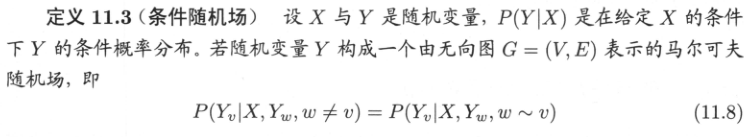
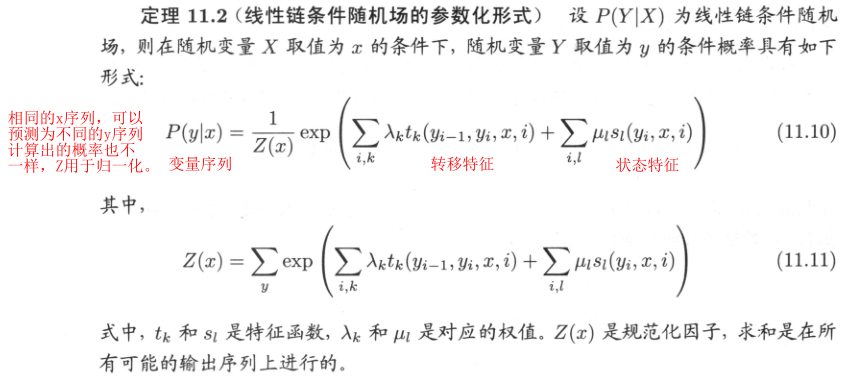
### 11.1.2 概率无向图模型的因子分解

将概率无向图模型的联合概率分布表示为最大团上的随机变量函数乘积形式的操作，称为概率无向图模型的因子分解。  
11.2 条件随机场的定义



### **11.2.2 条件随机场的参数化形式**



理解：求已知观测序列x的条件下，状态(标记)序列为y的概率：

1、主题部分为特征函数值的计算

2、以转移特征为例，在每一个时间步，可以计算k个特征值

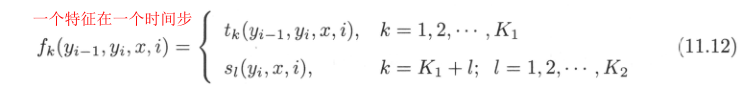
3、对特征值加权求和，在求其指数并归一化即为该条件概率

### 11.2.3 向量形式(时间维求和)

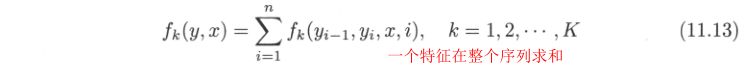
将参数形式的权重和特征函数统一符号描述，为此有：

1、假设K1个转移特征，K2个状态特征，令K=K1+K2

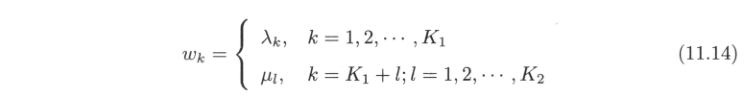
2、统一的特征描述为：



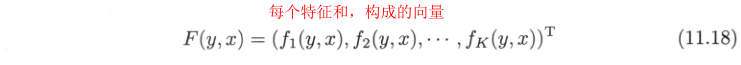
3、一个特征在所有的时间步的斩获(数量值)



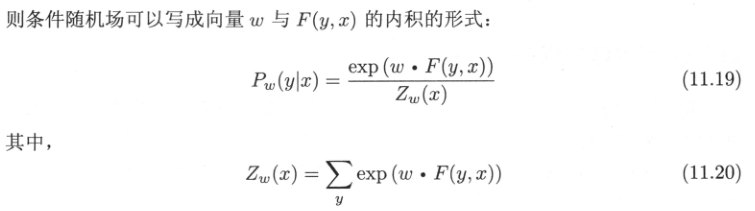
4、权值统一化：



5、特征向量(每个元素，表示一个特征在整个序列上的斩获)



6、内积形式

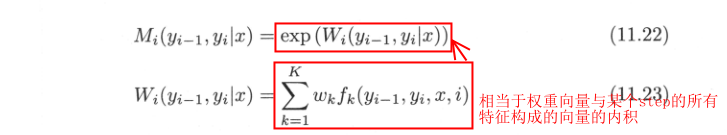


参数形式：求和

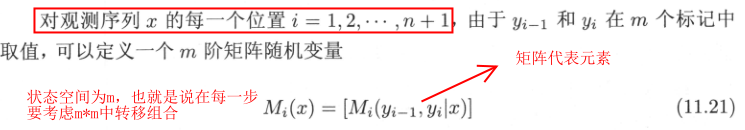
内积形式：内积(等价于求和)

### 11.2.4 矩阵形式(特征维求和)

1、单步特征和：单个时间步基于所有特征的斩获



2、单步矩阵：



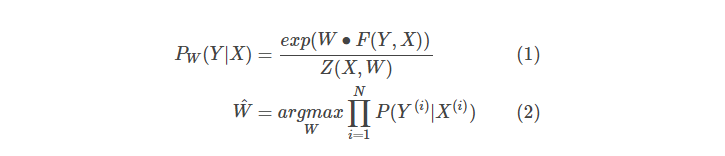
为什么是m阶呢？因为考虑不同的标记组合带入到M(x)中。基于此可以求所有y序列的条件概率

## 11.4 参数学习算法

input: (xi, yi) N对，最大似然函数

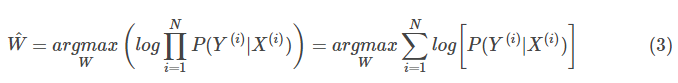
output: 参数λ(特征权重),η(状态权重)

向量形式：

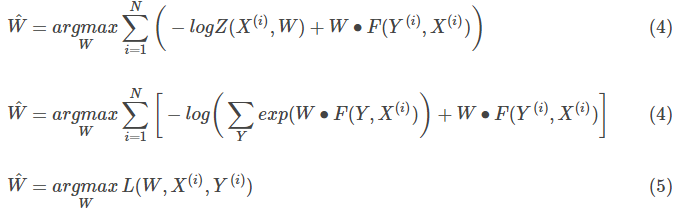


其中，F为K(特征总数)维特征向量，W为K为权重，目标就是最大似然。

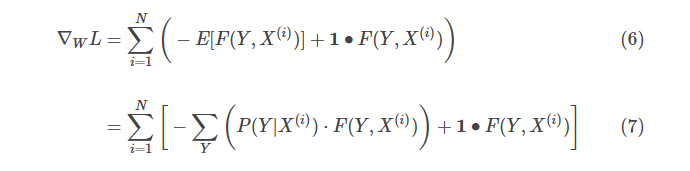
1、对数似然为：



2、带入条件概率，并将归一化因子展开为：



3、梯度计算



由于F为特征函数，第二项可以直接求解。现在考虑第一项z(x)：

将特征按照时间步展开：注意此时f为单步K维特征向量

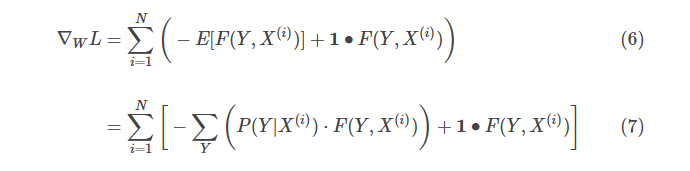
将时间步前提：

将Y拆分：

将pdf通过"积分"化为"边缘概率密度"

由于p可以由前后向算法计算得出，f可以直接计算，至此已经没有变量。

综上：



其中：F(Y,X)=∑tf(yt-1, yt, Xi)

参数更新：

W=step\*

注：求充分统计量的均值用到了：log-partition Function,具体理论来源于指数族分布相关知识。

参考1：<https://www.bilibili.com/video/BV19t411R7QU>

参考：2 <https://anxiang1836.github.io/2019/11/05/NLP_From_HMM_to_CRF/#%E7%BA%BF%E6%80%A7%E9%93%BE%E6%9D%A1%E4%BB%B6%E9%9A%8F%E6%9C%BA%E5%9C%BA-Linear-CRF>

实践文档：

2.4 真实路径得分

从Bi-lstm中可以得到每一个step标记为某个label的分值，然后相加。就是真实路径的发射得分。

从转移矩阵中可以查询出响应的转移分数。

2.5 所有可能序列的得分

归结为对



的计算，其中s就是序列的发射转移得分。下面举一个例子。

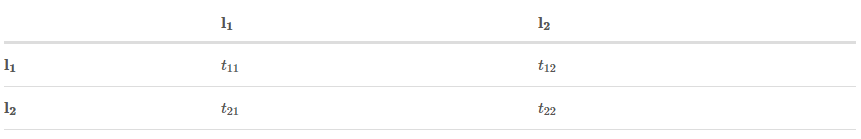
假设我们的观测序列和状态集合为：





lstm输出的发射分数，以及当前学习到的转移概率为：



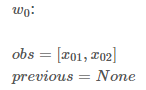


接下来的计算会涉及到两个变量：obs和previous

其中obs：当前step的信息

previous：前一步的计算结果(动态规划计算出的分数)

1、wo step

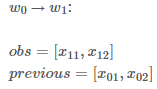


obs是当前step的两个可能状态的发射分数：X01、X02，那么到w0步的总分：



2、w1 step

前向分数：previous



计算到当前step的分数：

假如手写的话就是：

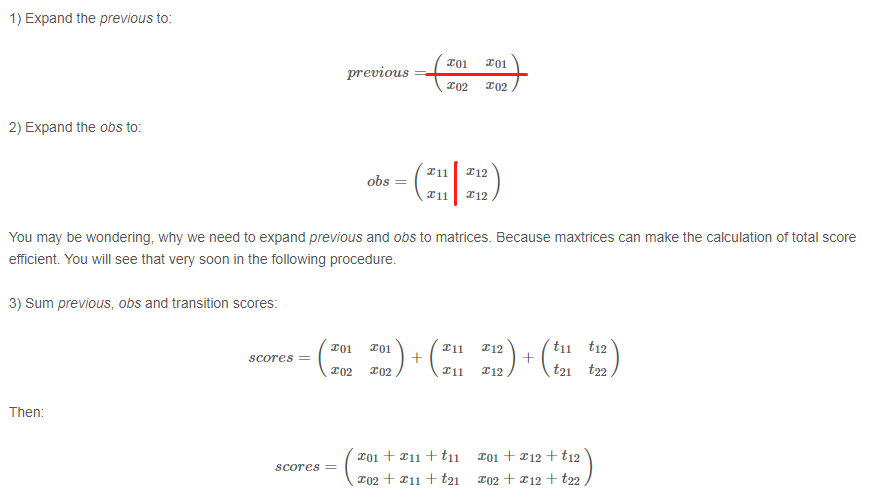
x01+t11+x11 # 前一步 转移 后一步

x01+t12+x12

x02+t21+x11

x02+t22+x12

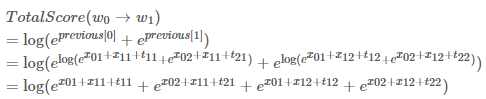
这可以对应处理为矩阵形式如下：

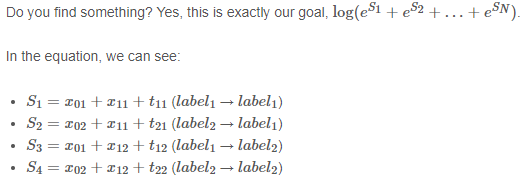


令N为状态集合的长度，则每一个step都要N种可能，T步就是NT。那么到第二步就有4种序列，更新previous为：(以状态1或者状态2结束)



总得分：

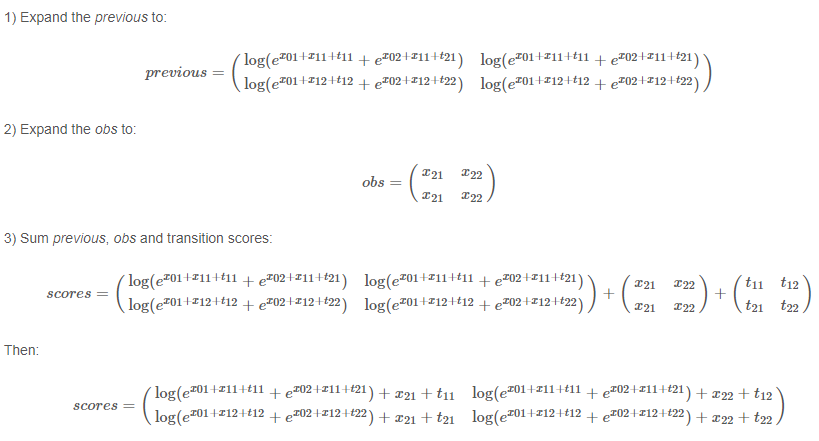


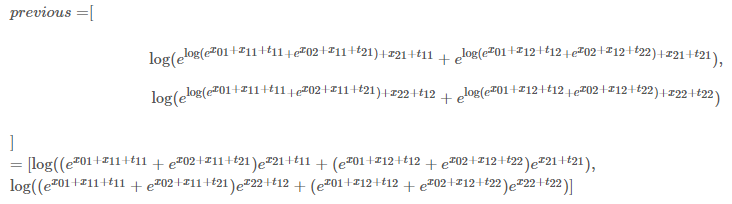


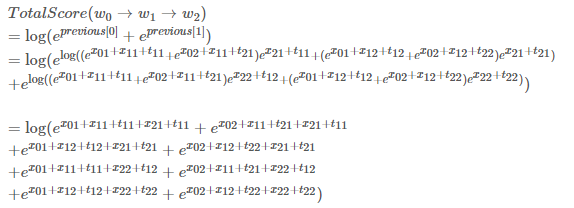
3、 w2 step



前面已经有4个状态序列，所以这一步可以转移为8个状态序列。







所以，哪里体现了动态规划呢？

previous中有两项，分别代表到当前步，以状态1、2结尾的分数。虽然看上去很复杂，但是就是两个float。

所以归纳一下就是：previous expand + 转移分数 + 发射分数

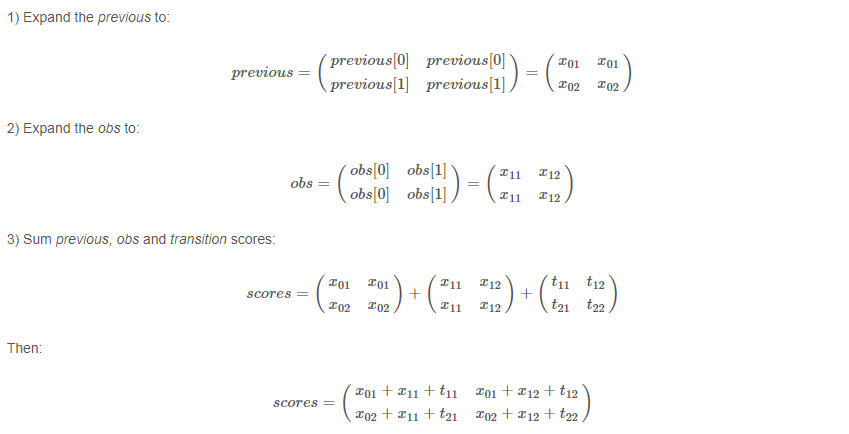
上面算出总分，然后更新当前previous

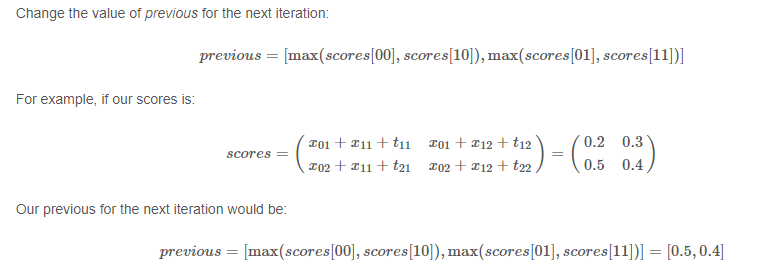
如果是最后一步，可以直接算出总分。

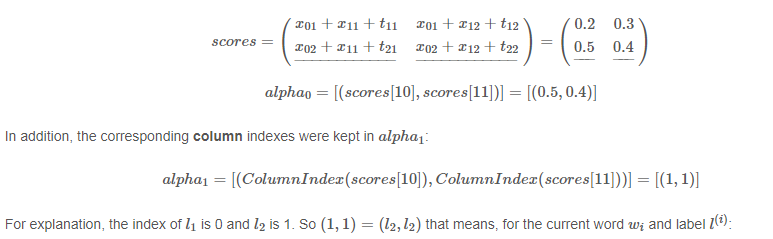
2.6 预测算法-维特比

第一步：因为不涉及到转移等，那么最好的label就是发射分数最高的。

第二步：分数求和贪心，并做记忆score&index(行对应label)：







参考：

<https://createmomo.github.io/2017/11/11/CRF-Layer-on-the-Top-of-BiLSTM-5/>

<https://zhuanlan.zhihu.com/p/97676647>