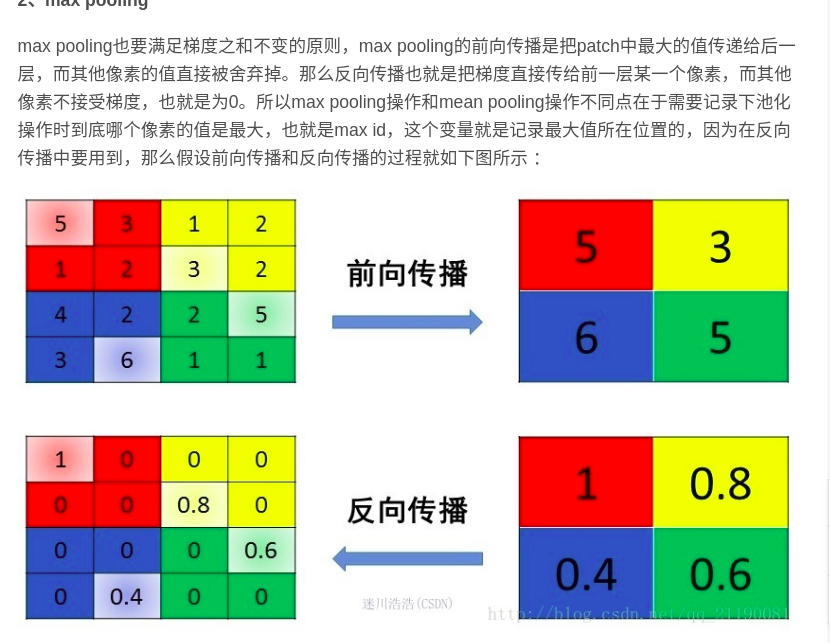
面试问题汇总

**1.香浓科技**

面试题目

# 1.CNN中max pooling 反向传播是怎么实现的？

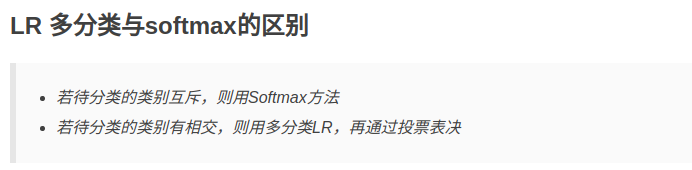
# [池化层（pooling）的反向传播是怎么实现的](https://blog.csdn.net/hellocsz/article/details/88096446)



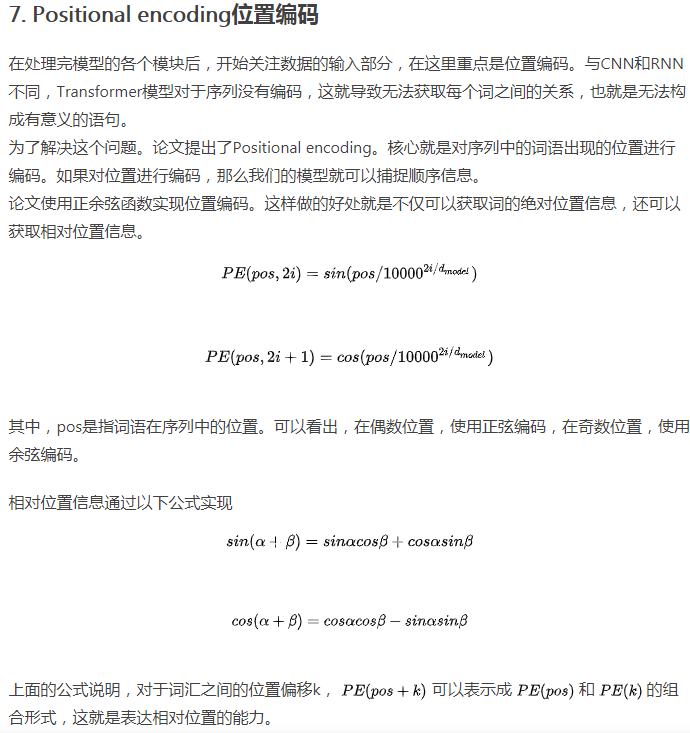
# 2.什么是cnn的稀疏交互和参数共享

# 3.softmax和lr的区别

# **[机器学习算法：LR多分类与softmax分类](https://www.jianshu.com/p/8b8cecf8836a)**



4.为什么transformer要加位置编码



6.batch normalization和layer normalization什么区别

# [Batch Normalization和Layer Normalization的原理分析对比，及Tensorflow代码实现](https://www.jianshu.com/p/367c456cc4cf)

Batch Normalization 针对多个样本的同一维度的输出值；

Layer Normalization 针对一个样本各个维度的输出值；

反正有多个输出值就可做归一化，只是横向和纵向的区别。完事～

经过归一化再输入激活函数，得到的值大部分会落入**非线性函数的线性区**，导数远离导数饱和区，避免了梯度消失，这样来加速训练收敛过程。

7.各种排序时间空间复杂度



8.



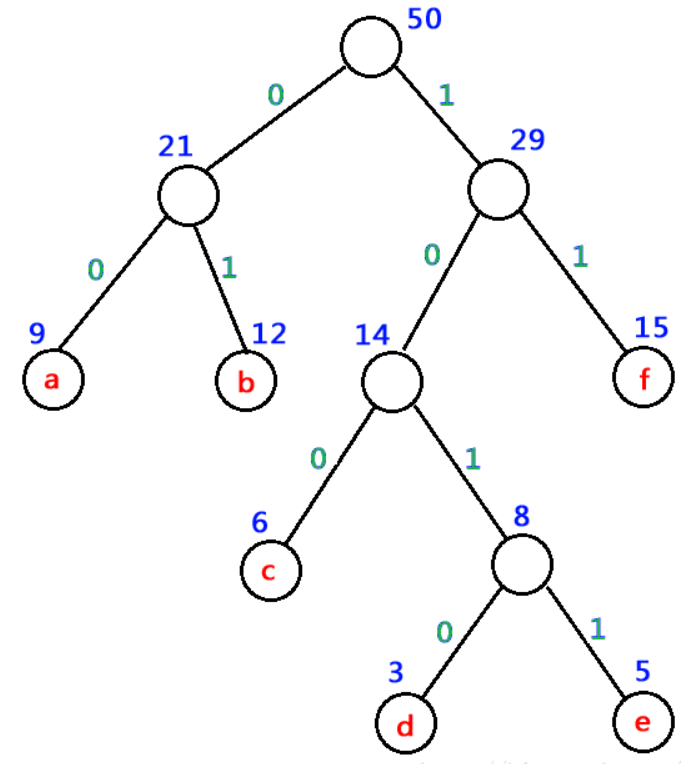
# 9.[哈夫曼树的构建、编码以及带权路径长计算](https://blog.csdn.net/yushupan/article/details/82735773)

哈夫曼树:左右中，后序。权重路径为value\*depth

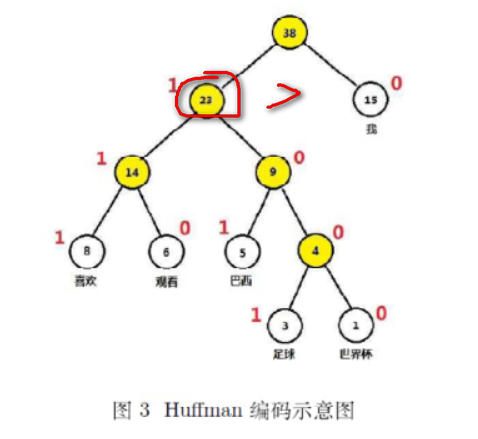
哈夫曼编码为左子树为0，右子树为1，左字树值小于右子树

哈夫曼编码：

从哈夫曼树根节点开始，对左子树分配代码“0”，对右子树分配“1”，一直到达叶子节点



word2vec哈夫曼树，将权重大的左子树值为1，右子树位0



10.相比于LSTM,Transformer的优缺点是啥？在中文上bert有啥不够完善的地方，该如何解决？

# [RNN结构，双向LSTM，Transformer， BERT对比分析](https://blog.csdn.net/weixin_41089007/article/details/96474760)

11.bagging 和boosting 的区别(<https://www.cnblogs.com/earendil/p/8872001.html>、<https://blog.csdn.net/cy_tec/article/details/80632280>)

# Bagging和Boosting的区别：

# variance（方差）

1）样本选择上：

Bagging：训练集是在原始集中有放回选取的，从原始集中选出的各轮训练集之间是独立的。

Boosting：每一轮的训练集不变，只是训练集中每个样例在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整。

2）样例权重：

Bagging：使用均匀取样，每个样例的权重相等

Boosting：根据错误率不断调整样例的权值，错误率越大则权重越大。

3）预测函数：

Bagging：所有预测函数的权重相等。

Boosting：每个弱分类器都有相应的权重，对于分类误差小的分类器会有更大的权重。

4）并行计算：

Bagging：各个预测函数可以并行生成

Boosting：各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。

12.模型过拟合有哪些解决方案，如何选择L1或者L2正则化？

正则化目的是为了约束参数

L1:参数稀疏

L2:更平滑

　　1.L1是模型各个参数的绝对值之和。

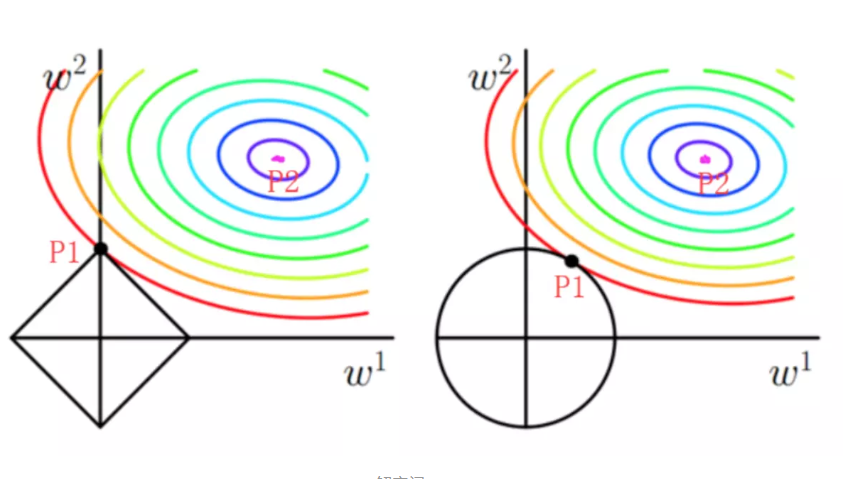
　　　L2是模型各个参数的平方和的开方值。

　　2.L1会趋向于产生少量的特征，而其他的特征都是0.

　　　 因为最优的参数值很大概率出现在坐标轴上，这样就会导致某一维的权重为0 ，产生稀疏权重矩阵

　　   L2会选择更多的特征，这些特征都会接近于0。

          最优的参数值很小概率出现在坐标轴上，因此每一维的参数都不会是0。当最小化||w||时，就会使每一项趋近于0



[解决过拟合问题通常的方法](https://www.cnblogs.com/tsruixi/p/10693101.html)如下：  
1.数据集增强  
2.Drop out

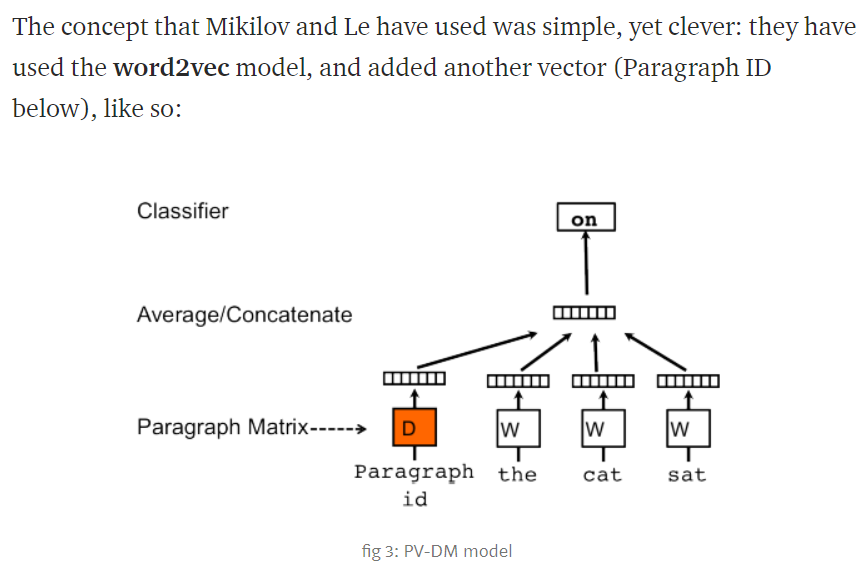
3.Early stopping  
4.正则化：保留所有的特征，但尽可能使参数θjθj尽量小。

13.什么是残差网络，它解决了哪些问题？

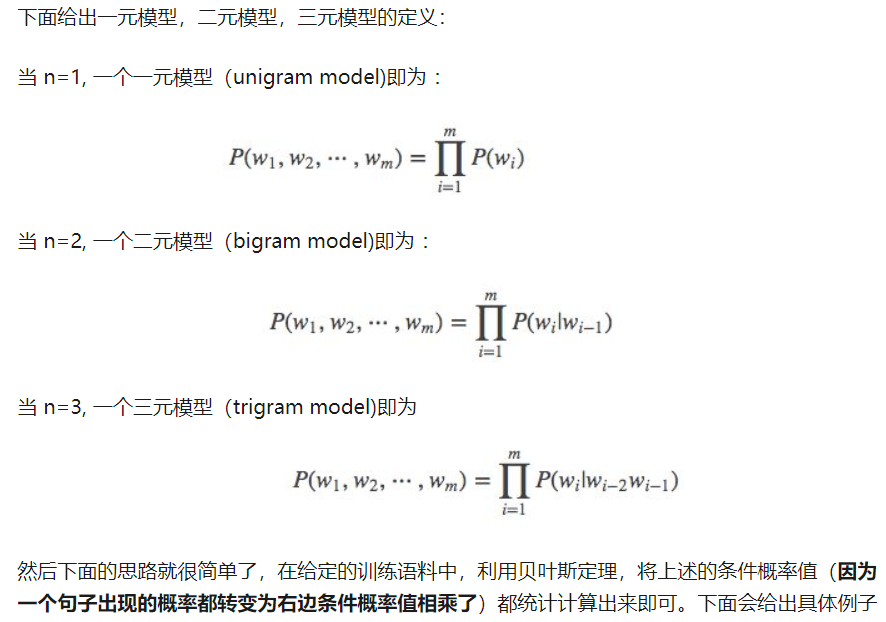
一方面: ResNet解决的不是梯度弥散或爆炸问题，kaiming的论文中也说了:臭名昭著的梯度弥散/爆炸问题已经很大程度上被normalized initialization and intermediate normalization layers解决了;

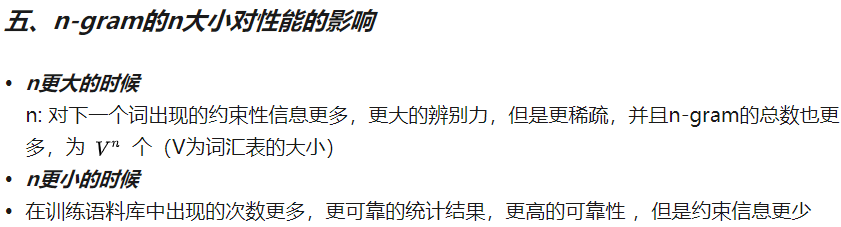
另一方面: 由于直接增加网络深度的(plain)网络在训练集上会有更高的错误率，所以更深的网络并没有过拟合，也就是说更深的网络效果不好，是因为网络没有被训练好。

14.Doc2Vec模型简介



14.N-gram 在n=2 和n=3 的公式





15贝叶斯公式，朴素贝叶斯？

16.分词用哪些方法？

# **[中文分词方法](https://blog.csdn.net/qq_34832393/article/details/89947290)**

**中文分词：**

     在语言理解中，词是最小的能够独立活动的有意义的语言成分，在处理中文文本时，需要进行分词处理，将句子转化为词的表示。这个切词处理的过程就是中文分词，它通过计算机自动识别出句子的词，在词间加入边界标记符，分割出各个词汇。

**常见的分词方法：**

**（一） 规则分词：**

      1.1  正向最大匹配法（Maximum Match method，MM法）

      1.2  逆向最大匹配法（Reverse Maximum Match method，MM法）

      1.3  双向最大匹配法（Bi-direction Match method）

<https://blog.csdn.net/chengzheng_hit/article/details/54752673>

**（二） 统计分词：**

      2.1  语言模型(bigram、trigram,n-gram)

      2.2  隐马尔科夫模型（HMM）

      2.3  条件随机场（CRF）

      2.4  神经网络分词算法

17.随机梯度下降，随机体现在哪？

随机一个样本更新参数

18.随机森林，随机体现在哪？

样本随机，特征随机

19.python几种数据格式

八种数据结构：number,string,None,Boolean;list,truple,dict,set

~~字典、字符串、列表、字典、数组~~

20.列表与元组的区别

一个是可变一个是不可变：列表长度可变，元祖长度不变

21.列表去重如何实现

set元组去重，然后list

22.字典通过key排序

a = sorted(d.items(), key=lambda x: x[1], reverse=True)

23.[]{}()正则表达式中的区别 匹配邮箱 匹配手机

24.python多进程与多线程

25.hmm与crf区别(hmm两个状态 显状态和隐藏状态 crf一个转移矩阵)

26.[RNN和LSTM的区别](https://blog.csdn.net/lanmengyiyu/article/details/79941486)

27.LDA主题模型

28.经常使用的优化函数SGD 和Adam

29.SVM的优化函数 <https://www.nowcoder.com/questionTerminal/58a0edc053644a8cbeb4e504d9f28a8b>

30.LR的优化函数 和损失函数时 <https://www.jianshu.com/p/fa411ffb5490cxa> <https://www.jianshu.com/p/dd8ee2a38dbc> <https://www.jianshu.com/p/dd8ee2a38dbc>

31.CRF介绍下

32.正则表达式用过吗

33.你用过的最大前向匹配和反向匹配是多少

35.理解Bias（偏差），Error（误差），和Variance（方差）的区别和联系？(http://baijiahao.baidu.com/s?id=1601092478839269810&wfr=spider&for=pc)

Bias反映的是模型在样本上的输出与真实值之间的误差，即模型本身的精准度

Variance（方差）反映的是模型每一次输出结果与模型输出期望之间的误差，即模型的稳定性。

所以我们可以得出结论：当模型复杂度低时，Variance更低，当模型复杂度高时，Variance更高。

36.优化算法:SGD 和Adam 、Adagrade([https://www.cnblogs.com/jayechan/p/9511486.html](https://www.cnblogs.com/ljygoodgoodstudydaydayup/p/7294671.html))

目的是让梯度下降更快更准更平稳的收敛

37.一阶和二阶优化算法

深度学习优化学习方法总结 （一阶为主）https://blog.csdn.net/sunflower\_sara/article/details/81321886

常用的优化算法：梯度下降法，牛顿法，拟牛顿法,共轭梯度法 （二阶为主） https://blog.csdn.net/sunflower\_sara/article/details/81215135

一阶方法：

随机梯度下降（Stochastic Gradient Descent，SGD）

动量（SGD with Momentum，SGDM）

牛顿动量法（Nesterov Acceleration Gradient，NAG）

AdaGrad（自适应梯度）

RMSProp（均方差传播）

Adam

Nadam

二阶方法：

牛顿法

拟牛顿法

共轭梯度法 CG

BFGS

L-BFGS

38[判别模型和生成模型](https://blog.csdn.net/u010358304/article/details/79748153)

判别模型: KNN,LR,SVM,树模型，神经网络

生成模型:贝叶斯,GMM,HMM,LDA

39.装饰器

import time

def wrapper(func): # 装饰器

def inner(\*args, \*\*kwargs):

'''函数执行之前的内容扩展'''

ret = func(\*args, \*\*kwargs)

'''函数执行之前的内容扩展'''

return ret

return inner

40Viterbi算法:解释下

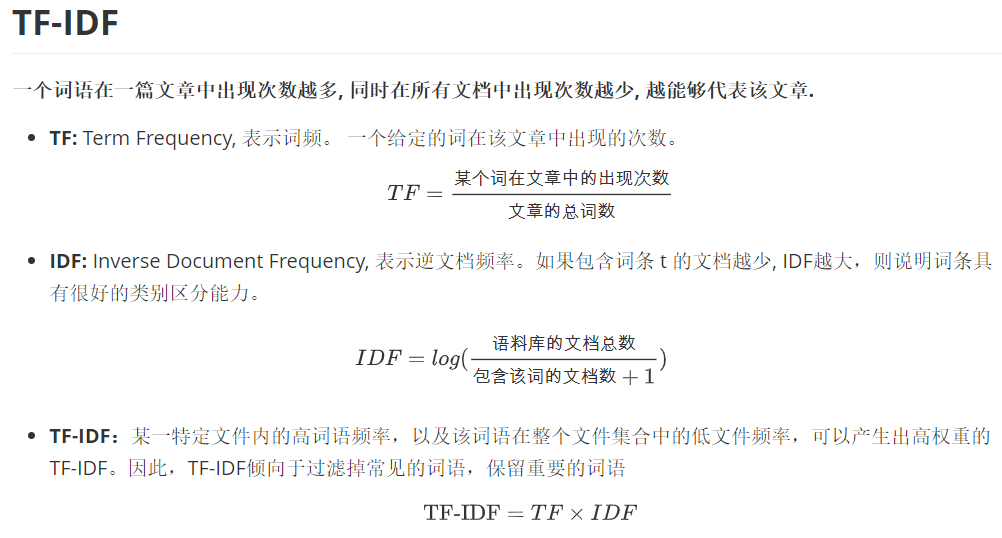
动态对方法中的一种https://blog.csdn.net/qq\_42189083/article/details/89350890

即维特比算法。是一种动态规划算法用于最可能产生观测时间序列的-维特比路径-隐含状态序列，特别是在马尔可夫信息源上下文、隐马尔科夫模型、条件随机场中

41.softmax函数

<https://www.cnblogs.com/alexanderkun/p/8098781.html>

42.TF-IDF

43.优化算法详细的解析

<https://blog.csdn.net/u012328159/article/details/80311892>

44.

解决这些不同的任务，从模型角度来讲什么最重要？是特征抽取器的能力。尤其是深度学习流行开来后，这一点更凸显出来。因为深度学习最大的优点是“端到端（end to end）”，当然这里不是指的从客户端到云端，意思是以前研发人员得考虑设计抽取哪些特征，而端到端时代后，这些你完全不用管，把原始输入扔给好的特征抽取器，它自己会把有用的特征抽取出来。

为了解决这个问题，后来引入了LSTM和GRU模型，通过增加中间状态信息直接向后传播，以此缓解梯度消失问题，获得了很好的效果，于是很快LSTM和GRU成为RNN的标准模型。

上述内容简单介绍了RNN在NLP领域的大致技术演进过程。那么为什么RNN能够这么快在NLP流行并且占据了主导地位呢？主要原因还是因为RNN的结构天然适配解决NLP的问题，NLP的输入往往是个不定长的线性序列句子，而RNN本身结构就是个可以接纳不定长输入的由前向后进行信息线性传导的网络结构，而在LSTM引入三个门后，对于捕获长距离特征也是非常有效的。所以RNN特别适合NLP这种线形序列应用场景，这是RNN为何在NLP界如此流行的根本原因。

RNN本身的序列依赖结构对于大规模并行计算来说相当之不友好。

cnn:

1.网络做深

2.Max Pooling层

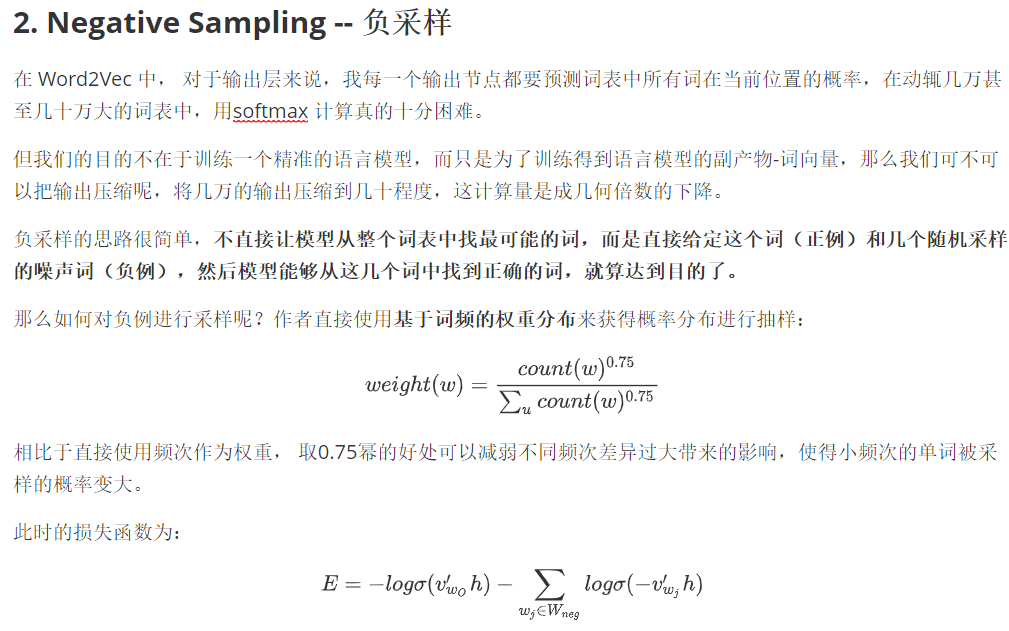
专门在输入部分给每个单词增加一个position embedding，将单词的position embedding和词向量embedding叠加起来形成单词输入，这样也可以，也是常规做法。

以，Transformer不像RNN或CNN，必须明确的在输入端将Positon信息编码，Transformer是用位置函数来进行位置编码的，而Bert等模型则给每个单词一个Position embedding，将单词embedding和单词对应的position embedding加起来形成单词的输入embedding，类似上文讲的ConvS2S的做法

而关于NLP句子中长距离依赖特征的问题，Self attention天然就能解决这个问题，因为在集成信息的时候，当前单词和句子中任意单词都发生了联系，所以一步到位就把这个事情做掉了。不像RNN需要通过隐层节点序列往后传，也不像CNN需要通过增加网络深度来捕获远距离特征，Transformer在这点上明显方案是相对简单直观的。

45.word2vec优化：分层softmax和负采样

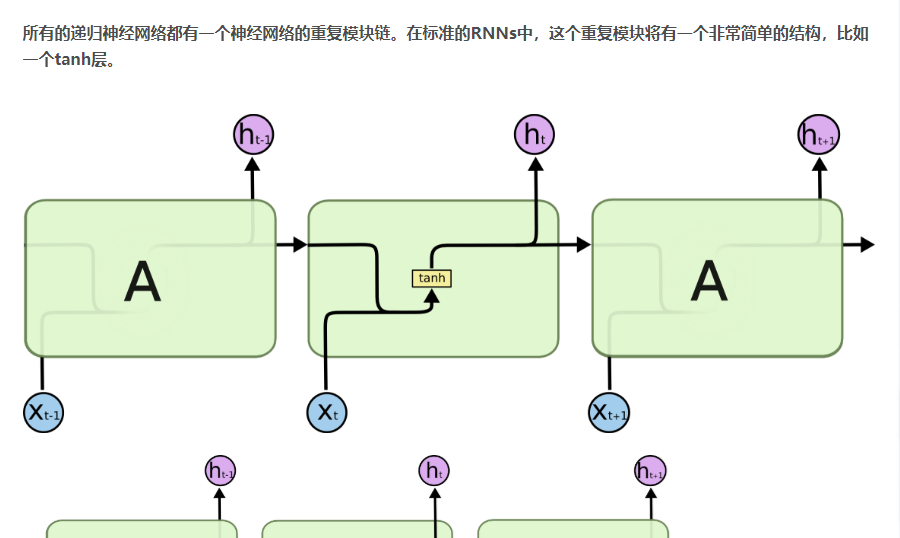
1. 分层softmax
2. 负采样



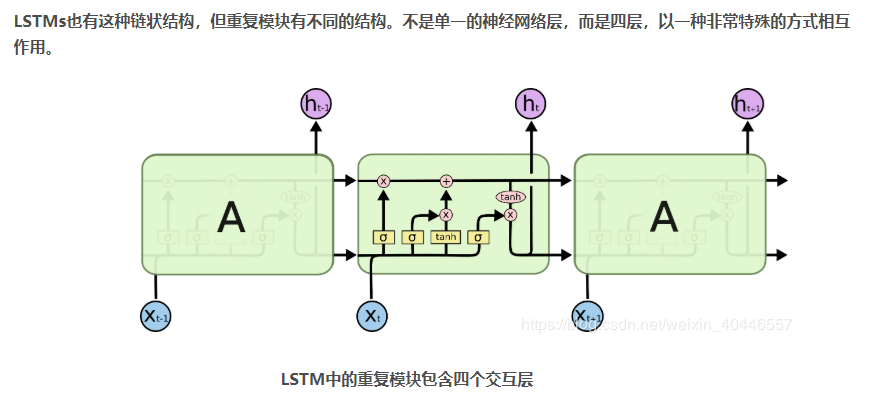
46.查全率与查准率

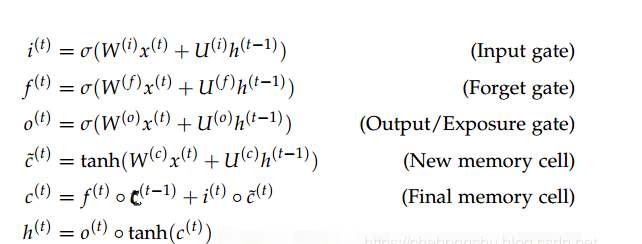
<https://blog.csdn.net/program_developer/article/details/79937291>

47.RNNS和LSTM

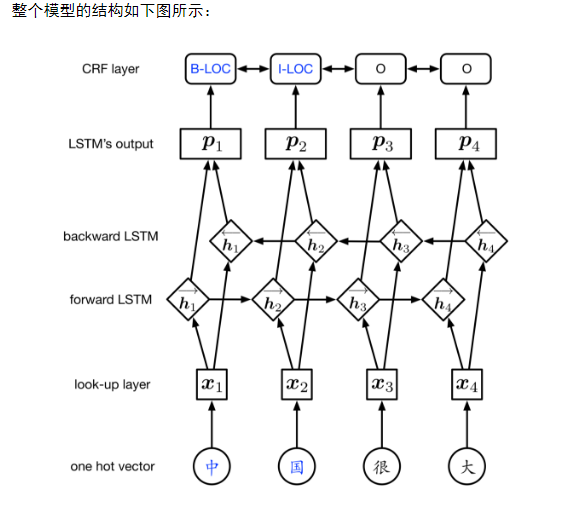


LSTM





48.bilstm-ner的模型结构



49.nlp语言模型

<https://blog.csdn.net/miner_zhu/article/details/82729192>

1. 平滑化

我们在传统统计语言模型提出了两个问题：自由参数数目和数据稀疏，上述N-gram只是解决了第一个问题，而平滑化就是为了解决第二个问题。

假设有一个词组在训练语料中没有出现过，那么它的频次就为0，但实际上能不能认为它出现的概率为0呢？显然不可以，我们无法保证训练语料的完备性。那么，解决的方法是什么？如果我们默认每一个词组都出现1次呢，无论词组出现的频次是多少，都往上加1，这就能够解决概率为0的问题了。

上述的方法就是加1平滑，也称为拉普拉斯平滑。平滑化还有许多方法，这里就不展开介绍了：

（1）加法平滑

（2）古德-图灵平滑

（3）K平滑

50.textcnn

<https://www.jianshu.com/p/f69e8a306862>

51维特比算法

<https://blog.csdn.net/athemeroy/article/details/79339546>

52.简单介绍下tensorflow计算图

<https://cloud.tencent.com/developer/article/1357335>

TensorFlow哲学

对定义计算和计算的执行做了分离。

tensorflow的编程和以往接触的编程方式有很大差异。以前的编程，无论是编译类型语言还是动态解释型语言，变量计算后，就会得到结果，比如c=a+b，当执行完语句后，就会得到c的值。

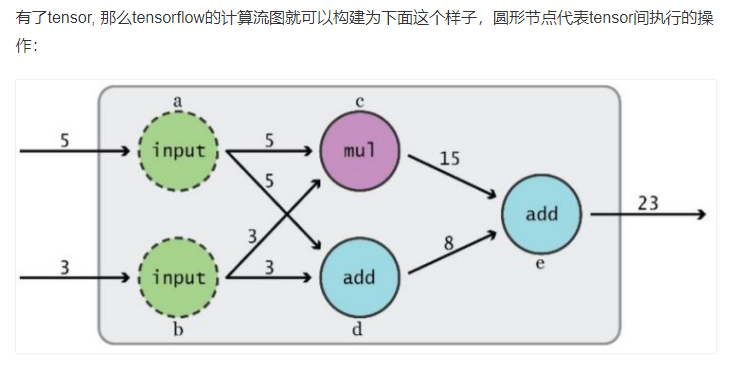
但， tensorflow不是！

首先看一下，tensor是什么？它是一个 n 维数组：

0-d tensor: scalar （标量）

1-d tensor: vector （向量）

2-d tensor: matrix（矩阵）





这就验证了之前所说，计算图和执行计算图做了严格的分离，所以此时打印得不到a值。

如何得到a值？

创建session，并在当前的sess中执行构建的图得到a的值。

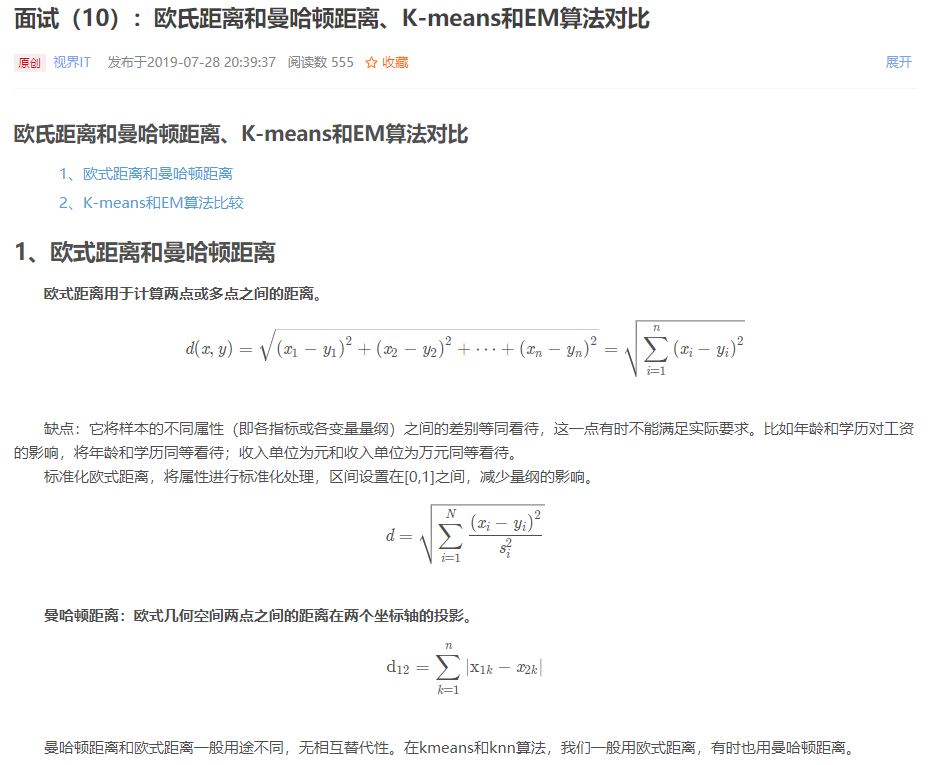
1with tf.Session() as sess:

2 print(sess.run(a))

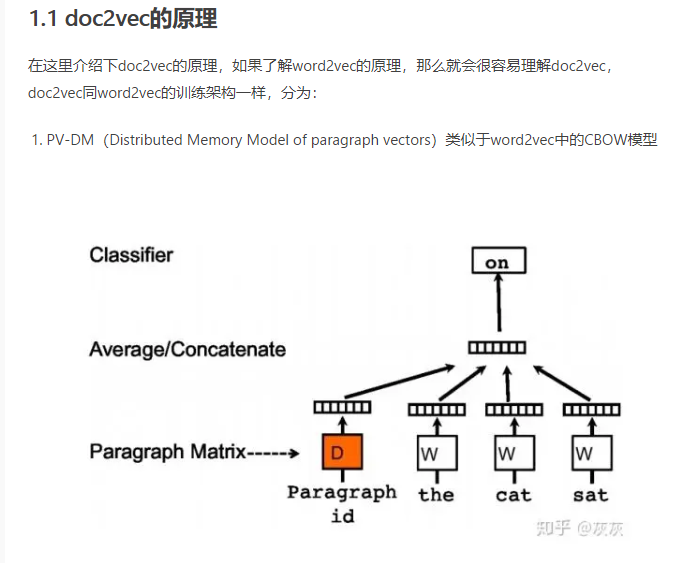
总结

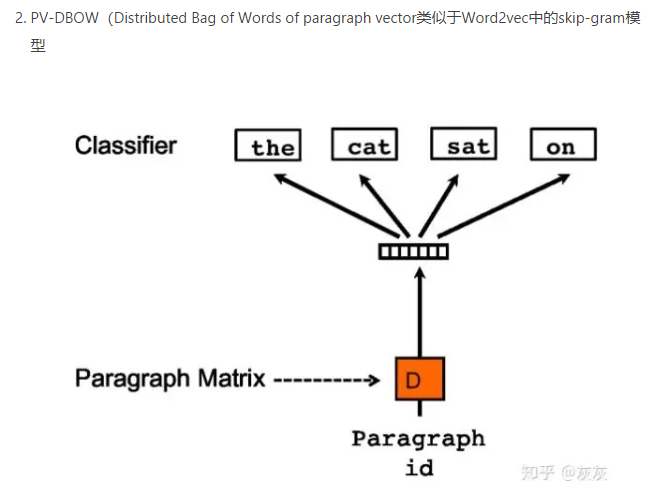
Session对象封装了tensorflow的执行环境。记住：计算图只是定义了operations，operations只有在session里面执行才能真正得到计算图的结果。

53.k-means or knn中通常用欧式距离来介绍最近邻之间的距离，简述Manhattman距离和欧式距离之间区别.

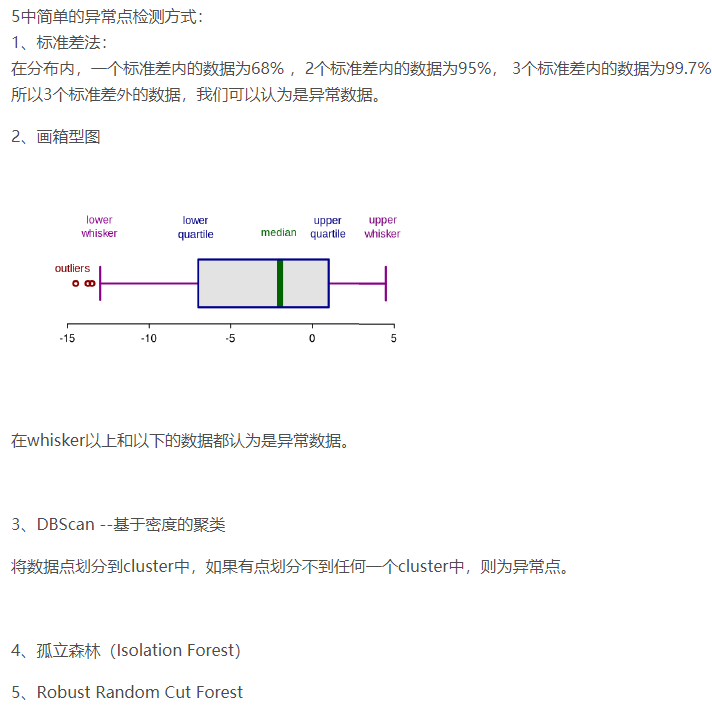


54 [句向量的表示方式](https://www.jianshu.com/p/d6f4ab115465)





55.[常见5种异常检测(Anomaly Detection)的方法](https://blog.csdn.net/u012073033/article/details/92782863)



56.词嵌入时的OOV(out of vacabulary)问题

<https://www.zhihu.com/question/308543084>

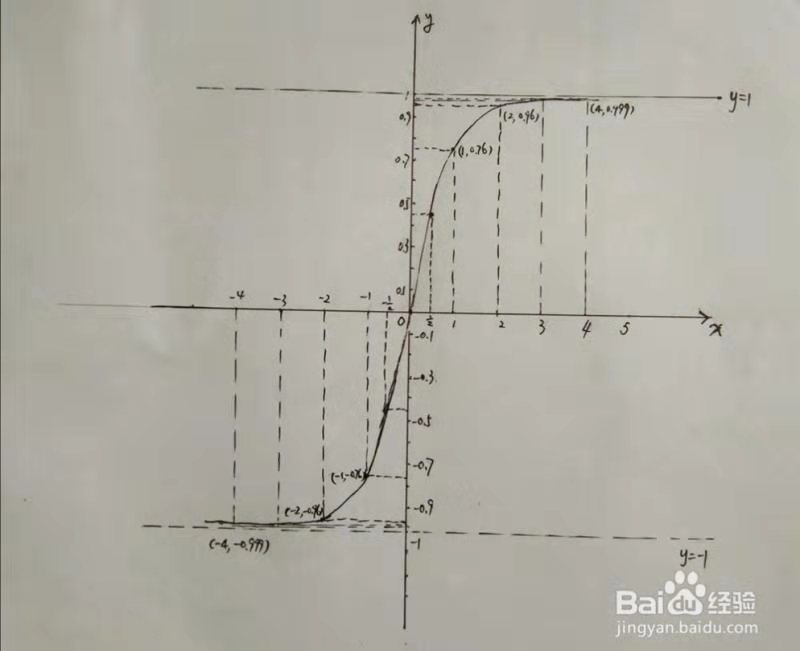
未登录词就是训练时未出现，测试时出现了的单词。在自然语言处理或者文本处理的时候，我们通常会有一个字词库（vocabulary）。这个vocabulary要么是提前加载的，或者是自己定义的，或者是从当前数据集提取的。假设之后你有了另一个的数据集，这个数据集中有一些词并不在你现有的vocabulary里，我们就说这些词汇是Out-of-vocabulary，简称OOV。

Instead of assigning all the Out of Vocabulary tokens to a common UNK vector (zeros), it is better to assign them a unique random vector. At-least this way when you find the similarity between them with any other word, each of them will be unique and the model can learn something out of it. In the UNK case, they will all be same and so all the UNK words will be treated as having the same context.

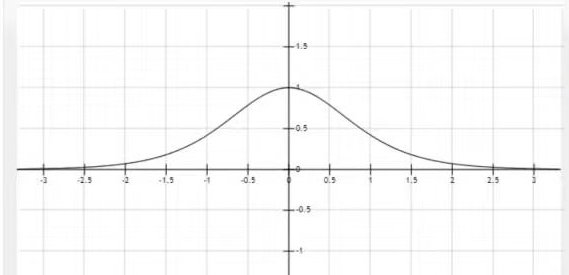
I tried this approach and got a 3% accuracy improvement on the Quora Duplicate question pair detection dataset using an LSTM model.

57.

Tanh激活函数曲线，导数曲线。



单调递增，值域为(-1,1),单调递增



经过(0,1)偶函数。

58介绍下CRF

<https://blog.csdn.net/xueyingxue001/article/details/51498902>

59.常用的调参方法

<https://zhuanlan.zhihu.com/p/81956439>

1.试错法

2.网格搜索法

3.二分搜过法

4.随机搜索法

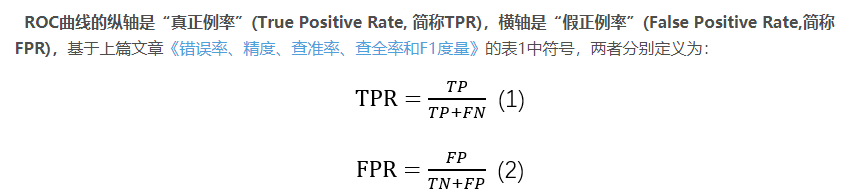
5.贝叶斯优化

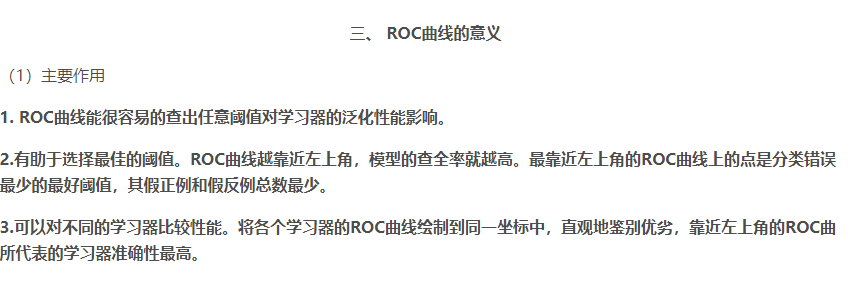
60.[ROC](十亿像素级动态大场景数据集 PANDA)

Precison（查准率）：预测为正例的样本中真正正例的比例。

Recall（召回率）：真正为正例的样本有多少被预测出来。

****ROC全称是“受试者工作特征”(Receiver OperatingCharacteristic)曲线****

********

********

****AUC就是ROC曲线下的面积，衡量学习器优劣的一种性能指标****

****AUC是衡量二分类模型优劣的一种评价指标，表示预测的正例排在负例前面的概率。****

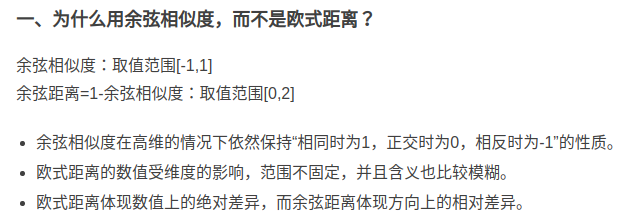
ROC曲线和P-R曲线的区别:

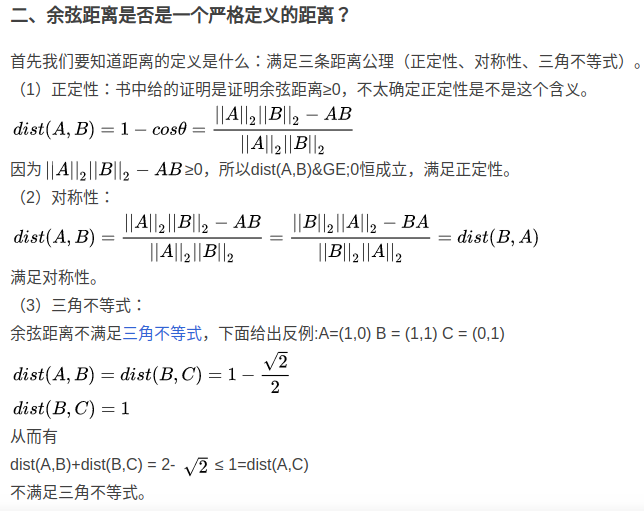
当正负样本的分布发生变化时，ROC曲线的形状基本保持不变，而P-R曲线的形状会发生剧烈变化。反应ROC曲线能客观的反应模型本身的性能。尽量降低不同测试集带来的干扰。

选择哪种曲线因实际问题而已，如果希望看到特点数据集的表现，P-R曲线更直观反应其性能。

61.余弦距离

# 为什么用余弦相似度，而不是欧式距离？





62.neo4j操作语法

<https://blog.csdn.net/xingxiupaioxue/article/details/71747284>

63.HMM和CRF区别，bi-LSTM+CRF相比于bi-LSTM+HMM的效果和优势?

# [HMM ,MEMM,CRF 优缺点与区别](https://blog.csdn.net/losteng/article/details/51037927)

已知输入x,类别标签y:

    判别式模型估计条件概率分布P(y|x),常见有：LR，SVM，NN,KNN,CRF,LDA,线性回归

    产生式模型估计联合概率分布P(x,y), 常见有：NB，HMM

1. 生成式模型or 判别式模型

HMM模型对转移概率和表现概率直接建模，统计共同出现的概率式模型。

MEMM不是一种生成式模型，它是一种基于下状态分类的有限状态模型。

CRF模型对条件概率 P(m|o) 建模，就是判别模型。其基本思想是有限样本条件下建立判别函数，不考虑样本的产生模型，直接研究预测模型。

1. 拓扑结构

HMM和MEMM是一种有向图，CRF是一种无向图

1. 全局最优or局部最优

**HMM对转移概率和表现概率直接建模，统计共现概率。**

**MEMM是对转移概率和表现概率建立联合概率，统计时统计的是条件概率，由于其只在局部做归一化，所以容易陷入局部最优。**

**CRF是在全局范围内统计归一化的概率，而不像是MEMM在局部统计归一化概率。是全局最优的解。解决了MEMM中标注偏置的问题。**

1. CRF与MEMM和HMM的优缺点比较

**优点：**

**a）与HMM比较。CRF没有HMM那样严格的独立性假设条件，因而可以容纳任意的上下文信息。特征设计灵活）**

**b）与MEMM比较。由于CRF计算全局最优输出节点的条件概率，它还克服了最大熵马尔可夫模型标记偏置（Label-bias）的缺点。**

**c）与ME比较。CRF是在给定需要标记的观察序列的条件下，计算整个标记序列的联合概率分布，而不是在给定当前状态条件下，定义下一个状态的状态分布。**

**缺点：**

**训练代价大、复杂度高**

64.**FastText跟Word2Vec的区别？fasttext相比其他的深度学习模型比如TextCNN速度快的原因。**

* 相似处：

图模型结构很像，都是采用embedding向量的形式，得到word的隐向量表达。

都采用很多相似的优化方法，比如使用Hierarchical softmax优化训练和预测中的打分速度。

* 不同处：

模型的输出层：word2vec的输出层，对应的是每一个term，计算某term的概率最大；而fasttext的输出层对应的是分类的label。不过不管输出层对应的是什么内容，起对应的vector都不会被保留和使用。

模型的输入层：word2vec的输出层，是 context window 内的term；而fasttext 对应的整个sentence的内容，包括term，也包括 n-gram的内容。

两者本质的不同，体现在 h-softmax的使用：

Word2vec的目的是得到词向量，该词向量 最终是在输入层得到，输出层对应的 h-softmax  
也会生成一系列的向量，但最终都被抛弃，不会使用。

fastText则充分利用了h-softmax的分类功能，遍历分类树的所有叶节点，找到概率最大的label（一个或者N个）

* Fasttext在预测标签时使用非线性激活函数，在中间层不使用非线性激活函数，所以速度很快。

65.简述Attention，写出self-Attention公式表达。

[Attention&self-Attention](https://zkq1314.github.io/2018/10/11/NLP%E4%B8%AD%E7%9A%84attention%E5%92%8Cself-attention/)

66.bi-LSTM+CRF模型分数还有什么改善空间？