Word2Vec

背景 onehot 用向量表达词意义

两个模型

Skip-gram

总词数 背景词 中心词

$$\max \prod_{t=1}^{T} \prod_{-m \le j \le m,\ j \neq 0} P(w^{(t+j)} \mid w^{(t)}),$$

Skip-gram 最大似然估计

每个词都是独立

等价 log

$$\min -\sum_{t=1}^{T} \sum_{-m \le j \le m,\ j \neq 0} \log P(w^{(t+j)} \mid w^{(t)}).$$

损失函数

套入向量

引入所有的词，导致梯度训练很慢

背景 中心

$$\mathbb{P}(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

u0是WO的向量表达
Vc是Wc的向量表达

$$\log P(w_o \mid w_c) = \boldsymbol{u}_o^\top \boldsymbol{v}_c - \log\left(\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \boldsymbol{v}_c)\right)$$

求微分进行梯度下降

$$\frac{\partial \log P(w_o \mid w_c)}{\partial \boldsymbol{v}_c} = \boldsymbol{u}_o - \frac{\sum_{j \in \mathcal{V}} \exp(\boldsymbol{u}_j^\top \boldsymbol{v}_c) \boldsymbol{u}_j}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \boldsymbol{v}_c)} = \boldsymbol{u}_o - \sum_{j \in \mathcal{V}} \left(\frac{\exp(\boldsymbol{u}_j^\top \boldsymbol{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \boldsymbol{v}_c)} \right) \boldsymbol{u}_j$$

$$\frac{\partial \log P(w_c \mid \mathcal{W}_o)}{\partial \boldsymbol{v}_{o_i}} = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} \frac{\exp(\boldsymbol{u}_j^\top \bar{\boldsymbol{v}}_o)\boldsymbol{u}_j}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \bar{\boldsymbol{v}}_o)}\right) = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} P(w_j \mid \mathcal{W}_o)\boldsymbol{u}_j\right).$$

CBOW

$$\max \prod_{t=1}^{T} P(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}).$$

$$\min -\sum_{t=1}^{T} \log P(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}).$$

$$P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{\exp\left(\frac{1}{2m}\boldsymbol{u}_c^\top (\boldsymbol{v}_{o_1} + \ldots + \boldsymbol{v}_{o_{2m}})\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\boldsymbol{u}_i^\top (\boldsymbol{v}_{o_1} + \ldots + \boldsymbol{v}_{o_{2m}})\right)}.$$

随机梯度

$$\frac{\partial \log P(w_c \mid \mathcal{W}_o)}{\partial \boldsymbol{v}_{o_i}} = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} \frac{\exp(\boldsymbol{u}_j^\top \bar{\boldsymbol{v}}_o)\boldsymbol{u}_j}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \bar{\boldsymbol{v}}_o)}\right) = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} P(w_j \mid \mathcal{W}_o)\boldsymbol{u}_j\right).$$

注：如果用梯度下降的话，开销很大，
实际很难学习，所以引出下面训练方法

两个近似的训练方法

负采样

负采样

我们以跳字模型为例讨论负采样。

词典$\mathcal{V}$大小之所以会在目标函数中出现，是因为中心词w_c生成背景词w_o的概率$\mathbb{P}(w_o \mid w_c)$使用了softmax，而softmax正是考虑了背景词可能是词典中的任一词，并体现在softmax的分母上。

我们不妨换个角度，假设中心词w_c生成背景词w_o由以下两个相互独立事件联合组成来近似

- 中心词w_c和背景词w_o同时出现在该训练数据窗口
- 中心词w_c和第1个噪声词w_1不同时出现在该训练数据窗口（噪声词w_1按噪声词分布$\mathbb{P}(w)$随机生成）
- ...
- 中心词w_c和第K个噪声词w_K不同时出现在该训练数据窗口（噪声词w_K按噪声词分布$\mathbb{P}(w)$随机生成）

我们可以使用$\sigma(x) = 1/(1+\exp(-x))$函数来表达中心词$w_c$和背景词$w_o$同时出现在该训练数据窗口的概率：

公式

$$P(D=1 \mid w_c, w_o) = \sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c),$$

max $$\prod_{t=1}^{T} \prod_{-m \le j \le m,\, j \ne 0} P(D=1 \mid w^{(t)}, w^{(t+j)}).$$

然而，以上模型中包含的事件仅考虑了正类样本。这导致当所有词向量相等且值为无穷大时，以上的联合概率才被最大化为1。很明显，这样的词向量毫无意义。负采样通过采样并添加负类样本使目标函数更有意义。设背景词w_o出现在中心词w_c的一个背景窗口为事件P，我们根据分布$P(w)$采样K个未出现在该背景窗口中的词，即噪声词。设噪声词w_k（$k=1,\ldots,K$）不出现在中心词w_c的该背景窗口为事件N_k。假设同时含有正类样本和负类样本的事件$P, N_1, \ldots, N_K$相互独立。负采样将以上需要最大化的仅考虑正类样本的联合概率改写为

max $$\log \mathbb{P}(w_o \mid w_c) = \log[\mathbb{P}(D=1 \mid w_o, w_c) \prod_{k=1, w_k \sim \mathbb{P}(w)}^{K} \mathbb{P}(D=0 \mid w_k, w_c)]$$

max $$\log \mathbb{P}(w_o \mid w_c) = \log \frac{1}{1+\exp(-\mathbf{u}_o^\top \mathbf{v}_c)} + \sum_{k=1, w_k \sim \mathbb{P}(w)}^{K} \log[1 - \frac{1}{1+\exp(-\mathbf{u}_k^\top \mathbf{v}_c)}]$$

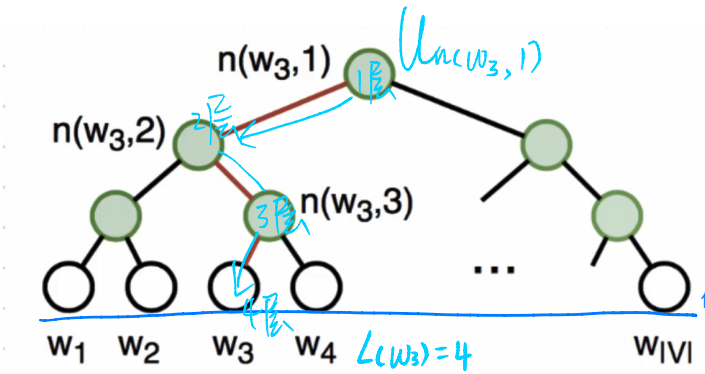
min $$-\log \mathbb{P}(w_o \mid w_c) = -\log \frac{1}{1+\exp(-\mathbf{u}_o^\top \mathbf{v}_c)} - \sum_{k=1, w_k \sim \mathbb{P}(w)}^{K} \log \frac{1}{1+\exp(\mathbf{u}_k^\top \mathbf{v}_c)}$$

注：计算大小从词数量V降到K

Trick:噪声词设为单词概率的3/4次方（0.99 0.01 各己的3/4次方）

层次Soft max

用了哈夫曼树，根据词频（字频）构造



$$P(w_o \mid w_c) = \prod_{j=1}^{L(w_o)-1} \sigma\left([\![n(w_o, j+1) = \text{leftChild}(n(w_o, j))]\!] \cdot \mathbf{u}_{n(w_o,j)}^\top \mathbf{v}_c\right),$$

例子

$$\mathbb{P}(w_3 \mid w_i) = \sigma(\mathbf{u}_{n(w_3,1)}^\top \mathbf{v}_i)) \cdot \sigma(-\mathbf{u}_{n(w_3,2)}^\top \mathbf{v}_i)) \cdot \sigma(\mathbf{u}_{n(w_3,3)}^\top \mathbf{v}_i))$$

按词频公式去掉

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

用有限的非叶子结点向量表达了所要生成表达词语的向量

Question:
1、如何处理无意义的高频词
2、如何学习词组
3、噪声选3/4平方原因

unigram and bigram count
如果某些词语经常在一起就判定他们是短语

输入两个词若大于某值就认定它们在一起

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}.$$

通常拿2-4个词

在取样的过程中，我们把每个词频的值取四分之三次方的时候效果最好。
这样做是因为对频率较大的数进行取 3/4 次幂，原数字损失的多，反之较小的，去掉的就少，这样做是一种平滑方式。