詞嵌入法 (Word Embedding)

處理文字資料的時候，bag-of-word、n-gram、TF-IDF 是常見的方法。這三個方法的共同特徵是「計算資料裡出現某些單詞的頻率」。這樣的方法處理小量文字資料效果滿好，然而如果面對龐大的文字資料，比如說文章甚至是書本，就會發現轉換出來的結果是一個充滿 0 的稀疏矩陣（Sparse Matrix）。

為什麼會是稀疏矩陣呢？舉例來說，假設現在有兩句話：「This is a sentence」、「You need to read that book」，我們來計算一下兩句話的單詞出現頻率：

	This	is	a	Sentence	You	need	to	read	that	book
第 1 句	1	1	1	1	0	0	0	0	0	0
第 2 句	0	0	0	0	1	1	1	1	1	1

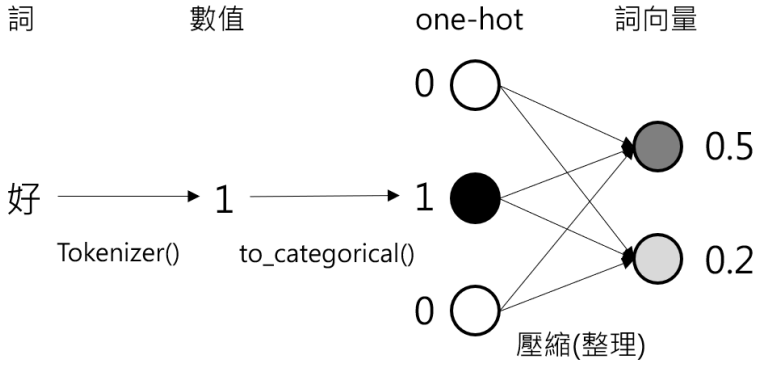
可以發現轉換後，矩陣內竟然有一半的位置都是 0。如果我們原始的文字資料當中，有很多罕見的單詞，那麼轉換之後會出現一個超級大矩陣，但是很多位置都是 0（大部分的文字內容都不會用罕見字）。這顯然是一個沒有效率的方式甚至可能會導致記憶體不足而建模失敗。雖然 Python 中的 Sparse Matrix 資料結構也可以壓縮儲存空間，不過處理起來較花時間，仍然不是好的做法。

為了解決上述問題，我們可以使用詞嵌入法。這個方法是如何用少少的數值矩陣來表示龐大的文字呢？我們先看一個生活的範例：我們現在有一個塞滿滿東西的背包，然後要找背包裡面的一支筆。該怎麼找？雖然把背包的東西全部倒出來能找到筆，但是散落一地的東西會變得很佔位，那麼有什麼方法能讓下一次找筆時，不用再一次把東西倒出來呢？有的！那就是整理你的背包，把東西倒出來後應該有規律地將東西重新放回背包中，例如把「文具」統一放在背包的前袋中，這樣下次要找

筆時，就知道要去背包的前袋裡找。由於經過了很好的整理，所以能很快地找出東西放在哪。

詞嵌入法便是運用了上述原理，將文字資料轉成 one-hot 格式後，再把這些資料進行壓縮，並且在降低資料維度的同時，使用高度有序的方法，重新分配詞在空間中的位置，製造出低維的詞向量。

不過，每個人整理背包的方式不一樣，例如有些人可能會按照物品的屬性來整理：文具放在前袋、3C 產品放在側袋…；有些人可能會用物品是否常用來整理：常用的放前袋、不常用的放後袋…。不同的擺放方式適合不同的應用場景，像是找筆的話就可能較適合用第一種整理方式。而詞嵌入法也是，可以用不一樣的方式來壓縮（整理）資料，好的做法是依照不同任務，採用適合的整理方式。換句話說，最好能把這個高維到低維的轉換過程也作為一層網路，讓它自行從訓練資料中學習。以下用一個簡化的例子來說明如何做到這件事：首先假設單詞量為 3、對照表為 {' 你 ': 0, ' 好 ': 1, ' 嗎 ': 2}，那麼就能用以下網路架構將「好」這個詞轉換成 2 維的詞向量：



接下來我們實際看看這個網路內部的運算。這裡令輸入資料（單詞）為 x、權重（壓縮方式）為 w、輸出資料（詞向量）為 y，並以「好」這個詞為例代入網路：

$$
\begin{array}{ccccc}
x & & w & & y \\
 & & [[0.1,\ 0.3], & & \\
[0,1,0] & \text{dot} & [0.5,\ 0.2], & = & [0.5,\ 0.2] \\
 & & [-0.4,\ 0.6]] & & \\
\text{好} & & \text{權重} & & \text{詞向量}
\end{array}
$$

從以上的運算結果我們可以發現，由於 one-hot 編碼的特性，所以 x 和 w 做點積運算等於「用 x 的類別作為索引對 w 取值」：

▼ **驗證用類別作為索引**

```
import numpy as np
x=np.array([0,1,0])
w=np.array([0.1,0.3],[0.5,0.2],[-0.4,0.6])
print(np.dot(x,w))
print(w[1])
```

▼ **輸出**

```
array([0.5, 0.2])
array([0.5, 0.2])
```

也就是說其實權重 w 中第 0 列的向量，便是第 0 個詞的詞向量；而第 1 列就是第 1 個詞向量，後面依此類推：

- [0.1,0.3]，第 0 個詞，也就是「你」的詞向量
- [0.5,0.2]，第 1 個詞，也就是「好」的詞向量
- [-0.4,0.6]，第 2 個詞，也就是「嗎」的詞向量

了解這個道理後，我們得知如果用索引取值，取代數值轉 One-hot encoding 再做點積運算，將能大幅縮短網路正向傳播的時間。Keras 早就想到了這點，只要使用嵌入層（Embedding layer）便能達到這樣的效果，通常這一層要放在網路的第一層，並傳入整數數值（一個數值代表一個詞），接著它就會將數值轉為指定維度的向量囉！