◂ BIT ▸

情感分析

Sentiment Analysis

小组成员：李杨晓、郝家辉、李生椰、韩亚辉、李思远

北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

德以明理　学以精工

1 情感分析概述

什么是情感分析

情感分析，又称倾向性分析、意见抽取、意见挖掘、情感挖掘、主观分析。

即对带有情感色彩的主观性文本进行分析、处理、归纳和推理的过程。

情感分析的参与主体主要包括：

观点持有者、评价对象、评价观点和评价文本

情感分析研究方向

情感分类

情感分析技术的核心问题，其目标是判断主观文本的情感取向。

评价点抽取

包括对评价对象的抽取以及情感词的抽取。

观点概括

将用户的评论概括成一系列的评论点和情感词的pair，并统计比例。

垃圾观点识别

识别出观点中的虚假评论、非评论和针对性评论。

情感分类

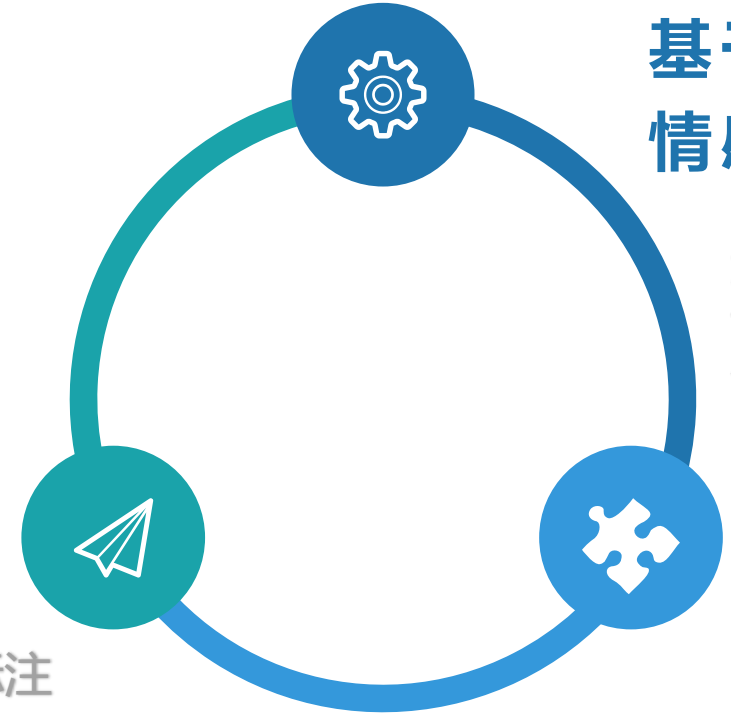
文本粒度

- 篇章级：对篇章所表达的情感进行分类。
- 句子级：判断句子的主客观，以及主观性句子的情感极性。
- 方面级：提取句子中针对不同实体的不同方面的意见，得到<方面，意见>二元组。

情感粒度

- 正/负二分类（或者正面/负面/中立三分类）：主客观信息的二元分类&主观信息的情感分类。
- 多元分类 （e.g.新闻评论中的“乐观” “悲伤” “愤怒” “惊讶”，商品评论的1~5星）

方法



基于机器学习的情感分析

通过对数据进行特征提取，
训练生成文本情感分析模型。
然后进行文本情感分类

基于情感词典的情感分析

构建情感词典，
对词典进行极性和强度标注
然后进行文本情感分类

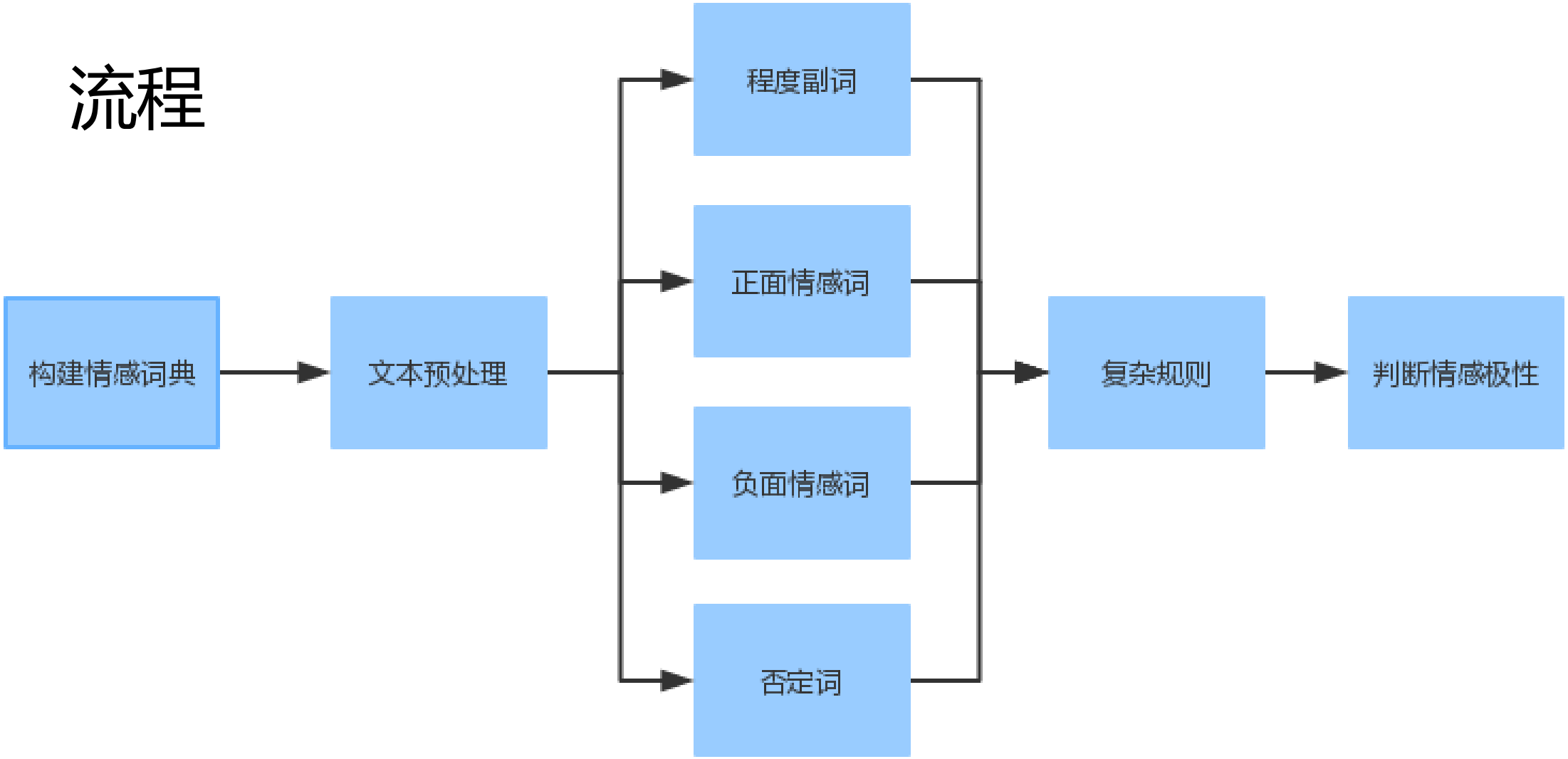
基于深度学习的情感分析

不依赖特征工程，
端到端的对输入文本进行语义理解，
并基于语义表示进行情感倾向的判断

2 基于情感词典的情感分析

基于情感词典的情感分析

流程



基于情感词典的情感分析

这个电视剧剧情有点辣鸡，但是男主女主颜值特别高

预处理

这个/电视剧/剧情/有点/辣鸡/但是/男主/女主/颜值/特别/高

进行计算

(-1)*2+1*3=1

positive!

$$s(\mathrm{i}) = \sum_{j}^{n} (-1)^{t} * k * word(j)$$

1.基于启发式规则的方法

观察大量语料的特性，找到一些语法模式、语法规则、语义特征和语言学特性，然后抽取情感词并判断其极性。

1.通过文本中的连词连接的形容词对判断极性
“她的这篇文章内容丰富又有趣”

2.通过目标词语与种子情感词的互信息来判断极性

$$PMI(\omega_1, \quad \omega_2) = \log(\frac{p(\omega_1 \& \omega_2)}{p(\omega_1)p(\omega_2)})$$

2.基于图的方法

- 建立一个图，顶点是待抽取的目标词或目标词组组成，边的权值为两个顶点之间的相似度
- 在这个图上用一个图传播的算法进行迭代，计算顶点的情感值。

3.基于词对齐模型的方法

Liu等人提出利用机器翻译模型IBM挖掘情感词和评价对象之间的关系

- 构建二分图的顶点，选取所有的形容词作为候选情感词定点，所有名词或名词词组作为候选评价对象顶点
- 利用机器翻译模型IBM挖掘情感词和评价对象之间的关系
- 采用基于图的算法计算候选词的置信度，置信度越高，被选择的概率越大

4.基于表示学习的方法

- 将文本情感信息加入Skip-gram模型词组的向量表示中
- 开发具有混合损失函数的专用神经结构
- 结合来自文本的情感极性的监督
- 用分类器预测词汇表中每个短语的情感分数

基于情感词典的情感分析

优点：

可解释性强

缺点：

情感词典无法包含所有的情感表达词语，不具有普遍性，不同的领域需要不同的情感词典，并且一些新词不能够及时地加入到词典当中，导致情感分析准确率较低。

另外，中文会出现各种语言现象，如反讽、褒义贬用、贬义褒用等。

3 基于机器学习的情感分析

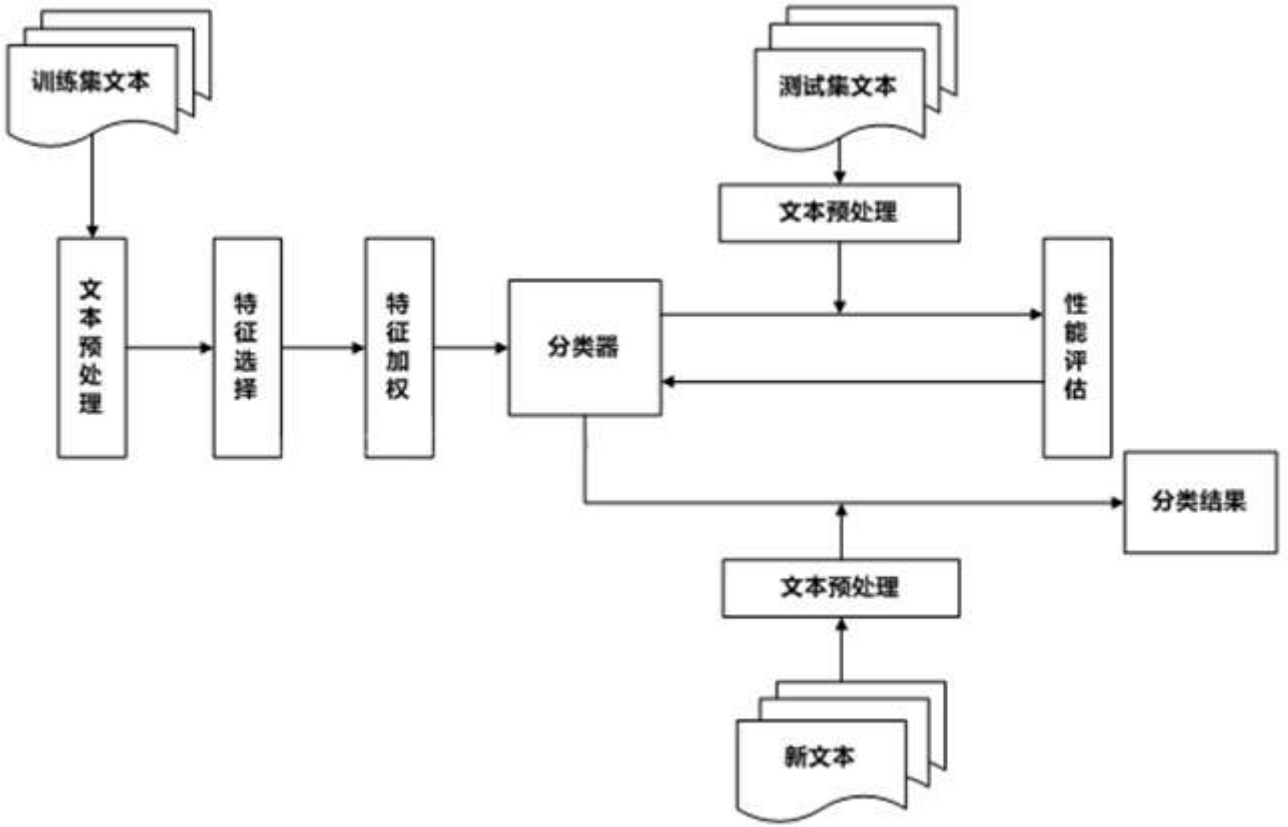
机器学习+特征工程

机器学习+特征工程

- 需要大量标注训练集
- 需要做特征工程

基本过程

- 数据预处理
- 特征提取
- 特征选择
- 使用分类器分类



数据预处理

中文分词

中文文档不像英文文档有特定的间隔符来对词进行分割，需要显示的进行分词任务将句子分割成词。

去除停用词

为节省存储空间和提高搜索效率，在处理自然语言数据时会自动过滤掉某些字或词，这些字或词即被称为Stop Words(停用词)如“的”“都”“此”以及一些十分常见的符号

文本表示

词袋模型
主题模型，LSA，LDA
分布式表示，Word2Vec，Glove

基于词袋的模型

- Naïve版本：只考虑词出现与否
- Term Frequency：考虑词出现的频率
- TF-IDF：考虑逆文档频率
- 特点:高维度，高稀疏
- 没有考虑语义的顺序

TF-IDF

- 词频 (TF)：某一个给定的词语在该文件中出现的次数。反映该词描述文档的能力

$$TF_w = \frac{\text{在某一类中词条}w\text{出现的次数}}{\text{该类中所有的词条数目}}$$

- 逆向文件频率 (IDF)：由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。IDF的思想：如果包含词条t的文档越少, IDF越大，该词区分文档的能力越好。

$$IDF = log(\frac{\text{语料库的文档总数}}{\text{包含词条}w\text{的文档数}+1})$$

- 文件内的高词频，整个文件集合低文件词频会产生高TF-IDF权重。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。

$$TF-IDF(t,d) = TF(t,d) \times IF(t)$$

- 将其用于特征权重计算，十分普遍。

卡方检验（Chi-square）

卡方检验一种假设检验方法。最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。常常先假设两个变量确实是独立的（“原假设”），然后观察实际值（观察值）与理论值（这个理论值是指“如果两者确实独立”的情况下应该有的值）的偏差程度，如果大于某个阈值，则认为两者相关，否定原假设。

$$x^2 = \sum \frac{(A-T)^2}{T}$$

A：观察值
T：实际值
X2: 结果衡量差异程度

卡方检验用于特征选择

在特征选择阶段，对于一个词条t和情感类别c，我们做出“词t与类别c不相关”的原假设，然后计算卡方值，若卡方值较大，则倾向于“词t与类别c存在相关关系”，也就是词t对于类别c有更强的表征能力。
与一般卡方检验不同，我们不用设定阈值，只需将卡方值降序排序，取前k个作为特征。

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

N：训练数据集文档总数
A：包词词条t，同时属于类别c的文档的数量
B：包含词条t，但是不属于类别c的文档的数量
C：属于类别c，但是不包含词条t的文档的数量
D：不属于类别c，同时也不包含词条t的文档的数

分类器

将从特征工程中获得的特征向量送入送入分类其进行训练与分类

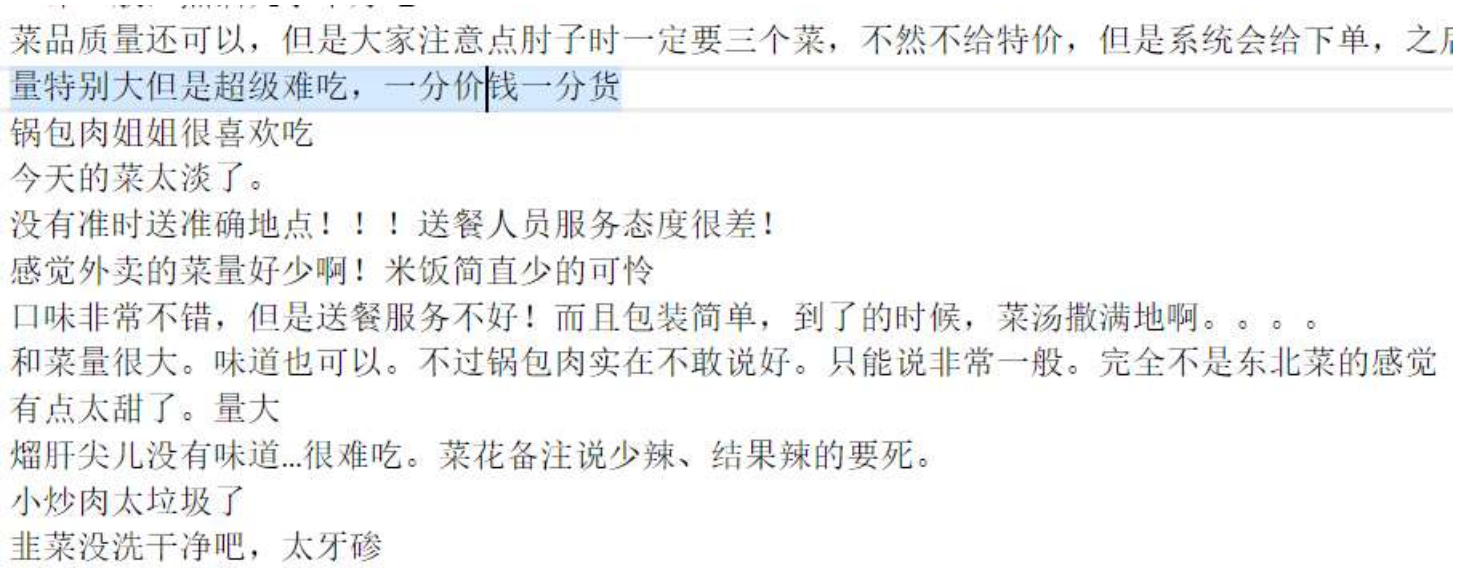
常见的分类方法

- 朴素贝叶斯
- K近邻
- 决策树
- 支持向量机
- ……

外卖评论数据的情感极性分析

流程

- 分词，去除停用词
- 卡方检验选择特征
- TF-IDF计算特征权重
- 使用SVM分类



外卖评论数据的情感极性分析

预处理

- 分词有许多现成的库，使用jieba进行分词
- 停用词：分词的直接结果包含许多无效项，要剔除
- 使用哈工大的停用词表，包括近800条的无用符号和无效词

```
——
》），
）÷（1－
”，
）、
＝（
：
→
℃
&
*
——
~~~~
，
```

```
有的是
也就是说
末##末
啊
阿
哎
哎呀
哎哟
唉
俺
俺们
按
按照
吧
吧哒
```

长得	精神	，	服务态度	好	，	速度	快								
都	很	好吃	，	量	很	足	！	快递	准时						
软炸	里脊	应该	有	蘸	料	的	啊	，	但是	菜	真的	很	好	哦	
谢谢	快递	师傅	，	谢谢	厨师	师傅	啦								
小伙子	人	太好了													
速度	挺快	菜	也	不错	。										

外卖评论数据的情感极性分析

特征提取和选择

- 使用卡方检验，选出800个词作为特征
- 计算tf-idf作为特征权重

分类

- 使用SVM进行分类
- L2正则化处理过拟合
- 使用sklearn的现成方法

外卖评论数据的情感极性分析

训练与测试

- 选去8000条外卖评论数据，正负评论各4000条
- 选取正负各3000条用于训练，1000条测试
- 实验的F1值均不是很高

POS-P	POS-R	POS-F1	NEG-P	NEG-R	NEG-F1
86.72566372	68.6	76.60525	74.028122	89.5	81.032141

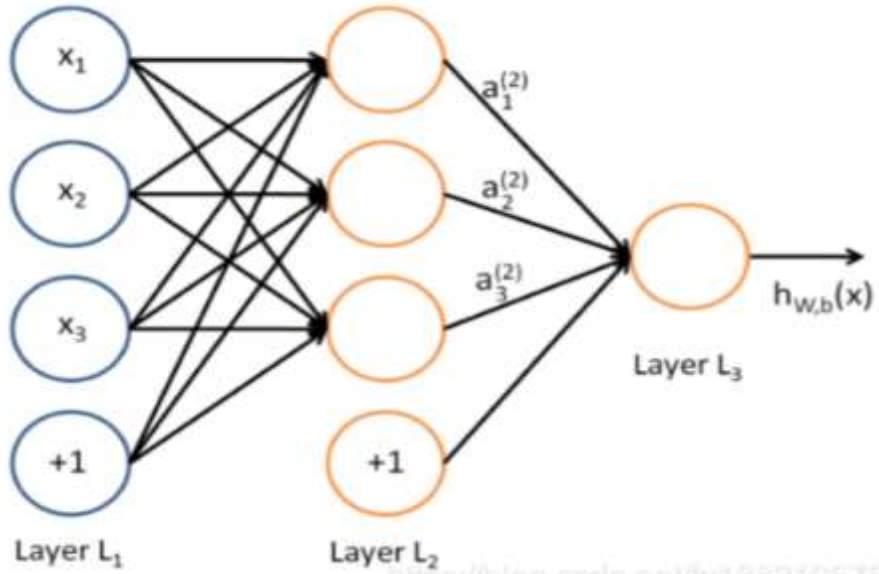
总结

- 词袋模型忽略了上下文信息。高维且稀疏。
- 特征工程优化太繁琐，需要神经网络自动提取特征

4 基于深度学习的情感分析

MLP（多层感知机）

- 一个三层的神经网络图



$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$
$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$

MLP缺点

1） 有隐藏层的MLP包含一个非凸性损失函数，存在超过一个最小值，所以不同的随机初始权重可能导致不同验证精确度

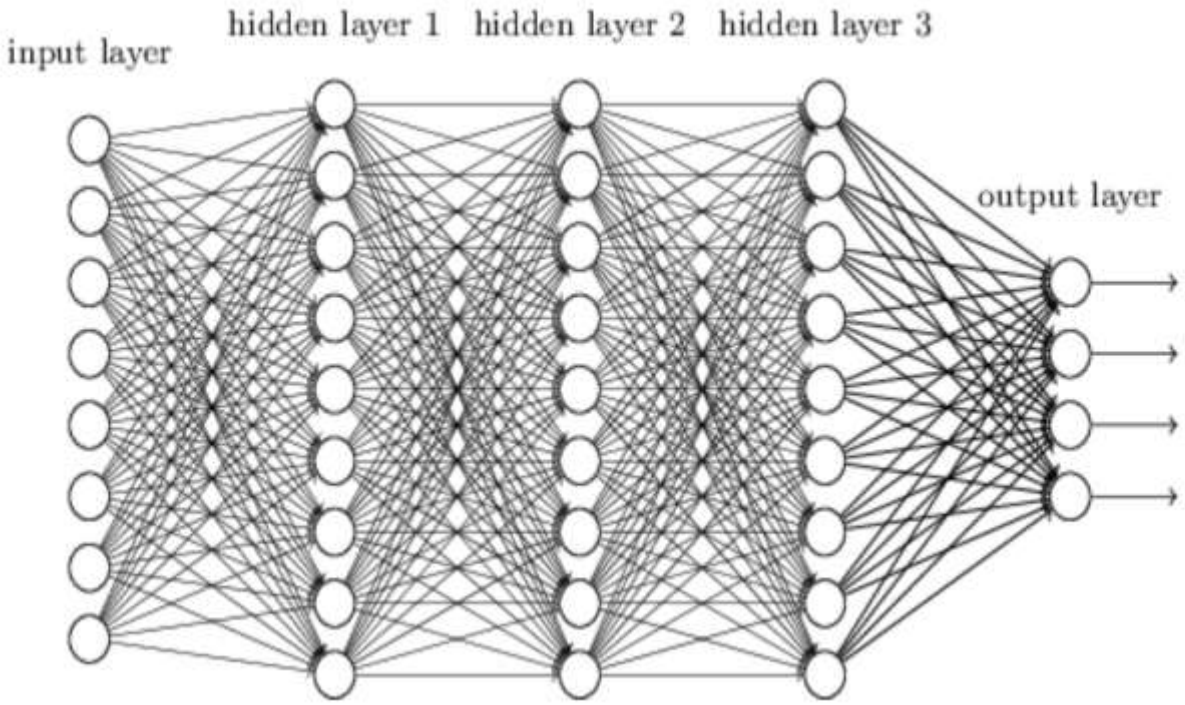
2） MLP要求调整一系列超参数，比如隐藏神经元，隐藏层的个数以及迭代的次数

3） MLP对特征缩放比较敏感

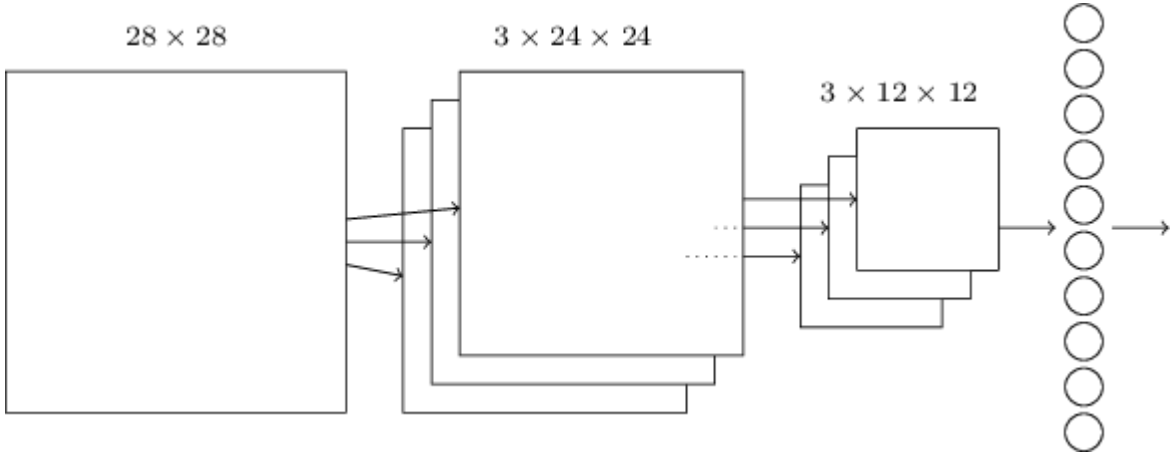
4） 陷入局部最优解

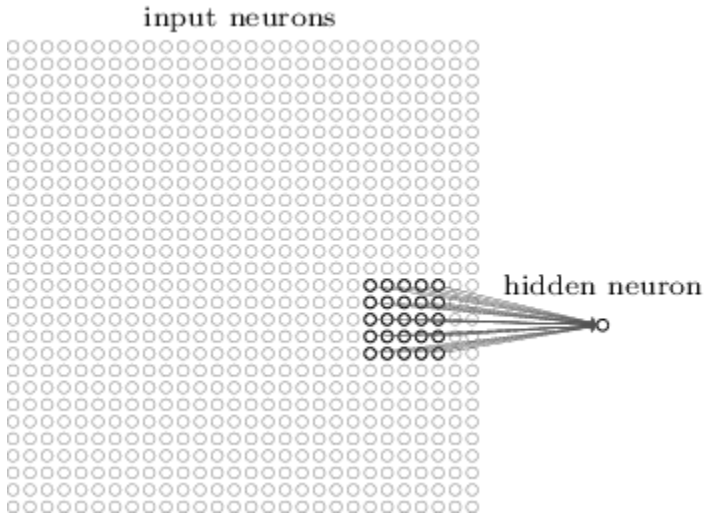
5） 梯度消失问题

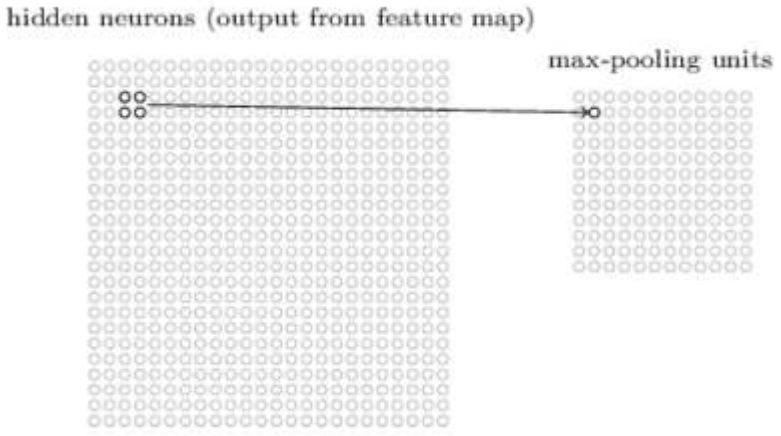
DNN（深度神经网络）


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

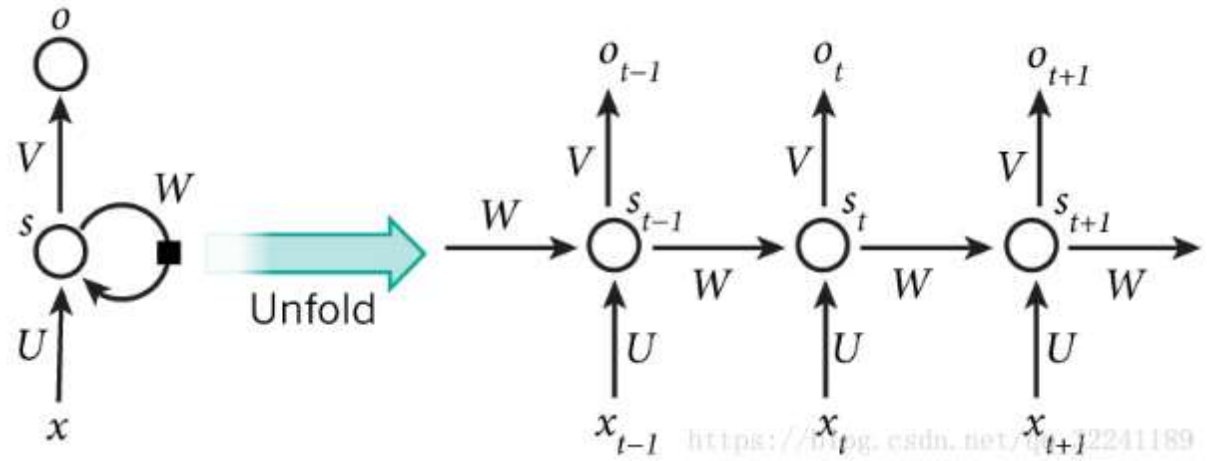
CNN（卷积神经网络）


28 × 28
3 × 24 × 24
3 × 12 × 12


input neurons
hidden neuron


hidden neurons (output from feature map)
max-pooling units

RNN（循环神经网络）



$$h_t = Ux_t + Ws_{t-1}$$
$$s_t = f(h_t)$$
$$o_t = g(Vs_t)$$

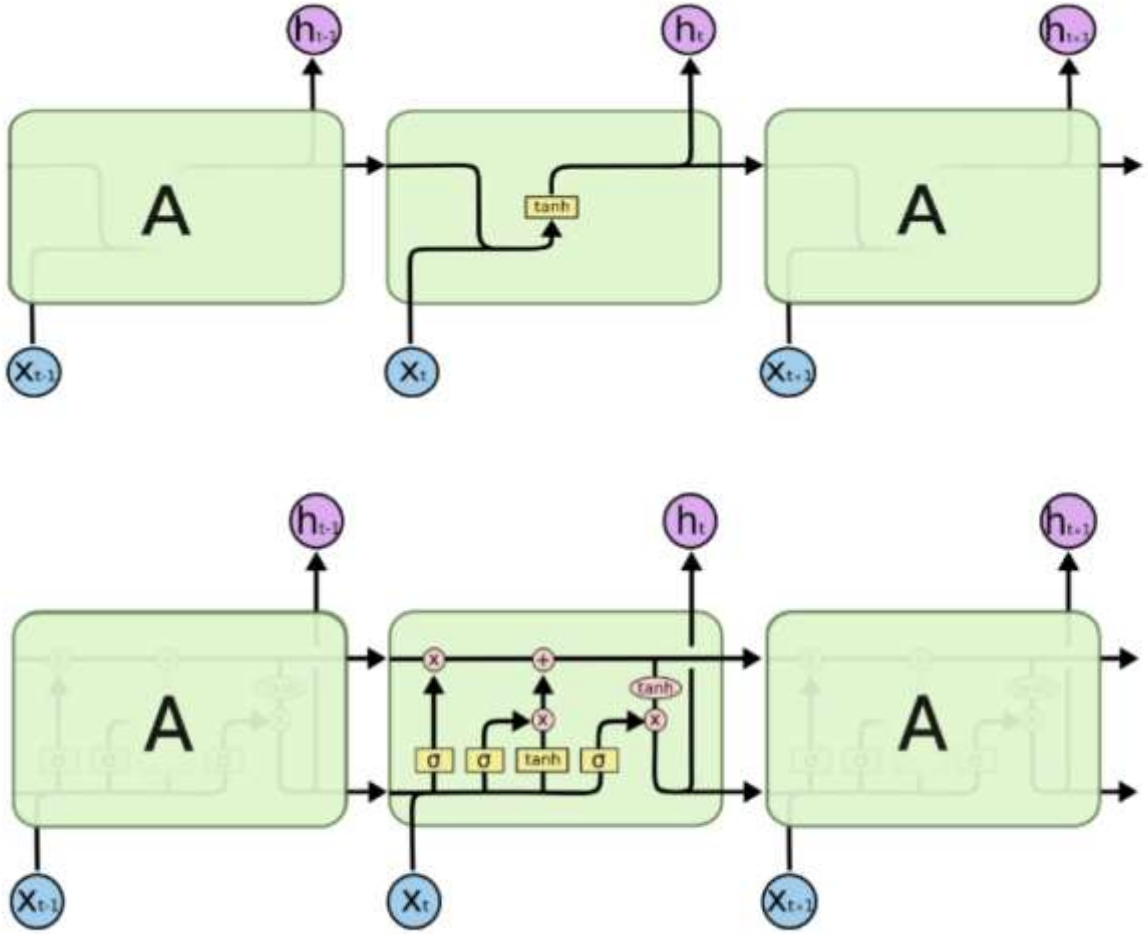
RNN的缺点

- 长期依赖问题

The cat, which already ate a bunch of food, was full.

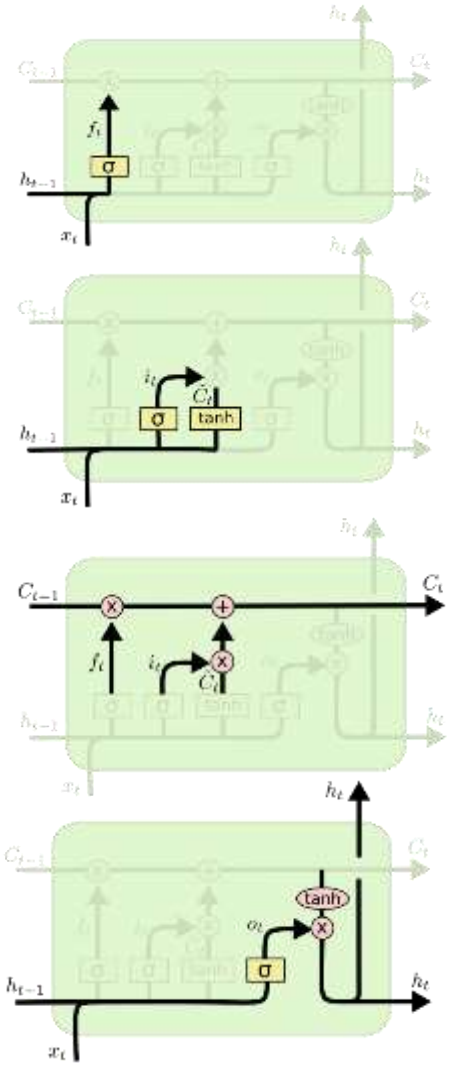
The cats, which already ate a bunch of food, were full.

LSTM VS 普通的RNN



LSTM

- 遗忘门
- 输入门
- 细胞状态更新
- 输出门



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

基于深度学习的文本分类

数据预处理

1）分词：中文特有，将文本分割成由词语组成的形式（例如：中国 人民解放军，分词工具很多，jieba分词工具比较好。）

2）除去停用词：将文本中的无用词去掉（如 标点符号，了，也等），使后期模型简单、可靠。

3）词嵌入：把词从高维的稀疏空间嵌入到低维的连续空间，极大提升模型的效果

深度学习模型分类

1）基于CNN的模型textCNN（原理和实例）

2）基于LSTM的模型（原理）

3）HAN/BERT/texGCN...

单词的one-hot表示

词典 {China, America, Japan}

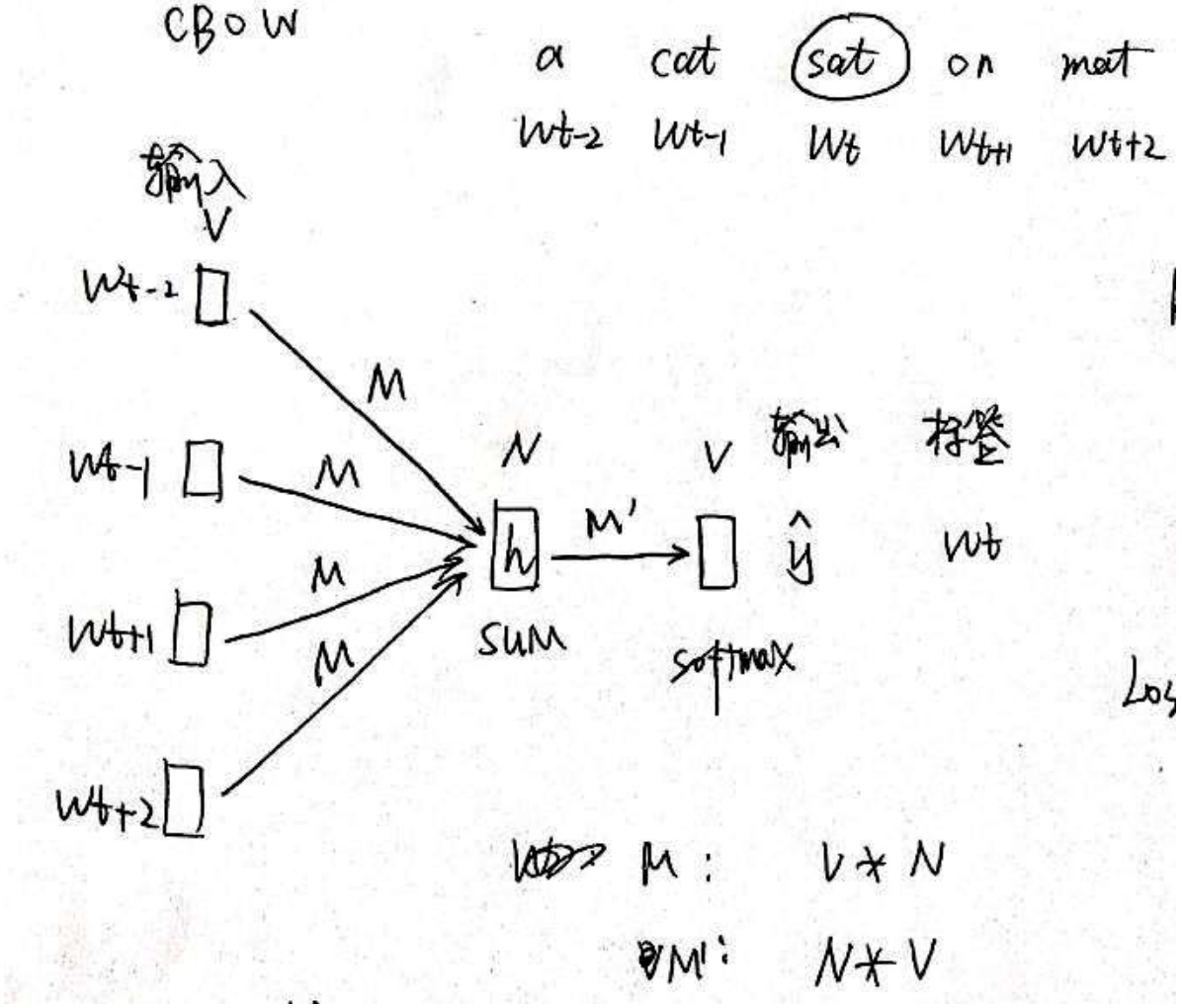
one-hot 表示

China $\begin{bmatrix}1\\0\\0\end{bmatrix}$ America $\begin{bmatrix}0\\1\\0\end{bmatrix}$ Japan $\begin{bmatrix}0\\0\\1\end{bmatrix}$

one-hot 的维度 = 词典中词语个数 |V|

缺点：稀疏、维数灾难.

word2vec-CBOW



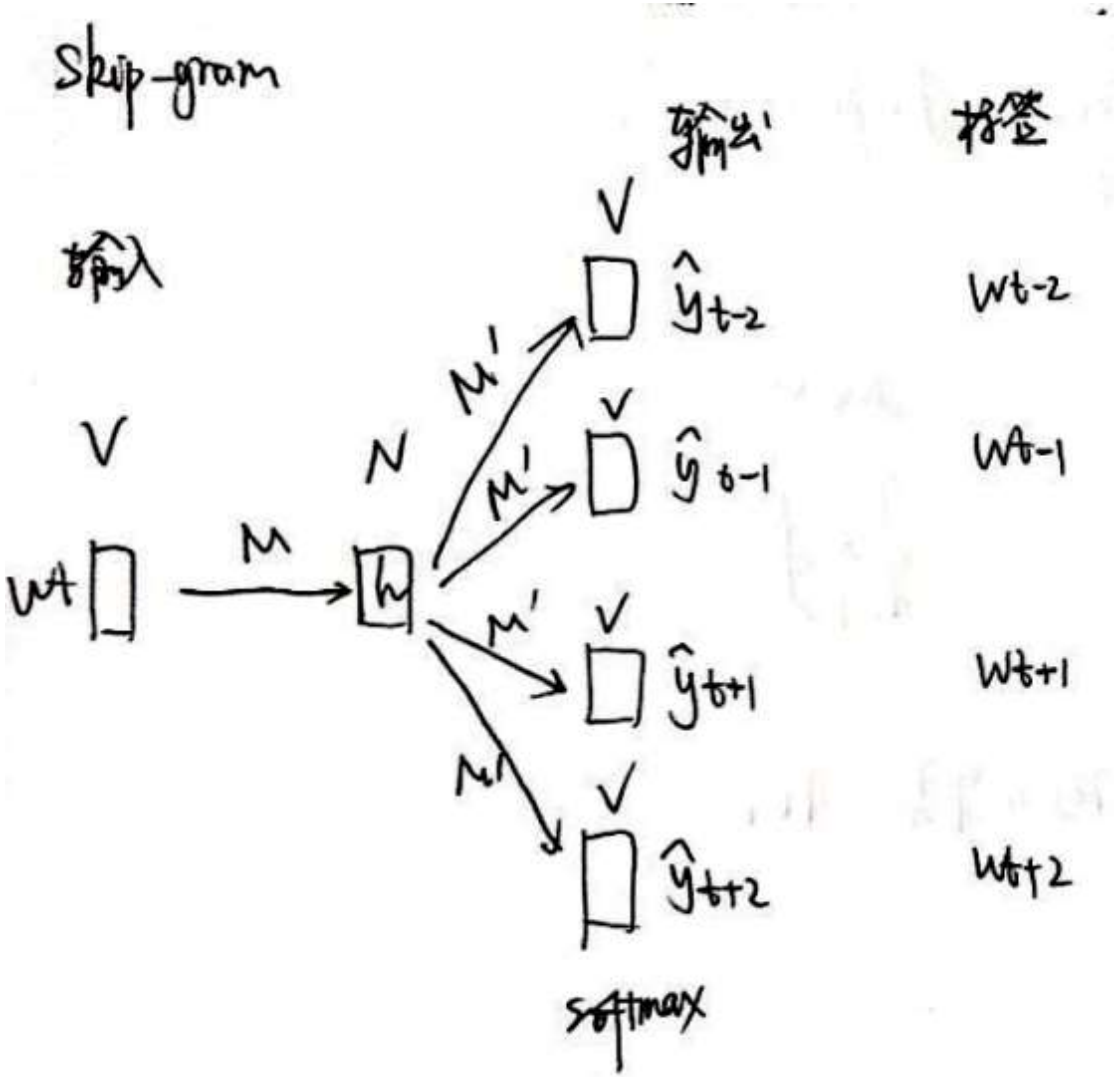
$$h = W_{t-2} * M + W_{t-1} * M$$

$$+ W_{t+1} * M + W_{t+2} * M$$

$$\hat{y} = \text{softmax}(h * M')$$

$$\text{Loss} = \text{CrossEntropy}(\hat{y}, W_t)$$

word2vec-skipgram



$$h = w_t * M$$

$$\hat{y}_{t-2} = \text{softmax}(h * M')$$

$$\hat{y}_{t+2} = \hat{y}_{t+1} = \hat{y}_{t-1} = \hat{y}_{t-2}$$

$$\text{Loss} = \text{CrossEntropy}(\hat{y}_{t-2}, w_{t-2}) + \text{CrossEntropy}(\hat{y}_{t-1}, w_{t-1}) + \text{CrossEntropy}(\hat{y}_{t+1}, w_{t+1}) + \text{CrossEntropy}(\hat{y}_{t+2}, w_{t+2})$$

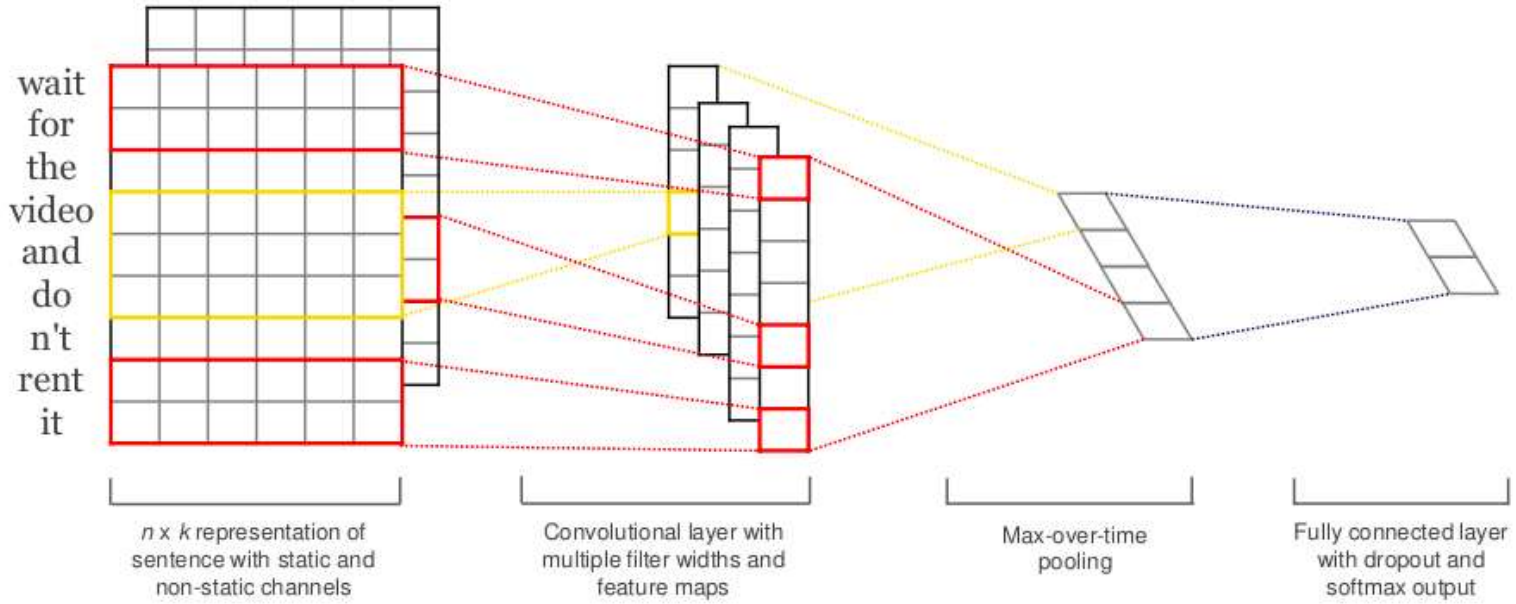
word2vec训练后会得到一个权值矩阵M，这个矩阵就是所有词的向量表示。这个矩阵的每一行，就是一个词的矢量表示。如图，如果两个矩阵相乘...

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

根据ont-hot编码的特点，在矩阵相乘的时候，就是选取出矩阵中的某一行，而这一行就是我们输入这个词语的word2vec表示

textCNN

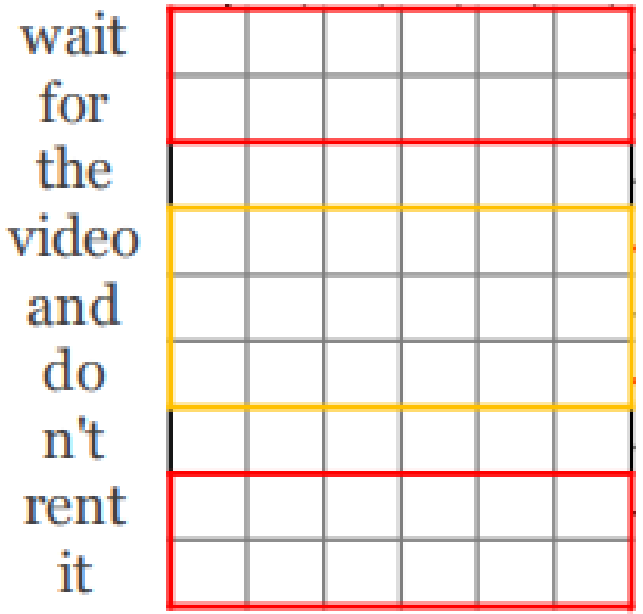
Yoon Kim在论文(2014 EMNLP) Convolutional Neural Networks for Sentence Classification提出TextCNN。将卷积神经网络CNN应用到文本分类任务，利用多个不同size的kernel来提取句子中的关键信息（类似于多窗口大小的ngram），从而能够更好地捕捉局部相关性。



textCNN数据预处理

使用预训练的word2vec模型，将单词表示为词向量。句子单词个数为n，词向量长度为k，则这个句子就可以使用n*k的矩阵表示，如下图，n=9, k=6

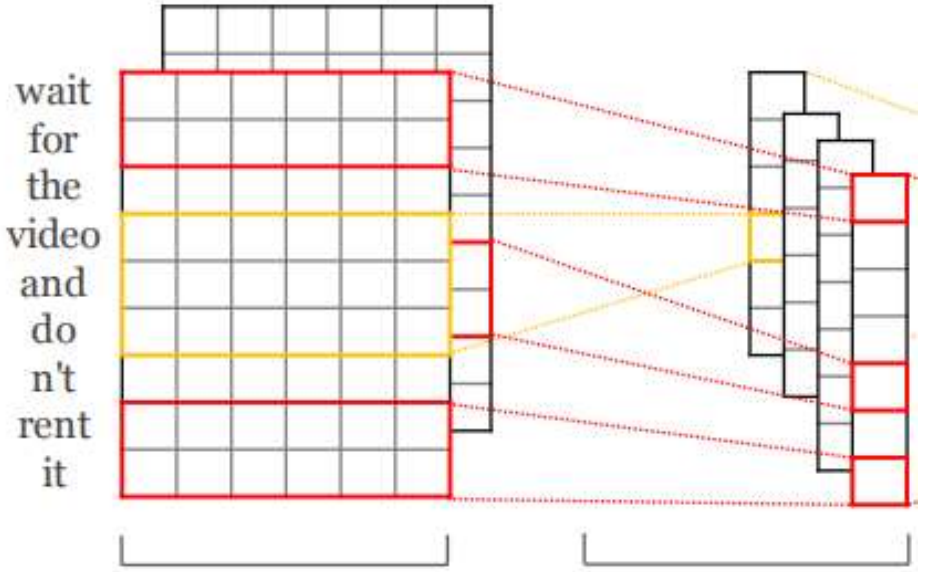
通常句子的长度不是固定的，那么需要预先设置n值，如果句子太长就截断，太短就补0.



textCNN模型结构

卷积

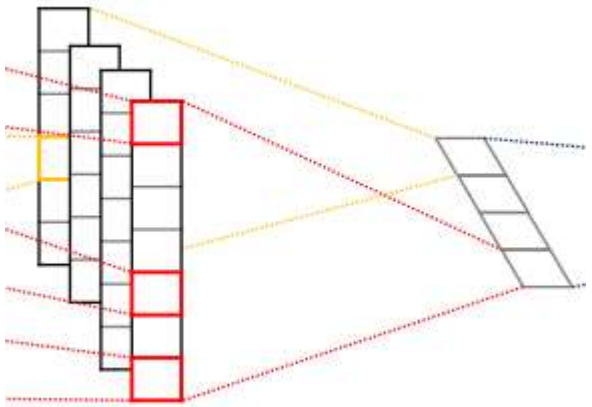
CNN的输入是n*k的矩阵，使用2*k的卷积核进行卷积运算输出一个值，对整个矩阵从上到下进行卷积，就生成一个向量。使用不同大小的卷积核，如3*k，4*k...等就能生成具有不同感受野的向量。



textCNN模型结构

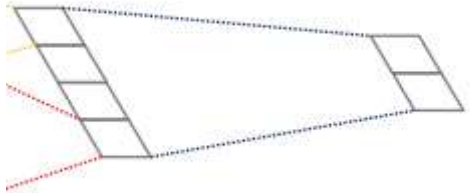
池化

对向量进行最大池化，生成特征向量， 以右图为例，前面通过卷积生成4个向量，最大池化就是把向量的各个分量相加，最终生成4个标量，这4个标量组成一个特征向量.



全连接层

特征向量接全连接层，使用softmax输出分类结果。作者为了防止过拟合，使用了dropout.



textCNN实现

```
class CNN_Text(nn.Module):
    def __init__(self, args):
        super(CNN_Text, self).__init__()
        self.args = args

        V = args.embed_num          # 语料中词典大小
        D = args.embed_dim                  # 词向量长度k
        C = args.class_num          # 类的个数
        Ci = 1                              # 输入通道数
        Co = args.kernel_num       # 卷积核的个数

        Ks = args.kernel_sizes          # 卷积核的大小

        self.embed = nn.Embedding(V, D)         # 词嵌入

        self.convs1 = nn.ModuleList([nn.Conv2d(Ci, Co, (K, D)) for K in Ks])

        self.dropout = nn.Dropout(args.dropout) # dropout，防止过拟合

        self.fc1 = nn.Linear(len(Ks)*Co, C)         # 全连接层

    def forward(self, x):
        x = self.embed(x)  # (N, W, D)

        if self.args.static:
            x = Variable(x)

        x = x.unsqueeze(1)  # (N, Ci, W, D)

        x = [F.relu(conv(x)).squeeze(3) for conv in self.convs1]  # [(N, Co, W), ...]*len(Ks)

        x = [F.max_pool1d(i, i.size(2)).squeeze(2) for i in x]  # [(N, Co), ...]*len(Ks)

        x = torch.cat(x, 1)

        x = self.dropout(x)  # (N, len(Ks)*Co)
        logit = self.fc1(x)  # (N, C)
        return logit
```

textCNN微博文本情感分类

数据集

10 万多条带情感标注新浪微博数据，正负向评论约各 5 万条 。将数据集随机划分成70%的训练集共83991条，30%的测试集共35997条，训练集和测试集中正负评论比例为1:1

[网盘链接](https://pan.baidu.com/s/1DoQbki3YwqkuwQUOj64R_g)

1	不用拒绝[哈哈] //@王振华:这次你再也不能拒绝[抱抱]
1	#舌尖上的望京一号#多好的媳妇/@雨落-牛妈: 在第五大道给我买份礼物，我带你和你爹娘到
0	现在的鞋都什么质量？！我老妈穿了几十年的高跟鞋，也没听她说过卡脚，更别说是卡烂还
1	//@飞乐鸟: 我是胖纸，卖萌来得~[亲亲]
1	感谢大家的支持和关注哈，这次的免费课已经报满了。请报上名的同学准时来上课，没有报
0	今年我不要嘴勤屁股懒、今年我不要拖延症。[抓狂]
0	[悲伤][悲伤][挖鼻屎]//@我的名字招谁惹谁了: 商家也太用心了。[抓狂]
1	在家就可以吃到正宗的四川火锅//@那小宁子是好娃: 这个好这个号[爱你][馋嘴][馋嘴][馋嘴]
1	[鼓掌] //@行娟子:[围观]
0	《这个杀手不太冷》，亦或《杀手里昂》，值得一看的片子：）图中是两主角：）//@小鱼妈

一共训练了两轮，测试集准确率达到98.22%.