1. **skip-gram模型：**

用一个中心词去预测它周围的单词，但是经过这么一段数据预处理，生成的数据和标签其实是一一对应的，一个中心词对应一个周围单词，而不是一个中心词对应它周围的多个单词。

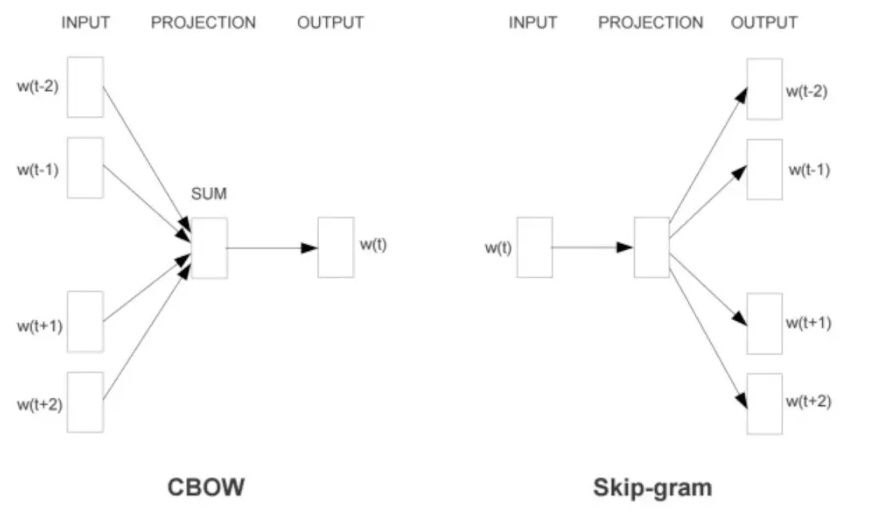
1. **word2vec模型：**

有两种训练模型：

1. CBOW（Continuous Bag-Of-Words Model词袋）
2. Skip-gram（Continuous Skip-gram Model）

**CBOW：**是采用周围的词来预测中心词的；CBOW算法对于很多分布式信息进行了平滑处理（例如将一整段上下文信息视为一个单一观察量）。很多情况下，对于小型的数据集，这一处理是有帮助的。

**Skip-Gram模型**：采取CBOW的逆过程的动机在于：相形之下，Skip-Gram模型将每个“上下文-目标词汇”的组合视为一个新观察量，这种做法在大型数据集中会更为有效。



1. **Softmax：**

（1）softmax函数本质：就是将一个K 维的任意实数向量压缩（映射）成另一个K维的实数向量，其中向量中的每个元素取值都介于（0，1）之间。

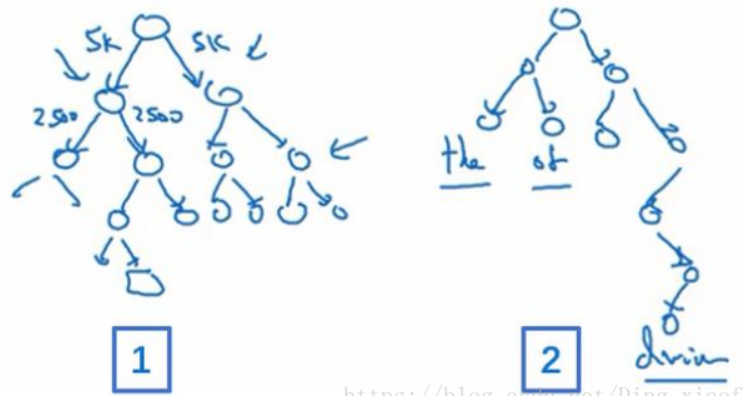
（2）softmax函数：

（3）softmax分类器：

词向量的基本任务还是预测任务；单词每计算一个样本数据，需要计算一次词典里面全部词是要预测的那个词的概率，这是很耗费资源。

**使用softmax分层，减少计算量：**

**eg：**确定到底是属于 10,000 （词典数）类中的哪一类。softmax分类器标明，目标词是在词汇表的前 5000 个中还是在词汇表的后 5000 个词中，假如这个二分类器告诉你这个词在前 5000 个词中，然后第二个分类器会告诉你这个词在词汇表的前 2500 个词中，或者在词汇表的第二组 2500 个词中，诸如此类，直到最终你找到一个词准确所在的分类器，那么就是这棵树的一个叶子节点。

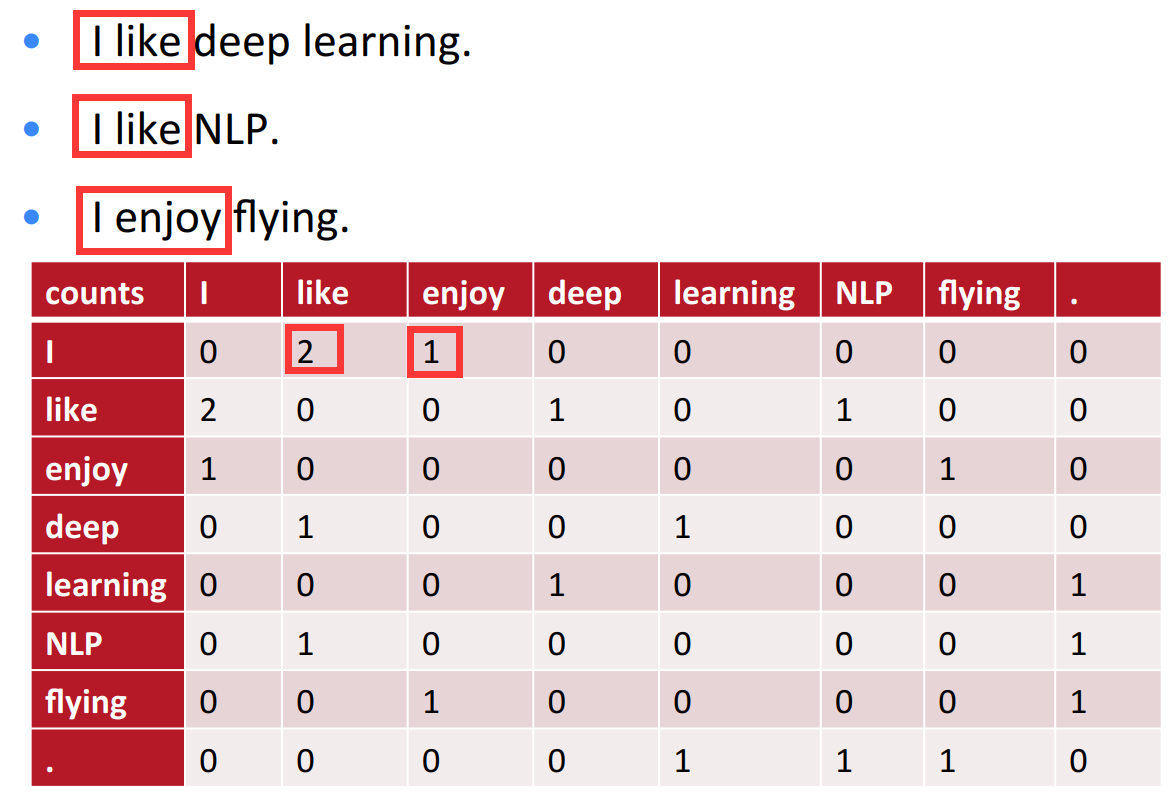


（4）sigmoid、softmax的区别：

sigmoid将一个real value映射到（0,1）的区间（当然也可以是（-1,1）），这样可以用来做二分类。

softmax把一个k维的real value向量（a1,a2,a3,a4….）映射成一个（b1,b2,b3,b4….）其中bi是一个0-1的常数，然后可以根据bi的大小来进行多分类的任务，如取权重最大的一维。

1. **共现矩阵（co-occurrence）**



I：旁边有2个like，所以第1行第2列=2

I：旁边有1个enjoy，所以第1行第3列=1

**应用：**可以将每一行作为词向量，可以用SVD对矩阵降维。