word2vec：

**1.什么是word2vec**

如果用一句比较简单的话来总结，word2vec是用一个一层的神经网络(即CBOW)把one-hot形式的稀疏词向量映射称为一个n维(n一般为几百)的稠密向量的过程。为了加快模型训练速度，其中的tricks包括Hierarchical softmax，negative sampling, Huffman Tree等。

在NLP中，最细粒度的对象是词语。如果我们要进行词性标注，用一般的思路，我们可以有一系列的样本数据(x,y)。其中x表示词语，y表示词性。而我们要做的，就是找到一个x -> y的映射关系，传统的方法包括Bayes,SVM等算法。但是我们的数学模型，一般都是数值型的输入。但是NLP中的词语，是人类的抽象总结，是符号形式的（比如中文、英文、拉丁文等等），所以需要把他们转换成数值形式，或者说——嵌入到一个数学空间里，这种嵌入方式，就叫词嵌入（word embedding)，而 Word2vec，就是词嵌入（ word embedding) 的一种。

在 NLP 中，把 x 看做一个句子里的一个词语，y 是这个词语的上下文词语，那么这里的 f，便是 NLP 中经常出现的『语言模型』（language model），这个模型的目的，就是判断 (x,y) 这个样本，是否符合自然语言的法则，更通俗点说就是：词语x和词语y放在一起，是不是人话。

Word2vec 正是来源于这个思想，但它的最终目的，不是要把 f 训练得多么完美，而是只关心模型训练完后的副产物——模型参数（这里特指神经网络的权重），并将这些参数，作为输入 x 的某种向量化的表示，这个向量便叫做——词向量。

**2.词向量是什么**

自然语言理解的问题要转化为机器学习的问题，第一步肯定是要找一种方法把这些符号数学化。

　　NLP 中最直观，也是到目前为止最常用的词表示方法是 One-hot Representation，这种方法把每个词表示为一个很长的向量。这个向量的维度是词表大小，其中绝大多数元素为 0，只有一个维度的值为 1，这个维度就代表了当前的词。

　　举个栗子，

　　“话筒”表示为 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 ...]

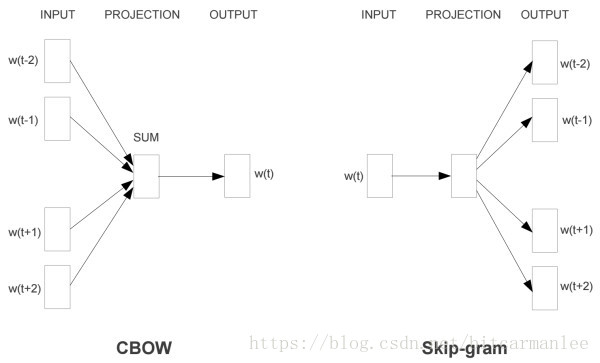
　　“麦克”表示为 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 ...]

　　每个词都是茫茫 0 海中的一个 1。

3.两个重要的模型CBOW模型和Skip-gram模型

a、如果是用一个词语作为输入，来预测它周围的上下文，那这个模型叫做Skip-gram 模型

b、而如果是拿一个词语的上下文作为输入，来预测这个词语本身，则是CBOW 模型



CBOW就是根据某个词前面的C个词或者前后C个连续的词，来计算某个词出现的概率。Skip-Gram Model相反，是根据某个词，然后分别计算它前后出现某几个词的各个概率。

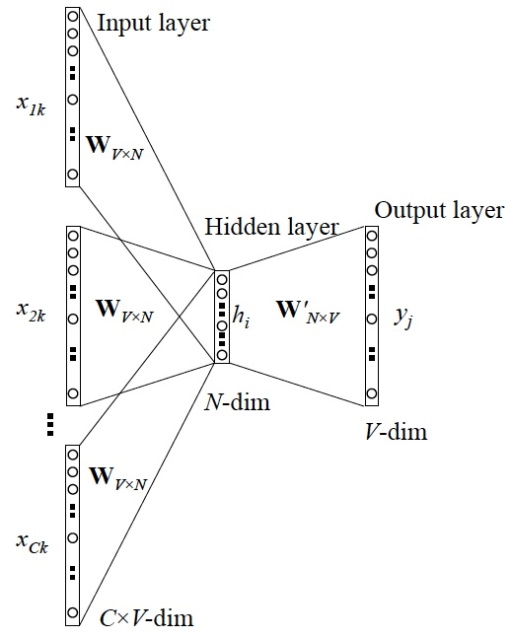
**4.tricks训练技巧**

hierarchical softmax 和 negative sampling ，它们并不是 Word2vec 的精髓，只是它的训练技巧，但也不是它独有的训练技巧。 Hierarchical softmax 只是 softmax 的一种近似形式

为什么要用训练技巧呢？ 如我们刚提到的，Word2vec 本质上是一个语言模型，它的输出节点数是 V 个，对应了 V 个词语，本质上是一个多分类问题，但实际当中，词语的个数非常非常多，会给计算造成很大困难，所以需要用技巧来加速训练。

（1）.hierarchical softmax  
最后预测输出向量时候，大小是1\*V的向量，本质上是个多分类的问题。通过hierarchical softmax的技巧，把V分类的问题变成了log(V)次二分类。

（2）.negative sampling  
本质上是对训练集进行了采样，从而减小了训练集的大小。  
每个词𝑤的概率由下式决定

5、word2vec  
  
把这张图看懂，基本word2vec就懂了一大半。下面来详细说说这张图。

词向量最简单的方式是1-of-N的one-hot方式。onehot对于同学们来说都很熟悉了，也就是从很大的词库corpus里选V个频率最高的词(忽略其他的) ，V一般比较大，比如V＝10W，固定这些词的顺序，然后每个词就可以用一个V维的稀疏向量表示了，这个向量只有一个位置的元素是1，其他位置的元素都是0。One hot方式其实就是简单的直接映射，所以缺点也很明显，维数很大，也没啥计算上的意义。