Word2vec公式推导

(objective function + 梯度更新公式)

skip-gram model:

最大化似然

$$\max J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} p(w_{t+j} \mid w_t ; \theta) \quad \Rightarrow \theta: \text{word vectors}$$

除了outside word其他位置为0

实际为最小化交叉熵.

$$\Leftrightarrow \text{minimize } J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log p(w_{t+j} \mid w_t ; \theta) \quad - p \cdot \log(p)$$

$$\left(p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w v_c)} \right)$$

: [V : vocabulary size
u_o : word vector for "outside" words
v_c : word vector for "center word"]

梯度下降

$$A: \frac{\partial}{\partial v_c} \log \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)} = \frac{\partial}{\partial v_c} \log \exp(u_o^T v_c) - \frac{\partial}{\partial v_c} \log \sum_{w=1}^{V} \exp(u_w^T v_c)$$

$$\left[① \ (a^x)' = a^x \ln a \qquad ② \ (e^x)' = e^x \qquad ③ \ (\ln x)' = \frac{1}{x} \right]$$

$$\frac{\partial}{\partial v_c} \log \exp(u_o^T v_c) = \frac{\partial}{\partial v_c} (u_o^T v_c) = u_o$$

$$\frac{\partial}{\partial v_c} \log \sum_{w=1}^{V} \exp(u_w^T v_c) = \frac{1}{\sum_{w=1}^{V} \exp(u_w^T v_c)} \cdot \underbrace{\sum_{w=1}^{V} \frac{\partial}{\partial v_c} \exp(u_w^T v_c)}_{= \sum_{w=1}^{V} \left(\exp(u_w^T v_c) \cdot u_w \right)}$$

Therefore:

$$A = u_o - \frac{\sum_{w=1}^{V} \left(\exp(u_w^T \cdot v_c) \cdot u_w \right)}{\sum_{w=1}^{V} \exp(u_w^T \cdot v_c)} = u_o - \sum_{w=1}^{V} \frac{\exp(u_w^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)} \cdot u_w$$

$$= u_o - \sum_{w=1}^{V} P(w|c) \cdot u_w$$

B: $\frac{\partial}{\partial u_o}\log\frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V}\exp(u_w^T v_c)} = \frac{\partial}{\partial u_o}\log\exp(u_o^T v_c) - \frac{\partial}{\partial u_o}\log\sum_{w=1}^{V}\exp(u_w^T v_c)$

$(\ln x)' = \frac{1}{x}$

$= \frac{\partial}{\partial u_o}u_o^T v_c - \frac{1}{\sum_{w=1}^{V}\exp(u_w^T v_c)}\cdot\frac{\partial}{\partial u_o}\left(\sum_{w=1}^{V}\exp(u_w^T v_c)\right)$

$= v_c - \frac{1}{\sum_{w=1}^{V}\exp(u_w^T v_c)}\cdot\frac{\partial}{\partial u_o}\exp(u_o^T v_c) = v_c - \frac{\exp(u_o^T v_c)\cdot v_c}{\sum_{w=1}^{V}\exp(u_w^T v_c)}$

$= v_c - p(u_o|v_c)\cdot v_c$

△交叉熵：机器/深度学习中常用来描述目标与预测值差距，即定义目标函数

$\boxed{\text{信息量}}$：一个事件发生的概率越低，获取到的信息量就越大

A：巴西队进入2018年世界杯决赛　B：中国队进入2018年世界杯决赛

显然~~A~~B发生的概率~~高~~低，因此~~A~~B的信息量更大　$I(x=A) = -\log(p(A))$

$\boxed{\text{熵}}$：所有信息量的期望　$H(x) = -\sum_{i=1}^{n}p(x_i)\log(p(x_i)) \geq 0$

$\boxed{\text{相对熵}}$：也称KL散度(Kullback-Leibler divergence)

用于衡量两个分布的差异

机器学习中，P⇒真实分布[1,0,0]；Q⇒模型预测分布[0.7,0.2,0.1]

显然Q用来描述样本分类不够完美，信息量不足，需要额外的"信息增量"

$\underbrace{D_{KL}(p||q)}_{\text{用p比用q多出来的信息增量}} = \sum_{i=1}^{n}p(x_i)\log\frac{p(x_i)}{q(x_i)} \geq 0$

交叉熵　$D_{KL}(p||q) = \sum_{i=1}^{n}p(x_i)\log p(x_i) - \sum_{i=1}^{n}p(x_i)\log q(x_i)$

$= \underbrace{-H(p(x))} - \sum_{i=1}^{n}p(x_i)\log q(x_i)$

机器/深度学习中此项为固定值，只需优化 $-\sum_{i=1}^{n}p(x_i)\log q(x_i)$ 使其最小