

6 Skip-Gram与CBOW的数学推导

6.1 基本的假设

他们的任务都是学习两个权重矩阵，词向量矩阵 C ，和输出权重矩阵 θ ，通过两个矩阵可以表示词 w 上下文中有词 u 的概率，由于对单个词来说，这是一个伯努利分布（ w 的上下文有或者没有 u ），所以用sigmoid归一化：

$$P(u \in context(w)|w; C, \theta) = \text{sigmoid}(C_w^T \cdot \theta_u)$$

其中 C_w 是词 w 对应的词向量，也就是矩阵 C 中的一行， θ_u 是输出矩阵的一行，它也同样与一个词 u 对应，因为某个词向量与输出矩阵的某一行做点积实际上得到了输出向量中对词 u 的概率。

6.2 CBOW

CBOW通过确定的 w 的上下文 $context(w)$ 来预测 $w = u$ 的概率，对每一个属于 w 上下文的词 k ，按照上面的假设给出 u 在它的上下文范围内的概率，且对每个属于 $context(w)$ 的词 k 的预测都能增强或削弱这种概率，不妨假设对所有 k 的预测结果互相独立：

$$\begin{aligned} P(w = u|context(w); C, \theta) \\ &= \prod_{k \in context(w)} p(u \in context(k)|w; C, \theta) \\ &= \prod_{k \in context(w)} \text{sigmoid}(C_k^T \cdot \theta_u) \end{aligned}$$

若采用NCE损失，在词库上以任意分布(Noise Contrast Estimation要求分布律覆盖所有训练样本，但在实际中这个限定条件很弱)做负采样采样，记所有采样以及正确标记的集合为 N_w ，对 N_w 中的任意词 u ，用 l_u^w 做类别标记，1为正类，0为负类，只有 $u = w$ 时 $l_u^w = 1$ ，希望正类的似然最大而负类似然最小，那么对单个样本所有采样的似然：

$$\begin{aligned} g(w; C, \theta) \\ &= \prod_{u \in N_w} \{l_u^w \cdot p(u|context(w); C, \theta) + (1 - l_u^w)(1 - p(u|context(w); C, \theta))\} \\ &= \prod_{u \in N_w} \{l_u^w \prod_{k \in context(w)} \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w)(1 - \prod_{k \in context(w)} \text{sigmoid}(C_k^T \cdot \theta_u))\} \end{aligned}$$

关于NCE损失的详细推导可以参考(<https://github.com/Y-WJ/NLP-PLAYER/blob/master/%E4%B8%80%E4%BA%9B%E8%BE%85%E5%8A%A9%E6%96%87%E6%A1%A3/NCE.pdf>),对于整个训练集上的样本NCE损失应该写成：

$$\begin{aligned} G(C, \theta) &= \prod_{w \in V} g(w; C, \theta) \\ &= \prod_{w \in V} \prod_{u \in N_w} \{l_u^w \prod_{k \in context(w)} \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w)(1 - \prod_{k \in context(w)} \text{sigmoid}(C_k^T \cdot \theta_u))\} \end{aligned}$$

取对数：

$$\begin{aligned} L &= \ln G(C, \theta) \\ &= \sum_{w \in V} \sum_{u \in N_w} \{l_u^w \sum_{k \in context(w)} \ln \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w)(1 - \sum_{k \in context(w)} \ln \text{sigmoid}(C_k^T \cdot \theta_u))\} \\ &= \sum_{w \in V} \sum_{u \in N_w} \{ \sum_{k \in context(w)} (l_u^w - 1) \cdot C_k^T \cdot \theta_u - \sum_{k \in context(w)} \ln(1 + e^{C_k^T \cdot \theta_u}) \} \\ &= \sum_{w \in V} \sum_{u \in N_w} \sum_{k \in context(w)} \{l_u^w \cdot C_k^T \cdot \theta_u - C_k^T \cdot \theta_u - \ln(1 + e^{C_k^T \cdot \theta_u})\} \end{aligned}$$

L 就是优化的目标，当然，若采用随机梯度下降， V 是从训练集中随机选取的一批样本，对 θ 求梯度：

$$\begin{aligned}\frac{\partial L}{\partial \theta} &= \sum_{w \in V} \sum_{u \in N_w} \sum_{k \in \text{context}(w)} \frac{\partial \{l_u^w \cdot C_k^T \cdot \theta_u - C_k^T \cdot \theta_u - \ln(1 + e^{-C_k^T \cdot \theta_u})\}}{\partial \theta_u} \\ &= \sum_{w \in V} \sum_{u \in N_w} \sum_{k \in \text{context}(w)} \{l_u^w \cdot C_k - C_k + (1 - \text{sigmoid}(C_k^T \cdot \theta_u)) \cdot C_k\} \\ &= \sum_{w \in V} \sum_{u \in N_w} \sum_{k \in \text{context}(w)} \{l_u^w - \text{sigmoid}(C_k^T \cdot \theta_u)\} C_k\end{aligned}$$

由于 θ_u 与 C_k 在 L 中地位是等价的，交换两者可以得出对偶式：

$$\frac{\partial L}{\partial C_k} = \sum_{w \in V} \sum_{u \in N_w} \sum_{k \in \text{context}(w)} \{l_u^w - \text{sigmoid}(C_k^T \cdot \theta_u)\} \theta_u$$

在学习率 α 下，采用随机梯度下降更新权重以及词向量的算法如下：

```

C,  $\theta$  初始化
while(!终止条件) do
{
    随机从V中选取样本集batch;
    for( $w \in \text{batch}$ ) do:
    {
        抽样得到 $N_w$ 
        for( $u \in N_w$ ) do:
        {
            for( $k \in \text{context}(w)$ ) do:
            {
                 $g = l_u^w - \text{sigmoid}(C_k^T \cdot \theta_u)$ 
                 $\Delta C_k = g \cdot \theta_u$ 
                 $\Delta \theta_u += g \cdot C_k$ 
                 $C_k += \alpha \cdot \Delta C_k$ 
            }
             $\theta_u^+ = \alpha \cdot \Delta \theta_u$ 
        }
    }
}

```

注意对每个 $k \in \text{context}(w)$, C_k 会累加地影响对 θ_u 的梯度，因此在更新 C_k 前要先记录这种影响，这等效于先计算出所有向量的更新量，再对所有向量更新。

6.3 Skip-Gram

Skip-Gram确定了当前词 w 而预测上下文的似然,对每一个属于上下文的词 u ，希望由上面概率计算出的似然最大

$$\prod_{u \in \text{context}(w)} P(u \in \text{context}(w) | w; C, \theta) = \prod_{u \in \text{context}(w)} \text{sigmoid}(C_u^T \cdot \theta_w)$$

同样考虑NCE损失，对每个 w 取若干负采样，与其真实上下文构成集合 N_w ,用 l_u^w 做标记，与CBOW不同的是，由于 w 的上下文可能不止一个词，因此有多个可能的 u 使 $l_u^w = 1$,其余负采样所得的 u 均使 $l_u^w = 0$,于是对单个样本有：

$$\begin{aligned}g(w; C, \theta) &= \prod_{u \in N_w} \{l_u^w \cdot p(u | \text{context}(w); C, \theta) + (1 - l_u^w)(1 - p(u | \text{context}(w); C, \theta))\} \\ &= \prod_{u \in N_w} \{l_u^w \cdot \text{sigmoid}(C_u^T \cdot \theta_w) + (1 - l_u^w)(1 - \text{sigmoid}(C_u^T \cdot \theta_w))\}\end{aligned}$$

在整个训练集上，NCE损失为：

$$\begin{aligned} G(C, \theta) &= \prod_{w \in V} g(w; C, \theta) \\ &= \prod_{w \in V} \prod_{u \in N_w} \{l_u^w \cdot \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w)(1 - \text{sigmoid}(C_k^T \cdot \theta_u))\} \end{aligned}$$

取对数：

$$\begin{aligned} L &= \ln G(C, \theta) \\ &= \sum_{w \in V} \ln g(w; C, \theta) \\ &= \sum_{w \in V} \sum_{u \in N_w} \ln \{l_u^w \cdot \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w)(1 - \text{sigmoid}(C_k^T \cdot \theta_u))\} \\ &= \sum_{w \in V} \sum_{u \in N_w} \{l_u^w \ln \text{sigmoid}(C_k^T \cdot \theta_u) + (1 - l_u^w) \ln(1 - \text{sigmoid}(C_k^T \cdot \theta_u))\} \\ &= \sum_{w \in V} \sum_{u \in N_w} \{l_u^w \cdot C_k^T \cdot \theta_u - C_k^T \cdot \theta_u - \ln(1 + e^{-C_k^T \cdot \theta_u})\} \end{aligned}$$

梯度计算与更新算法与CBOW的情况类似，只是少了对 k 求和的一层循环，这里略去不讲。