设词典长度是K，文本长度是T, 隐藏层向量长度是H，则输入到隐藏层的矩阵I的形状是$K \times H$，隐藏层到输出的矩阵O的形状是$K \times H$。

输入我们使用`one-hot`向量，长度是 K，记作a(索引为$u : K \times 1$)，隐藏层向量和输出向量分别记作b和c，长度分别是H和K，则正向传播的公式是

$$c = O \cdot (I^T \cdot a) \ (1)$$

。

括号内的$I^T \cdot a$得到隐藏层向量h，但因为a是`one-hot`向量，所以h其实是I中的一行，我们记作I_m，那么我们可以说$h = I_m$。为了之后的反向传播方便，我们将输出向量进行`softmax`处理，将结果归一化道$c \in (0, 1)$之间:

`softmax`公式：$softmax(x_i) = \frac{e^{x_i}}{\sum_1^n e^{x_i}}$

$$d = \frac{e^{c_i}}{\sum_{i=1}^K e^{c_i}} = \frac{1}{\sum_{i=1}^K e^{c_i}} \begin{bmatrix} e^{c_1} \\ \vdots \\ e^{c_k} \end{bmatrix} \ (2)$$

样本$s_n = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$（其中索引$n$元素为1），则:

$$s_n^T \cdot d = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \cdot \frac{1}{\sum_{i=1}^K e^{c_i}} \begin{bmatrix} e^{c_1} \\ \vdots \\ e^{c_k} \end{bmatrix} = \frac{e^{c_n}}{\sum_1^K e^{c_i}} \ (3)$$

其中的c_n是c的n元素，可以通过$s_n^T \cdot c$得到，将c由公式（1）替代，得到:

$$c_n = s_n^T O \cdot (I^T \cdot a) = s_n^T O \cdot (a^T I)^T \ (4)$$

我们已经知道公式（4）中的$I^T a = I_m$是I中的一行，而$s_n^T O$是矩阵O中的一行，我们记作O_n，所以我们可以将公式（4）改写成$c_j = O_n \cdot I_m^T \ (5)$。

将公式（5）带入到公式（3），得到:

$$s_j^T \cdot d = \frac{e^{O_n \cdot I_m^T}}{\sum_1^K e^{c_i}} \ (6)$$

公式（6）是单一背景词对某一中心词（设为l)的似然估计函数，skip-gram模型是通过一个中心词推测出多个背景词，我们假设背景词窗口是z，则这些背景词的似然估计公式如下：

$$P(\text{背景词}|\text{中心词}) = \prod_{n=l-z}^{l+z} \frac{e^{O_n \cdot I_m^T}}{\sum_{i=1}^{K} e^{c_i}} \text{ (7)}$$

单个的背景词的似然估计就是：$P(w_{\text{背景词}}|w_{\text{中心词}}) = \frac{e^{O_n \cdot I_m^T}}{\sum_{i=1}^{K} e^{c_i}} = \frac{e^{c_j}}{\sum_{i=1}^{K} e^{c_i}}$

似然估计值在$(0,1)$，其越接近于1，说明预测越准确。如果我们对似然估计取对数，值在$(-\infty,0)$，1的对数是0，所以似然估计和1的对数差可以表征两者的差距。如果这个差值取负值，则越接近0，说明预测越准。基于上述，我们可以得到我们的损失函数（`loss`）是：

$$\text{loss} = -\log \prod_{n=l-z}^{l+z} \frac{e^{O_n \cdot I_m^T}}{\sum_{i=1}^{K} e^{c_i}} = -\sum_{n=l-z}^{l+z} O_n \cdot I_m^T + 2z \log \sum_{i=1}^{K} e^{c_i} \text{ (8)}$$

如果一次取q个样品进行反向传播学习，则公式（8）可以改写成：

$$\begin{aligned} \text{loss} &= \sum_{r=m}^{m+q} (-\sum_{n=l-z}^{l+z} O_n \cdot I_m^T + 2z \log \sum_{i=1}^{K} e^{c_i}) \\ &= -\sum_{m=g}^{g+q} \sum_{n=l-z}^{l+z} O_n \cdot I_m^T + 2z \sum_{r=m}^{m+q} \log \sum_{i=1}^{K} e^{c_i} \\ &= -\sum_{m=g}^{g+q} \sum_{n=l-z}^{l+z} (O_n \cdot I_m^T - \log \sum_{i=0}^{K} e^{c_i}) \\ &= -\sum_{m=g}^{g+q} \sum_{n=l-z}^{l+z} logP(w_{\text{背景词}}|w_{\text{中心词}}) \end{aligned} \qquad (9)$$

在反向传播中，我们需要找到损失函数对于矩阵O的梯度：

$$\begin{aligned}\frac{\partial logP(w_{\text{背景词}}|w_{\text{中心词}})}{\partial O_n} &= I_m^T - \frac{\partial \log \sum_{i=1}^{K} e^{c_i}}{\partial O_n} \\ &= I_m^T - \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot \frac{\partial \sum_{i=1}^{K} e^{c_i}}{\partial O_n} \\ &= I_m^T - \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot \frac{(\partial e^{O_1 I_m^T} + \cdots + \partial e^{O_n I_m^T} + \cdots + \partial e^{O_K I_m^T})}{\partial O_n} \\ &= I_m^T - I_m^T \cdot \frac{e^{O_n I_m^T}}{\sum_{i=1}^{K} e^{c_i}} \\ &= (1 - \frac{e^{O_n I_m^T}}{\sum_{i=1}^{K} e^{c_i}}) \cdot I_m^T \\ &= (1 - P(w_{\text{背景词}}|w_{\text{中心词}})) \cdot I_m^T\end{aligned}$$

得到损失函数对于矩阵I的梯度：

$$\begin{aligned}\frac{\partial P(w_{\text{背景词}}|w_{\text{中心词}})}{\partial I_m^T} &= O_n + \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot \frac{\partial \sum_{i=1}^{K} e^{c_i}}{\partial l_m^T} \\ &= O_n - \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot \frac{\partial(e^{c_1} + \cdots + e^{c_K})}{\partial I_m^T} \\ &= O_n - \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot (\frac{\partial e^{c_1}}{\partial I_m^T} + \cdots + \frac{\partial e^{c_K}}{\partial I_m^T}) \\ &= O_n - \frac{1}{\sum_{i=1}^{K} e^{c_i}} \cdot (e^{c_1} O_1 + \cdots + e^{c_K} \cdot O_K) \\ &= O_n - \sum_{i=1}^{K} \frac{e^{c_i} \cdot O_i}{\sum_{i=1}^{K} e^{c_i}}\end{aligned}$$