**文本分类技术研究报告（上下文语义）**

**1、文本分类、情感分类的概念**

**文本分类**是在预定义的分类体系下，根据文本的特征，将给定文本与一个或多个类别相关联的过程。

传统的文本分类是基于文本内容的，研究如何将文本自动划分成政治的 、经济的、军事的、体育的、娱乐的等各种类型。这是人们提到文本分类这一术语时通常所指的含义。

**情感分析**是对带有情感色彩的主观性文本进行分析、处理、归纳和推理。情感分析包含较多任务，如情感分类、观点抽取、观点问答和观点摘要等。

**情感分类**是指文本所表达的含义和情感信息将文本划分成褒扬的或贬义的两种或几种类型。是对文本作者倾向性和观点、态度的划分，因此有时也称为倾向性分析 (以上定义来自宗成庆的《统计自然语言处理》一书)。

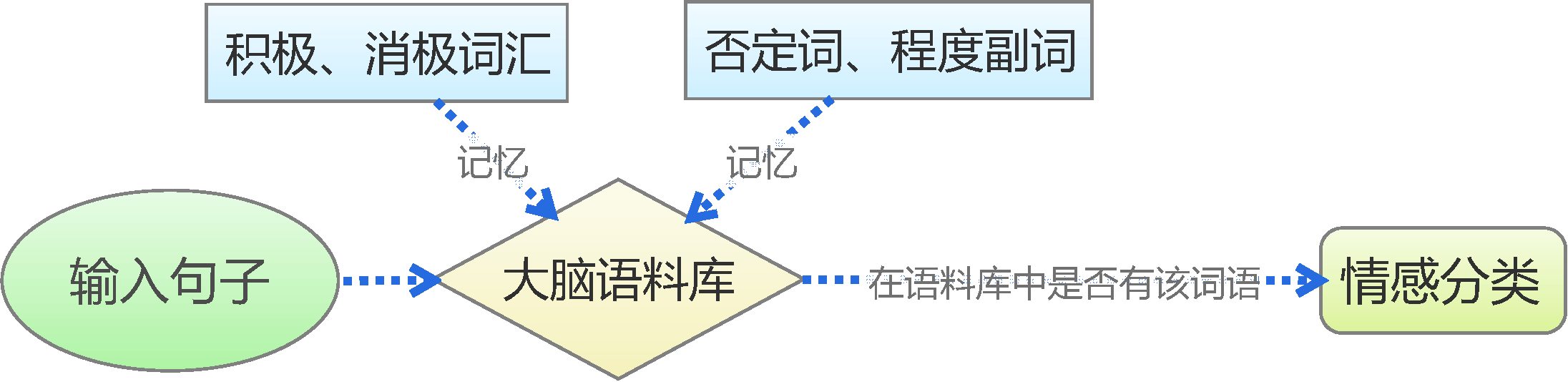
文本情感分类问题是一类特殊的文本分类任务，它和传统的主题分类的区别是，主题分类中，主题词是最重要的特征，而情感分类中，情感词是最重要的特征。

**2、文本情感分类算法**

一般有两种方式实现文本情感分类：一是传统的基于情感词典的文本分类，二是使用基于机器学习的情感分析模型。

**2.1 基于情感词典的文本情感分类**

传统的基于情感词典的文本情感分类，是对人的记忆和判断思维的最简单的模拟。我们首先构建**情感词典**——积极情感词典、消极情感词典、否定词典以及程度副词词典。然后对输入的句子进行分词处理，判断分词后的词语是否出现在对应情感词典中，比如“我喜欢数学”，“喜欢”这个词在我们所记忆的积极词汇表中，所以我们判断它具有积极的情感。



**图1 基于情感词典的文本分类流程**

基于情感词典的文本情感分类规则比较机械化：将每个积极情感词语赋予权重1，将每个消极情感词语赋予权重-1，如果分词后的词语向量包含相应的词语，就加上向前的权值，其中，否定词和程度副词会有特殊的判别规则，否定词会导致权值反号，而程度副词则让权值加倍。最后，根据总权值的正负性来判断句子的情感。

**存在的问题：**

①词典的完备性问题，覆盖范围有限。

②基于情感字典的方法往往采用无监督的分类模型，在缺少大量人工标注的 训练文本的情况下，分类的精确率比起基于监督的分类模型要差。

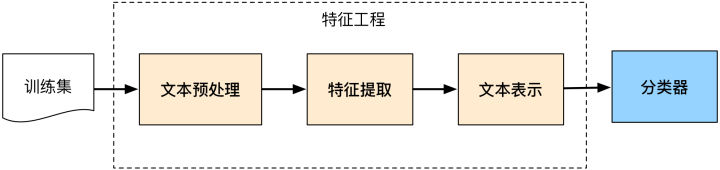
③情感词典的构建，对专业背景知识依赖较大，不仅需要是数据挖掘专家，还需要语言学家。

**2.2 基于机器学习的文本情感分类**

伴随着统计学习方法的发展，特别是90年代后互联网在线文本数量增长和机器学习学科的兴起，逐渐形成了一套解决大规模文本分类问题的经典方法，这个阶段的主要工作是特征工程+分类模型：

**文本特征工程**分为文本预处理、特征提取、文本表示三个部分，最终目的是把文本转换成计算机可理解的格式，并封装足够用于分类的信息，即很强的特征表达能力。

**分类器**基本都是统计分类方法，基本上大部分机器学习方法都在文本分类领域有所应用，比如朴素贝叶斯分类算法、KNN、SVM和神经网络等等。



**图2 基于情感词典的文本分类流程**

基于传统机器学习的文本情感分类的主要问题是，文本表示是高纬度高稀疏的，特征表达能力很弱，并且需要人工进行特征工程，成本很高。

**2.3 基于深度网络的情感分析**

应用深度学习解决大规模文本分类问题最重要的是解决文本表示，再利用CNN/RNN等网络结构自动获取特征表达能力，去掉繁杂的人工特征工程，端到端的解决问题。

对分词后的数据，第一步是使用word2vec技术将词语转换为词向量（见**第3章节word2vec）。**

句子是词向量的集合，也就是矩阵，类似于图像处理，图像数字化后也对应一个像素矩阵，因为模型的输入一般只接受一维的特征，所以需要将矩阵进行转化。一个比较简单的想法是将矩阵展平，也就是将词向量一个接一个，组成一个更长的向量。但是这样会使得输入维度高达几千维甚至几万维，事实上是难以实现的。

对于**图像**处理来说，已经有一套成熟的方法了，叫做**卷积神经网络**（CNN），它是神经网络的一种，专门用来处理矩阵输入的任务，能够将矩阵形式的输入编码为较低维度的一维向量，而保留大多数有用信息。卷积神经网络那一套也可以直接搬到自然语言处理中，尤其是文本情感分类中，效果也不错。但是句子的原理不同于图像，直接将图像那一套用于语言，虽然略有小成，但总让人感觉不伦不类。因此，这并非自然语言处理中的主流方法。

在**自然语言处理**中，通常用到的方法是**循环神经网络**（recurrent neural network RNN）。它们的作用跟卷积神经网络是一样的，将矩阵形式的输入编码为较低维度的一维向量，而保留大多数有用信息。跟卷积神经网络的区别在于，卷积神经网络更注重全局的模糊感知（好比我们看一幅照片，事实上并没有看清楚某个像素，而只是整体地把握图片内容），而RNN则是注重邻近位置的重构，由此可见，对于语言任务，语言总是由相邻的字构成词，相邻的词构成短语，相邻的短语构成句子，因此需要有效地把邻近位置的信息进行有效的整合。（RNN具体技术实现见**第4章节RNN循环神经网络）**

**示例**

**语料1**：我有抑郁症，所以就去死一死，没什么重要的原因，大家不必在意我的离开

**语料2**：[抑郁症的懒和健康人的懒有啥区别](https://weibo.com/ttarticle/p/show?id=2309404368650185379544)

**3、Word2vec**

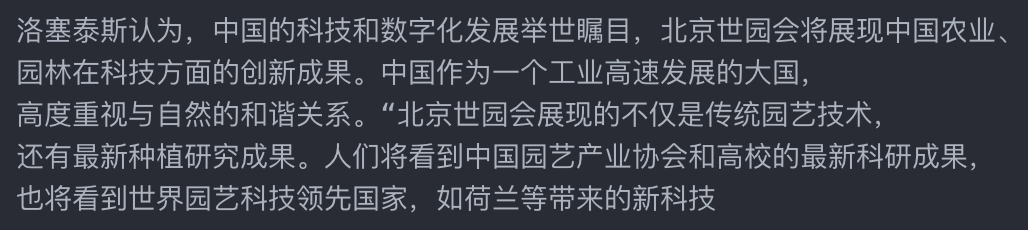
要将自然语言交给机器学习中的算法来处理，首先需要将语言数学化，可以利用各种词表示模型，将“词”这一符号信息表示成数学上的向量形式。

**3.1 词表示**

词在计算机中的表示通常有两种方式：one-hot 表示和distribution 表示。

**3.1.1 one-hot representation**

考虑一个词表V，里面的每一个词 wi都有一个编号 i∈{1,...,|V|}，|V|表示词表大小，那么词 wi的one-hot表示就是一个维度为|V|的向量，其中第i个元素值非零，其余元素全为0，举例：



**图2 语料库示例**

根据这段文本我们构建的一个词表：{1:洛塞泰斯;2:认为;3:中国;4:科技; ... |V|:新科技}，其中每个词的向量表示为：

洛塞泰斯：[1,0,0,...,0]，认为：[0,1,0,...,0]，新科技：[0,0,...,0,1]

one-hot表示的词向量的缺点：一是容易受到**维数灾难**的困扰，二是**无法表示出词与词之间的语义**信息。

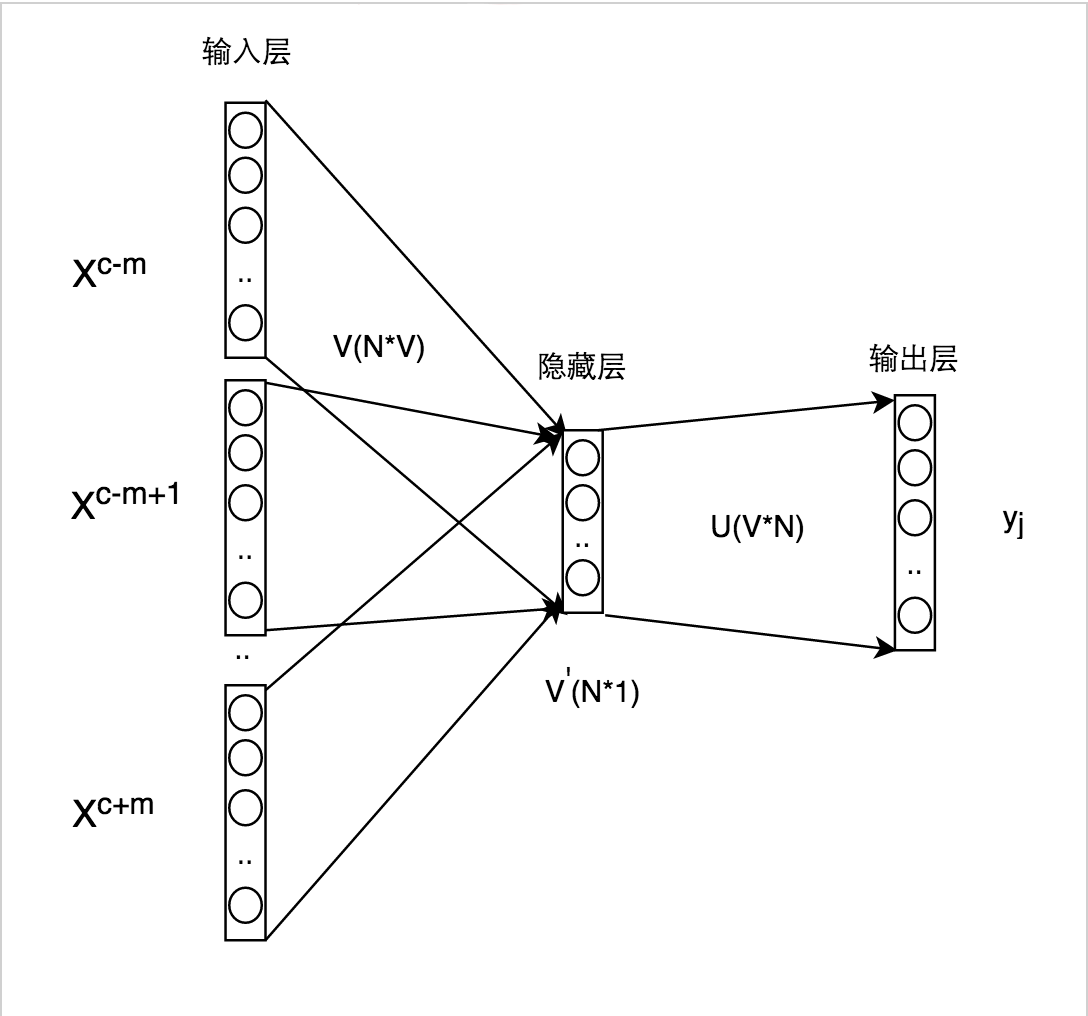
**3.1.2 distribution representation**

Harris在1954年提出的分布假说（ distributional hypothesis）为将语义融入到词表示中提供了理论基础：**上下文相似的词，其语义也相似**。Firth 在 1957 年对分布假说进行了进一步阐述和明确：**词的语义由其上下文决定**。

到目前为止，基于分布假说的词表示方法，按照使用技术手段不同，主要分为三类：基于矩阵的分布式表示、基于聚类的分布式表示、基于神经网络的分布式表示（经典Bengio神经网络语言模型）。

在经典Bengio神经网络语言模型中，Bengio 等人的工作只考虑对语言模型的建模，词向量只是语言模型结束的副产品，因此他们并没有指出哪一套向量作为词向量效果更好。Mikolov 等人在NNLM等模型的基础之上，设计了CBOW（Continuous Bag-of-Words）和 Skip-gram 模型两个模型，高效地获取词向量。

**3.2 CBOW**

****

**图3 CBOW网络结构图**

CBOW有三层机构：输入层，投影层（中间词），输出层。

输入层：使用上下文各词的词向量的简单求和作为输入

投影层：所谓投影层就是神经网络中的隐藏层概念，由于在CBOW模型中隐藏层的作用是投影，所以也叫投影层更好理解，投影层输出的就是输入向量的加权平均

投影层哈夫曼树：计算在输出层每个结点的输入

输出层：

|  |  |
| --- | --- |
| m | 窗口大小 |
| N | 词汇表大小 |
|  | 投影层维度 |
|  | 输入层到投影层权重 |
| U | 投影层到输出层权重 |

NNLM计算复杂度为：

CBOW计算复杂度：

梯度计算：最后要使我们计算的概率达到最大，所以通过梯度上升的方法来更新哈夫曼的非叶子节点和投影层输入的上下文，上下文来自原始各个词。

**模型实例：**

假设语料库有10个词[今天，我，你，他，小明，玩，北京，去，和，好]，有一句话：‘今天我和小明去北京玩’，对这个句话分词之后就成了：‘今天’‘我’‘和’‘小明’‘去’‘北京’‘玩’，如果选择小明最为中间词，n=3，就是与小明前后三个词有关，然后将词语用one-hot表示：

今天：[1,0,0,0,0,0,0,0,0] >x1

我： [0,1,0,0,0,0,0,0,0] >x2

和： [0,0,0,0,0,0,0,1,0] >x3

去： [1,0,0,0,0,0,1,0,0] >x4

北京：[1,0,0,0,0,1,0,0,0] >x5

玩： [1,0,0,0,1,0,0,0,0] >x6

预测向量也就是小明的表示：小明：[0,0,0,1,0,0,0,0,0] 。可以看出来向量的维度就是语料库的大小，接下来将这六个词向量传入神经网络，累加得到：

x=x1+x2+x3+x4+x5+x6=[1,1,0,0,0,1,1,1,1,0]

则x就是输入层，即输入层是由小明的前后三个词生成的一个向量，为1\*10维（这里的10是语料库中词语的个数），我们这里是根据一个词语的上下文来预测这个词究竟是哪个，这个例子中就是根据小明这个词的前后三个词来预测小明这个位置出现各个词的概率，因为训练数据中这个词就是小明，所以小明出现的概率应该是最大的， 所以我们希望输出层的结果就是小明对应的向量，所以本例中，输出层期望的数据实际就是小明这个词构成的向量(可以认为是训练数据的标签) ，小明：[0,0,0,1,0,0,0,0,0]

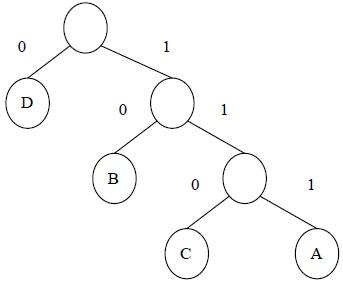
这里我们可以随意设定投影层的维度，即m的值，则：输入层和投影层的连线就可以构成一个参数矩阵1, 即w，该矩阵的维度为10\*m(这里10是语料库中词语的个数)， 可以随机初始化这个矩阵的参数值。投影层和输出层的连线就可以构成一个参数矩阵2，即w’， 该矩阵的维度为m\*10， 可以随机初始化这个矩阵的参数值，这样最终的输出层的维度1\*10 = (1\*10) \* (10\*m) \* (m\*10) ，首先，期望输出就是“小明” 对应的向量：

小明：[0,0,0,1,0,0,0]

模型输出的结果不会刚好就是一个one-hot向量， 可以认为趋近于0的就是0，趋近于1的就是1，当然为1的肯定只有一个元素，最终训练结束后，就要将一个词表示成一个向量。输入X中第一个元素的值为1， 这表示的其实就是“今天” 这个词，那么“今天”就用它对应的连线上的权重参数来表示

今天：[w11,w12,…..,w1m]

输出层连接的是一个哈夫曼树，树的叶子结点是语料库中的所有词，以个词在语料库中出现过的次数作为为权值，树的每一次分枝都是一次二分类。从二分类的角度来考虑问题，那么对于每一个非叶子结点，都需要对左右孩子结点指定类别，也就是正负标签，一般最自然的做法是将一个结点进行分类时，分到左边即为正类，分到右边即为负类，正类和负类的判断采用sigmoid函数。



**图4 哈夫曼树结构图**

**哈夫曼计算实例**：

还是以“今天 我 和 小明 去 北京 玩” 这句为例，输入层： “小明”的前后三个词向量，即首先要把这些词全部表示成向量，投影层： 将这写词的向量求和，得到的结果就是：

将投影层的向量输入到哈夫曼树中，哈夫曼树怎么来的？ 将语料库中词以及词出现的频率构造一颗哈夫曼树，默认左边(编码为0)是负类，右边(编码为1)是正类，哈夫曼树的叶子节点就是语料库中的所有的词，将向量Xw输入到哈夫曼树，在每个节点上进行二分类，每个节点上二分类的概率如下：

正类概率：

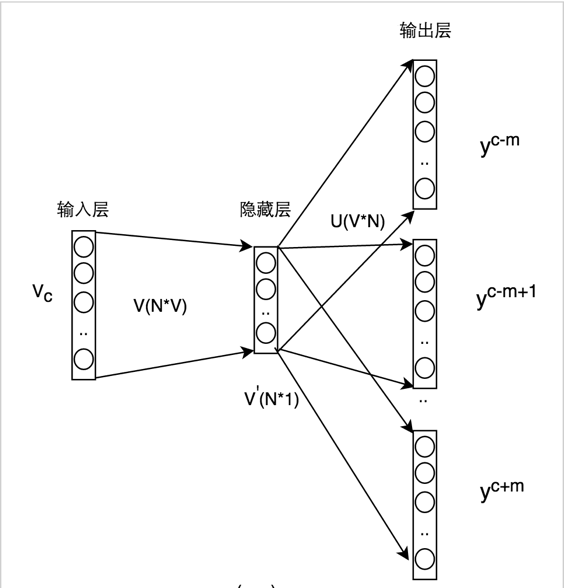
负类概率：

这里的概率是逻辑回归概率模型，那么我们的目标是沿着哈夫曼树找到最终的叶子节点是“小明”， 那么就求出走到“小明”这个叶子节点的路径的概率，其实就是将这条路线上的概率全部乘起来， 然后极大化这个概率，这里也是采用随机梯度上升法，最后要对各个词的词向量进行更新。

该模型使用一段文本的中间词作为目标词，并且CBOW模型去掉了隐藏层，加快了运算速度。此外 CBOW模型使用上下文各词的词向量的平均值代替NNLM模型各个拼接的词向量，由于CBOW去掉了隐藏层，所以其输入层就是语义上下文表示。CBOW对目标词的概率计算：

目标函数为最大化式子：

**3.3 Skip-gram**



**图4 Skip\_gram网络结构**

输入层：使用上下文各词的词向量的简单求和作为输入

投影层：没有上下文所以相对于做一个恒等映射

投影层哈夫曼树：计算在输出层每个结点的输入

输出层：

|  |  |
| --- | --- |
| m | 窗口大小 |
| N | 词汇表大小 |
|  | 投影层维度 |
|  | 输入层到投影层权重 |
| U | 投影层到输出层权重 |

Skip-gram计算复杂度：

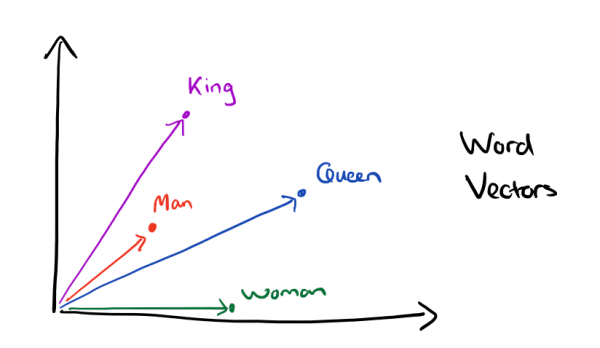
梯度计算：要使得最会预测词概率最大，因此要用梯度上升的方法。

理解了CBOW模型，Skip-gram也很简单。Skip-gram模型与CBOW模型类似，主要包含三层结构，输入层，投影层，输出层。Skip-gram模型的思想与CBOW相反，它是用中间词来预测上下，Skip-gram模型同样没有设置隐藏层，Skip-gram模型是从目标词w的上下文选择一个词，将其词向量组成上下文的表示。输出层是但前词的词向量。投影层因为没有上下文，所有是一个恒等映射，和CBOW模型一样输出层是一棵哈夫曼树，Skip-gram模型的目标函数为：

Skip-gram和CBOW模型实际上是word2vec两种不同思想的现实：CBOW的目标是根据上下文来预测但前词语的概率，且上下文所有的词对当前词出现概率的影响影响的权重是一样的。Skip-gram刚好相反，其是根据当前词语预测上下文概率的。在实际使用中，算法本身并无高下之分，读者可根据最后呈现的效果来进行算法的选择。

**上下文语义分析实例**：

经过大量的预料训练之后，我们的到有趣的发现，我们在二维坐标上面表示词向量国王，男人，女人，皇后：发现一个有趣的结果：



**图5 词向量的空间分布**

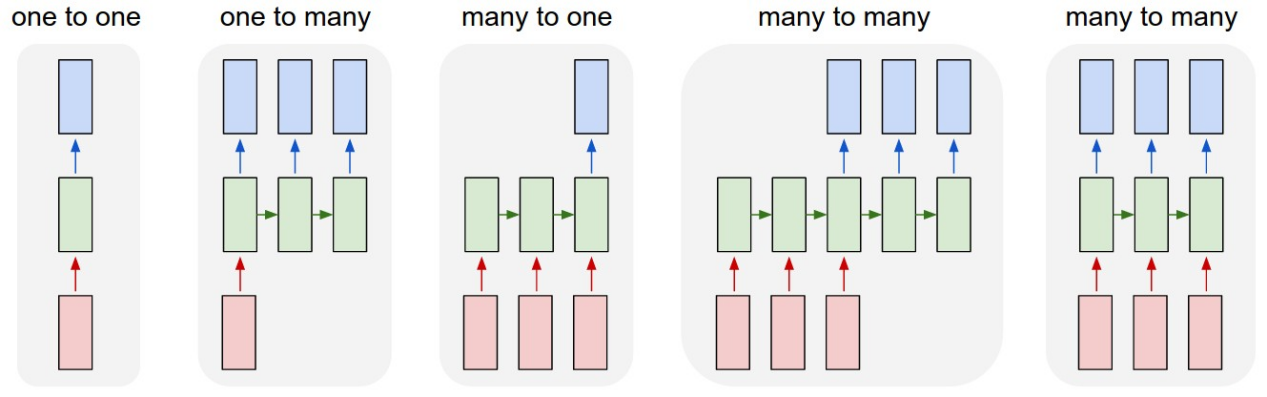
因为经过大量的预料训练之后，词向量包含了上下文语义，当我们使用国王的词向量减去男人的词向量之后，剩下的是信息是包含皇室特质的信息，当这些信息加上女人词向量之后，女人有了皇室的特质就成了皇后。

**4、 RNN循环神经网络**

RNN是一类用于处理序列数据的神经网络。这种网络的内部状态可以展示动态时序行为。不同于[前馈神经网络](https://baike.baidu.com/item/%E5%89%8D%E9%A6%88%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/7580523" \t "/Users/cy/Documents\\x/_blank)的是，RNN可以利用它内部的记忆来处理任意时序的输入。在人工神经网络中，单独的输入完全确定了剩下的层上的神经元的激活值。而在RNN中隐藏神经元的行为不只受到前一层的隐藏神经元的影响，还会受到更早的层上的神经元激活值的影响，也就是说隐藏层和输出层的神经元的激活值不会单单由当前的网络输入决定，而且包含了前面的输入的影响。

**4.1 序列模型**

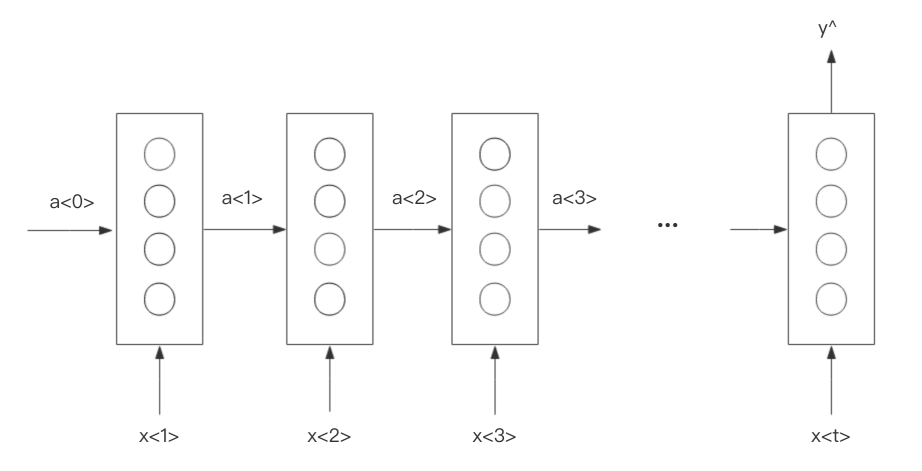
语音识别、音乐生成(o2m)、文本情感分类(m2o)、DNA序列分析、机器翻译、视频行为识别、命名实体识别(m2m)



**图6 序列模型的分类**

对自然语言这种时序数据直接建立标准的神经网络会存在两个问题：①每个**句子的长度不一样**②从文本不同位置学习到的**特征不共享**（假设神经网络学习已经到在位置1出现的harry是人名，如果能够学习到在其他位置出现的harry也是人名就更好了），并且同卷积网络类似，它可以**减少模型中的参数数量。**

**4.2 循环神经网络**



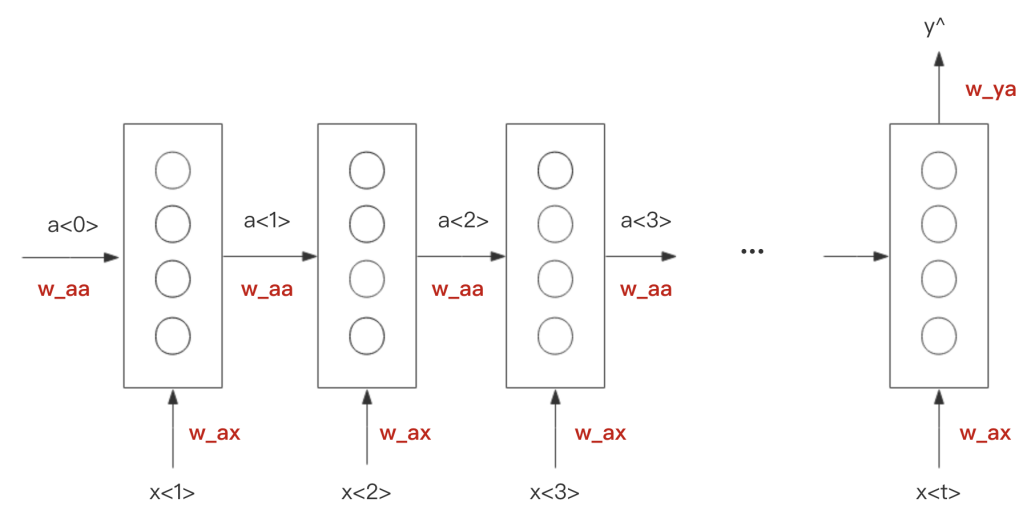
**图7 RNN网络结构图**

x<0>、x<1>、...、x<t>代表句子分词后的词向量，y代表句子标签。

a<t>是神经单元的激活值，一般使用零向量作为零时刻的伪激活值a<0>，即

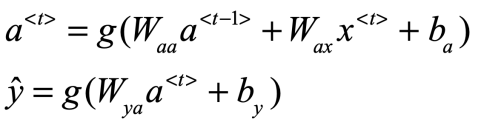
屏幕快照 2019-05-07 下午4.02.20

RNN的前向传播过程如下：

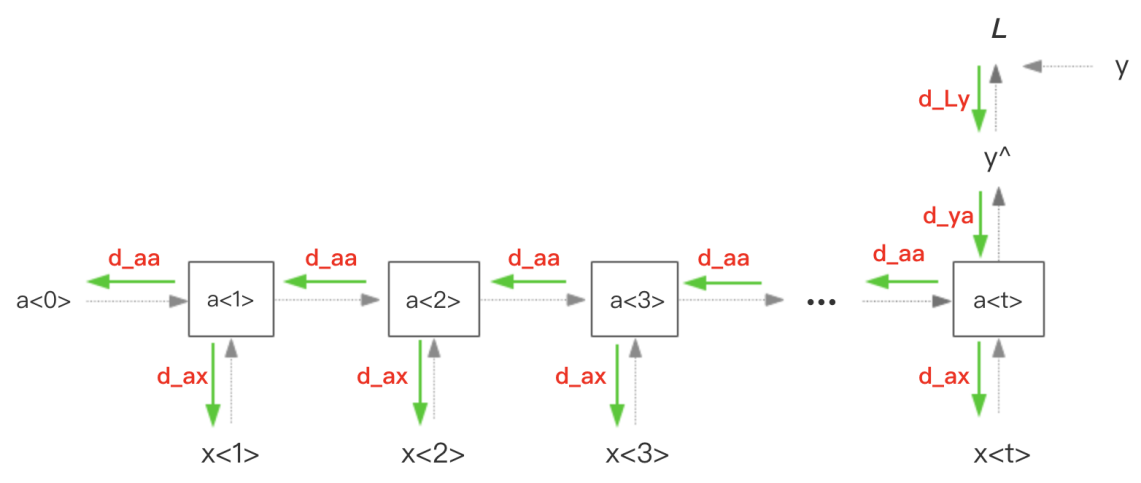


**图8 循环神经网络的前向传播**

前向传播公式：

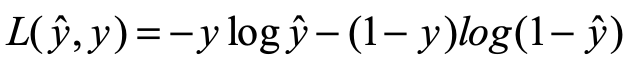


RNN的反向传播：



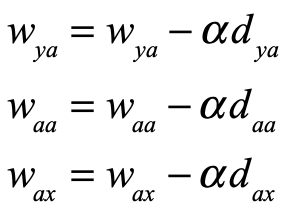
**图9 循环神经网络的反向传播**

反向传播需要定义损失函数：



d\_是求偏导符号，d\_Ly代表损失函数L对y求偏导,其他类似。

梯度更新公式：



超参a是步长值，这个值对于神经网络训练非常重要，决定了训练时间的长短，它的取值一般是从0.1到0.0001。

**4.3 长距离依赖问题**

影响 RNN 的一个挑战是前期的模型会很难训练，甚至比前馈神经网络更难。原因就是**不稳定梯度**的问题，该问题的通常表现就是在反向传播的时候梯度越变越小。这就使得前期的层学习非常缓慢。在 RNN 中这个问题更加糟糕，因为梯度不仅仅通过层反向传播，还会根据时间进行反向传播。如果网络运行了一段很⻓的时间，就会使得梯度特别不稳定，学不到东西。

**长短时记忆网络**（LSTM） 最早是由 Hochreiter 和 Schmidhuber 在 1997 年提出，就是为了解决RNN中不稳定梯度的问题。LSTM是RNN的一种特殊形式，特点是能够学习长距离依赖关系，由Hochreiter & Schmidhuber (1997)首先提出，其设计之初就是为了解决长距离依赖问题，之后被很多学者改善和推广，它在很多问题上都得到很好的表现，现在被广泛使使用。

1. **极端言论和情感积极消极算法**

**5.1 算法现状**

**极端言论**（自杀）算法是基于极端言论词典的文本分类方法，主要分为2个步骤，①使用**全网新闻语料库，**带入CBOW模型，模型内部自动提取上下文关系进行预测，不需人工打标签，模型训练好后，可以得到每个词的向量表示，根据向量间的距离计算，输出和“自杀”相关性最大的n个极端词，n是自定义的。②判断学生上网数据中是否包含词库中的词汇，包含就计数。

**情感积极消极**算法是基于深度网络的情感分类方法。算法思路：①使用**搜狗用户搜索行为记录**作为原始语料，通过人工打标签（积极、消极）得到模型训练所需数据，模型的输入为搜索行为记录，模型的输出为标签。先使用CBOW模型将输入和输出转化为词向量，然后带入LSTM模型，训练得到最优参数。②输出一条学生上网行为记录，根据最优参数，计算得到该记录是积极/消极的概率。

**5.2 存在的问题及改进建议**

①学生上网行为中包含两部分可用数据，一是搜索关键词，二是学生浏览的网页标题，因为关键词数据是学生的主动行为，所以在得分计算中需要给关键词数据设置比标题数据更高的权重，以消极得分为例：

是消极得分，是关键词的消极得分，是网页标题的消极得分，是关键词消极得分的权重，是标题消极得分的权重。

②学生的上网行为数据是一种**时序数据**，在对学生进行行为预测时，使用的是30天的平均得分，考虑到时序数据的特性，因此应该按根据时间设置权重，时间越接近当前时间，权重越高。

，且当时，

是30天的消极得分，是每天的消极得分，是每天的权重。

此外，我们还缺少对上网行为数据波动性的分析，比如，30天内每天的消极得分波动指标对最终得分的影响。

③情感积极消极算法中标签的覆盖范围不够，考虑到现实情况，大部分学生的搜索行为都是**中性**的，但模型的预测结果只有积极、消极两类，在具体应用时不合理，需要把中性行为这种情况也加入到模型进行训练。

④当前极端言论算法在预测学生上网行为是否极端时，是通过极端言论词典匹配的方式，并没有利用到上下文信息。一个解决方案就是使用深度网络RNN训练，但前提是需要人工打标签。那么就会存在这样一个问题，对于消极和极端言论标签，我们难以给出一个清晰的定义，比如，打游戏是否属于消极，看视频是否属于消极，这需要和专家交流再确定。

考虑到以上问题，我们可以换一种思路，先挖掘那些有明显异常的行为的学生，我们给出三个维度：色情、反动、暴力，暂时以这三个维度的情况去衡量学生的极端倾向。

具体做法是，给数据打色情、反动、暴力、其他标签，使用RNN训练四分类模型，计算色情、反动、暴力的综合得分，作为极端倾向得分，如果后续有其他相关指标，可以再增加。