

刘建平Pinard

十年研发，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

博客园 首页 新随笔 联系 订阅 管理

word2vec原理(二) 基于Hierarchical Softmax的模型

word2vec原理(一) CBOW与Skip-Gram模型基础

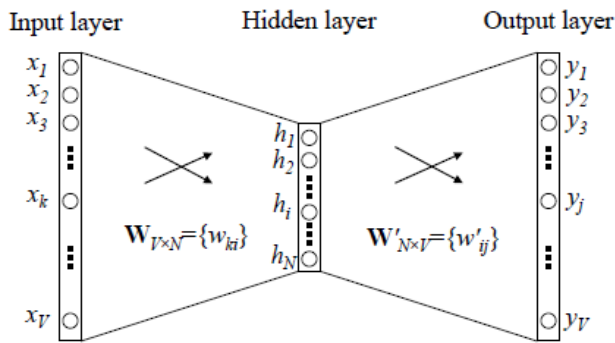
word2vec原理(二) 基于Hierarchical Softmax的模型

word2vec原理(三) 基于Negative Sampling的模型

在word2vec原理(一) CBOW与Skip-Gram模型基础中，我们讲到了使用神经网络的方法来得到词向量语言模型的原理和一些问题，现在我们开始关注word2vec的语言模型如何改进传统的神经网络的方法。由于word2vec有两种改进方法，一种是基于Hierarchical Softmax的，另一种是基于Negative Sampling的。本文关注于基于Hierarchical Softmax的改进方法，在下一篇讨论基于Negative Sampling的改进方法。

# 1. 基于Hierarchical Softmax的模型概述

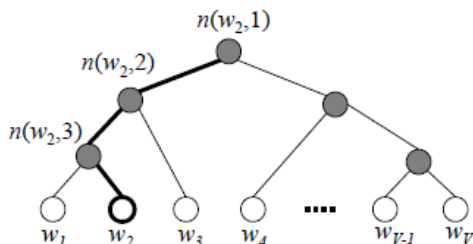
我们先回顾下传统的神经网络词向量语言模型，里面一般有三层，输入层（词向量），隐藏层和输出层（softmax层）。里面最大的问题在于从隐藏层到输出的softmax层的计算量很大，因为要计算所有词的softmax概率，再去找概率最大的值。这个模型如下图所示。其中 $V$ 是词汇表的大小，



word2vec对这个模型做了改进，首先，对于从输入层到隐藏层的映射，没有采取神经网络的线性变换加激活函数的方法，而是采用简单的对所有输入词向量求和并取平均的方法。比如输入的是三个4维词向量： $(1, 2, 3, 4), (9, 6, 11, 8), (5, 10, 7, 12)$ ，那么我们word2vec映射后的词向量就是 $(5, 6, 7, 8)$ 。由于这里是从多个词向量变成了一个词向量。

第二个改进就是从隐藏层到输出的softmax层这里的计算量个改进。为了避免要计算所有词的softmax概率，word2vec采样了霍夫曼树来代替从隐藏层到输出softmax层的映射。我们在上一节已经介绍了霍夫曼树的原理。如何映射呢？这里就是理解word2vec的关键所在了。

由于我们把之前所有都要计算的从输出softmax层的概率计算变成了一颗二叉霍夫曼树，那么我们的softmax概率计算只需要沿着树形结构进行就可以了。如下图所示，我们可以沿着霍夫曼树从根节点一直走到我们的叶子节点的词 $w_2$ 。



和之前的神经网络语言模型相比，我们的霍夫曼树的所有内部节点就类似之前神经网络隐藏层的神经元，其中，根节点的词向量对应我们的投影后的词向量，而所有叶子节点就类似于之前神经网络softmax输出层的神经元，叶子节点的个数就是词汇表的大小。在霍夫曼树中，隐藏层到输出层的softmax映射不是一下子完成的，而是沿着霍夫曼树一步步完成的，因此这种softmax取名为"Hierarchical Softmax"。

## 公告

★珠江追梦，饮岭南茶，恋鄂北家★  
你的支持是我写作的动力：



昵称：刘建平Pinard  
园龄：2年8个月  
粉丝：4329  
关注：15  
+加关注

## 随笔分类(135)

- 0040. 数学统计学(9)
- 0081. 机器学习(71)
- 0082. 深度学习(11)
- 0083. 自然语言处理(23)
- 0084. 强化学习(19)
- 0121. 大数据挖掘(1)
- 0122. 大数据平台(1)

## 随笔档案(135)

- 2019年7月 (1)
- 2019年6月 (1)
- 2019年5月 (2)
- 2019年4月 (3)
- 2019年3月 (2)
- 2019年2月 (2)
- 2019年1月 (2)
- 2018年12月 (1)
- 2018年11月 (1)
- 2018年10月 (3)
- 2018年9月 (3)
- 2018年8月 (4)
- 2018年7月 (3)
- 2018年6月 (3)
- 2018年5月 (3)
- 2017年8月 (1)
- 2017年7月 (3)
- 2017年6月 (8)
- 2017年5月 (7)
- 2017年4月 (5)
- 2017年3月 (10)
- 2017年2月 (7)
- 2017年1月 (13)
- 2016年12月 (17)
- 2016年11月 (22)

如何“沿着霍夫曼树一步步完成”呢？在word2vec中，我们采用了二元逻辑回归的方法，即规定沿着左子树走，那么就是负类(霍夫曼树编码1)，沿着右子树走，那么就是正类(霍夫曼树编码0)。判别正类和负类的方法是使用sigmoid函数，即：

$$P(+) = \sigma(x_u^T \theta) = \frac{1}{1 + e^{-x_u^T \theta}}$$

其中 $x_u$ 是当前内部节点的词向量，而 $\theta$ 则是我们需要从训练样本求出的逻辑回归的模型参数。

使用霍夫曼树有什么好处呢？首先，由于是二叉树，之前计算量为 $V$ ,现在变成了 $\log_2 V$ 。第二，由于使用霍夫曼树是高频的词靠近树根，这样高频词需要更少的时间会被找到，这符合我们的贪心优化思想。

容易理解，被划分为左子树而成为负类的概率为 $P(-) = 1 - P(+)$ 。在某一个内部节点，要判断是沿左子树还是右子树走的标准就是看 $P(-)$ ,  $P(+)$ 谁的概率值大。而控制 $P(-)$ ,  $P(+)$ 谁的概率值大的因素一个是当前节点的词向量，另一个是当前节点的模型参数 $\theta$ 。

对于上图中的 $w_2$ ，如果它是一个训练样本的输出，那么我们期望对于里面的隐藏节点 $n(w_2, 1)$ 的 $P(-)$ 概率大， $n(w_2, 2)$ 的 $P(-)$ 概率大， $n(w_2, 3)$ 的 $P(+)$ 概率大。

回到基于Hierarchical Softmax的word2vec本身，我们的目标就是找到合适的所有节点的词向量和所有内部节点 $\theta$ ，使训练样本达到最大似然。那么如何达到最大似然呢？

## 2. 基于Hierarchical Softmax的模型梯度计算

我们使用最大似然法来寻找所有节点的词向量和所有内部节点 $\theta$ 。先拿上面的 $w_2$ 例子来看，我们期望最大化下面的似然函数：

$$\prod_{i=1}^3 P(n(w_i), i) = (1 - \frac{1}{1 + e^{-x_u^T \theta_1}})(1 - \frac{1}{1 + e^{-x_u^T \theta_2}}) \frac{1}{1 + e^{-x_u^T \theta_3}}$$

对于所有的训练样本，我们期望最大化所有样本的似然函数乘积。

为了便于我们后面一般化的描述，我们定义输入的词为 $w$ ,其从输入层词向量求和平均后的霍夫曼树根节点词向量为 $x_w$ ,从根节点到 $w$ 所在的叶子节点，包含的节点总数为 $l_w$ ,  $w$ 在霍夫曼树中从根节点开始，经过的第 $i$ 个节点表示为 $p_i^w$ ,对应的霍夫曼编码为 $d_i^w \in \{0, 1\}$ ,其中 $i = 2, 3, \dots, l_w$ 。而该节点对应的模型参数表示为 $\theta_i^w$ ，其中 $i = 1, 2, \dots, l_w - 1$ ，没有 $i = l_w$ 是因为模型参数仅仅针对于霍夫曼树的内部节点。

定义 $w$ 经过的霍夫曼树某一个节点 $j$ 的逻辑回归概率为 $P(d_j^w | x_w, \theta_{j-1}^w)$ ，其表达式为：

$$P(d_j^w | x_w, \theta_{j-1}^w) = \begin{cases} \sigma(x_u^T \theta_{j-1}^w) & d_j^w = 0 \\ 1 - \sigma(x_u^T \theta_{j-1}^w) & d_j^w = 1 \end{cases}$$

那么对于某一个目标输出词 $w$ ,其最大似然为：

$$\prod_{j=2}^{l_w} P(d_j^w | x_w, \theta_{j-1}^w) = \prod_{j=2}^{l_w} [\sigma(x_u^T \theta_{j-1}^w)]^{1-d_j^w} [1 - \sigma(x_u^T \theta_{j-1}^w)]^{d_j^w}$$

在word2vec中，由于使用的是随机梯度上升法，所以并没有把所有样本的似然乘起来得到真正的训练集最大似然，仅仅每次只用一个样本更新梯度，这样做的目的是减少梯度计算量。这样我们可以得到 $w$ 的对数似然函数 $L$ 如下：

$$L = \log \prod_{j=2}^{l_w} P(d_j^w | x_w, \theta_{j-1}^w) = \sum_{j=2}^{l_w} ((1 - d_j^w) \log[\sigma(x_u^T \theta_{j-1}^w)] + d_j^w \log[1 - \sigma(x_u^T \theta_{j-1}^w)])$$

要得到模型中 $w$ 词向量和内部节点的模型参数 $\theta$ ，我们使用梯度上升法即可。首先我们求模型参数 $\theta_{j-1}^w$ 的梯度：

$$\frac{\partial L}{\partial \theta_{j-1}^w} = (1 - d_j^w) \frac{(\sigma(x_u^T \theta_{j-1}^w)(1 - \sigma(x_u^T \theta_{j-1}^w))}{\sigma(x_u^T \theta_{j-1}^w)} x_w - d_j^w \frac{(\sigma(x_u^T \theta_{j-1}^w)(1 - \sigma(x_u^T \theta_{j-1}^w))}{1 - \sigma(x_u^T \theta_{j-1}^w)} x_w \tag{1}$$

$$= (1 - d_j^w)(1 - \sigma(x_u^T \theta_{j-1}^w))x_w - d_j^w \sigma(x_u^T \theta_{j-1}^w)x_w \tag{2}$$

$$= (1 - d_j^w - \sigma(x_u^T \theta_{j-1}^w))x_w \tag{3}$$

如果大家看过之前写的[逻辑回归原理小结](#)，会发现这里的梯度推导过程基本类似。

同样的方法，可以求出 $x_w$ 的梯度表达式如下：

$$\frac{\partial L}{\partial x_w} = \sum_{j=2}^{l_w} (1 - d_j^w - \sigma(x_u^T \theta_{j-1}^w)) \theta_{j-1}^w$$

有了梯度表达式，我们就可以用梯度上升法进行迭代来一步步的求解我们需要的所有的 $\theta_{j-1}^w$ 和 $x_w$ 。

## 3. 基于Hierarchical Softmax的CBOW模型

由于word2vec有两种模型：CBOW和Skip-Gram,我们先看看基于CBOW模型时， Hierarchical Softmax如何使用。

首先我们要定义词向量的维度大小 $M$ ，以及CBOW的上下文大小 $2c$ ,这样我们对于训练样本中的每一个词，其前面的 $c$ 个词和后面的 $c$ 个词作为了CBOW模型的输入,该词本身作为样本的输出，期望softmax概率最大。

2016年10月 (8)

### 常去的机器学习网站

52 NLP  
Analytics Vidhya  
机器学习库  
机器学习路线图  
强化学习入门书  
深度学习进阶书  
深度学习入门书

### 积分与排名

积分 - 432286  
排名 - 491

### 阅读排行榜

1. 梯度下降（Gradient Descent）小结(237769)
2. 梯度提升树(GBDT)原理小结(178441)
3. word2vec原理(一) CBOW与Skip-Gram模型基础(145153)
4. 线性判别分析LDA原理总结(122158)
5. 奇异值分解(SVD)原理与在降维中的应用(119131)

### 评论排行榜

1. 梯度提升树(GBDT)原理小结(376)
2. word2vec原理(二) 基于Hierarchical Softmax的模型(232)
3. 集成学习之Adaboost算法原理小结(207)
4. 决策树算法原理(下)(180)
5. 谱聚类（spectral clustering）原理总结(168)

### 推荐排行榜

1. 梯度下降（Gradient Descent）小结(80)
2. 奇异值分解(SVD)原理与在降维中的应用(66)
3. 谱聚类（spectral clustering）原理总结(33)
4. 集成学习原理小结(33)
5. 梯度提升树(GBDT)原理小结(32)

在做CBOW模型前，我们需要先将词汇表建立成一颗霍夫曼树。

对于从输入层到隐藏层（投影层），这一步比较简单，就是对 $w$ 周围的 $2c$ 个词向量求和取平均即可，即：

$$x_w = \frac{1}{2c} \sum_{i=1}^{2c} x_i$$

第二步，通过梯度上升法来更新我们的 $\theta_{j-1}^w$ 和 $x_w$ ，注意这里的 $x_w$ 是由 $2c$ 个词向量相加而成，我们做梯度更新完毕后会用梯度项直接更新原始的各个 $x_i (i = 1, 2, \dots, 2c)$ ，即：

$$\begin{aligned} \theta_{j-1}^w &= \theta_{j-1}^w + \eta(1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)) x_w \\ x_i &= x_i + \eta \sum_{j=2}^{l_w} (1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)) \theta_{j-1}^w \quad (i = 1, 2, \dots, 2c) \end{aligned}$$

其中 $\eta$ 为梯度上升法的步长。

这里总结下基于Hierarchical Softmax的CBOW模型算法流程，梯度迭代使用了随机梯度上升法：

输入：基于CBOW的语料训练样本，词向量的维度大小 $M$ ，CBOW的上下文大小 $2c$ ，步长 $\eta$

输出：霍夫曼树的内部节点模型参数 $\theta$ ，所有的词向量 $w$

1. 基于语料训练样本建立霍夫曼树。
2. 随机初始化所有的模型参数 $\theta$ ，所有的词向量 $w$
3. 进行梯度上升迭代过程，对于训练集中的每一个样本( $context(w), w$ )做如下处理：

a)  $e=0$ ，计算 $x_w = \frac{1}{2c} \sum_{i=1}^{2c} x_i$

b) for  $j = 2$  to  $l_w$ ，计算：

$$\begin{aligned} f &= \sigma(x_w^T \theta_{j-1}^w) \\ g &= (1 - d_j^w - f) \eta \\ e &= e + g \theta_{j-1}^w \\ \theta_{j-1}^w &= \theta_{j-1}^w + g x_w \end{aligned}$$

c) 对于 $context(w)$ 中的每一个词向量 $x_i$  (共 $2c$ 个) 进行更新：

$$x_i = x_i + e$$

d) 如果梯度收敛，则结束梯度迭代，否则回到步骤3继续迭代。

## 4. 基于Hierarchical Softmax的Skip-Gram模型

现在我们先看看基于Skip-Gram模型时，Hierarchical Softmax如何使用。此时输入的只有一个词 $w$ ，输出的为 $2c$ 个词向量 $context(w)$ 。

我们对于训练样本中的每一个词，该词本身作为样本的输入，其前面的 $c$ 个词和后面的 $c$ 个词作为Skip-Gram模型的输出，期望这些词的softmax概率比其他的词大。

Skip-Gram模型和CBOW模型其实是反过来的，在上一篇已经讲过。

在做CBOW模型前，我们需要先将词汇表建立成一颗霍夫曼树。

对于从输入层到隐藏层（投影层），这一步比CBOW简单，由于只有一个词，所以，即 $x_w$ 就是词 $w$ 对应的词向量。

第二步，通过梯度上升法来更新我们的 $\theta_{j-1}^w$ 和 $x_w$ ，注意这里的 $x_w$ 周围有 $2c$ 个词向量，此时如果我们期望 $P(x_i|x_w), i = 1, 2, \dots, 2c$ 最大。此时我们注意到由于上下文是相互的，在期望 $P(x_i|x_w), i = 1, 2, \dots, 2c$ 最大化的同时，反过来我们也期望 $P(x_w|x_i), i = 1, 2, \dots, 2c$ 最大。那么是使用 $P(x_i|x_w)$ 好还是 $P(x_w|x_i)$ 好呢，word2vec使用了后者，这样做的好处就是在一个迭代窗口内，我们不是只更新 $x_w$ 一个词，而是 $x_i, i = 1, 2, \dots, 2c$ 共 $2c$ 个词。这样整体的迭代会更加均衡。因为这个原因，Skip-Gram模型并没有和CBOW模型一样对输入进行迭代更新，而是对 $2c$ 个输出进行迭代更新。

这里总结下基于Hierarchical Softmax的Skip-Gram模型算法流程，梯度迭代使用了随机梯度上升法：

输入：基于Skip-Gram的语料训练样本，词向量的维度大小 $M$ ，Skip-Gram的上下文大小 $2c$ ，步长 $\eta$

输出：霍夫曼树的内部节点模型参数 $\theta$ ，所有的词向量 $w$

1. 基于语料训练样本建立霍夫曼树。
2. 随机初始化所有的模型参数 $\theta$ ，所有的词向量 $w$ ，
3. 进行梯度上升迭代过程，对于训练集中的每一个样本( $w, context(w)$ )做如下处理：

a) for  $i = 1$  to  $2c$ :

i)  $e=0$

ii)for  $j = 2$  to  $l_w$ , 计算:

$$f = \sigma(x_i^T \theta_{j-1}^w)$$

$$g = (1 - d_j^w - f)\eta$$

$$e = e + g\theta_{j-1}^w$$

$$\theta_{j-1}^w = \theta_{j-1}^w + gx_i$$

iii)

$$x_i = x_i + e$$

b)如果梯度收敛，则结束梯度迭代，算法结束，否则回到步骤a继续迭代。

## 5. Hierarchical Softmax的模型源码和算法的对应

这里给出上面算法和word2vec源码中的变量对应关系。

在源代码中，基于Hierarchical Softmax的CBOW模型算法在435-463行，基于Hierarchical Softmax的Skip-Gram的模型算法在495-519行。大家可以对着源代码再深入研究下算法。

在源代码中，neule对应我们上面的 $e$ ，syn0对应我们的 $x_w$ ，syn1对应我们的 $\theta_{j-1}^i$ ，layer1\_size对应词向量的维度，window对应我们的 $c$ 。

另外，vocab[word].code[d]指的是，当前单词word的，第d个编码，编码不含Root结点。vocab[word].point[d]指的是，当前单词word，第d个编码下，前置的结点。

以上就是基于Hierarchical Softmax的word2vec模型，下一篇我们讨论基于Negative Sampling的word2vec模型。

(欢迎转载，转载请注明出处。欢迎沟通交流：liujianping-ok@163.com)

分类: 0083. 自然语言处理

标签: 自然语言处理


好文要顶

关注我

收藏该文







刘建平Pinard

关注 - 15

粉丝 - 4329

±加关注

« 上一篇: word2vec原理(一) CBOW与Skip-Gram模型基础

» 下一篇: word2vec原理(三) 基于Negative Sampling的模型

22

0

posted @ 2017-07-27 17:26 刘建平Pinard 阅读(61888) 评论(232) 编辑 收藏

### 评论列表

#201楼[楼主] 2019-04-17 10:02 刘建平Pinard

@ Process520

你好！

“ns或softmax比hs准确”，这个结论是否成立，还是得看你的训练数据的分布情况。并不绝对。

1. 其实nst也是近似计算，因为使用了负采样。而hs如果一定也要说是近似计算，那么是因为它使用了梯度下降法，这是近似的求最优解的方法。
2. “hs当作softmax近似”，这句话不太理解。hs是使用霍夫曼树建立的softmax模型。但是近似的不是树或者softmax的原因，只是优化方法。
3. 损失函数，一般使用交叉熵损失函数即可

支持(0) 反对(0)

#202楼 2019-04-17 10:47 Process520

@ 刘建平Pinard

谢谢，明白了很多。第三个问题我没表述清楚，再请教下您。

问题：

fasttext作分类时，有一个函数参数：loss。官方提到有三种选择： {ns,hs,softmax} default softmax，我想问下这三个选择各自的适用范围。我目前的样本实例如下，结果表明准确率ns和softmax比hs要好，训练速度而言，hs>softmax>ns。

1. 家里的电表不用了，帮我注销掉\_\_label\_\_居民销户

2. 我厂里生产设备同时用就跳闸，容量不过够大吗？\_\_label\_\_高压扩容

3. 你好，我想咨询分时业务分了哪几段价格\_\_label\_\_居民分时电价

fasttext.classifier参数：

https://blog.csdn.net/qq\_32023541/article/details/80845913

支持(0) 反对(0)

#203楼[楼主 ] 2019-04-18 11:18 刘建平Pinard

@ Process520

你好！

官网最新文档的loss默认参数是ns

<https://fasttext.cc/docs/en/options.html>

我的个人经验是，数据量比较大的时候，一般使用ns，而中小数据量使用hs。softmax由于不是word2vec的默认方法，我一般不使用。

训练速度我不关心，因为不是瓶颈。

支持(0) 反对(0)

#204楼 2019-04-18 13:50 wangzaistone

刘老师您好，在Hierachical Softmax中，这个哈夫曼树的建立，是V中的每个词的位置是根据在语料中的频率来作为其权重建立的吗？

“第二，由于使用霍夫曼树是高频的词靠近树根，这样高频词需要更少的时间会被找到，这符合我们的贪心优化思想。”看到您这句话，但是还是不确定，遂觉得还是要请教确认下。

支持(0) 反对(0)

#205楼 2019-04-18 13:59 wangzaistone

@ 刘建平Pinard

引用

@hwcmh

你好！

这里是否是用词汇表中的所有词对应的词向量建立一个哈夫曼树？

【严格的说是用训练语料中的词和该词的上下文，训练出一颗哈夫曼树，同时可以得到训练样本中每个词的词向量。如果词汇表的某些词没有训练样本，那么就无法得到它的词向量】

刘老师，看到您之前的一个回复？训练出一颗哈夫曼树，我在最开始，这课树的建立（每个词放在这棵树的那些叶子节点位置）是根据词频来的吗？我训练的是得到关于这棵树上每个非叶子节点的向量theta?与更新后的词向量？

支持(0) 反对(0)

#206楼 2019-04-18 21:22 办公室李主任

博主，此处是否笔误？

“第二步，通过梯度上升法来更新我们的 $\theta_{wj-1}$ 和  $x_w$ ，注意这里的  $x_w$  是由2c个词向量相加而成，我们做梯度更新完后会用梯度项直接更新原始的各个  $x_i$  (i=1,2,,,,2c)，即...”后面的第二个式子应该是更新  $x_i$  而不是  $x_w$  ？

支持(0) 反对(0)

#207楼[楼主 ] 2019-04-19 09:37 刘建平Pinard

@ wangzaistone

你好！

1. 在Hierarchical Softmax中，这个哈夫曼树的建立，的确每个词的位置是根据在语料中的频率来作为其权重建立的。

2. 我训练的是得到关于这棵树上每个词对应的每个非叶子节点的向量theta，与更新后的每个词的词向量。这里要注意，非叶子节点的向量theta在每个位置对不同的词来说是不同的。

支持(0) 反对(0)

#208楼 2019-04-19 10:11 wangzaistone

@ 刘建平Pinard

谢谢您

支持(0) 反对(0)

#209楼[楼主 ] 2019-04-19 10:16 刘建平Pinard

@ 办公室李主任

你好，这里说的是常规方法应该是更新中心词的词向量和模型参数，但是后续我们发现更新窗口词更加合理均衡，因此更新的是窗口词的词向量。

<https://www.cnblogs.com/pinard/p/7243513.html>

5/8

支持(0) 反对(0)

#210楼 2019-04-19 13:21 Process520

@ 刘建平Pinard  
谢谢大佬。

支持(0) 反对(0)

#211楼 2019-04-19 22:31 办公室李主任

@ 刘建平Pinard  
emmm，更新窗口词就是更新这2c个词向量得意思咩，不理解得是这个式子

$$x_w = x_w + \eta \sum_{j=2}^{l_w} (1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)) \theta_{j-1}^w \quad (i = 1, 2 \dots, 2c)$$

的 $x_w = x_w + \dots$ 不应该是 $x_i = x_i + \dots$ 吗? 整个式子没有出现*i*后面却有“(i = 1, 2..., 2c)”, 强迫症表示很奇怪 :P  
烦请博主受累再次答疑解惑~

支持(0) 反对(0)

#212楼[楼主 ] 2019-04-20 20:20 刘建平Pinard

@ 办公室李主任  
你好，这里写错了，感谢指出错误，你的理解是对的。

支持(0) 反对(0)

#213楼 2019-04-26 10:19 cyZoe

有两个问题我想请问一下：  
1. CBOW和Skip-Gram更新的都是每个词的词向量xi吗？如果是这样的话，那您说的“Skip-Gram模型并没有和CBOW模型一样对输入进行迭代更新，而是对2c个输出进行迭代更新。”这句话怎么理解呢？  
  
2. CBOW中最后得到的w词向量具体是怎么得到的呢？是将更新后的xi像开始根节点的处理进行加和平均吗？

支持(0) 反对(0)

#214楼[楼主 ] 2019-04-26 10:32 刘建平Pinard

@ cyZoe  
你好！  
1. CBOW和Skip-Gram更新的都是每个词的词向量和词的模型参数，这个不错。后面这句话主要是在一个样本(2c个窗口词+1个中心词)训练时，CBOW输入的是2c个词向量，输出是1个中心词向量，**迭代更新的是输入的**2c个词向量。而Skip-Gram输入的是1个中心词向量，输出是2c个词向量，但是**迭代更新的是输出的**那2c个词向量。不是输入的那个中心词向量。  
  
2. 这里你说的是中心词吗？中心词在其他样本里是窗口词，在作为窗口词的样本中才能更新。迭代完毕后即为最终词向量。在作为中心词的阶段里，只会更新对应的模型参数，不更新它的词向量。

支持(0) 反对(0)

#215楼 2019-05-04 15:42 小于同学(\*^\_^\*)

@ 刘建平Pinard  
您好，Skip-Gram模型梯度上升迭代过程，f是不是应该和CBOW是一样的，我觉得不然梯度迭代过程和输入词向量没有关系呀

支持(0) 反对(0)

#216楼[楼主 ] 2019-05-04 18:43 刘建平Pinard

@ 小于同学(\*^\_^\*)  
你好，这里不一样。  
对于CBOW，f的计算使用输入词所有窗口词的平均词向量。而对于Skip-Gram，每次f的计算使用输入词各个窗口词的词向量，共2c个f。

支持(0) 反对(0)

#217楼 2019-05-04 19:46 小于同学(\*^\_^\*)

这2c循环中霍夫曼树中的路径都是从根节点到输入词向量到路径对吗？

支持(0) 反对(0)

#218楼[楼主 ] 2019-05-04 21:36 刘建平Pinard

@ 小于同学(\*^\_^\*)  
你好，循环的路径都是中心词的从根节点到叶子节点的路径。不能说是“输入词向量”在霍夫曼树中的路径。  
  
在这2c次循环中，每次循环输入的词向量是2c中的某一个。

支持(0) 反对(0)

#219楼 2019-05-06 10:39 千千世界

您好，这个哈夫曼树是虚拟建立的，中间的节点其实是不存在的，那怎么把这些节点和真是存在的context节点对应起来，为什么计算所得对这些节点的更新可以直接作用于2c个节点呢

支持(0)

反对(0)

#220楼

[楼主]

2019-05-06 10:50

刘建平Pinard

@ 千千世界  
你好，哈夫曼树在word2vec训练之前建立起来，你说的“虚拟建立”有点怪。中间节点也都会存在的。中间节点对每个经过词向量都有一个自己独有的模型参数。具体过程你可以再理解下算法流程

支持(0)

反对(0)

#221楼

2019-05-06 10:54

千千世界

好的，谢谢您~我应该没理解，我自己再理解一下

支持(0)

反对(0)

#222楼

2019-05-18 10:59

GodIsAWord

刘老师你好，计算量由V变成LOG2V，分别是怎么得到的呢？

支持(0)

反对(0)

#223楼

2019-05-18 16:12

GodIsAWord

刘老师你好，请问这个霍夫曼树是怎么构建的，是通过训练文本里面每个词出现的频率为权重构建嘛？

支持(0)

反对(0)

#224楼

[楼主]

2019-05-19 17:33

刘建平Pinard

@ GodIsAWord  
你好！  
1. 计算量由V变成LOG2V，这个是使用树结构二分查找的好处，其他的树搜索也都是这个改进。  
  
2. 霍夫曼树的确是根据训练文本里面每个词出现的频率为权重构建的，可以参考word2vec源码里构建霍夫曼树的源代码。

支持(0)

反对(0)

#225楼

2019-06-10 17:35

儒雅随和老实人

老师你好，我在别的博客看到，使用分层softmax训练，训练的是每个非叶子节点的参数，也就是你博客里的节点向量theta，非叶子节点个数为V-1，V为词汇表大小。而使用softmax的结构，最后会得到两个词向量矩阵，一个是输入层到隐藏层的矩阵W1，一个是隐层层到输出层的矩阵W2。我的问题是，分层softmax改变了隐藏层到输出层的结构，也就是W2没了，那么分层softmax的词向量如何表示呢，是直接使用输入层到隐层那个矩阵W1吗？还是说综合这个W1矩阵和哈弗曼树非叶子节点的theta，如果综合的话，哈弗曼树非叶子节点只有V-1个，即只有V-1个向量，如何综合呢？

支持(0)

反对(0)

#226楼

[楼主]

2019-06-11 10:17

刘建平Pinard

@ 儒雅随和老实人  
你好！  
word2vec其实和神经网络还有有些不同，里面并不能直接说“输入层到隐藏层的矩阵W1，隐层层到输出层的矩阵W2”，因为我们的HS的树结构不止2层，可以有好多层，而且每个词在树里面的深度不同，词向量参数的个数也不同。  
  
HS的词向量其实只是训练模型的副产品，它本身不是HS树结构中的模型参数。不好直接和神经网络对应上。  
  
个人建议你抛开神经网络的定式，单独理解word2vec这里的模型会更加准确。

支持(0)

反对(0)

#227楼

2019-06-18 17:43

hnaaa1

我觉得hierarchical softmax一章 第二节，梯度推导的时候，符号有些错误，Xw里面的w应该是输入的词向量，但是 $\theta_{w_j}$ 里面的w应该是指输出的词向量，因为要更新的权重只和输出向量挂钩，文中两个东西搞混了，输入输出都用w表示的。。。。。

支持(0)

反对(0)

#228楼

[楼主]

2019-06-19 09:53

刘建平Pinard

@ hnaaa1  
你好，的确词向量是输入的，而模型参数是中心词的，我这里 $x_w$ 代表输入词向量，而 $\theta_j^w$ 代表的是中心词的某个模型参数。  
  
都使用w是为了强调都是基于w为中心词的样本来的，符号有些让人迷惑，但是某种特定的符号没有混用。

支持(0)

反对(0)

#229楼

2019-06-26 12:28

千殇白芷

刘老师，您好：  
1、CBOW中，哈弗曼树的构建，其叶子节点权重 是指什么呢？是word在文档集中出现的频率吗？  
2、Skip-Gram中，如果概率采用 $p(xw \mid xi)$ 的形式，那context word的 word2vec更新量 怎么求呢？

感觉用 最大似然估计——梯度上升，只能求出 center word以及softmax中的weight的更新量，但是，context word更新量要怎么求呢？  
此外， $p(x_w | x_i)$ ，这种条件概率怎么求解呢？

支持(0)

反对(0)

#230楼[楼主]

2019-06-27 10:34

刘建平Pinard

@ 千殇白芷  
你好！  
1. 对的，是word在文档集中出现的频率，出现的频率越高，则会更靠近根节点，编码长度短。出现频率低，则会在比较深的树层次里，编码长度长。  
  
2. 上下文词的词向量更新的确就是梯度上升，参见第2节的原理和第4节的算法流程。这里更新的是上下文词的词向量，而不是中心词的词向量。可能我的符号用的让你觉得有点混淆。第2节我已经讲到了 $x_w$ 是根节点的输出词向量了。  

支持(0)

反对(0)

#231楼

2019-06-29 12:43

千殇白芷

刘老师，您好：  
还是有几个疑问：  
1、在Skip-Gram中，更新的是context word（也就是output word）的word2vec，那他们的更新量是用 由 center word 计算得到的  $x_w$  的 梯度更新量 来更新吗？  
  
2、在Skip-Gram中，目标优化是 $P(X_w|X_i)$ ，那这个概率值是怎么求得呢？因为，在Skip-Gram中， $X_i$ 为output word，可以通过softmax来求 $P(X_i)$ ，但是， $X_w$  相当于是 softmax 的输入，那 $P(X_w|X_i)$ 要怎么求解呢？感觉有点儿混乱。  

支持(0)

反对(0)

#232楼[楼主]

2019-06-30 16:18

刘建平Pinard

@ 千殇白芷  
你好！  
1. 这里窗口词的更新量的确是基于中心词计算得到的，也就是根据中心词的树参数选择是走左子树还是右子树得到的损失更新量。  
  
2) 在Skip-Gram中，树结构的输入并不是中心词的词向量，而是每个窗口词的词向量 $x_i$ ，参见第4节算法流程的3-a-ii的的表达式。  

支持(0)

反对(0)

刷新评论

刷新页面

返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万C++/C#源码：大型实时仿真组态图形源码
- 【前端】SpreadJS表格控件，可嵌入系统开发的在线Excel
- 【推荐】码云企业版，高效的企业级软件协作开发管理平台
- 【推荐】程序员问答平台，解决您开发中遇到的技术难题

相关博文：

- word2vec原理与代码
- word2vec生成词向量原理
- （六）语言模型 Language Madel 与 word2vec
- 自然语言处理--Word2vec （二）
- word2vec原理简述