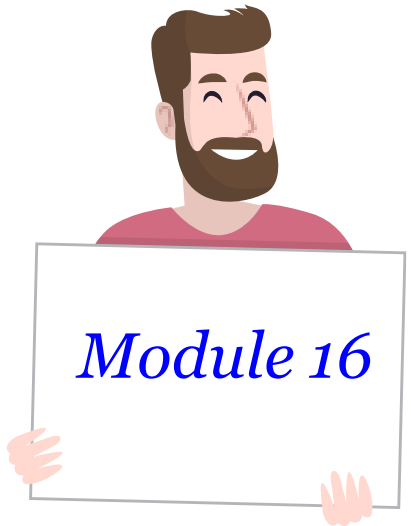
Skip-Gram模型介紹



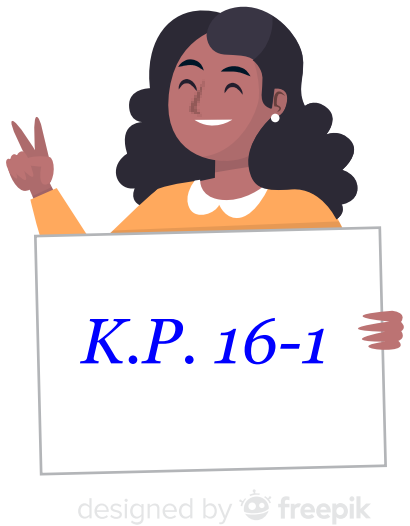
Estimated time:
45 min.

學習目標

- **16-1: Bag of words問題**
- **16-2: Skip-gram**
- **16-3: Skip-gram實作上的問題**

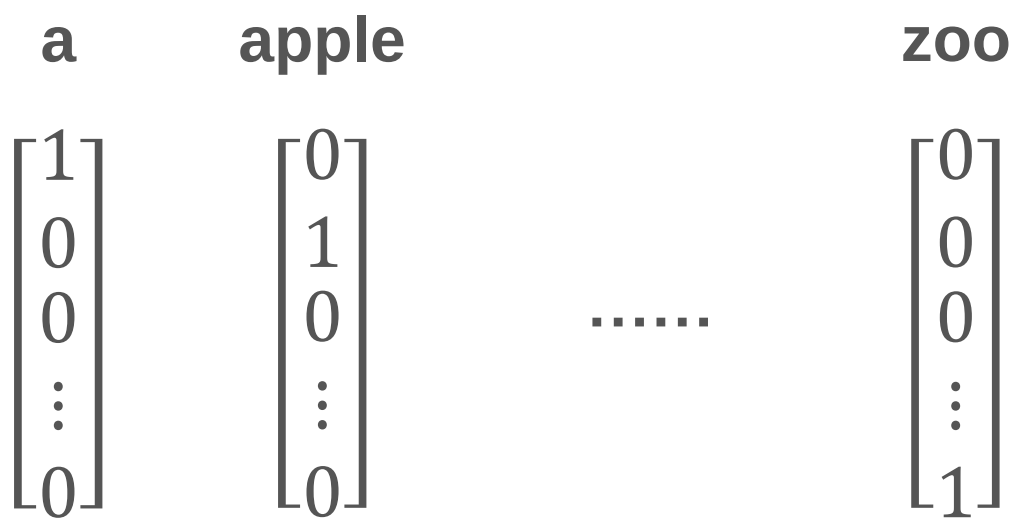
16-1: Bag of words問題

- Bag of words問題
- 期望Word2vec模型的特徵



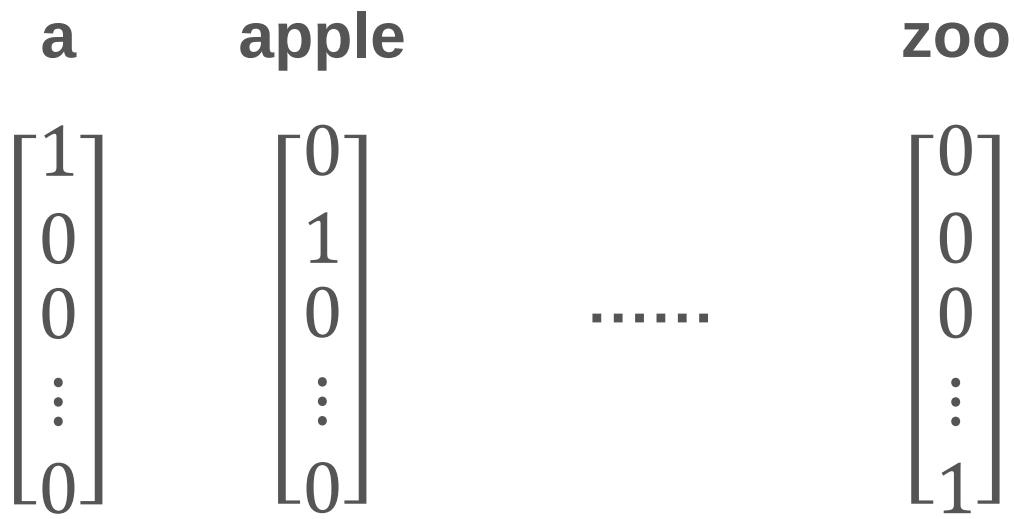
Bag of words問題

- **Bag of words雖然可以把文字很快的轉成向量，但對電腦來說還是無法理解文字的意思**
  - **單純把字變向量是不夠的**



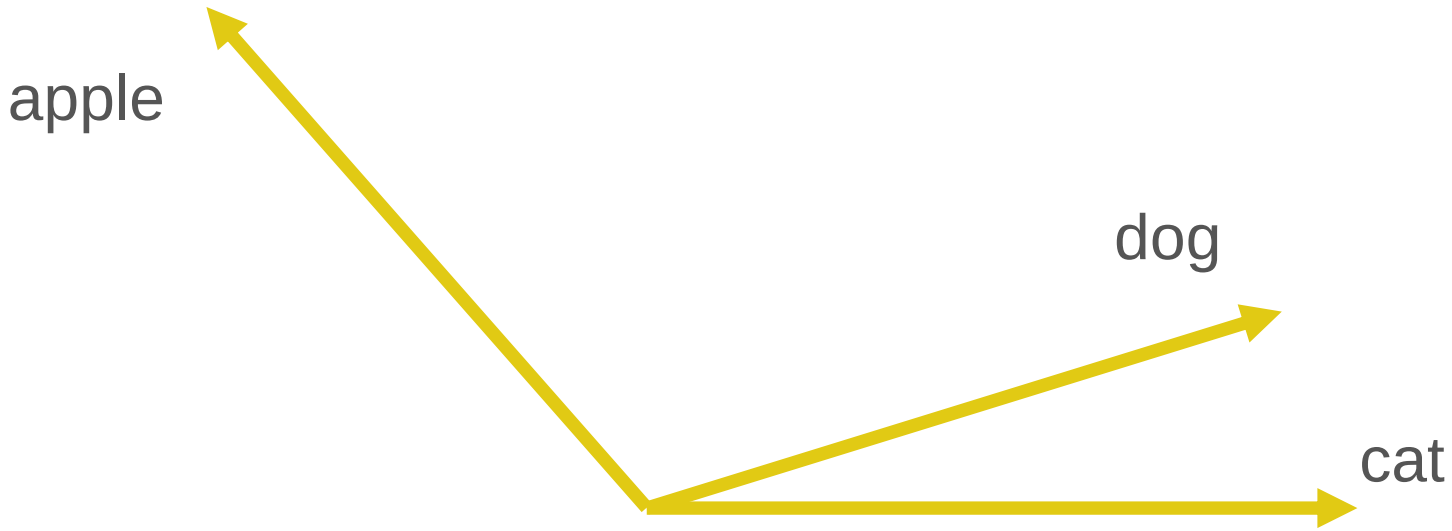
Bag of words問題

- **Bag of words模型的缺點有**
  - **相似的字之間的向量沒有關係，即電腦無法分辨哪些字是有關係的(例如，”Dog”跟“Cat”應該有要關係)**
  - **字向量的維度(字彙量)非常高，浪費空間外且大部分元素為0**



期望Word2vec模型的特徵

- 我們會希望如果能有一個Word2vec模型，它能夠將
  - 意思”相似”的字轉換成距離接近的字向量
  - 意思”相異”的字轉換成距離較遠的字向量



期望Word2vec模型的特徵

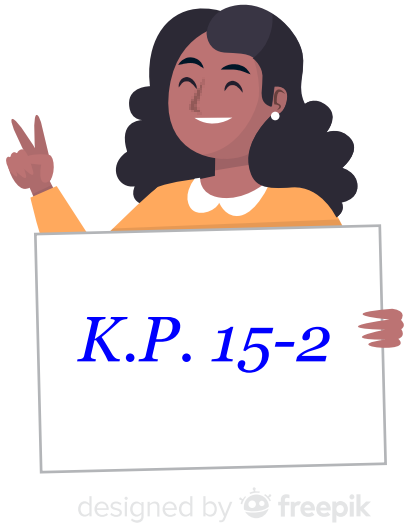
- 然而，如何定義出字之間意思”相似”以及”相異”這件事情很複雜
  - 有人提出根據字的前後文來判斷字是否相似
  - 只要任兩個字的前後文相似，此兩字應該有某種”相似”特性

I am ten years old

I am five years old

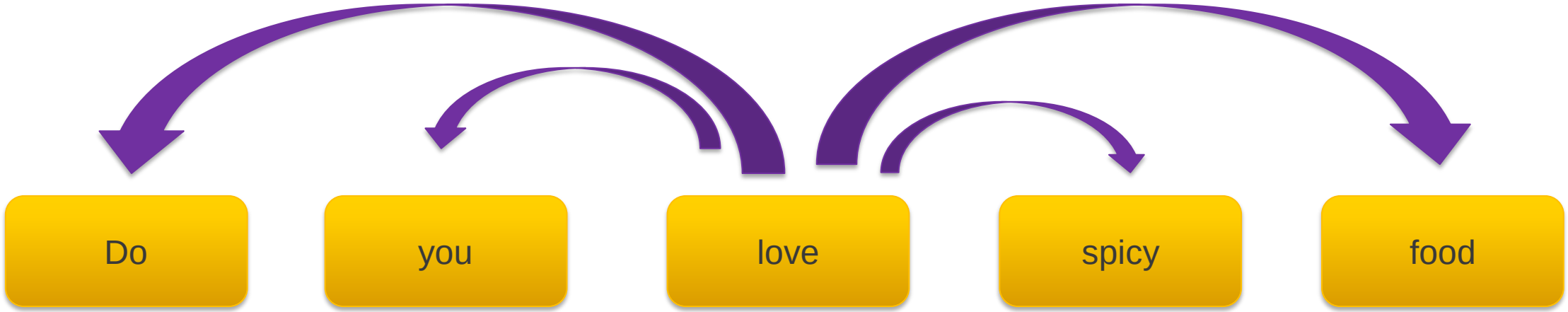
16-2: Skip-gram

- **Skip-gram介紹**
- **Skip-gram演算法**



Skip-gram介紹

- **Skip-gram是一種基於神經網路的word2vec模型**
  - 它的核心思想是給定目標字的情況下，去預測上下文的字
  - 換句話說，某些字如果常出現相同的上下字，那麼這些字可能意思相似



Skip-gram演算法

- 為了將每個字所出現對應的上下字蒐集起來，skip-gram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

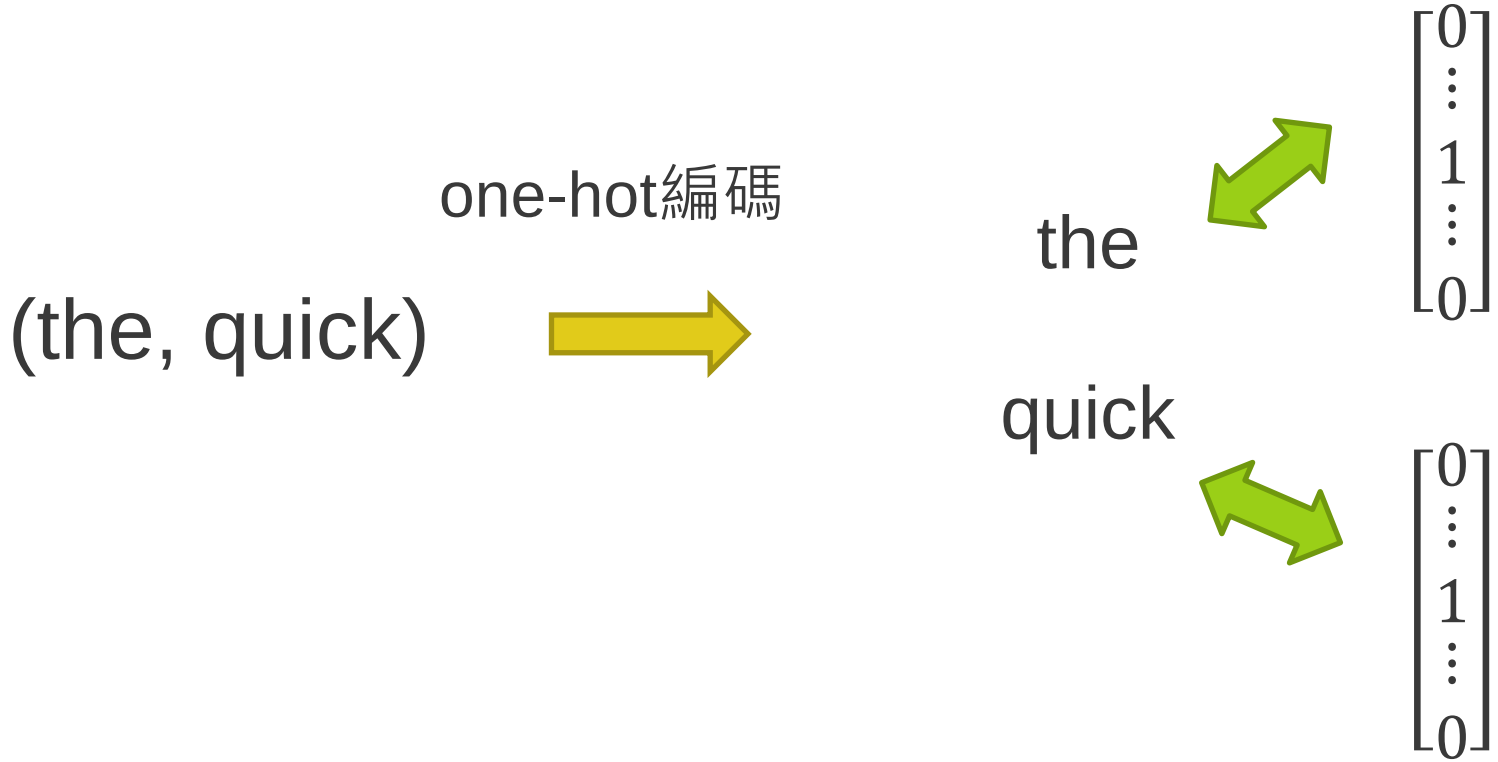
the quick brown fox jumps over the lazy dog. → (the, quick)
(the, brown)

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

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

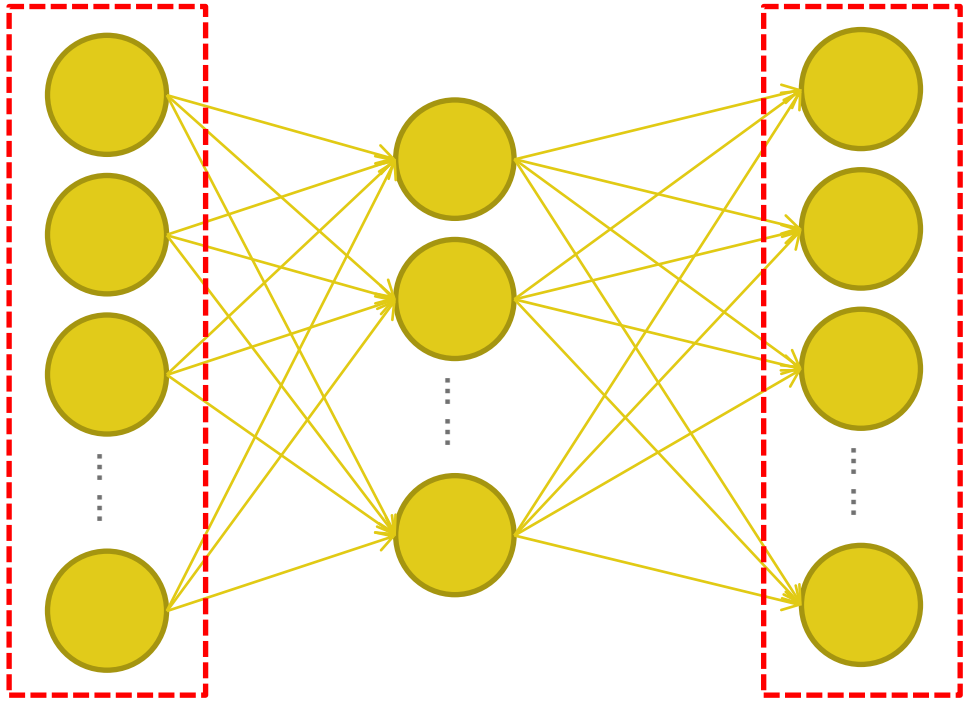
Skip-gram演算法

- 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



Skip-gram演算法

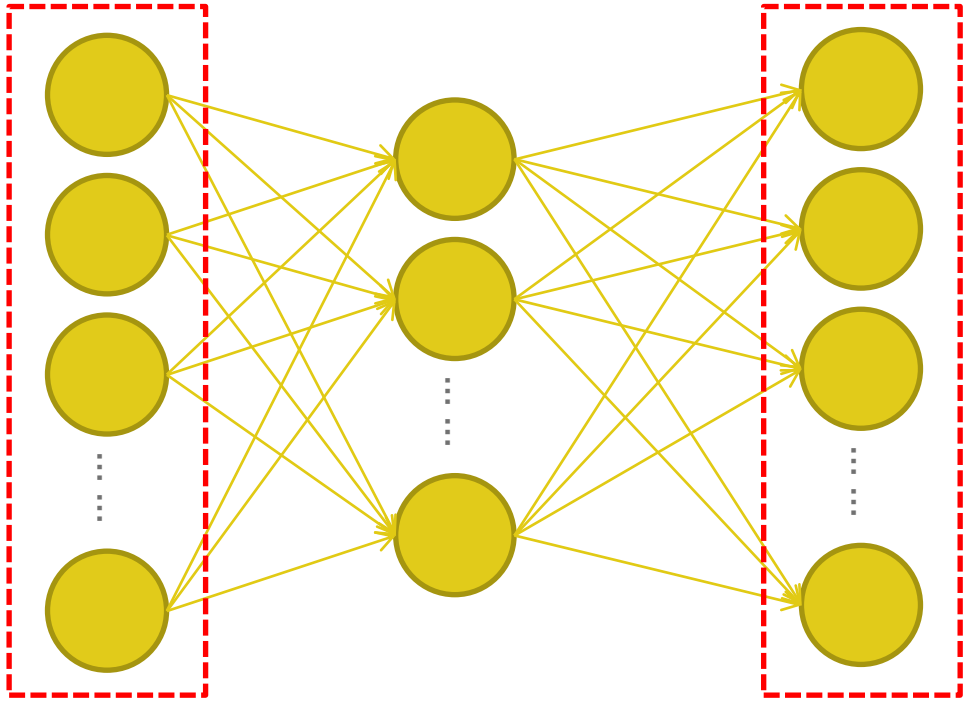
- **Skip-gram的做法主要是先建立一個只有一個隱藏層的神經網路**
  - **為了要將每個字one-hot編碼的結果輸入網路，此網路輸入層以及輸出層的神經元數量需為字彙量**



輸入層、輸出層之神經元數量為字彙量

Skip-gram演算法

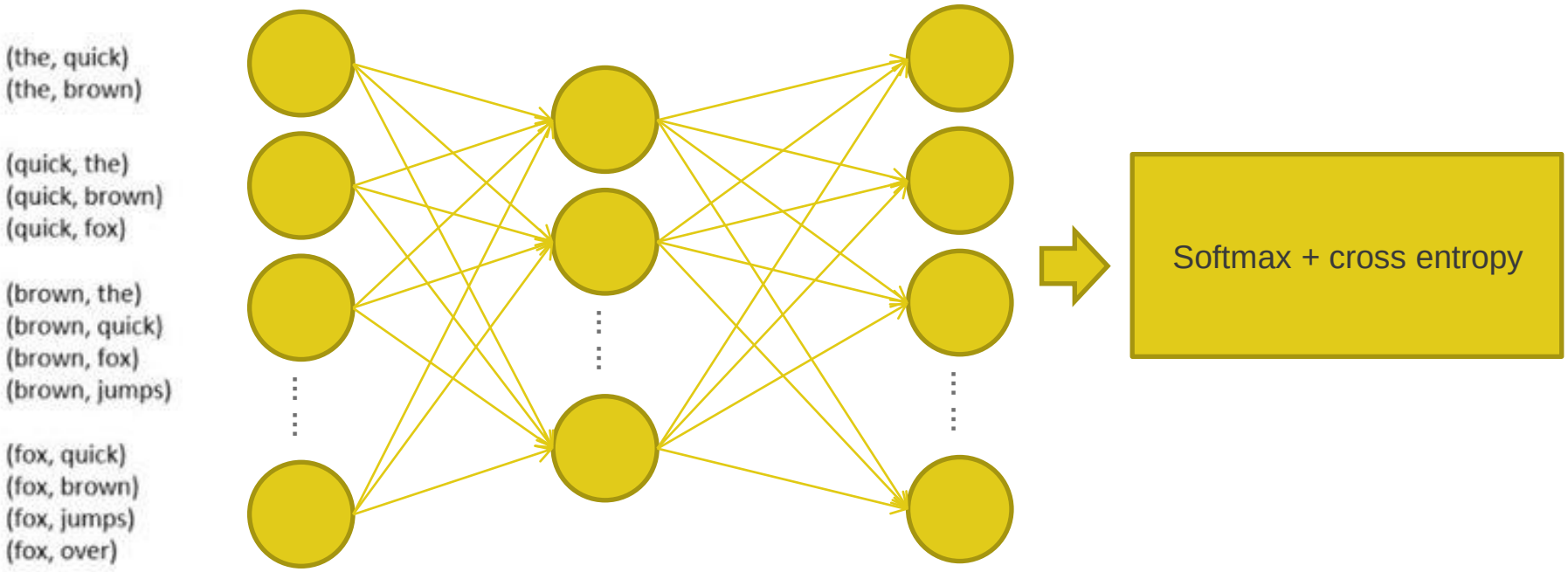
- 要特別注意的是，Skip-gram的隱藏層沒有激活函數，其餘計算方法跟DNN神經網路一樣
  - 這樣的設計是為了讓學到的字向量之間，會有線性的關係



輸入層、輸出層之神經元數量為字彙量

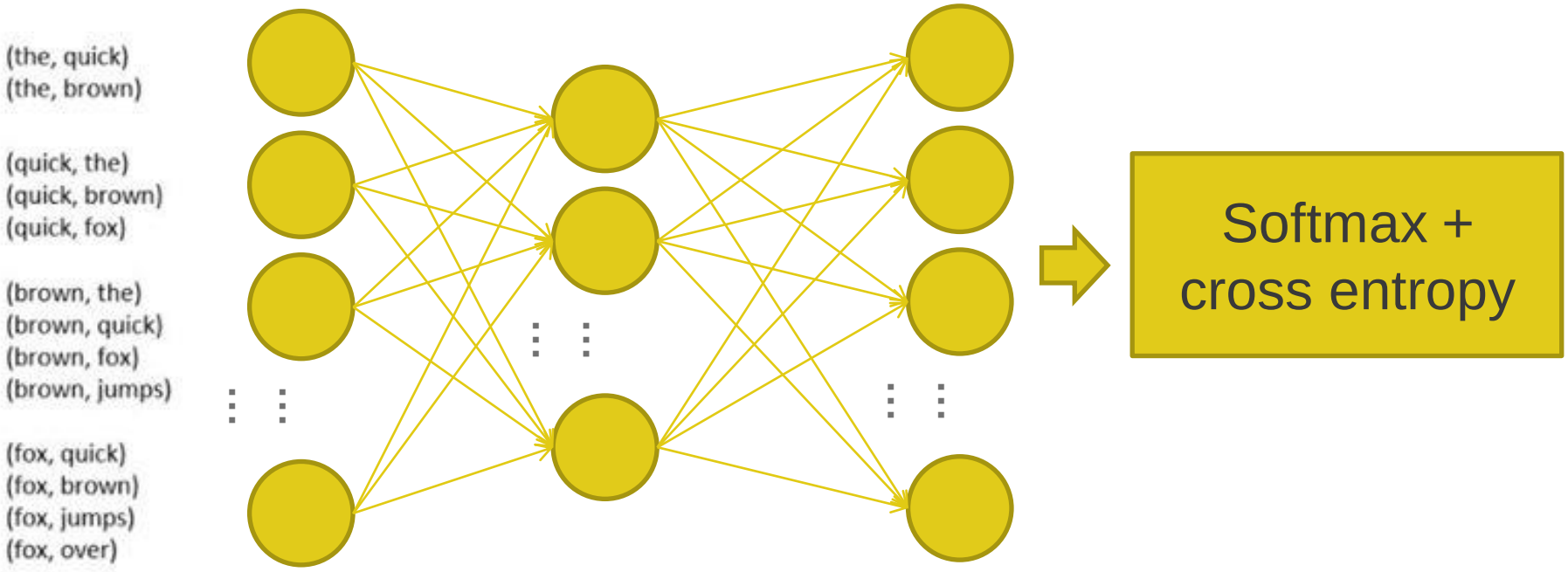
Skip-gram演算法

- 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
  - 每筆資料的輸入為第一個元素，期望輸出為第二個元素，例如(the, quick)這筆資料，輸入為”the”這個字，期望輸出為”quick”這個字



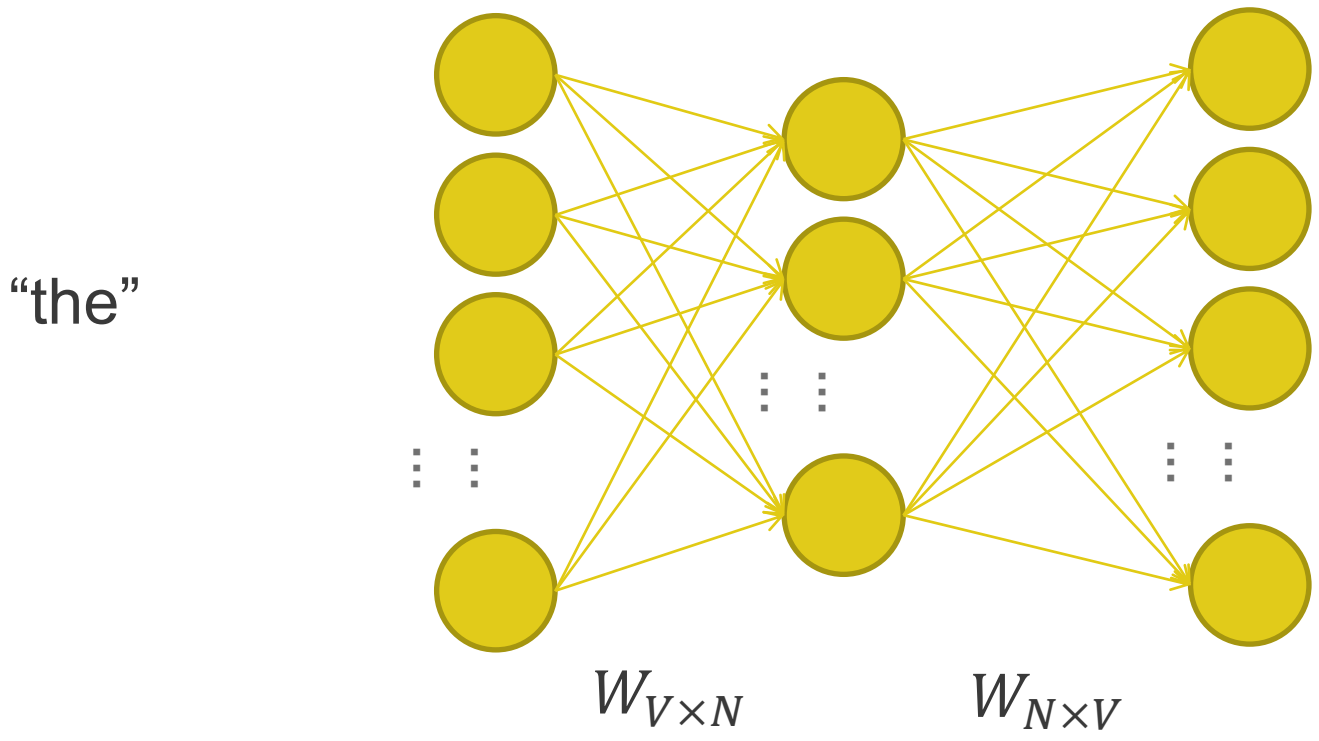
Skip-gram演算法

- **Skip-gram的訓練方法與DNN神經網路一樣，可以定義損失函數並經由優化去去調整網路裡的權重**



Skip-gram演算法

- 訓練完的網路會得到一組$W_{V\times N}$的參數，此參數稱為lookup table
  - 可以藉由lookup table去查詢任一個字的字向量，例如，我們把”the”的one-hot編碼輸入，得到的隱藏層則代表”the”這個字的字向量

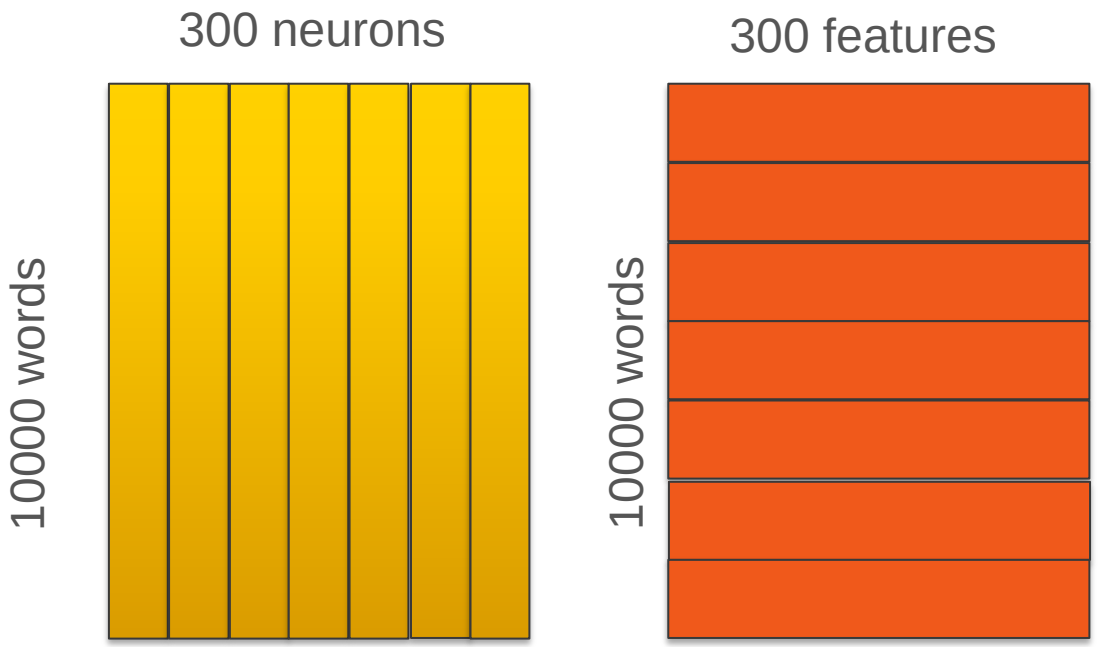


Skip-gram演算法

- $W_{V\times N}$這個矩陣被稱為lookup table是有原因的，因為一個one-hot編碼的向量與某矩陣相乘，就好像是把某個特定的row抽取出來
  - 我們可以把skip-gram網路裡$W_{V\times N}$的每個row視為不同的字對照的字向量

$$[0 \quad 0 \quad 1] * \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & 1 \\ 3 & 2 & 1 \end{bmatrix} = [3 \quad 2 \quad 1]$$

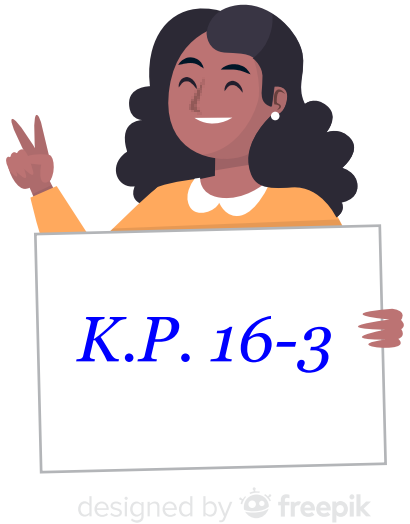
線性代數範例



假設字彙量為10000

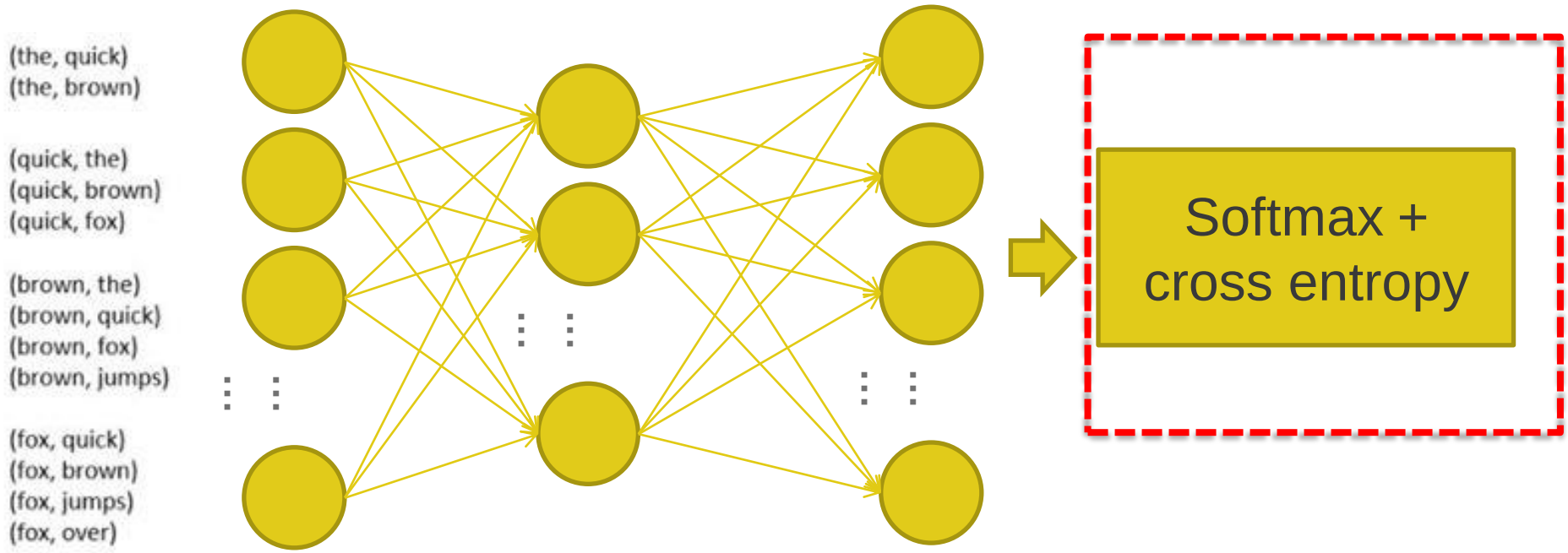
16-3: Skip-gram實作上的問題

- **Skip-gram實作上的問題**



Skip-gram實作上的問題

- **Skip-gram在訓練網路時，如果損失函數選擇為cross-entropy，會遇到前面的softmax層計算量非常大**
  - **因為輸出層的長度為字彙量，一個語料庫字彙量往往非常大**



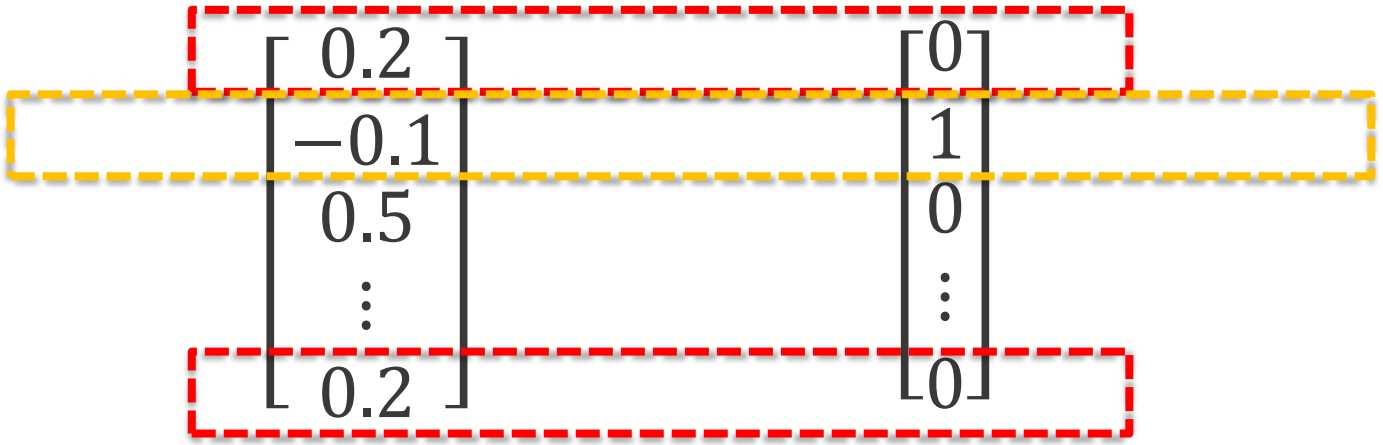
Skip-gram實作上的問題

- 當使用skip-gram時，softmax層的運算量會非常大，因為輸出層的神經元等於字彙量(很大)
  - 即因為下面式子k很大，所以分母加總的時候會非常久
  - 因此目前現有的作法是，為了加快softmax計算，我們常常使用簡化版的softmax，叫做sampled softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Skip-gram實作上的問題

- 負採樣(Negative sample)是一種將sampled softmax的方法，其概念是將標籤裡，取出不是答案的N個欄位以及答案欄位來做softmax
  - 減少要算的向量長度
  - N可以人為自行設定

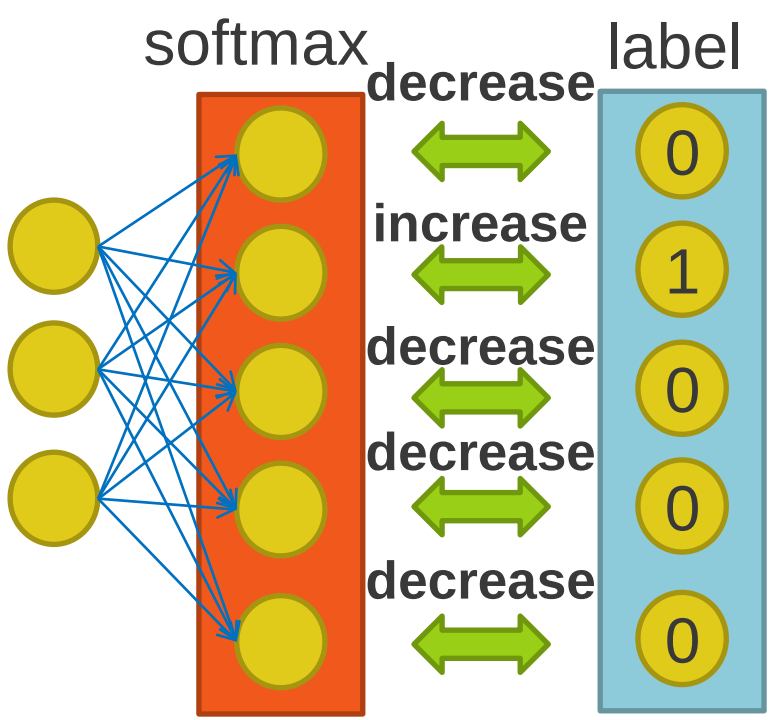


N=2

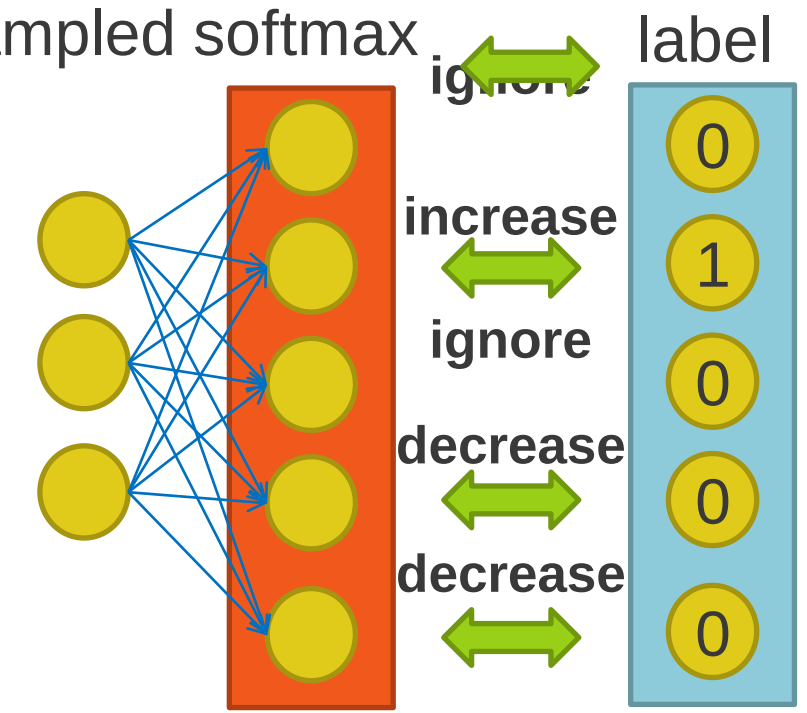
$$\begin{bmatrix} 0.2 \\ -0.1 \\ 0.2 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Skip-gram實作上的問題

- 正常softmax以及sample softmax的意義如下
  - 正常softmax是輸出層所有神經元均須更新參數，而sampled softmax只更新答案欄位以及不是答案欄位且被選取到的



正常softmax



sample softmax

Demo 16-3

- 產生word2vec訓練資料
- word2vec模型建立
- 可視化word2vec結果

線上Corelab

- 題目1： 將word2vec模型結果繪圖出來
- 題目2： 中文文字word2vec
  - 給予中文語料庫，實作word2vec
- 題目3：電影影評分類
  - 給予電影影評資料集，分類評論好或壞

本章重點精華回顧

- bag of words問題
- Word2vec目標
- skip-gram神經網路
- skip-gram實際上會遇到的問題

Lab: skip-gram模型介紹

- **Lab01: 產生word2vec訓練資料**
- **Lab02: word2vec模型建立**
- **Lab03: 可視化word2vec結果**

Estimated time:
20 minutes


THANK YOU