刘建平Pinard

十年码农，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。

word2vec原理(三) 基于Negative Sampling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

在上一篇中我们讲到了基于Hierarchical Softmax的word2vec模型，本文我们我们再来看看另一种求解word2vec模型的方法：Negative Sampling。

1. Hierarchical Softmax的缺点与改进

在讲基于Negative Sampling的word2vec模型前，我们先看看Hierarchical Softmax的的缺点。的确，使用霍夫曼树来代替传统的神经网络，可以提高模型训练的效率。但是如果我们的训练样本里的中心词w是一个很生僻的词，那么就得在霍夫曼树中辛苦的向下走很久了。能不能不用搞这么复杂的一颗霍夫曼树，将模型变的更加简单呢？

Negative Sampling就是这么一种求解word2vec模型的方法，它摒弃了霍夫曼树，采用了Negative Sampling（负采样）的方法来求解，下面我们就来看看Negative Sampling的求解思路。

2. 基于Negative Sampling的模型概述

既然名字叫Negative Sampling（负采样），那么肯定使用了采样的方法。采样的方法有很多种，比如之前讲到的大名鼎鼎的MCMC。我们这里的Negative Sampling采样方法并没有MCMC那么复杂。

比如我们有一个训练样本，中心词是w,它周围上下文共有$2c$个词，记为$context(w)$。由于这个中心词w,的确和$context(w)$相关存在，因此它是一个真实的正例。通过Negative Sampling采样，我们得到neg个和w不同的中心词$w_i,i=1,2,..neg$，这样$context(w)$和w_i就组成了neg个并不真实存在的负例。利用这一个正例和neg个负例，我们进行二元逻辑回归，得到负采样对应每个词w_i对应的模型参数θ_i，和每个词的词向量。

从上面的描述可以看出，Negative Sampling由于没有采用霍夫曼树，每次只是通过采样neg个不同的中心词做负例，就可以训练模型，因此整个过程要比Hierarchical Softmax简单。

不过有两个问题还需要弄明白：1）如果通过一个正例和neg个负例进行二元逻辑回归呢？ 2） 如何进行负采样呢？

我们在第三节讨论问题1，在第四节讨论问题2.

3. 基于Negative Sampling的模型梯度计算

Negative Sampling也是采用了二元逻辑回归来求解模型参数，通过负采样，我们得到了neg个负例$(context(w),w_i)i=1,2,..neg$。为了统一描述，我们将正例定义为$w_0$。

在逻辑回归中，我们的正例应该期望满足：

$$P(context(w_0),w_i)=\sigma(x_{w0}^T\theta^{wi}),y_i=1,i=0$$

我们的负例期望满足：

$$P(context(w_0),w_i)=1-\sigma(x_{w0}^T\theta^{wi}),y_i=0,i=1,2,..neg$$

我们期望可以最大化下式：

$$\prod_{i=0}^{neg}P(context(w_0),w_i)=\sigma(x_{w0}^T\theta^{w0})\prod_{i=1}^{neg}(1-\sigma(x_{w0}^T\theta^{wi}))$$

利用逻辑回归和上一节的知识，我们容易写出此时模型的似然函数为：

$$\prod_{i=0}^{neg}\sigma(x_{w0}^T\theta^{wi})^{y_i}(1-\sigma(x_{w0}^T\theta^{wi}))^{1-y_i}$$

此时对应的对数似然函数为：

$$L=\sum_{i=0}^{neg}y_ilog(\sigma(x_{w0}^T\theta^{wi}))+(1-y_i)log(1-\sigma(x_{w0}^T\theta^{wi}))$$

和Hierarchical Softmax类似，我们采用随机梯度上升法，仅仅每次只用一个样本更新梯度，来进行迭代更新得到我们需要的$x_{w_i}, \theta^{w_i}, i=0,1,..neg$, 这里我们需要求出$x_{w_0}, \theta^{w_i}, i=0,1,..neg$的梯度。

首先我们计算θ^{w_i}的梯度:

$$\frac{\partial L}{\partial \theta^{w_i}} = y_i(1-\sigma(x_{w_0}^T\theta^{w_i}))x_{w_0} - (1-y_i)\sigma(x_{w_0}^T\theta^{w_i})x_{w_0} \quad (1)$$

$$= (y_i - \sigma(x_{w_0}^T\theta^{w_i}))x_{w_0} \quad (2)$$

同样的方法，我们可以求出x_{w_0}的梯度如下:

$$\frac{\partial L}{\partial x^{w_0}} = \sum_{i=0}^{neg}(y_i - \sigma(x_{w_0}^T\theta^{w_i}))\theta^{w_i}$$

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$x_{w_0}, \theta^{w_i}, i=0,1,..neg$。

4. Negative Sampling负采样方法

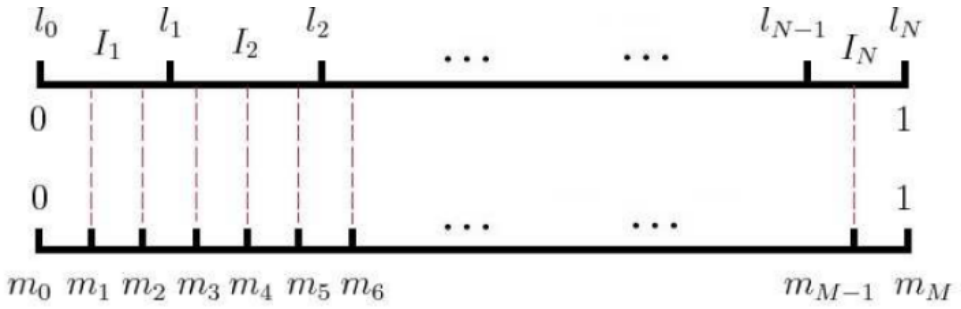
现在我们来看看如何进行负采样，得到neg个负例。word2vec采样的方法并不复杂，如果词汇表的大小为V,那么我们就将一段长度为1的线段分成V份，每份对应词汇表中的一个词。当然每个词对应的线段长度是不一样的，高频词对应的线段长，低频词对应的线段短。每个词w的线段长度由下式决定:

$$len(w) = \frac{count(w)}{\sum\limits_{u \in vocab} count(u)}$$

在word2vec中，分子和分母都取了3/4次幂如下:

$$len(w) = \frac{count(w)^{3/4}}{\sum\limits_{u \in vocab} count(u)^{3/4}}$$

在采样前，我们将这段长度为1的线段划分成M等份，这里$M >> V$，这样可以保证每个词对应的线段都会划分成对应的小块。而M份中的每一份都会落在某一个词对应的线段上。在采样的时候，我们只需要从M个位置中采样出neg个位置就行，此时采样到的每一个位置对应到的线段所属的词就是我们的负例词。



在word2vec中，M取值默认为10^8。

5. 基于Negative Sampling的CBOW模型

有了上面Negative Sampling负采样的方法和逻辑回归求解模型参数的方法，我们就可以总结出基于Negative Sampling的CBOW模型算法流程了。梯度迭代过程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$Mcount$，CBOW的上下文大小$2c$,步长η, 负采样的个数neg

输出：词汇表每个词对应的模型参数θ，所有的词向量x_w

1. 随机初始化所有的模型参数θ，所有的词向量w

2. 对于每个训练样本$(context(w_0), w_0)$,负采样出neg个负例中心词$w_i, i=1,2,...neg$

3. 进行梯度上升迭代过程，对于训练集中的每一个样本$(context(w_0), w_0, w_1, ...w_{neg})$做如下处理:

a) e=0， 计算$x_{w_0} = \frac{1}{2c}\sum\limits_{i=1}^{2c} x_i$

b) for i= 0 to neg, 计算:

$$f = \sigma(x_{w_0}^T\theta^{w_i})$$

$$g = (y_i - f)\eta$$

$$e = e + g\theta^{w_i}$$

$$\theta^{w_i} = \theta^{w_i} + gx_{w_0}$$

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

$$x_k = x_k + e$$

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。

6. 基于Negative Sampling的Skip-Gram模型

有了上一节CBOW的基础和上一篇基于Hierarchical Softmax的Skip-Gram模型基础，我们也可以总结出基于Negative Sampling的Skip-Gram模型算法流程了。梯度迭代过程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$Mcount$，Skip-Gram的上下文大小$2c$,步长η，，负采样的个数neg。

输出：词汇表每个词对应的模型参数θ，所有的词向量x_w

1. 随机初始化所有的模型参数θ，所有的词向量w

2. 对于每个训练样本$(context(w_0), w_0)$,负采样出neg个负例中心词$w_i, i = 1, 2, \ldots neg$

3. 进行梯度上升迭代过程，对于训练集中的每一个样本$(context(w_0), w_0, w_1, \ldots w_{neg})$做如下处理：

a) for i =1 to 2c:

i) e=0

ii) for j= 0 to neg, 计算：

$$f = \sigma(x_{w0}^T \theta^{wj})$$

$$g = (y_j - f)\eta$$

$$e = e + g\theta^{wj}$$

$$\theta^{wj} = \theta^{wj} + g x_{w0i}$$

iii) 词向量更新：

$$x_{w0i} = x_{w0i} + e$$

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。

7. Negative Sampling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Negative Sampling的CBOW模型算法在464-494行，基于Negative Sampling的Skip-Gram的模型算法在520-542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, syn1neg对应我们的θ^{wi}, layer1_size对应词向量的维度，window对应我们的c。negative对应我们的neg, table_size对应我们负采样中的划分数M。

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。这些和基于Hierarchical Softmax的是一样的。

以上就是基于Negative Sampling的word2vec模型，希望可以帮到大家，后面会讲解用gensim的python版word2vec来使用word2vec解决实际问题。

分类: 0083. 自然语言处理

标签： 自然语言处理

刘建平Pinard
关注 - 15
粉丝 - 4329
+加关注

11 0

« 上一篇：word2vec原理(二) 基于Hierarchical Softmax的模型
» 下一篇：用gensim学习word2vec

posted @ 2017-07-28 15:56 刘建平Pinard 阅读(43182) 评论(96) 编辑 收藏

评论列表

#51楼[楼主] 2018-12-03 15:14 刘建平Pinard

@ 努力wy努力
你好，还没有弄明白你的上下文，可以说的更详细些吗?

支持(0) 反对(0)

#52楼 2018-12-25 19:01 liangxinxin

老师您好，请教一下
Negative Sampling的Skip-Gram模型中
iii 对于context(w)中的每一个词向量xk(共2c个)进行更新:
xk=xk+e 不是 应该 xi = xi+e吗

支持(0) 反对(0)

#53楼[楼主] 2018-12-26 11:10 刘建平Pinard

@ liangxinxin
你好，的确前面已经有i了，没有必要再多一个k出来，感谢指出！原文已修改。

支持(0) 反对(0)

#54楼 2019-01-18 18:26 Austinak

老师您好:
Negative Sampling的Skip-Gram模型中
ii： f = 那里，X的下角标为何是woi，而不是wi,woi具体代表什么呢?

谢谢

支持(0) 反对(0)

#55楼 2019-01-18 18:27 Austinak

woi是否代表wo的上下文2c个词还是2c个词的词向量

支持(0) 反对(0)

#56楼[楼主] 2019-01-20 19:14 刘建平Pinard

@ Austinak
你好，注意这里除了中心词w_0以外有两组数据，第一组是neg个negative sample的数据，即$w_1, \ldots w_{neg}$第二组是2c个窗口词，即$w_{0i}(i = 1, 2, \ldots, 2c)$。
而作为x的角标，则表示式一个词向量，$x_{w0_i}^T$表示中心词的第i个窗口词的词向量。

支持(0) 反对(0)

#57楼 2019-01-25 12:09 linda_xu

博主您好，negative sampling并没有使用层次softmax，那么是不是和最原始的模型一样，隐藏层到输出层是全连接的，只是输出层个数不是字典V，而是正例个数+负例个数 = 1+neg ?

支持(0) 反对(0)

#58楼[楼主] 2019-01-27 21:56 刘建平Pinard

@ linda_xu
你好，还是应该来说还是有sigmoid的，也算是softmax的2分类的特例。
输出层的个数现在没有决策树结构，所以只是neg+1个"叶子",更新的模型参数每次也是neg+1个。

支持(0) 反对(0)

#59楼 2019-03-25 00:53 Bigheart

博主您好，能否讲下nceloss和negative sampling的关系呢?

支持(0) 反对(0)

#60楼[楼主] 2019-03-25 09:54 刘建平Pinard

@ Bigheart
你好，Negative Sampling是NCE的一种近似。negative sampling对于负例分布有强依赖，比如word2vec里的基于上下文的负采样，NCE则没有这个问题，可以从任意已知分布来采样噪音，当然有可能不准确。

这篇总结的不错，你可以看看:
https://blog.csdn.net/friyal/article/details/84875266

支持(0) 反对(0)

#61楼 2019-03-27 20:47 dobestself_994395

楼主您好，想请教一下无论是cbow还是skip-gram梯度上升求解中e是含义是什么，为什么要做累加然后更新相应的词向量

支持(0) 反对(0)

#62楼[楼主] 2019-03-28 10:19 刘建平Pinard

@ dobestself_994395
你好，这部分你要熟悉下梯度下降法，梯度上升法的算法原理。这里的e其实就是梯度值乘以步长，作为词向量在当前迭代要更新的变化值。上下文词向量用当前的词向量原值加上这个变化值，就算进行了一次梯度上升的迭代更新。

支持(0) 反对(0)

#63楼 2019-03-30 11:18 York_Chu

老师您好！请问cbow这个输出：词汇表每个词对应的模型参数θ，所有的词向量xw。
模型参数是指什么？是输入层与隐含层之间的权重，词向量应该是隐含层的权重。我怎么感觉这个模型参数与词向量是一个东西，不是很理解，求解答。

支持(0) 反对(0)

#64楼 2019-03-30 11:20 York_Chu

如果模型参数是是输入层与隐含层之间的权重，那么通过one-hot编码直接可以得到该词的词向量，我认为模型参数与词向量本质上是同一个东西。

支持(0) 反对(0)

#65楼[楼主] 2019-03-31 18:25 刘建平Pinard

@ York_Chu
你好，模型参数指的是每个词向量x_w,对应的参数θ^w，具体你可以看CBOW算法流程中中心词和neg个负采样词对应的参数θ如何使用。

而词向量是x_w,你也可以看看算法中词向量x_w是怎么使用了。

支持(0) 反对(0)

#66楼 2019-04-08 17:20 hhduola

刘老师您好，想问一下：
1、在CBOW中，正例是Xw0，即窗口词的词向量均值，负例是和中心词w不同的neg个词；在Skip-gram中，正例是每一个窗口词Xw0i，负例是和中心词w不同的neg个词，这样理解对吗？在计算量上，Skip-gram相当于是CBOW的2c倍?
2、如果是这样的话，对于CBOW和Skip-gram，在每一次迭代中并没有更新中心词w的词向量，而都是更新的窗口词的词向量，中心词的词向量也没有参与计算（在一次迭代中）？
谢谢~

支持(0) 反对(0)

#67楼[楼主] 2019-04-09 11:15 刘建平Pinard

@ hhduola
你好!
1. 理解是对的。Skip-Gram和CBOW相比，多了一层循环（2c次），所以的确是计算量大约为CBOW的2c倍。

2. 的确在每次迭代中，更新的都是窗口词的词向量。中心词的词向量没有参与更新。只是中心词的模型参数θ参与了更新。

支持(2) 反对(0)

#68楼 2019-04-09 11:35 hhduola

@ 刘建平Pinard
谢谢您的回复!
还想问一下，这样的话在Skip-gram中，为什么不直接用中心词w作为正例，而是要用窗口词Xw0i呢？用中心词w作为正例的话，计算量还会少很多（和CBOW一样了）

支持(0) 反对(0)

#69楼[楼主] 2019-04-09 12:07 刘建平Pinard

@ hhduola
你好，这里的原因和在第2篇里的Skip-gram中是一样的。你对比下可以发现都是使用的窗口词词向量参与迭代，只是迭代的模型参数是中心词的（如果是neg sample还有负采样词的模型参数）

这样做的原因我在第二篇第四节有讲到，这样我们每次迭代不是只更新中心词一个词，而是2c个词窗口词，这样整体的迭代会更加的均衡

支持(0) 反对(0)

#70楼 2019-04-11 15:53 simmonssong

刘老师您好，Negative Sampling的word2vec模型相比于Hierarchical Softmax的，是不是增加了负样本？在您的第二篇文章中，Hierarchical Softmax只用正例（句子内的单词）更新对应的权重向量，没有用到句子之外的其他单词。

支持(0) 反对(0)

#71楼[楼主] 2019-04-12 10:08 刘建平Pinard

@ simmonssong
你好，对的，增加了负样本。

Negative Sampling相比于Hierarchical Softmax，每个词的模型参数从l_w-1个减少到了一个，但是代价就是，需要若干负采样的词的 模型参数一起帮助来做迭代更新。

总的来说，计算量比起Hierarchical Softmax树结构变小了。

支持(1) 反对(0)

#72楼 2019-04-17 10:17 niar

老师您好!
我的问题是:替换中心词Xw0为负例Xwi(i>0)后，在算法中好像没有用到这些负例词本身?那样负例词本身什么用
Negative Sampling的CBOW模型算法流程中:仅仅对正例和负例对应的θ参数进行更新,最后对窗口的2c个单词进行更新,这样负例词本身就没有用到??
麻烦老师帮忙解惑

支持(0) 反对(0)

#73楼[楼主] 2019-04-18 09:49 刘建平Pinard

@ niar
你好!
对于负例词，我们使用了他们的模型参数$\theta^{w_j}, j=1,2,...neg$, 更新了它们的模型参数，但是并不更新他们的词向量。

如果你想说的是在迭代时负例词的词向量本身没有用到，这是对的。不过这些负例词在其他场景可能是窗口词，那么词向量还是可以的更新的。

支持(0) 反对(0)

#74楼 2019-04-18 11:25 wangzaistone

@ 刘建平Pinard

引用

@zhuyunxiu
你好！这里负采样的词的个数是neg，而不是2c，2c代表的是正例中心词对应的2c个真实窗口数据。
在CBOW中，neg个负采样的词不需要取平均值，直接和刚才2c个真实窗口数据平均得到的词一起，共neg+1个词，进行拟合优化，这些词中只有一个正例，其余全部都是负例。
在skip-gram里则稍微复杂一点，由于没有取平均，所以多了一层循环，也就是每个窗口词都去和neg个负例一起去拟合优化。

刘老师您好，“您这个解答回复中“neg个负采样的词不需要取平均值，直接和刚才2c个真实窗口数据平均得到的词一起，””，在CBOW中，可是既然context(w)是取平均值了，那为何又会对context(w)中的每一个词向量Xk（共2c）个进行更新？您原文是“ c) 对于context(w)中的每一个词向量xk(共2c个)进行更新：
xk=xk+e”

这个很懵了。

支持(0) 反对(0)

#75楼[楼主] 2019-04-18 12:03 刘建平Pinard

@ wangzaistone
你好!
关键点还是看算法流程比较准确。可能上面的描述让你不好理解。我重新梳理下整个ns的流程。

对于word2vec negative sample来说，关键点是一个中心词，2c个窗口词，neg个负采样的词。

在CBOW中，2c个窗口词平均得到根节点输入词向量，使用中心词和负采样词的模型参数来分别做梯度下降，得到梯度更新总量。最后用于更新2c个窗口词的词向量。

在Skip-gram中，2c个窗口词分别做为根节点输入词向量，因此Skip-gram的运算量比CBOW多2c倍，也就是2c次循环。在每个循环中，对于当前窗口词根节点，使用中心词和负采样词的模型参数来分别做梯度下降，得到梯度更新总量。最后用于更新当前个窗口词的词向量。

支持(1) 反对(0)

#76楼 2019-04-18 12:11 wangzaistone

@ 刘建平Pinard
非常感谢，这个概述的很清晰了！感谢!

支持(0) 反对(0)

#77楼 2019-04-18 21:18 niar

@ 刘建平Pinard

引用

@niar
你好!
对于负例词，我们使用了他们的模型参数θwj,j=1,2,...neg" role="presentation" style="position: relative;">θwj,j=1,2,...negθwj,j=1,2,...neg\theta^{w_j}, j=1,2,...neg, 更新了它们的模型参数，但是并不更新他们的词向量。

如果你想说的是在迭代时负例词的词向量本身没有用到，这是对的。不过这些负例词在其他场景可能是窗口词，那么词向量还是可以的更新的。

老师您好!
也就是说这些neg的参数是全局共享的吧,同一个词作为不同负例或作为正例用的同一个参数
明白了,,谢谢老师!!

支持(0) 反对(0)

#78楼 2019-04-20 20:38 办公室李主任

博主您好，想请教两个问题~
1. 为什么正例w_0对应的向量是2c个向量的平均向量，而不是直接用我们的中心词对应的词向量？中心词不才是正例吗?
2. 个人的理解是，Hierarchical Softmax和Negative Sampling其实都是word2vec的训练方法，word2vec的主要思想是在训练过程中不断更新词向量让词向量的形式更接近我们需要的形式。对吗？如果其重点还是这个的话，其实好像传统的DNN也是这种思想，那word2vec的贡献就是提出了两种更高效的训练方法改进了传统的DNN模型?

支持(0) 反对(0)

#79楼[楼主] 2019-04-20 20:51 刘建平Pinard

@ 办公室李主任
你好!
1. 看你说的正例w_0对应的向量是2c个向量的平均向量，应该你说的是CBOW。此时输入是2c个向量的平均向量，分别使用中心词正例和neg个负采样负例来迭代更新。

对于skip-gram，输入是2c个窗口词向量一个个的进来，然后分别使用中心词正例和neg个负采样负例来迭代更新。

2. word2vec的贡献主要是提出了高效得到低维词向量的简单方法。之前提出词向量后，维度太高了，实际中难以使用。维度降下来后，使用就很广泛了。

支持(0) 反对(0)

#80楼 2019-04-20 20:57 办公室李主任

@ 刘建平Pinard
博主你好！感谢光速回复 :P
第一个问题懂了。关于第二个问题，传统的DNN模型也可以定义低维的词向量呀，看您之前在第一篇的评论里面说过"最早的词向量神经网络是onehot的，后面慢慢也开始有低于onehot维度的词向量训练出现"，难道说这是在受了word2vec的启发之后对传统的DNN所作出的改进?

支持(0) 反对(0)

#81楼[楼主] 2019-04-22 09:55 刘建平Pinard

@ 办公室李主任
你好!
这里"传统"这个词我可能滥用了。你理解为最早出现的是onehot的即可。后续演化是不是受了word2vec的启发，其实我也没有研究。：）我们不纠结这个。

我们只能说现在word2vec做到的，使用深度学习DNN一样可以得到低维词向量。

支持(0) 反对(0)

#82楼 2019-04-26 11:17 办公室李主任

@ 刘建平Pinard
好的~谢谢博主^.^

支持(0) 反对(0)

#83楼 2019-04-27 10:43 wangzaistone

刘老师，CBOW采样neg个上下文词好理解，softmax层时间复杂度由logV降到了neg.但是为什么skip-gram采样的负例还是中心词呢？那不增大了计算量了吗？我采样neg个负例中心词，而中心词在Skipgram中是输入啊（原来只有一个），现在对于一个整理中心词，我还采样neg个吗?

今天再来读时，就想到了这个问题

支持(0) 反对(0)

#84楼[楼主] 2019-04-27 23:41 刘建平Pinard

@ wangzaistone
你好，skip-gram采样的负例，应该说还是中心词对应的负例。此时循环次数比CBOW多了2c倍。

对于一个正例中心词，也还是采样neg个负例，用于更新窗口词的词向量，以及负例的模型参数。

至于现在为什么skip-gram中心词是输入，还要采用neg的原因，其实和第二篇的Hierarchical Softmax里的原因是一样的。这样可以一次更新2c个词的词向量，而不是中心词一个词的词向量。

支持(0) 反对(0)

#85楼 2019-05-10 16:19 千千世界

您好，我想问下，怎么确定一个词的context呢，因为我想到一个语料库中如果这个词出现了好几次，那它在每句话中的context都不一样呀。还有就是，我看了源码，虽然模模糊糊，但感觉word2vec的待训练文件都是词库，而非语料库，

那词库中更加不知道每个词的context呀，烦请解答，谢谢您~

支持(0) 反对(0)

#86楼[楼主] 2019-05-12 17:37 刘建平Pinard

@ 千千世界
你好，词的context很简单，就是上下文。所以程序就可以很简单拿到。

比如假设我们定义的窗口词个数为3，则句子：我/爱/祖国/的/大好/河山

第一个词"我"的context就是3个词：爱/祖国/的
第4个词"的"的context就是5个词：我/爱/祖国/大好/河山

一个词出现了好几次，那它在每句话中的context都不一样：这是很正常的，这样可以拿到不同的context对应的训练样本。

支持(0) 反对(0)

#87楼 2019-05-20 19:58 乱花丛中独秀

您好，请教下：
根据词频将[0,1]线段带权分割之后，为什么非要再建立一个M个均分的结果来映射带权划分之后表示的词频线段来进行负样本的选择，为什么不直接创建随机数来对应负样本？建立M等分之后有什么好处？

支持(0) 反对(0)

#88楼[楼主] 2019-05-21 10:40 刘建平Pinard

@ 乱花丛中独秀
你好，当然可以创建随机数来做，这里的采样方法有很多。word2vec这个方法的实现也很简单。M等分的目的只在于在[1,M]这M个数字中随机采样。

支持(0) 反对(0)

#89楼 2019-06-08 22:58 szcCL2

老师您好，我的问题是，CBOW在计算的时候使用了2c个上下文词的平均向量作为输入，如果按照链式求导法则，误差对每个上下文词向量的梯度应该乘以1/2c（即w0对xk的梯度）才对啊，但是回传更新的时候却并没有乘以这个系数。而且研究gensim的源码发现，如果不使用2c个上下文词的平均向量而是简单的sum，回传的时候反而要乘以1/2c，这是为什么呢？gensim中这部分的代码如下：
if not model.cbow_mean and input_word_indices:
neu1e /= len(input_word_indices)
for i in input_word_indices:
context_vectors[i] += neu1e * context_locks[i]

支持(0) 反对(0)

#90楼[楼主] 2019-06-10 16:08 刘建平Pinard

@ szcCL2
你好，CBOW的计算可以使用2c个上下文词的平均向量作为输入，如我上文所示，也可以直接使用2c个上下文词的向量和作为输入。这两种选择都是可以的。

至于回传的时候反而要乘以1/2c，这里你可以把它理解为梯度下降(上升)法更新时的步长变化了，乘以1/2c，那么梯度迭代更新的步长就变小了，迭代收敛会变慢，但是结果可能会更加准确。理论上你可以 不乘1/2c，结果也不会有很大的影响。

支持(0) 反对(0)

#91楼 2019-06-13 19:52 szcCL2

@ 刘建平Pinard
是的，理论上确实是一个步长的影响，但是按照梯度下降法的链式求导法则来是没错的吧。
当使用2c个向量平均的时候乘以1/2c，而使用2c个向量的和的时候求导就没有1/2c这个系数，这符合求导公式应该没问题的吧。
我的问题不是理论上乘不乘是不是等效，而是为什么gensim的代码里面刚刚好是相反的。。？

支持(0) 反对(0)

#92楼[楼主] 2019-06-14 10:40 刘建平Pinard

@ szcCL2
你好，因为之前我是基于google的word2vec的代码来写的文章
https://github.com/tmikolov/word2vec/blob/master/word2vec.c

没有看过gensim的word2vec代码，有时间我看下，不过个人认为这里只是工程上的差异，且没有太多的影响。

支持(0) 反对(0)

#93楼 2019-06-17 20:31 木头人开开

博主你好，请问在skip-gram中的模型参数的数量（在迭代中貌似每个词对应的参数不一致）,和词汇表的数量是一致的，那这个是不是比hierarchical方法的中的参数多了，因为树结构可能会有共享的节点参数。

支持(0) 反对(0)

#94楼[楼主] 2019-06-18 10:37 刘建平Pinard

@ 木头人开开
你好，negative sample的参数数量一般要比hierarchical softmax要少。

对于negative sample来说，你使用了中心词，neg个负采样词的模型参数，一次样本迭代对应neg+1个模型参数。

对于hierarchical softmax来说，不同的样本迭代时模型参数数量不同，与对应的中心词在树结构的位置有关，如果 比较深，则模型参数多，否则参数少。

这样平均下来的话，每次迭代时一般negative sample的参数会少一些，当然不绝对。

站在整体来说，negative sample总参数数量和词汇表的数量是一致的，而hierarchical softmax则是词汇表每个词的树参数个数之和。肯定超过negative sample的参数数量了。

支持(0) 反对(0)

#95楼 2019-06-26 21:54 默文

刘老师，您好!
在第7个小节里:
在源代码中，基于Negative Sampling的CBOW模型算法在464-494行，基于Hierarchical Softmax的Skip-Gram的模型算法在520-542行。大家可以对着源代码再深入研究下算法。

后面的基于的原理写错了，应该都是Negative Sampling

支持(0) 反对(0)

#96楼[楼主] 2019-06-27 10:36 刘建平Pinard

@ 默文
你好，的确是写错了，非常 感谢指出错误，原文已改正。

支持(0) 反对(0)

< Prev 1 2

刷新评论 刷新页面 返回顶部

相关博文:
· word2vec改进之NegativeSampling
· word2vec原理(二)基于HierarchicalSoftmax的模型
· 基于word2vec训练词向量(二)
· DL4NLP——词表示模型（三）word2vec（CBOW/Skip-gram）的加速：Hierarchical Softmax与Negative Sampling
· [DeeplearningAI笔记]序列模型2.7负采样Negativesampling