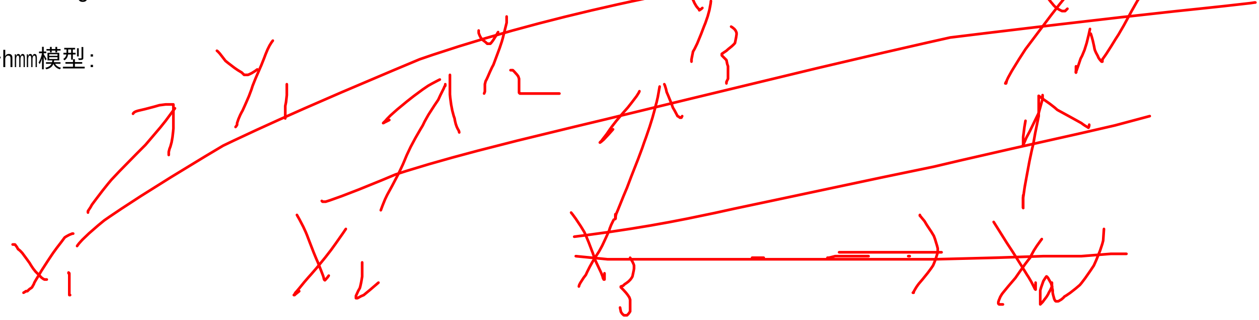
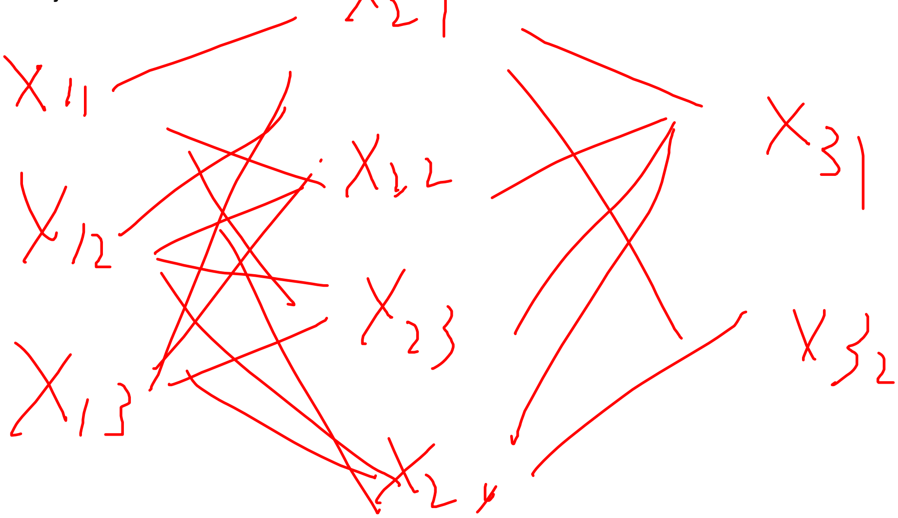
Viterbi-Algorithm算法:很简单的一个算法,但是是最实用的.

适用于hmm模型:



xij表示状态xi的第j个可能的值



下面就是计算这个图的最短路径了.

Viterbi

假设上面的网络结构所有的边长度都已知,那么求起点到重点的最短距离

1. 从start出发.计算d(S,x1i)　　其中S表示start, x1i表示第一个节点的所有状态.　d是距离

2. $d(S,x2i)=\min_j\ d(S,x1j)+d(x1j,x2i)$　　不停的跑2一直到网络结束即可.

完毕

如果假定这个在这个隐含马尔可夫链中节点最多的状态有D个节点，也就是说整个网格的宽度为D，那么任何一步的复杂度不超过O(D2)，由于网格长度是N，所以整个维特比算法的复杂度是O(N·D2).显然正确.

总结一下复杂度.= 序列长*状态数max的平方

最常用的梯度公式: 因为l就表示loss_function 是一个scala,然后里面的变量y,x都是列向量.这种情况是最常遇到的.

$$l = g(\vec{y})$$

$$\vec{y} = y(x_1 - - - x_n)$$

$$\Rightarrow \frac{\partial l}{\partial \vec{x}} = \text{列向量} = \left(\frac{\partial \vec{y}}{\partial \vec{x}}\right)' \times \frac{\partial l}{\partial \vec{y}}$$

左式

记忆: 右边l对y求导是基本的链式法则第一步, 因为要一个列向量,所以必须左乘一个矩阵,根据相消原则,左边的jacobi还需要写一个转职!!!!!!!!

逆一般推广 $l \triangleq$ loss function

$$l = l(\vec{y}_1)$$

$$\vec{y}_1 = \vec{y}_1(\vec{y}_2)$$

$$\vdots$$

$$\vec{y}_{n-1} = \vec{y}_{n-1}(\vec{y}_n)$$

除了第一个l是scala以外,其他量都是列向量.这基本是所有学习任务计算梯度的抽象了.
下面计算l对于任意中间变量的梯度.

$$\frac{\partial l}{\partial y_1} = \text{就是正常的函数分量求导数而已} = \begin{pmatrix} \frac{\partial l}{\partial y_{11}} \\ \vdots \\ \frac{\partial l}{\partial y_{1n}} \end{pmatrix}$$

$$\frac{\partial l}{\partial y_2} = \left(\frac{\partial y_1}{\partial y_2}\right)' \times \frac{\partial l}{\partial y_1}$$

上面证明过了

$$\frac{\partial l}{\partial y_3} = \left(\frac{\partial y_2}{\partial y_3}\right)' \times \frac{\partial l}{\partial y_2} = \left(\frac{\partial y_2}{\partial y_3}\right)' \times \left(\frac{\partial y_1}{\partial y_2}\right)' \times \frac{\partial l}{\partial y_1}$$

因为右边要一个列向量和一个矩阵做,
所以显然要这么写.

带入上面即可

所有链式法则的精髓公式!!!!!!!!!

下面写出最一般的,作为收尾!!!!!!!!!三花

$$\boxed{\frac{\partial L}{\partial y_n} = \left(\frac{\partial y_{n-1}}{\partial y_n}\right)' \times \cdots \times \left(\frac{\partial y_1}{\partial y_2}\right)' \times \frac{\partial L}{\partial y_1}}$$

$$y = c'(y) \oplus v$$

$$z(y) = c'(y) \oplus u$$

$$y = c'(y) \oplus u \oplus w$$

$$h \triangleq c'(y) \oplus w$$

$$z(h) = c'(g)$$