Model on NLP

Todo List

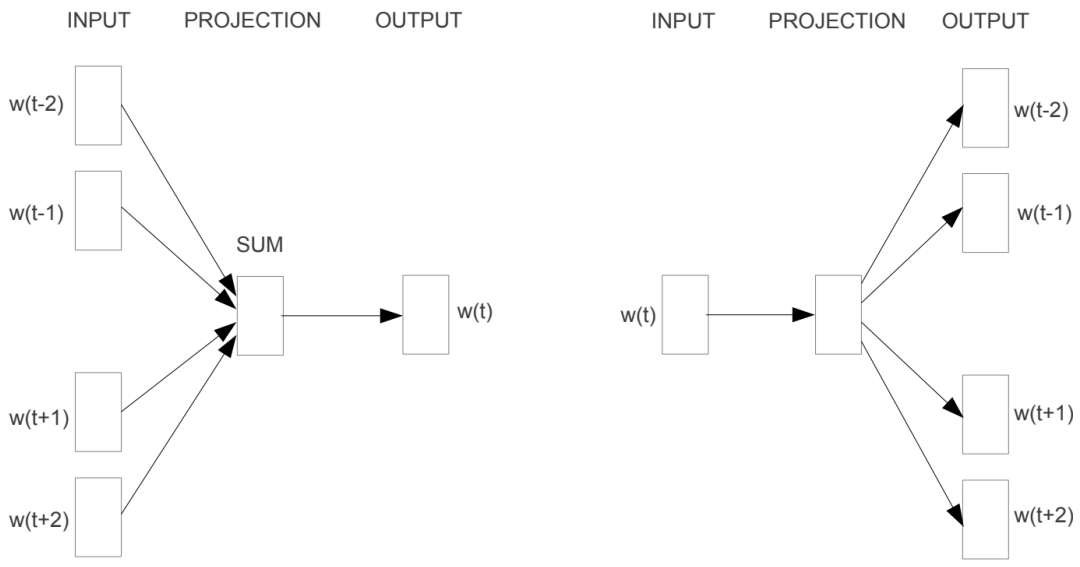
Word2vec

Contribution

1. Word2vec 是一种**考虑上下文，维度少，速度快的 Embedding 方法**，适用于各种 NLP 任务；
2. Word2vec 在词和向量之间是一个一对一的关系，对于**一词多义的问题**它是没有办法解决的；

Notes

1. Word2vec 分为 **CBOW** 和 **Skip-gram** 模型。**CBOW 模型为根据单词的上下文预测当前词的可能性**；**Skip-gram 模型恰好相反，根据当前词预测上下文的可能性。**两种模型相比，Skip-gram的学校效果会好一些，它对生僻词的处理更好，但训练花费的时间也会更多一些。两种模型的结构如下所示：



CBOW **Skip-gram**

2. 解决 Word2Vec 中 softmax 计算开销巨大的问题：

   (1) **Hierarchical Softmax**：使用哈夫曼树结构来代替 softmax，每一个树的非叶子节点都是一个二分类问题；

   (2) **Negative Sampling**: 使用负采样的方法来代替 softmax，随机采样几个不在 window 中的词，训练的目标变为期望在 window 内的词出现的概率越高越好，而负采样得到的词出现的概率越小越好；

Code Implementation

1. $\sigma(x)$ 的近似计算：$sigmoid$ 函数在 $x=0$ 附近变化剧烈，往两边趋近平缓，故 K 等分存储函数值，每次访问数组近似取值，以减少计算量；
2. 词典的存储：使用哈希表存储词典，地址的冲突解决使用线性探测的开放定址法；
3. 低频词的处理：对低频词在构建词表时直接进行删除，并且为了计算效率，在词表中单词超过容量的 70%时，即清理低频词 (当然，这有一定概率误删部分低频词)；
4. 高频词的处理：对高频词超过一定词频后，会以一定概率舍弃，具体公式如下

$$prob(w)=1-\left(\sqrt{\frac{t}{f(w)}}+\frac{t}{f(w)}\right)$$

   其中 $f(w)$ 为词频；

Links

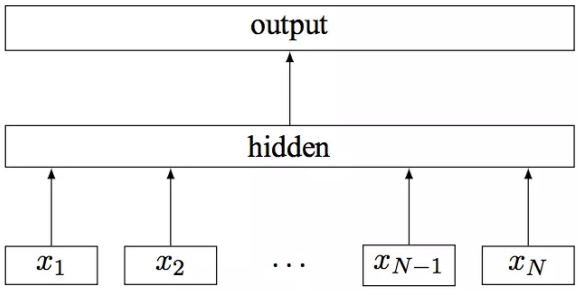
- 论文链接：
  Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
  Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." *International conference on machine learning*. 2014.
- Gensim 库：RaRe-Technologies/gensim: Topic Modelling for Humans (github.com)
- 大佬的 Word2Vec 讲解：word2vec 中的数学原理详解（一）目录和前言*peghoty-CSDN博客*
  word2vec数学原理
- C 代码实现：dav/word2vec (github.com)

FastText

Contribution

Notes

1. word2vec 等工作，是以词汇表中的独立单词作为基本单元来进行训练的，存在如下问题：
    - 低频词、罕见词组，未能得到充分训练，甚至是与最终结果无关的；
    - OOV的问题；
    - 英语中的词形变换问题；
2. FastText 架构图：



    FastText 的架构和 word2vec 的 CBOW 模型十分相似，都是对输入的 Embedding 做平均后使用 Hierarchical Softmax输出。但不同的是，CBOW 输出的是中间词的 Embedding，而 FastText 输出的则是分类，在输入端使用了 n-gram；

3.

Links