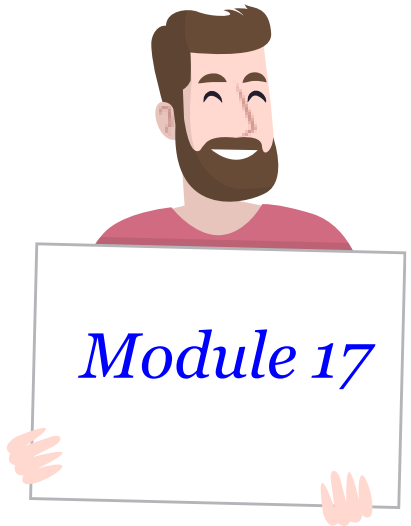
CBOW模型介紹



Estimated time:

45 min.

學習目標

- 17-1: CBOW介紹
- 17-2: CBOW演算法
- 17-3: CBOW與Skip-gram

17-1: CBOW介紹

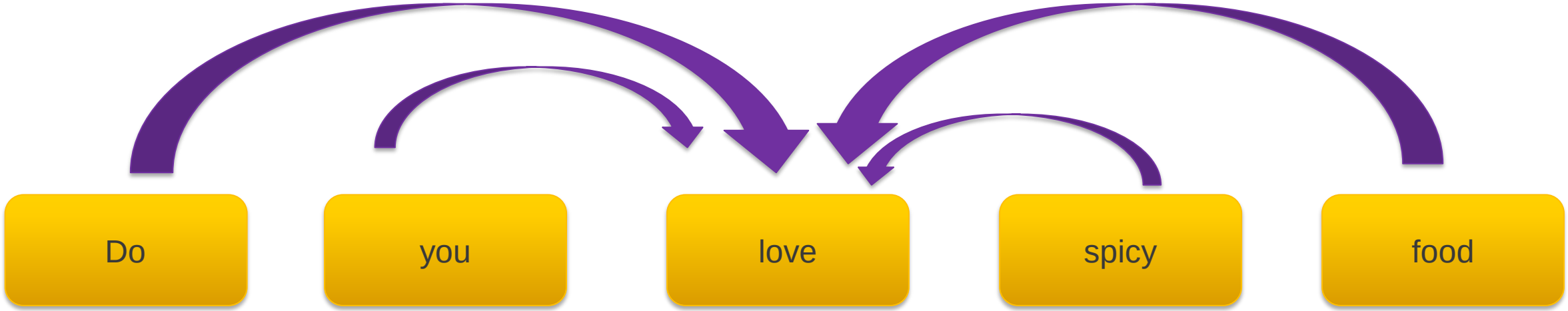
- CBOW模型介紹
- CBOW訓練資料

CBOW模型介紹

- **CBOW(Continuous Bag Of Words)是一個基於神經網路Word2vec模型**
  - **觀念跟skip-gram非常相似，只是在蒐集訓練資料的時候不太一樣**
  - **CBOW模型比skip-gram來得快**

CBOW訓練資料

- CBOW使用的核心思想是給定上下文字的情況下，去預測目標字
- 換句話說，某些上下文字常常對應到同一些字，那麼這些字可能意思相近



CBOW訓練資料

- 為了將上下字所對應的目標字蒐集起來，CBOW模型首先會去閱讀所有語料庫並產生訓練資料集
- 可以設定window size大小來去蒐集訓練資料，如果window size為2，代表每看到某個字的時候往前往後2個字都被視為上下字

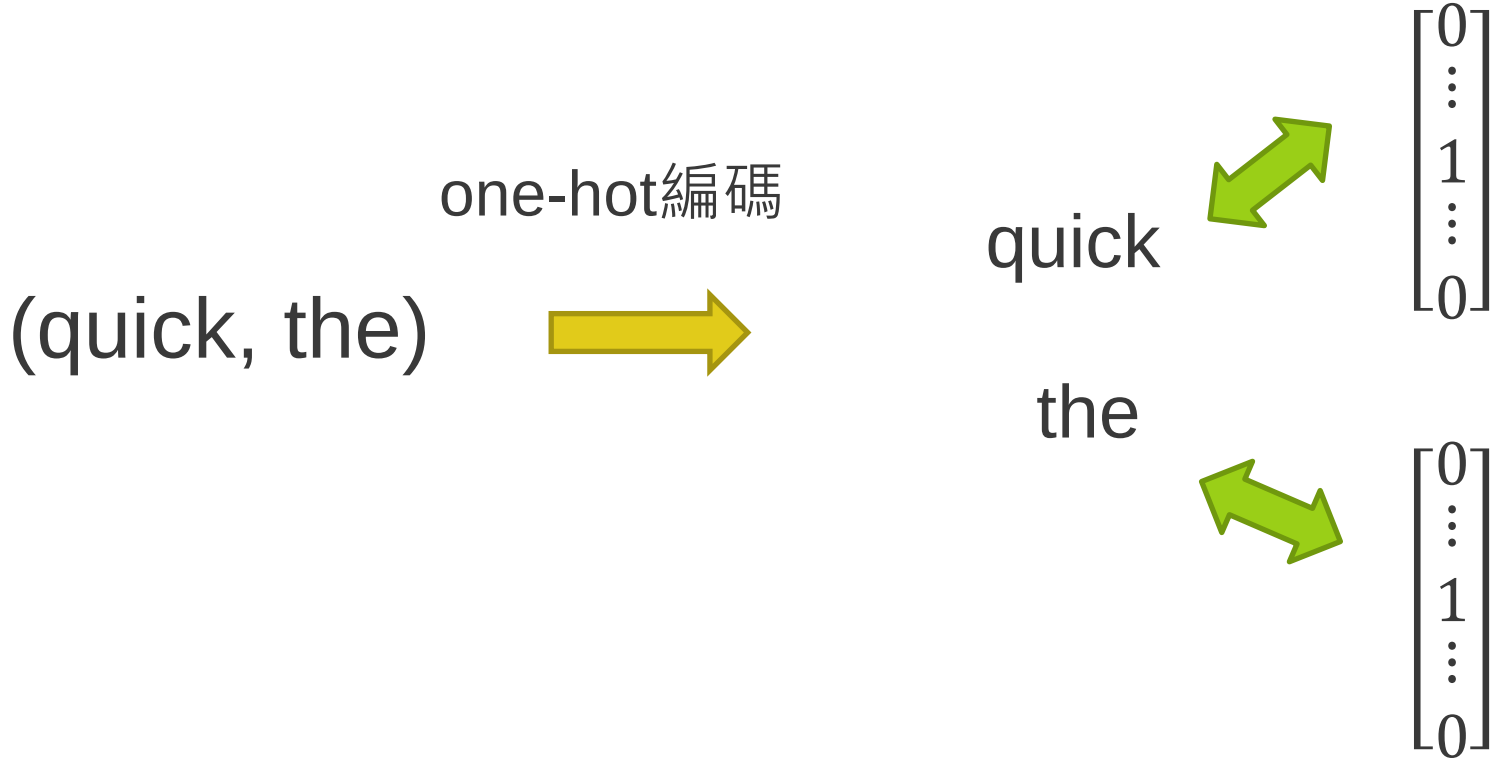
the quick brown fox jumps over the lazy dog. → (quick, the)
(brown, the)

the quick brown fox jumps over the lazy dog. → (the, quick)
(brown, quick)
(fox, quick)

the quick brown fox jumps over the lazy dog. → (the, brown)
(quick, brown)
(fox, brown)
(jump, brown)

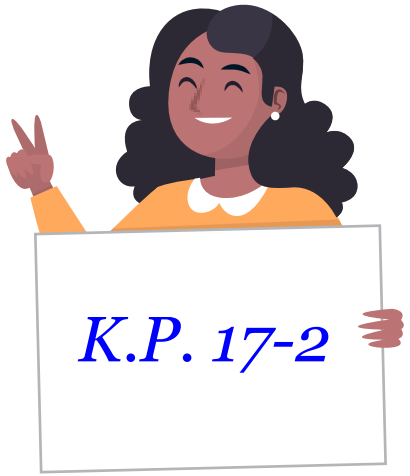
CBOW訓練資料

- 將上一個步驟所蒐集的資料集，每個字做one-hot編碼
  - 此步驟跟bag of words一樣，即給予每個不同的字獨特的ID，並將其轉換成一個向量



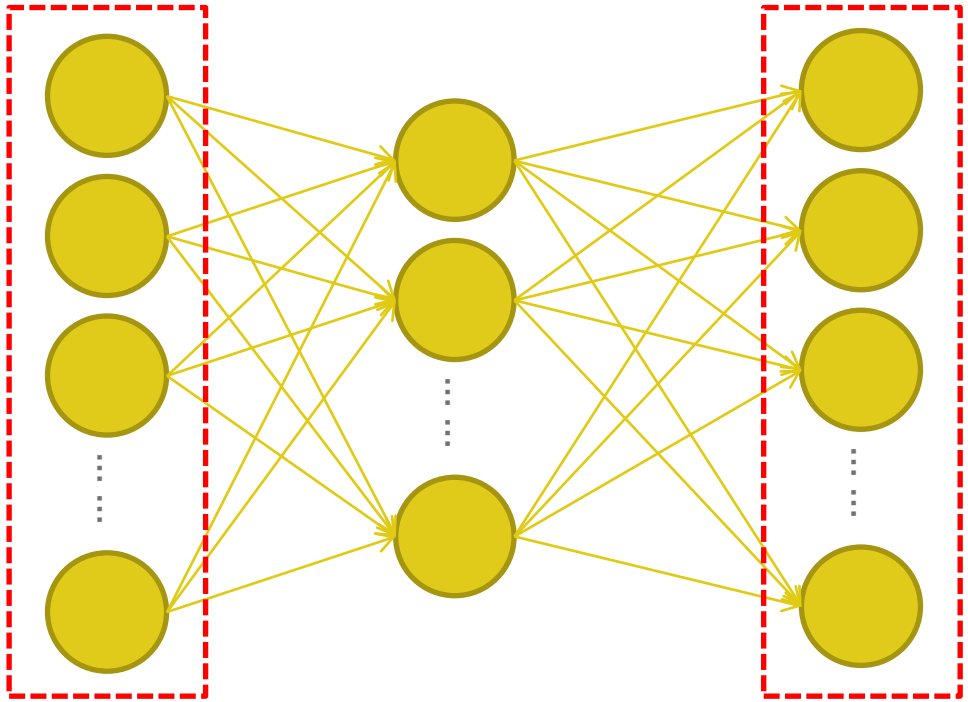
17-2: CBOW演算法

- **CBOW演算法**



CBOW演算法

- CBOW的做法主要是先建立一個只有一個隱藏層的神經網路
- 為了要將每個字one-hot編碼的結果輸入網路，此網路輸入層以及輸出層的神經元數量需為字彙量

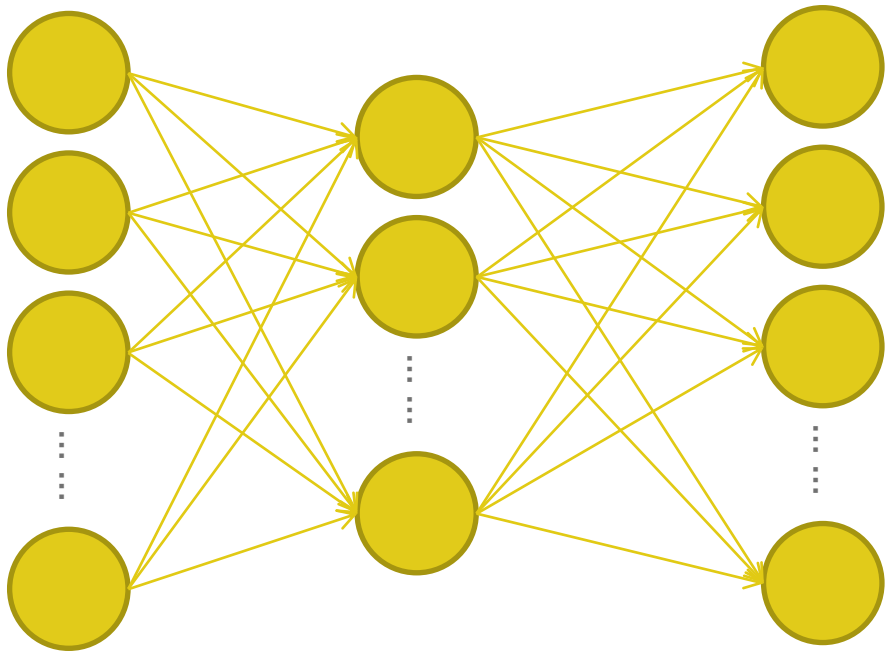


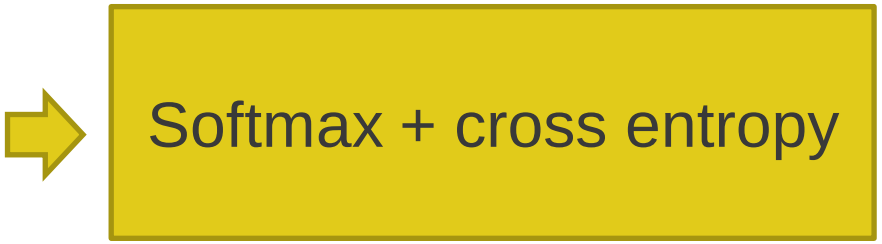
輸入層、輸出層之神經元數量為字彙量

CBOW演算法

- 將之前蒐集來的訓練集輸入
  - 此時每個字已經是one-hot編碼的結果
  - 每筆資料的輸入為目標字所有的上下文，期望輸出為目標字，例如輸入the, quick, fox, jump這些資料，期望輸出為”brown”這個字

(the, brown)
(quick, brown)
(fox, brown)
(jump, brown)



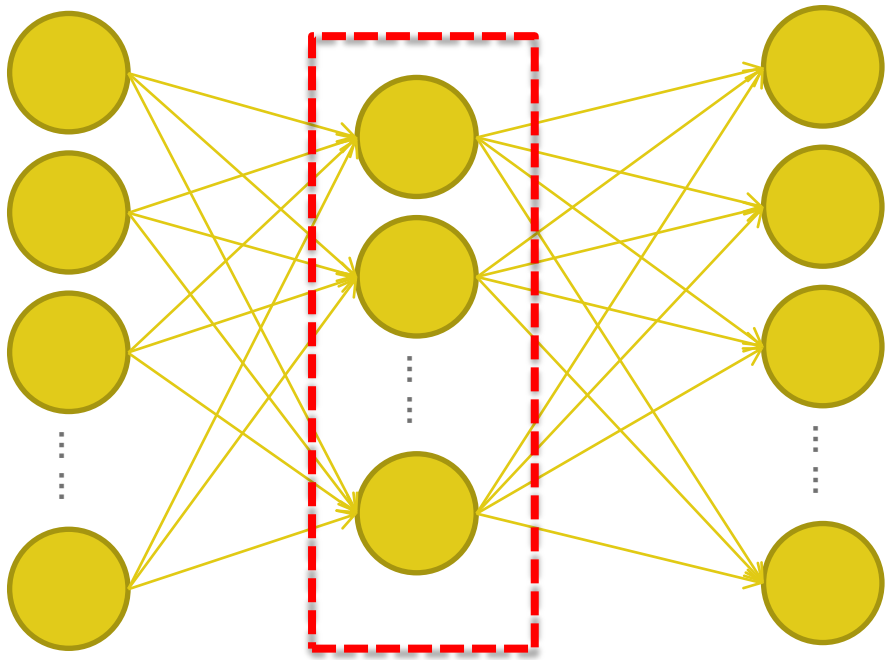


CBOW演算法

- 將the, quick, fox, jump輸入的時候
  - 請用同一組W去轉換到隱藏層
  - 當依序將the, quick, fox, jump當輸入時，請將他們隱藏層的向量平均在往下算

平均(將the, quick, fox, jump的隱藏層向量做平均來讓網路繼續往下算)

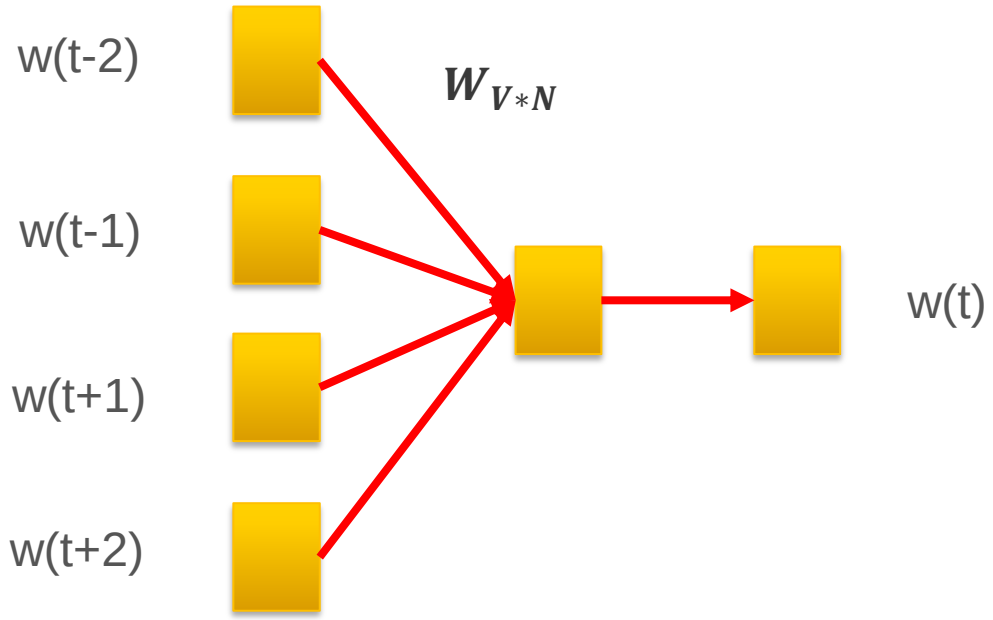
(the, brown)
(quick, brown)
(fox, brown)
(jump, brown)



Softmax + cross entropy

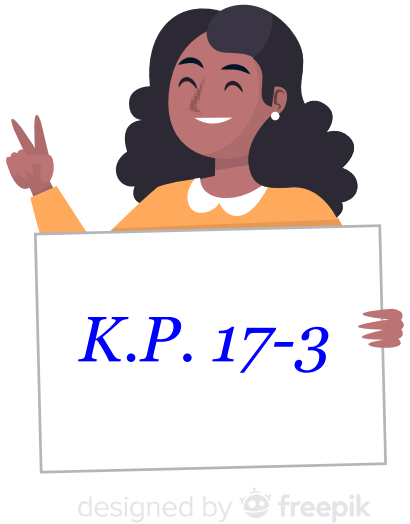
CBOW演算法

- 也有人習慣用以下圖來表示CBOW演算法
  - 不過特別注意W_{V*N}是同一組參數



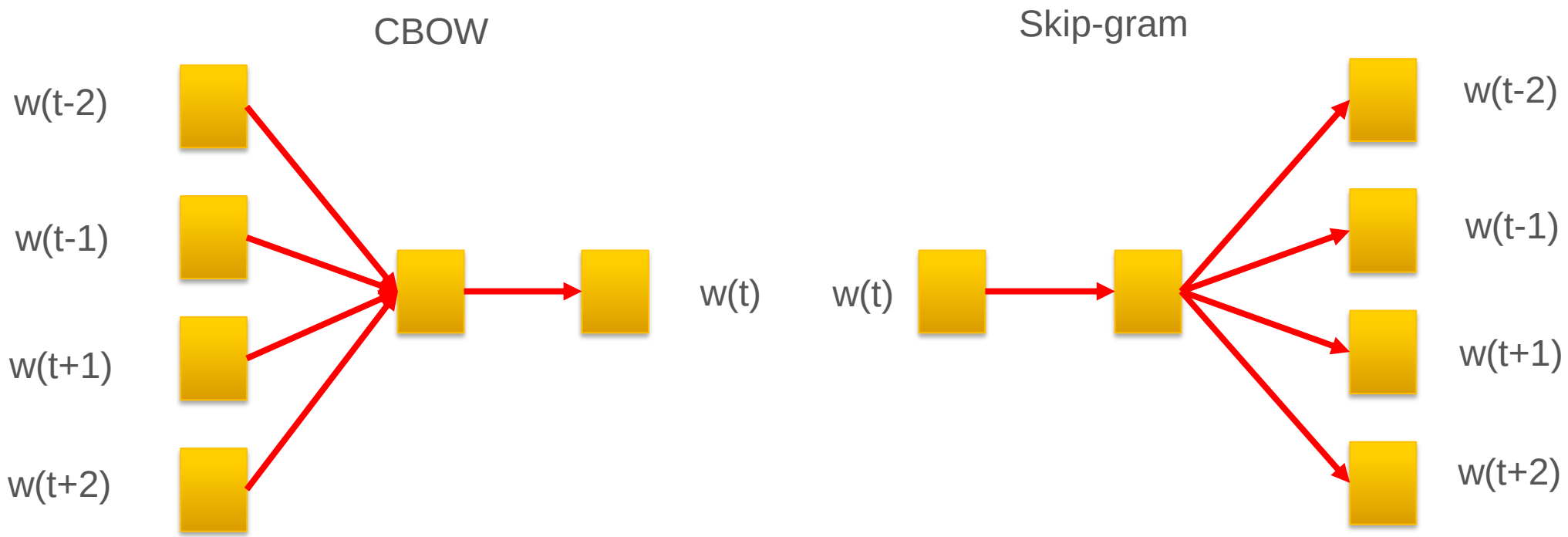
17-3:CBOW與Skip-gram

- CBOW與Skip-gram差異
- CBOW與Skip-gram的應用
- CBOW與Skip-gram可視化



CBOW與Skip-gram差異

- **CBOW與Skip-gram最大的差異在於**
  - **CBOW是用上下文去預測目標字**
  - **Skip-gram是用目標字去預測上下文**

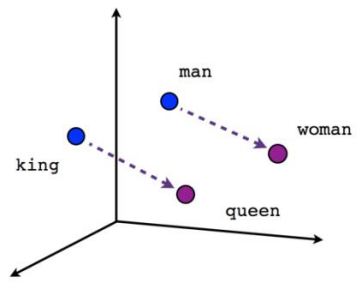


Skip-gram及CBOW比較

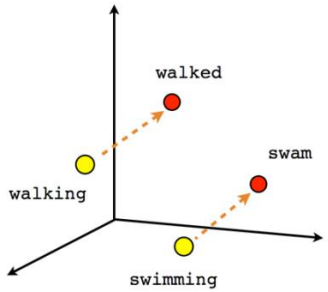
- 一般認為，Skip-gram在小資料集表現得比較好，尤其是當資料集內出現罕見字或是罕見詞的時候
- CBOW在訓練速度上比skip-gram來得快

CBOW與Skip-gram的應用

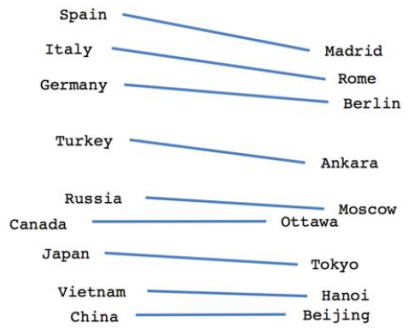
- **CBOW及Skip-gram訓練出來的字向量，可以拿來做類推**
  - **例如向量(”king”)-向量(”queen”)=向量(”man”)-向量(”woman”)**



Male-Female



Verb tense



Country-Capital

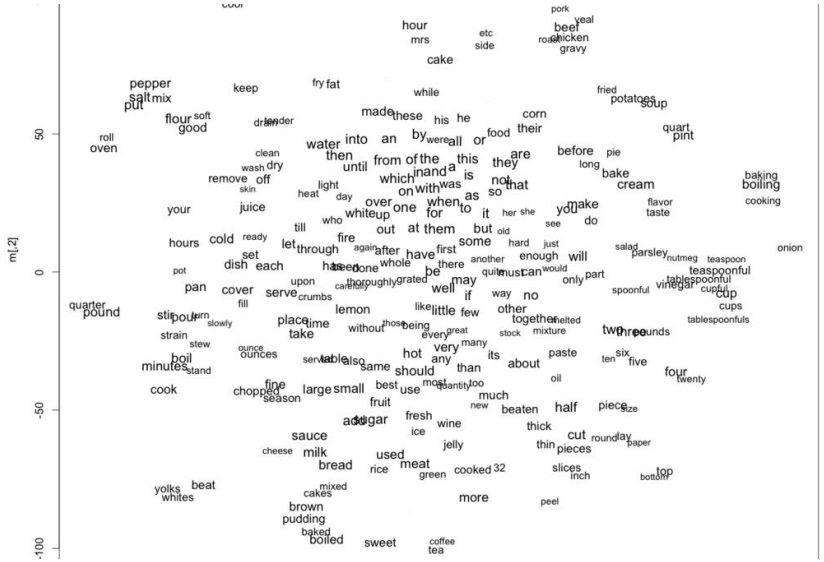
vector(king) – vector(queen) = vector(man) – vector(woman)

vector(walking) – vector(walked) = vector(swimming) – vector(swam)

vector(Spain) – vector(Italy) = vector(Madrid) – vector(Rome)

CBOW與Skip-gram可視化

- 由於CBOW與Skip-gram會把所有字壓到一個高維度的向量
  - 我們很難去觀察每個字在空間中的分布，因此有人提出我們可以用t-sne這個機器學習演算法，去將向量做降維，並在二維座標平面上可視化所有文字
  - t-sne演算法我們會在後面的課程提到



Demo 17-3

- CBOW模型建立
- Gensim套件實作CBOW
- 文字類推

線上Corelab

- 題目1： Gensim套件實作CBOW(基礎)
  - 使用Gensim完成CBOW模型
- 題目2： Gensim套件實作CBOW (進階)
  - 使用Gensim完成CBOW模型並做文字類推
- 題目3：CBOW模型建立
  - 使用TensorFlow完成CBOW模型

本章重點精華回顧

- CBOW介紹
- CBOW演算法
- CBOW與skip-gram

Lab:CBOW模型建立

- **Lab01: CBOW模型建立**
- **Lab02: Gensim套件實作CBOW**
- **Lab03: 文字類推**

Estimated time:
20 minutes


THANK YOU