自然语言处理一直是个人比较感兴趣的一块内容，近期又抽空学习了下word2vec的内容，就简单将一些前人研究和自身想法整理成文字。

随着深度学习的发展，对于图像、音频、文字等的处理方式也更加多样化，但其本质仍旧是将非结构化数据转换为结构化数据，重点依旧是数据的表示。如何将这些非结构化的数据，通过一定的形式转换为计算机可以识别的向量型数据，同时又尽可能多的保留非结构化数据所要表达的信息。

1. 词向量的表示

单词/文字的结构化表示需要考虑的因素相对来说还是比较多的

①每个单词都包含不同的意思；

②不同单词之间的组合所表达的含义也不一样

③同样一个单词放在不同的语句之中，结果又是不一样的

这几点可以概括为词序信息和语义信息，之后涉及到的模型都是为了解决如果有效的用尽可能少的维度去表示更多的信息，同时还能尽量减少计算复杂度（模型的参数）

1. One-Hot Representation

最原始的单词表示方法为0-1表示，有多少个不同的单词就相应的会有多少维的数据；该方法结果下，单词矩阵非常稀松，而且单词与单词之间默认不存在相关性，同时也没有考虑单词在不同位置出现时的不同情况

1. Distributed Representation

这种方法是通过一定的计算将单词转化为一个低维度的向量（远远小于语料库单词N个数），而每个单词之间的相关性则由向量之间的欧式距离决定；该方法是根据现有语料库里的单词间结构，训练得到的（无监督和有监督），因此在一定程度上受语料库的影响（除非语料库足够大，否则训练出来的单词间相关性只适用于该语料库）

对于这种单词分布表示方法，大致可以归为基于矩阵的表示、基于聚类的表示以及基于神经网络的表示（word2vec此）三种，且这三种表示方法都依赖于分布假说——单词语义由上下文决定。

1. 分布表示
2. 基于矩阵的表示（词的计数）

该方法是将词与其上下文之间建立一个矩阵关系；行表示单词，列表示单词对应的上下文（单词前后N个词作为一个上下文，即n\_gram）出现的次数，当然更加复杂的可以是单词对应的文档出现的次数。通常情况下矩阵内的值为单词和上下文同时出现的频次/频率（tf-idf属于词和词的矩阵），用以表明单词与上下文之间的关联性；构建完矩阵后，通过对矩阵的SVD、PCA等形式的分解，就可以得到单词的低维向量表示，此类表示的经典方法包括LSA、GloVe。（词-词矩阵构建了词的替换关系，词-文档矩阵构建了词的组合关系）

分解的矩阵N\*M中，N\*n1表示的是每个单词对应的向量表示，**那n1\*M中每一列（M为单词对应的上下文类别）则表示什么意思？是指上下文对应的向量表示吗？如果是这样，那么这样是不是可以计算出每个上下文与每个单词之间的相关性？**

该方法下，n\_gram取值不能太大，否则对应的总数将指数级增加；n=1为词-词，n>1为词-上下文，n>n1为词-文档；随着n的增加，矩阵的稀疏性增加。

1. 基于聚类的表示

通过聚类的方法构建单词与其上下文之间的关系，经典方法为布朗聚类。

*P*(*wijwi􀀀*1) = *P*(*wijci*)*P*(*cijci􀀀*1)

通过该词对应的类别和上下文对应的类别，求出相应的概率值，**问题在于初始词的类别如何进行划分？**目前了解的一种方法是单词对应的词性作为对应的类别，其余的暂时不太了解，且该方法应该是属于一种无监督方法。

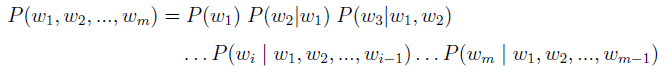
1. 基于神经网络（词的预测）

基于矩阵的表示，很大程度上只能表示词与词之间的关系，当在表达词与上下文之间的联系时，其上下文不能够有太长的跨度（上下2、3个词），而神经网络则相对来说能更灵活的表达词与上下文之间的关系。

神经网络的核心在于将上下文和目标词之间建立联系，即y=f(x)，其中x为上下文的向量表示。基于神经网络的表示，可以理解为一种有监督的学习。

1. 语言模型

语言模型本质是通过给定的一些单词（文档），判断对应的单词（文档）出现的概率，模型优势在于考虑了各个单词出现的次序与单词间的相关性



正常情况下，一些单词或文档出现的概率公式如上图所示，但是随着m值得增加，在一个较大的语料库中进行计算时，其计算量和计算难度将大幅度提升（组合个数增加）

1、n-gram模型

对于比较长的语句，n-gram模型只考虑对应单词前的n个上下文，超过n的上下文将被忽略



当n=1时，单词不受其上下文的影响，只考虑自身的出现频率；而当n>1时，单词出现的概率除了受本身的影响外，还受其上下文n个单词的影响（后续的循环神经网络思路在此基础上衍生）；随着n得增加，词序信息得损失也将会越小，但同时n个单词的组合可能性也呈指数级上升（尤其是语料库较大时）；另一方面，n得增加使得符合条件的组合出现的频次更低，数据越稀疏，最终导致结果稳定性较差。

对于那些n个单词组合在语料库中没有出现的情况，通常可以考虑两种方式解决：

①平滑法（基本上是分子分母都加一个数）

②回退法（利用n-1的元组的概率去代替n元组的概率）

解释说明：4个词的语料库，n=4个时，对应的计算量为4！+3！+2！+1！；语料库越大计算量越大

n-pos模型整体逻辑与n-gram类似，只是不用上下文的词，而是用上下文的词对应的词性或类别；即将所有的词分为K类，然后上下文出现的对应类别出现的频次计算该词出现的概率

**该模型下，貌似只能计算出对应单词或句子出现的概率，对于如何将单词表示成低维向量是无法实现的。**

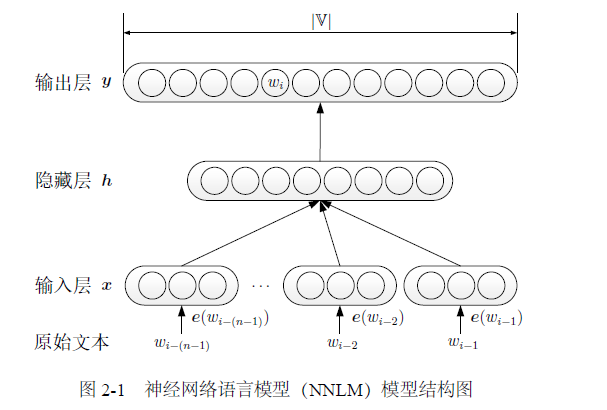
2、神经网络语言模型（NNLM）

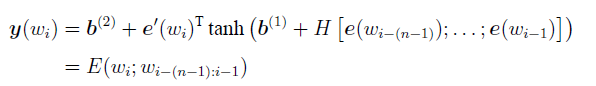
采用三层神经网络，其中输入层为单词前n个上下文的向量表示，初始为one-hot编码，输入的矩阵维度为n\*N；通过权重矩阵后输出到隐藏层，权重矩阵为N\*T（T为隐藏层神经元个数，即单词表示的低维向量维度）；隐藏层到输出层矩阵为T\*N，最终输出的结果为N维的向量（表示语料库中N个单词出现的概率）

N\*T矩阵中表示每个单词对应的T维向量，表示单词作为上下文时的向量表示；T\*N则表示单词作为目标值得向量表示，**到底用哪个来对单词进行降维？**

按照上述逻辑最终计算出来的矩阵应该是n\*N维，在输入到隐藏层之间应该是进行处理过，比如word2vec采用的是直接求向量均值，那么**传统的神经网络算法采用的是什么方法？——（采用词向量的拼接，将输入的词向量拼接成1\*(N\*n)的向量）**

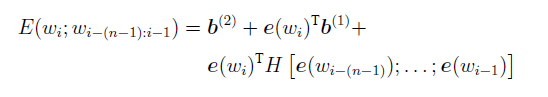
NNLM的大部分计算集中在隐藏层和输出层之间的矩阵向量运算，以及输出层上的softmax归一化计算。





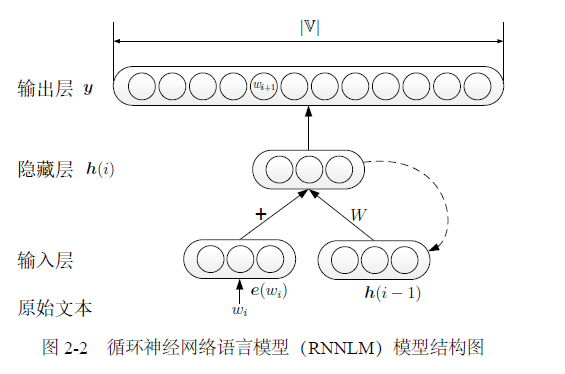
3、log双线性语言模型（LBL）

在神经网络语言模型中，涉及到的是非线性形式（tanh），且会涉及到两个词向量表示（上下文和目标值），但在log双线性语言模型中则词向量只涉及到一个向量表示矩阵，能量函数应该属于一个线性的函数？当然，这类语言模型后续还有很多优化后的变种模型，



1. 循环神经网络语言模型（RNNLM）

循环即重复的利用之前的数据输出结果，与当前的数据相结合来预测未来的一个结果。在该结果下，神经网络的输入数据为当前单词向量与前一单词的隐藏层输出，该模型的优势在于能够利用所有的过去的词信息（上下文信息）来预测下一个单词。



**该方法的不足点在哪里？如果能够用到所有的过去词向量信息，那该方法是不是属于最优的？最终的输出层貌似仍旧是N维的词向量（语料库里单词个数）**

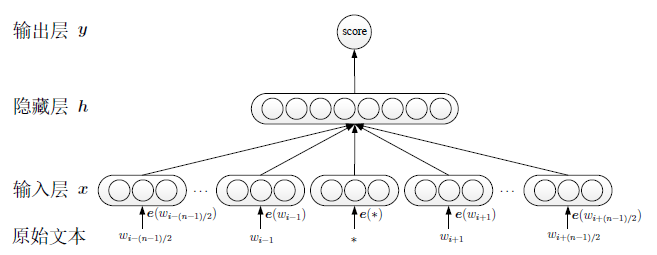
循环神经网络的结构属于一种有偏的，句子后面输入的信息所占影响要高于句子前面输入的，但实际情况中一个句子的重点并不一定在后半部分。

1. C&W模型

之前的几种神经网络语言模型，在最终的输出层都需要输出V维的数据，而C&W模型则通过一定的方式将输出结果减少至1维，其输出结果反映的是对n元短语的打分，且模型是以生成词向量为目标的。同时该模型还有一个最大的区别在于输入层中包含当前单词向量（即目标词），**对于此方法并没有完全理解，需要重新找资料进行学习，而且该方法貌似与负采样技术有相似点？——（目标词和上下文联合打分）**

**C&W模型与神经网络模型类似，差异点在于①C&W模型的激活函数为HardTanh函数；②输出结果只有1个值（1维）。模型的目标函数中，x为正常的输入语句，x(w)则为将中间目标词替换为其他词之后的非正常语句，根据目标函数最大化估计出对应的参数值，进而求得对应的词向量。其目标函数的含义可表示为正确词序列相对于不正确词序列高出的分数。**

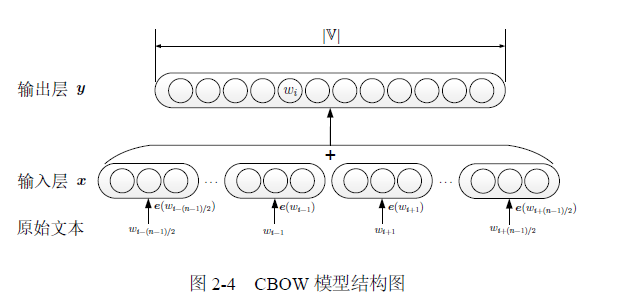




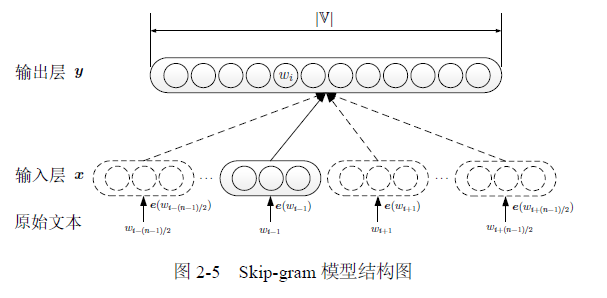
1. CBOW模型&Skip-gram模型

这两个模型是在之前的几个模型基础上进行的优化，个人理解CBOW是将NNLM和C&W进行了融合。CBOW采用的是将目标词上下n/2个单词输入模型中，但是去掉了原先的隐藏层，而是直接将输入向量求均值；该部分其实与log双线性模型近似，属于线性部分。CBOW对原先的语言模型进行了简化（1、去掉隐藏层；2、输入词向量直接求线性均值），大幅度提升了训练速度；

CBOW直接用上下文词向量的均值来预测目标词向量，默认上下文各词向量具有相同的权重（与目标词向量关联性一致），**能不能在权重这块上进行优化，上下文每个词向量与目标词之间肯定存在不同的相关性**。



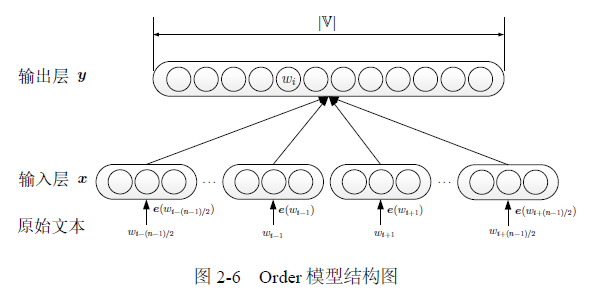
Skip-gram模型与CBOW模型近似，区别在于CBOW输入的是目标词的上下n/2个词向量，而Skip-gram模型则输入的是1个词向量（**有些文章说输入的是目标词，但有些文章说输入的是目标词上下文中的一个词，具体仍需验证**）



7、Order模型

CBOW模型将输入词向量求均值，在一定程度上忽略了词序信息（**在权重上的优化能不能解决这个问题点**）；Skip-gram每次只输入一个词向量，对于词序信息也没有考虑；

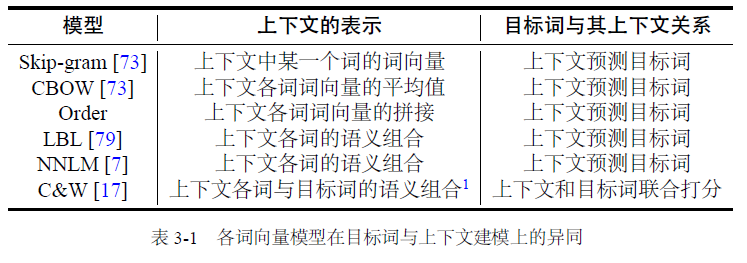
相比之下，Order模型则是在考虑词序信息得基础上，将原先神经网络的隐藏层去掉；但是如果这样的话，那么**最终输出的应该是属于线性结构，即上下文与目标词之间是线性关系，但实际情况并不一定如此，如何解决目标词与上下文之间的非线性关系？——（与非线性方法相比，线性关系在一定程度上会损失部分的语义信息）**



8、小结

各种神经网络词向量模型中，除了Skip-gram 模型使用词作为上下文表示之外，其它模型均使用n-gram 作为上下文表示，而这些表示使用不同的组合策略构造n-gram 的表示。如CBOW 模型使用n-gram 中各词词向量的平均值作为上下文表示；Order 模型使用n-gram 中各词词向量的拼接作为上下文表示，这种方法可以**看做词向量的线性组合**；LBL 模型则是直接对n-gram 中各词的词向量做了线性变换；NNLM 和C&W 模型更是做了非线性变换。

CBOW 模型与Skip-gram 模型相比，采取了更复杂的上下文表示方法 ，Order 相比CBOW 模型在上下文表示时，保留了词序信息。LBL 在保留词序信息的同时，还进一步使用线性变换，使模型具有上下文的语义组合能力。NNLM与C&W 模型进一步采用了非线性激活函数，使得整个模型为神经网络结构，表达能力强于LBL 的双线形结构。



1. Word2Vec

终于讲到word2vec模型了，该模型是在CBOW和Skip-gram两种语言模型的基础上，进一步优化了输出层的计算。其对输出层的优化又可以分为Hierarchical Softmax和Negative Sampling两种框架，故最终能形成4种模型。

1. Hierarchical Softmax

在讲该框架前，首先需要解释下Huffman树，该结构是Hierarchical Softmax方法的基础结构，Huffman树在一定程度上能代替隐藏层和输出层。

1. Huffman树

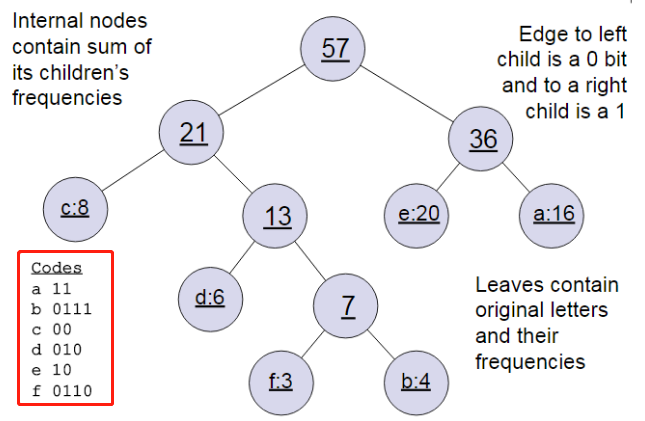
Huffman树其实属于一种非线性结构（二叉树），**因此能够表示上下文与目标词之间的非线性关系（？？？）**

1）将(w1,w2,...wn)看做是有n棵树的森林，每个树仅有一个节点。

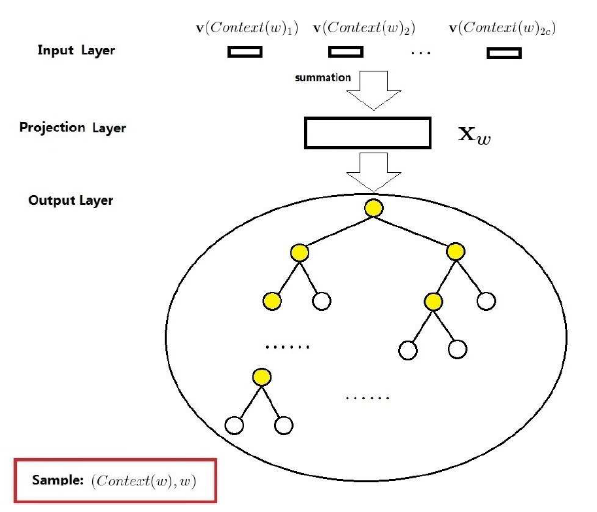
2）在森林中选择根节点权值最小的两棵树进行合并，得到一个新的树，这两颗树分布作为新树的左右子树。新树的根节点权重为左右子树的根节点权重之和。

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。



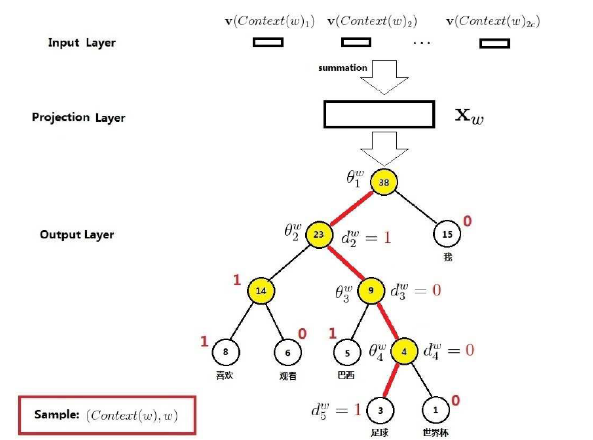
Huffman树的优势在于①对于高频词汇，通常会更加靠近根节点，在进行搜索时更容易搜索到，但假如文章中出现的词都比较生僻，则该方法反而会加大计算量；②语料库中每一个单词都可以被编码成一串01数据（每个根节点左右分别代表0或1），那么对于目标词的预测其实就是一连串的二分类模型组合。



1. 模型逻辑

模型的第一步就是根据语料库构建Huffman树对语料库中的词进行编码（0-1），之后其实就是一系列的逻辑回归分类问题。而在迭代优化权重和参数时，其目标函数则为相应的各分类节点上的概率值乘积（CBOW和Skip-gram整体逻辑类似）。

根据目标函数，结合梯度下降法求得对应的权重向量及参数，但问题是求得的权重向量表示的是输入参数的权重向量累加值（Skip-gram貌似直接就是输入词的向量），需要将此根据输入各词的贡献（向量权重）分配到各词中去。



1. Negative Sampling
2. 模型逻辑

目标词以外的所有词都属于负样本（目标词为正样本），那么该模型就转化为一个二分类问题。从语料库中抽取若干个目标词以外的词，与目标词一起构建一个二分类样本集；输入为目标词的上下文，输出则为N维词向量（N-1个负样本词和目标词），模型优化的目标函数则是正样本和所有负样本的概率乘积。

1. 负采样算法

其本质就是一个带权采样问题，根据单词在语料库中出现的频率确定权重，采用等距、等比方式抽取出负样本

二次采样技术中，将高频词剔除采样样本中（高频词通常都是停用词，包含的语义信息量较少）

1. 博士论文的启示

《基于神经网络的词和文档语义向量表示方法研究》该文章对现有语言模型的适用条件进行了系统的梳理，相当全面，绝对是大神之作

1. **回答了下述几个问题：**

①使用哪个模型效果更好？具体而言，在多种不同的上下文表示之中，以及在上下文与目标词的两种不同的关系之间，应当如何选择合适的模型？

②训练语料的大小及领域对词向量有什么样的影响？

③在迭代训练中，选择什么样的迭代次数可以获得足够好的词向量，同时避免过拟合？

1. 多少维的词向量效果最理想？

**2、评价方法：**

①利用词向量的语言学特性完成任务；（语义相关性、同义词检测、单词类比）

②将词向量作为特征，提高自然语言处理任务的性能；（基于平均词向量的文本分类、命名实体识别）

③将词向量作为神经网络的初始值，提升神经网络模型的优化效果。（基于卷积神经网络的文本分类、词性标注）

为了解决不同衡量指标之间的差异性（指标与指标之间、指标本身在不同方法下），作者采用了性能增益率来进行对比（与随机生成的词向量对应的结果进行对比，相应的语言模型所带来的提升比例）

**3、得到的结论**

**问题一：**

①对于小语料，像Skip-gram 这样的简单的模型会取得更好的效果。对于更大的语料，CBOW 和Order 这样对上下文有更复杂建模的模型，会有更好的效果。

②对于实际的自然语言处理任务（比如将词向量用作现有任务的特征，或者用于神经网络模型的初始值），使用Skip-gram、CBOW 和Order 这样的简单模型就已经足够好。③对于评价语言学特性的任务（比如词汇相似度），通过上下文预测目标词的模型，比上下文与目标词联合打分的C&W 模型效果更好。其中比较特别的是，C&W 模型得到的词向量完全不包含线性平移关系

**问题二：**

1. 使用领域内的语料，对同领域的任务有明显的帮助。领域内的语料可以让词向量拥有领域内的语义，对同领域任务的促进是最明显的。而且，使用纯领域内语料比混合领域外的大规模语料的效果更好。
2. 如果选择了不合适的语料，很可能没有办法从语料中得到与任务相匹配的语义或者用法，会对相应的任务起到负面效果。如使用网络评论语料训练的词向量，用于词性标注任务时，性能下降。
3. 对于同领域的语料，语料规模越大，词向量性能越好。

**问题三：**

在大多数情况下，可以选取一个简单任务的性能峰值作为训练词向量的迭代终止条件。在条件允许的情况下，选择目标任务的验证集性能作为参考标准，是最合适的选择。

**问题四：**

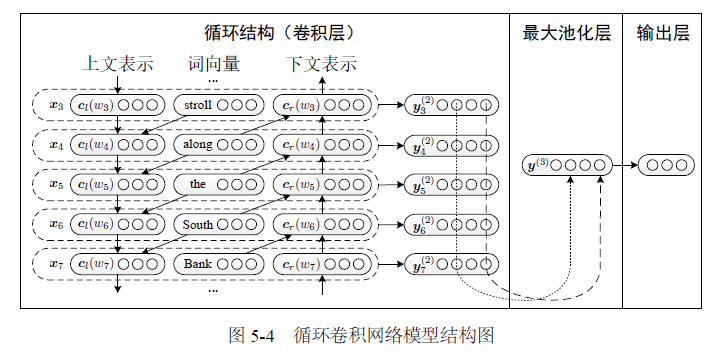
对于分析词向量语言学特性的任务，维度越大效果越好（除C&W 模型以外，在3.3.2 节中已有解释）；对于提升自然语言处理任务而言，50 维词向量通常就足够好。

上述提到的语言模型中，其前提假设都是分布假说，即单词语义由其上下文决定；但另外一种可能就是单词语义只受单词本身以及组成单词的字所决定（比较适用于中文）

句子的语义由一系列词的语义所组合而成，常用的语义组合方式包括线性加权、矩阵乘法、张量乘法、：递归神经网络、循环神经网络和卷积神经网络

**4、文档表示方法**

对循环网络而言，其缺点在于其语义偏向于靠后的词向量；对于卷积网络而言，其窗口需要人为的进行设定调整，无法判断多少窗口合理；因此作者将循环和卷积网络结合，试图解决两者各自的问题。（**这样结合确实能解决循环网络的问题，但也能解决卷积网络的问题吗？**）



深度学习word2vec笔记基础篇

<https://blog.csdn.net/han____shuai/article/details/50882135>

word2vec 中的数学原理详解

<https://blog.csdn.net/itplus/article/details/37969519>

基于神经网络的词和文档语义向量表示方法研究【博士论文】