Word2Vec

2021-10-16

1. 背景知识

1.1. Log-linear Model

定义（Log Linear Models）：将语言模型的建立看成是一个多分类问题，相当于线性分类器加上softmax操作。

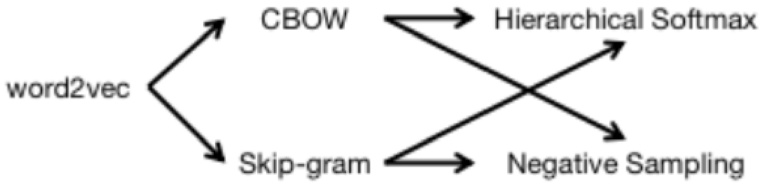
$$Y = \mathrm{softmax}(wx + b) \tag{52}$$

1.2. 符号定义

1. 单词：w
2. 词典：$\mathcal{D} = \{w_1, w_2, \cdots, w_N\}$，其中，$N$为单词个数。由单词组成的集合。
3. 语料库：$\mathcal{C}$，由单词组成的文本序列。
4. 上下文：$\mathrm{Context}(w_t)$是单词w_t在语料库中前c个单词和后c个单词组成的文本序列，w_t称为中心词，c为窗口长度
5. 词向量：$\mathbf{v}(w)$表示单词w对应的词向量

2. 两种模型

2.1. Word2Vec简介



语言模型基本思想：句子中下一个词的出现和前面的词是有关系的，所以可以使用**前面的词**预测**下一个词**。

Word2Vec基本思想：句子中**相近的词**之间是有联系的，比如今天后面经常出现上午、下午。所以Word2Vec的基本思想就是用词来预测词，**CBOW**使用周围词预测中心词，**Skip-gram**使用中心词预测周围词。

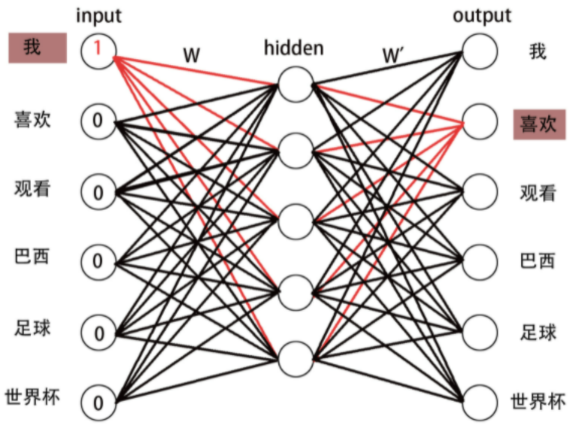
2.2. 单个词到单个词

为了便于理解CBOW和Skip-gram模型，先介绍一个词到一个词的简单模型。

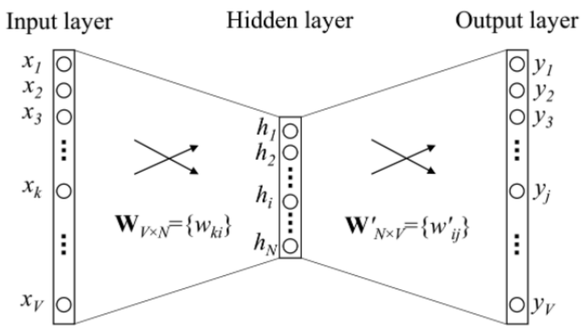
假设输入为：我喜欢观看巴西足球世界杯。经过分词，得到词表：['我','喜欢','观看','巴西','足球','世界杯']。为了构建一个词到一个词的模型，将单词两两分组，得到**数据集**：[['我','喜欢'],['喜欢','观看'],['观看','巴西'],['巴西','足球'],['足球','世界杯']]。使用one-hot对单词进行表示如下图所示：

我	1	0	0	0	0	0
喜欢	0	1	0	0	0	0
观看	0	0	1	0	0	0
巴西	0	0	0	1	0	0
足球	0	0	0	0	1	0
世界杯	0	0	0	0	0	1

接下来，输入数据：['我', '喜欢']，在输出中，期望'喜欢'的概率最大。



扩展到一般的网络结构：



定义如下符号：

1. V：词表大小（或语料库中不同单词的数目）
2. N：词向量维度
3. $\mathbf{X}_{N\times 1}$：输入单词，使用**one-hot编码**表示
4. w：原始单词
5. $\text{Context}(w_i)_c$：单词w_i的第c个周围词，其中，$1 \le c \le C$
6. $\mathbf{W}_{V\times N}$：输入层与隐藏层之间的权重
7. $\mathbf{W'}_{N\times V}$：隐藏层与输出层之间的权重
8. $\mathbf{u}_{V\times 1}$：每个单词的预测值，通过softmax计算可得到概率
9. $\mathbf{y}_{V\times 1}$：每个单词的概率

前向计算：

1. $\mathbf{h}_{N\times 1}=\mathbf{W}^{\mathrm{T}}\mathbf{X}$
   $h_i=\sum_{k=1}^{V}w_{ki}x_k$
2. $\mathbf{u}_{V\times 1}={\mathbf{W}'}^{\mathrm{T}}\mathbf{h}$
   $u_j=\sum_{i=1}^{N}w'_{ij}h_i$，注意，$h_i$出现在每一个$u_j$当中，如果对$h_i$求导，需要遍历每一个$u_j$
3. $\mathbf{y}=\operatorname{softmax}(\mathbf{u})$

损失函数：

1. 给定中心词w_i，预测词w_j的概率

$$p(w_j\mid w_i)=y_j=\frac{e^{u_j}}{\sum_{k=1}^{V}e^{u_k}} \tag{53}$$

2. 损失函数
   假设a^*是正确的单词对应的索引，我们期望$p(w_{a^*}\mid w_i)$最大，等价地，可以导出如下损失函数：

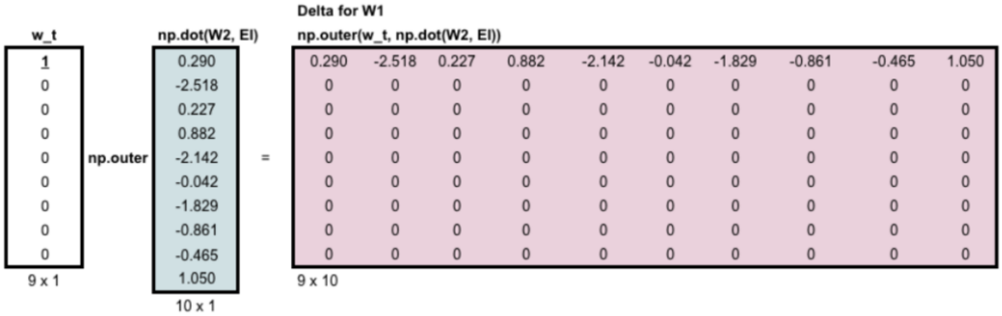
$$E=-\log\left\{\frac{e^{u_{a^*}}}{\sum_{k=1}^{V}e^{u_k}}\right\}=-u_{a^*}+\log\left(\sum_{k=1}^{V}e^{u_k}\right) \tag{54}$$

反向传播：

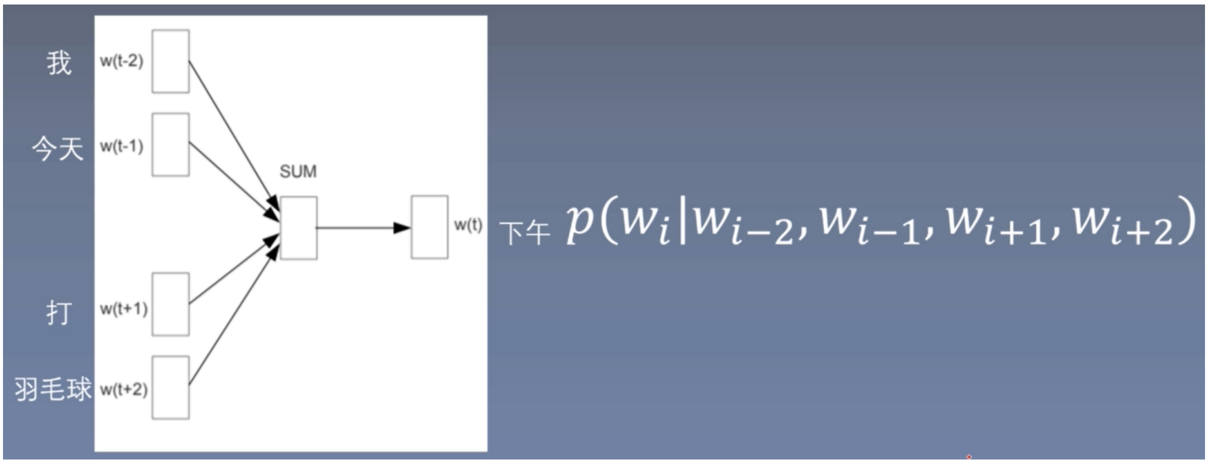
$$\begin{aligned}
&(1)\text{对}\mathbf{W}'\text{求导}\\
&\qquad\frac{\partial E}{\partial u_j}=-t(j,a^*)+y_j:=e_j\\
&\qquad\frac{\partial E}{\partial w'_{ij}}=\frac{\partial E}{\partial u_j}\cdot\frac{\partial u_j}{\partial w'_{ij}}=e_jh_i\\
&\qquad\frac{\partial E}{\partial \mathbf{W}'}=\sum_{i=1}^{N}\sum_{j=1}^{V}h_i\cdot e_j=\mathbf{h}\otimes\mathbf{e}^T\\
&(2)\text{对}\mathbf{W}\text{求导}\\
&\qquad\frac{\partial E}{\partial h_i}=\sum_{j=1}^{V}\frac{\partial E}{\partial u_j}\cdot\frac{\partial u_j}{\partial h_i}=\sum_{j=1}^{V}e_j\cdot w'_{ij}:=EH_i\\
&\qquad\frac{\partial E}{\partial w_{ki}}=\frac{\partial E}{\partial h_i}\cdot\frac{\partial h_i}{\partial w_{ki}}=(\sum_{j=1}^{V}e_j\cdot w'_{ij})\cdot x_k=EH_i\cdot x_k\\
&\qquad\frac{\partial E}{\partial \mathbf{W}}=\sum_{k=1}^{V}\sum_{i=1}^{N}EH_i\cdot x_k=\mathbf{X}\otimes EH^T\text{，得到}V\times N\text{的矩阵，由于}\mathbf{X}\text{只有}1\text{行非}0\text{，所以矩阵只有}1\text{行非}0\text{，为}EH^T
\end{aligned} \tag{55}$$

参数更新：

1. $\mathbf{W}'$更新：需要更新整个矩阵
2. $\mathbf{W}$更新：只需要更新a^*对应的行。梯度更新如下图所示：（注：w_t应为$\mathbf{X}$，$\mathbf{W}_2$应为$\mathbf{W}'$）



2.3. CBOW



首先，需要定义window，即选取多少个周围词，上图中window=2。通过周围词预测中心词，该问题为多分类问题。

词向量：$\mathbf{v}$

输入：周围词，$w_{i-1},w_{i-2},w_{i+1},w_{i+2}$，将其词向量相加得到周围词词向量$\mathbf{v}_o$

标签：中心词，w_i，M为语料库$\mathcal{C}$中的中心词个数

使用向量内积表示词向量的相似度，那么，中心词的预测概率为：

$$p\left(w_i \mid w_{i-2},w_{i-1},w_{i+1},w_{i+2}\right)=\frac{\exp\left(\mathbf{v}_o^T\mathbf{v}_{w_i}\right)}{\sum_{j=1}^{N}\exp\left(\mathbf{v}_o^T\mathbf{v}_j\right)} \tag{56}$$

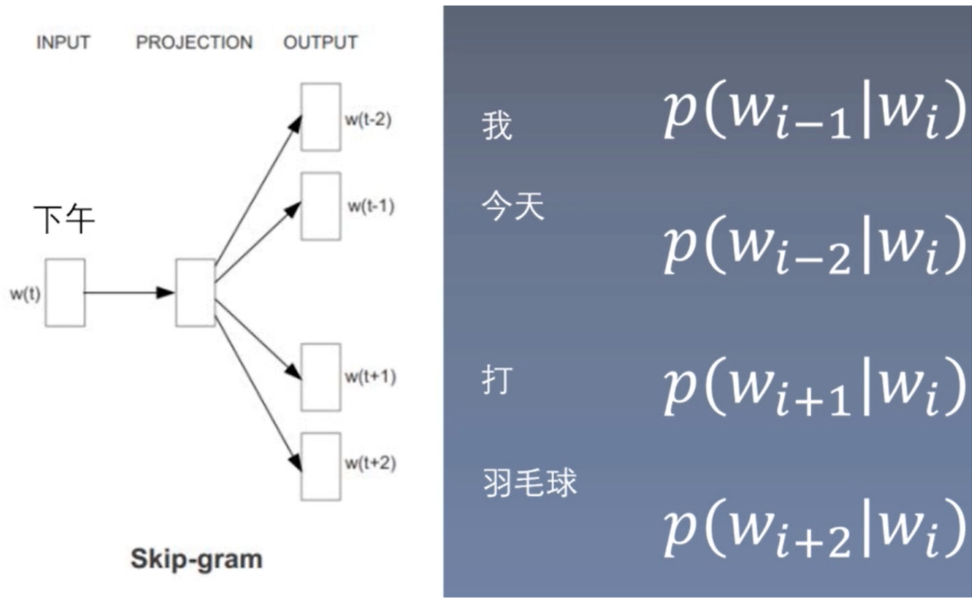
其中，$\mathbf{v}_{w_i}$、$\mathbf{v}_j$为中心词的词向量

损失函数（使中心词的概率最大）：

$$\begin{aligned} J(\theta) &= -\frac{1}{M}\sum_{i=1}^{M}\log p\left(w_i \mid w_o\right) \\ &= -\frac{1}{M}\sum_{i=1}^{M}\frac{\exp\left(\mathbf{v}_o^T\mathbf{v}_{w_i}\right)}{\sum_{j=1}^{M}\exp\left(\mathbf{v}_o^T\mathbf{v}_j\right)} \end{aligned} \tag{57}$$

其中，M为中心词个数，w_i为中心词，w_o为w_i对应的周围词，$\mathbf{v}_o$为周围词词向量的和，$\mathbf{v}_j$为中心词词向量

2.4. Skip-gram



首先，需要定义window，即选取多少个周围词，上图中window=2。通过中心词预测周围词，该问题为多分类问题。

词向量：$\mathbf{v}$

输入：中心词，w_i

标签：周围词，$w_{i-1},w_{i-2},w_{i+1},w_{i+2}$，$M$为周围词个数

使用向量内积表示词向量的相似度，那么，周围词的预测概率为：

$$p\left(w_{i-1} \mid w_{i}\right)=\frac{\exp \left(\mathbf{v}_{w_{i-1}}^{T} \mathbf{v}_{w_{i}}\right)}{\sum_{j=1}^{M} \exp \left(\mathbf{v}_{j}^{T} \mathbf{v}_{w_{i}}\right)} \text{其中，} v_{j} \text{为周围词的词向量} \tag{58}$$

损失函数（使周围词的概率最大）：

$$J(\theta)=-\frac{1}{M} \sum_{m=1}^{M} \sum_{-c \leq j \leq c, j \neq 0} \log p\left(w_{m+j} \mid w_{m}\right) \text{其中，} c \text{为窗口大小} \tag{59}$$

3. 计算优化

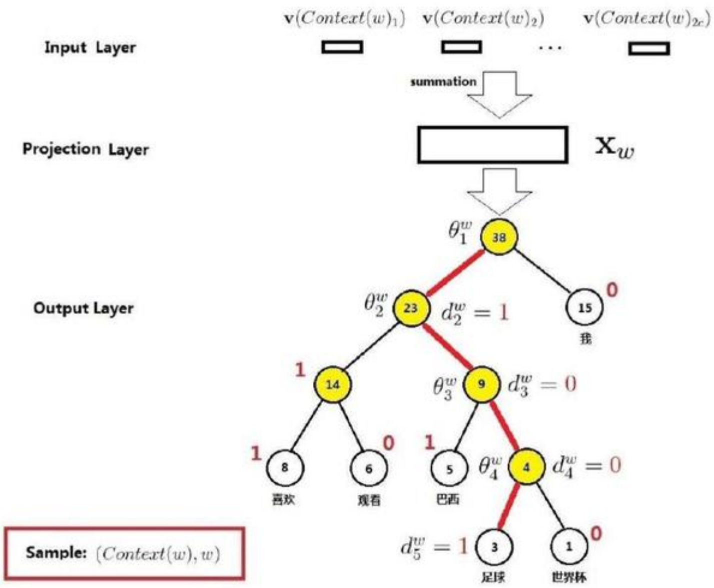
为了解决上述模型中softmax计算量太大的问题，使用以下两种方法进行优化。

3.1. Hierarchical softmax

核心思想：将多分类问题转化为多个二分类问题。

方法：将词表中的单词，按照频率构建一颗哈夫曼树，这样，对每一个单词，便有了唯一正确的一个搜索路径。如果将**搜索路径**视为输入X，将**单词**视为真实标签y，那么，从根结点到目标单词（叶子结点），便是若干个二分类过程。至此，完成了问题的转化。

3.1.1. CBOW



针对上述哈夫曼树，各层解释如下：

1. 输入层

   $\mathbf{v}\left(Context(w)_{1}\right)$，$\mathbf{v}\left(Context(w)_{2}\right)$，$\cdots$，$\mathbf{v}\left(Context(w)_{2c}\right) \in \mathbb{R}^{m}$，其中，$\mathbf{v}(\cdot)$为单词的向量化表示

2. 投影层

$$\mathbf{x}_{w}=\sum_{i=1}^{2c} \mathbf{v}\left(\operatorname{Context}(w)_{i}\right) \in \mathbb{R}^{m} \tag{60}$$

3. 输出层

   给定w的周围词，单词w出现的概率，即$p(w \mid \operatorname{Context}(w))$

为方便计算损失函数，定义如下变量：

1. **路径**：$p^{w}=\left(p_{1}^{w}, p_{2}^{w}, \cdots, p_{l_{w}}^{w}\right)$

   从根结点出发，到达w对应的叶子结点的路径。其中，l^{w}为路径长度，即路径中结点数目；p_{i}^{w}为路径中的结点，p_{1}^{w}为根结点，$p_{l_{w}}^{w}$为w对应的叶子结点。

2. **编码**：$d^{w}=\left(d_{1}^{w}, d_{2}^{w}, \cdots, d_{l_{w}}^{w}\right)$

   w的Huffman编码。其中，$d_{i}^{w} \in\{0,1\}$为路径p^{w}中第i个结点对应的编码（根结点不对应编码）。

3. **权值**：$\theta^{w}=\left(\theta_{1}^{w}, \theta_{2}^{w}, \cdots, \theta_{l_{w}-1}^{w}\right)$

路径p^w中非叶子结点对应的参数向量。其中，$\theta_i \in \mathbb{R}^m$为路径$p^w$中第$i$个非叶子结点对应的参数向量。

那么，在已知周围词$Context(w)$的条件下，中心词w的概率为：

$$p(w \mid \operatorname{Context}(w))=\prod_{j=2}^{l^w} p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right) \text{注：顺着路径走} \tag{61}$$

其中，

$$p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right)=\begin{cases}\sigma\left(\mathbf{x}_w^T \theta_{j-1}^w\right), d_j^w=0 \\ 1-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right), d_j^w=1\end{cases} \text{注：该概率决定往左还是往右} \tag{62}$$

该概率也可以写成如下形式：

$$p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right)=\left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]^{1-d_j^w} \cdot\left[1-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]^{d_j^w} \tag{63}$$

那么，似然函数为：

$$\begin{aligned} \ell & =\prod_{w \in C} p(w \mid \text { Context }(w)) \\ & =\prod_{w \in C} \prod_{j=2}^{l^w} p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right)\end{aligned} \tag{64}$$

对数似然函数为：

$$\begin{aligned} \mathcal{L} & =\log \prod_{w \in C} \prod_{j=2}^{l^w} p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right) \\ & =\sum_{w \in \mathcal{C}} \log \prod_{j=2}^{l^w} p\left(d_j^w \mid \mathbf{x}_w, \theta_{j-1}^w\right) \\ & =\sum_{w \in C} \sum_{j=2}^{l^w}\left\{\left(1-d_j^w\right) \cdot \log \left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]+d_j^w \cdot \log \left[1-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]\right\}\end{aligned} \tag{65}$$

对数似然函数$\mathcal{L}$关于θ_{j-1}^w求偏导为：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_{j-1}^w} & =\frac{\partial}{\partial \theta_{j-1}^w}\left\{\sum_{w \in C} \sum_{j=2}^{l^w}\left\{\left(1-d_j^w\right) \cdot \log \left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]+d_j^w \cdot \log \left[1-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right]\right\}\right\} \\ & =\left(1-d_j^w\right)\left[1-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right] \mathbf{x}_w-d_j^w \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right) \mathbf{x}_w \\ & =\left[1-d_j^w-\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta_{j-1}^w\right)\right] \mathbf{x}_w\end{aligned} \tag{66}$$

所以，θ_{j-1}^w的更新公式为：

$$\theta_{j-1}^w=\theta_{j-1}^w+\eta\left[1-d_j^w-\sigma\left(\mathbf{x}_w^{\top} \theta_{j-1}^w\right)\right] \mathbf{x}_w \text{其中，}\eta\text{为学习率} \tag{67}$$

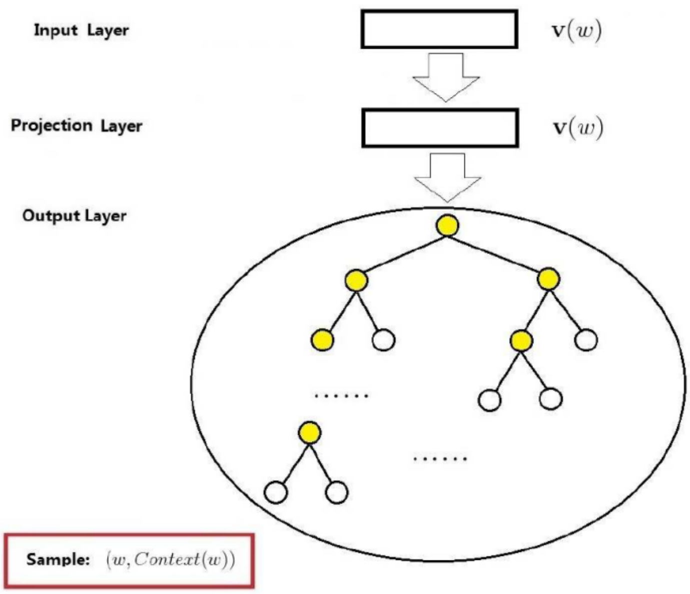
对数似然函数$\mathcal{L}$关于$\mathbf{x}_w$求偏导为：

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_w}=\sum_{j=2}^{l^w}\left[1-d_j^w-\sigma\left(\mathbf{x}_w^{\top} \theta_{j-1}^w\right)\right] \theta_{j-1}^w \tag{68}$$

$\mathbf{v}(\tilde{w})$的更新公式为：

$$\mathbf{v}(\tilde{w})=\mathbf{v}(\tilde{w})+\eta \frac{\partial \mathcal{L}}{\partial \mathbf{x}_w} \text{其中，}\tilde{w} \in Context(w) \tag{69}$$

3.1.2. Skip-gram



针对上述哈夫曼树，各层解释如下：

1. 输入层

   当前样本的中心词w对应的词向量$\mathbf{v}(w) \in \mathbb{R}^m$

2. 映射层

   恒等映射，多余，为了和CBOW模型的网络结构进行对比。

3. 输出层

那么，给定中心词w的条件下，其周围词$Context(w)$的条件概率为：

$$p(\text{Context}(w) \mid w) = \prod_{u \in \text{Context}(w)} p(u \mid w) \tag{70}$$

其中，

$$p(u \mid w) = \prod_{j=2}^{l^u} p\left(d_j^u \mid \mathbf{v}(w), \theta_{j-1}^u\right) \tag{71}$$

并且：

$$p\left(d_j^u \mid \mathbf{v}(w), \theta_{j-1}^u\right) = \left[\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]^{1-d_j^u} \cdot \left[1-\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]^{d_j^u} \tag{72}$$

所以，似然函数为：

$$\begin{aligned} \ell &= \prod_{w \in \mathcal{C}} p(\text{Context}(w) \mid w) \\ &= \prod_{w \in \mathcal{C}} \prod_{u \in Context(w)} \prod_{j=2}^{l^u} p\left(d_j^u \mid \mathbf{v}(w), \theta_{j-1}^u\right) \end{aligned} \tag{73}$$

对数似然函数为：

$$\begin{aligned} \mathcal{L} &= \sum_{w \in \mathcal{C}} \log \prod_{u \in \text{Context}(w)} \prod_{j=2}^{l^u} \left\{\left[\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]^{1-d_j^u} \cdot \left[1-\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]^{d_j^u}\right\} \\ &= \sum_{w \in \mathcal{C}} \sum_{u \in \text{Context}(w)} \sum_{j=2}^{l^u} \left\{\left(1-d_j^u\right) \cdot \log\left[\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] + d_j^u \cdot \log\left[1-\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]\right\} \end{aligned} \tag{74}$$

对数似然函数$\mathcal{L}$关于θ_{j-1}^u的偏导为：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_{j-1}^u} &= \frac{\partial}{\theta_{j-1}^u}\left\{\sum_{w \in \mathcal{C}} \sum_{u \in Context(w)} \sum_{j=2}^{l^u} \left\{\left(1-d_j^u\right) \cdot \log\left[\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] + d_j^u \cdot \log\left[1-\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right]\right\}\right\} \\ &= \sum_{w \in \mathcal{C}} \left\{\left(1-d_j^u\right)\left[1-\sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] \mathbf{v}(w) - d_j^u \sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right) \mathbf{v}(w)\right\} \end{aligned} \quad (7$$

$$= \sum_{w \in \mathcal{C}} \left[1 - d_j^u - \sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] \mathbf{v}(w)$$

θ_{j-1}^u的更新公式为:

$$\theta_{j-1}^u = \theta_{j-1}^u + \eta \sum_{w \in \mathcal{C}} \left[1 - d_j^u - \sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] \mathbf{v}(w) \quad \text{其中，}\eta\text{为学习率} \tag{76}$$

对数似然函数$\mathcal{L}$关于$\mathbf{v}(w)$的偏导为:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}(w)} = \sum_{u \in \mathrm{Context}(w)} \sum_{j=2}^{l^u} \left[1 - d_j^u - \sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] \theta_{j-1}^u \tag{77}$$

$\mathbf{v}(w)$的更新为:

$$\mathbf{v}(w) = \mathbf{v}(w) + \eta \sum_{u \in \mathrm{Context}(w)} \sum_{j=2}^{l^u} \left[1 - d_j^u - \sigma\left(\mathbf{v}(w)^{\mathrm{T}} \theta_{j-1}^u\right)\right] \theta_{j-1}^u \tag{78}$$

3.2. Negative Sampling

3.2.1. CBOW

设$Context(w)$的负样本子集（非w的周围词）为:

$$NEG(w) \neq \varnothing$$

对于$\forall \tilde{w} \in \mathcal{D}$，定义$L^w(\tilde{w})$表示词$\tilde{w}$的标签，即$\tilde{w}$是否是中心词$w$的周围词，正样本标签为1，负样本标签为0:

$$L^w(\tilde{w}) = \begin{cases} 1, \tilde{w} = w \\ 0, \tilde{w} \neq w \end{cases} \tag{79}$$

在给定周围词$\mathrm{Context}(w)$的条件下，中心词与负样本数据集，即$\{w\} \cup NEG(w)$，其似然函数为:

$$g(w) = \prod_{u \in \{w\} \cup NEG(w)} p(u \mid \mathrm{Context}(w)) = \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^w\right) \prod_{u \in NEG(w)} \left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^w\right)\right] \tag{80}$$

其中,

$$p(u \mid \mathrm{Context}(w)) = \begin{cases} \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right), L^w(u) = 1 \\ 1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right), L^w(u) = 0 \end{cases} \text{其中，}\mathbf{x}_w\text{为 Context}(w)\text{词向量之和，} \tag{81}$$

上式也可以写为:

$$p(u \mid \mathrm{Context}(w)) = \left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right]^{L^w(u)} \cdot \left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right]^{1 - L^w(u)} \tag{82}$$

关于词表$\mathcal{C}$的对数似然函数为:

$$\begin{aligned} \mathcal{L} &= \log \prod_{w \in \mathcal{C}} g(w) \\ &= \sum_{w \in C} \log g(w) \\ &= \sum_{w \in \mathcal{C}} \log \prod_{u \in \{w\} \cup NEG(w)} \left\{ \left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right]^{L^w(u)} \cdot \left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right]^{1 - L^w(u)} \right\} \\ &= \sum_{w \in \mathcal{C}} \sum_{u \in \{w\} \cup NEG(w)} \left\{ L^w(u) \cdot \log\left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \right\} \end{aligned} \tag{83}$$

对数似然函数$\mathcal{L}$关于θ^u的偏导为:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta^u} &= \frac{\partial}{\partial \theta^u} \left\{ \sum_{w \in \mathcal{C}} \sum_{u \in \{w\} \cup NEG(w)} \left\{ L^w(u) \cdot \log\left[\sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \right\} \right\} \\ &= L^w(u)\left[1 - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \mathbf{x}_w - \left[1 - L^w(u)\right] \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right) \mathbf{x}_w \\ &= \left[L^w(u) - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \mathbf{x}_w \end{aligned} \tag{84}$$

参数θ^u的更新公式为:

$$\theta^u = \theta^u + \eta\left[L^w(u) - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \mathbf{x}_w \tag{85}$$

对数似然函数$\mathcal{L}$关于$\mathbf{x}_w$的偏导为:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_w} = \sum_{u \in \{w\} \cup NEG(w)} \left[L^w(u) - \sigma\left(\mathbf{x}_w^{\mathrm{T}} \theta^u\right)\right] \theta^u \tag{86}$$

参数$\mathbf{v}(\tilde{w})$的更新公式为：

$$\mathbf{v}(\tilde{w}) = \mathbf{v}(\tilde{w}) + \eta \frac{\partial \mathcal{L}}{\partial \mathbf{x}_w} \tag{87}$$

其中，$\tilde{w} \in \text{Context}(w)$

3.2.2. Skip-gram

给定中心词w，对应多个周围词$\tilde{w}$，$NEG^{\tilde{w}}(w)$表示相对于$(w, \tilde{w})$的负样本。

给定周围词$\tilde{w}$，$\{w\} \cup NEG^{\tilde{w}}(w)$的似然函数为：

$$g(w) = \prod_{\tilde{w} \in \text{ Context } (w)} \prod_{u \in \{w\} \bigcup NEG^{\tilde{w}}(w)} p(u \mid \tilde{w}) \qquad (88)$$

其中，$NEG^{\tilde{w}}(w)$为处理周围词$\tilde{w}$时生成的负样本子集

其中，

$$p(u \mid \tilde{w}) = \begin{cases} \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right), L^w(u) = 1 \\ 1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right), L^w(u) = 0 \end{cases} \qquad (89)$$

上式又可以写为：

$$p(u \mid \tilde{w}) = \left[\sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]^{L^w(u)} \cdot \left[1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]^{1 - L^w(u)}$$

关于词表$\mathcal{C}$的对数似然函数为：

$$\begin{aligned} \mathcal{L} &= \log \prod_{w \in \mathcal{C}} g(w) \\ &= \sum_{w \in \mathcal{C}} \log g(w) \\ &= \sum_{w \in \mathcal{C}} \log \prod_{\tilde{w} \in \text{ Context(w)}} \prod_{u \in \{w\} \bigcup NEG^{\tilde{w}}(w)} \left\{\left[\sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]^{L^w(u)} \cdot \left[1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]^{1 - L^w(u)}\right\} \\ &= \sum_{w \in \mathcal{C}} \sum_{\tilde{w} \in \text{Context}(w)} \sum_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} \left\{L^w(u) \cdot \log\left[\sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]\right\} \end{aligned} \tag{91}$$

对数似然函数$\mathcal{L}$关于θ^u的偏导为：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta^u} &= \frac{\partial}{\partial \theta^u}\left\{\sum_{w \in \mathcal{C}} \sum_{\tilde{w} \in \text{Context}(w)} \sum_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} \left\{L^w(u) \cdot \log\left[\sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right]\right\}\right\} \\ &= L^w(u)\left[1 - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] \mathbf{v}(\tilde{w}) - \left[1 - L^w(u)\right] \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right) \mathbf{v}(\tilde{w}) \\ &= \left[L^w(u) - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] \mathbf{v}(\tilde{w}) \end{aligned} \tag{92}$$

参数θ^u的更新公式为：

$$\theta^u = \theta^u + \eta\left[L^w(u) - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] \mathbf{v}(\tilde{w}) \tag{93}$$

对数似然函数$\mathcal{L}$关于$\mathbf{v}(\tilde{w})$的偏导为：

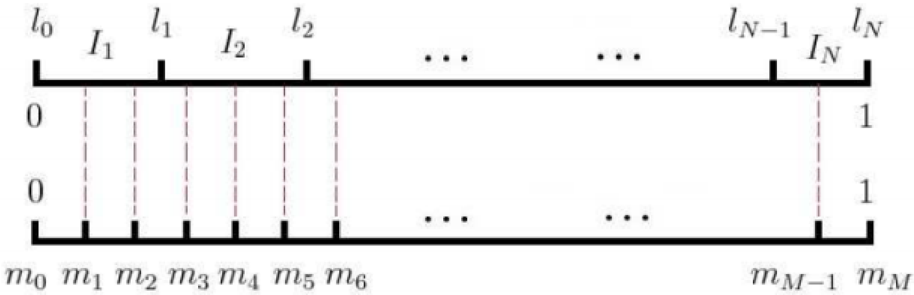
$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}(\tilde{w})} = \sum_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} \left[L^w(u) - \sigma\left(\mathbf{v}(\tilde{w})^{\mathrm{T}} \theta^u\right)\right] \theta^u \tag{94}$$

参数$\mathbf{v}(\tilde{w})$的更新公式为：

$$\mathbf{v}(\tilde{w}) = \mathbf{v}(\tilde{w}) + \eta \frac{\partial \mathcal{L}}{\partial \mathbf{v}(\tilde{w})} \tag{95}$$

4. 采样

4.1. 负采样



非等距剖分：

设词典$\mathcal{D}$中单词w_i对应的线段为$l(w_i)$，其长度为：

$$\operatorname{len}(w_i)=\frac{\operatorname{counter}(w_i)}{\sum_{u\in\mathcal{D}}\operatorname{counter}(u)} \tag{96}$$

其中，$\operatorname{counter}(\cdot)$为词在语料$\mathcal{C}$中的出现次数

可将线段$l(w_1),\cdots,l(w_N)$拼接为长度为1的单位线段。记：

$$\begin{aligned} l_0 &= 0 \\ l_k &= \sum_{j=1}^{k}\operatorname{len}(w_j), k=1,2,\cdots,N \end{aligned} \tag{97}$$

则以$l_j, j\in\{0,1,\cdots,N\}$为剖分点可得到区间$[0,1]$上的一个非等距剖分：

$$I_i=(l_{i-1},l_i], i=1,2,\cdots,N \tag{98}$$

等距剖分：

在区间$[0,1]$上以剖分点$m_j, j\in\{0,1,\cdots,M\}$做等距剖分，其中$M\gg N$。

等距与非等距映射：

将等距剖分的内部点$\{m_j\}_{j=1}^{M-1}$投影到非等距剖分，则可建立$\{m_j\}_{j=1}^{M-1}$与区间$\{I_j\}_{j=1}^{N}$的映射，进一步建立与词$\{w_j\}_{j=1}^{N}$之间的映射。

$$\operatorname{Table}(i)=w_k,\ \text{where } m_i\in I_k,\quad i=1,2,\cdots,M-1 \tag{99}$$

4.2. 重采样

重采样的目的：提高低频次出现的频率，降低高频词出现的概率。

训练集中的词w_i会以$P(w_i)$的概率被删除，概率$P(w_i)$计算公式为：

$$P(w_i)=1-\sqrt{\frac{t}{f(w_i)}} \tag{100}$$

其中，$f(w_i)$为词w_i在数据集中出现的频率，论文中t取10^{-5}

5. 复杂度分析

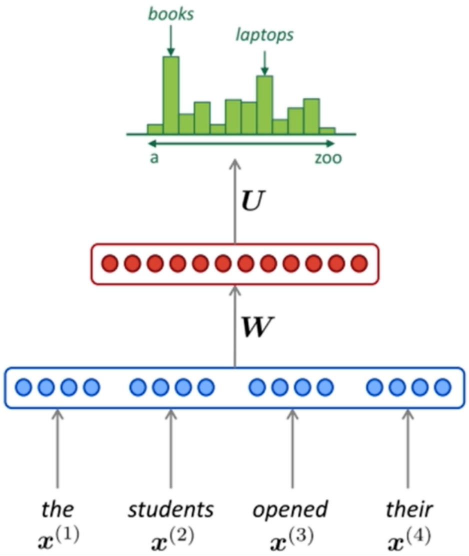
使用模型参数量表征模型复杂度。

定义符号：

1. O：训练复杂度
2. E：迭代次数
3. T：数据集大小
4. Q：参与本次计算的参数的数目

   那么：$O=E\times T\times Q$

5.1. NNLM(前馈神经网络)



模型：使用前N个词预测下一个词

$$y=\mathbf{U}\tanh(\mathbf{W}\mathbf{X}+d)$$

$$p\left(x_i \mid x_{i-1}, \ldots, x_{i-\mathrm{N}}\right)=\frac{e^{y_i}}{\sum_{j=1}^{V} e^{y_j}} \tag{101}$$

$$\text{其中，}\mathbf{X}=[\mathbf{x}^{(1)},\mathbf{x}^{(2)},\cdots,\mathbf{x}^{(N)}],\mathbf{y}=[y_1,y_2,\cdots,y_V]$$

符号定义：

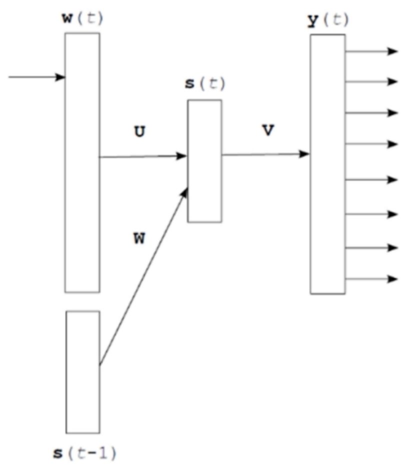
1. N：上文词（训练数据）个数
2. D：词向量维度
3. V：词表的大小
4. H：隐藏层大小
5. $\mathbf{x}_{N\times D}$
6. $\mathbf{W}_{N\times D\times H}$
7. $\mathbf{U}_{V\times H}$

参数个数：

1. 输入层：$N\times D$
2. 隐藏层：$N\times D\times H$
3. 输出层：$V\times H$。如果使用HS对输出进行优化，则需要进行$\log_2 V$次二分类，参数θ的长度为H，故参数量为$H\cdot\log_2 V$

总参数量：$Q=N\times D+N\times D\times H+V\times H$

5.2. RNNLM(循环神经网络语言模型)



$$\begin{gathered}\mathbf{s}(t)=\mathbf{U}\mathbf{w}(t)+\mathbf{W}\mathbf{s}(t-1)+d\\ \mathbf{y}(t)=\mathbf{V}\mathbf{s}(t)\end{gathered} \tag{102}$$

$$\text{其中，}\mathbf{w}(t)\text{表示时刻}t\text{的当前输入单词的词向量，}\mathbf{s}(t-1)\text{代表隐藏层的前一次输出}$$

符号定义：

1. $\mathbf{w}(t)$：维度为$D \times 1$
2. $\mathbf{U}_{H \times D}$
3. $\mathbf{W}_{H \times H}$
4. $\mathbf{s}(t)$：维度为$H \times 1$
5. $\mathbf{V}_{V \times H}$
6. $\mathbf{y}(t)$：维度为$V \times 1$

复杂度计算：

1. 输入层：$1 \times D$
2. 隐藏层：$D \times H + H \times H$
3. 输出层：$H \times V$
4. 总复杂度：假设$D \approx H$

$$
\begin{aligned}
&1 \times D + D \times H + H \times H + H \times V \\
&\approx 1 \times H + H \times H \times 2 + V \times H \\
&\approx H \times H + V \times H
\end{aligned}
$$

5.3. Skip-gram

符号定义：

1. C：中心词个数
2. D：词向量维度
3. V：单词个数

原始复杂度：

1. 输入：$1 \times D$
2. 输出：通过矩阵$\mathbf{W}_{D \times V}$进行计算得到$\mathbf{u}$，再通过softmax操作得到最终结果$\mathbf{y}$。因此，本次参与计算的参数个数为$D \times V$
3. 总复杂度：$C(1 \times D + D \times V)$

Hierarchical softmax复杂度：

1. $\log_2 V$次二分类，参数θ的长度为D，故复杂度为$C(1 \times D + D \times \log_2 V)$

Negative Sampling复杂度：

1. 正样本个数为1，负样本个数为K，词向量维度为D，故总的复杂度为$C(1 \times D + D \times (K+1))$

5.4. CBOW

周围词个数为N，每个词的向量维度为D，所以输入的参数量为$N \times D$。经过sum操作，得到汇总的词向量，维度为$1 \times D$，再经过权重矩阵$\mathbf{W}_{D \times V}$计算，得到最终预测结果。所以输出层的参数量为$N \times D$。

原始复杂度：$N \times D + D \times V$

HS复杂度：$N \times D + D \times \log_2 V$

NEG复杂度：$N \times D + D \times (K+1)$

5.5. 复杂度总结

如下对各模型复杂度进行总结：

1. NNLM：$Q = N \times D + N \times D \times H + H \times \log_2 V$
2. RNNLM：$Q = H \times H + H \times \log_2 V$
3. Skip_gram+HS：$Q = C(1 \times D + D \times \log_2 V)$
4. Skip_gram+NEG：$Q = C(1 \times D + D \times (K+1))$
5. CBOW+HS：$Q = N \times D + D \times \log_2 V$
6. CBOW+NEG：$Q = N \times D + D \times (K+1)$

6. 问题

1. 为什么可以使用向量内积度量相似度?

   $\mathbf{a}\cdot\mathbf{b}=|\mathbf{a}||\mathbf{b}|\cos\theta$，其中，$\theta$为两向量的夹角。如果向量已被**单位化**，那么向量内积等于$\cos\theta$，此时，两向量越接近，夹角越小，内积越大。

2. 对比Skip-gram和CBOW。训练速度上CBOW更快；对低频词，Skip-gram效果更好，因为Skip-gram是用当前词预测上下文，当前词是低频还是高频没有区别，但是CBOW相当于是完形填空，更倾向于选择常见的词而不是低频词。总体上讲，Skip-gram模型的效果更好。

7. 参考链接

1. https://spaces.ac.cn/archives/4122
2. 源码 GitHub - tmikolov/word2vec: Automatically exported from code.google.com/p/word2vec
3. Embedding从入门到专家必读的十篇论文 - 知乎 (zhihu.com)
4. 词向量模型word2vector详解 - 空空如也_stephen - 博客园 (cnblogs.com)
5. word2vec公式推导及python简单实现 - Kayden_Cheung - 博客园 (cnblogs.com)
6. Python implementation of Word2Vec | Marginalia (claudiobellei.com)
7. The backpropagation algorithm for Word2Vec | Marginalia (claudiobellei.com)
8. word2vec公式推导及python简单实现 - Kayden_Cheung - 博客园 (cnblogs.com)
9. 深度之眼：NLP-baseline 体验课 (deepshare.net)
10. 七月在线：补充视频：陈博士带你从头到尾通透word2vec (julyedu.com)
11. 【机器学习】白板推导系列(三十六) ～ 词向量(Word Vector)哔哩哔哩bilibili
12. 【双语字幕】斯坦福CS224n《深度学习自然语言处理》课程(2019) by Chris Manning哔哩哔哩bilibili