Distributed Representations of Words and Phrases and their Compositionality

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/925cff5b-0973-47e4-be4d-6766fd79fda8/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

1. Introduction

- Distributed representation

  強調的是每個詞都對應著一個低維（例如100維）的向量，這個詞的“詞義”或者這個詞與其他詞的關係就蘊含在這個向量的不同維度中，例如題圖中例子，但是我們一般並不知道每個維度對應著什麼含義。和one hot encoding形成對比。

- Skip-gram
  - ▼ Original skip gram

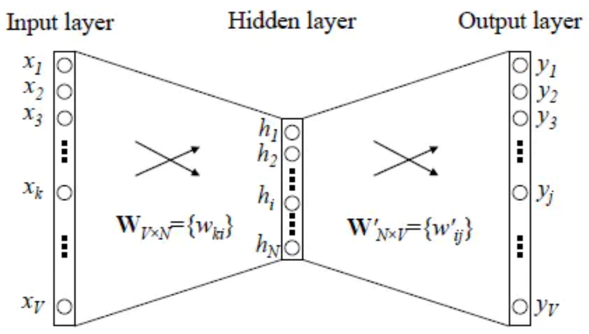
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

Skip-gram 就是要maximize這個式子

假設有T個training words, c 是training context的大小。c越大就有越多的training examples ，但相對的訓練時間就比較長，而Skip-gram的目標就是maximize上面的式子。

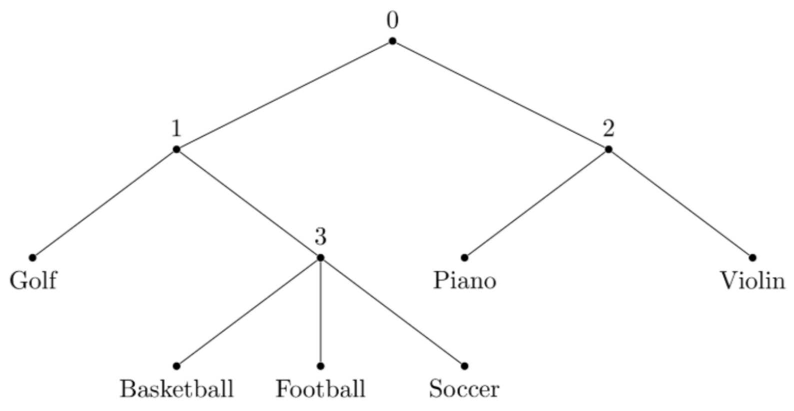
$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^\top v_{w_I}\right)}$$

vw 代表著 "input" vector representations of w, v'w 則代表 "output" vector representations of w. "input" vector representation 則是 W 中一個row的weight. 而"output" vector representation代表W'中一個column的weight.



▼ Hierarchical Softmax

原先softmax需要對所有vocabulary加總，很花時間。



Corpus Tree Example

The probability of the word Football will be

$$P^h_\theta(w = \text{Football}) = P^h_\theta(0 \to 1)P^h_\theta(1 \to 3)P^h_\theta(3 \to \text{Football})$$

走左子樹的機率:

$$p(n, left) = \frac{1}{1+e^{-x}} = \sigma(x)$$

走右子樹的機率:

$$p(n, right) = 1 - \frac{1}{1+e^{-x}} = \frac{e^{-x}}{1+e^{-x}} = \frac{1}{1+e^{x}} = \sigma(-x)$$

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \text{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^{\top} v_{w_I} \right)$$

而在這邊我們判斷走左或右子樹的機率，是根據到目前這個node，與原先的word的cosine similarity。再將機率連成得到的機率。

▼ Negative sample

要解決的問題是: 當我們採取一個pair的時候，例如(fox, quick)。則quick的概率為1，其他的為0，透過negative sample，我們隨機採取幾個negative的詞，來更新我們的頻率。

將原本要算的 $logP(w_O|w_I)$ 代換成

$$\log \sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^{\top} v_{w_I}) \right]$$

第一項的意思根據 "input" 以及 "output" vector 的 cosine similarity。而後面那項的意義就是避免noise抽取出來的word的vector與原先的"input" vector越不像越好。有點像變成二分類的問題，要把noise跟原本的分布拉的越開越好。

在實作上，我們根據字詞出現的頻率來選擇negative sample。每個字被選擇的機率為

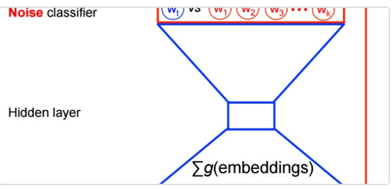
$$P(w_i) = \frac{f(w_i)^{0.75}}{\sum_{j=0}^{n} f(w_i)^{0.75}}$$

其中0.75是來自實驗的結果。

[译] Noise Contrastive Estimation

英文原文来自于我自己的博客，感谢自然语言处理百科全书的邀稿。简介在一个以 \theta 为参数的神经网络语言模型中，给予一定的语境 h ，我们通常使用softmax函数来预测下一个词的分布。这里我们把预测下一个词的条...

知 https://zhuanlan.zhihu.com/p/76568362



似然值→

如何理解似然函数?

似然函数在统计推断中意义重大，但是不是很理解为什么定义是L(θ|x)=f(x|θ)。我们在似然函数和概率密度...

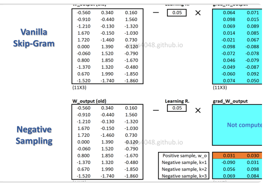
知 https://www.zhihu.com/question/54082000

NCE notes

Optimize Computational Efficiency of Skip-Gram with Negative Sampling | Pythonic Excursions

The code snippet assumes Anaconda 5.2.0 version of Python virtual environment Acknowledgement The materials on this post are based the on five NLP papers, Distributed Representations of Words and Phrases and their Compositionality (Mikolov et al., 2013), word2vec Parameter Learning Explained (Rong, 2014), Distributed

https://aegis4048.github.io/optimize_computational_efficiency_of_skip-gram_with_negative_sampling

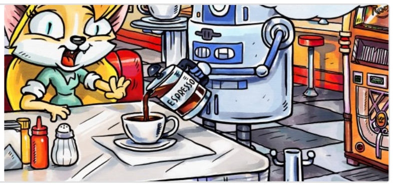


這篇很清楚→

NLP 102: Negative Sampling and GloVe

One way to generate a good quality word embedding from a corpus is using Word2Vec - CBOW or Skip-gram model. Both models have a few things in common: The training samples consisted of a pair of words selected based on proximity of occurrence. The last layer in the network was

https://towardsdatascience.com/nlp-101-negative-sampling-and-glove-936c88f3bc68



▼ subsampling of frequent words

認為少出現的字彙相較於常出現的字彙來說，包含著更多的資訊量，(Taiwan vs The)，所以為了要平衡各種字彙不同的頻率，在train set中，每個字有一定的機率會被捨棄掉，機率如下

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

f是word出現的頻率，t是threshold