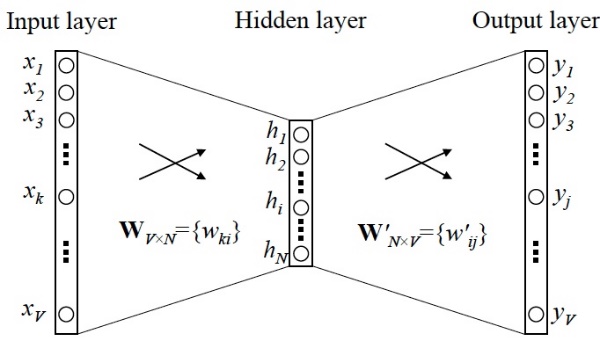
# Word2Vector

论文：Efﬁcient Estimation of Word Representations in Vector Space

通过训练处理简单任务的语言模型，来学习一个合适的词表，将词表示成计算机能够处理的词向量形式，同时这种形式的词向量要能够表示出词与词之间的联系。

常见的有两种形式的语言模型Skip-gram和CBOW，它们都是由输入层、隐藏层、输出层构成。



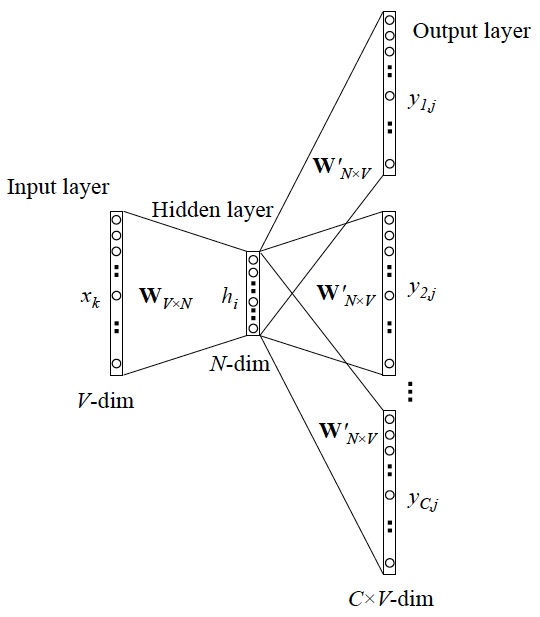
模型最终目的都是获取到输入层和隐藏层之间的权重矩阵，这样对于任意一个词的输入X，都可以得到其词向量。

在模型中，一般有两个词表，一个是中心词表，一个是上下文词表，一个词在作为中心词和上下文词时它的向量表示是不同的。

## Skip-gram

### 模型介绍

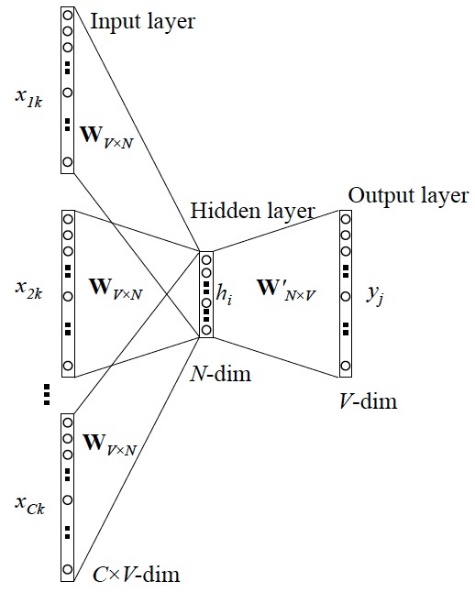
模型目标：输入一个词，预测该词的上下文，T是预测窗口的大小。



1. 从输入词的上下文单词随机抽出num\_skip(超参数)个词组成num\_skip组数据
2. 使用这num\_skip组数据训练模型
3. 隐藏层参数矩阵，输出层参数矩阵
4. 对于数据，，，

## CBOW

模型目标：输入词的上下文，预测该词



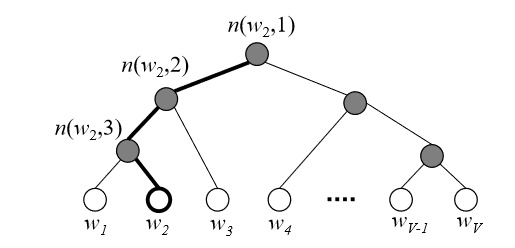
1. 隐藏层参数矩阵，输出层参数矩阵
2. 隐藏层向量，使用同一个参数矩阵
3. 输出向量

## 分层softmax(Hierarchical Softmax)

面临问题：当词表太大时，直接使用softmax计算量太大。

解决方法：

1. 将词表中所有的词构成一个树，树中叶子节点即为对应词表中的词，非叶子节点可以理解成一个可学习的神经元，拥有一组可学习的参数向量。



1. 表示从根节点到叶子节点的路径
2. 则
   1. 其中代表在节点处向左走
   2. 代表在节点处向右走
   3. 是节点的参数向量
   4. 是隐藏层的输出向量
   5. 是激活函数，将值映射到0~1
3. 这样一来
4. 损失函数：
5. 具体实现是构建的是哈夫曼树，方便计算路径中每一步是向左走还是向右走；同时由于哈夫曼树是根据权值(词频)构建的，越常出现的词路径越短，需要的计算量越低。

## 负采样

分层softmax解决了计算loss时的计算量问题，负采样则解决了更新词向量矩阵(隐藏层参数矩阵)的计算量问题。

思路：

1. 对标签词进行采样，作为正样本；对词表中剩余词进行随机采样，作为负样本
2. 对于词向量矩阵而言，每一列对应一个词
3. 原始梯度下降更新参数矩阵时是所有列都更新，现在只更新采样样本对应的列。

# Word Embedding

## 首次提出

《A Neural Probabilistic Language Model》

## 思想

## 做法

## 缺点

无法处理同义词问题，一个词最终只能被映射到一个词向量上，但一个词可以有多种不同的含义，使用在不同的场景下。

# ELMO

## 论文

《Deep contextualized word representation》

## 思想

## 做法