Unsupervised Learning - Word Embedding

(1)該主題解決什麼問題?

當我們要用 vector 來表示 word 的時候，典型的做法會使用 1-of-N Encoding。但是 1-of-N encoding 並不能表示 word 之間關聯性的資訊及詞彙含意，因為每個 word 都被 encode 成 independent 的 vector。

要能表示更多 word 間關係的資訊，我們需要做的是 word embedding。Word embedding 把每個 word 都 project 到一個多維的空間上。在這個空間中，類似語意的 word vector 是比較接近的，而且每個 dimension 都能夠表示特別的含意。

(2)怎麼解決，簡單說明解決方法?

- 基本概念
  - 處理 word embedding 的方法是 input 大量的文章,
    跟據 context 來得到 word 的含意與關聯性。
  - 我們只知道 input，不知道 output,
    因此這是一個 unsupervised learning 的問題。
- 方法
  - Count-based
    - 概念：word 越常一起出現，word vector 會越接近。
    - 假設有兩個 word，w_i 和 w_j。
      代表 w_i 和 w_j 的 vector 為 $V(w_i)$ 和 $V(w_j)$,
      $V(w_i)$ 和 $V(w_j)$ 的內積會與 w_i 和 w_j 一起出現的次數有正相關。
  - Prediction-based
    - 概念：給前一個 word，算出其他 word 出現在下一個的可能性。
    - 做法
      - 把 word 做 1-of-N Encoding 作為 input 來 train 一個 model,
        產生的 output 是其他 word 出現在下一個的可能性。
      - 利用 train 好的 model 的第一個 hidden layer,
        input 進到這個 layer 的結果作為一個 vector,
        這個 vector 就可以作為 word embedding。

- Prediction-based 拓展
  - 只看前一個 word 是不夠的，可以把 input 拓展為 N 個 words。
  - 這時候會做 sharing parameter，讓各個 input 的 weight 都一樣，
    好處是：
    - 避免同一個 word 因為在不同位置 input 而產生出不同的 word vector
    - 減少參數量
- Prediction-based 變形
  - Continuous bag of word (CBOW)：
    拿前後的 word 去 predict 中間的 word
  - Skip-gram：
    拿中間的 word 去 predict 前後的 word
- Word Embedding 特性
  - 同類型的 word vector 擺在一起，可以看得出固定關係

(3)提出影片中不懂的地方或已找到的問題答案

- 用 prediction-based 方法的時候我們要 train 一個 model 來算各個 word 出現在上下文的機率，好奇這個機率是怎麼算出來的？會用怎麼樣的 model 來處理？

Unsupervised Learning - Neighbor Embedding

(1)該主題解決什麼問題?

當 data point 其實是分布在低維的空間中，只是被扭曲塞進一個高維的空間時，我們不能直接以 Euclidean distance 作為兩個點相似程度的依據。這時就需要做 neighbor embedding 來把 data point 降維到可以用 Euclidean distance 來得到點和點之間的相似關係。

(2)怎麼解決，簡單說明解決方法?

課程提到三個方法

- Lolly Linear Embedding (LLE)
  - 取某一個點 x_i，再選幾個 x_i 的 neighbor x_j，找 x_i 跟 x_j 的關係 w_{ij}
  - 找出 wij: 每一個 neighbor xj 乘上 wij 兜出的 linear combination 可以跟 x_i 越接近越好
  - 根據 w_{ij} 把 x_i, x_j 轉成低維度的 z_i, z_j :
    找一組 z， 讓 w_{ij}, z_j 的 linear combination 跟 z_i 越接近越好
- Laplacian Eigenmaps
  - Construct 所有的 data points 變成一個 graph :
    - 計算 data points 兩兩間的相似度,
      如果大於一個 threshold 就把它們連起來
    - 如果 x_1 跟 x_2 在 high density 的 region 上接近,
      做完 dimension reduction 出來的 z_1 跟 z_2 也是相近的。
- T-distributed Stochastic Neighbor Embedding
  - 問題
    - LLE 和 Laplacian Eigenmaps 沒有限制不相近的點要分開,
      不同 class 的點可能還是會擠在一起。
  - 概念
    - 計算 x 兩兩間的 similarity 並做 normalization，得到 x 的 distribution
    - 假設已經找出一組低維的 z,
      計算 z 兩兩間的 similarity 並做 normalization，得到 z 的 distribution
    - 找出一組 z，可以讓 x 跟 z 的 distribution 越接近越好

(3)提出影片中不懂的地方或已找到的問題答案

- 要怎麼知道 data 需不需要做 neighbor embedding 來降維？
    - 特徵過多的時候可能會造成 overfitting、參數多而處理速度慢，或者是做視覺化的時候，就需要對特徵來做降維

- 怎麼知道要降維到什麼程度？
- LLE 和 Laplacian Eigenmaps 感覺相似，兩者在應用情境上有什麼不同？