**Word2Vec**

**1.Word2Vec的原理或者什么是word2vec：简单来说，就是将一个个单词转换为包含信息的向量。具体来说， 就是通过skip-gram模型或者CBOW神经网络模型对词向量进行训练，得到高质量的分布式词向量表达，这些词向量获取了单词之间的语法和语义关系。**

**2.Word2Vec中加速词向量训练过程的技巧（tips）？**

**Hierarchical softmax：构建一棵哈夫曼二叉树，不用计算N个样本的概率，只需要计算log(N)个**

**Negative sampling：**

在样本中根据**概率随机挑选出k个负样本**，**将上下文的单词作为正样本，将多分类问题转换为二分类问题**，通过正样本的对数概率和负样本的对数负概率之和作为损失函数求解。

在训练过程中，使用了一种简化的NCE变体（NEG），两者的差别：**NEG只需要采用几个负样本，**而NCE需要噪声样本和样本分布；NCE最大化softmax的对数概率。

**细节：**NEG采样时需要每个单词的概率p(w),先对概率取**3/4**次方，再平均化（/N）,作为单词采样频率的结果更好。

**3.什么是负采样（negative sampling）？**

1）Word2Vec两个算法模型的原理是什么，网络结构怎么画？

（2）网络输入输出是什么？隐藏层的激活函数是什么？输出层的激活函数是什么？

（3）目标函数/损失函数是什么？

（4）Word2Vec如何获取词向量？

（5）推导一下Word2Vec参数如何更新？

（6）Word2Vec的两个模型哪个效果好哪个速度快？为什么？

（7）Word2Vec加速训练的方法有哪些？

（8）介绍下Negative Sampling，对词频低的和词频高的单词有什么影响？为什么？

（9）Word2Vec和隐狄利克雷模型(LDA)有什么区别与联系？

以上问题可以通过本文和参考文章找到答案，这里不再详细解答。

（10）介绍下Hierarchical Softmax的计算过程，怎么把 Huffman 放到网络中的？参数是如何更新的？对词频低的和词频高的单词有什么影响？为什么？

Hierarchical Softmax利用了Huffman树依据词频建树，词频大的节点离根节点较近，词频低的节点离根节点较远，距离远参数数量就多，在训练的过程中，低频词的路径上的参数能够得到更多的训练，所以效果会更好。

（11）Word2Vec有哪些参数，有没有什么调参的建议？

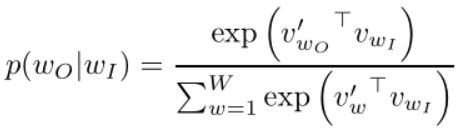
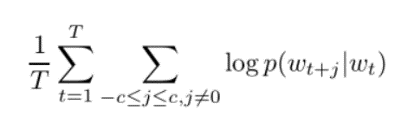
* Skip-Gram 的速度比CBOW慢一点，小数据集中对低频次的效果更好；
* Sub-Sampling Frequent Words可以同时提高算法的速度和精度，Sample 建议取值为  ；
* Hierarchical Softmax对低词频的更友好；
* Negative Sampling对高词频更友好；
* 向量维度一般越高越好，但也不绝对；
* Window Size，Skip-Gram一般10左右，CBOW一般为5左右。

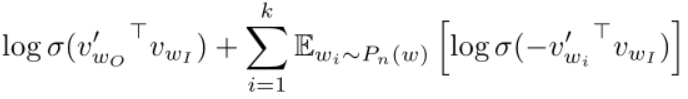
（12）Word2Vec有哪些局限性？

Word2Vec作为一个简单易用的算法，其也包含了很多局限性：

* Word2Vec只考虑到上下文信息，而忽略的全局信息；
* Word2Vec只考虑了上下文的共现性，而忽略的了彼此之间的顺序性；

**Skip-gram model?**

**最大化对数概率：**

****

**目录：**

**介绍**

**CBOW模型**

**Skip-gram模型**

**Result**

介绍：

Word2Vec即单词的向量空间表示，是Google公司Tomas Mikolov等人提出的

之前的许多NLP系统都用词汇中的索引(index)来表示单词，单词之间没有相似度联系。这种单词的表示方法虽然比较简洁，在小数据训练中具有鲁棒性。但是这种简单的表示在许多任务中有局限性，其单词的表示 质量低导致性能局限。

目前，一种主流的方法就是使用单词的分布式表示。单词的分布式表示，相似单词不仅趋于彼此接近，而且这些词可以具有多个相似度，例如词义和语法。单词表示的相似性超出了简单句法规律，可以在词向量空间中进行简单的几何运算，例如vector(”King”) - vector(”Man”) + vector(”Woman”) = vector(Queen)

在本文中，提出了两种模型结构，有效地保留单词之间的线性规则，提高了单词向量表示的准确率。

单词向量显著改善和简化许多NLP应用问题

Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA).

单词的分布式表达由神经网络学习得到，其单词之间的线性保持比LSA表现更好，LDA在大数据集的应用中，需要花费更大的计算量。

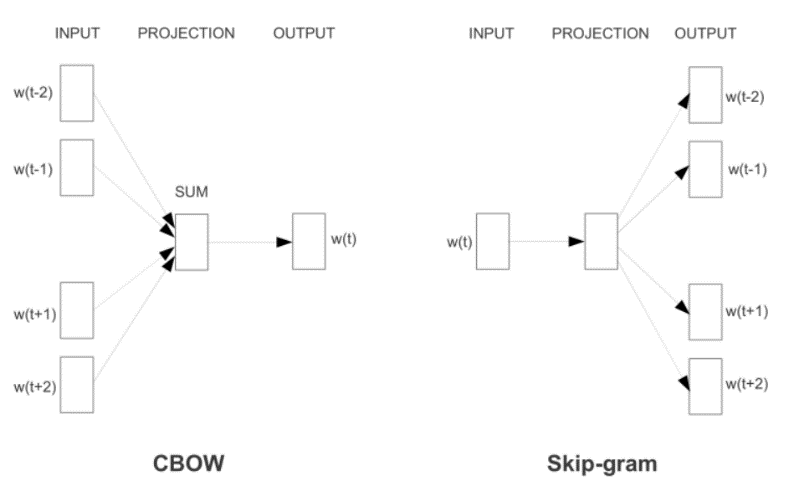
所有神经网络的模型训练都采用SGD和BP实现

Hierarchical softmax

两种模型：

本文提出的两种学习单词分布式表示的模型结构，它们可能无法像神经网络那样精确地表示数据，但减少了计算的复杂度，可以有效地对更多数据进行训练。

神经网络语言模型目前主流的训练模式就是：先使用简单模型学习连续单词向量，然后再用这些具有分布式表示的单词向量训练神经网络。



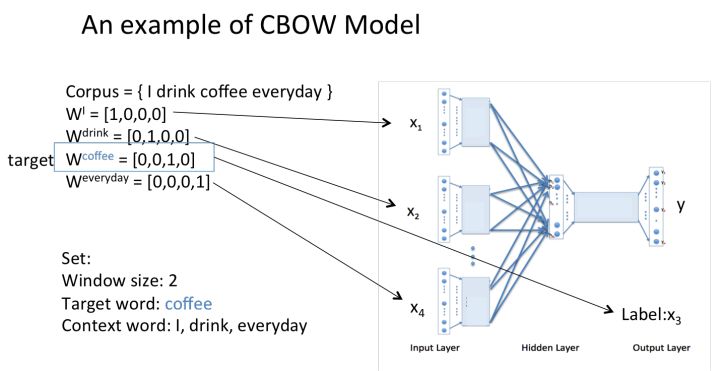
模型一：continuous bag-of-words model （CBOW）词袋模型 根据上下文预测中心词

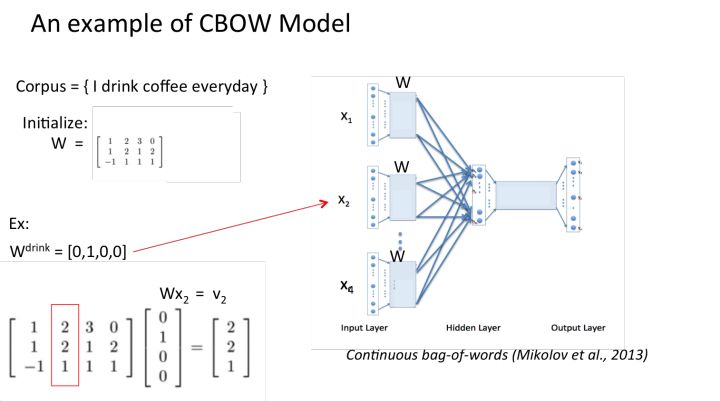
CBOW模型为三层结构：输入层，映射层和输出层。模型的输入为某一个特征词上下文相关的词向量，输出为该特征词的词向量。值得注意的是，上下文的单词是平等的，其顺序不影响映射结果。在本文测试的结果中，4个future words + 4个history words 构成的上下文结构可以达到 best performance。

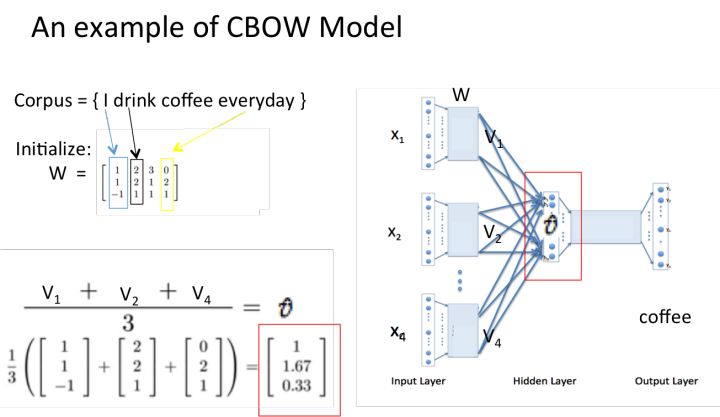
模型训练的时间复杂度是：

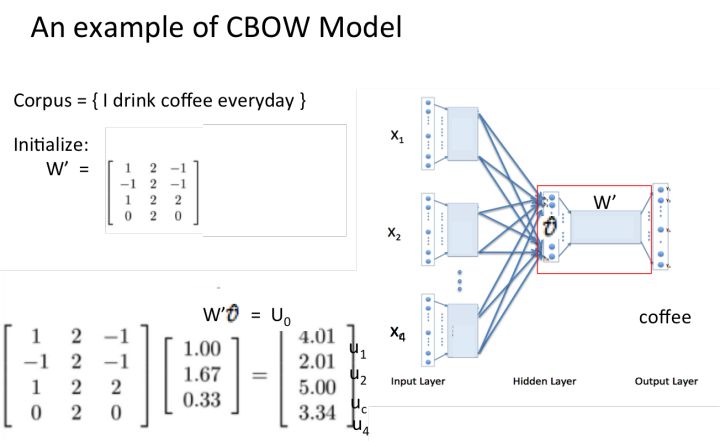
Q = N × D + D × log 2 (V )

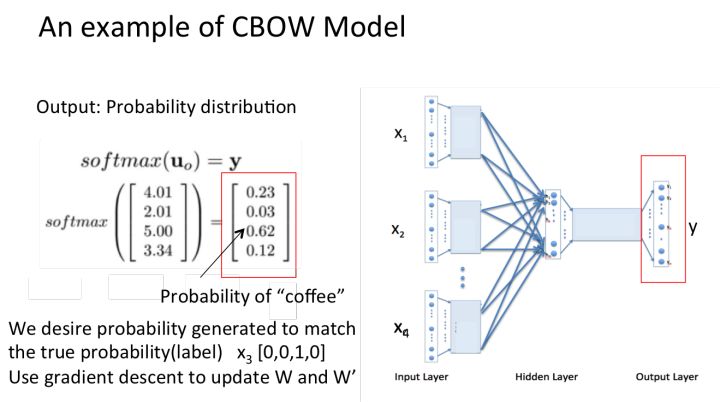
其中，N为输入层神经元的个数，D为单词向量的维度，V为词汇量。











模型二：continuous skip-gram model 给定中心词预测周围词

Skip-gram模型结构也是三层,不过它恰好与CBOW模型相反，其输入为每一中心词的词向量，输出为其上下文词向量。

模型训练的时间复杂度是：

Q = C × (D + D × log 2 (V ))

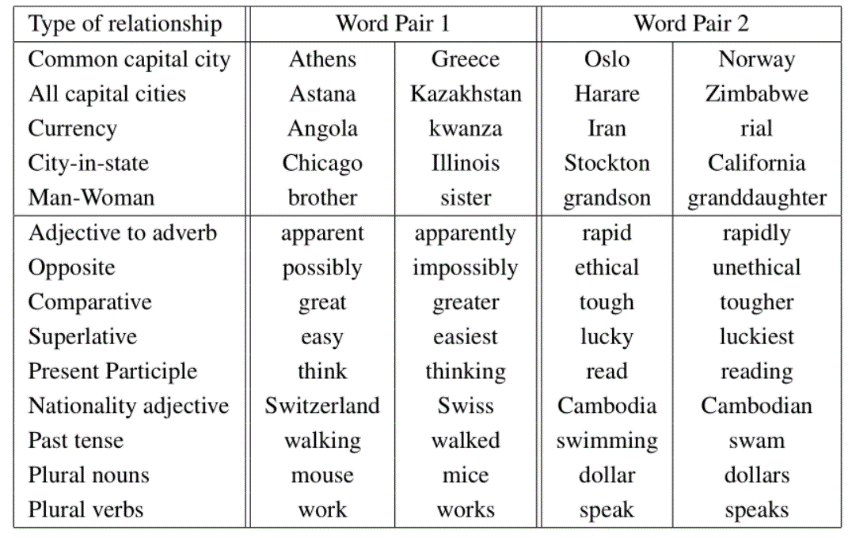
其中，C为单词间最大距离，D为单词向量维度，V为词汇量大小。

**Result：**

由两种模型训练得到的词向量通过对比示例单词和与之最接近的单词，来检测其语义(semantic)和语法(syntactic)的表达，例如：big-biggest和man-women。

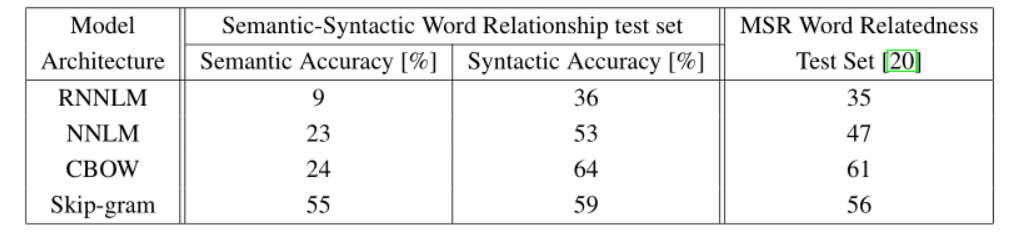
这些对比问题可以通过用单词向量表示执行简单的代数运算来回答：例如vector X = vector(”biggest”)−vector(”big”)+vector(”small”) =>vector（“smallest”）.

下图是5种语义和9种语法测试例子：



使用更多的数据和更高的单词向量维度可以提高准确率，但是达到某种程度以后，增加词向量维度和数据规模却带来了退化问题。

本文也对比这两种模型与其他模型的性能，下图为同一数据下训练各个模型的测试结果，词向量维度为640。



基于词向量的神经网络已应用于许多其他NLP任务，例如情感分析和释义检测。