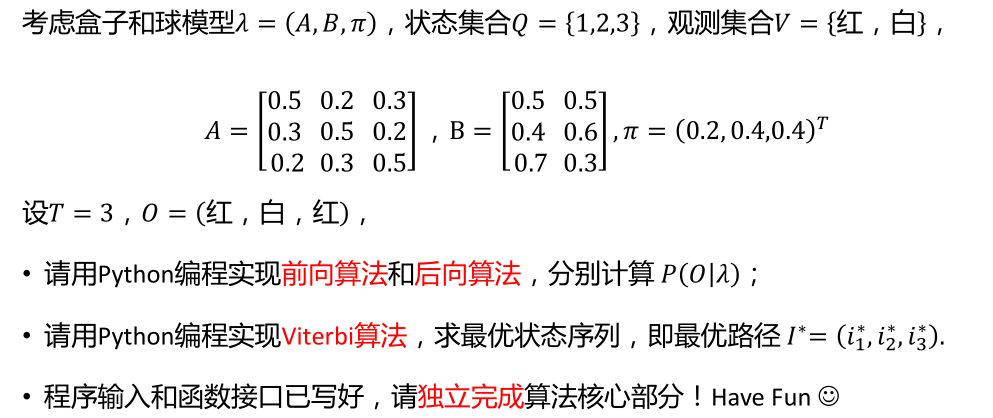
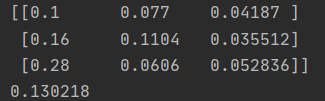
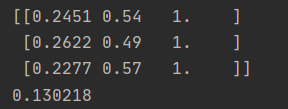
Q：

A：

前向算法

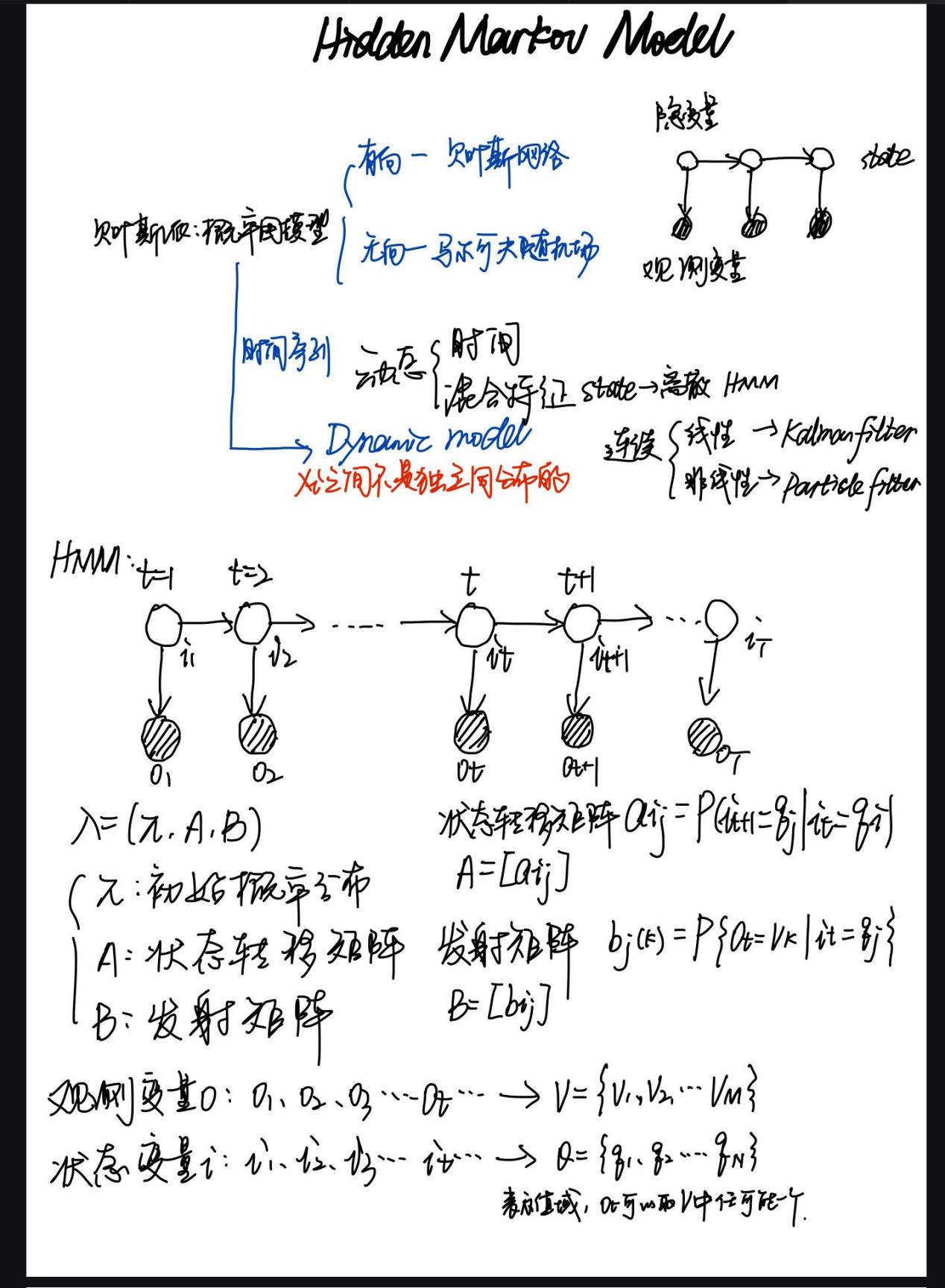


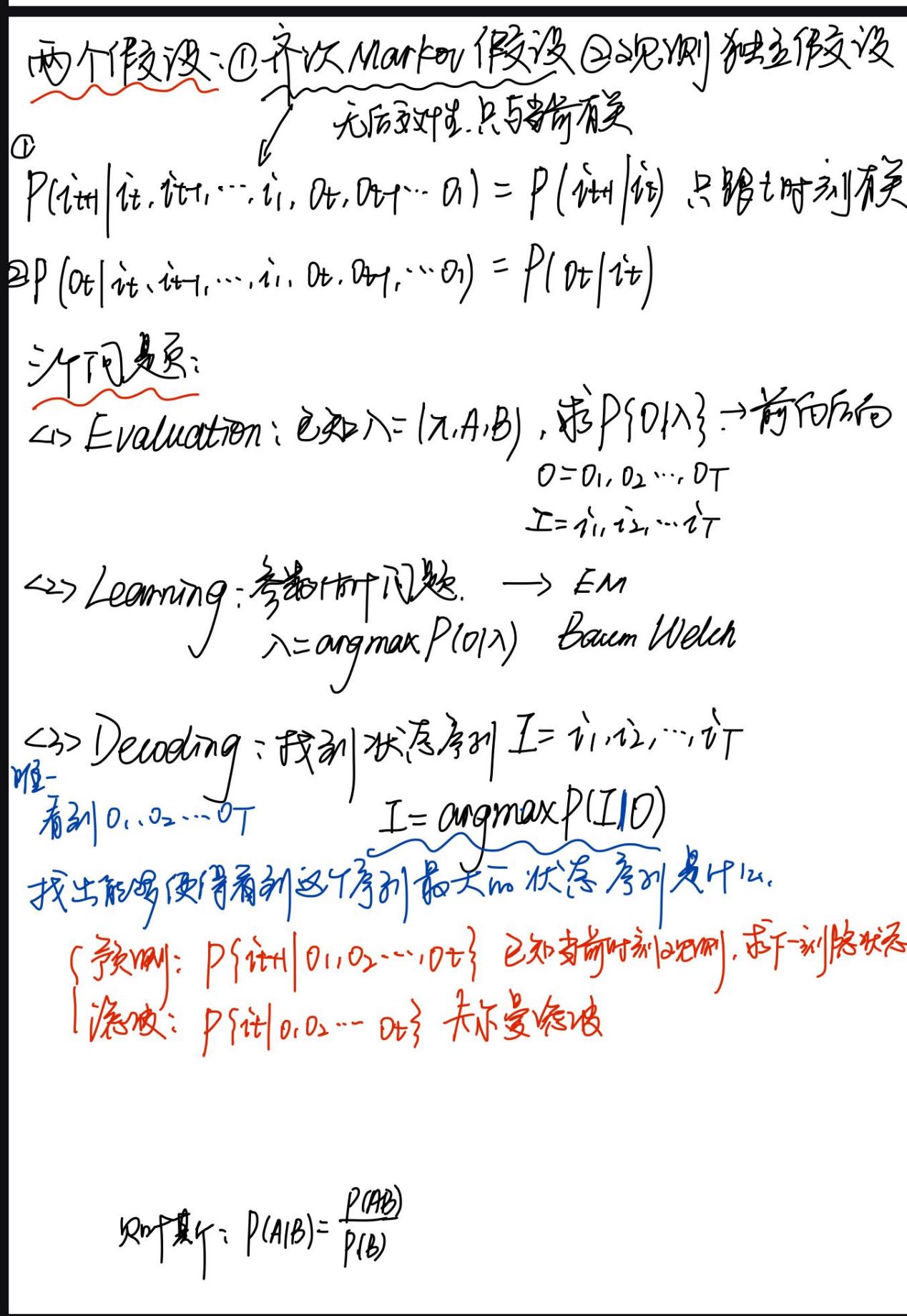
后向算法

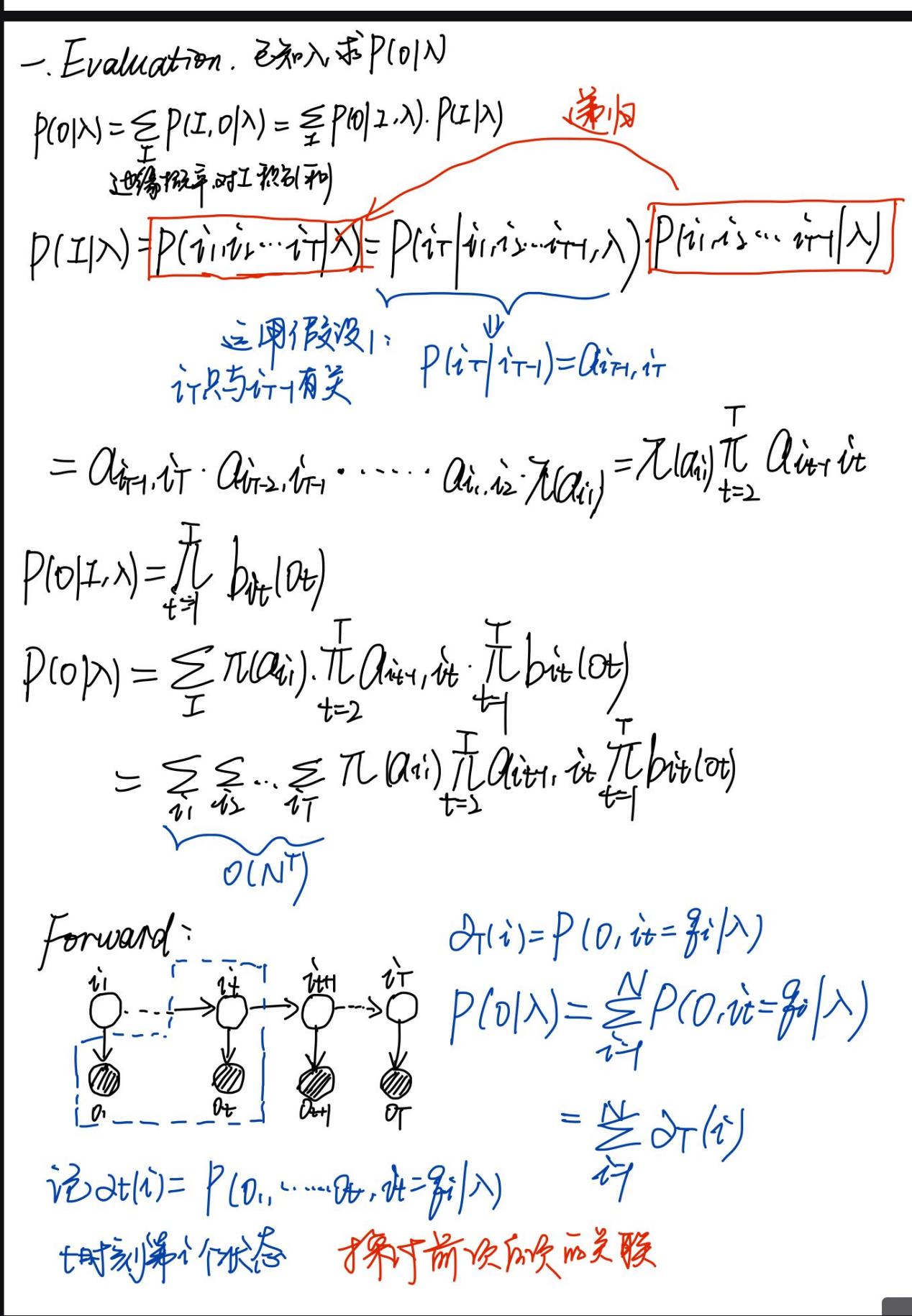


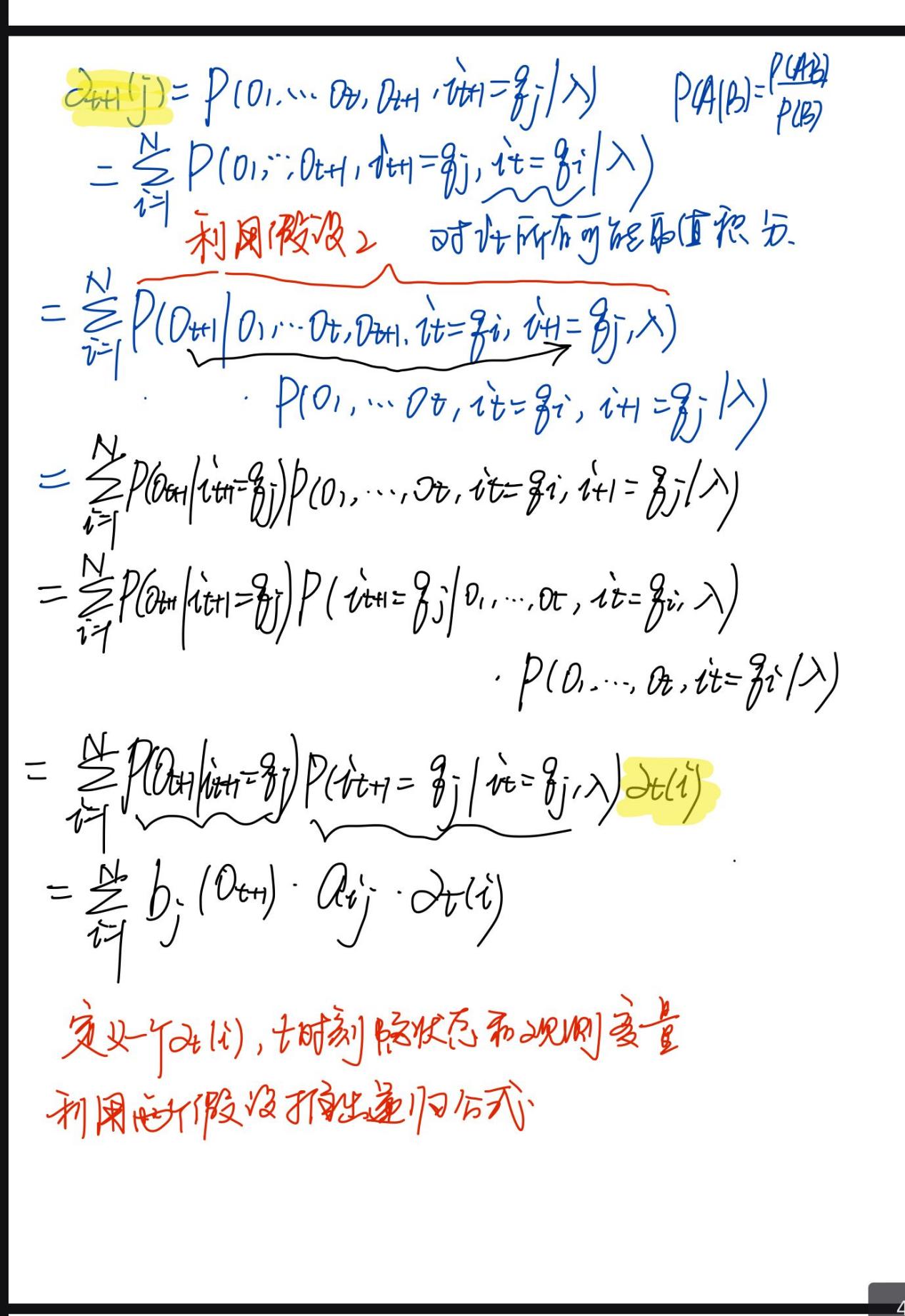
Viterbi解码

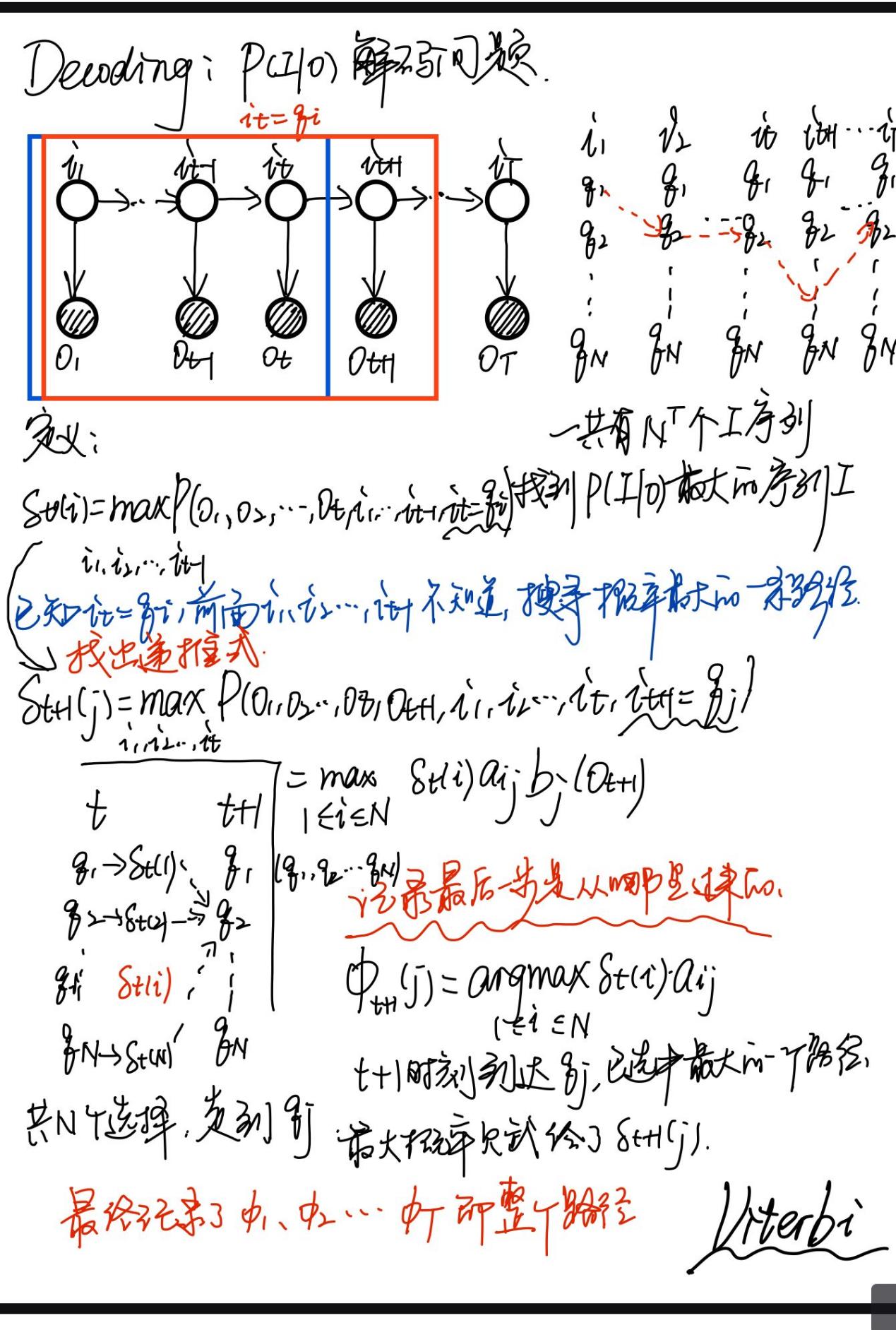
IMG_256











综上所述，Viterbi算法编写步骤如下：

1. 初始化：
2. 迭代计算： 

1. 最后计算最终节点和输出概率:





算法终止时记录的数据便是最佳状态序列

Learning问题

模型训练问题可以定义为：给定一个观察值序列，确定一个,使得最大，用公式表示为：



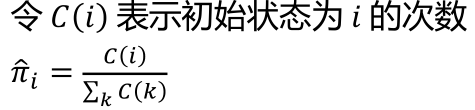
但没有一种方法能直接估计最佳的模型参数。因此需要需求替代的方法，即根据观察值序列选取初始模型，然后求一组新的参数，保证第二次的概率大于第一次的输出概率。重复这个过程，逐步改进模型参数，直到收敛。

一般有俩种学习算法：Viteribi硬对齐和Balum Welch软对齐

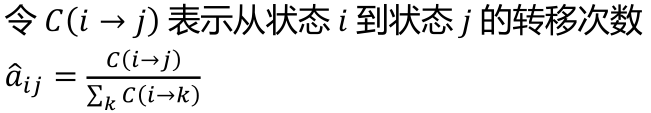
顾名思义：viterbi就是把观测和状态一一对齐了，告诉你这一帧就是那个状态产生的，不是0就是1，然后再去训练模型参数。Balum Welch是一种软对齐的方式，就是说帧和状态的对应关系是以概率分布的。

Viterbi学习算法：

* 通过viterbi解码算法得到状态-观测的对齐序列
* 可通过最大似然估计得到(数数)

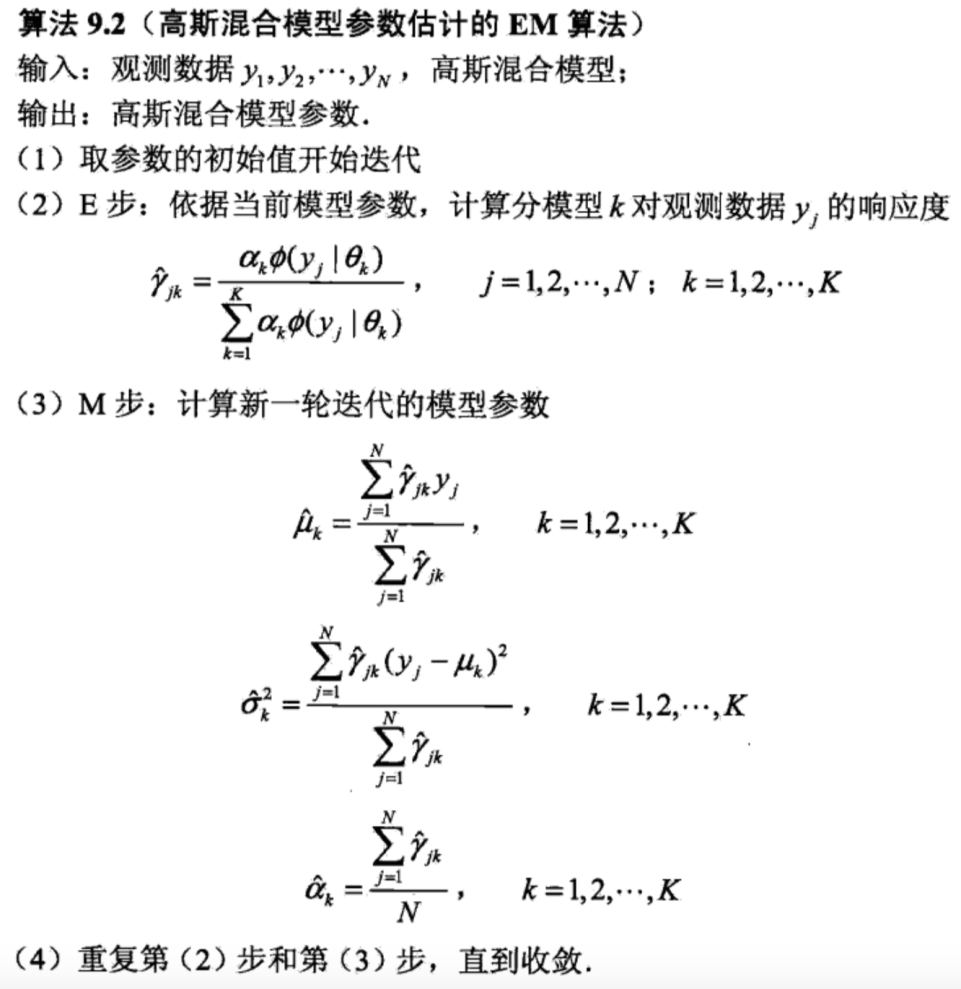


* 也可通过最大似然估计得到(数数)



可得到每个状态j对应的观测集合，而每个状态对应一个GMM，也就得到了每个GMM对应的观测集合。

参数更新公式：

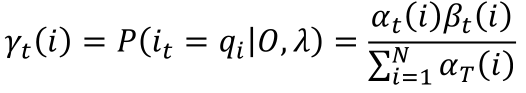


Viterbi学习算法总结：

1. 初始化GMM-HMM参数
2. 基于GMM-HMM参数和Viterbi算法得到状态-观测对齐，得到每个观测对应的隐藏状态
3. 更新参数，包括初始状态概率分布、转移概率、GMM参数
4. 重复2、3步，直至收敛

Baum Welch学习算法

定义状态占用概率：给定模型参数和观测，在时刻t处于状态的概率为为



可以将此概率用于一个迭代算法，学习到参数：EM算法(在hmm中称为Baum-Welch)

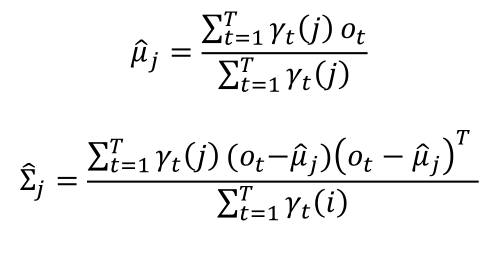
每次迭代分为两步：

①E步：估计状态占用概率

②M步：基于估计的状态占用概率，重新估计参数

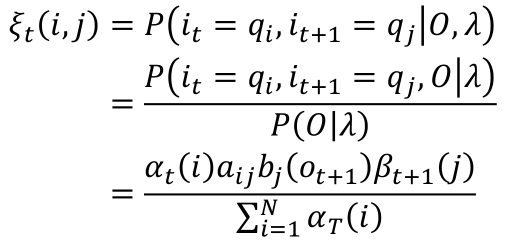
对于某个状态，将所有时刻的状态占用概率相加，可以认为是一个软次数

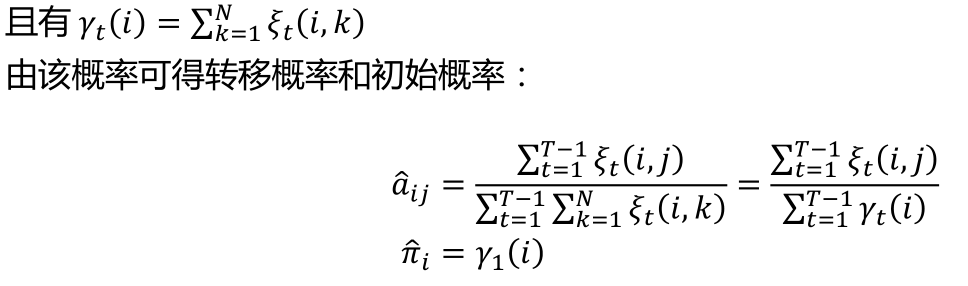
可以使用该软次数重新估计HMM参数



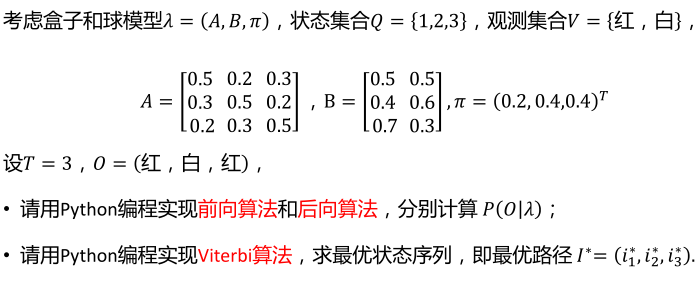
与状态占用概率类似，定义给定模型参数和观测，在时刻t处于状态qi且在时刻t + 1处于

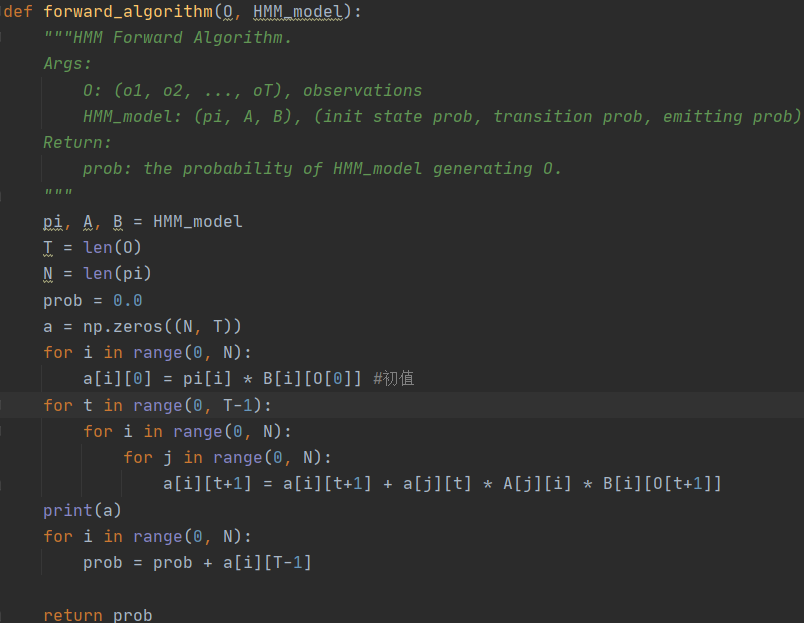
状态qi的概率：

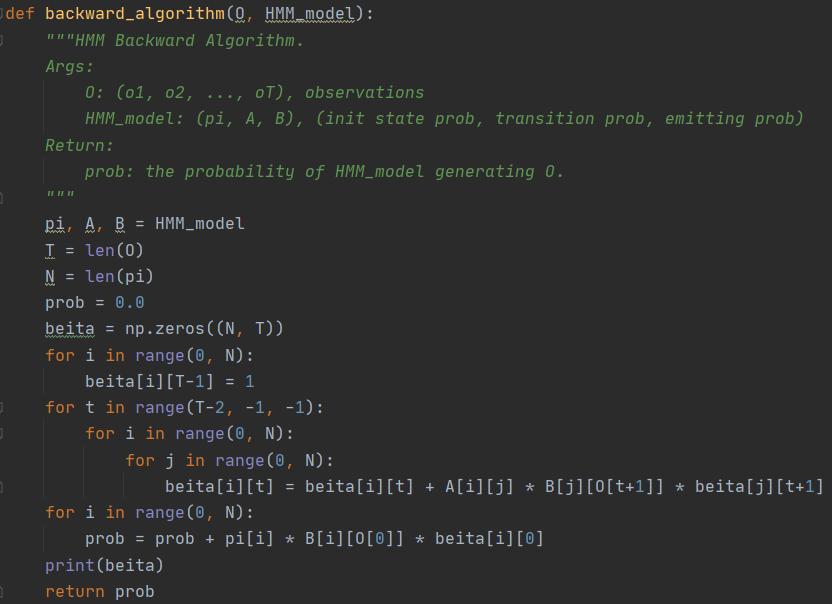


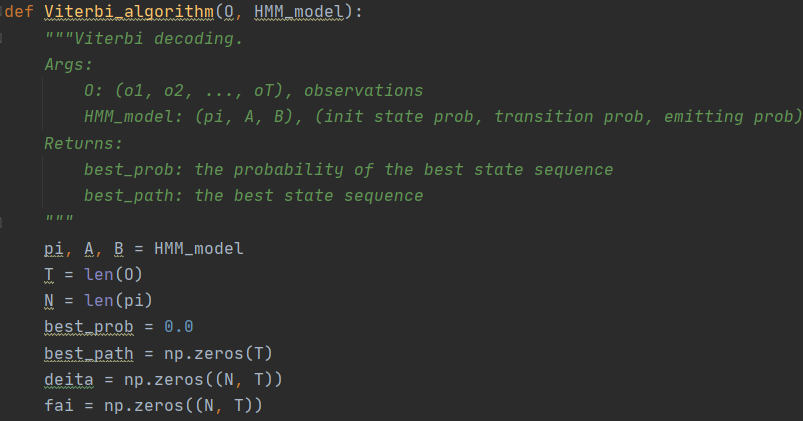


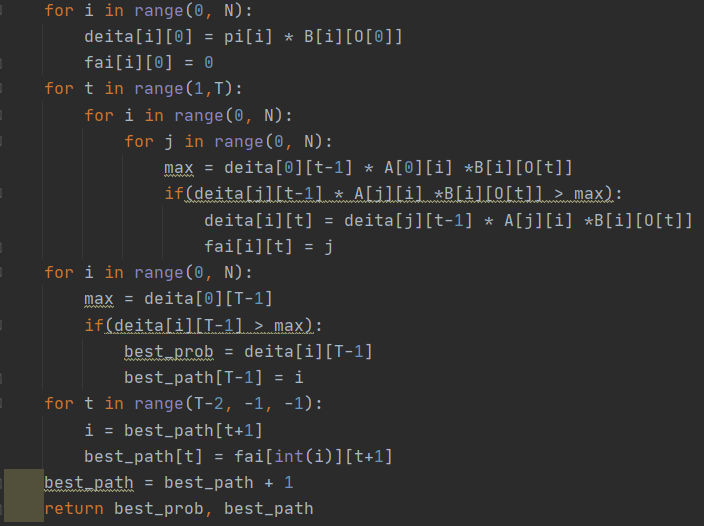
练习：



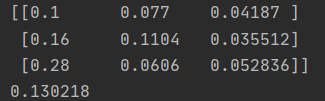




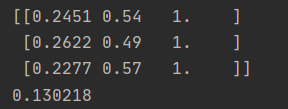




前向算法



后向算法



Viterbi解码

IMG_256

最终解得，最可能都是从第三个盒子中取出来的。