3. Dimension Reduction

你可能會遇到......

搜集變數要越多越好嗎？
變數之間的相關性怎麼看？
變數和群體之間的關係怎麼看？
有沒有辦法可以縮減變數呢？

Abstract

- 維度詛咒
- 生活中的降維
- PCA
- MDS
- 算法差異

維度詛咒

“

Is Big Data A Miracle or a Curse?

維度詛咒

(Curse of Dimensionality)

當維數提高時

所需要的數據量通常隨著維數的提高

而呈指數級增長

可用數據就會變得很稀疏

“

數學的本質不是將簡單的事情變複雜
而是將複雜的事物簡化

— Stan Gudder

降低維度方法

變數篩選

降維

減少變數的數量，儘量用最少的變數解釋最大的資訊

- PCA (principal components analysis)
- MDS (Multidimensional Scaling)
- SVD (singular value decomposition)
- t-SNE (t-distributed Stochastic Neighbor Embedding)

……………………………

生活中的降維

線性降維

彼此間存在高度正相關

國文(1)　英文(2)　數學(3)

線性降維

加權平均
12*1/6 + 13*2/6 + 15*3/6

代表申請大學門檻是否達成

非線性降維

彼此間存在高度正相關

身高

體　重

非線性降維

BMI(體重/身高平方)

用一維變數代表兩維變數

圖片降維
空間壓縮

一張圖片 321 x 261 個數字

同一人 32 張圖片共 268 萬個數字

每一張圖片用4個數字表達， 每1個數字背後有1張圖片

- 總共儲存數字:4x32+4x(321x 261)=約33萬個數字

Reference: *http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/11ImageProc/15PCA.pdf*

PCA 主成份分析
Principal Component Analysis

PCA 目標

代表性

保有原來變數的資訊

獨立性

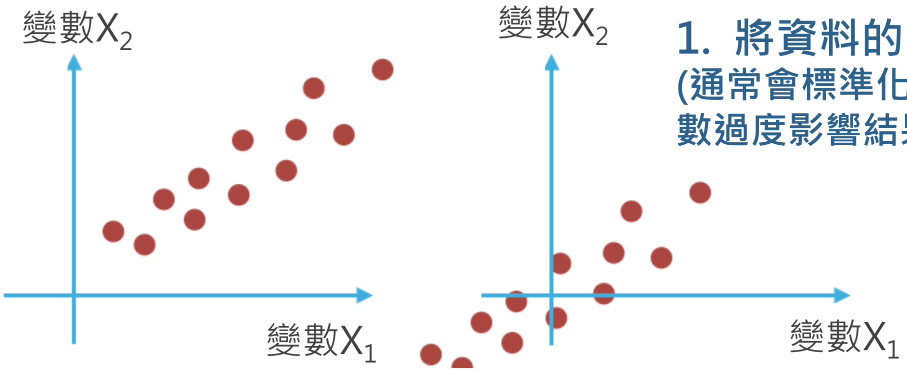
主成份間不能重疊

精簡性

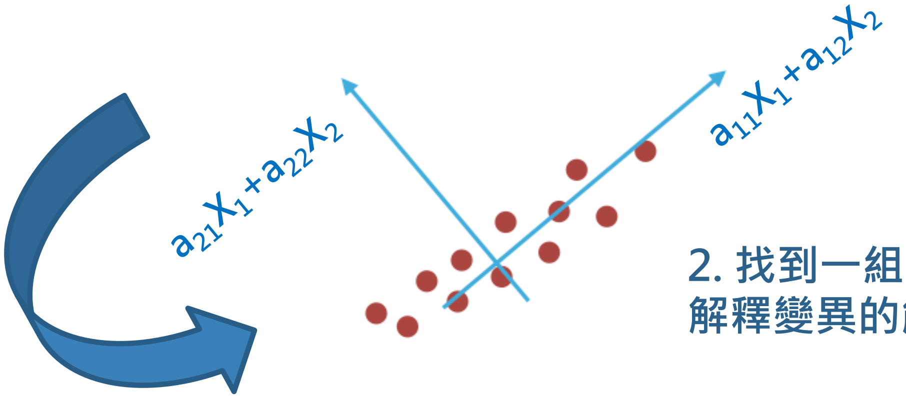
以少數主成份代替原來多個變數

PCA

資料經過轉換過後，保留最大的變異



1. 將資料的中心點移至原點 (通常會標準化，避免變異大的變數過度影響結果)

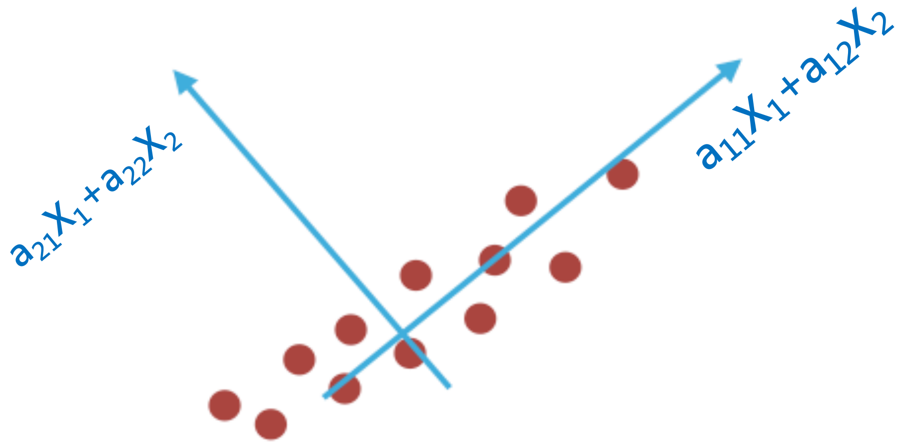


2. 找到一組新的軸，且這些軸解釋變異的能力由大排到小

PCA

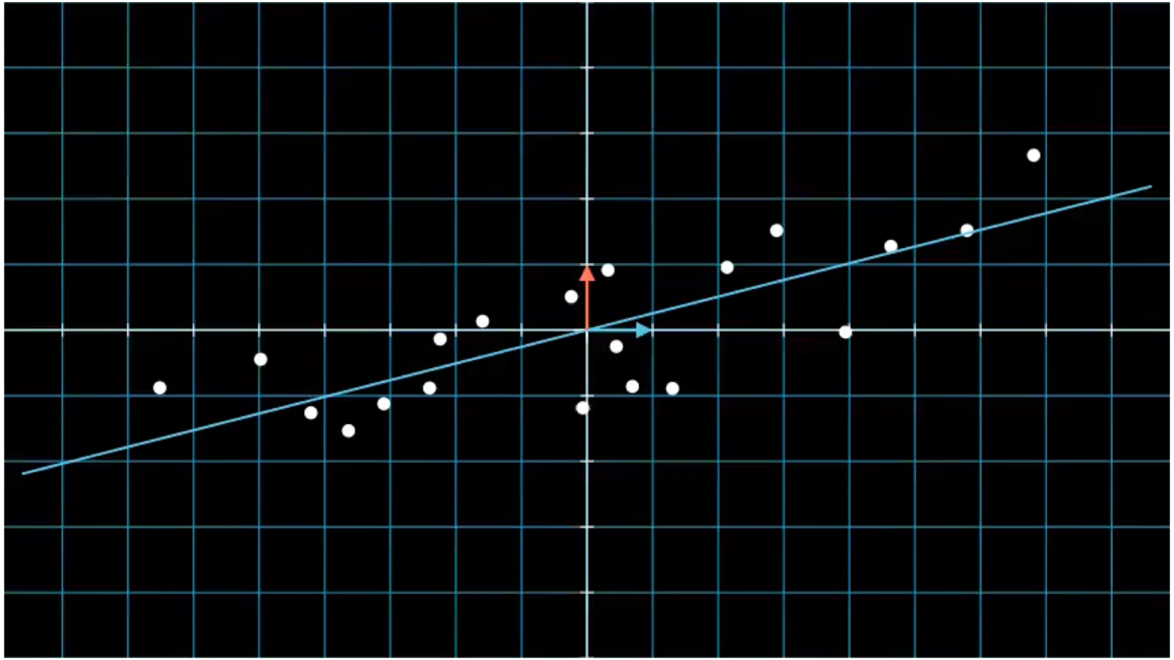
找出可有效表達資料的新變數 (軸)

- X_1與X_2的線性組合
- 以捕捉最多的資料變異



- X_1與X_2的線性組合
- 與第一個軸垂直(捕捉相異的資訊)
- 亦能捕捉更多的資料變異

x 軸並不一定得水平展開
這世上的任何直線都能是你的 x 軸



PCA

Input & Output

Input

Output

n個變數 → 特徵值分解 (Eigen-decomposition) → P個主成分

	變數1	…	n
個體1	V11	…	V1n
個體2	V21	…	V2n
…	…	…	…
個體m	Vm1	…	Vmn

1. 每個主成份都是原始變數的加權平均。
2. 主成份之間不相關。
3. 越前面的主成份解釋越多資料變異。
4. 拋棄較後面的主成份進行降維。

PCA

變數X_1,……,X_p透過「線性轉換」得到主成份F_1 ,……,F_i

P個主成份

$$\boxed{\begin{array}{l} F_1 \\ \\ F_2 \\ \\ \\ F_i \end{array}} \begin{array}{l} = a_{11}X_1 + a_{12}X_2 + \ldots + a_{1p}X_p \\ \\ = a_{21}X_1 + a_{22}X_2 + \ldots + a_{2p}X_p \\ \\ \\ = a_{i1X1} + a_{i2}X_2 + \ldots + a_{ip}X_p \end{array}$$

1. 越前面的主成份解釋越多變異，$Var(F_1) >= Var(F_2) >= \ldots >= Var(F_i)$。
2. 主份彼此不相關，$Cor(F_i , F_j) = 0$，$i \neq j$。
3. 每一個主成份的係數和為1，$a_{i1} + a_{i2} + \ldots + a_{ip} = 1$

估計a_{ip}達成兩大目標：極大化變異數與與保持主成份間不相關

$$
\begin{aligned}
Var(F_i) &= Var(\boxed{a_{i1}}X_1 + \cdots + \boxed{a_{ip}}X_p) \\
&= a_{i1}^2 Var(X_1) + \cdots + a_{ip}^2 Var(X_p) \\
&\quad + 2a_{i1}a_{i2}Cov(X_1, X_2) + \cdots + 2a_{ip-1}a_{ip}Cov(X_{p-1}, X_p)
\end{aligned}
$$

- **極大化$Var(F_i)$**
- **限制式：保持第i個主成份與前(i-1)個主成份不相關**

PCA

降維運算 ： T

還原該轉換的轉換 ： T^{-1}

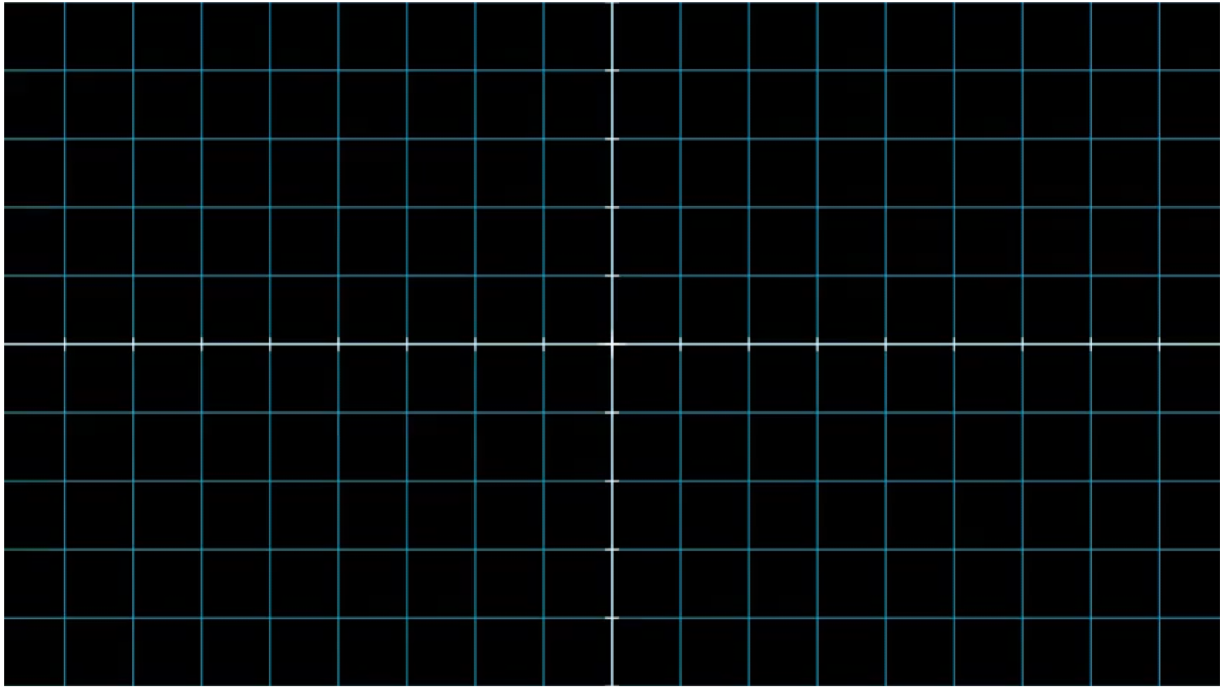
$$\mathrm{RE} = \sum_{i=1}^{n_{samples}} \left\| \vec{x_i} - T^{-1}T(\vec{x_i}) \right\|$$

重建錯誤(Reconstruction Error, RE) :

將所有樣本從二維，降維後所得到的1維特徵L後，再度還原回原2維空間。

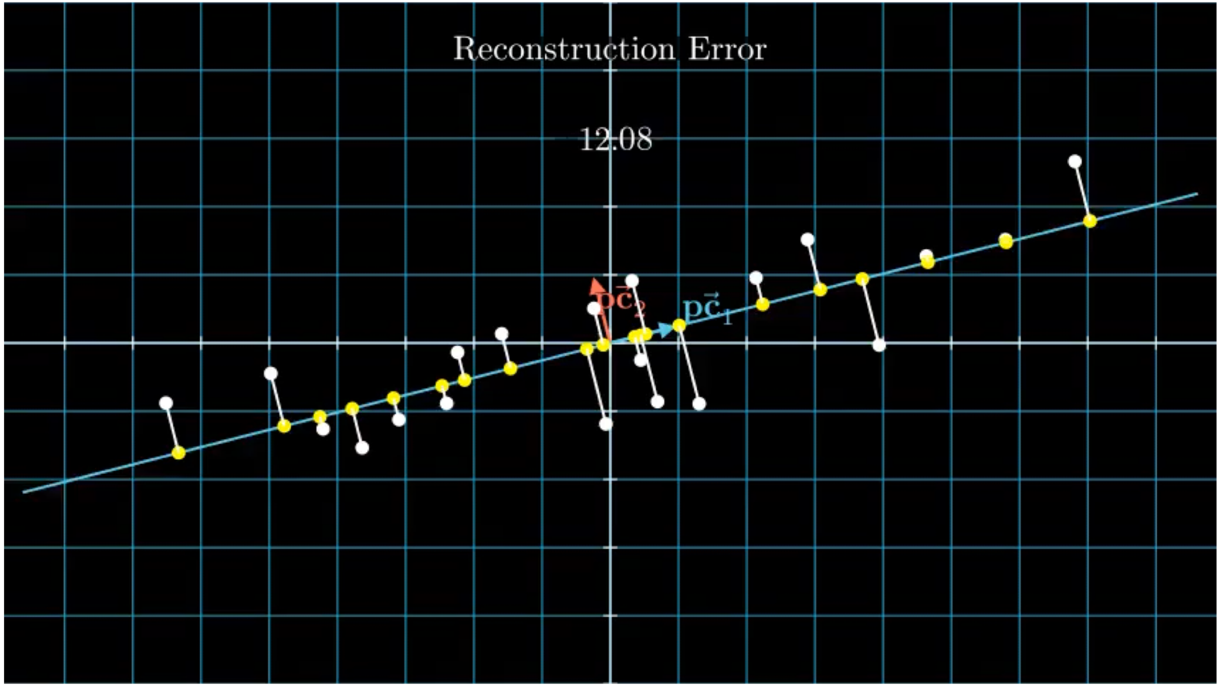
而還原後的結果 X_{proj} 跟原始數據 X 之間的距離總和就是 RE。

RE 越低代表降維結果越成功，代表找出的新特徵 L越具代表性。



PCA

1. PCA找出來的主成分pc_1是能找出最能代表數據 X 變異傾向的向量，它會使得投影後的RE_1最小
2. pc_1在最小化 RE_1的同時，也使得投影後的黃點分布最廣，這代表最大化新特徵 L的變異



如果將數據 X 投影到 PCA 中第二大的pc_2上，將得到最大的 RE_1。
而因投影到 pc_1能得到最小的 RE_1，這間接透露了 pc1 跟 pc2互相垂直的事實（兩者為正交）。

PCA 缺點

PCA 只能捕捉到變量之間線性的關聯

PCA 的計算要對平均中心平移，因此資料集中若有極端離群值發生會影響到估計的結果

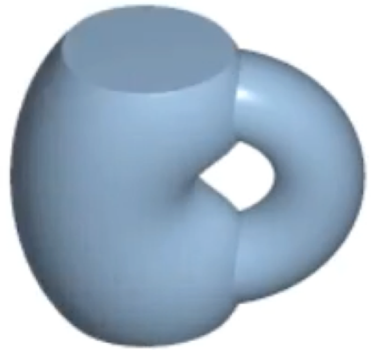
大資料量會讓PCA無法很快的計算完畢

“

線性降維的核心精神
是將原始數據拆解成更具代表性的主成分
並以其作為新的基準
由此獲得更能描述數據本質的新成分表徵

MDS 多元尺度分析
Multidimensional Scaling

流形
Manifold



MDS

「一張圖，勝過千言萬語」

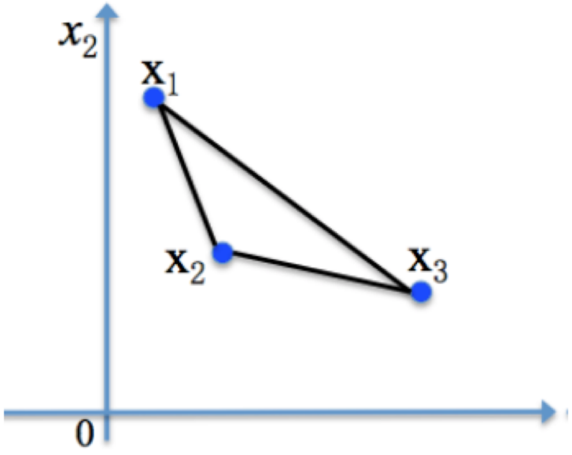
透過保持資料間歐式距離（Euclidean distance）關係，將高維對象非線性投影到低維空間，使得由降維所引起的任何變形達到最小

兩點間的歐式距離d(x, y)d(x, y)為：

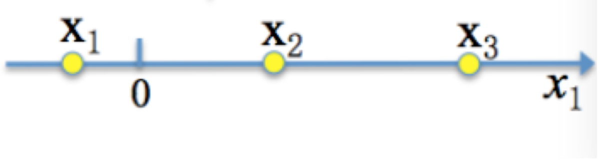
$$d(x, y) = \sqrt{(x-y)(x-y)^T}$$

是一種拓撲保留方法，用於降維外，亦可實現數據視覺化。

MDS



樣本維度由兩維
降成一維



MDS

給定n個d維的樣本數據

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

其中這n個樣本之間的歐式距離表示成以下矩陣

$$d(\mathbf{X}) = \begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \cdots & \delta_{nn} \end{bmatrix} = \begin{bmatrix} 0 & ||\mathbf{x}_1 - \mathbf{x}_2|| & \dots & ||\mathbf{x}_1 - \mathbf{x}_m|| \\ ||\mathbf{x}_2 - \mathbf{x}_1|| & 0 & \dots & ||\mathbf{x}_2 - \mathbf{x}_m|| \\ \vdots & \vdots & \vdots & \vdots \\ ||\mathbf{x}_m - \mathbf{x}_1|| & ||\mathbf{x}_m - \mathbf{x}_2|| & \dots & 0 \end{bmatrix}$$

$\sigma_{ij}(\mathbf{X}) = \delta_{ij}$ 是樣本 i 和樣本 j 之間的歐式距離

MDS

MDS的基本原理是在一個低維空間中，將n個物體鑲嵌到該空間中，使得彼此間的相似度盡可能保留。如果空間是2-3維，則可畫出n個物體的**視覺化結果**。

假設將n個樣本映射到新的p維空間中，映射成的矩陣形式如下：

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix}$$

其中p<d。MDS目的是最小化如下的目標函數。

$$\sigma(\mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{n} (\delta_{ij} - ||\mathbf{y}_i - \mathbf{y}_j||)^2$$

算法差異

算法差異

PCA

關注維度本身，尋求最大化解釋變異

PCA不會保留數據的拓撲結構，本質上是捕捉變數間的線性關聯。因此，若變數具有非線性關係，用此方法即不適當

MDS

關注縮放對象之間的關係

試圖保留數據的拓撲結構，它本質上是一種非線性變換，且不受異常值的影響，可檢測異常值

補充

為什麼在機器學習中，要把圖片做 flatten

許多傳統機器學習模型只能吃平坦的向量

模型	特徵要求
支持向量機（SVM）	需要一維向量
KMeans 聚類	需要一維向量
隨機森林（Random Forest）	需要一維向量
PCA（主成分分析）	需要一維向量

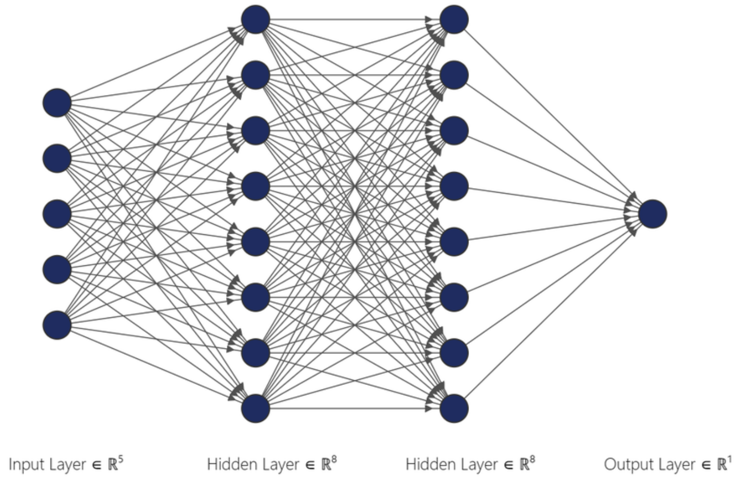
這些經典模型都只接受「一筆資料 → 一個向量」的形式。
它們沒辦法直接處理「高度、寬度、RGB 通道」這種多維圖片。
→ **圖片必須攤平成一條長向量，才能送進這些模型裡面！**

為什麼在機器學習中，要把圖片做 flatten

2. 類神經網絡不一定需要一維向量

(1) FCNN (Fully Connected Neural Network)：

- 只能處理一維向量，一開始就要 flatten，適合表格資料或數字特徵



為什麼在機器學習中，要把圖片做flatten

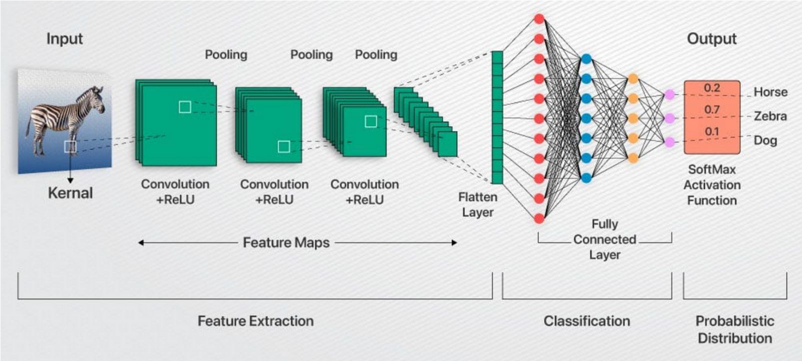
2. 類神經網絡不一定需要一維向量

(2) CNN

- **卷積神經網路 CNN**：CNN可以直接吃4維輸入 (batch_size, height, width, channels)

- **全連接層 (Fully Connected Layer)**：即使在 CNN 中，到了後段要接 MLP 的時候還是要 flatten 一次！

🔵 提示
CNN 要保留圖片的「局部結構」和「空間資訊」（像是 3×3、5×5 卷積核掃過圖片）。
通常 CNN 是到 **卷積層和池化層結束**後，
在進入最後的 **Dense全連接層**之前，才做 flatten。



為什麼在機器學習中，要把圖片做flatten

2. 類神經網絡不一定需要一維向量

比較項目	FCNN	CNN
輸入	一維向量	保持圖片（多維結構）
特徵感知	完全失去空間結構	保留局部空間結構
優勢	簡單、快速	對圖片、空間結構超有效
缺點	對圖片處理能力差	複雜、需要設計卷積核

flatten 前後：資料特徵變了什麼？

1. flatten 前（原本是 3D 結構：height × width × channel）

- ✓ **保有**圖片的**空間資訊**
  （例：一個 pixel 左邊是什麼？上面是什麼？）
- ✓ **局部特徵**還存在
  （像英雄的眼睛、鼻子、嘴巴，是局部連續特徵）
- ✓ 適合給 **CNN（卷積神經網路）** 或其他空間感知模型

2. flatten 後（變成 1D：一條向量）

- ✓ **打散了空間結構**
  只剩下一堆數值，不知道哪個 pixel 原本在哪
- ✓ **局部鄰近關係消失**
  原本左右相連的 pixel，在向量裡可能隔很遠
- ✓ 適合給 **SVM、KMeans、PCA、全連接層（Dense）** 等不需要空間關係的模型

什麼時候該 flatten？什麼時候不要？

任務類型	是否需要 flatten？	原因
傳統機器學習（SVM、KMeans、PCA）	要 flatten	因為模型只懂向量，不懂空間結構
Fully Connected Neural Network（全連接層神經網路）	要 flatten	因為每個輸入神經元要獨立接收一個數字
CNN（卷積神經網路）處理前面幾層	不要 flatten	CNN需要保留局部圖形結構，才能卷積掃描
CNN結束進入分類器時（如Dense層）	那時候才 flatten	要從 feature map 進 Dense做分類

🔵 **總結**

如果你需要模型理解「這是張圖」→ 不要 flatten。
如果你只需要模型看一堆特徵數字→ 就可以 flatten。

Thanks