PCA

機器學習降維

降維有三種

維持原有特徵

- 過多的特徵在後續的UI/UX都會造成災難，寧可犧牲一些命中率，使用重要的特徵給使用者即可，我們可以在分類器的feature_importance_裡找到每個特徵的重要性

組合原有特徵

- 數千個特徵或者數萬個特徵(尤其文字&語言)是無法用傳統方式分析的，於是我們可以結合原有特徵，並且順便完成降維

- 畫圖：不多說！只能2D/3D

共變異數

共變異數如果你不了解的話可以先當成相關係數來思考

相關係數 $\rho = \frac{x和y的共變異數}{x的標準差 \times y的標準差}$

共變異數(covariance): $\mathrm{cov}(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$

變異數(variance): $\mathrm{var}(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)^2$

標準差(standard deviation): $std(x) = \sqrt{\mathrm{var}(x)}$

PCA第一步

✓ 預測資料

[x_1 x_2 x_3]

X

✓ 共變異數矩陣

[Cov(1, 1) Cov(2, 1) Cov(3, 1)]
[Cov(1, 2) Cov(2, 2) Cov(3, 2)]
[Cov(1, 3) Cov(2, 3) Cov(3, 3)]

→ x_1 * Cov(1, 1) + x_2 * Cov(1, 2) + x_3 * Cov(1, 3)

物理意義：把所有對於第一個特徵的影響全部組合起來，變成新特徵

PCA第二步

上面的第一步有個問題，就是你可能會重複計算影響，因為每個轉換不是垂直的！這時候我們就可以來個特徵值分解！對於對稱的矩陣，我們可以把它分解成三個矩陣相乘，而且

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

→ 正交(垂直)矩陣

↓ 對角線矩陣

[常數1　　0　　0　]
[　0　　常數2　　0　]
[　0　　0　　常數3]

PCA第三步

儘管我們可以直接把Q當作我們新的轉換了！但是還可以順便做一件事，只留常數最大的幾個數值就好！
這個留下最大貢獻度的轉換就是我們PCA的精髓了

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$ → 正交(垂直)矩陣

↓

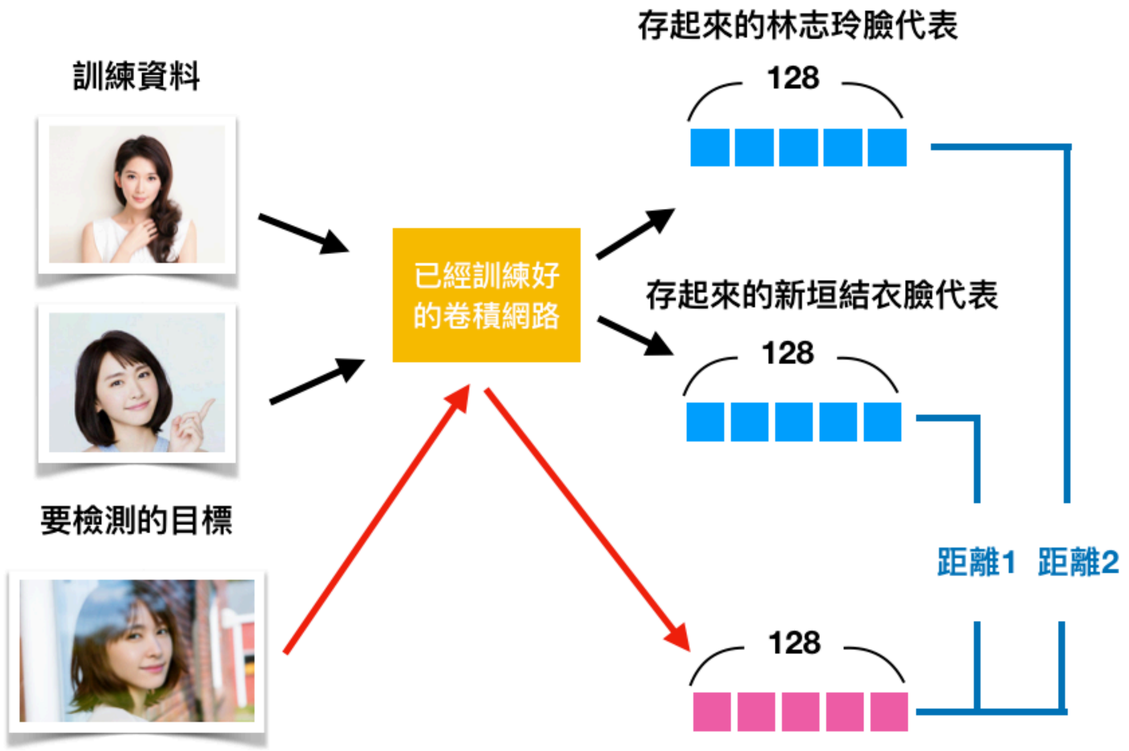
對角線矩陣

[常數1　0　0]
[0　常數2　0]
[0　0　常數3]

人臉辨識

人臉降維

一圖流



我們只要比較檢測的目標和訓練資料的距離即可

FastText

語意分析

NLP

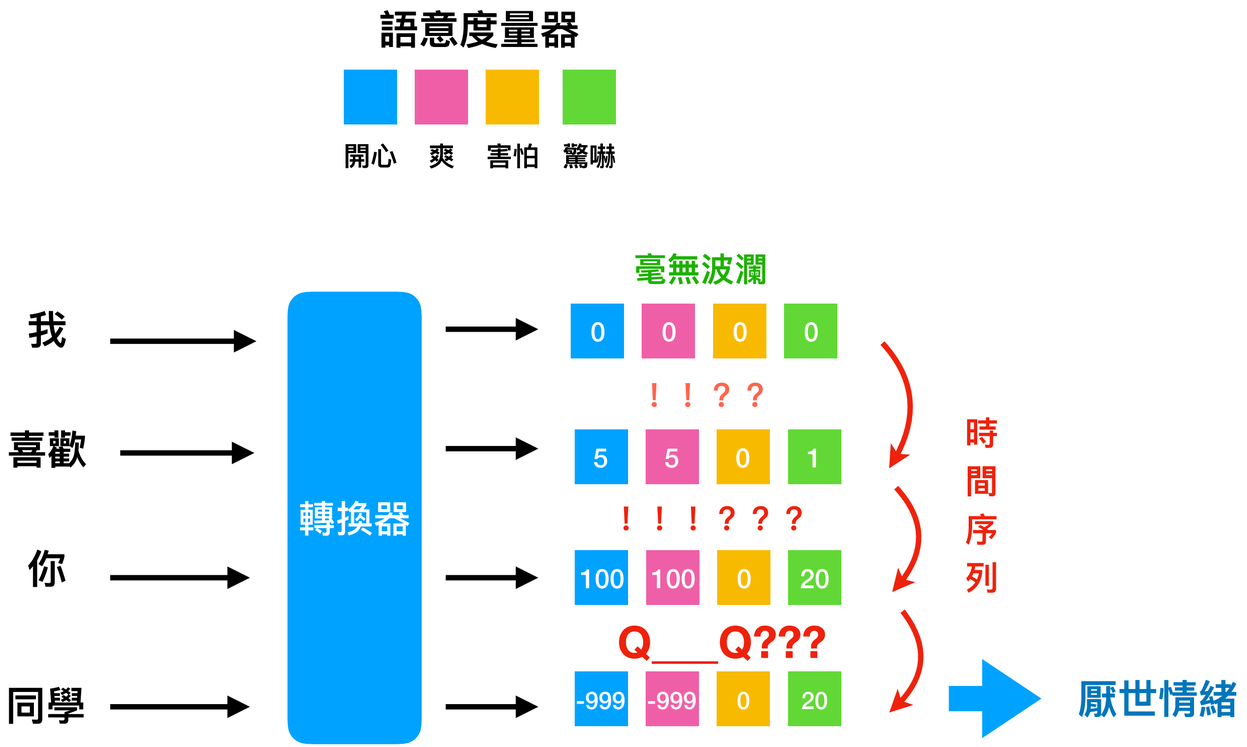
NLP(自然語言處理): Natural Language Processing

不管是什麼任務，中心思想就一個，讓電腦能夠瞭解人類語言

了解人類語言一定要搞定的兩件事！

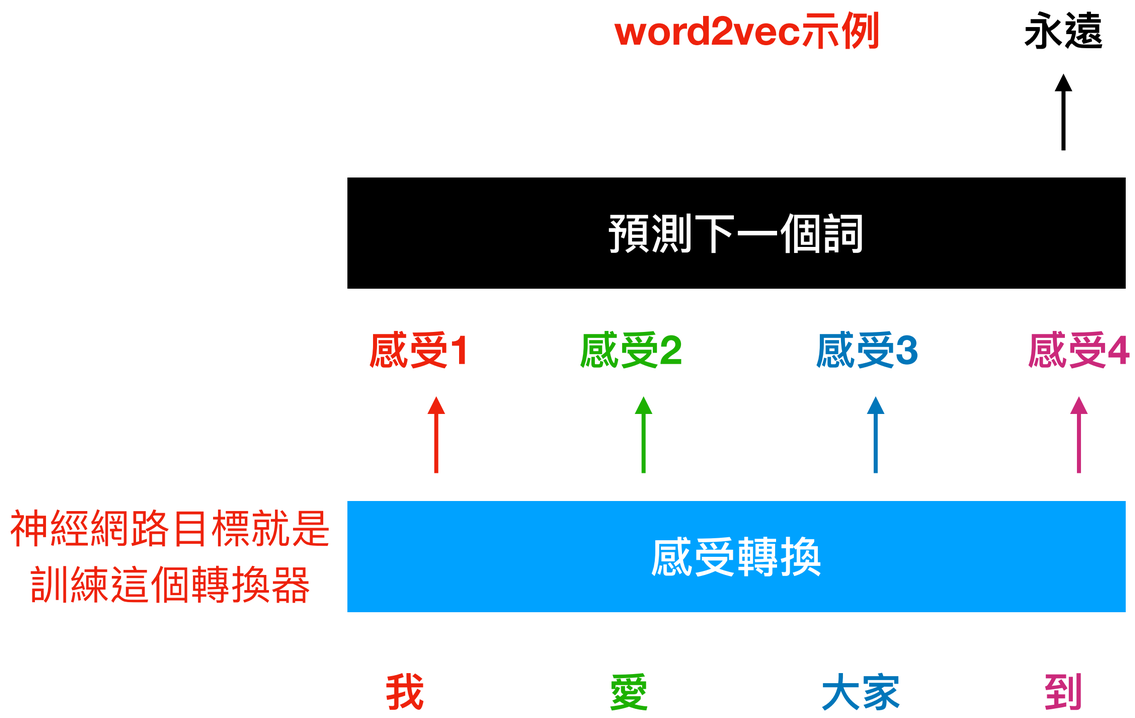
- **詞義(語意): 人類在聽的時候其實不是聽每個詞，而是你的大腦在聽到某個詞的時候會產生對應的情感感受**
- **上下文(序列): 人類在聽的時候其實還會參考上下文，根據上下文的累積，我們可能會有不一樣的意思**
  e.g. 討厭(正常), 討厭(裝可愛), 你沒那麼讓人討厭(喜歡)

一圖流


語意度量器
開心
爽
害怕
驚嚇
我
喜歡
你
同學
轉換器
毫無波瀾
0 0 0 0
！！？？
5 5 0 1
！！！？？？
100 100 0 20
Q___Q???
-999 -999 0 20
時間序列
厭世情緒

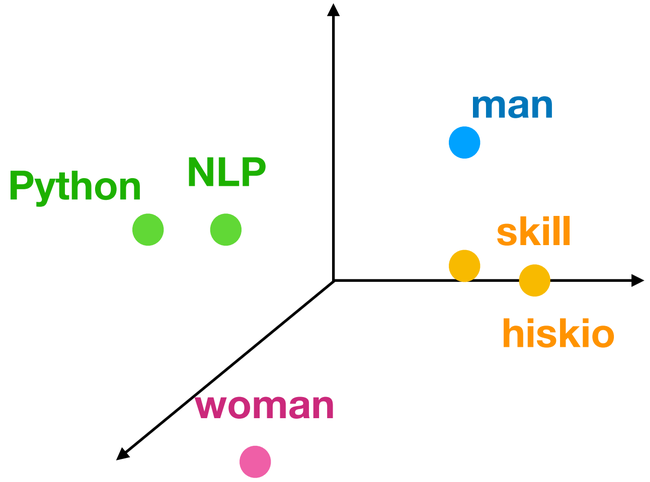
詞向量

我們在NLP任務裡可能會訓練很多種不同模型，但他們在中間大部分都會先做一件事，產生語意



相似度

有了感受我們就可以把感受排列在空間中，比較兩個感受的相似度或者計算兩個感受間的差距



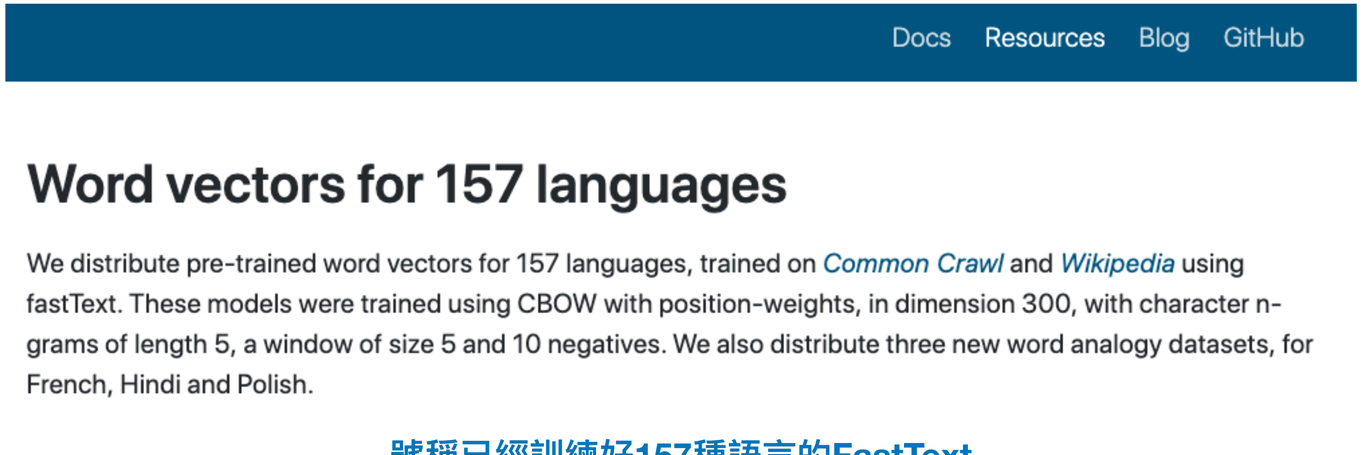
有了詞向量
我們就可以把詞表示在空間中!

cos距離
只計算方向，不計算大小
-1(180度,最不相似)
1(0度,最相似)

FastText

今天我們為何介紹FastText呢

- NLP一個很大的問題是資料有無辦法撐得起你的模型，何不當個『不勞而獲的人』利用大公司訓練好的完整模型，100%比你自己訓練出的語意還準！
- FastText主要以標籤來訓練詞向量，並且加入n-gram的機制，字詞級別的n-gram，尤其字級別的n-gram對我們的英文分析如虎添翼，例如tourism，模型會同時帶入tou, our, ism...等等3-gram字，那我們就會發現他跟其他以ism結尾的字有點類似



號稱已經訓練好157種語言的FastText

Demo Time

Colab網址: https://reurl.cc/xDDYG1

進階回歸

回歸比較

回歸有三種

- RandomForestRegressor
  - ✓ 最無腦的選擇，不用多做任何特徵處理，但誤差相對大，因為你回答的是整群的平均
- 線性回歸(Lasso, Ridge)
  - ✓ 效果好，但一定要將特徵處理成常態分布！
- 深度學習
  - ✓ 將最後一層的Activation調成None，Loss調整成MSE，效果好，也不用做太多特徵處理，但需要你的訓練資料較多