

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

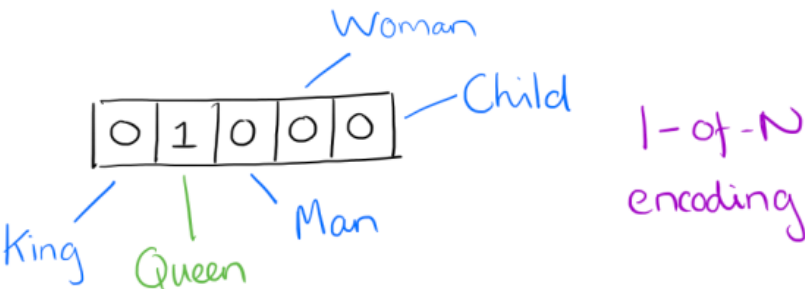
word2vec原理(一) CBOW与Skip-Gram模型基础

- 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

word2vec是google在2013年推出的一个NLP工具，它的特点是将所有的词向量化，这样词与词之间就可以量化的去度量他们之间的关系，挖掘词之间的联系。虽然源码是开源的，但是谷歌的代码库国内无法访问，因此本文的讲解word2vec原理以Github上的word2vec代码为准。本文关注于word2vec的基础知识。

1. 词向量基础

用词向量来表示词并不是word2vec的首创，在很久之前就出现了。最早的词向量是很冗长的，它使用是词向量维度大小为整个词汇表的大小，对于每个具体的词汇表中的词，将对应的位置置为1。比如有下面的5个词组成的词汇表，词"Queen"的序号为2，那么它的词向量就是(0, 1, 0, 0, 0)。同样的道理，词"Woman"的词向量就是(0, 0, 0, 1, 0)。这种词向量的编码方式我们一般叫做1-of-N representation或者one hot representation。



One hot representation用来表示词向量非常简单，但是却有很多问题。最大的问题是我们的词汇表一般都非常大，比如达到百万级别，这样每个词都用百万维的向量来表示简直是内存的灾难。这样的向量其实除了一个位置是1，其余的位置全部都是0，表达的效率不高，能不能把词向量的维度变小呢？

Distributed representation可以解决One hot representation的问题，它的思路是通过训练，将每个词都映射到一个较短的词向量上来。所有的这些词向量就构成了向量空间，进而可以用普通的统计学的方法来研究词与词之间的关系。这个较短的词向量维度是多大呢？这个一般需要我们在训练时自己来指定。

比如下图我们将词汇表里的词用"Royalty", "Masculinity", "Femininity"和"Age" 4个维度来表示，King这个词对应的词向量可能是(0.99, 0.99, 0.05, 0.7)。当然在实际情况中，我们并不能对词向量的每个维度做一个很好的解释。

	King	Queen	Woman	Princess	...
Royalty	0.99	0.99	0.02	0.98	
Masculinity	0.99	0.05	0.01	0.02	
Femininity	0.05	0.93	0.99	0.94	
Age	0.7	0.6	0.5	0.1	
...	...				

有了用Distributed Representation表示的较短的词向量，我们就可以较容易的分析词之间的关系了，比如我们将词的维度降低到2维，有一个有趣的研究表明，用下图的词向量表示我们的词时，我们可以发现：

$$\vec{King} - \vec{Man} + \vec{Woman} = \vec{Queen}$$

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：



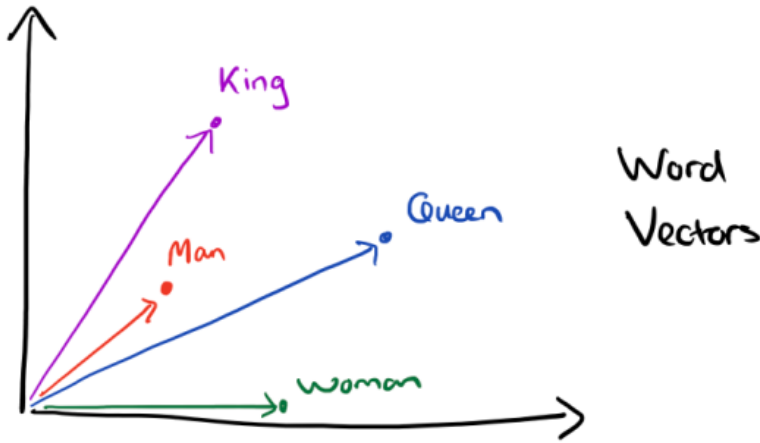
昵称：刘建平Pinard
园龄：2年8个月
粉丝：4328
关注：15
+加关注

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)



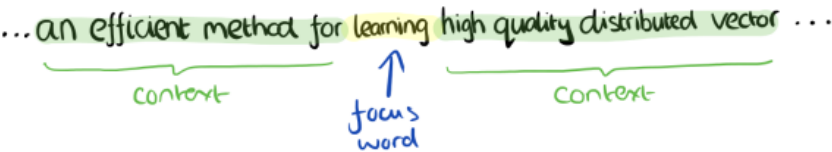
可见我们只要得到了词汇表里所有词对应的词向量，那么我们就可以做很多有趣的事情了。不过，怎么训练得到合适的词向量呢？一个很常见的方法是使用神经网络语言模型。

2. CBOW与Skip-Gram用于神经网络语言模型

在word2vec出现之前，已经有神经网络DNN来用训练词向量进而处理词与词之间的关系了。采用的方法一般是一个三层的神经网络结构（当然也可以多层），分为输入层，隐藏层和输出层(softmax层)。

这个模型是如何定义数据的输入和输出呢？一般分为CBOW(Continuous Bag-of-Words 与Skip-Gram两种模型。

CBOW模型的训练输入是某一个特征词的上下文相关的词对应的词向量，而输出就是这特定的一个词的词向量。比如下面这段话，我们的上下文大小取值为4，特定的这个词是"Learning"，也就是我们需要的输出词向量,上下文对应的词有8个，前后各4个，这8个词是我们模型的输入。由于CBOW使用的是词袋模型，因此这8个词都是平等的，也就是不考虑他们和我们关注的词之间的距离大小，只要在我们上下文之内即可。



这样我们这个CBOW的例子，我们的输入是8个词向量，输出是所有词的softmax概率（训练的目标是期望训练样本特定词对应的softmax概率最大），对应的CBOW神经网络模型输入层有8个神经元，输出层有词汇表大小个神经元。隐藏层的神经元个数我们可以自己指定。通过DNN的反向传播算法，我们可以求出DNN模型的参数，同时得到所有的词对应的词向量。这样当我们有新的需求，要求出某8个词对应的最可能的输出中心词时，我们可以通过一次DNN前向传播算法并通过softmax激活函数找到概率最大的词对应的神经元即可。

Skip-Gram模型和CBOW的思路是反着来的，即输入是特定的一个词的词向量，而输出是特定词对应的上下文词向量。还是上面的例子，我们的上下文大小取值为4，特定的这个词"Learning"是我们的输入，而这8个上下文词是我们的输出。

这样我们这个Skip-Gram的例子，我们的输入是特定词，输出是softmax概率排前8的8个词，对应的Skip-Gram神经网络模型输入层有1个神经元，输出层有词汇表大小个神经元。隐藏层的神经元个数我们可以自己指定。通过DNN的反向传播算法，我们可以求出DNN模型的参数，同时得到所有的词对应的词向量。这样当我们有新的需求，要求出某1个词对应的最可能的8个上下文词时，我们可以通过一次DNN前向传播算法得到概率大小排前8的softmax概率对应的神经元所对应的词即可。

以上就是神经网络语言模型中如何用CBOW与Skip-Gram来训练模型与得到词向量的大概过程。但是这和word2vec中用CBOW与Skip-Gram来训练模型与得到词向量的过程有很多的不同。

word2vec为什么不用现成的DNN模型，要继续优化出新方法呢？最主要的问题是DNN模型的这个处理过程非常耗时。我们的词汇表一般在百万级别以上，这意味着我们DNN的输出层需要进行softmax计算各个词的输出概率的的计算量很大。有没有简化一点点的方法呢？

3. word2vec基础之霍夫曼树

word2vec也使用了CBOW与Skip-Gram来训练模型与得到词向量，但是并没有使用传统的DNN模型。最先优化使用的数据结构是用霍夫曼树来代替隐藏层和输出层的神经元，霍夫曼树的叶子节点起到输出层神经元的作用，叶子节点的个数即为词汇表的大小。而内部节点则起到隐藏层神经元的作用。

具体如何用霍夫曼树来进行CBOW和Skip-Gram的训练我们在下一节讲，这里我们先复习下霍夫曼树。

霍夫曼树的建立其实并不难，过程如下：

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)

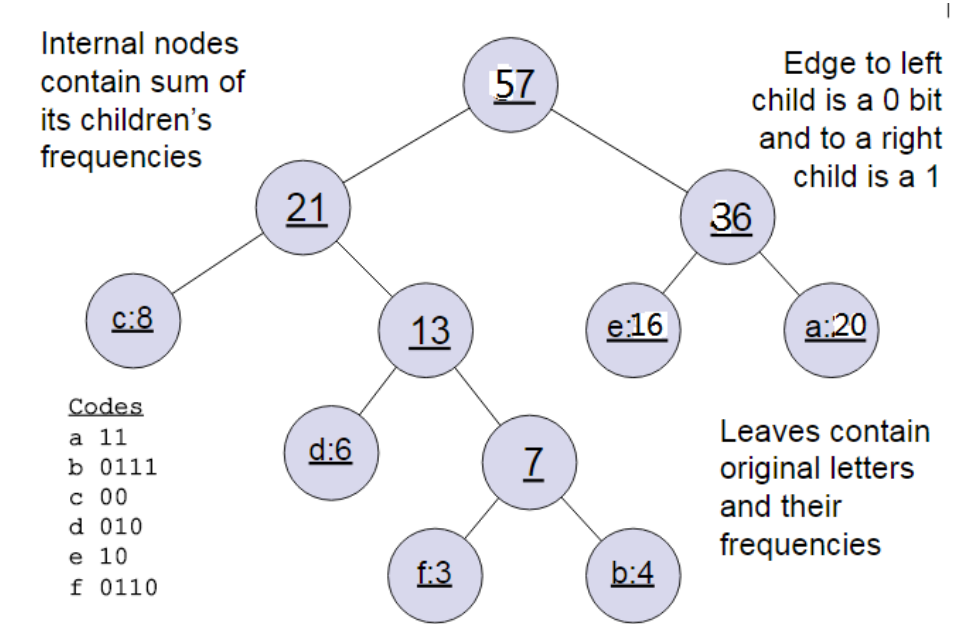
输入：权值为 (w_1, w_2, \dots, w_n) 的 n 个节点

输出：对应的霍夫曼树

- 1) 将 (w_1, w_2, \dots, w_n) 看做是有 n 棵树的森林，每个树仅有一个节点。
- 2) 在森林中选择根节点权值最小的两棵树进行合并，得到一个新的树，这两颗树分布作为新树的左右子树。新树的根节点权重为左右子树的根节点权重之和。
- 3) 将之前的根节点权值最小的两棵树从森林删除，并把新树加入森林。
- 4) 重复步骤2) 和3) 直到森林里只有一棵树为止。

下面我们用一个具体的例子来说明霍夫曼树建立的过程，我们有(a,b,c,d,e,f)共6个节点，节点的权值分布是(20,4,8,6,16,3)。

首先是最小的b和f合并，得到的新树根节点权重是7.此时森林里5棵树，根节点权重分别是20,8,6,16,7。此时根节点权重最小的6,7合并，得到新子树，依次类推，最终得到下面的霍夫曼树。



那么霍夫曼树有什么好处呢？一般得到霍夫曼树后我们会对叶子节点进行霍夫曼编码，由于权重高的叶子节点越靠近根节点，而权重低的叶子节点会远离根节点，这样我们的高权重节点编码值较短，而低权重值编码值较长。这保证的树的带权路径最短，也符合我们的信息论，即我们希望越常用的词拥有更短的编码。如何编码呢？一般对于一个霍夫曼树的节点（根节点除外），可以约定左子树编码为0，右子树编码为1.如上图，则可以得到c的编码是00。

在word2vec中，约定编码方式和上面的例子相反，即约定左子树编码为1，右子树编码为0，同时约定左子树的权重不小于于右子树的权重。

我们在下一节的Hierarchical Softmax中再继续讲使用霍夫曼树和DNN语言模型相比的好处以及如何训练CBOW&Skip-Gram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

19

0

« 上一篇：条件随机场CRF(三). 模型学习与维特比算法解码

» 下一篇：word2vec原理(二). 基于Hierarchical Softmax的模型

posted @ 2017-07-13 16:34 刘建平Pinard 阅读(145156) 评论(69) 编辑 收藏

评论列表

#51楼[楼主]2019-03-31 18:21 刘建平Pinard	@ cly24 你好！ 1. 传统网络里，onehot的词向量维度都是语料库大小V。这里并不需要训练词向量。因为可以根据每个词在语料库的索引号得到为1的位置，其余V-1个位置为0。训练只是为了得到的可以正确预测各个词输出概率的网络参数。 2. 这里如果你说的还是传统网络，输出的是V个概率值，即V个词出现的概率值，V个神经元。 3. 对的，这种情况不针对onehot，只对应小于V的N维词向量的情况。	支持(0) 反对(0)
#52楼2019-03-31 23:52 南宫城	@ 刘建平Pinard 好的，现在我没问题了。感谢楼主的细心解答，我继续看您的其它博客去了	支持(0) 反对(0)
#53楼2019-04-13 14:11 zzzzzzzcheng	楼主，你好！skip gram这个模型的意思是通过训练来把一个词转化为n个上下文词汇吗？那他是通过什么步骤实现word embedding的呢	支持(0) 反对(0)
#54楼2019-04-14 15:45 帆小徐	您好： 请问Skip-Gram会考虑context距离中心词的距离作为权重吗？	支持(0) 反对(0)
#55楼[楼主]2019-04-14 17:44 刘建平Pinard	@ 帆小徐 你好，理论上是可以这么做。不过常见的word2vec，比如我上文里的算法都没有讨论距离的权重问题。也就是说，当做词袋模型了，不考虑顺序。	支持(0) 反对(0)
#56楼2019-04-19 06:53 wangzaistone	刘老师您好，如何理解CBOW与word2vec中CBOW的关系？“ 以上就是神经网络语言模型中如何用CBOW与Skip-Gram来训练模型与得到词向量的大概过程。但是这和word2vec中用CBOW与Skip-Gram来训练模型与得到词向量的过程有很多的不同。” word2vec本身是一种浅层神经网络语言模型还是只能说它是一种工具？在word2vec没有出现前，是不是就已经存在CBOW于Skip-gram了？后面word2vec中无论是基于层次softmax还是NG的CBOW与skip-gram，他们的输出层，都是词汇表大小个概率？还是只是logV 与neg+1？ 还是有点迷糊，还恳请老师能指点一下？	支持(0) 反对(0)
#57楼[楼主]2019-04-19 10:28 刘建平Pinard	@ wangzaistone 你好，CBOW和Skip-gram的思想很早就有了。word2vec出来后，低维度的词向量成了主流。传统的词向量还是onehot的方式。 个人认为word2vec虽然也有神经网络的思想，但是更像是一个工具。有了这个工具，可以独立所有词的距离，相似度。而传统的神经网络还是局限于onehot高维词向量，求解预测上下文词或者中心词的出现概率。 现在最新的深度学习的方法都是基于词嵌入的，这和word2vec的思想类似。但是不用单独训练词向量了，词向量嵌入矩阵可以在训练模型的时候一起得到。	支持(0) 反对(0)
#58楼2019-04-19 10:58 wangzaistone	@ 刘建平Pinard 嗯嗯，清晰了，谢谢您	支持(0) 反对(0)
#59楼2019-05-05 16:16 千千世界	您好，我对输出和输出层感到有点乱。skip-gram中的输入是中心词词向量，输出为该中心词的context的词向量(有多个)，但输出层是词汇表大小个softmax概率，那最终得到的中心词的词向量是词汇量维度的，还是context维的，使用得到的概率表示吗	支持(0) 反对(0)
#60楼[楼主]2019-05-06 10:14 刘建平Pinard	@ 千千世界 你好，假设你说的应该是最传统的神经网络模型，那里输出的维度是词汇表大小，而不是窗口词的个数大小。每个维度是输出的softmax概率，如果是skip-gram则是选择概率最大的若干词序号作为预测的窗口词。	支持(0) 反对(0)
#61楼2019-05-06 10:19 千千世界		

谢谢~您是说skip-gram最后将一个词表示的向量形式是这样的吗：
维度：窗口大小
向量元素：概率排在前边的概率

支持(0) 反对(0)

#62楼[楼主] 2019-05-06 10:23 刘建平Pinard

@ 千千世界
这倒没有，输出所有的元素里面并没有词向量，只有所有词(词汇表打标)的softmax概率值。输入的才是词向量，但是维度是自己定义的。

这个最传统的模型并没有将词向量的输入和输出完全统一，不像word2vec。

支持(0) 反对(0)

#63楼 2019-05-06 10:37 千千世界

啊skip-gram不是word2vec吗，我可能没表述清楚，我是说skip-gram模型最终表示一个词向量的维度是自己定的，还是2c维，还是词汇量维~又麻烦您解答

支持(0) 反对(0)

#64楼[楼主] 2019-05-06 10:47 刘建平Pinard

@ 千千世界
哈哈，我上面“假设你说的应该是最传统的神经网络模型”。所以误解了。

其实word2vec的skip-gram模型最终表示的词向量的维度是自己定的，有默认值，可以调参。与窗口的维度2c，词汇表大小无关。

支持(0) 反对(0)

#65楼 2019-05-06 23:53 千千世界

好的 谢谢您，一路打扰过来的~

支持(0) 反对(0)

#66楼 2019-05-18 12:09 郑瀚Andrew.Hann

good post

支持(0) 反对(0)

#67楼 2019-06-09 11:06 code-life

我觉得评论里边，80%的人都是对基于传统的神经网络模型的CBOW和Skip-Gram的输入输出，与您说的Distributed representation的对应关系的疑问，
1、您可以在博文里补充上这个地方更细致的讲解
2、另外，看了一圈您的回复，还是没看懂啥意思。以cbow为例，比如输入是上下文共四个单词，定义词向量的维度为5，那么输入层的神经元个数是20吗？另外您说输出是词汇表的长度，就是说输出层是词汇表长度个神经元吗？那么这样的话，怎么对应到输出单词的5个维度的词向量来做损失呢？

支持(0) 反对(0)

#68楼[楼主] 2019-06-10 16:24 刘建平Pinard

@ code-life
你好，其实传统的神经网络模型的CBOW和Skip-Gram这里并不是重点。所以写的会简单一些，毕竟这种建模方法已经少用了。

你举的例子，如果按传统的CBOW来看，词汇表大小为V，上下文为4，则输入的维度为4V，如果你词汇表大小就是5，那么就是20个输入神经元。输出层则是V个神经元，每个神经元代表一个softmax概率。即你说的词汇表长度个神经元。

对应输出单词的5个维度的词向量来做损失，其实就是对于你输入的4V个神经元，输出是V个神经元，期望最小的损失是，只有那个期望的CBOW输出的位置的概率值尽可能靠近1，其余位置的概率值尽可能靠近0。

支持(0) 反对(0)

#69楼 2019-06-10 19:15 code-life

@ 刘建平Pinard
好的谢谢，我目前的理解是传统方法输入输出其实都是 one hot类型的向量，然后我们最终想获得的词向量（自定义长度）其实是输入层到隐藏层的权重矩阵。同时也是基于这样处理的计算量太大，所以才有了后面的层次softmax以及负采样方法训练出来的CBOW以及Skip-gram

支持(0) 反对(0)

【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的数学原理（一）——词向量基础及huffman树
- word2vec原理分析
- Word2Vec实现原理(HierarchicalSoftmax)
- Word2vec之CBOW
- 自然语言处理--Word2vec（二）

Copyright ©2019 刘建平Pinard