導向）
  - 就像射箭選手透由目測結果修正自身的射擊姿勢

三、人工智慧相關技術整理

- 監督式學習
  - 最短距離分類器
  - KNN（K NEAREST NEIGHBOR）分類器
  - 決策樹





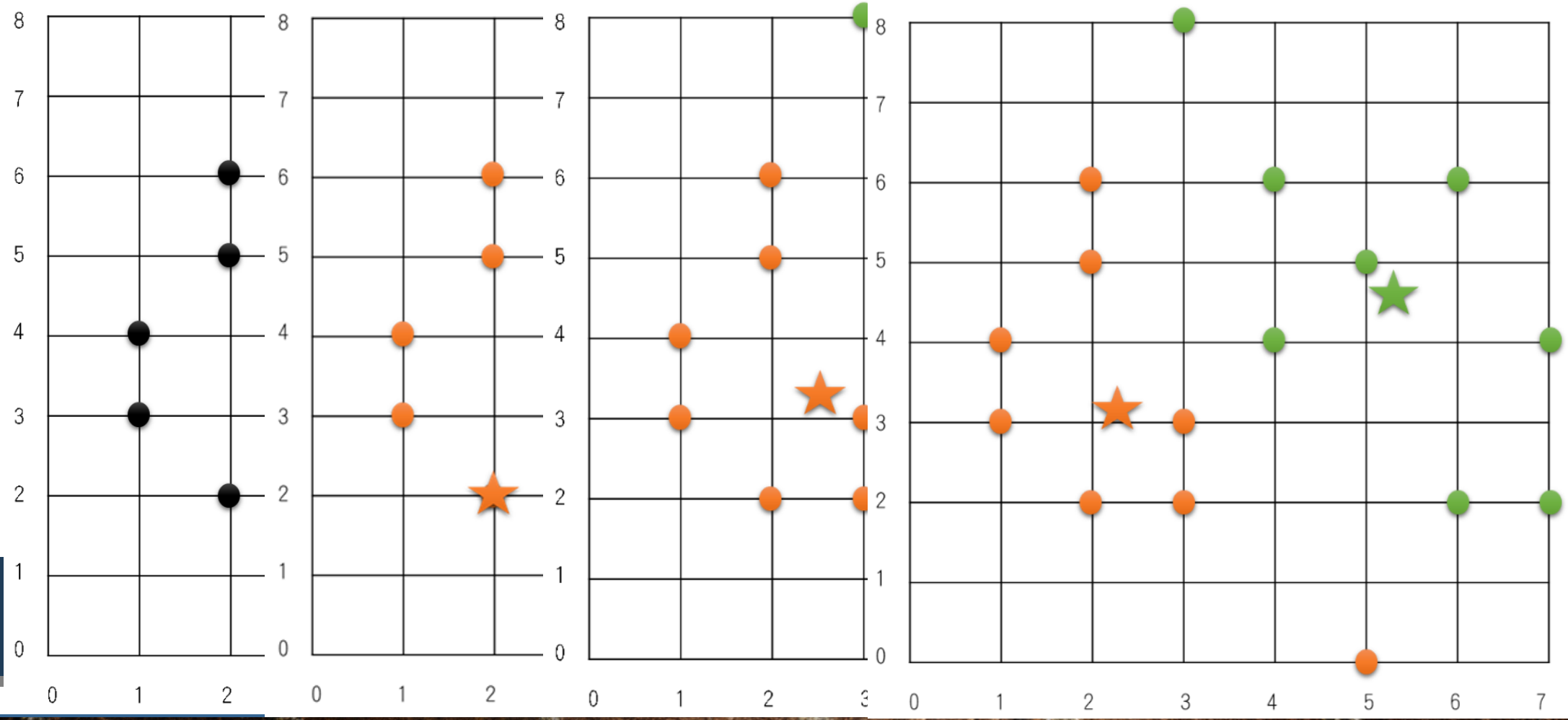
三、人工智慧相關技術整理

- 非監督式學習
  - 「沒有老師指導」的學習方法，在訓練階段無需提供訓練樣本所對應的標記資訊
  - 由於欠缺標記資訊，因此通常會就訓練樣本彼此之間的相似性進行聚類分群
  - 避免掉監督式學習因為需要取得每個訓練樣本所對應的標記資訊所要付出的可觀人力成本

三、人工智慧相關技術整理

- 非監督式學習
  - K-MEANS演算法（K個資料的群聚中心）
  - 階層式分群法

三、人工智慧相關技術整理



三、人工智慧相關技術整理

- 增強式學習：機器人找路徑
  - 貝爾曼方程式
  - Q-LEARNING
  - 知道結果是好的，但不知道原因

四、人工智慧與機器視覺

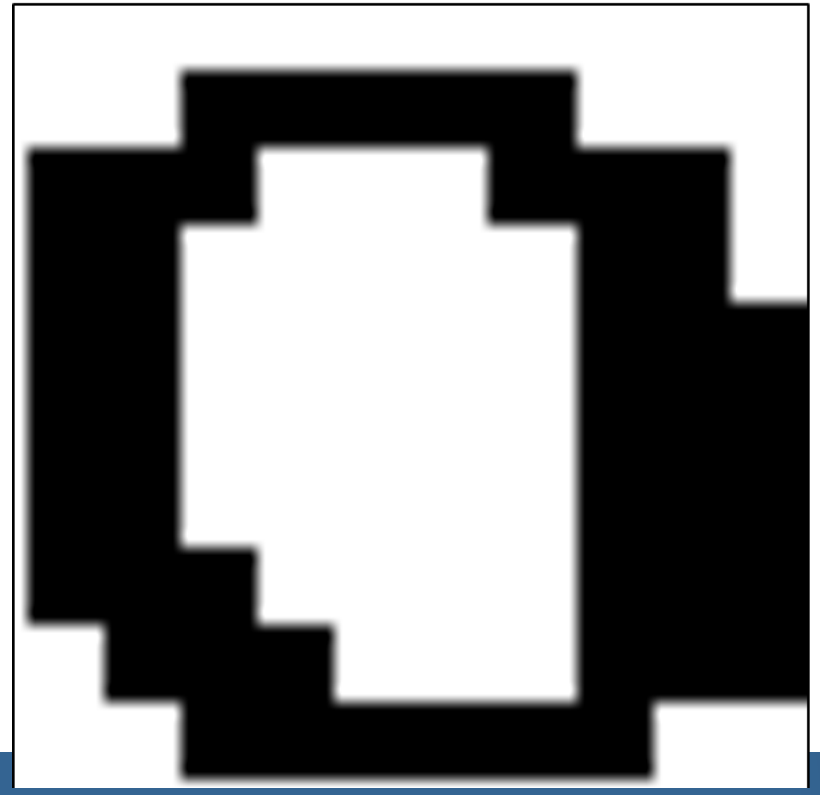
四、人工智慧與機器視覺

- 深度學習：
  - 為了處理更複雜的資料而生的
  - CONVOLUTIONAL NEURAL NETWORKS（CNN）：卷積類神經網路
  - RECURRENT NEURAL NETWORK（RNN）：遞歸神經網路

四、人工智慧與機器視覺

- 機器視覺：
  - 10*10的黑白圖片
  - 有100個像素
  - 每個像素都有一個數值

原始影像



數位化的值

255	255	255	255	255	255	255	255	255	255
255	255	0	0	0	0	0	255	255	255
0	0	0	255	255	255	0	0	0	255
0	0	255	255	255	255	255	0	0	255
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
255	0	0	0	255	255	255	0	0	0
255	255	0	0	0	0	0	0	255	255

四、人工智慧與機器視覺

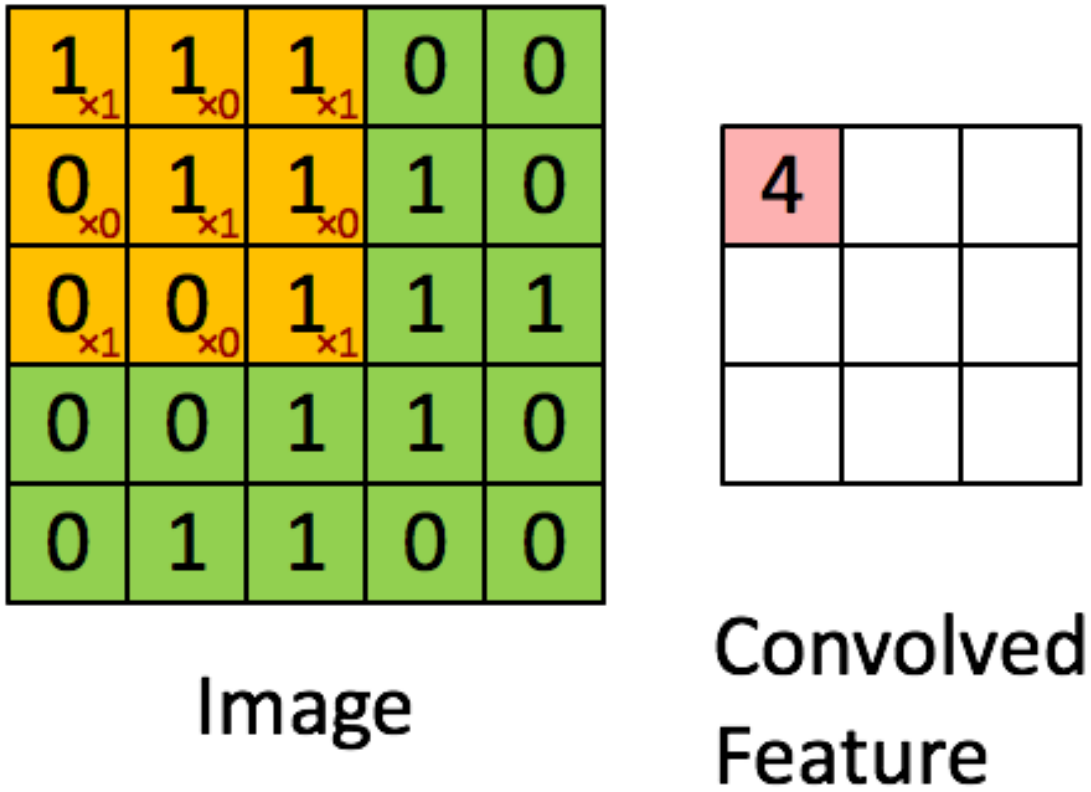
- 機器視覺：
  - 把所有像素的數值直接丟進去學習、分類，好嗎？

四、人工智慧與機器視覺

- **CNN:卷積類神經網路**
  - 卷積的技術可以減輕類神經網路的負擔
  - 人類的視覺本身就比較像卷積的方式看東西

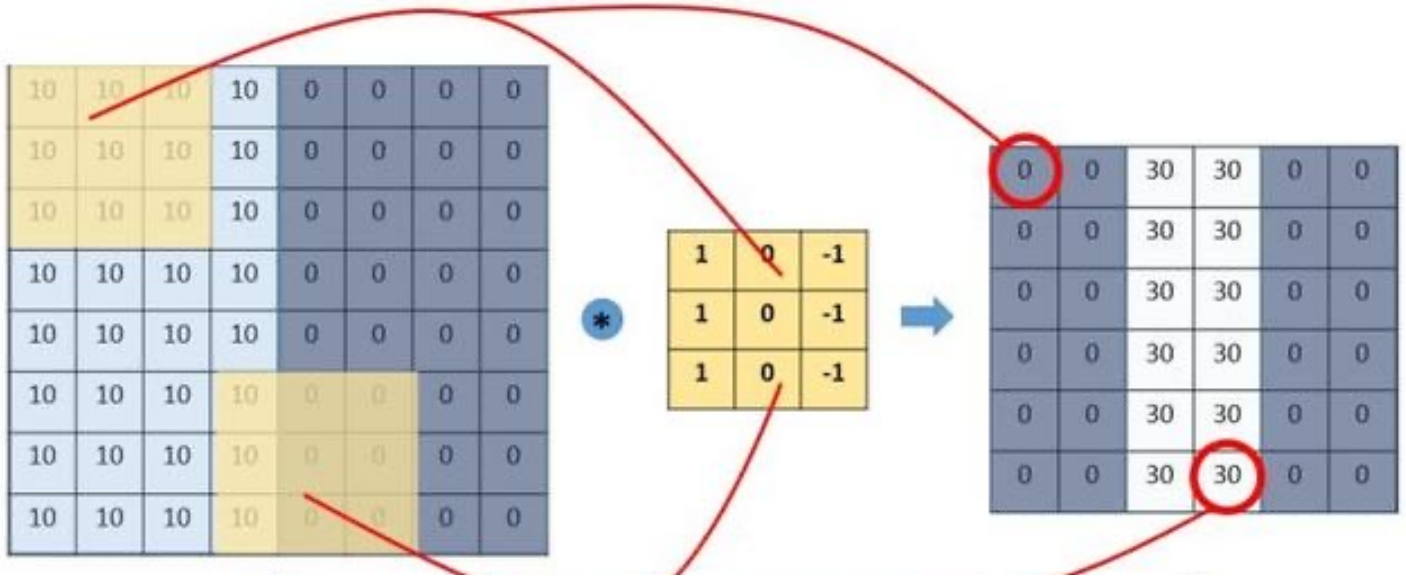
四、人工智慧與機器視覺

- 卷積：把圖片的一個區塊濃縮成一個值
  - 先經過**CONVOLUTION KERNEL**計算矩陣
  - 再總加矩陣，作為特徵值，處理後的圖片稱為**FEATURE MAP**



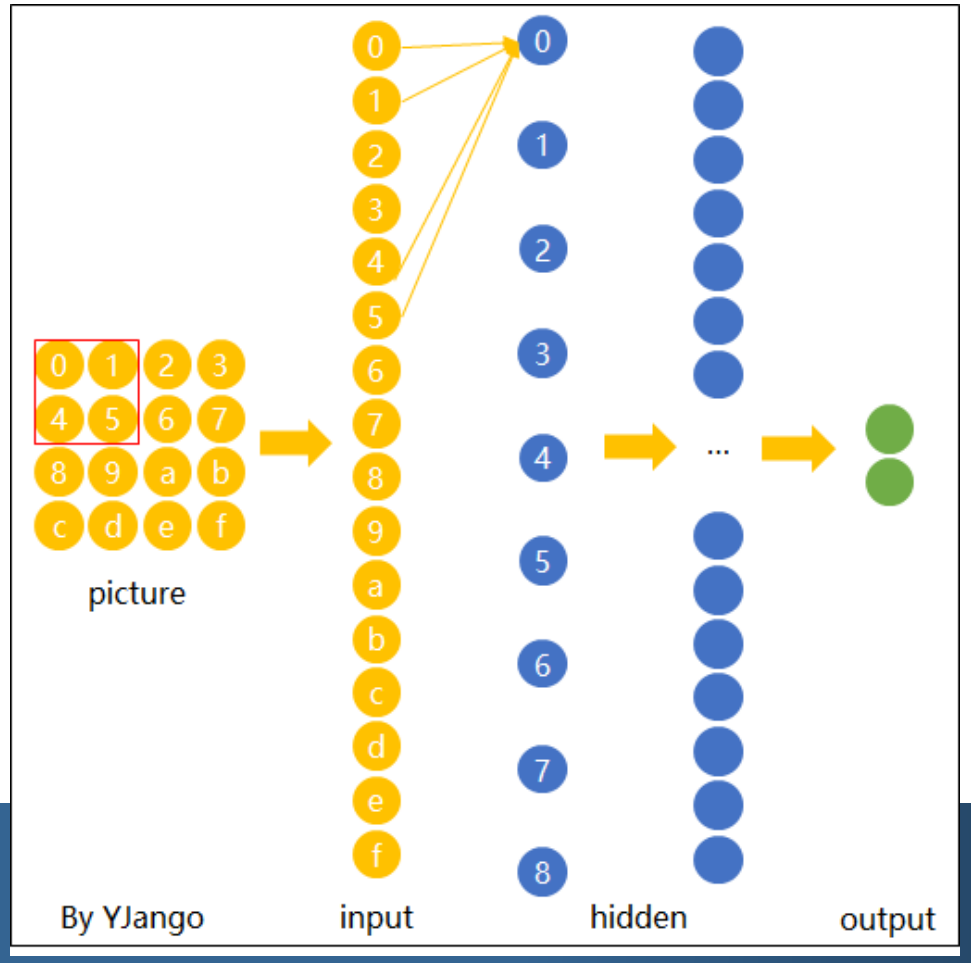
四、人工智慧與機器視覺

- **CNN:卷積類神經網路**
  - 好的**KERNEL**可以幫我們找出特徵



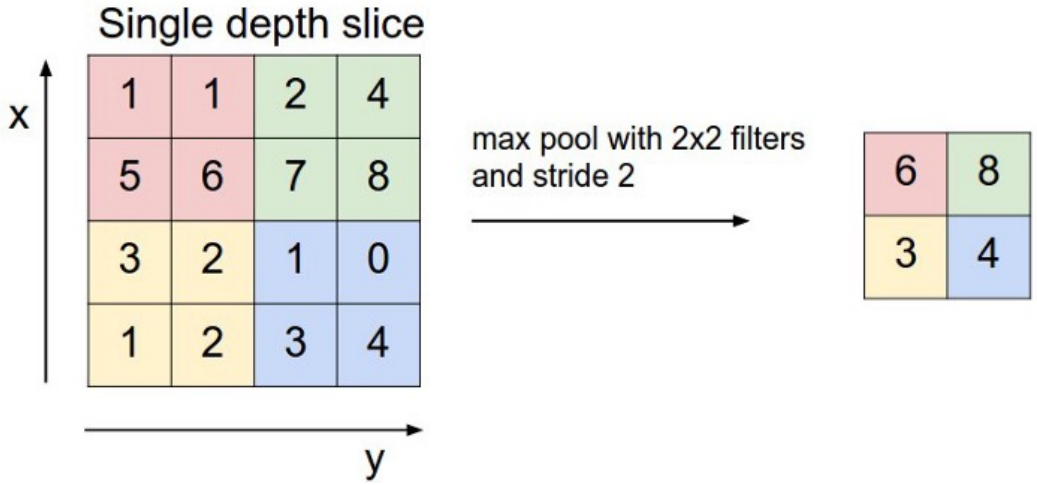
四、人工智慧與機器視覺

- **CNN:卷積類神經網路**
  - 卷積處理的神經連結
  - 部分連接層



四、人工智慧與機器視覺

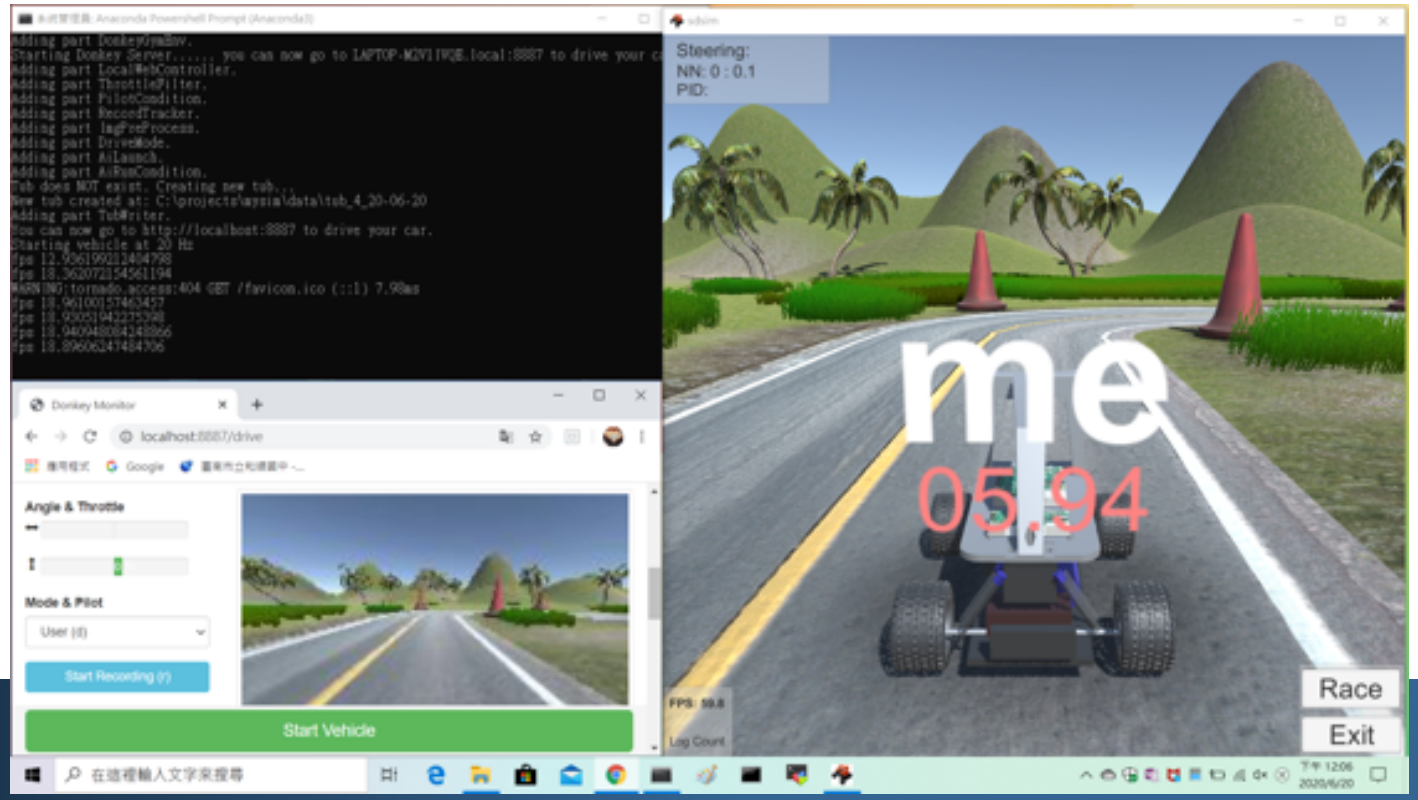
- **CNN:卷積類神經網路**
  - **POOLING（池化）：保留特徵**



    - **AVERAGE POOLING**：定義一部分的矩陣，所有值相加，取平均，會把圖片再變小
    - **MAX POOLING**：較常用，也有非常好的表現。因為失去的資訊往往不太重要。

四、人工智慧與機器視覺

- **AI影像辨識車**

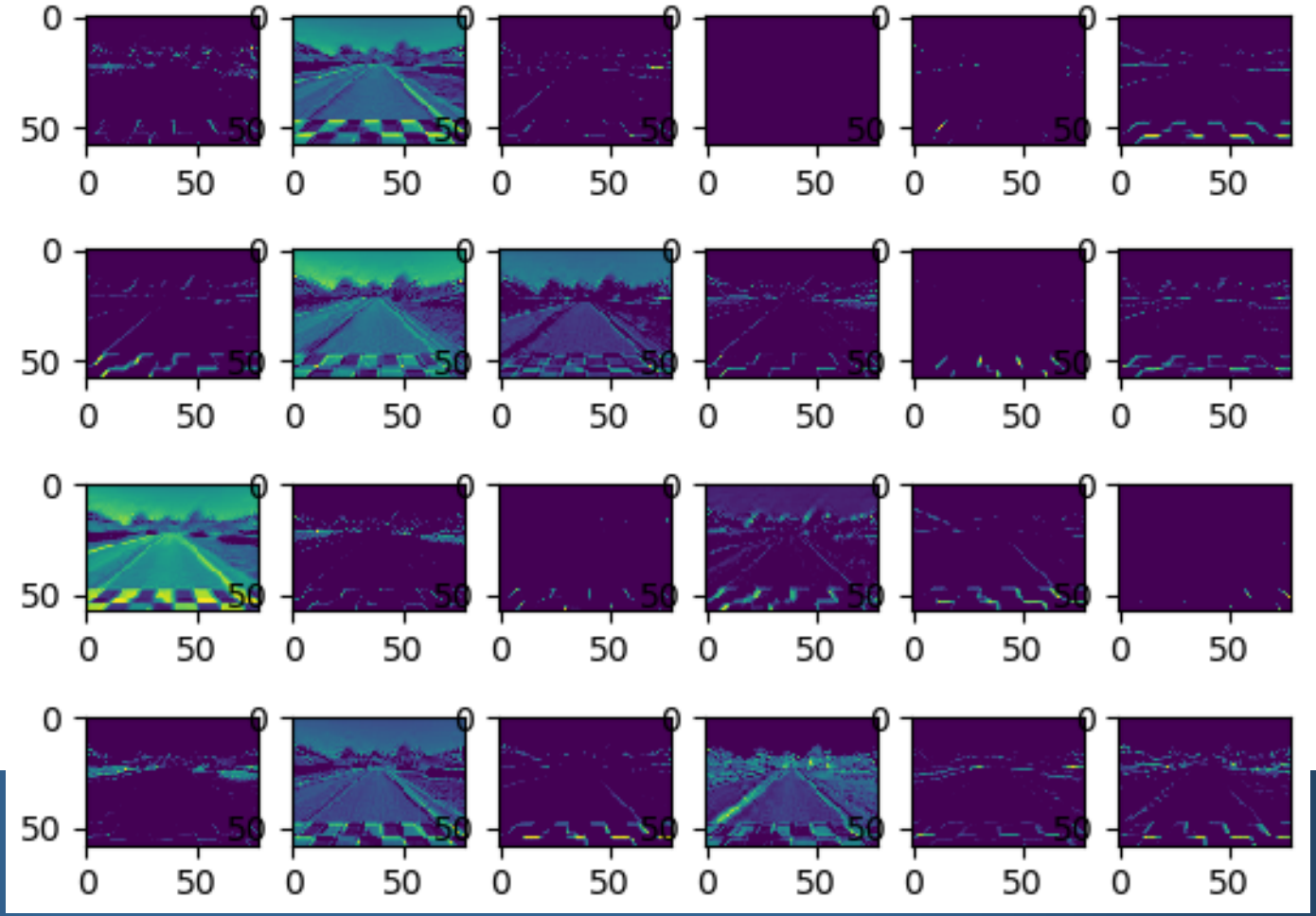


四、人工智慧與機器視覺

- 原始圖片：
  - 160*120

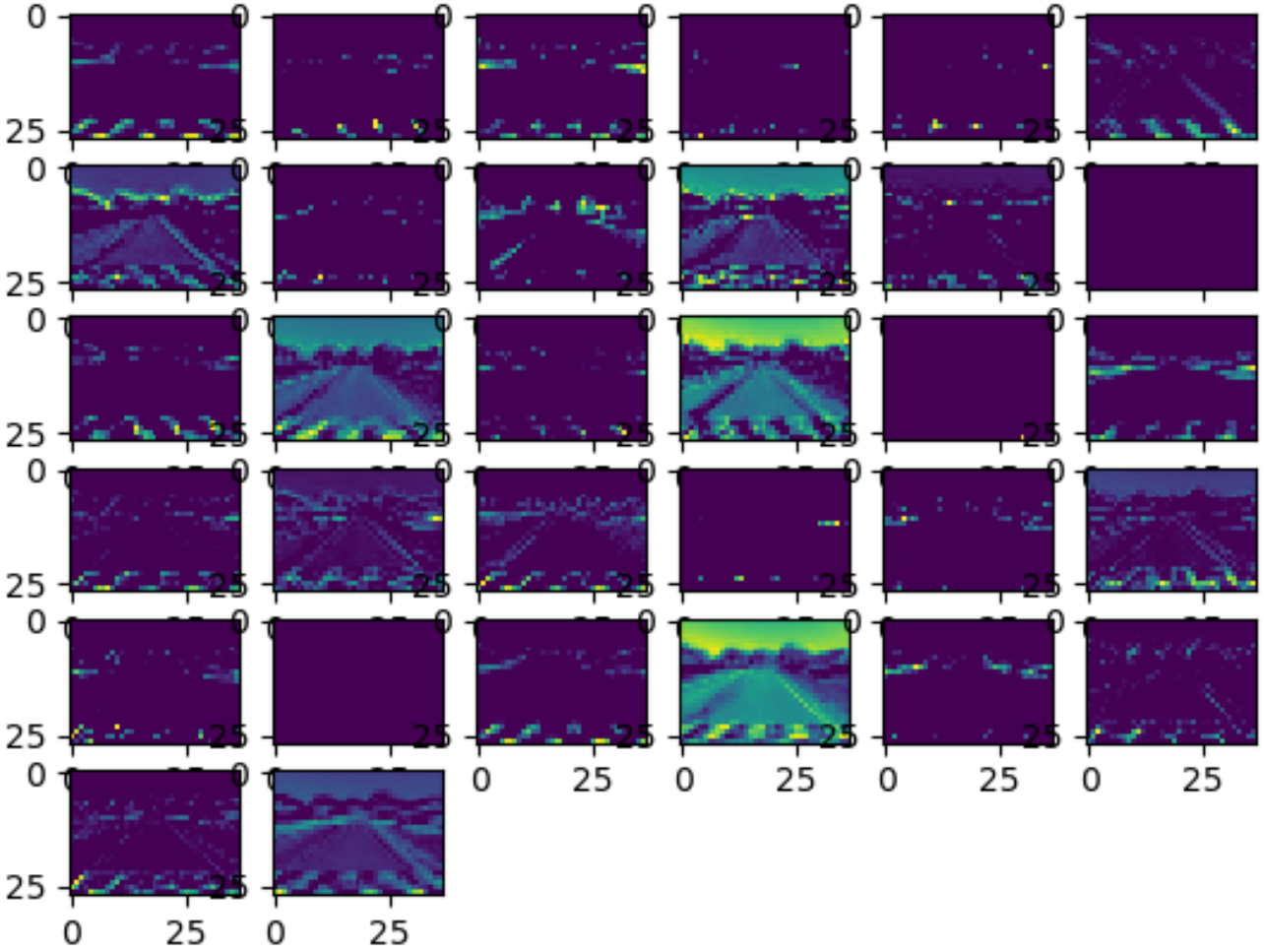
四、人工智慧與機器視覺

- 處理後：
  - 第一層：24張



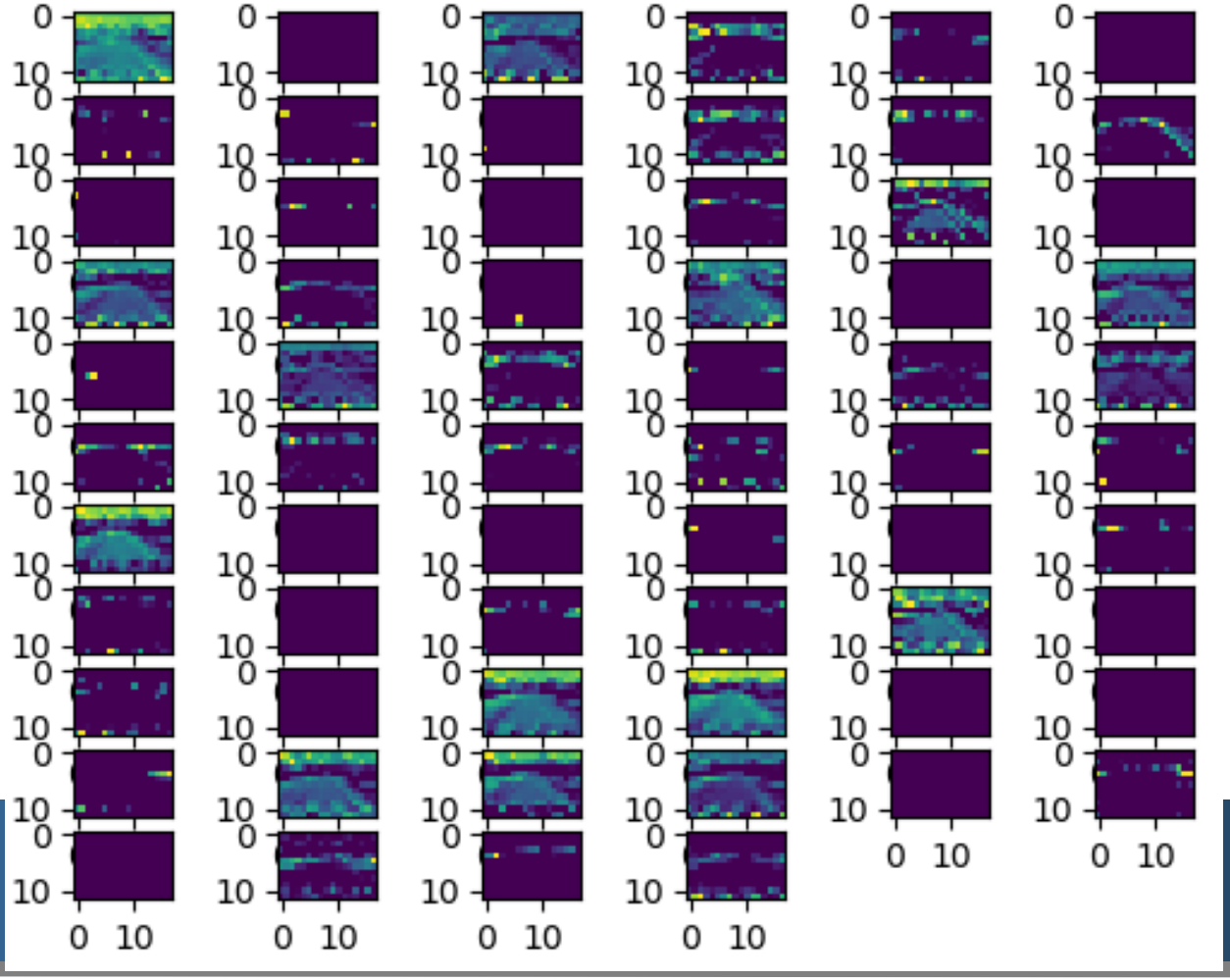
四、人工智慧與機器視覺

- **處理後：**
  - **第二層：32張**



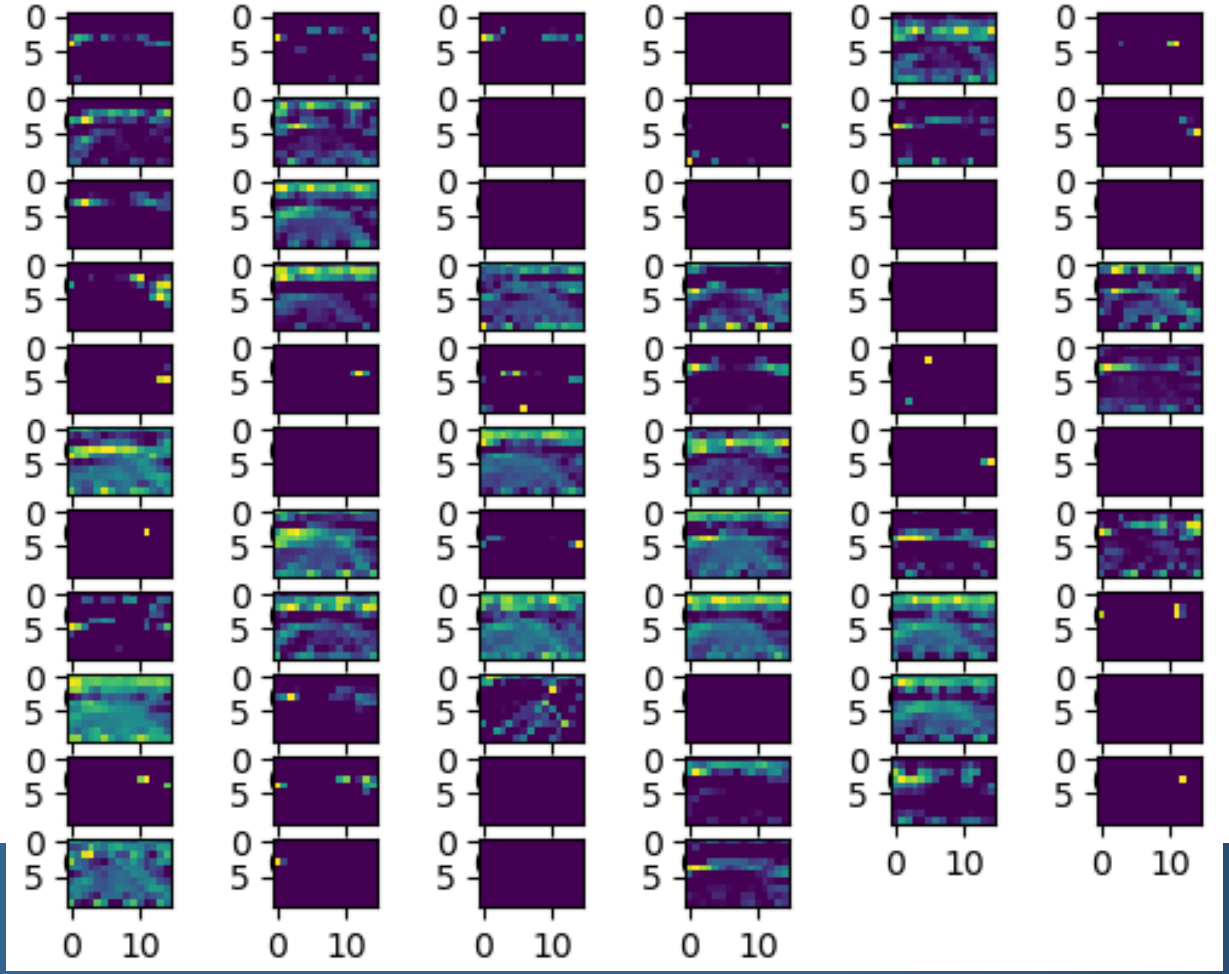
四、人工智慧與機器視覺

- 處理後：
  - 第三層：64張



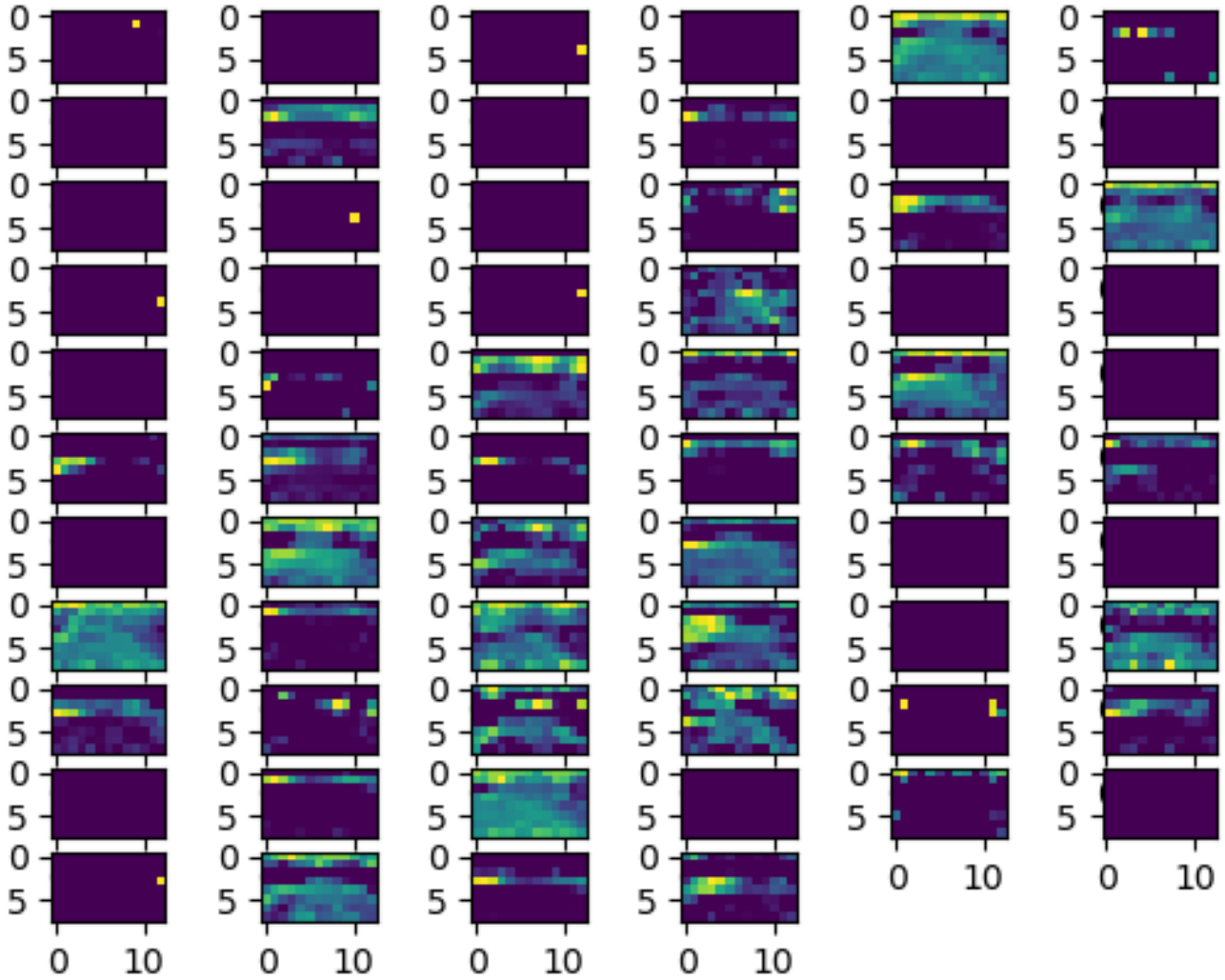
四、人工智慧與機器視覺

- 處理後：
  - 第四層：64張



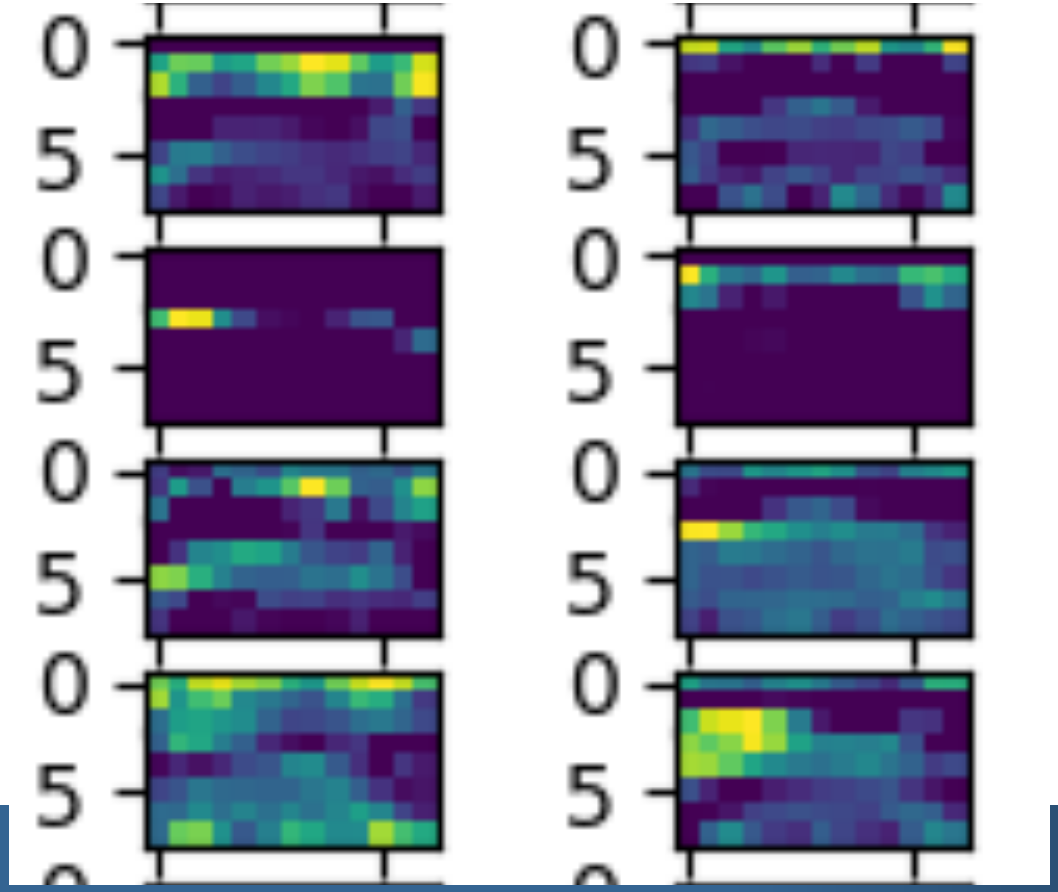
四、人工智慧與機器視覺

- 處理後：
  - 第五層：64張



四、人工智慧與機器視覺

- 處理後：
  - 第五層局部圖



四、人工智慧與機器視覺

- 數值：

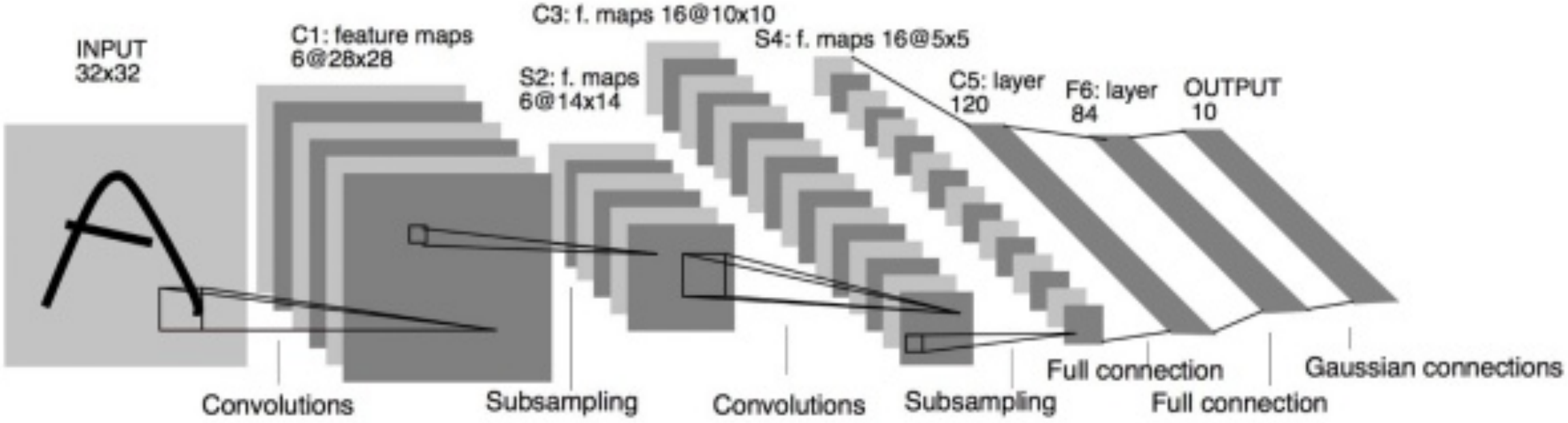
```
layer 1 shape: (58, 78, 24)
layer 2 shape: (27, 37, 32)
layer 3 shape: (12, 17, 64)
layer 4 shape: (10, 15, 64)
layer 5 shape: (8, 13, 64)
```

四、人工智慧與機器視覺

- **CNN:卷積類神經網路**
  - 全連接層：把二維資料攤平成一維後輸入的神經元
  - 又稱為密集層（DENSE）

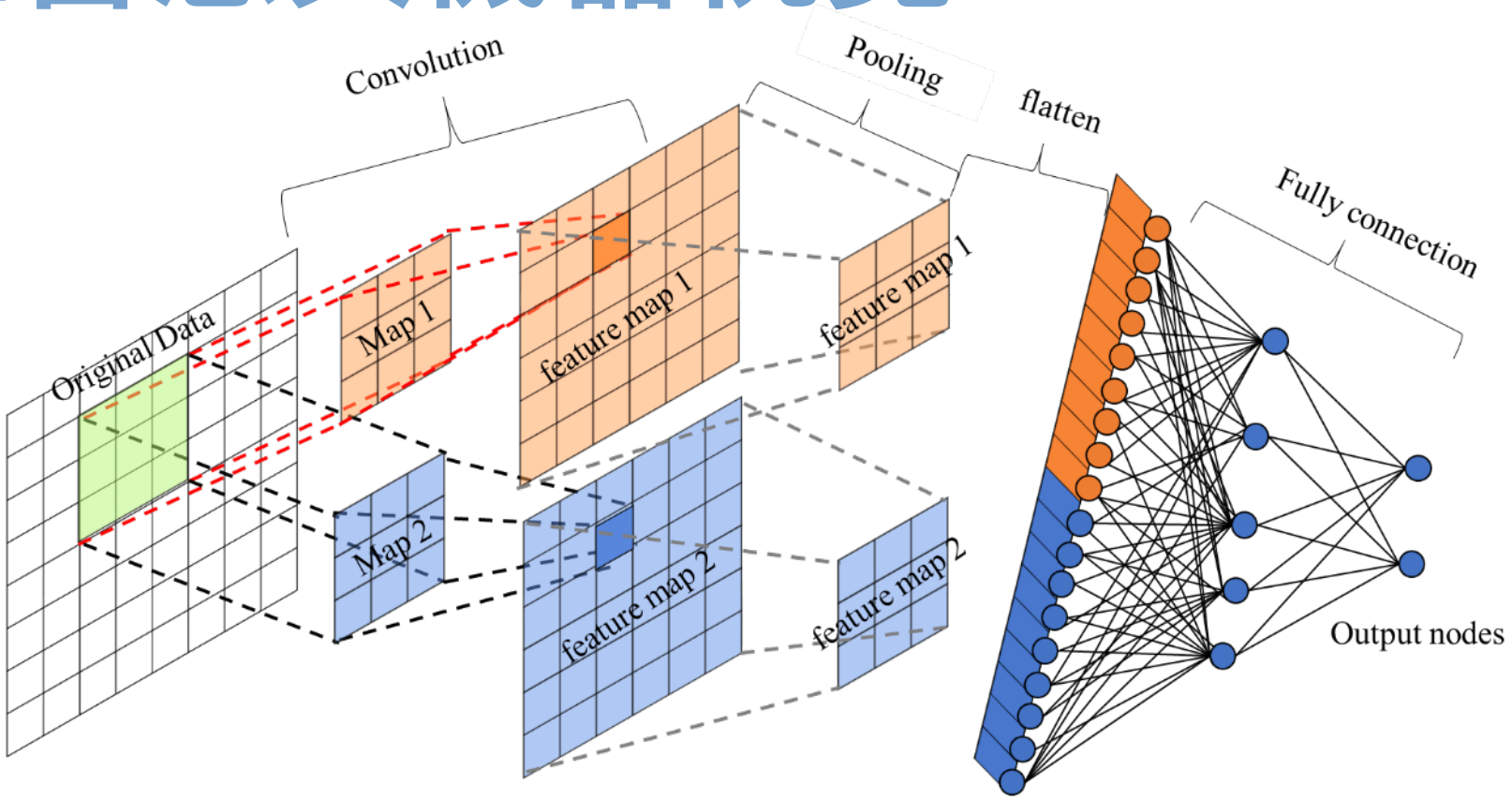
四、人工智慧與機器視覺

- **CNN:卷積類神經網路示意圖**



四、人工智慧與機器視覺

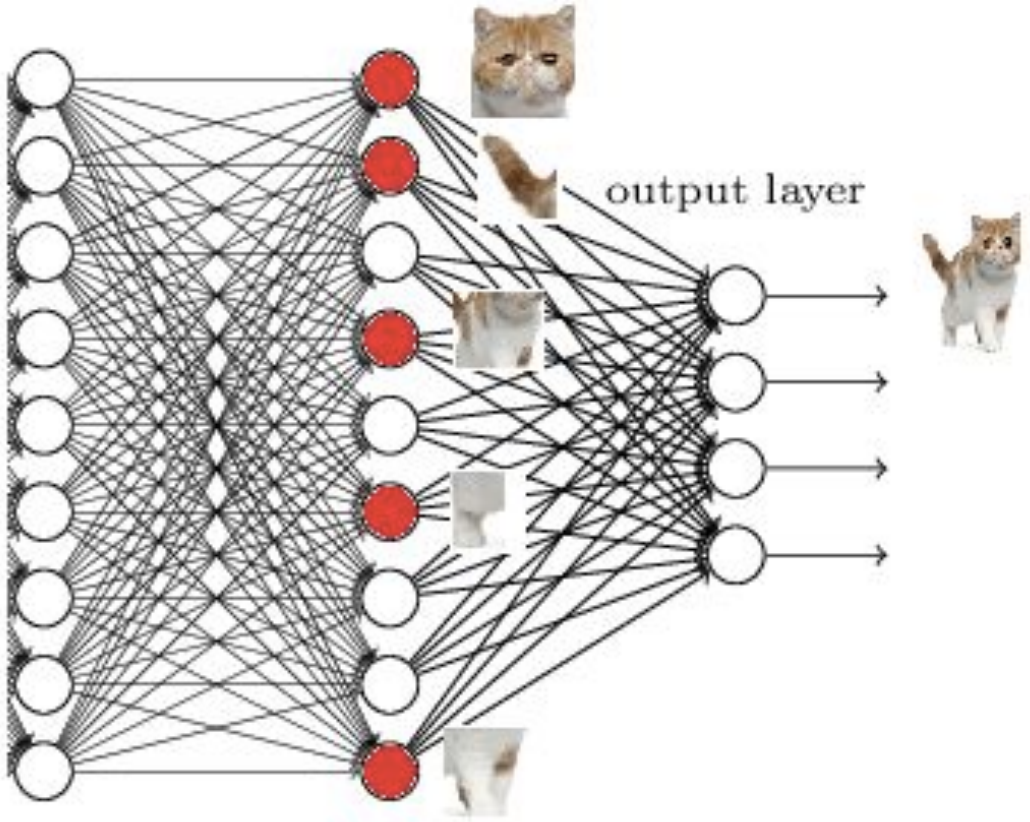
- **CNN:卷積類神經網路示意圖**



- 資料來源：HTTPS://MEDIUM.COM/@CHIH.SHENG.HUANG821/%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-CONVOLUTIONAL-NEURAL-NETWORK-CNN-CNN%E9%81%8B%E7%AE%97%E6%B5%81%E7%A8%8B-ECAEC240A631

四、人工智慧與機器視覺

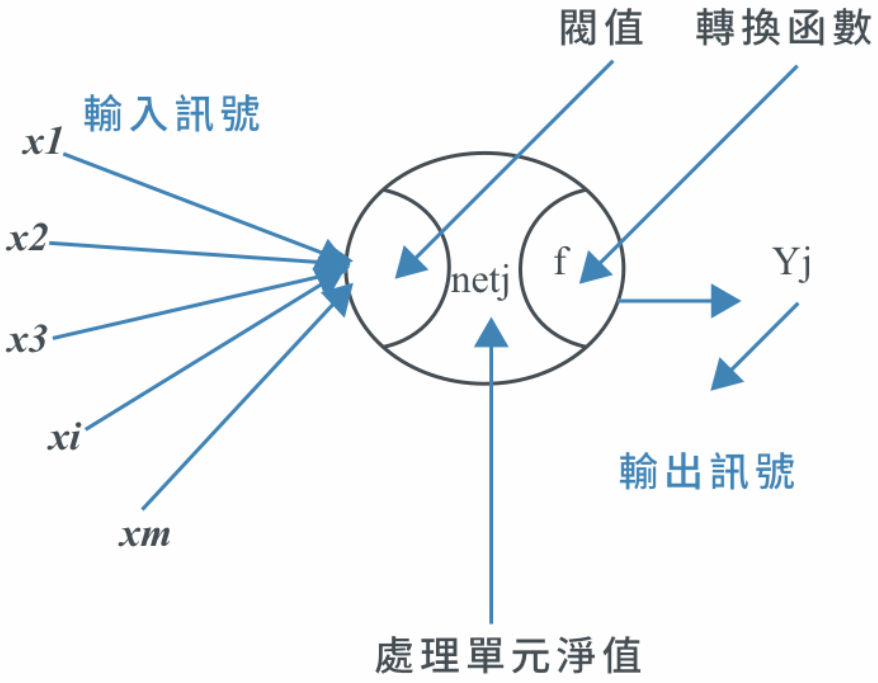
- **CNN:卷積類神經網路**
  - 輸出層



全連接層

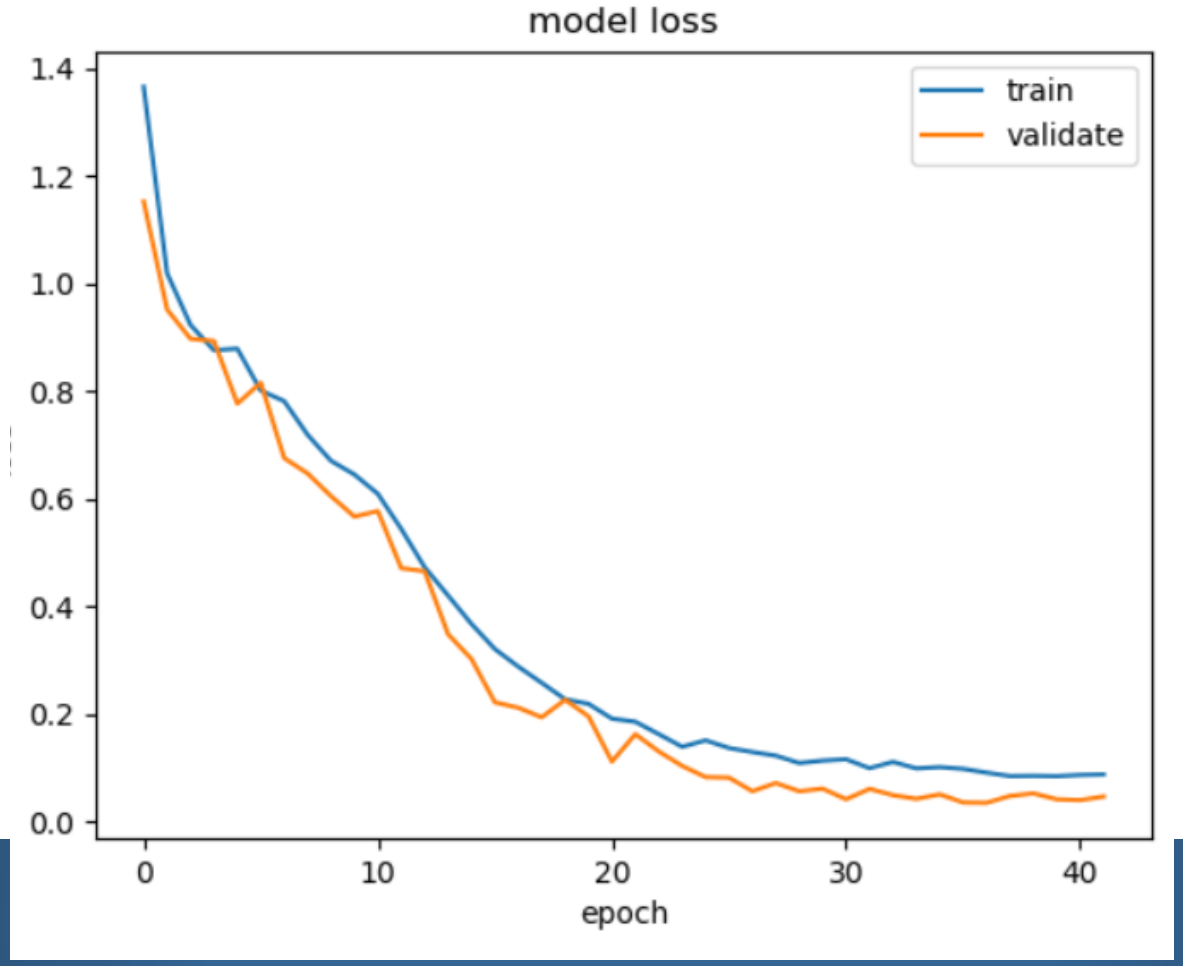
四、人工智慧與機器視覺

- **CNN:卷積類神經網路**
  - 到底什麼是「訓練」？



四、人工智慧與機器視覺

- 收斂圖：



五、數學時間

五、數學時間

- 數值有什麼秘密㊙️？
- 輸入是120*160

```
layer 1 shape: (58, 78, 24)
layer 2 shape: (27, 37, 32)
layer 3 shape: (12, 17, 64)
layer 4 shape: (10, 15, 64)
layer 5 shape: (8, 13, 64)
```

五、數學時間

- 你看得出來KERNEL是幾＊幾嗎？

```
training with model type <class 'donkeycar.parts.keras.KerasLinear'>
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
=================================================================
img_in (InputLayer)          [(None, 224, 224, 3)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 110, 110, 24)      1824      img_in[0][0]
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 24)      0         conv2d_1[0][0]
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 53, 53, 32)        19232     dropout[0][0]
_________________________________________________________________
dropout_1 (Dropout)          (None, 53, 53, 32)        0         conv2d_2[0][0]
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 64)        51264     dropout_1[0][0]
_________________________________________________________________
dropout_2 (Dropout)          (None, 25, 25, 64)        0         conv2d_3[0][0]
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 23, 23, 64)        36928     dropout_2[0][0]
```

System throttled

五、數學時間

- 參數量怎麼來的？

```
dropout_3 (Dropout)          (None, 23, 23, 64)   0          conv2d_4[0][0]
conv2d_5 (Conv2D)            (None, 21, 21, 64)   36928      dropout_3[0][0]
dropout_4 (Dropout)          (None, 21, 21, 64)   0          conv2d_5[0][0]
flattened (Flatten)          (None, 28224)        0          dropout_4[0][0]
dense (Dense)                (None, 100)          2822500    flattened[0][0]
dropout_5 (Dropout)          (None, 100)          0          dense[0][0]
dense_1 (Dense)              (None, 50)           5050       dropout_5[0][0]
dropout_6 (Dropout)          (None, 50)           0          dense_1[0][0]
n_outputs0 (Dense)           (None, 1)            51         dropout_6[0][0]
n_outputs1 (Dense)           (None, 1)            51         dropout_6[0][0]
==================================================================================
Total params: 2,973,828
Trainable params: 2,973,828
Non-trainable params: 0
```

System throttled due to Ov

五、數學時間

- 這在幹嘛？

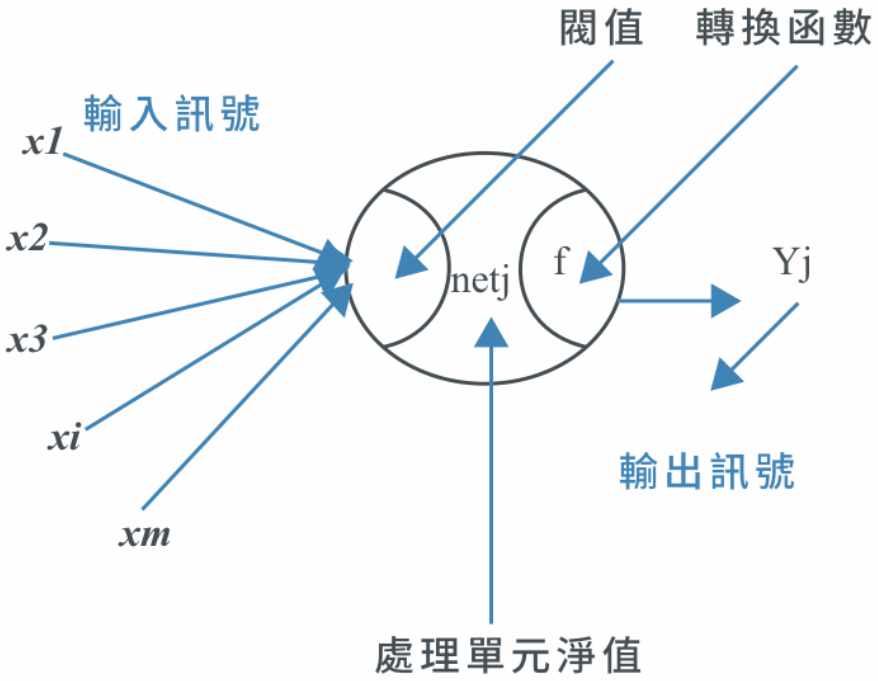
```
Epoch 4/100
16/16 [==============================] - 1s 45ms/step - loss: 0.3283 - n_outputs0_loss: 0

Epoch 00004: val_loss improved from 0.36250 to 0.32829, saving model to C:\projects\mysim
65/65 [==============================] - 12s 186ms/step - loss: 0.3497 - n_outputs0_loss:
loss: 0.0692
Epoch 5/100
16/16 [==============================] - 1s 45ms/step - loss: 0.2702 - n_outputs0_loss: 0

Epoch 00005: val_loss improved from 0.32829 to 0.27020, saving model to C:\projects\mysim
65/65 [==============================] - 12s 187ms/step - loss: 0.2954 - n_outputs0_loss:
loss: 0.0573
Epoch 6/100
16/16 [==============================] - 1s 46ms/step - loss: 0.2376 - n_outputs0_loss: 0

Epoch 00006: val_loss improved from 0.27020 to 0.23763, saving model to C:\projects\mysim
65/65 [==============================] - 12s 187ms/step - loss: 0.2567 - n_outputs0_loss:
loss: 0.0525
Epoch 7/100
16/16 [==============================] - 1s 46ms/step - loss: 0.2091 - n_outputs0_loss: 0

Epoch 00007: val_loss improved from 0.23763 to 0.20912, saving model to C:\projects\mysim
65/65 [==============================] - 12s 188ms/step - loss: 0.2119 - n_outputs0_loss:
loss: 0.0431
```

五、數學時間

- 耗費多少記憶體？
- HTTPS://BLOG.CSDN.NET/SWEETSEVEN_/ARTICLE/DETAILS/96026706

五、數學時間

- 神經元的數量？
  - 看輸出的數量？
  - 看參數的數量？
  - 看KERNEL的數量？
  - 看KERNAL掃過的數量？



五、數學時間

- 為何CNN模型常有兩個全連接層？
- CNN第一次的FILTER在做什麼？那第二次以後呢？
- 每個神經元的激勵函數（ACTIVATION FUNCTION）在幹嘛？

五、數學時間

- 獻給喜歡數學的你❤️
- HTTPS://CHIH-SHENG-HUANG821.MEDIUM.COM/機器學習-神經網路-多層感知機-MULTILAYER-PERCEPTRON-MLP-含詳細推導-EE4F3D5D1B41
- HTTPS://CHIH-SHENG-HUANG821.MEDIUM.COM/機器-統計學習-羅吉斯回歸-LOGISTIC-REGRESSION-AFF7A830FB5D

五、數學時間

- 獻給考不上台大的我們
- HTTPS://YOUTU.BE/CXGBEKL66JC
- HTTPS://GITHUB.COM/SAKURA-GH/ML-NOTES

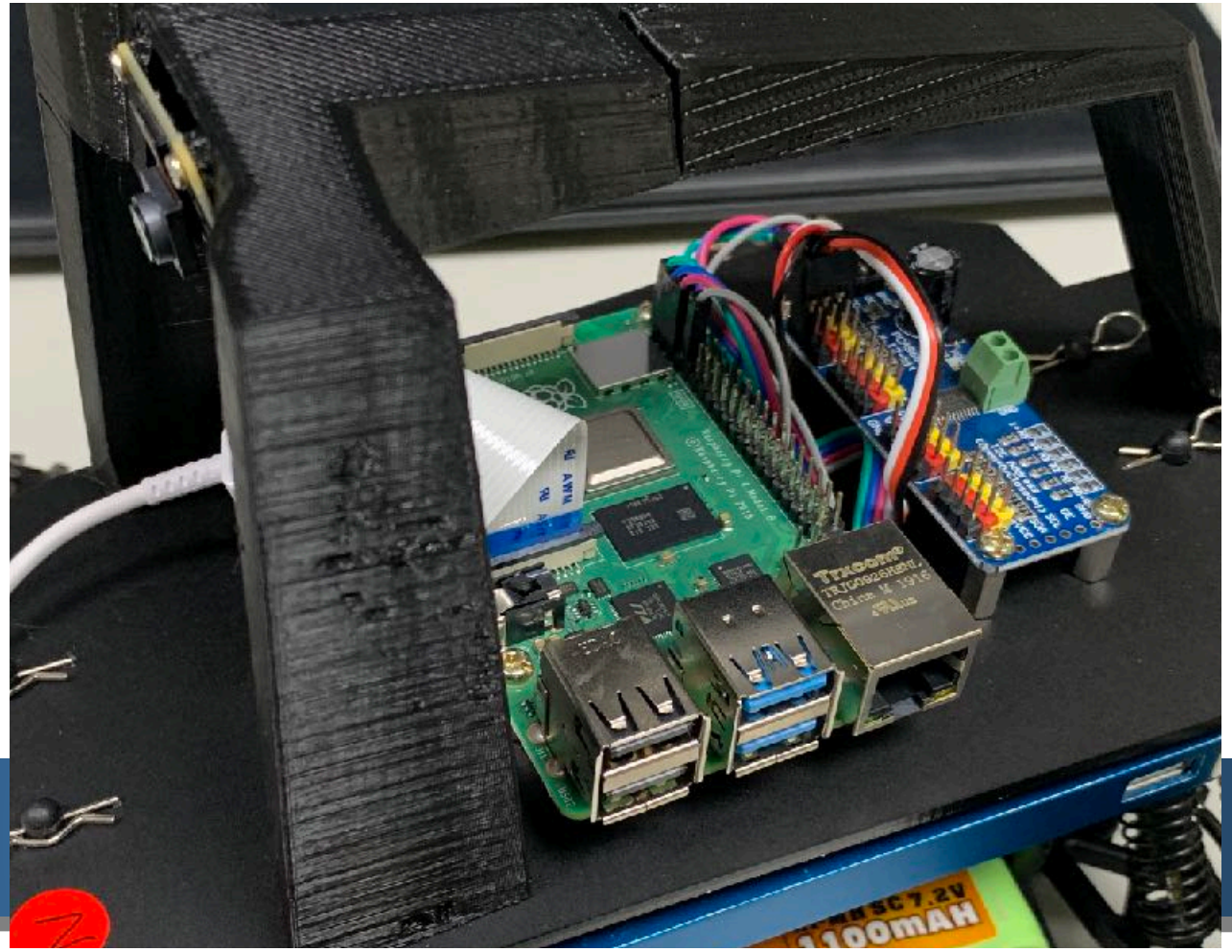
六、資源分享

- HTTPS://POLOCLUB.GITHUB.IO/CNN-EXPLAINER/
- HTTPS://WWW.CS.RYERSON.CA/~AHARLEY/VIS/CONV/

七、實體自駕車

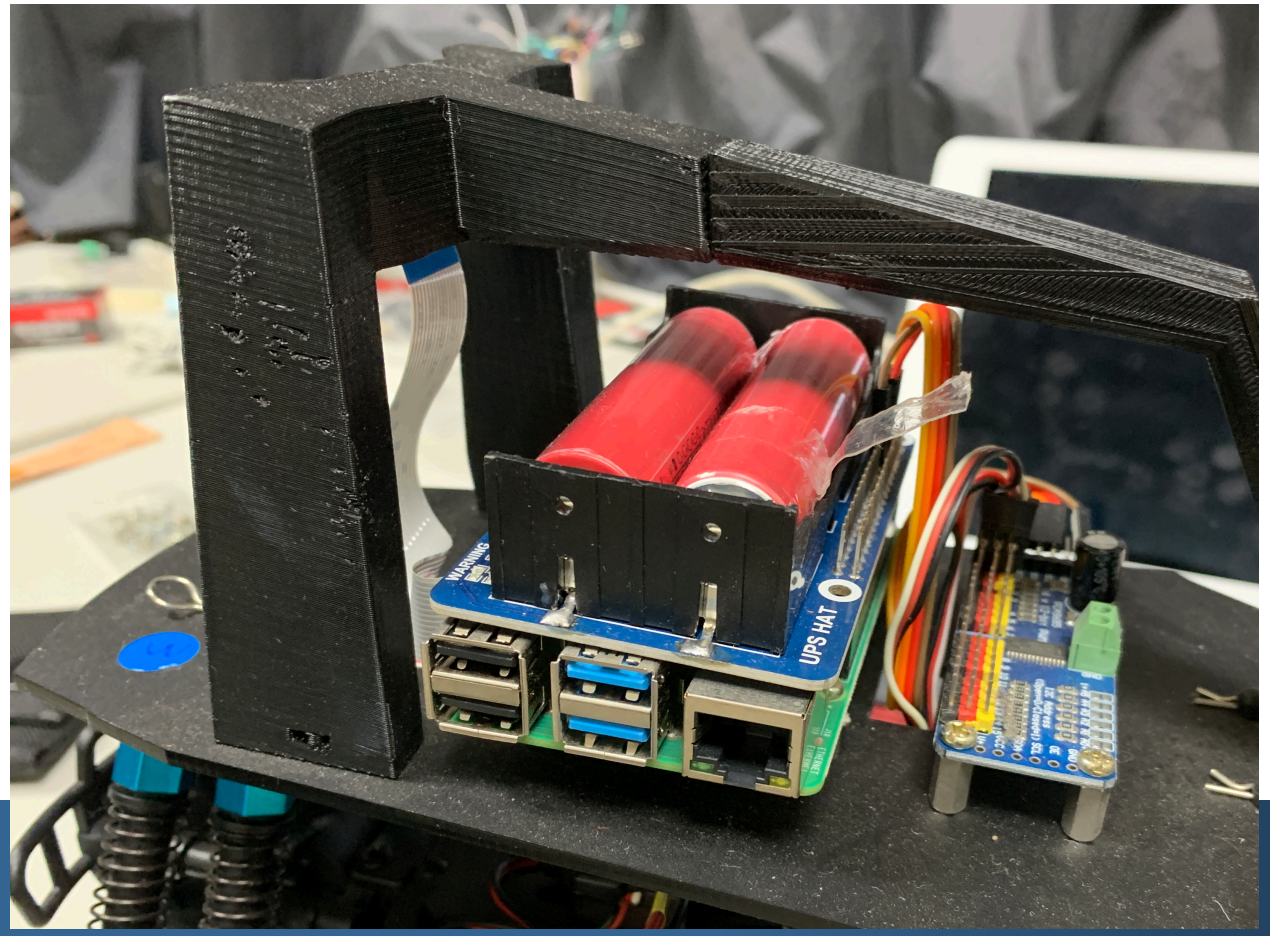
七、實體自駕車

- 核心：樹莓派



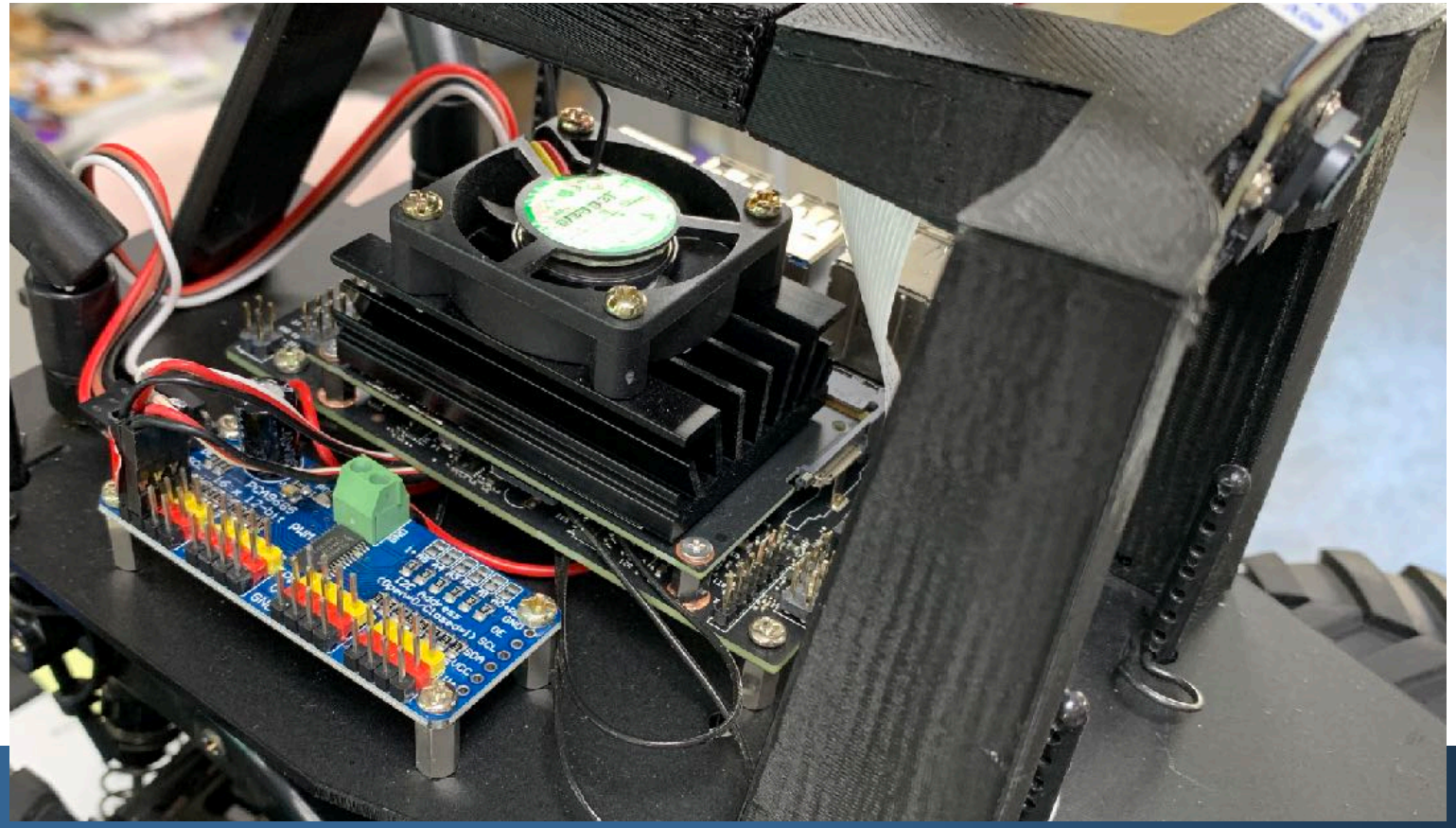
七、實體自駕車

- 核心：樹莓派＋電源



七、實體自駕車

- 核心：JETSONNANO

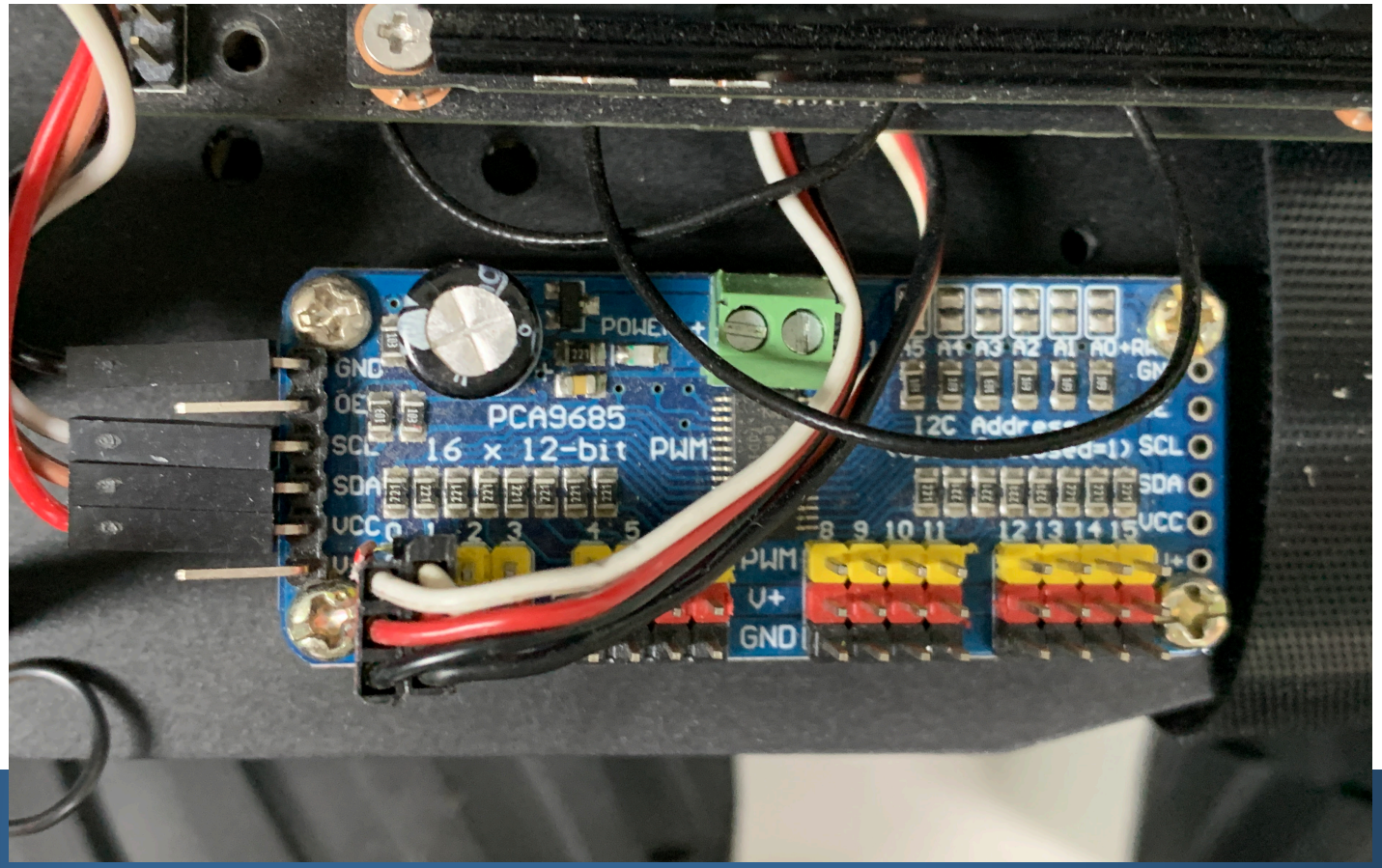


七、實體自駕車

- 碳刷馬達與電源

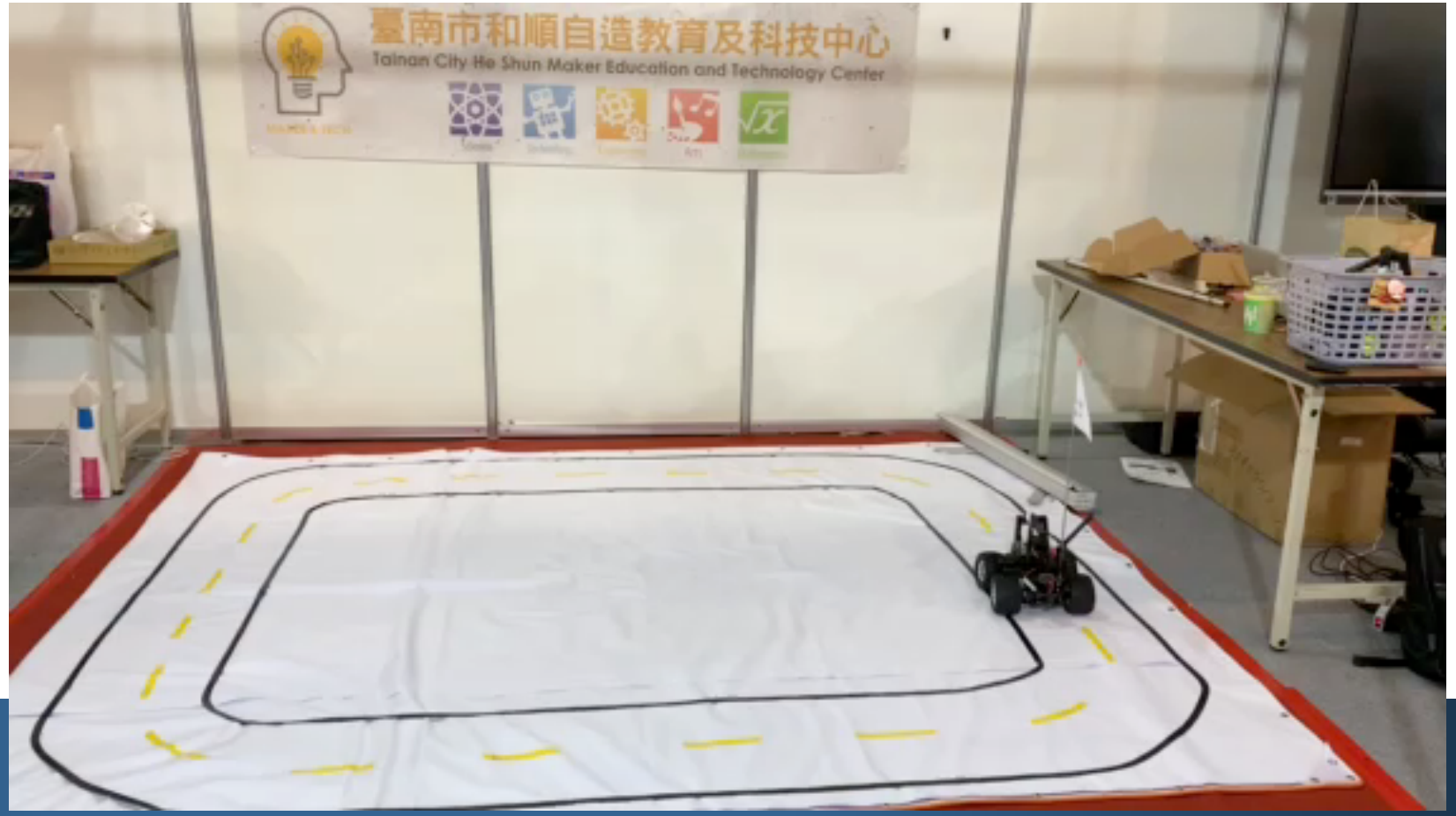
七、實體自駕車

- 馬達控制板：



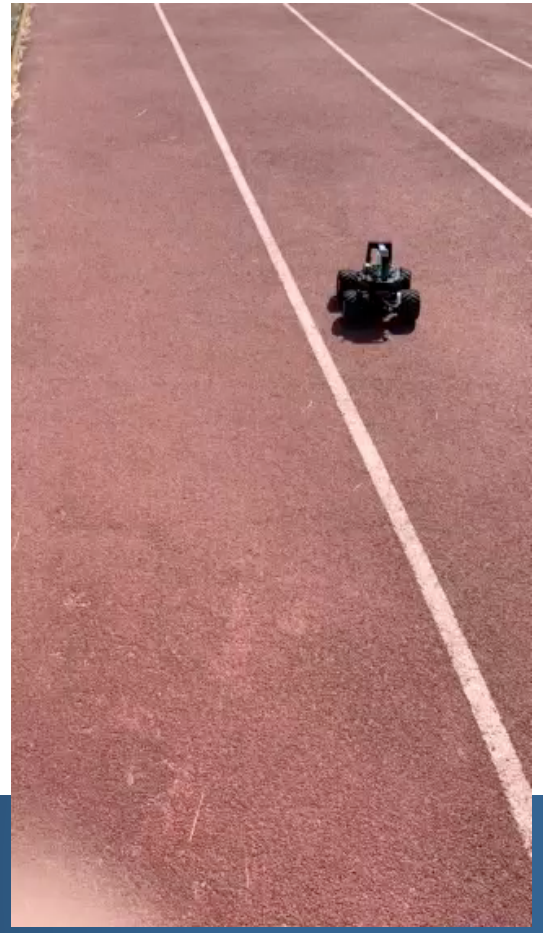
七、實體自駕車

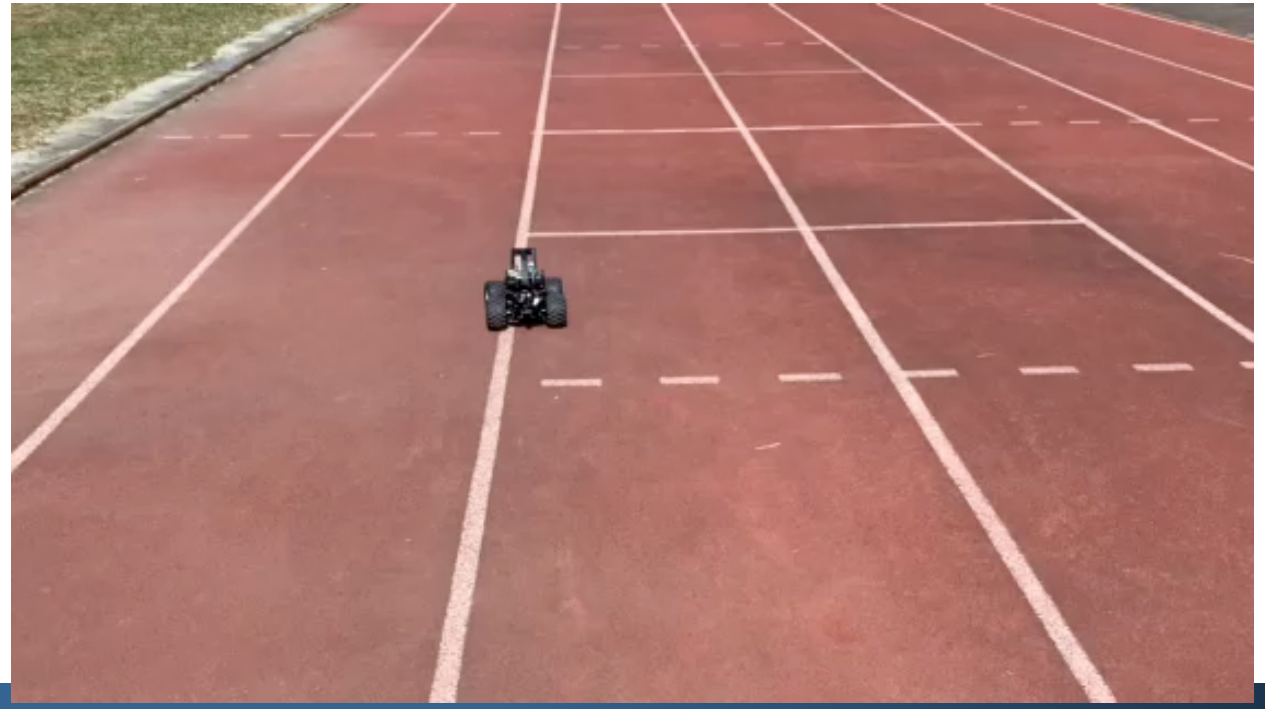
- 訓練結果：


臺南市和順自造教育及科技中心
Tainan City He Shun Maker Education and Technology Center

七、實體自駕車

- 訓練結果：





THANK YOU

OCEANBLUE0212@HSJH.TN.EDU.TW