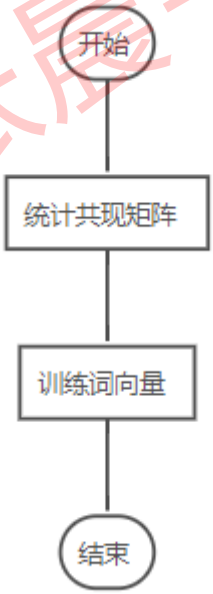
5. Glove

1．glove的输入输出及目标是什么

- 模型目标：进行词的向量化表示，使得向量之间尽可能多地蕴含语义和语法的信息。
- 输入：语料库
- 输出：词向量
- 方法概述：首先基于语料库构建词的共现矩阵，然后基于共现矩阵和GloVe模型学习词向量。
- Glove 算法结合了**矩阵分解（LSA）**和**浅窗口方法（word2vec）**的优点，充分地利用了全局的**统计信息**和**局部上下文窗口**的优势

2．GloVe构建过程是怎样的



（1）根据语料库构建一个共现矩阵，矩阵中的每一个元素 X_{ij} 代表单词 i 和上下文单词 j 在特定大小的上下文窗口内共同出现的次数。

（2）构建词向量（Word Vector）和共现矩阵之间的近似关系，其目标函数为：

$$J=\sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j+b_i+b_j-\log X_{ij}\right)^2$$

这个loss function的基本形式就是最简单的mean square loss，只不过在此基础上加了一个权重函数 $f(x_{ij})$：

$$f(x)=\begin{cases}(x/x_{\max})^{\alpha} & \text{if } x<x_{\max}\\ 1 & \text{otherwise.}\end{cases}$$

根据实验发现 $x_{\max}$ 的值对结果的影响并不是很大，原作者采用了 $x_{\max}=100$ 。而 $\alpha=3/4$ 时的结果要比 $\alpha=1$ 时要更好。下面是 $\alpha=3/4$ 时 $f(x)$ 的函数图象，可以看出对于较小的 X_{ij} ，权值也较小。这个函数图像如下所示：

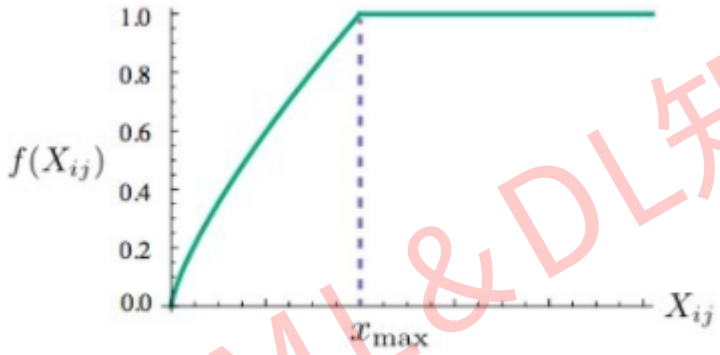


Figure 1: Weighting function f with $\alpha = 3/4$.

3．GloVe的训练过程是怎样的

1. 实质上还是监督学习：虽然glove不需要人工标注为无监督学习，但实质还是有label就是 $log(X_{ij})$ 。
2. 向量 w 和 $\tilde{w}$ 为学习参数，本质上与监督学习的训练方法一样，采用了AdaGrad的梯度下降算法，对矩阵 X 中的所有非零元素进行随机采样，学习曲率（learning rate）设为0.05，在vector size小于300的情况下迭代了50次，其他大小的vectors上迭代了100次，直至收敛。
3. 最终学习得到的是两个词向量是 $\tilde{w}$ 和 w ，因为 X 是对称的（symmetric），所以从原理上讲 $\tilde{w}$ 和 w ，是也是对称的，他们唯一的区别是初始化的值不一样，而导致最终的值不一样。所以这两者其实是等价的，都可以当成最终的结果来使用。但是为了提高鲁棒性，我们最终会选择两者之和 $w+\tilde{w}$ 作为最终的vector（两者的初始化不同相当于加了不同的随机噪声，所以能提高鲁棒性）。

4．Glove目标函数是如何确定的?

通过概率比而不是概率本身去学习词向量，结果更准确

思想：假设我们已经得到了词向量，如果我们用词向量v_i、v_j、v_k通过某种函数计算$ratio_{i,j,k}$，能够同样得到这样的规律的话，就意味着我们词向量与共现矩阵具有很好的一致性，也就说明我们的词向量中蕴含了共现矩阵中所蕴含的信息。
设用词向量v_i、v_j、v_k计算$ratio_{i,j,k}$的函数为$g(v_i, v_j, v_k)$（我们先不去管具体的函数形式），那么应该有：

$$\frac{P_{i,k}}{P_{j,k}} = ratio_{i,j,k} = g(v_i, v_j, v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

即二者应该尽可能地接近；
很容易想到用二者的差方来作为代价函数：

$$J = \sum_{i,j,k}^{N} (\frac{P_{i,k}}{P_{j,k}} - g(v_i, v_j, v_k))^2$$

但是仔细一看，模型中包含3个单词，这就意味着要在 $N * N * N$的复杂度上进行计算，太复杂了，最好能再简单点。
现在我们来仔细思考 $g(v_i, v_j, v_k)$，或许它能帮上忙；
作者的脑洞是这样的：
1. 要考虑单词 i和单词j之间的关系，那 $g(v_i, v_j, v_k)$中大概要有这么一项吧： $v_i - v_j$；嗯，合理，在线性空间中考察两个向量的相似性，不失线性地考察，那么 $v_i - v_j$大概是个合理的选择；
2. $ratio_{i,j,k}$是个标量，那么 $g(v_i, v_j, v_k)$最后应该是个标量啊，虽然其输入都是向量，那内积应该是合理的选择，于是应该有这么一项吧： $(v_i - v_j)^T v_k$。
3. 然后作者又往 $(v_i - v_j)^T v_k$的外面套了一层指数运算 $exp()$，得到最终的 $g(v_i, v_j, v_k) = exp((v_i - v_j)^T v_k)$；
最关键的第3步，为什么套了一层 $exp()$?
套上之后，我们的目标是让以下公式尽可能地成立：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp((v_i - v_j)^T v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp(v_i^T v_k - v_j^T v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = \frac{exp(v_i^T v_k)}{exp(v_j^T v_k)}$$

然后就发现找到简化方法了：只需要让上式分子对应相等，分母对应相等，即：

$$P_{i,k} = exp(v_i^T v_k) 并且 P_{j,k} = exp(v_j^T v_k)$$

然而分子分母形式相同，就可以把两者统一考虑了，即：

$$P_{i,j} = exp(v_i^T v_j)$$

本来我们追求：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

现在只需要追求：

$$P_{i,j} = exp(v_i^T v_j)$$

两边取个对数：

$$log(P_{i,j}) = v_i^T v_j$$

那么代价函数就可以简化为：

$$J = \sum_{i,j}^{N} (log(P_{i,j}) - v_i^T v_j)^2$$

现在只需要在 $N * N$的复杂度上进行计算，而不是 $N * N * N$，现在关于为什么第3步中，外面套一层 $exp()$就清楚了，正是因为套了一层 $exp()$，才使得差形式变成商形式，进而等式两边分子分母对应相等，进而简化模型。

然而，出了点问题。
仔细看这两个式子：

$$log(P_{i,j}) = v_i^T v_j 和 log(P_{j,i}) = v_j^T v_i$$

$log(P_{i,j})$不等于 $log(P_{j,i})$但是 $v_i^T v_j$等于 $v_j^T v_i$；即等式左侧不具有对称性，但是右侧具有对称性。
数学上出了问题。
补救一下好了。
现将代价函数中的条件概率展开：

$$log(P_{i,j}) = v_i^T v_j$$

即为:

$$log(X_{i,j}) - log(X_i) = v_i^T v_j$$

将其变为：

$$log(X_{i,j}) = v_i^T v_j + b_i + b_j$$

即添了一个偏差项 b_j，并将 $log(X_i)$吸收到偏差项 b_i中。
于是代价函数就变成了：

$$J = \sum_{i,j}^{N} (v_i^T v_j + b_i + b_j - log(X_{i,j}))^2$$

然后基于出现频率越高的词对儿权重应该越大的原则，在代价函数中添加权重项，于是代价函数进一步完善：

$$J = \sum_{i,j}^{N} f(X_{i,j})(v_i^T v_j + b_i + b_j - log(X_{i,j}))^2$$

具体权重函数应该是怎么样的呢？
首先应该是非减的，其次当词频过高时，权重不应过分增大，作者通过实验确定权重函数为：

$$f(x) = \begin{cases} (x/xmax)^{0.75}, & \text{if } x < xmax \\ 1, & \text{if } x >= xmax \end{cases}$$

5．GloVe 与 Word2Vec 进行对比

- 两者最直观的区别在于，word2vec是“predictive”的模型，而GloVe是“count-based”的模型
- 相比于word2vec，因为golve更容易并行化，所以速度更快
- 由于GloVe算法本身使用了全局信息，自然内存费的也就多一些，相比之

下，word2vec在这方面节省了很多资源

- Word2Vec 有神经网络，GloVe 没有；
- Word2Vec 关注了局部信息，GloVe 关注局部信息和全局信息；
- 都有滑动窗口但 Word2Vec 是用来训练的，GloVe 是用来统计共现矩阵的；
- GloVe 的结构比 Word2Vec 还要简单，所以速度更快；

6．将 GLoVe 与 SVD 进行对比

- SVD 所有单词统计权重一致，GloVe 对此进行了优化；
- GloVe 使用比值而没有直接使用共现矩阵。

7．glove的优点及缺点

优点：

- 相比于word2vec，因为golve更容易并行化，所以速度更快
- 使用了全局信息
- GloVe 的结构比 Word2Vec 还要简单，所以速度更快

缺点：

- 静态向量，没有一词多义的性质
- 内存占用更大

- https://zhuanlan.zhihu.com/p/56382372
  <https://zhuanlan.zhihu.com/p/56382372>
- https://blog.csdn.net/coderTC/article/details/73864097
  <https://blog.csdn.net/coderTC/article/details/73864097>
- https://www.it610.com/article/1306005579872899072.htm
  <https://www.it610.com/article/1306005579872899072.htm>