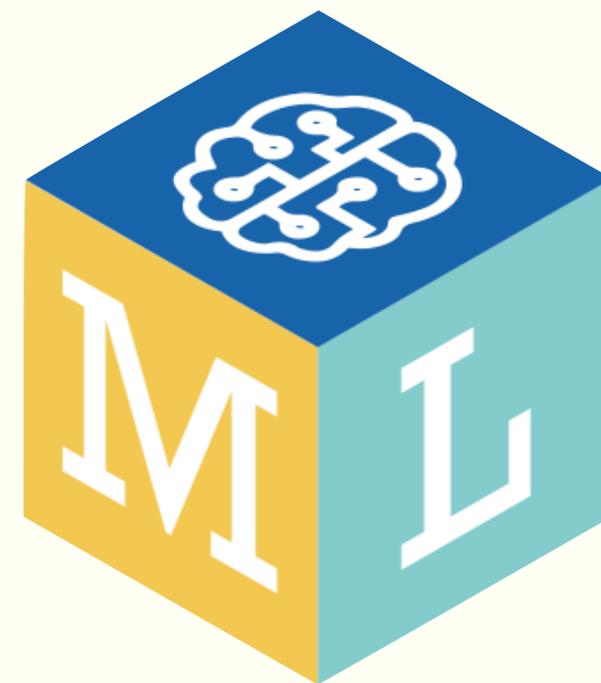


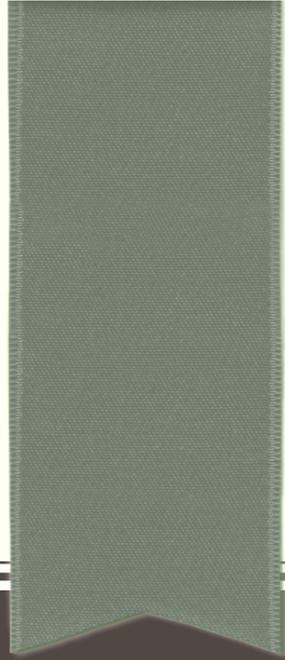
用SCRATCH玩AI

和順國中 林信廷



時間安排

- 人工智慧基本知識：1hr
- 初階AI體驗（預測水位）：1hr
- 高階AI體驗（井字遊戲）：1hr
 - 配合一個創客玩具
- 現場製作體驗（打磚塊）：3hr



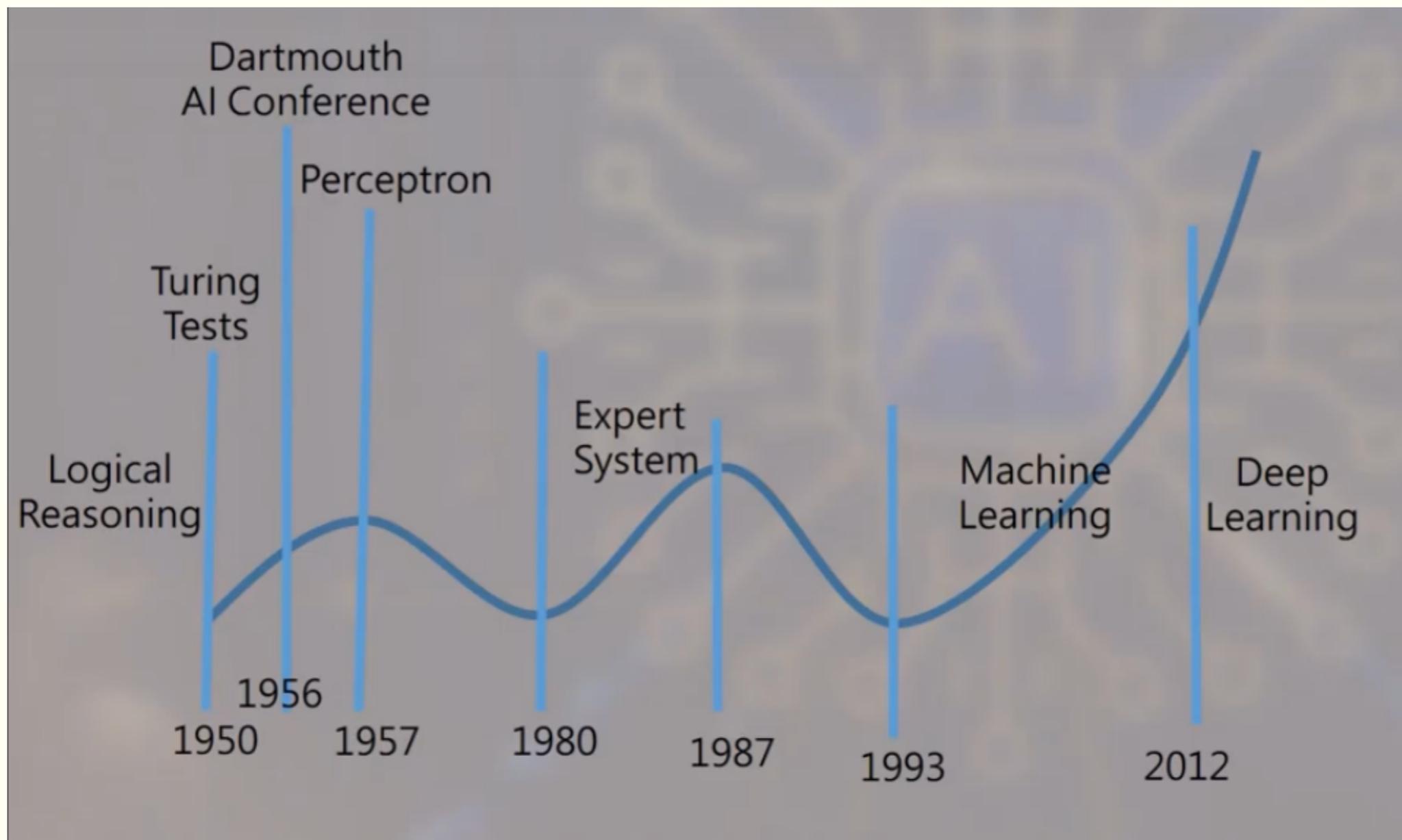
人工智慧基本知識

人工智慧基本知識

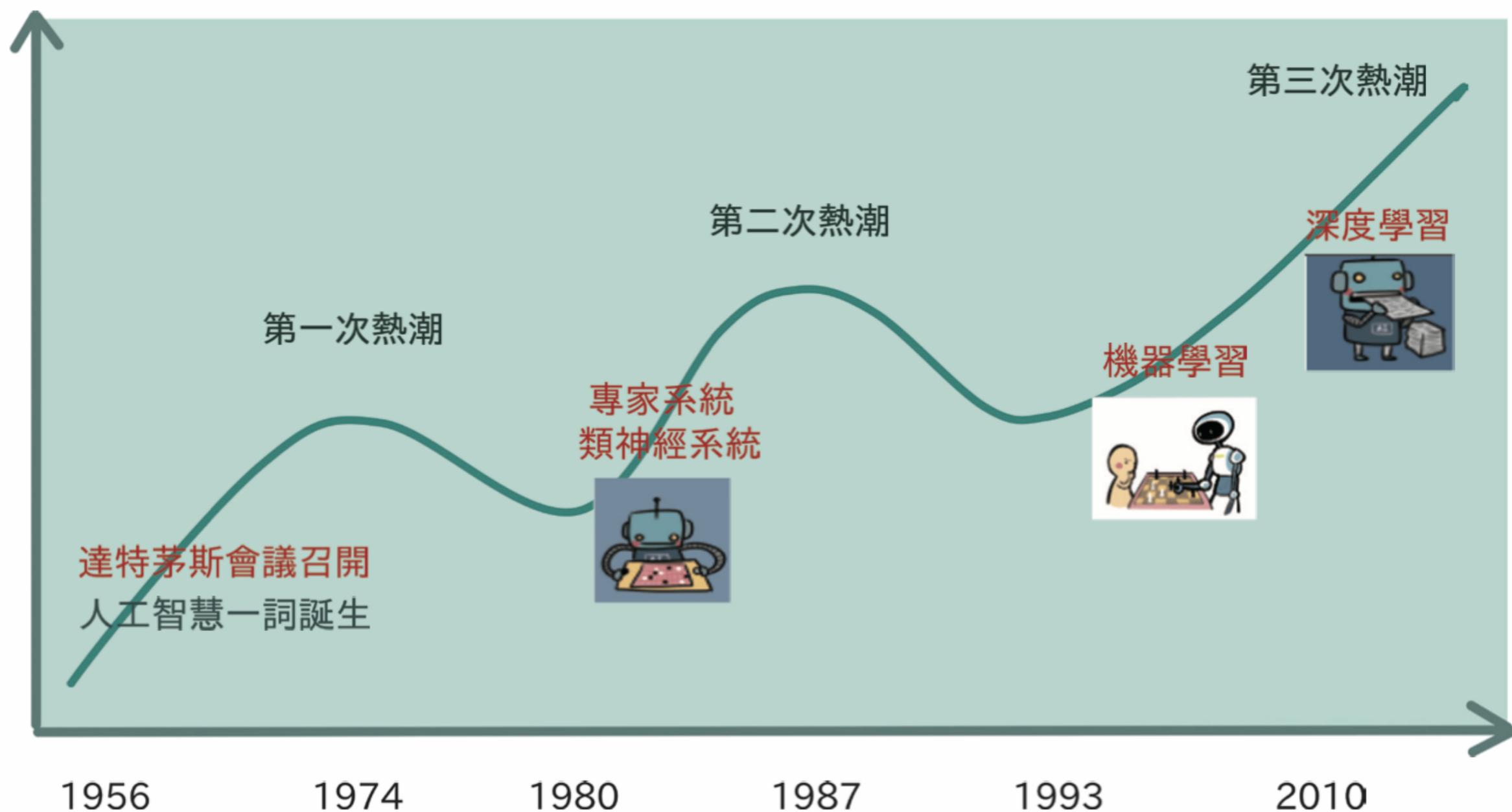
- 什麼是人工智慧？或者說，什麼是「智慧」？
 - 會思考？→那什麼叫做「會思考」？
 - 你每天都在思考什麼？→要吃什麼？要去哪裡玩？朋友為什麼不理我？
- 看來，智慧就是根據過去的經驗，做出判斷
- 所以簡單來說，人工智慧就是電腦根據我們提供的資料集，來做判斷

人工智慧基本知識

■ 人工智慧發展歷程



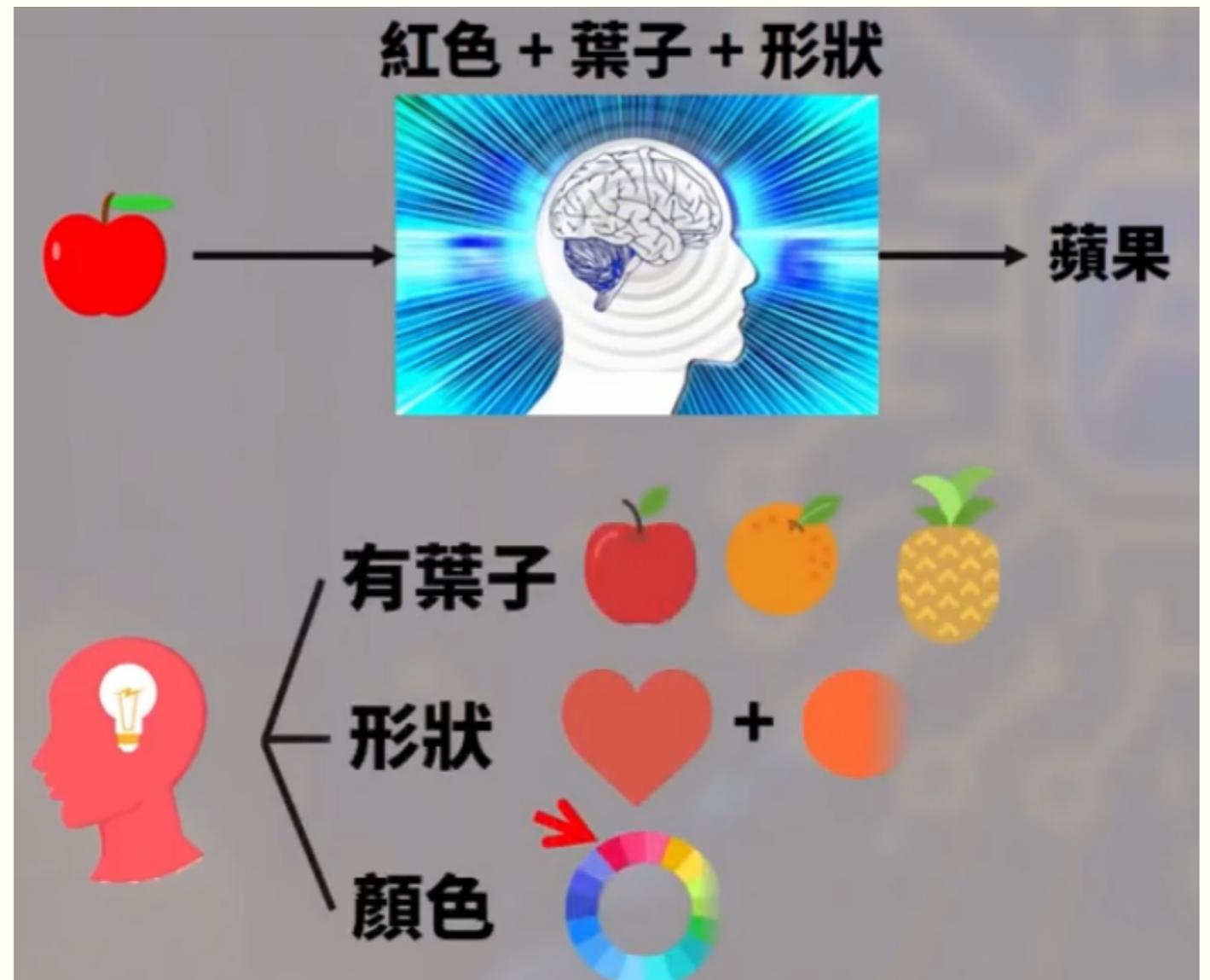
人工智慧基本知識



人工智慧基本知識

- 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

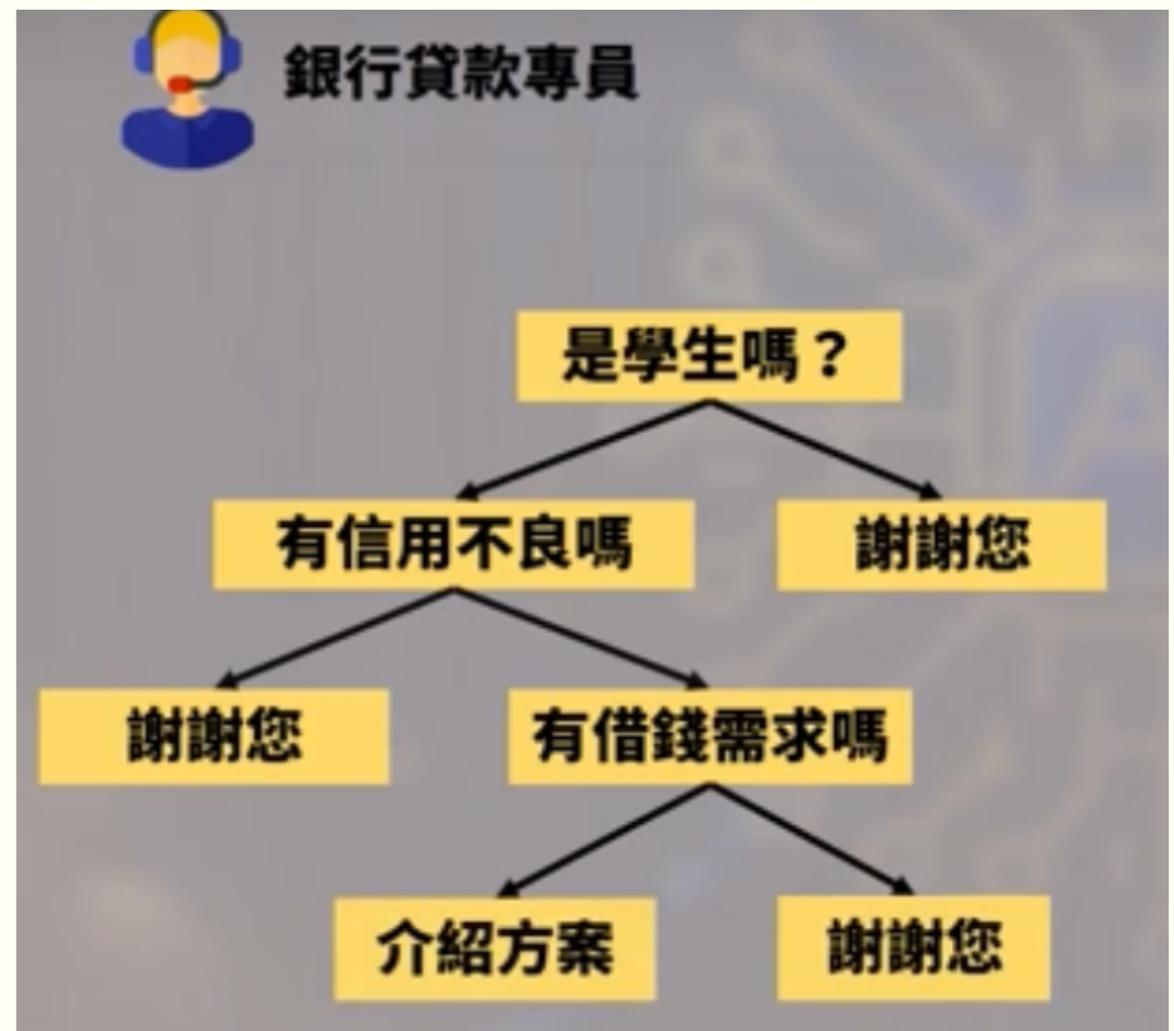


人工智慧基本知識

- 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。

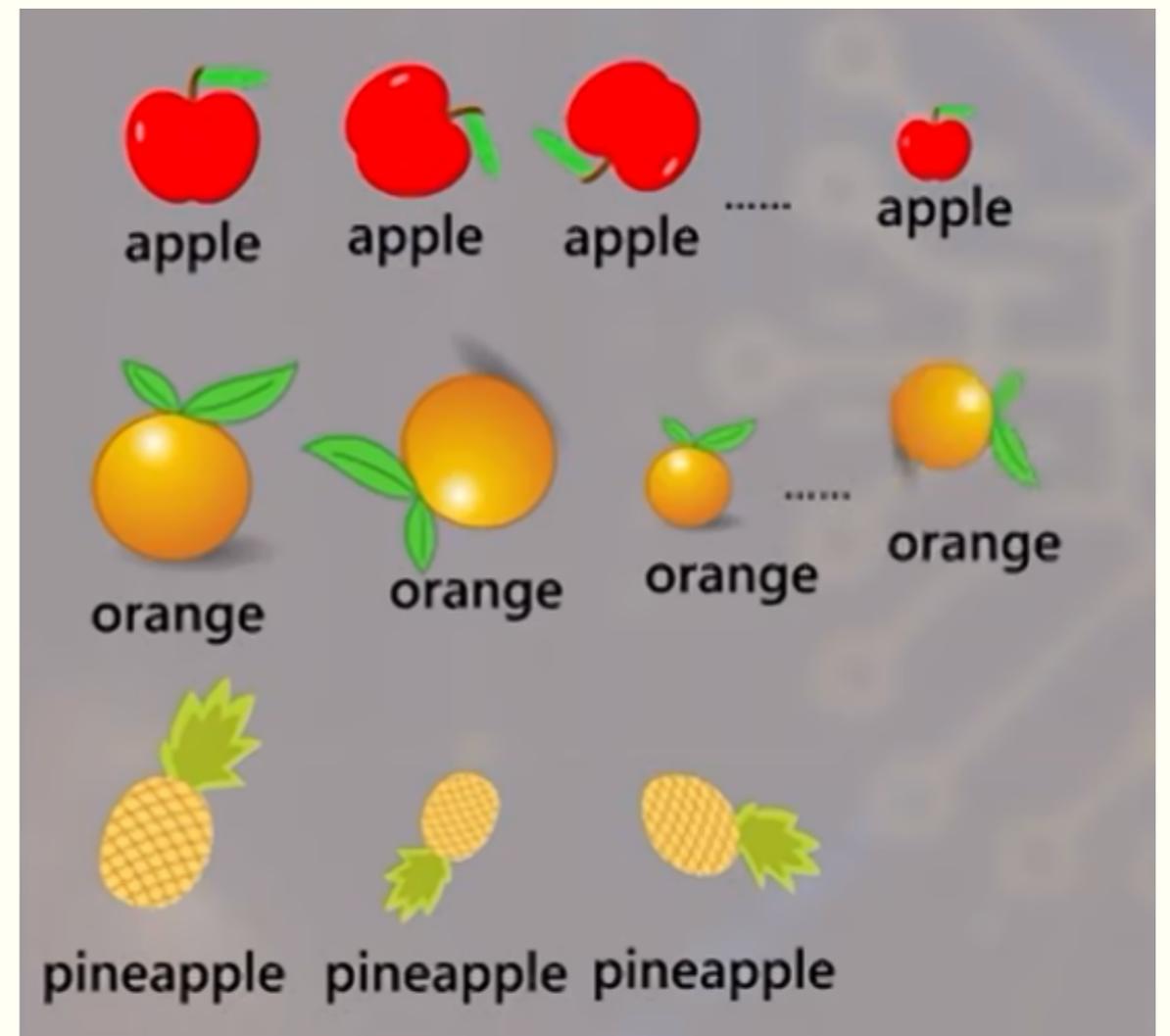
人工智慧基本知識

- 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



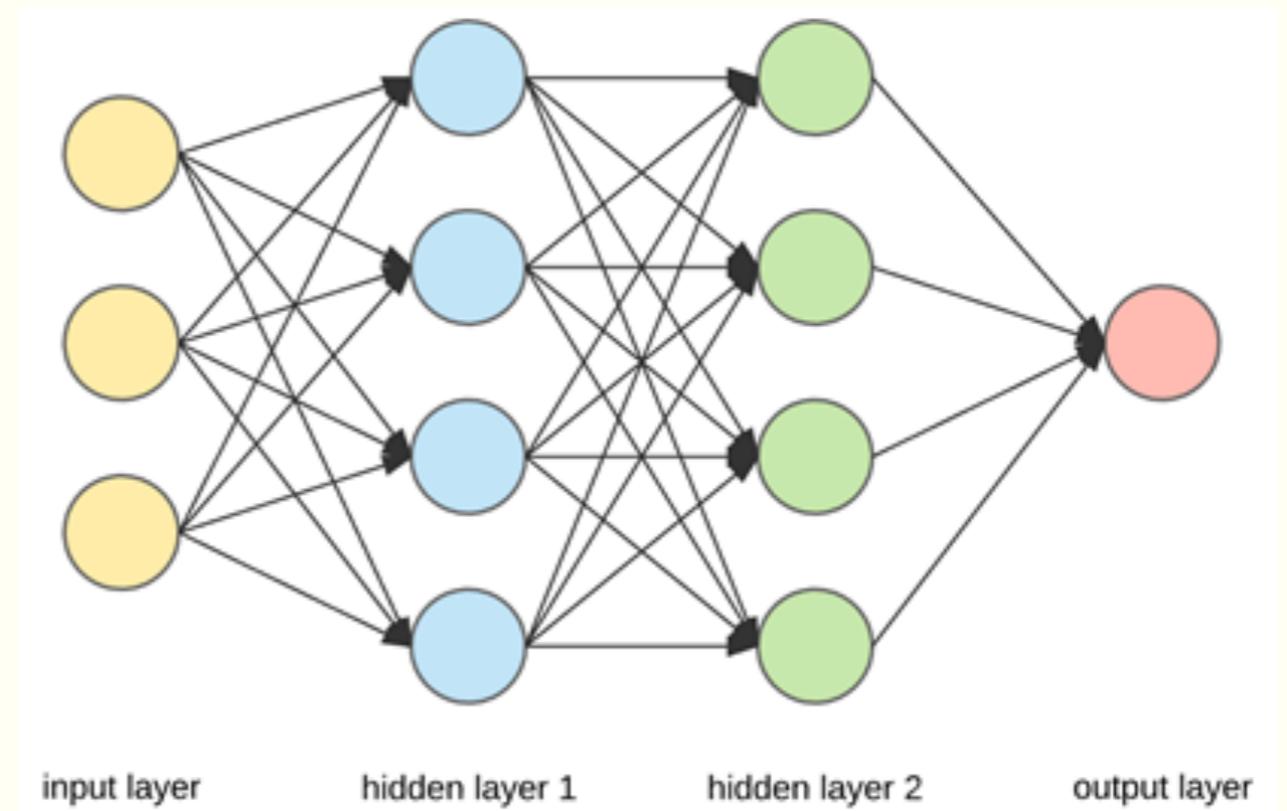
人工智慧基本知識

- 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
 - 學習什麼？
 - 學習如何分類
 - 找特徵
 - 從資料集比對



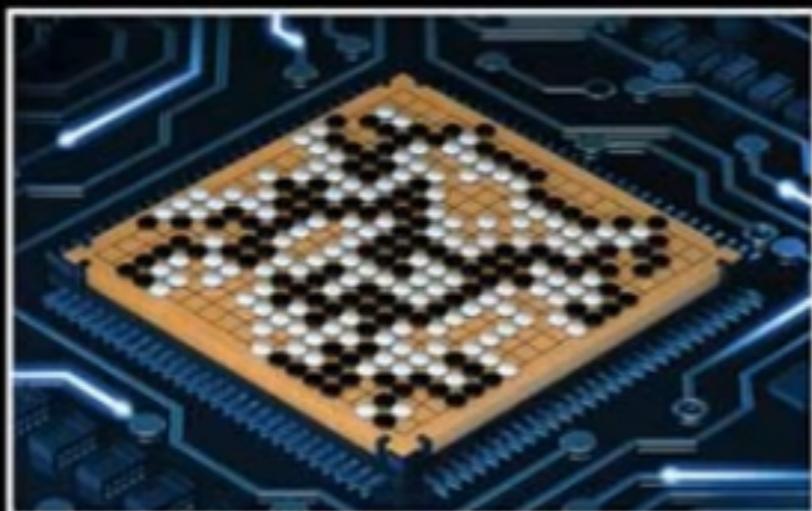
人工智慧基本知識

- 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



人工智慧基本知識

Deep Learning研究生



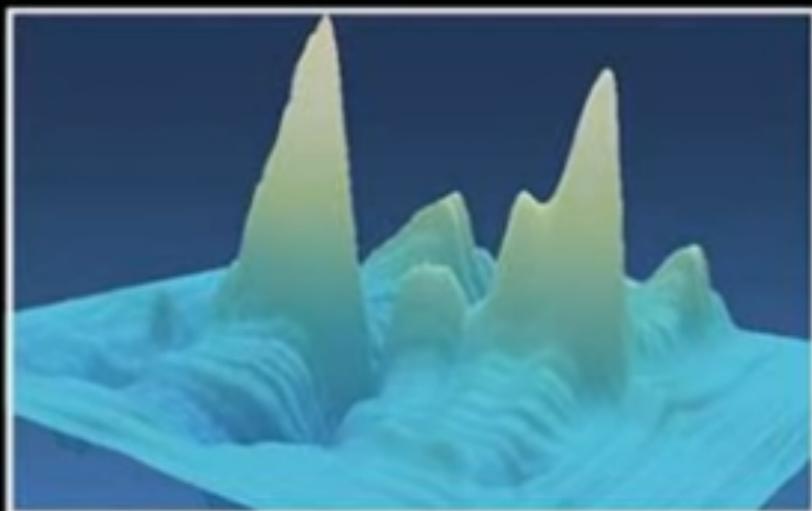
朋友覺得我在



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

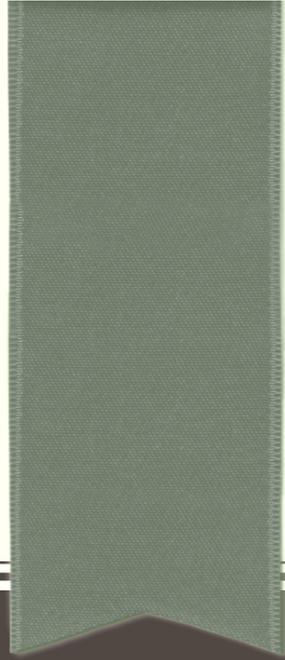


事實上我在

<https://blog.csdn.net/gqainrui>
Created with EverCam.

人工智慧基本知識

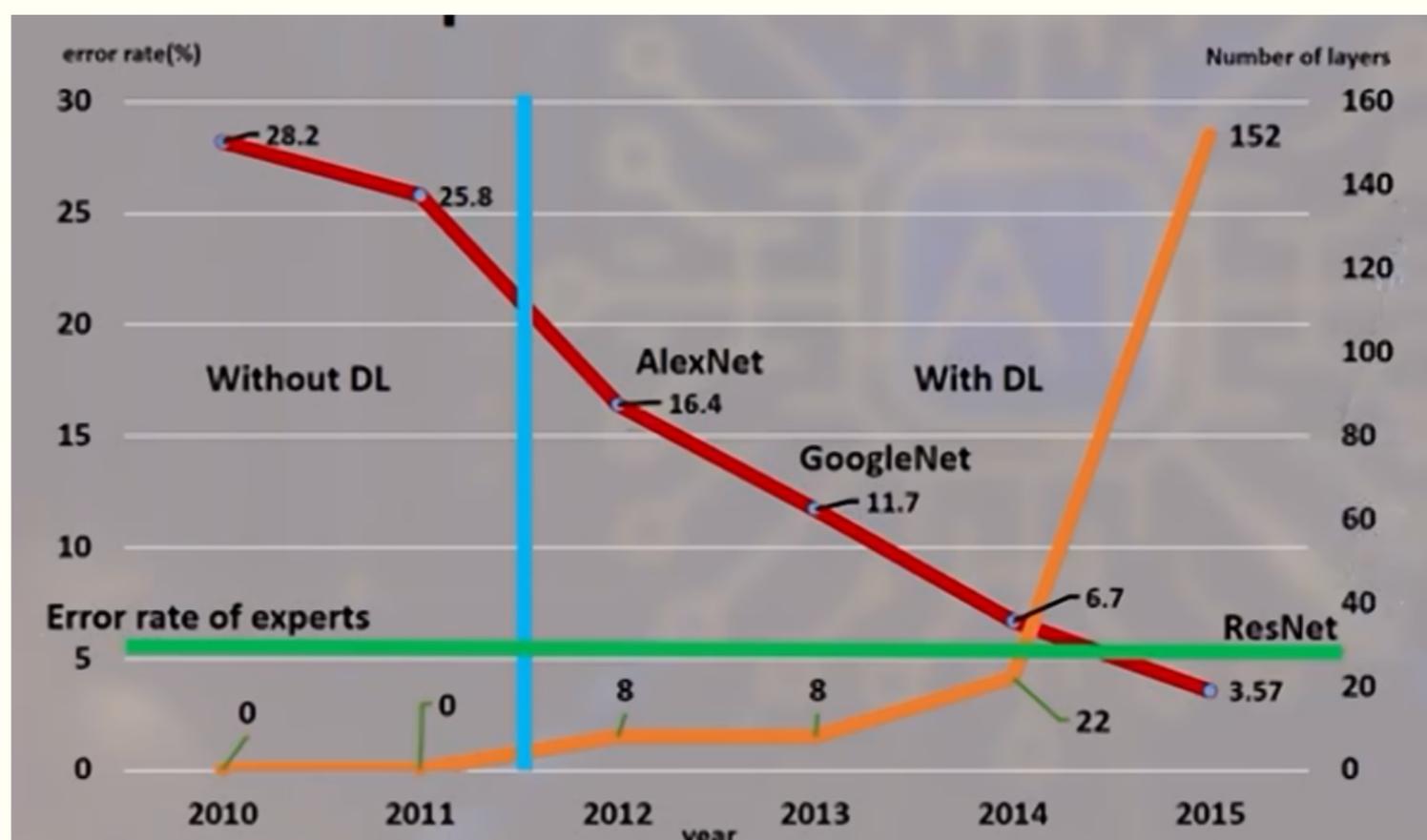
- 人工智慧常用於分類資料，那是因為先分類好了，才能做出判斷
 - 數值的、影像的、聲音的、文字的資料，就要用不同的技術處理
- 專家系統與機器學習的差異是：
 - 專家系統是人類直接告訴機器「特徵」，例如看到水果要先看顏色、葉子、形狀等等特徵來判斷
 - 機器學習是人類給電腦很多水果，電腦自己學著去找出特徵來分類資料。



人工智慧的應用

人工智慧的應用

- 圖片辨識比賽：
(ILSVRC)
 - 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
 - 2017年為最後一屆



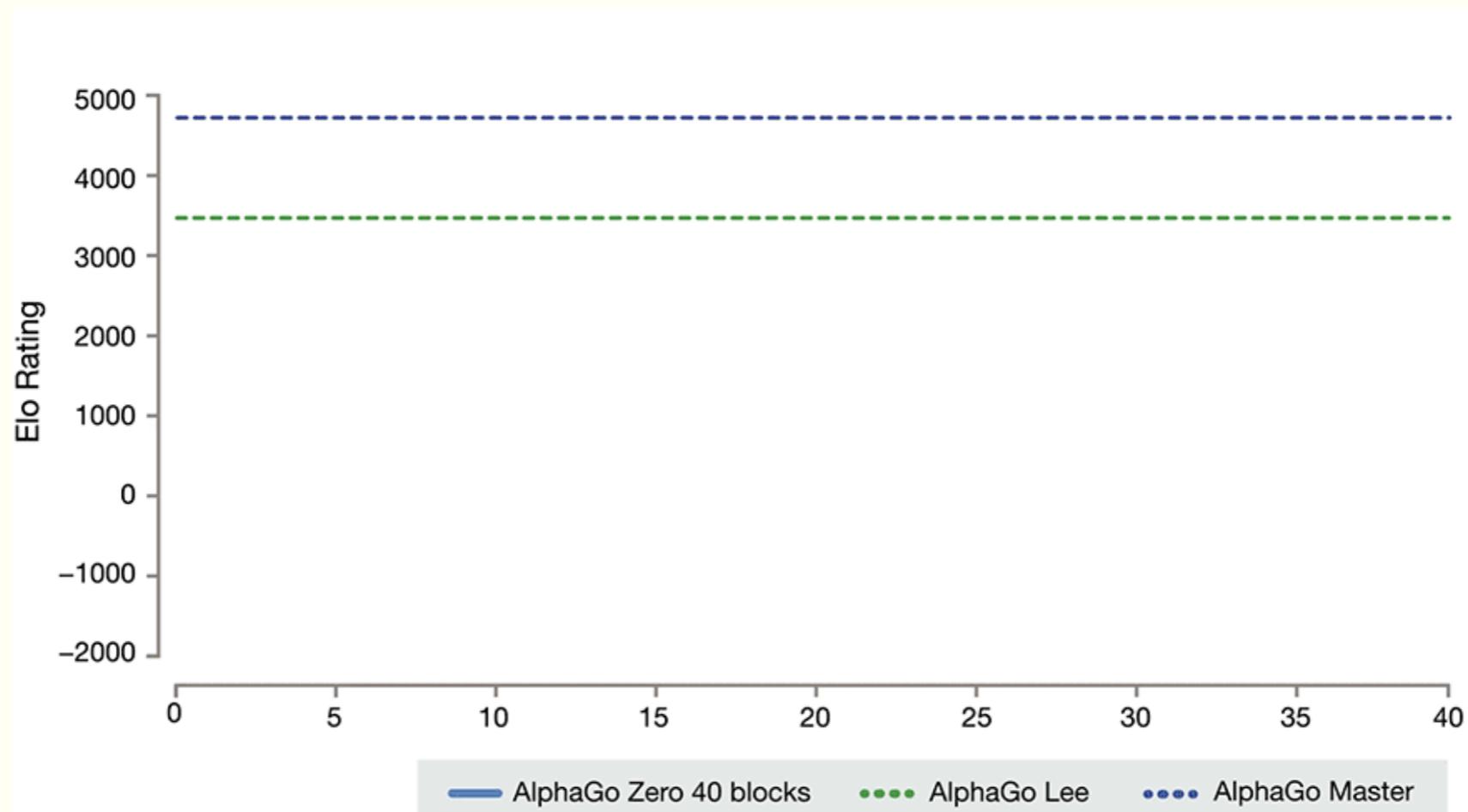
人工智慧的應用

- AlphaGO
 - 使用監督式學習與增強式學習

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戰勝李世石
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)

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>



人工智慧的應用

- Tesla



人工智慧的應用

■ 線上遊戲



人工智慧的應用

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



人工智慧的應用

- 影像辨識

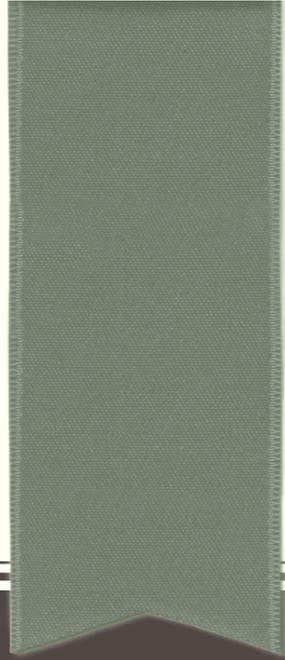
- 醫療
- 植物
- 車牌

- 創作

- 詩集：少女詩人小冰：<http://poem.msxiaobing.com/>
- 畫作

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

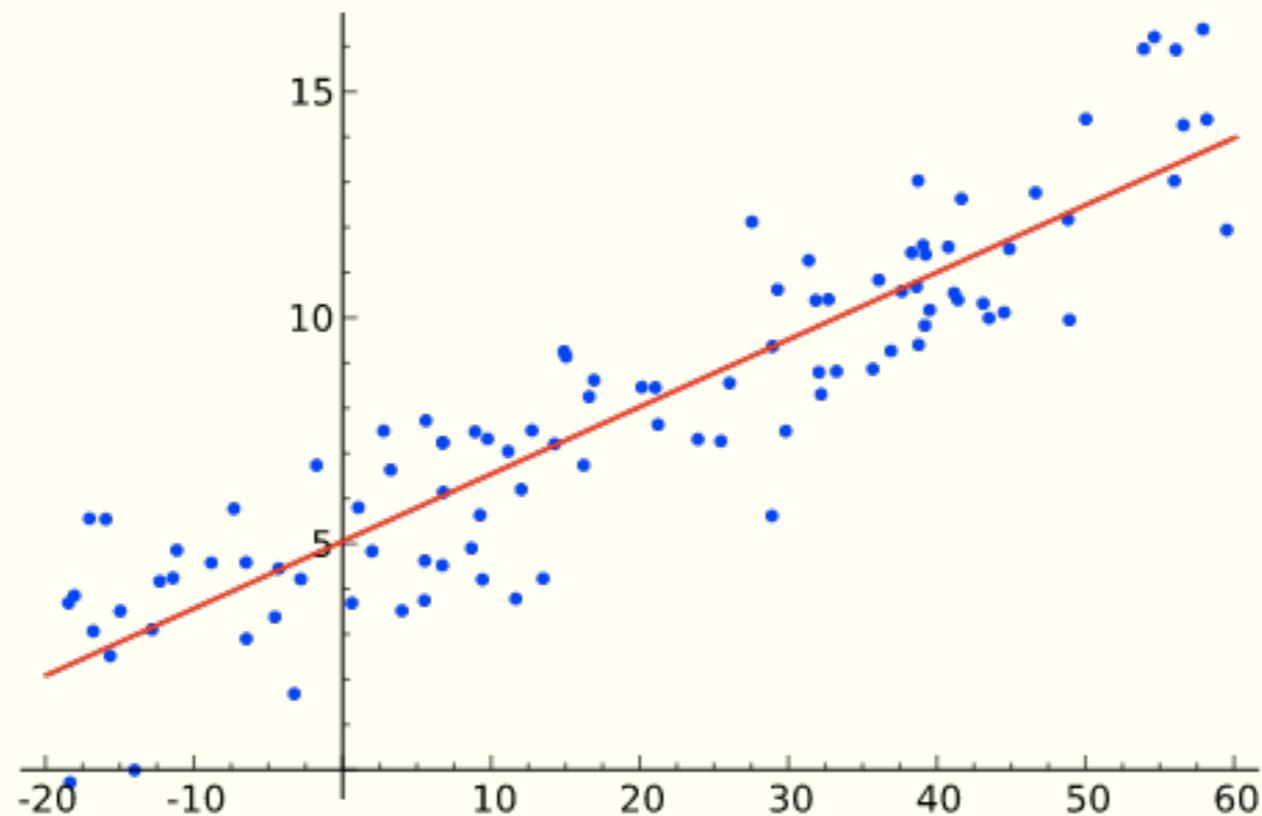


人工智慧相關技術說明

人工智慧相關技術說明

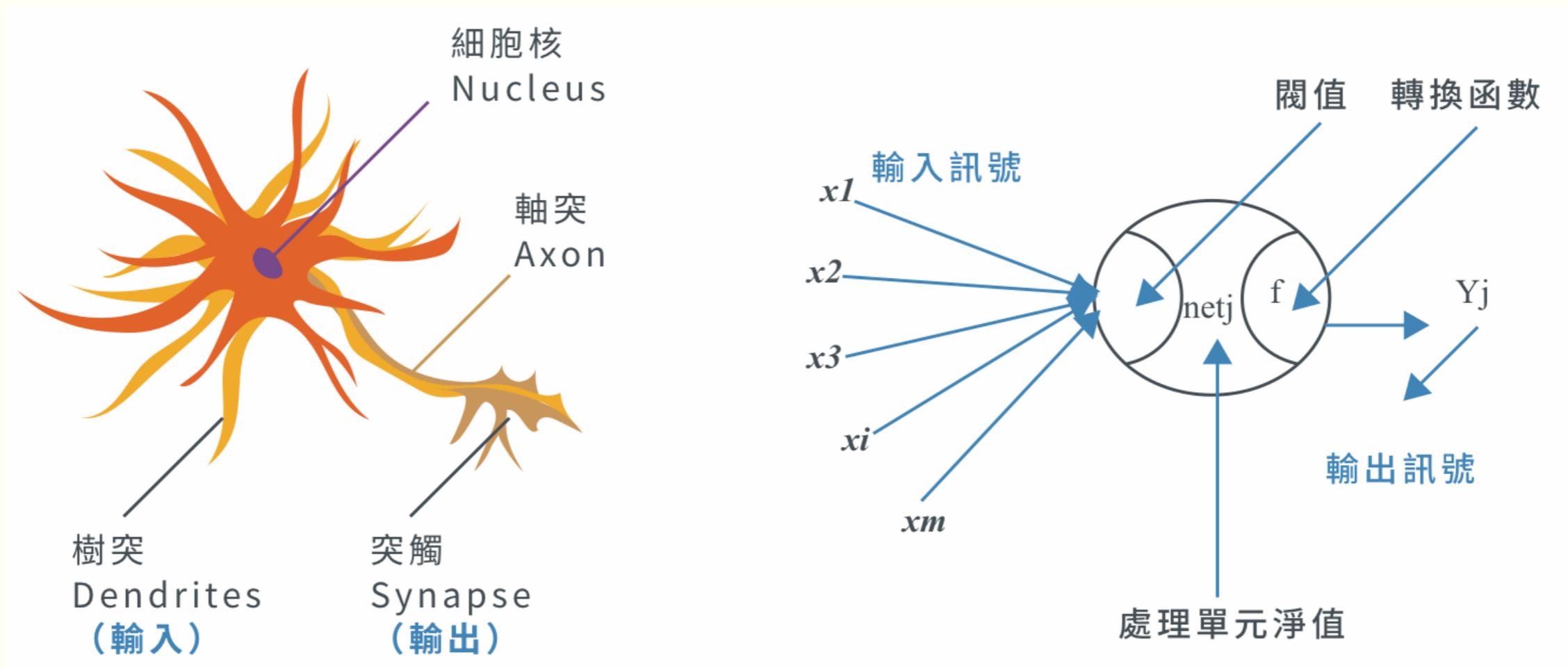
■ 線性迴歸

- 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



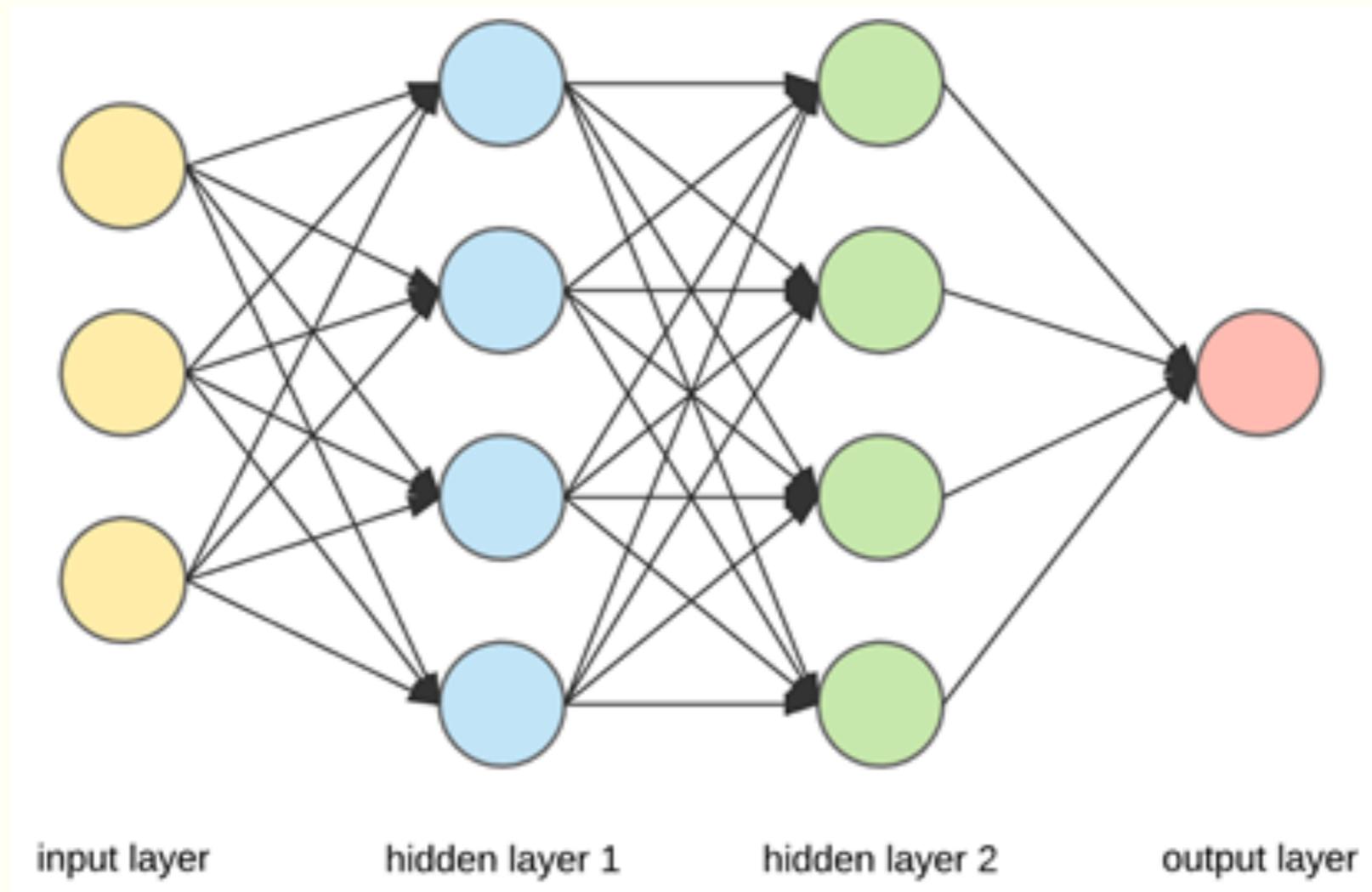
人工智慧相關技術說明

- Artificial Neural : 人工神經
 - 其實就是以前的感知器，換個名字重新出發



人工智慧相關技術說明

- Artificial Neural Networks：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

人工智慧相關技術說明

- 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

人工智慧相關技術說明

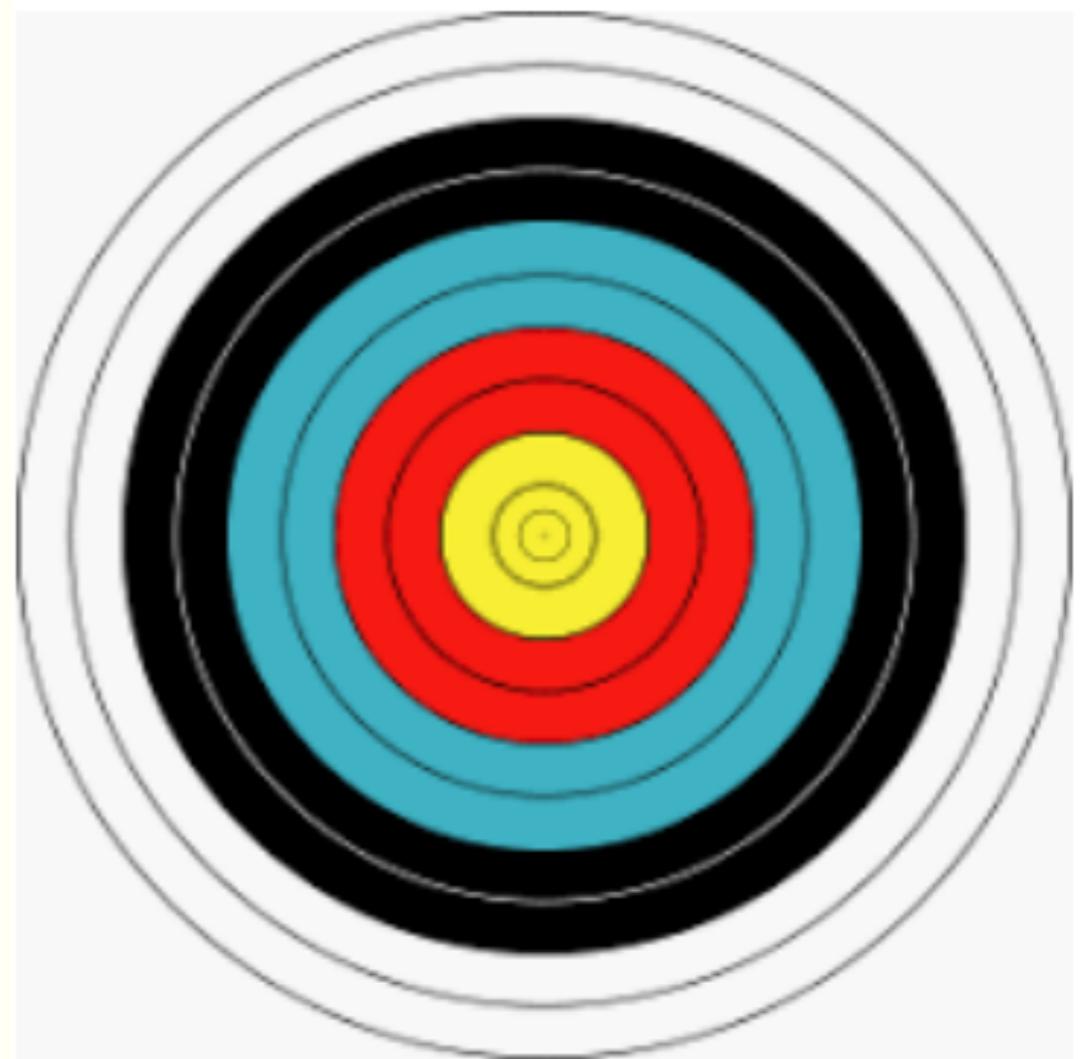
- machine learning：機器學習
 - 監督式：給電腦的資料是有標籤的
 - 非監督式：給電腦的資料是沒有標籤的
 - 增強式：電腦藉由跟環境的互動來學習
 - 電腦知道結果是好的，但不知道是哪一步造成優勢
 - 半監督式
 - 互動式
 - 轉換

人工智慧相關技術說明

- 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**！

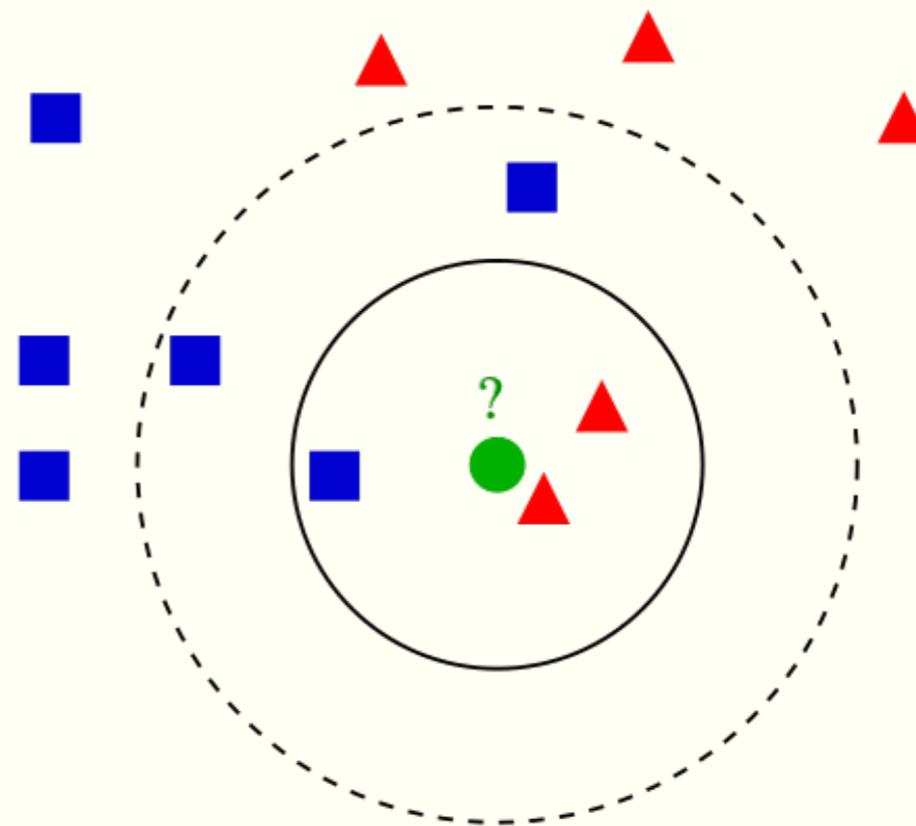
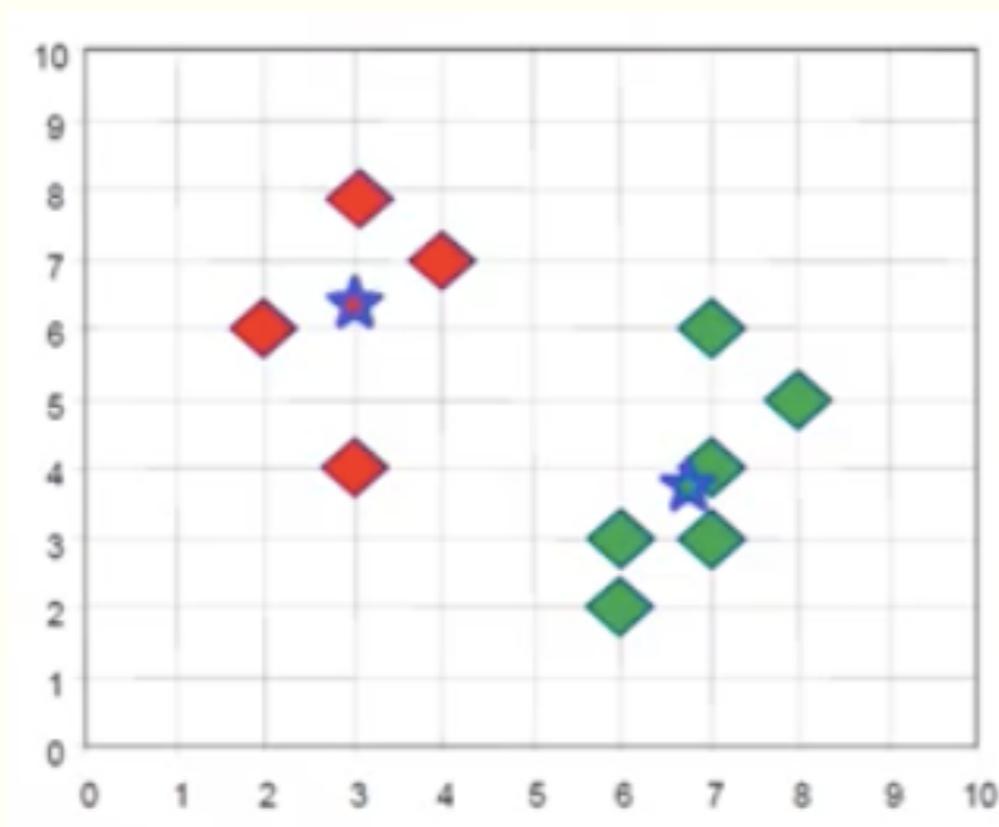
人工智慧相關技術說明

- 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



人工智慧相關技術說明

- 監督式學習
 - 最短距離分類器
 - KNN (k nearest neighbor) 分類器
 - 決策樹



人工智慧相關技術說明

■ 非監督式學習

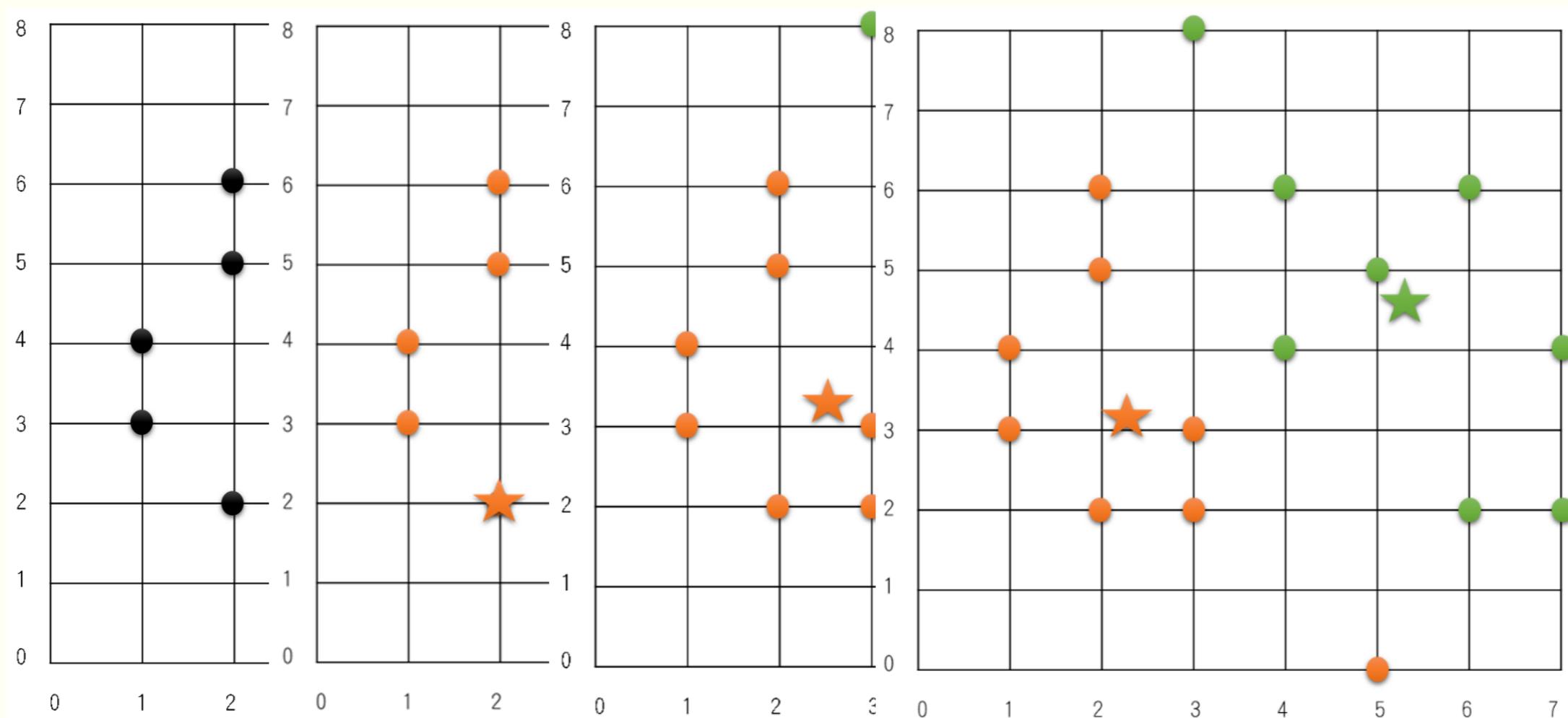
- 「**沒有老師指導**」的學習方法，在訓練階段無提供訓練樣本所對應的標記資訊
- 由於欠缺標記資訊，因此通常會就訓練樣本彼此之間的**相似性**進行**聚類分群**
- 避免掉監督式學習因為需要取得每個訓練樣本所對應的標記資訊所要付出的可觀**人力成本**

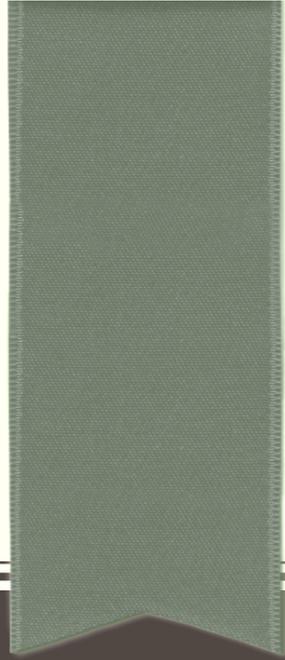
人工智慧相關技術說明

- 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





SCRATCH玩AI 基本説明

使用資源

- 網址：<https://machinelearningforkids.co.uk/>
- IBM軟體工程師建立的，看來是非官方的
- 安全性應該沒問題，因為密碼是網站給的，而且不能修改
- 不能用IE，開啟SCRATCH時會發生問題

基本說明

- 沒註冊也可以體驗
- 有註冊
 - 可以進行圖片的辨識
 - 可以建立三個專案
 - 專案可以長期保留
 - 專案可以分享給其他人

基本說明

- 網站設計為教學用途
 - 只有父母、老師或程式教育者可以申請帳號
 - 通過後，可以建立29個帳號給學生用（應該可以請他開多一點）
 - 每個帳號能建立三個專案
 - 帳號密碼也是網站配發，不能修改
- 再去IBM WATSON建立API，才可以訓練文字跟圖片
 - 免費版本可以有5個聲音模型，2個影像模型
 - 30天沒用會被刪除

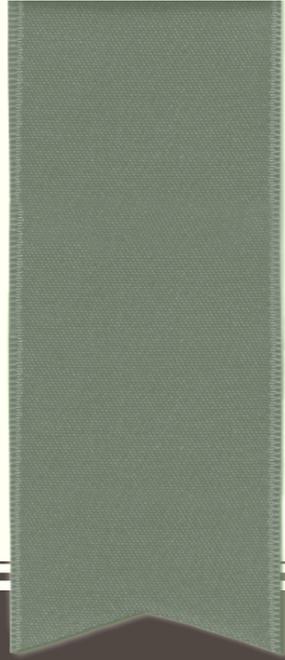
基本說明

- 老師的專案可以分享給學生，所以學生最多會有6個專案

project-all

使用 數字





課前操作說明

課前操作說明

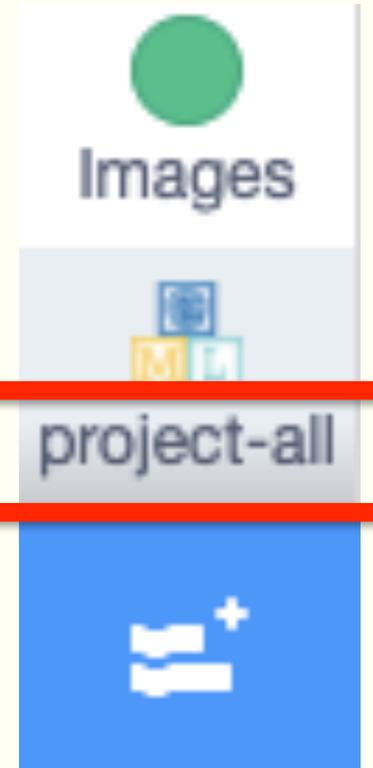
你的機器學習專案

+ 新增專案

使用範本

- 建立機器學習時，要先設定以下項目：

- 專案：訓練的主題、名稱
- value(值)：如果專案要辨識的是數字，就要馬上設定好想要辨識的數值的名稱。
 - 選項：如果要辨識的value是字串，就要用多選，建立每個選項
- 例如我想要辨識的是百分比(percent)的數值
- 或是想辨識的是在某個格子上有哪個東西
(多選)



課前操作說明

- 特別注意⚠
- 建立專案後，若匯出scratch檔案，刪除專案後，就算再重新製作一個名稱一模一樣的專案，也無法匯入
- 上課前備課時要注意

課前操作說明

- 建立機器學習時，要先設定以下項目：

全班的專案？

專案名稱*
Test

辨識*
數字

Value 1* Value1	數值的類型* 多選	Value 2* Value 2	數值的類型* 數字
--------------------	--------------	---------------------	--------------

Option
新增選項
Option2

If this field can be described as numbers, choose "number".
如果可以描述成從幾個選項中進行選擇，請選擇「多選」

ADD ANOTHER VALUE

建立 取消

課前操作說明

- 標籤：就是訓練與判斷的「結果」屬於哪一類
- 建立標籤：

建立資料模型

收集你希望電腦可以辨識的範例

建立資料模型

訓練 & 測試

使用資料模型訓練電腦學習辨識 numbers

訓練 & 測試

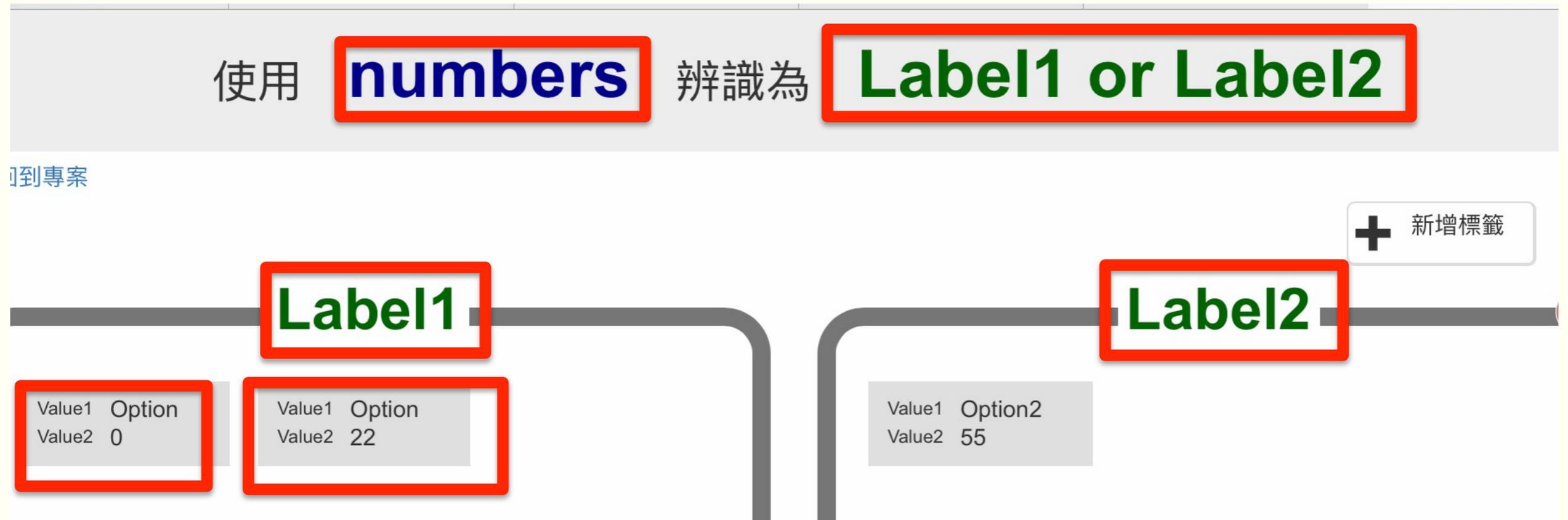
成果應用

在Scratch或Python中使用你訓練好的機器學習模型來製作遊戲或應用程式

成果應用

課前操作說明

- 欄位用途整理：

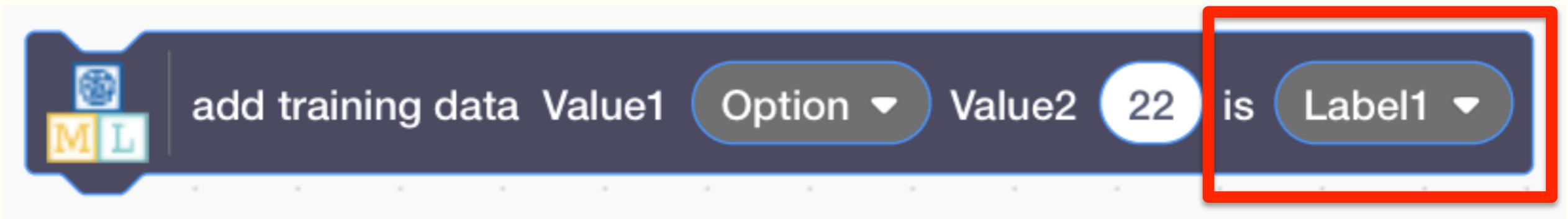


- 網站會拿value1跟value2的值來訓練其分類 (label)
- 網站會拿value1跟value2的值來分辨是哪一類(label)

課前操作說明

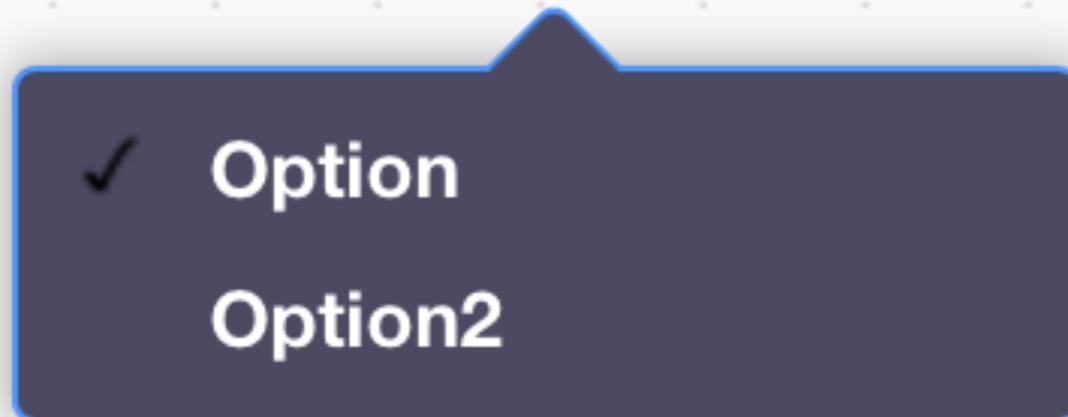
- 標籤(label)：

- 建好專案之後，要建立模型資料時，需要先建立標籤。
- scratch的積木會把資料丟進去每個標籤

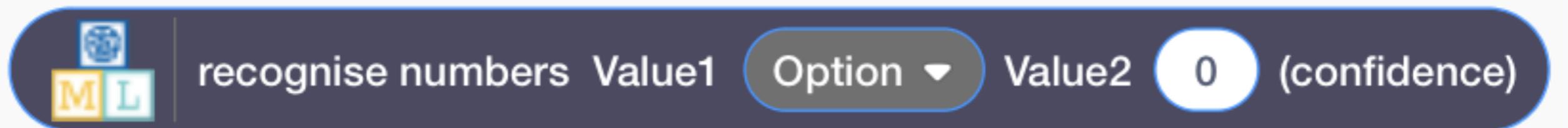


課前操作說明

- 學習後，可以撈出在某個組合下最有信心的label



- 也可以看到某個組合的信心分數 (0~100)



課前操作說明

- 專案畫面欄位說明

project-all

使用 **數字** 辨識為 **label1 or label2**

分享



課前操作說明

■ 範例：

prediction the water

使用 **數字** 辨識為 **half_empty or half_full**

- 專案名稱：水位預測（不能用中文）
- value名稱：percent，都只有一個值
- 分類（標籤(label)）：half_empty 跟 half_full

half_empty

percent 49

percent 51

percent 47

percent 40

+ 加入測試範例

24

half_full

percent 56

percent 64

percent 58

percent 64

+ 加入測試範例

18

課前操作說明

■ 範例：

Noughts and Crosses

使用 **數字** 辨識為 **top_left, top_middle or 7 other classes**

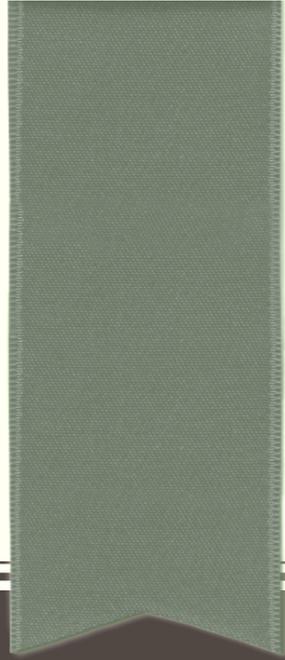
- 專案名稱：井字遊戲（不能用中文）
- value名稱：top_left等九個位置，每個都有三個選項
- 分類（標籤(label)）：top_left等九個位置

top_left

TopLeft	EMPTY
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	EMPTY
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY

top_middle

TopLeft	PLAYER
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	OPPONENT
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY



數學時間：決策樹

資料來源：中華開放教育平台

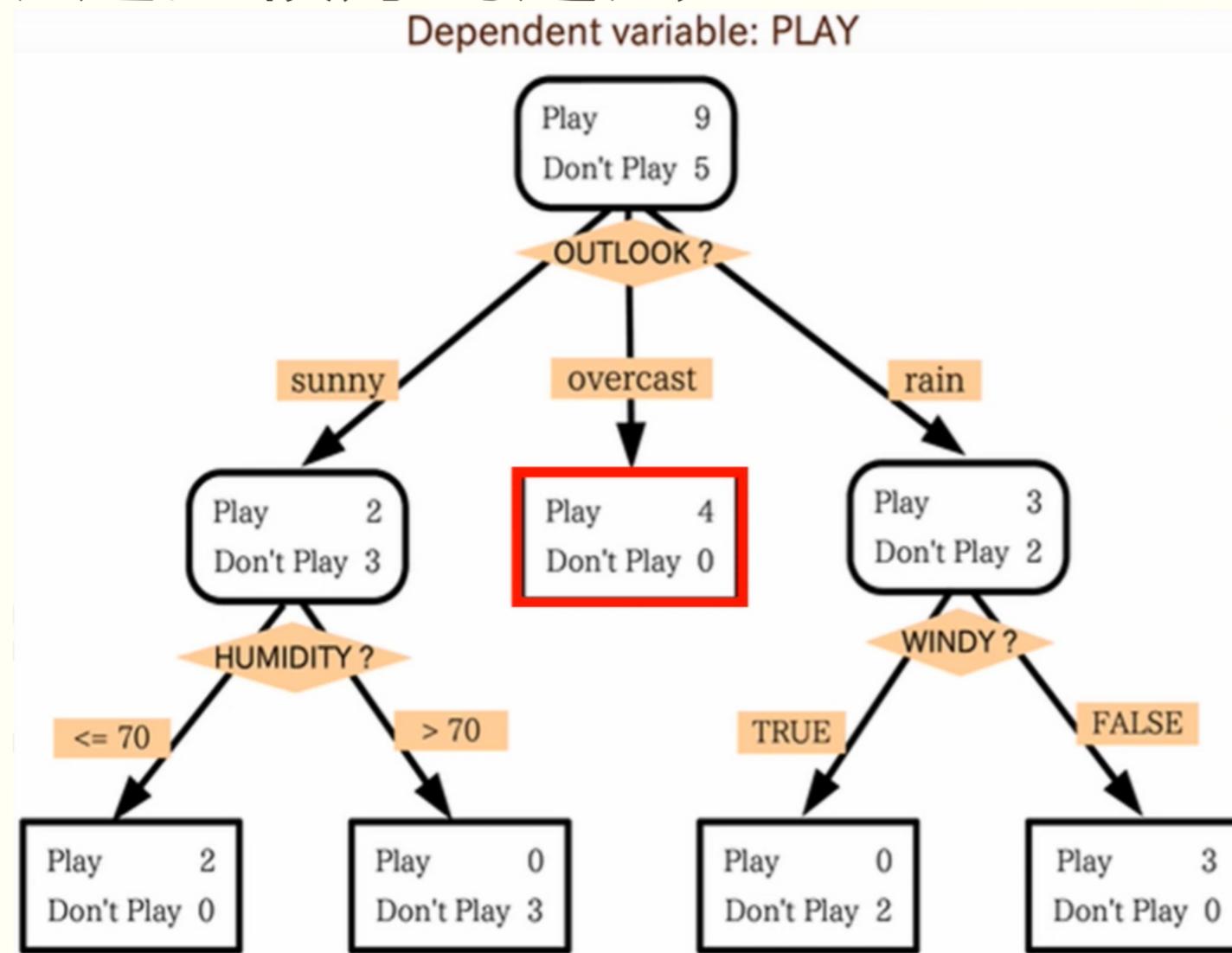
數學時間：決策樹

- 一個好的決策樹能夠幫助你更快地做出決定
- 假設一個打球的紀錄如下：

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

數學時間：決策樹

- 這是其中一種走法：先看天氣
- 這個可以快速地幫我把其中一類要打球跟不打球分開，所以是比較好的走法



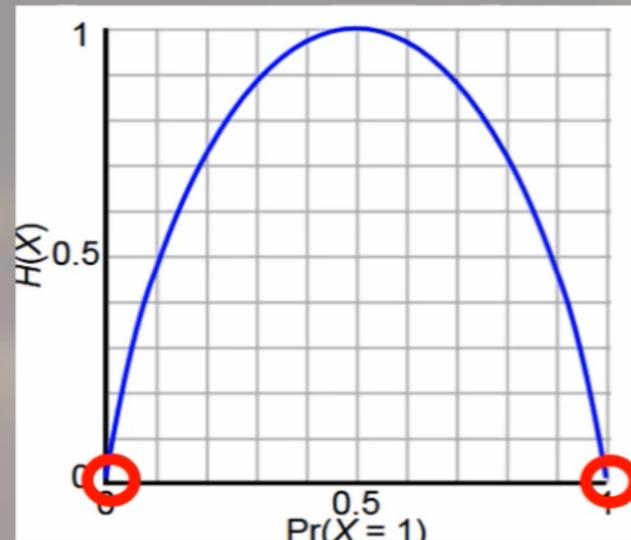
數學時間：決策樹

- 當某個值出現的機率是1或是0時，entropy(亂度)為0，代表資料很整齊

Entropy

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

$$H(x) = 0 \text{ if } P(x) = 0$$



數學時間：決策樹

- 我們可以計算某個資料集原本的亂度，再計算經過某種分類之後的亂度
- 比較這兩個亂度，就能夠知道這樣的分類有多少的幫助
- T資料集經過a特徵分類後的亂度

$$H(T|a) = \sum_{v \in \text{vals}(a)} \frac{|S_a(v)|}{|T|} \cdot H(S_a(v))$$

Overcast

Outlook

數學時間：決策樹

- 原始資料集的亂度

Play	9
Don't Play	5

$$H(T) = - \sum_{i=1}^n P(t_i) \log_b P(t_i)$$

$$H(T) = - \frac{9}{9+5} \log \frac{9}{9+5} - \frac{5}{9+5} \log \frac{5}{9+5} = 0.940$$

數學時間：決策樹

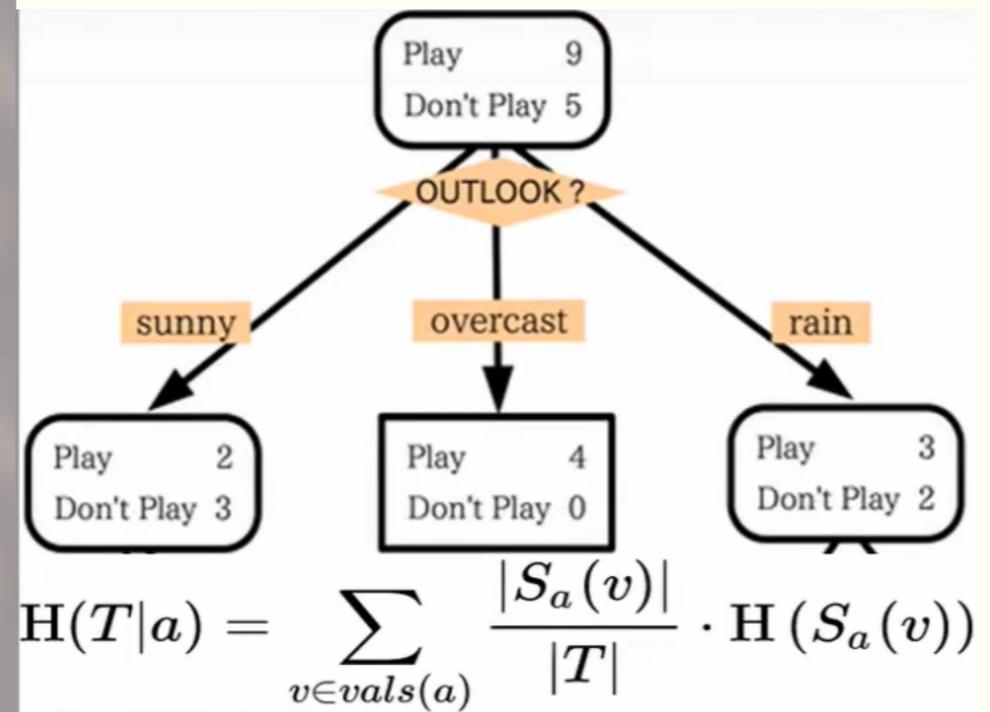
- 用天氣作為分類後的亂度

$$H(\text{sunny}) = -\frac{2}{2+3} \log \frac{2}{2+3} - \frac{3}{2+3} \log \frac{3}{2+3} = 0.971$$

$$H(\text{overcast}) = -\frac{4}{4+0} \log \frac{4}{4+0} - 0 = 0$$

$$H(\text{rain}) = -\frac{3}{3+2} \log \frac{3}{3+2} - \frac{2}{3+2} \log \frac{2}{3+2} = 0.971$$

$$H(T|\text{OUTLOOK?}) = \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.694$$



數學時間：決策樹

- 用氣溫分類的亂度：

Temperature	>70	<=70
Play	5	4
Don't play	4	1

$$H(T) = -\frac{9}{9+5} \log \frac{9}{9+5} - \frac{5}{9+5} \log \frac{5}{9+5} = 0.940$$

$$H(> 70) = -\frac{5}{5+4} \log \frac{5}{5+4} - \frac{4}{5+4} \log \frac{4}{5+4} = 0.991$$

$$H(\leq 70) = -\frac{4}{4+1} \log \frac{4}{4+1} - \frac{1}{4+1} \log \frac{1}{4+1} = 0.722$$

$$H(T|TEMPERATURE) = \frac{9}{14} \times 0.991 + \frac{5}{14} \times 0.722 = 0.895$$

數學時間：決策樹

- 用濕度分類的亂度：

Humidity	>70	<=70
Play	6	3
Don't play	4	1

$$H(T) = -\frac{9}{9+5} \log \frac{9}{9+5} - \frac{5}{9+5} \log \frac{5}{9+5} = 0.940$$

$$H(> 70) = -\frac{6}{6+4} \log \frac{6}{6+4} - \frac{4}{6+4} \log \frac{4}{6+4} = 0.971$$

$$H(\leq 70) = -\frac{3}{3+1} \log \frac{3}{3+1} - \frac{1}{3+1} \log \frac{1}{3+1} = 0.811$$

$$H(T|HUMIDITY) = \frac{10}{14} \times 0.971 + \frac{4}{14} \times 0.811 = 0.925$$

數學時間：決策樹

- 用是否起風分類之後的亂度

Windy	TRUE	FALSE
Play	3	6
Don't play	3	2

$$H(T) = -\frac{9}{9+5} \log \frac{9}{9+5} - \frac{5}{9+5} \log \frac{5}{9+5} = 0.940$$

$$H(TRUE) = -\frac{3}{3+3} \log \frac{3}{3+3} - \frac{3}{3+3} \log \frac{3}{3+3} = 1$$

$$H(FALSE) = -\frac{6}{6+2} \log \frac{6}{6+2} - \frac{2}{6+2} \log \frac{2}{6+2} = 0.811$$

$$H(T|WINDY) = \frac{6}{14} \times 1 + \frac{8}{14} \times 0.811 = 0.892$$

數學時間：決策樹

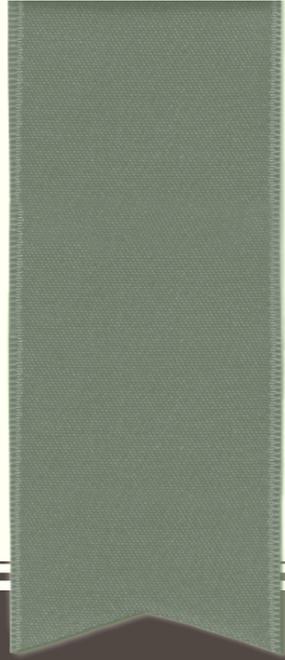
- 不同分法，亂度的比較

$$IG(T, OUTLOOK?) = 0.940 - 0.694 = 0.246$$

$$IG(T, TEMPERATUR) = 0.940 - 0.895 = 0.045$$

$$IG(T, HUMIDITY) = 0.940 - 0.925 = 0.015$$

$$IG(T, WINDY) = 0.940 - 0.892 = 0.048$$



網站範例分享：猜水位（低）

網站範例分享：猜水位（低）

- 人工智慧的運算思維：
 - 程式會隨機顯示水位的高低，我們把判斷的水位高低傳給電腦，讓電腦依循我們的經驗學習判斷
 - 監督式學習

網站範例分享：猜水位（低）

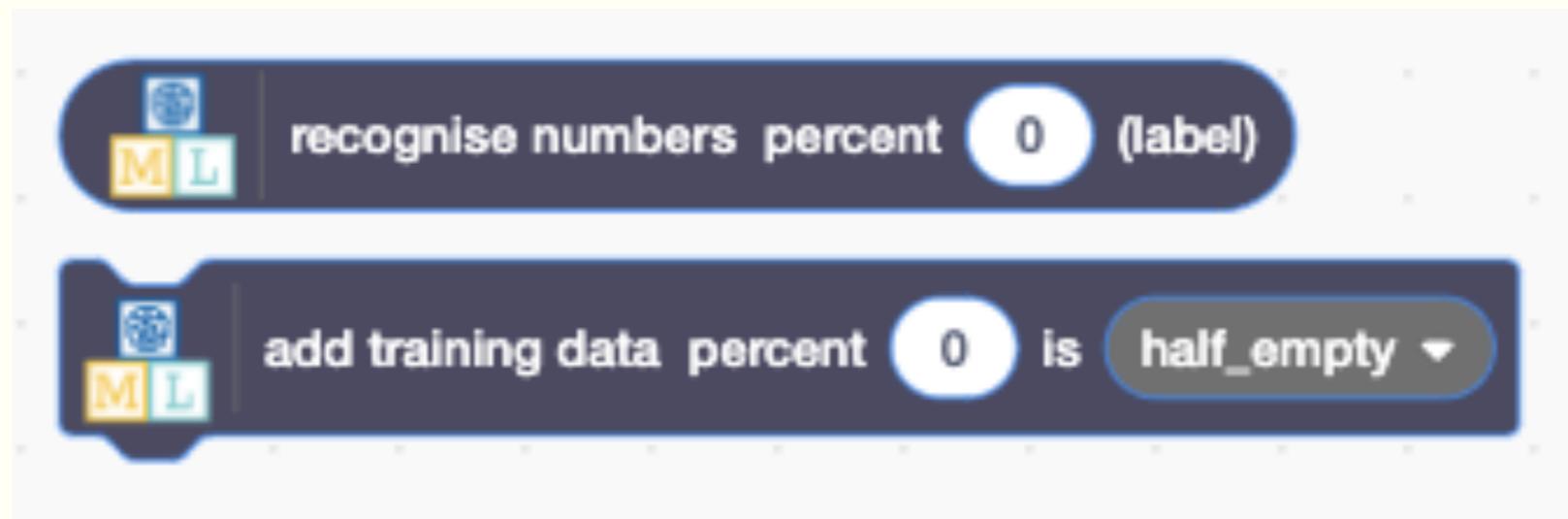
- 程式的運算思維：
 - 使用「liquid percent」變數來裝目前的水位
 - 使用「next target」變數來裝下一個水位
 - 用「在36~64之間隨機選數，直到next target != liquid percent」來找出下一題
 - 用next target 來設定 Y 軸的值+1或-1，製作藍色水位的動畫

網站範例分享：猜水位（低）

- 要做的事的運算思維：
 - 按下「half-empty」時，將目前的水量（liquid percent）傳給「half-empty」標籤紀錄。
 - 按下「half-full」時，將目前的水量（liquid percent）傳給「half-full」標籤紀錄。
 - 然後做完每一題都訓練

網站範例分享：猜水位（低）

- 準備資料庫：
 - 建立專案：
 - 專案名稱：prediction the water
 - value名稱：percent，種類：數值
 - 建立資料模型：
 - 標籤(label)：half_empty 跟 half_full



網站範例分享：猜水位（低）

■ 進入SCRATCH

建立資料模型

收集你希望電腦可以辨識的範例

建立資料模型

訓練 & 測試

使用資料模型訓練電腦學習辨識 numbers

訓練 & 測試

成果應用

在Scratch或Python中使用你訓練好的機器學習模型來製作遊戲或應用程式

成果應用

網站範例分享：猜水位（低）

■ 進入SCRATCH

使用你的機器學習模型製作一些作品吧

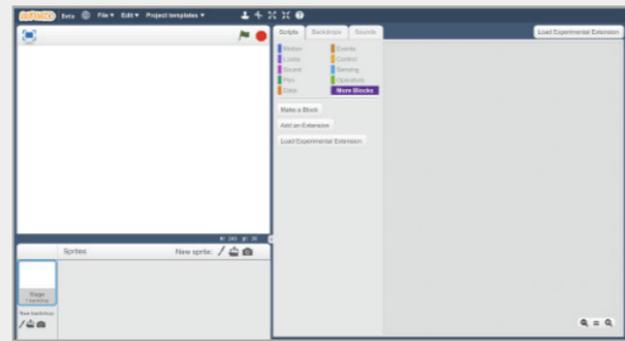
[< 返回到專案](#)

Scratch

使用舊版本的Scratch進行創作



Scratch

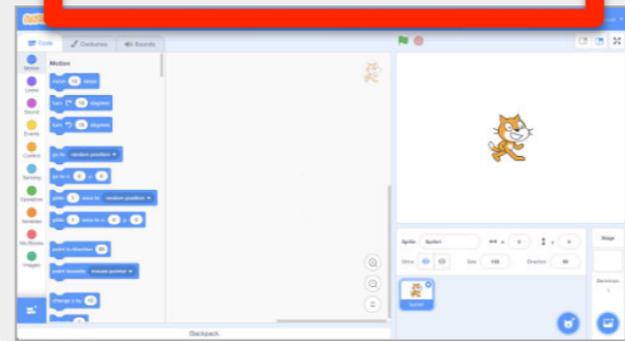


Scratch 3

使用新版的Scratch進行創作



Scratch 3

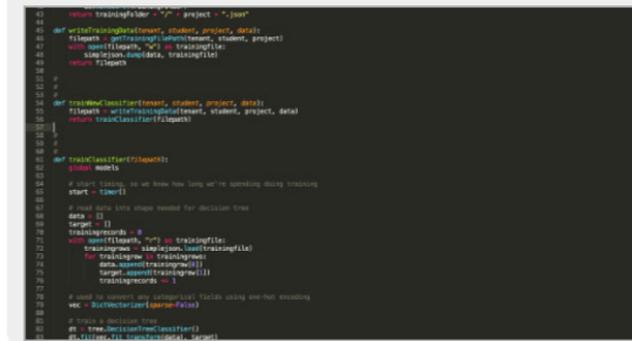


Python

寫一些Python程式碼來使用你的機器學習模型



Python



App Inventor

製作一個手機APP給你的手機或平板使用



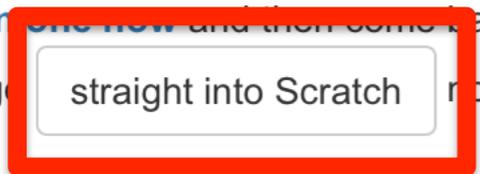
網站範例分享：猜水位（低）

■ 進入SCRATCH

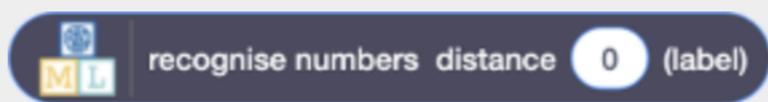
你還沒有訓練機器學習模型

You can **train one now** and then come back to open Scratch.

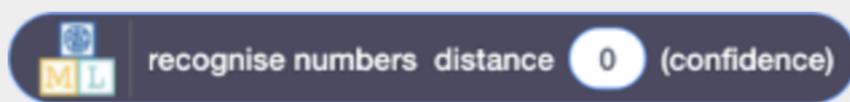
Or you can go **straight into Scratch** now.



你的專案將把這些積木加入到Scratch中

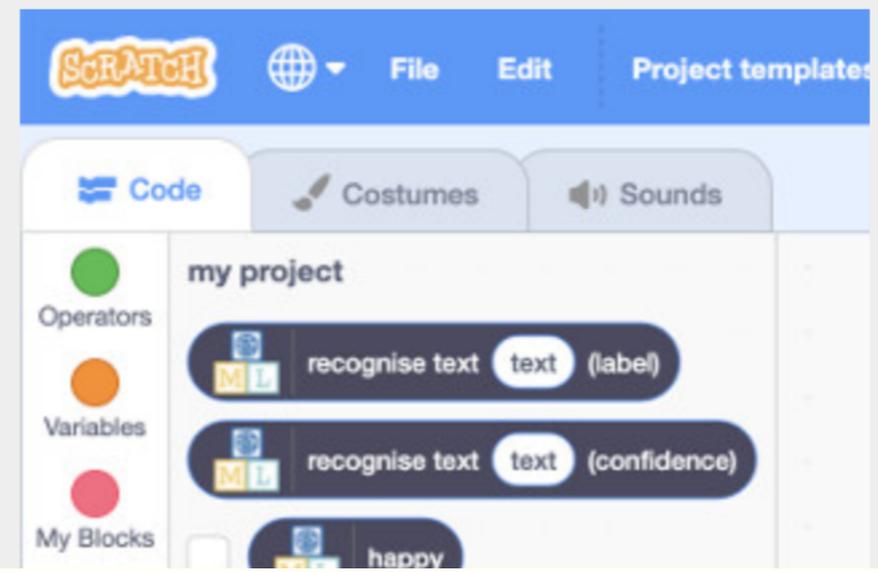


將numbers 放在輸入欄位中，他將回傳你的機器學習模型辨識後的標籤



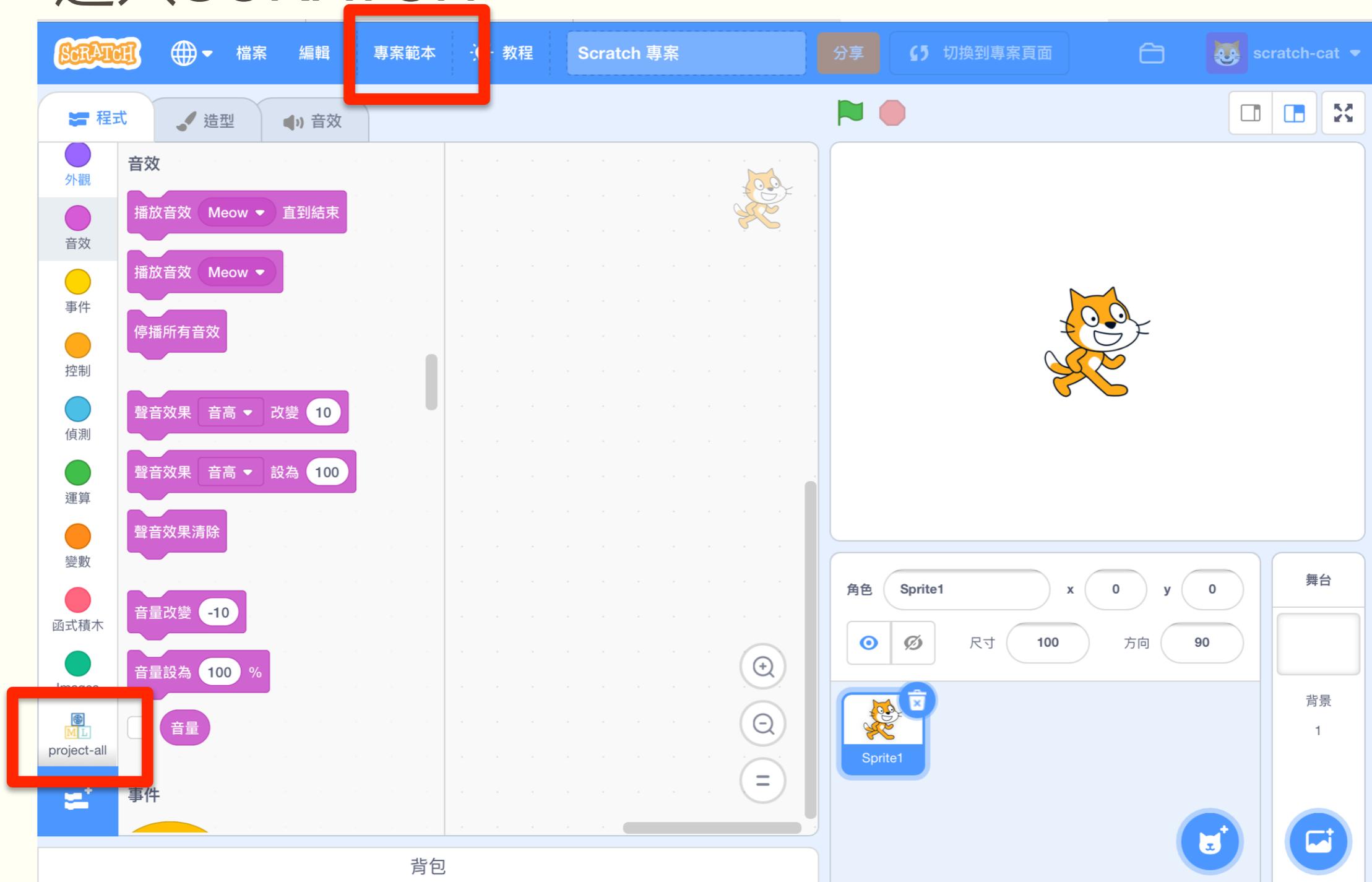
這會回傳你的機器學習模型對辨識結果是 numbers 有多少信心。(以數字0 - 100呈現)

它看起來會像是這樣—除了專案名稱與你的不同



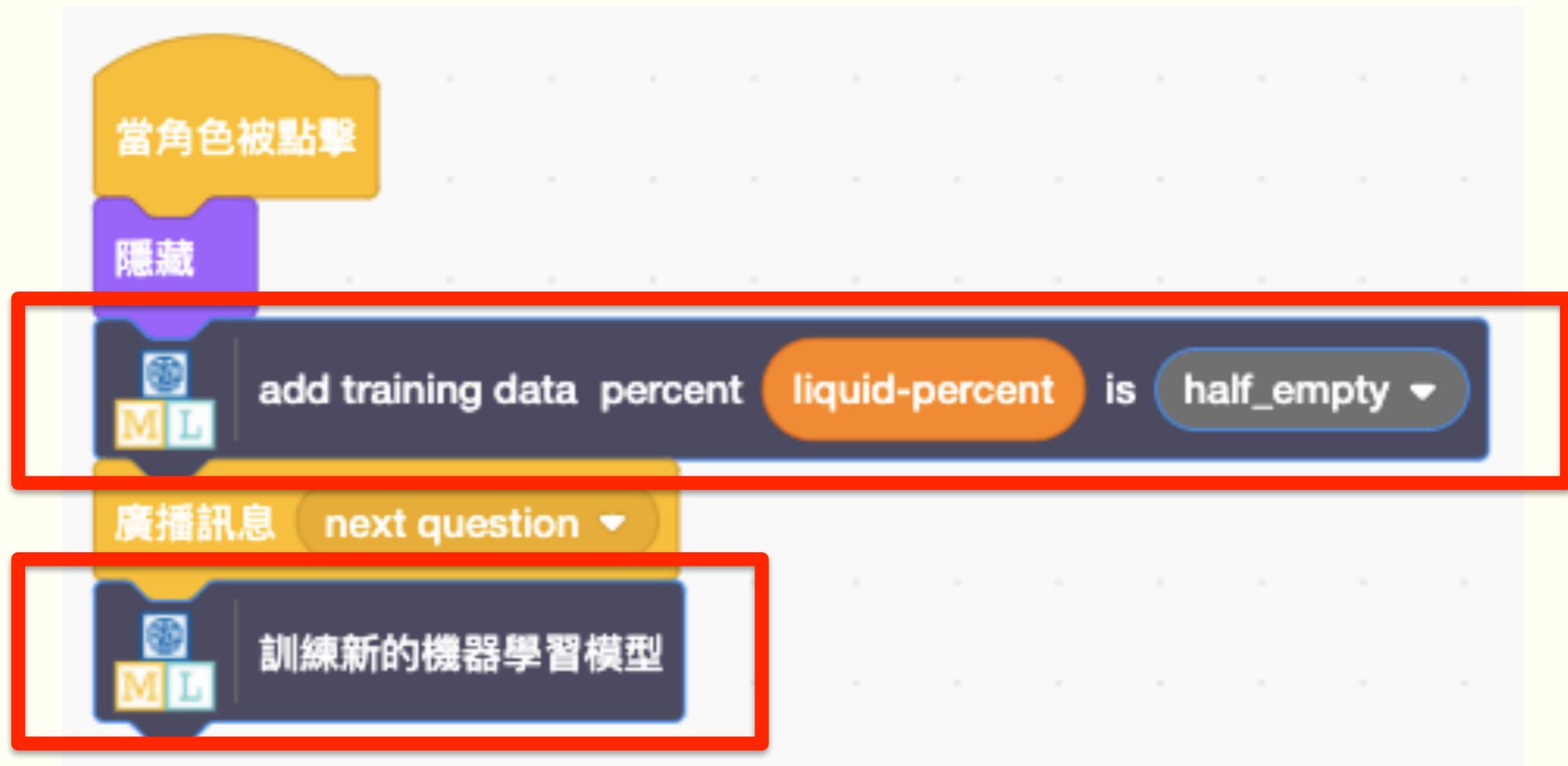
網站範例分享：猜水位（低）

■ 進入SCRATCH



網站範例分享：猜水位（低）

- 要做的-傳送資料並訓練：
 - 「half-empty」角色：按下後將目前的水量（liquid percent）傳給「half-empty」標籤紀錄。



The image shows a sequence of Scratch-like blocks in a block editor. The blocks are:

- 當角色被點擊 (When clicked)
- 隱藏 (Hide)
- add training data percent liquid-percent is half_empty (highlighted with a red box)
- 廣播訊息 next question (Broadcast message)
- 訓練新的機器學習模型 (Train new machine learning model, also highlighted with a red box)

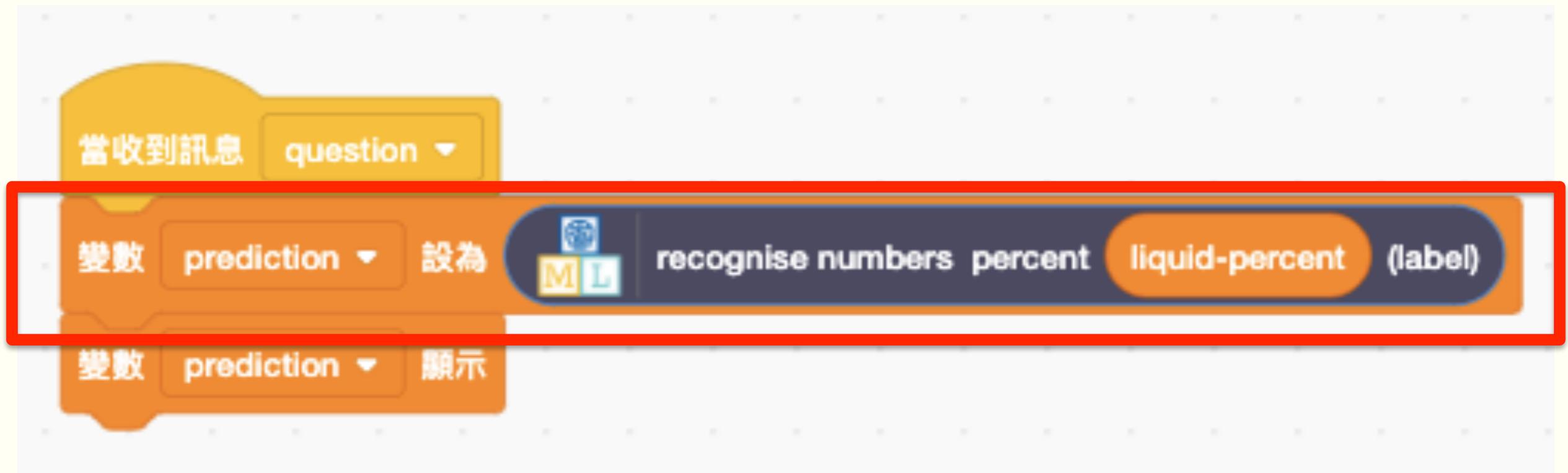
網站範例分享：猜水位（低）

- 要做的-傳送資料並訓練：
 - 「half-full」角色：按下後將目前的水量（liquid percent）傳給「half-empty」標籤紀錄。

The image shows a Scratch code editor with several blocks. The top block is a yellow '當角色被點擊' (When clicked) block. Below it is a purple '隱藏' (Hide) block. The main code block is a dark blue 'add training data percent' block with 'liquid-percent' in an orange input field and 'half_full' in a dropdown menu. Below this is a yellow '廣播訊息' (Broadcast message) block with 'next question' in a dropdown menu. The bottom block is a dark blue '訓練新的機器學習模型' (Train new machine learning model) block. The 'add training data percent' and '訓練新的機器學習模型' blocks are highlighted with red rectangles.

網站範例分享：猜水位（低）

- 要做的-觀看運算結果
 - 在「prediction」角色：



網站範例分享：猜水位（低）

- 觀看訓練資料：

建立資料模型

收集你希望電腦可以辨識的範例

建立資料模型

訓練 & 測試

使用資料模型訓練電腦學習辨識 numbers

訓練 & 測試

成果應用

在Scratch或Python中使用你訓練好的機器學習模型來製作遊戲或應用程式

成果應用

網站範例分享：猜水位（低）

- 觀看訓練資料：

嘗試輸入一些「數字」，觀察電腦如何透過你訓練的模型進行辨識

percent

[測試](#) [Describe your model!](#)

來自訓練伺服器的訊息：

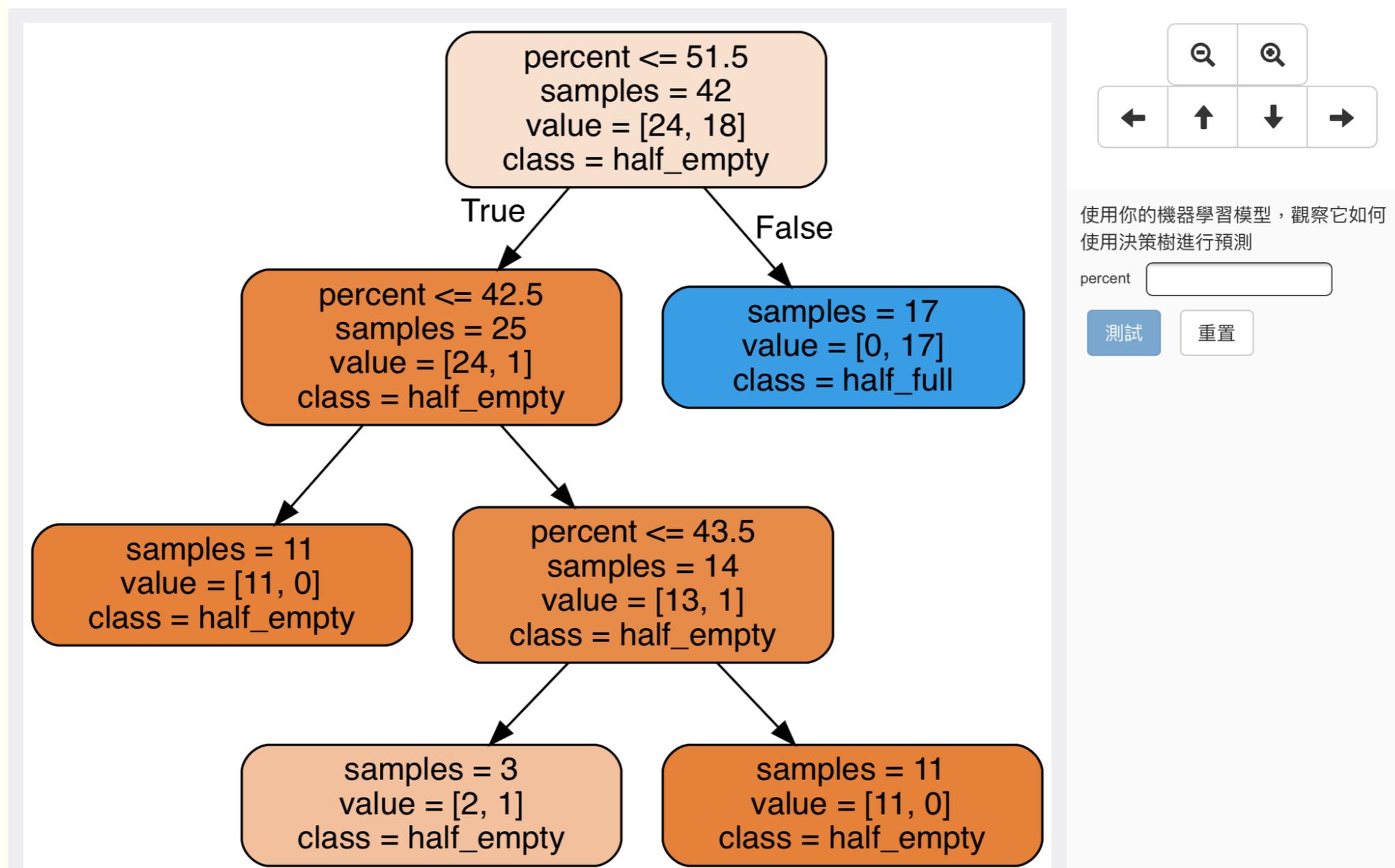
模組開始訓練的時間： Thursday, October 15, 2020 5:02 PM
模組的最新狀態： Available

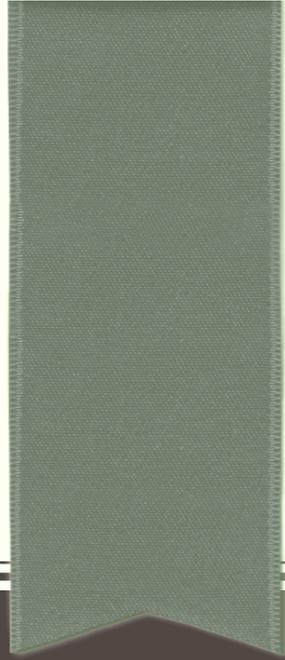
[刪除這個模型](#)

[訓練新的機器學習模型](#)

網站範例分享：猜水位（低）

- 決策樹：





網站範例分享：井字遊戲（高）

網站範例分享：井字遊戲（高）

- 人工智慧的運算思維：
 - 跟電腦玩井字遊戲，把贏家的下法傳給電腦讓他學習。
 - 所謂的「下法」指的是落子時當下的局面：哪裏是自己，哪裏是對手，哪裏是空的

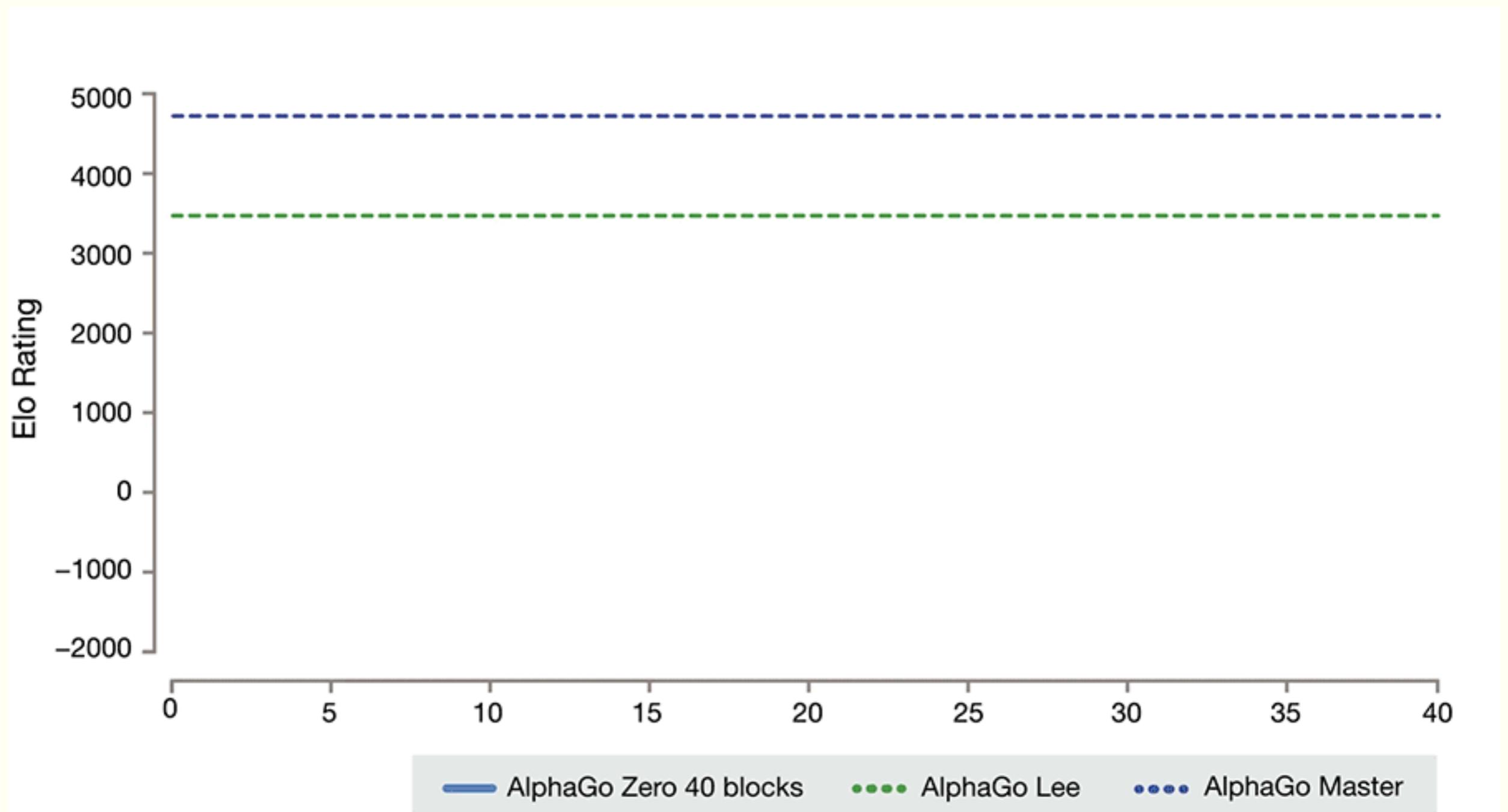
網站範例分享：井字遊戲（高）

■ 補充資料：

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戰勝李世乜
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)

網站範例分享：井字遊戲（高）

- 補充資料：



網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 如何控制落子後顯示O、X？
 - 落子後如何紀錄盤面，並傳給資料庫？
 - 傳給資料庫什麼資料？若是O、X、空格，如何訓練？（如何知道勝負）
 - 如何控制電腦下哪裡？
 - 如何決定遊戲結束？

網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 如何控制電腦下哪裡？
 - 因為資料庫會回傳的是標籤名稱，九個位置都要有各自的標籤
 - 用「computer-move」變數來存電腦要下的位置，下之前會依照目前的盤面去抓標籤來決定落子的位置。

網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 落子後如何紀錄盤面，並傳給資料庫？
 - 傳給資料庫什麼資料？若是O、X、空格，如何訓練？（如何知道該次的勝負）
 - 每回合，每個位置，都會把目前的狀況紀錄下來
 - 送去訓練的是以勝方為玩家的清單，在資料庫裡不是用O、X而是用「玩家(player)」與「對手(opponent)」，確保資料的一致性。
 - 每個位置都有兩份清單：以「O為玩家」的以及以「X為玩家」的。（共18個history清單）

網站範例分享：井字遊戲（高）

■ 程式設計的運算思維：



網站範例分享：井字遊戲（高）

- 程式設計的運算思維：

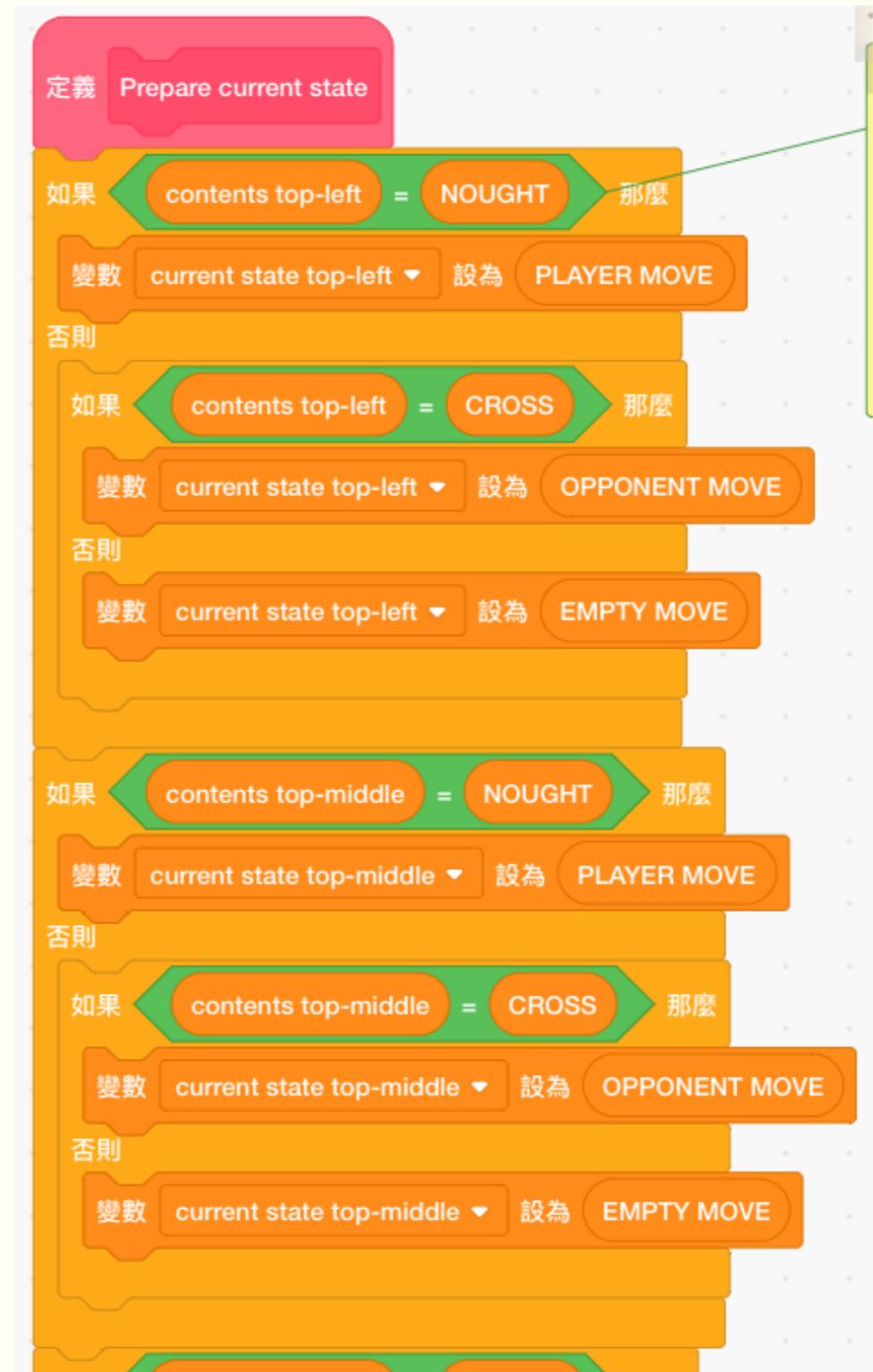
- 這部分有個比較複雜的操作：

- 我們記下來的是每次落子前的每個位置的紀錄，但要送給資料庫的是每次落子前的整體盤面，所以要從九個 history 清單中分別取出每一項，再配合落子當標籤送出。

	起始	1	2	3	4	5	6	7	8	9
左上	/	O	O	O	O	O	O	O	O	O
中上	/	/	/	O	O	O	O	O	O	O
右上	/	/	/	/	X	X	X	X	X	X
左中	/	/	/	/	/	/	X	X	X	X
中中	/	/	X	X	X	X	X	X	X	X
右中	/	/	/	/	/	/	/	O	O	O
左下	/	/	/	/	/	O	O	O	O	O
中下	/	/	/	/	/	/	/	/	X	X
右下	/	/	/	/	/	/	/	/	/	O
	O左上	X中中	O中上	X右上	O左下	X左中	O右中	X中下	O右下	

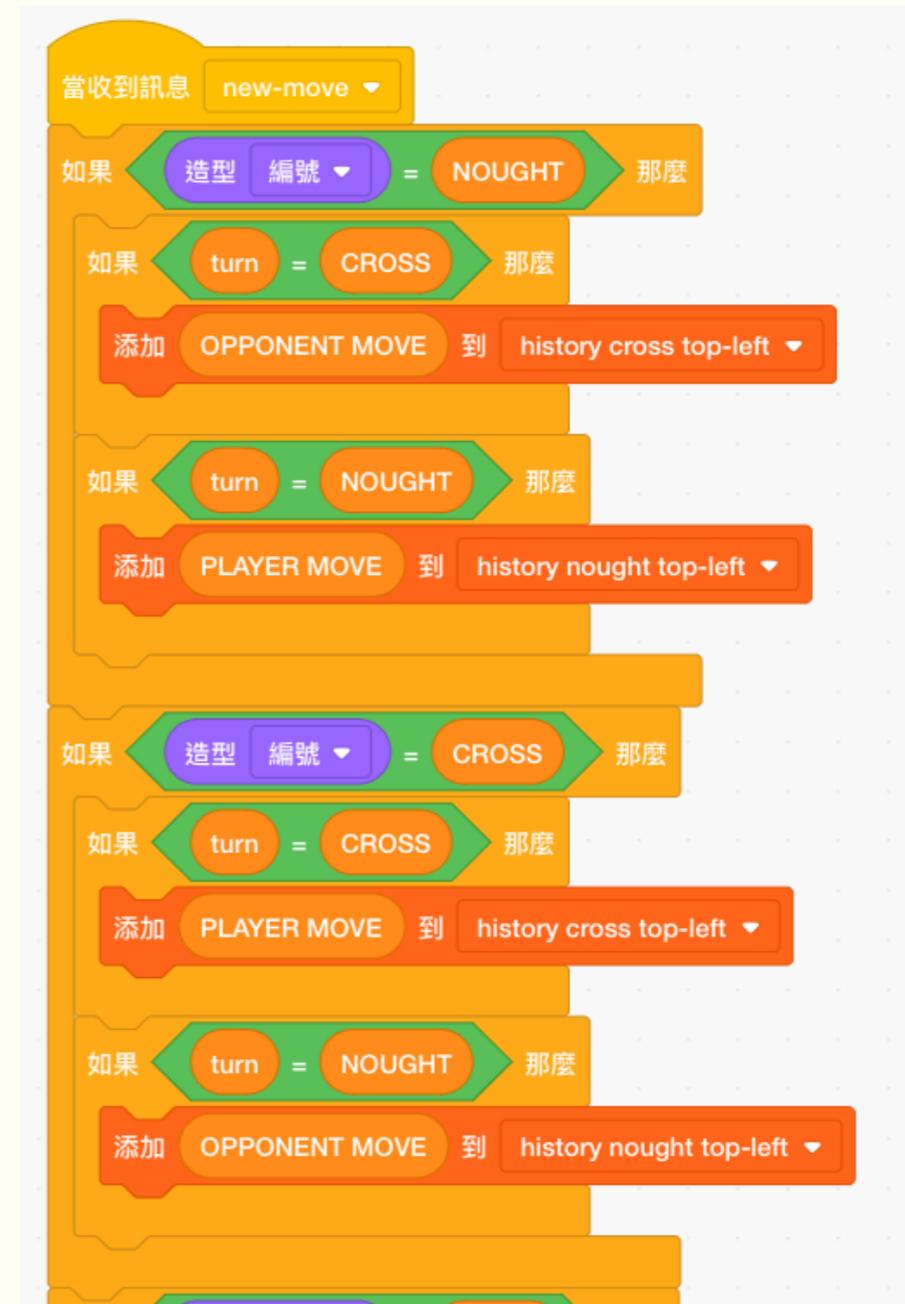
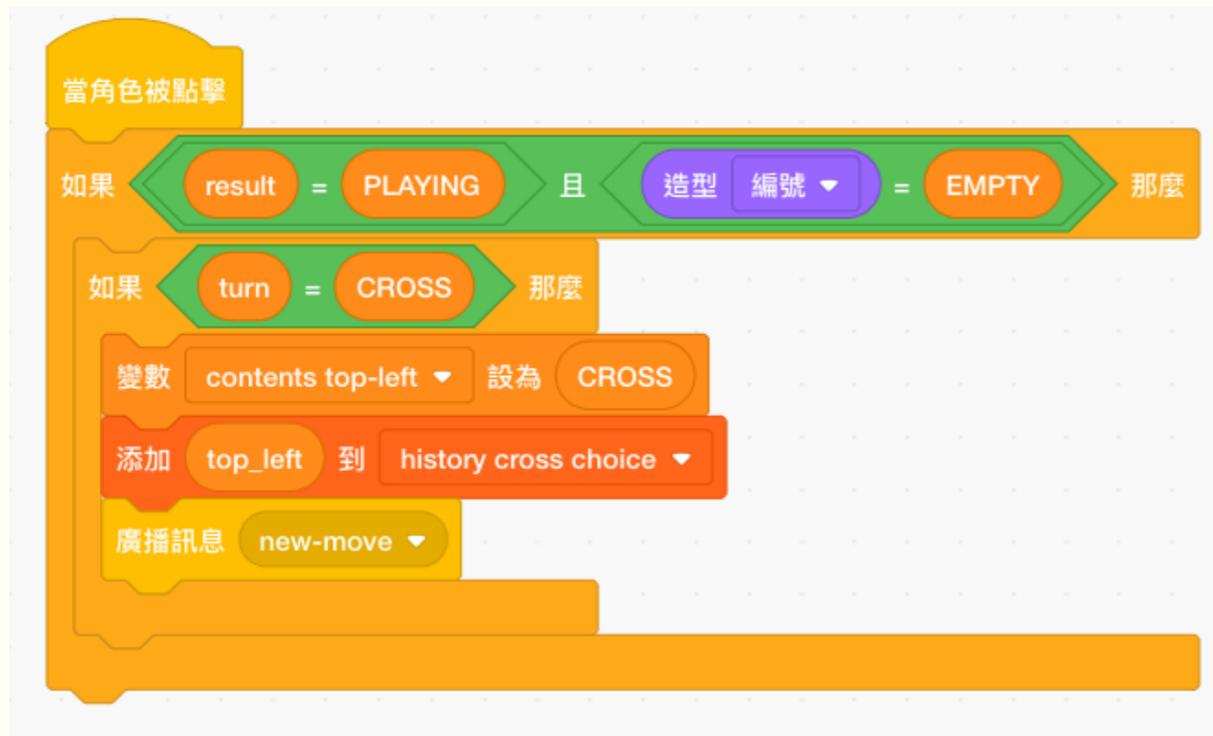
網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 此處衍伸出另一個問題：撈回來的是player跟opponent，電腦怎麼用？
 - 目前盤面用9個「current state xxx」變數來處理，把人類(X)記為對手(opponent)



網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 九個位置皆放置一個角色，皆有三個造型：空白、O、X，點擊會切換造型，並寫入清單。



網站範例分享：井字遊戲（高）

- 程式設計的運算思維：
 - 如何決定遊戲結束？
 - 就慢慢寫囉，反正也要讓線出現，就寫在線上面。



網站範例分享：井字遊戲（高）

- 要做的事的運算思維：
 - 要建立九個位置的標籤，傳給程式來判斷要下哪裡。
 - 每個標籤裡面，都是當下的盤面，所以都要有九個位置(value)，每個位置可能是玩家(player)、對手(opponent)或空的(empty)，所以是多選。

top_left

TopLeft	EMPTY
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	EMPTY
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY

TopLeft	EMPTY
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	EMPTY
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY

TopLeft	EMPTY
TopMiddle	EMPTY

TopLeft	EMPTY
TopMiddle	EMPTY

+ 加入測試範例

36

top_middle

TopLeft	PLAYER
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	OPPONENT
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY

TopLeft	PLAYER
TopMiddle	EMPTY
TopRight	EMPTY
MiddleLeft	EMPTY
MiddleMiddle	OPPONENT
MiddleRight	EMPTY
BottomLeft	EMPTY
BottomMiddle	EMPTY
BottomRight	EMPTY

TopLeft	PLAYER
TopMiddle	EMPTY

TopLeft	OPPONENT
TopMiddle	EMPTY

+ 加入測試範例

12

網站範例分享：井字遊戲（高）

- 要做的事：

定義 use machine learning model

變數 computer-move 設為 recognise numbers TopLeft current state top-left TopMiddle
top_left

定義 add cross moves to training data

add training data TopLeft history cross top-left 的第 move number 項 TopMiddle

定義 add nought moves to training data

add training data TopLeft history nought top-left 的第 move number 項 TopMiddle

recognise numbers BottomRight current state bottom-right (label)

add training data TopMiddle history cross choice 的第 move number 項 is history cross choice 的第 move number 項

add training data Bottom-right history nought choice 的第 move number 項 is history nought choice 的第 move number 項

網站範例分享：井字遊戲（高）

- 要做的事：

The image shows a Scratch script for training a machine learning model. It starts with a '當收到訊息' (When I receive a message) block for 'game-over'. There are two conditional blocks: one for 'result = PLAYER (CROSS) WIN' and one for 'result = COMPUTER (NOUGHT) WIN'. Each conditional block contains a '變數' (Variable) block to set 'move number' to 1, a '重複' (Repeat) block to loop through the length of a history list (cross choice or nought choice), an 'add cross moves to training data' or 'add nought moves to training data' block, and another '變數' block to change 'move number' by 1. At the bottom, a blue block with an ML icon and the text '訓練新的機器學習模型' (Train new machine learning model) is highlighted with a red box.

網站範例分享：井字遊戲（高）

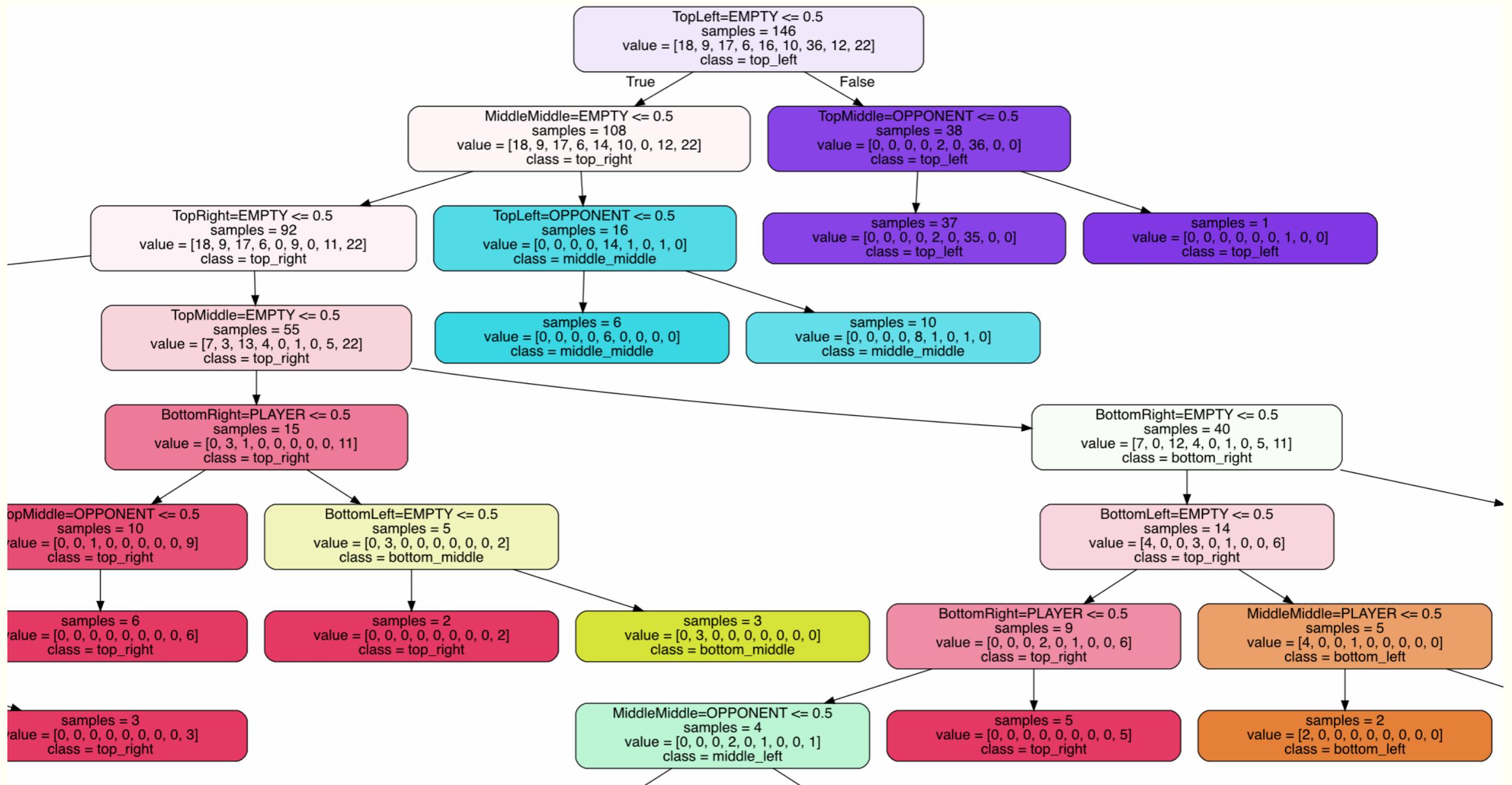
- 看訓練結果：

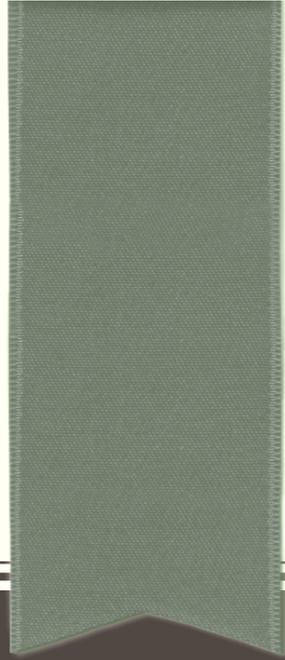
嘗試輸入一些「數字」，觀察電腦如何透過你訓練的模型進行辨識

TopLeft	EMPTY ▼
TopMiddle	EMPTY ▼
TopRight	EMPTY ▼
MiddleLeft	EMPTY ▼
MiddleMiddle	EMPTY ▼
MiddleRight	EMPTY ▼
BottomLeft	EMPTY ▼
BottomMiddle	EMPTY ▼
BottomRight	EMPTY ▼

網站範例分享：井字遊戲（高）

■ 看訓練結果：

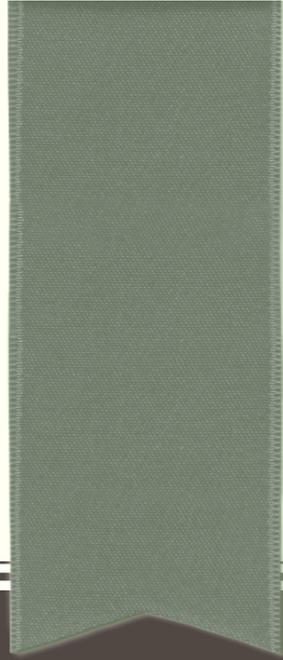




現場製作：打磚塊（中）

現場製作：打磚塊（中）

- 先說好：
 - 程式不完美
 - 規劃不完美
 - 訓練不完美
- 但！不完美才能更好



THANK YOU

oceanblue0212@hsjh.tn.edu.tw
