2019—2020学年(春)第二学期
中国科学院大学课程

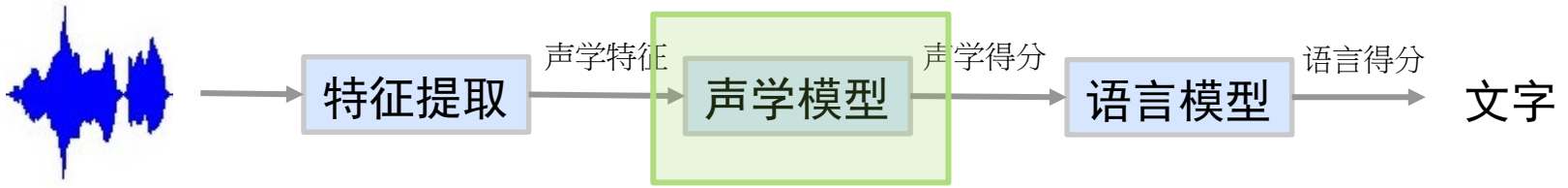
中国科学院自动化研究所
INSTITUTE OF AUTOMATION
CHINESE ACADEMY OF SCIENCES

语音交互技术
——语音识别（二）

中国科学院自动化研究所
模式识别国家重点实验室

陶建华
jhtao@nlpr.ia.ac.cn

经典语音识别系统结构

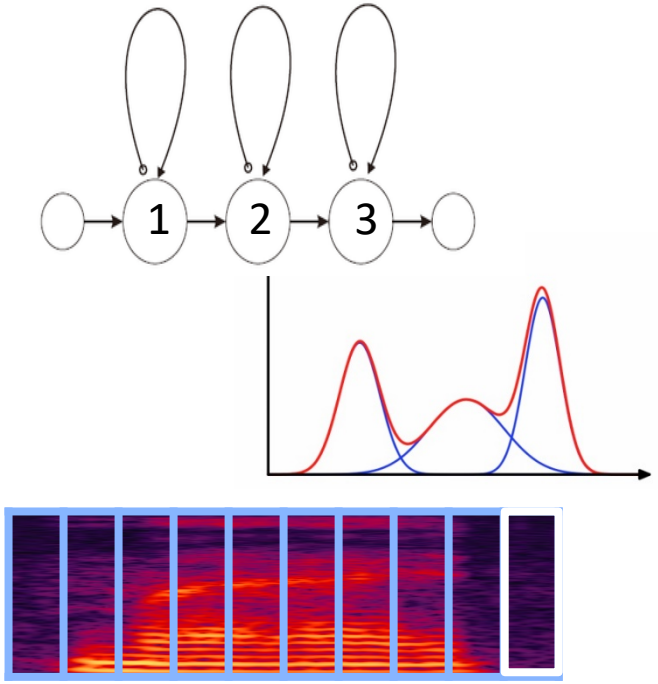


提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

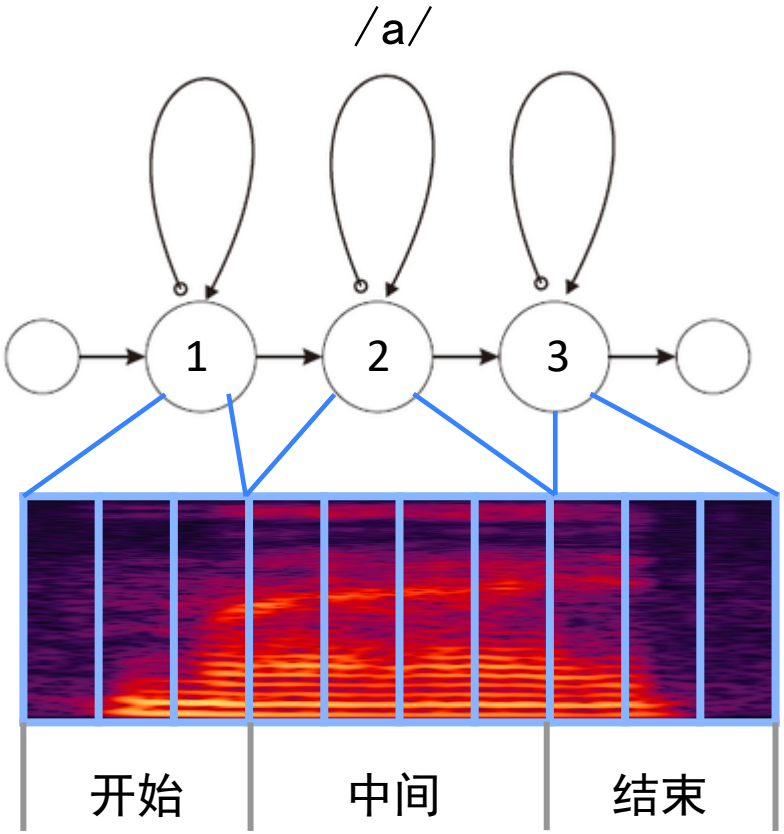
回顾GMM-HMM

- 利用GMM-HMM来对语音动态特性建模。
- 语音特征表示HMM的观测。GMM来表示给定状态下的观测概率，HMM状态之间有转移概率。
- GMM-HMM的训练采用期望最大化算法。
- 解码时采用Viterbi算法。



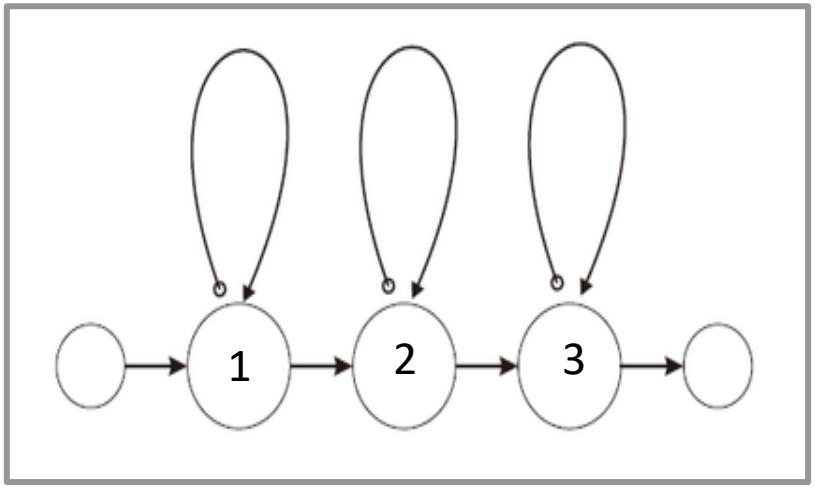
语音中常用的HMM拓扑结构

- 三状态串接拓扑结构的HMM表示一个发音单元。
- 也有采用五状态串接的形式的。



HMM建模单元的选择

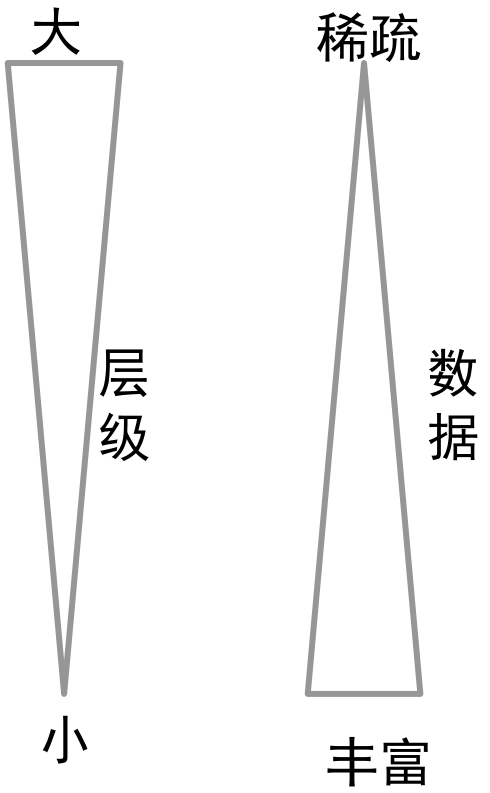
- 用一个HMM模型表示什么样的发音单元？



HMM建模单元的选择

汉语建模单元举例

建模单元	例子
汉字	人
音节	ren
声韵母	r en
音素	r e n



建模层级越高，对音段整体建模效果越好，但可供训练的数据越少。

选择合适的建模单元非常重要。

HMM建模单元的选择

- 音素：根据语音的自然属性划分出来的最小语音单位，由语言学专家标定出，是最自然的基本建模单元。
- 语音识别中，音素泛指基本的发音单位。
- 然而，音素的层级往往太小，难以解决协同发音问题。

带调的声韵母举例

爱国 ai4 g uo2

爱国主义 ai4 g uo2 zh u3 ii i4

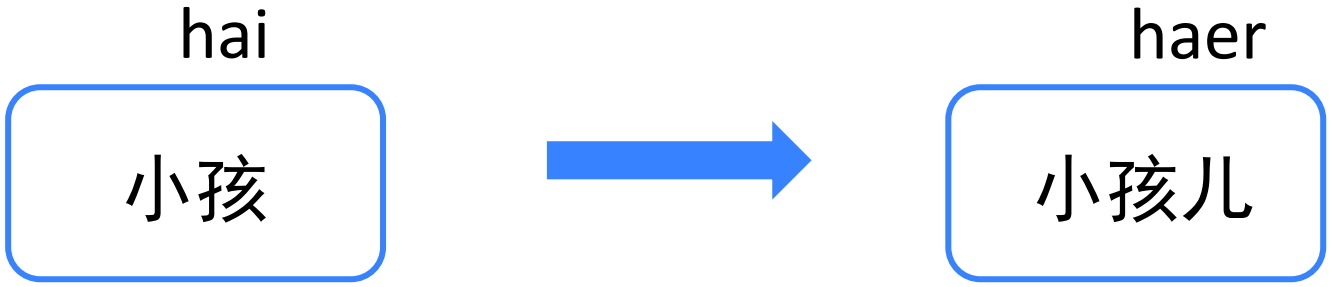
爱好者 ai4 h ao4 zh e3

爱花 ai4 h ua1

爱情 aa ai4 q ing2

三音素

- 协同发音就是音节相连而产生的音变现象。典型如儿化音。
- 对音素建模不能很好处理这种音变现象。
- 一个有效的做法，是将前后的多个音素连接起来，作为整体进行建模。
- 三音素即是将前后的音素，与当前的音素，三个看成整体来建模。比如 h ai er 三个音素连起来，即可以对应“孩儿”。

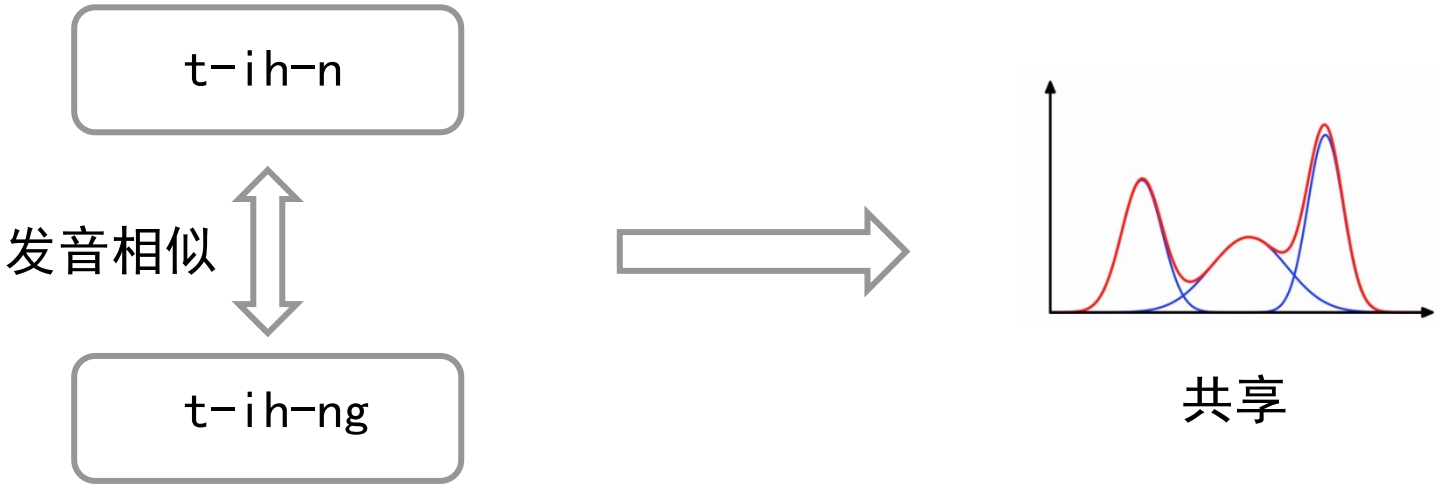


数据稀疏问题

- 三音素整体建模考虑了协同发音问题，能更好地具备发音的区分性。
- 然而这种方法存在数据稀疏问题。
- 举例来说，假设汉语常用声韵母约60个，那么其有可能组成60×60×60=216000种三音素。若每个三音素用3状态HMM建模，则共有3×216000=648000个状态。数量过于庞大。
- 这些三音素中，有的不会实际发生，有的会实际发生，但训练语料中并不存在，这造成模型无法有效训练。
- 采用决策树绑定来解决数据稀疏问题。

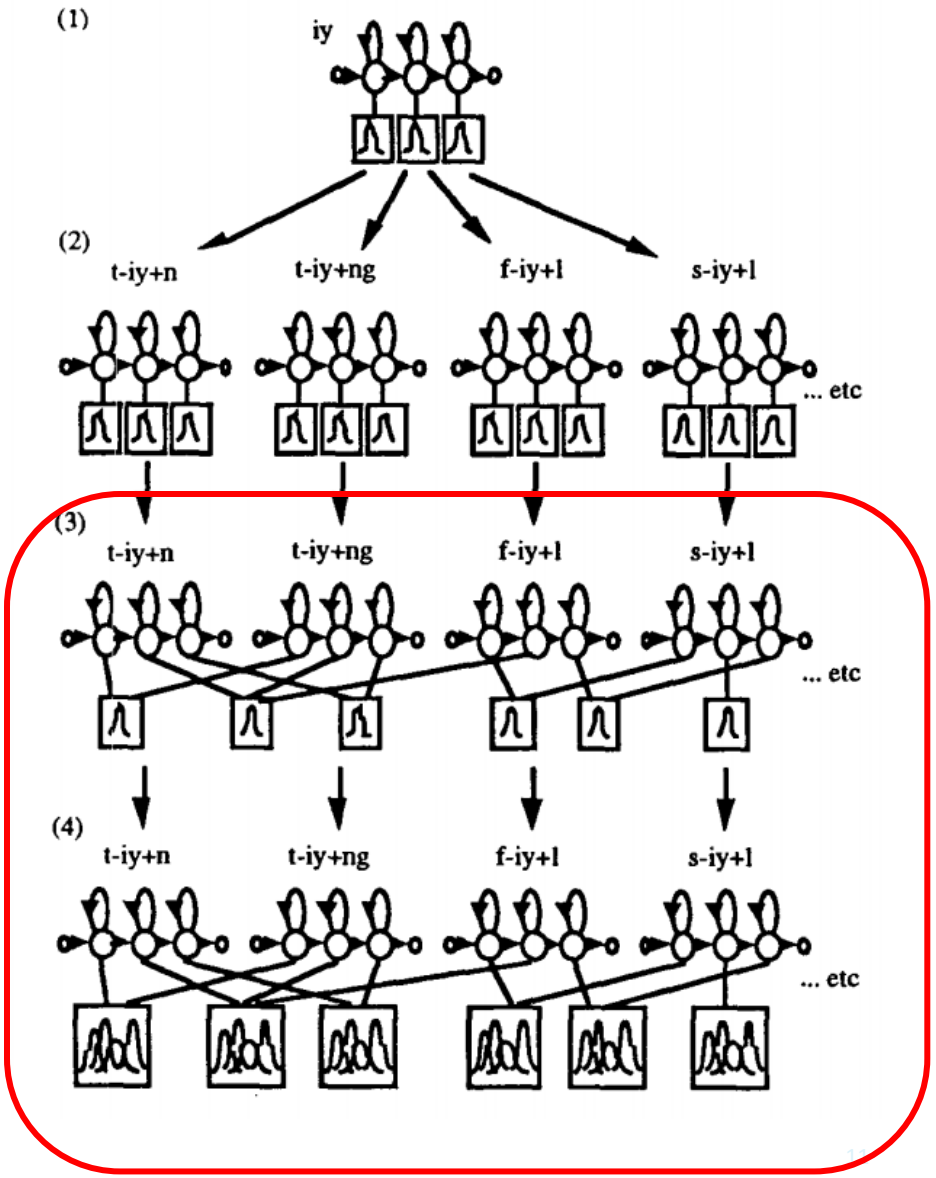
基于决策树的状态绑定

- 针对数据稀疏问题，Steve Young，黄美玉等分别提出了基于决策树的状态绑定。
- 状态绑定指的是让发音相似的三音素对应的HMM状态，使用同一个GMM概率密度函数。这样，即使未见的三音素也的GMM也可以得到训练。



状态绑定示意图

- 对于音素iy，可以构建出t-iy+n, t-iy+ng等三音素。
- 经过聚类，可以发现有一些三音素的状态可以共享一组概率密度。
- 聚类工具为决策树。决策树可以融合数据与人类的先验知识。



基于决策树的状态绑定

- 基于决策树的状态绑定融合了语言学知识。语言学家提供一个问题集来考量当前三音素应该归为哪一类。
- 问题类似“左边的音素是鼻音吗？”决策树自上而下分裂为二叉树。

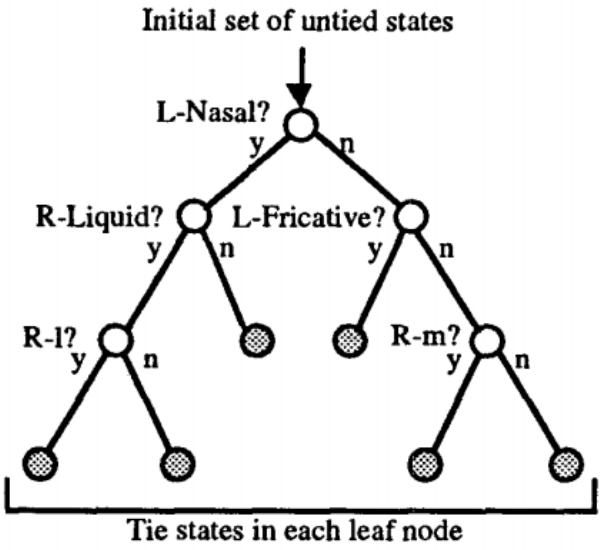


Figure 2: Example of a phonetic decision tree

基于决策树的状态绑定

- 每一个叶子结点表示一个绑定的状态集合。叶子节点数目就是最终的绑定的概率密度函数的个数。

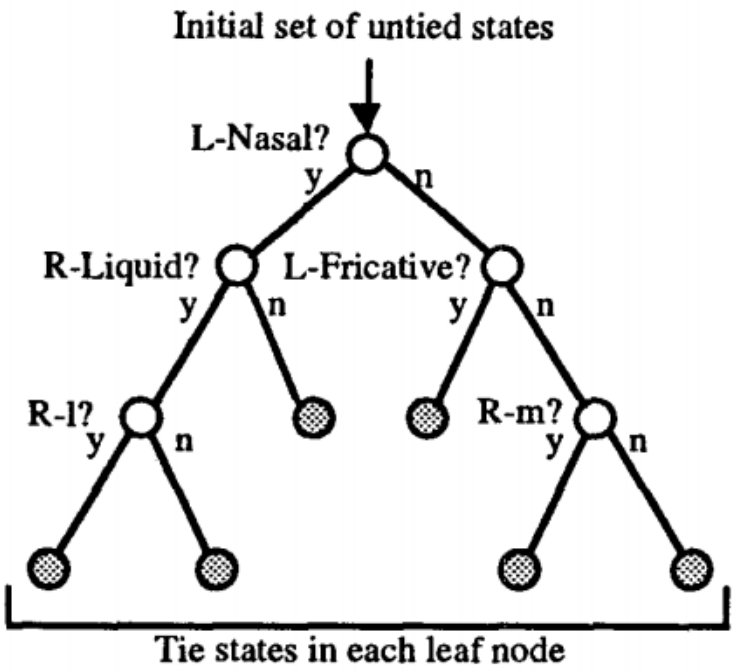


Figure 2: Example of a phonetic decision tree

基于决策树的状态绑定

- 绑定的每一个状态集合，都对应一个概率密度函数，称为**Senone**。

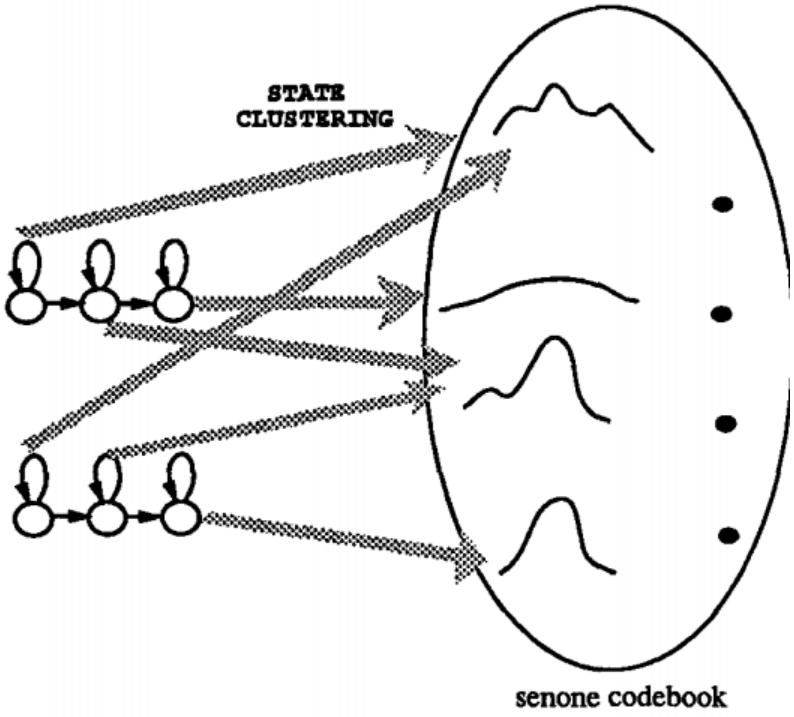


Figure 1: Creation of the senone codebook.

基于决策树的状态绑定

- 构建决策树的目标是最大化绑定后状态的对数似然

$$L(\mathbf{S}) = \sum_{f \in F} \sum_{s \in S} \log(Pr(\mathbf{o}_f; \mu(\mathbf{S}), \Sigma(\mathbf{S}))\gamma_s(\mathbf{o}_f)$$

- 其中F表示帧的集合，S表示绑定后状态集合，Pr为给定状态，帧发生的概率；γ为状态的后验概率，即状态发生的概率。所以L为绑定后状态的对数似然。

- 计算决策树分裂后，L的增益

$$\Delta L_q = L(\mathbf{S}_y(q)) + L(\mathbf{S}_n(q)) - L(\mathbf{S})$$

- 每次迭代，贪心地从问题集中挑选出能最大化增益的问题进行子树的分裂。

实践中的训练

1. 训练单音素的单高斯HMM模型。
2. 将训练的单高斯复制多份，对各复制的高斯成分添加随机噪声，然后进行参数重估，训练单音素GMM，并用此模型再训练语料上进行维特比搜索，将状态对齐到每一帧。
3. 根据对齐的结果，计算训练决策树的统计量，进行决策树绑定。
4. 根据标注，进行维特比搜索，将状态对齐到每一帧，然后重估GMM参数。
5. 反复迭代直到收敛。

提纲

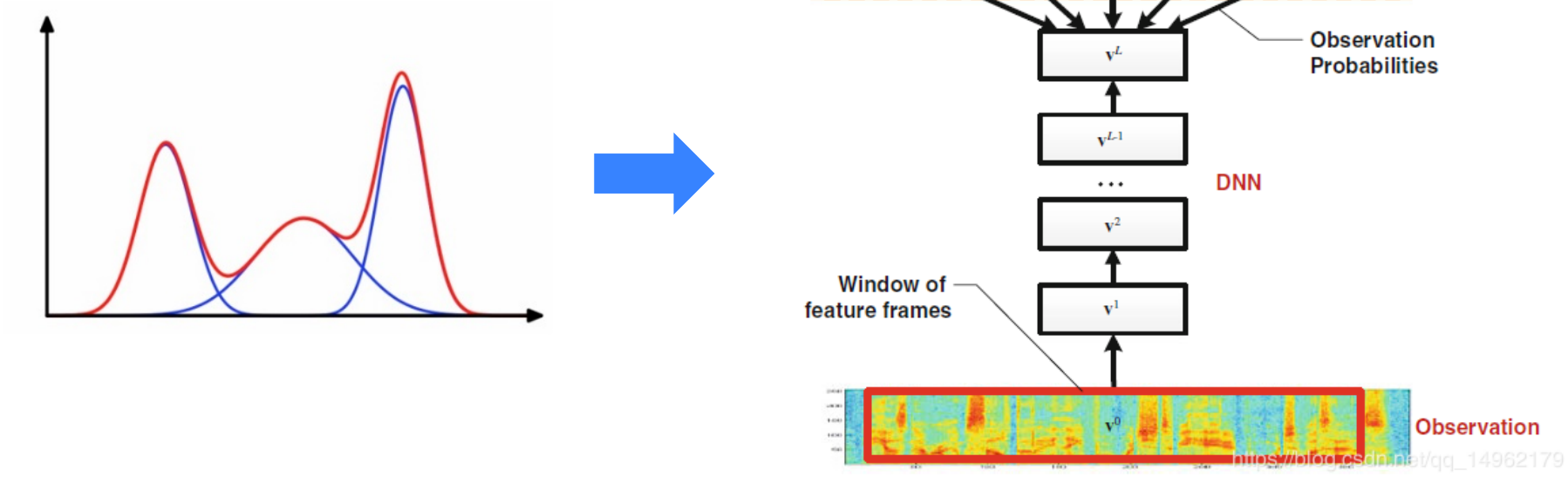
- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

DNN-HMM

- GMM-HMM框架一直引领语音识别技术近30年，取得了很大发展。然而到后期基于GMM-HMM模型的语音识别系统性能提升幅度很小。
- 2009年左右，俞栋，Hinton等人认为，GMM在拟合复杂高维流形能力存在不足，并将深度神经网络首先用于语音识别声学建模，取得了成功，大大提升了语音识别性能。自此开始，深度学习开始蓬勃发展。
- 近十年来，各种深度学习技术的应用，开始将语音识别系统带向真正实用化。

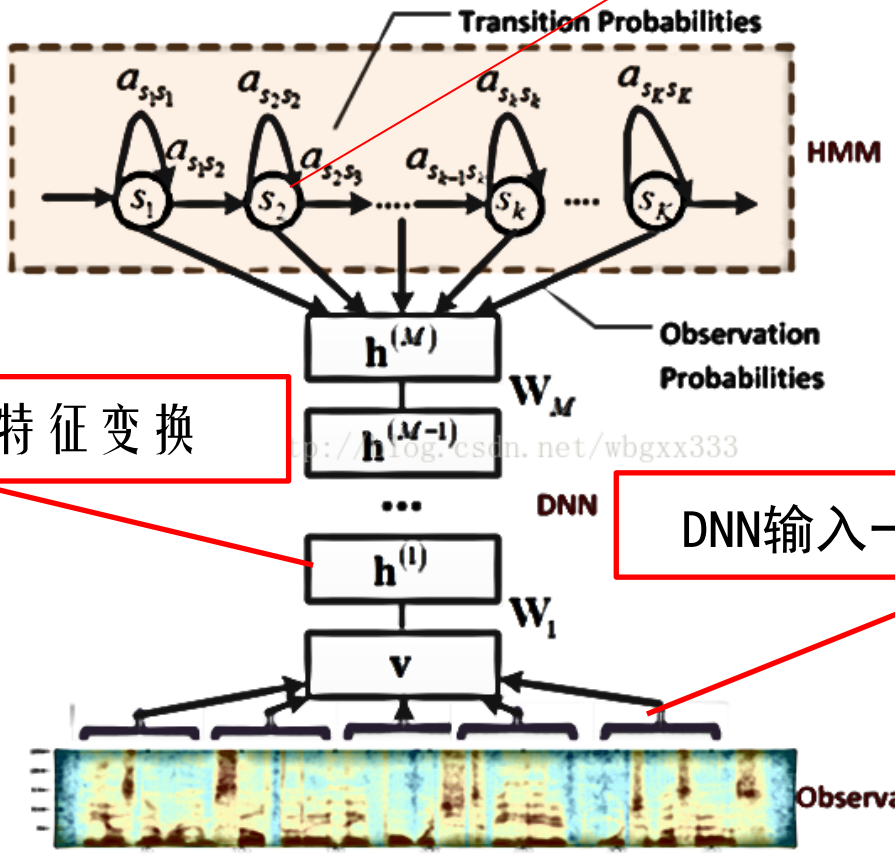
DNN-HMM

- 在DNN-HMM中，DNN被用来替换GMM建模观测概率。



DNN-HMM

DNN的输出为Senones的概率分布



多层非线性特征变换

DNN输入一个或多个语音帧

DNN-HMM

- 深度神经网络的输出一般为Senone的概率分布，由最高层Softmax函数得到。
- 深度神经网络的训练数据由GMM-HMM模型进行对齐得到，给每帧标注Senone的ID号。
- 深度神经网络的训练准则：交叉熵（CE）

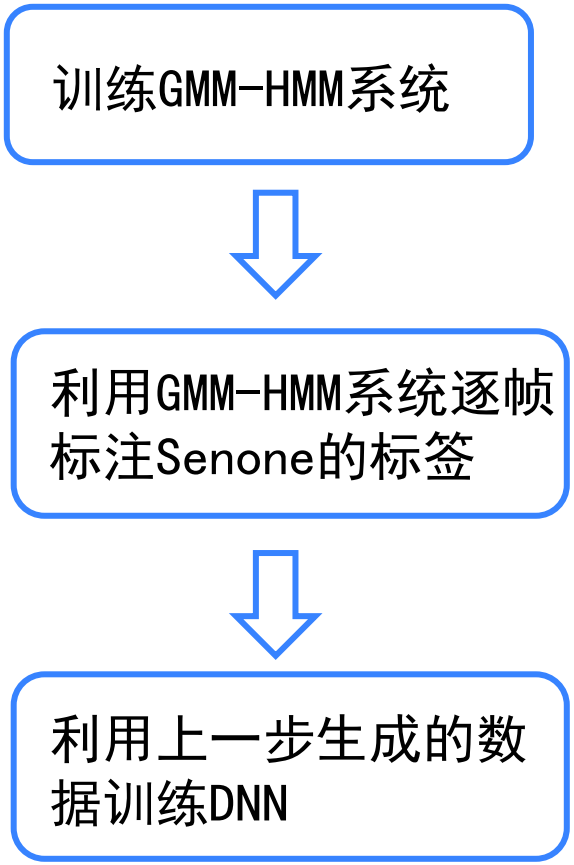
$$E^i = -\sum_{t=1}^{K} t_t^i \log y_t^i$$

Senone的one-hot编码

DNN的输出

DNN-HMM

- DNN-HMM训练过程：三步

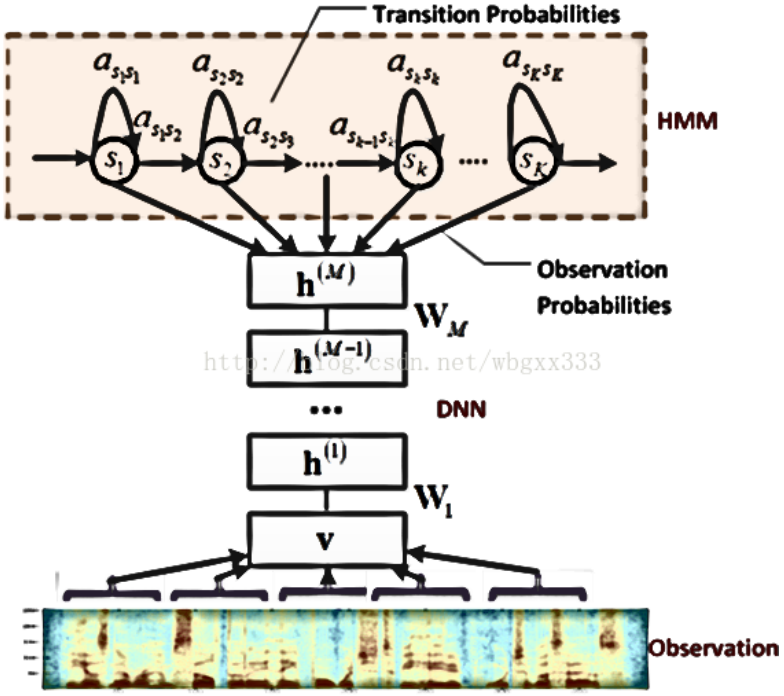


DNN-HMM

- 神经网络的结构可以根据需求灵活选取。
- 常见的神经网络-隐马尔可夫混合声学模型结构
  - 神经网络–隐马尔科夫模型（DNN-HMM）
  - 时延神经网络–隐马尔科夫模型（TDNN-HMM）
  - 循环神经网络–隐马尔科夫模型（RNN-HMM）
  - 卷积神经网络–隐马尔科夫模型（CNN-HMM）

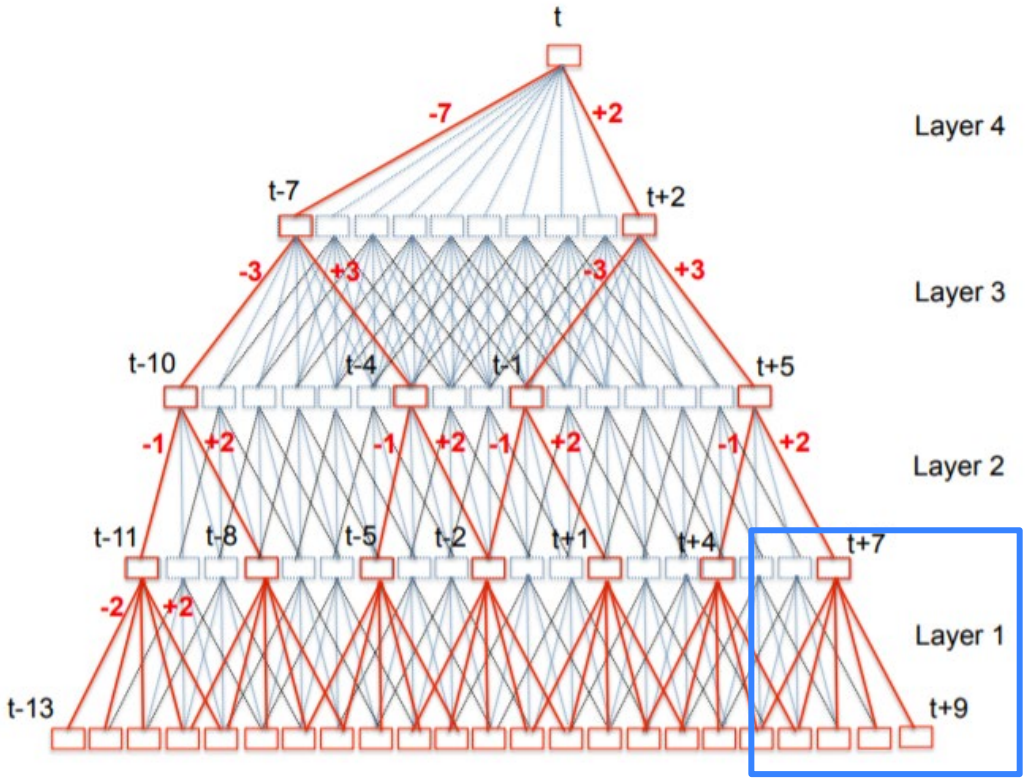
DNN-HMM

- 早期的DNN-HMM采用最基本的前馈神经网络，已经可以取得超越GMM-HMM的效果。
- 然而简单的前馈神经网络捕捉上下文能力不足。



DNN-HMM

- 时延神经网络（Time-delay neural networks，TDNN)在每一层都将前后多帧拼接在一起输入下一层。可以捕捉很长的上下文信息，效果优于普通DNN。

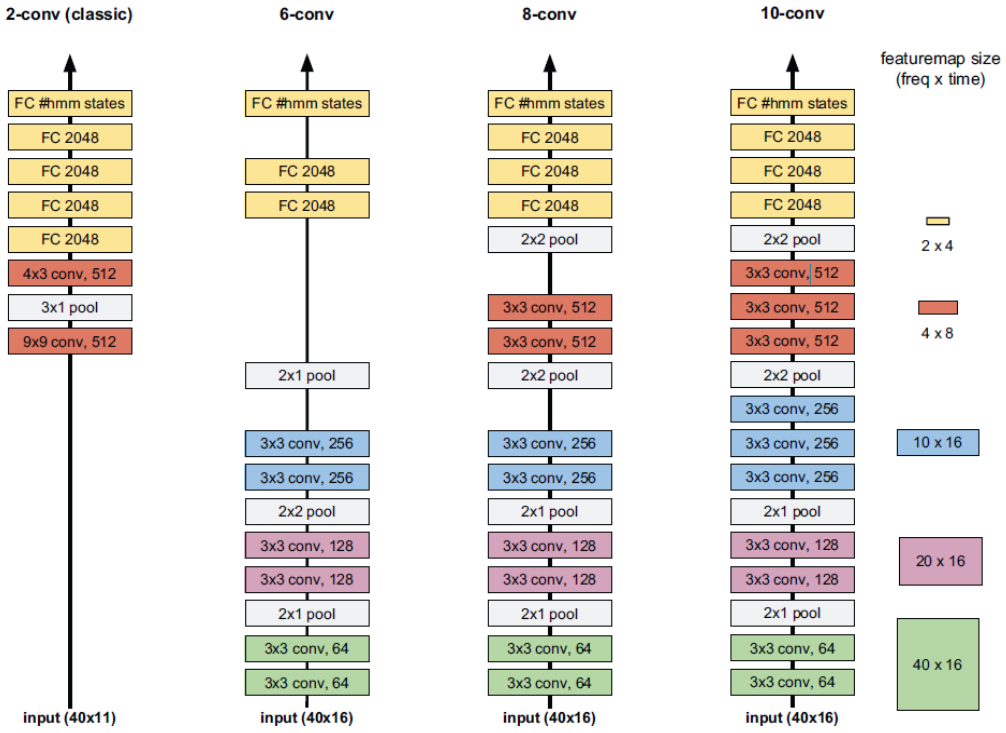


5帧拼接

DNN-HMM

基于CNN-HMM的声学模型

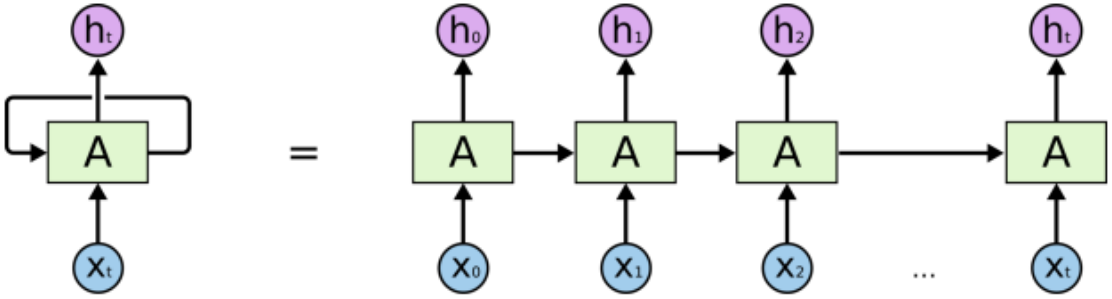
- CNN可以捕捉局部不变性，在图像分析中取得了成功。
- 对于语音的分析，可以看作是对语谱图的分析过程。
- 利用CNN来进行语谱图分析，提高了声学模型的鲁棒性。



DNN-HMM

基于RNN-HMM的声学模型

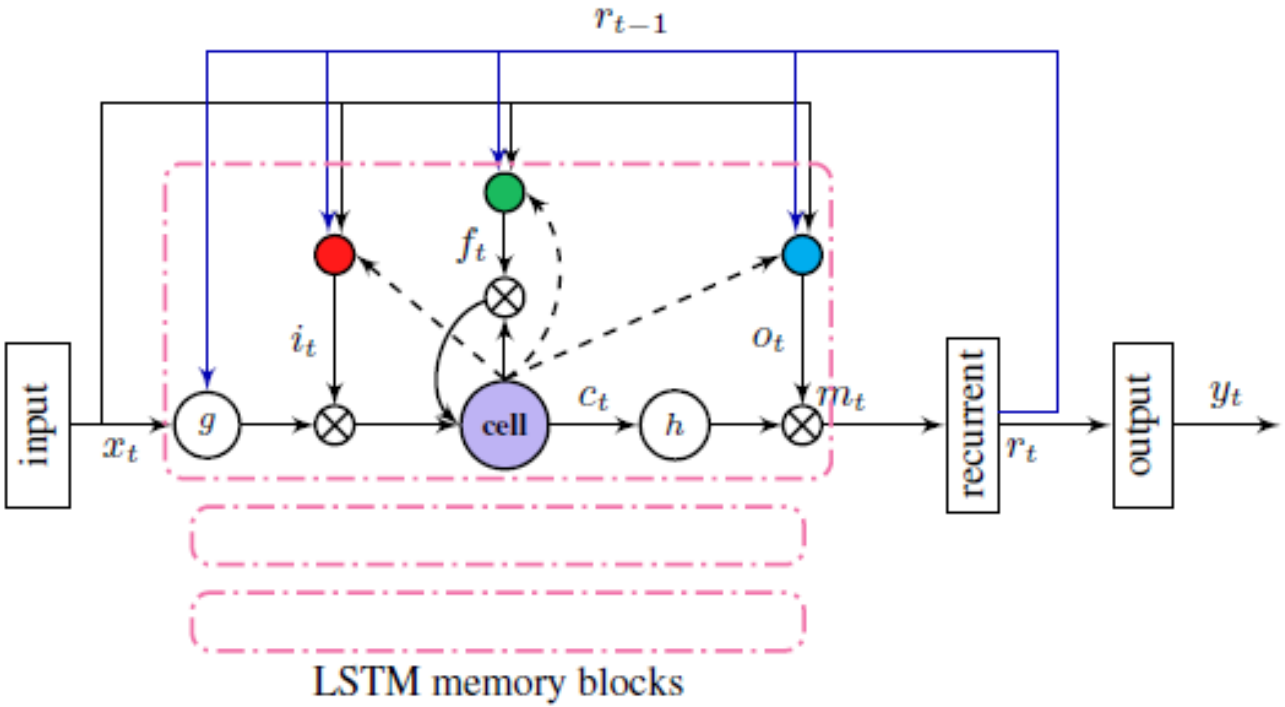
- RNN 善于捕捉特征的时序关系
- RNN 容易梯度爆炸或梯度消失，难以训练
- 一般采用LSTM，GRU等特殊的RNN



展开的 RNN

DNN-HMM

- 基于LSTM-HMM的声学模型
- LSTM 采用一些控制门（输入门，遗忘门和输出门）来减少梯度累积的长度，一定程度上解决了RNN 训练时梯度消失和梯度扩散的问题。



跳出HMM框架

- DNN-HMM系统训练复杂：需要首先训练GMM-HMM系统生成帧级别标注。
- DNN-HMM系统建模单元不灵活：依赖GMM-HMM的生成的Senone。

能否跳出HMM框架，端到端地进行声学建模？

提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

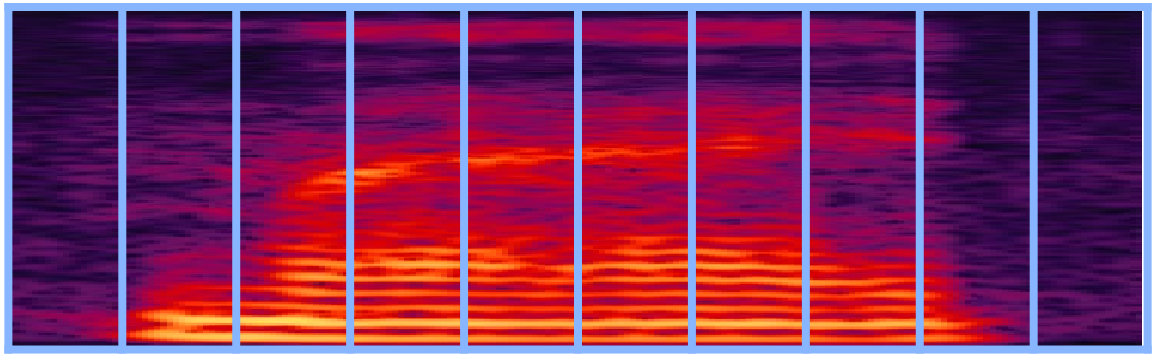
困难点

- 神经网络是逐帧给出预测。
- 标签却不是逐帧给出的。交叉熵损失无法训练。

天气很好

t i an q i h en h ao

如何对应？



联结主义时序分类（CTC）

- 能否不进行逐帧标注，让神经网络自动地学习出对齐？
- 联结主义时序分类：给标注加入blank符号，枚举出各种可能。
- 规则：允许标注重复，允许标注中间插入blank符号。

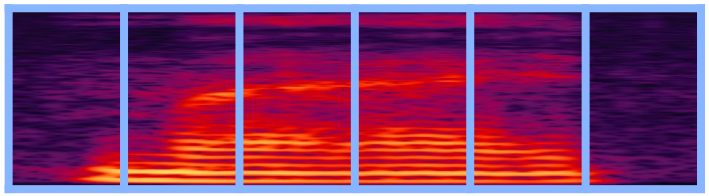
标注cat和7帧语音可能的对应

c _ _ a _ _ t

c c _ a _ _ t

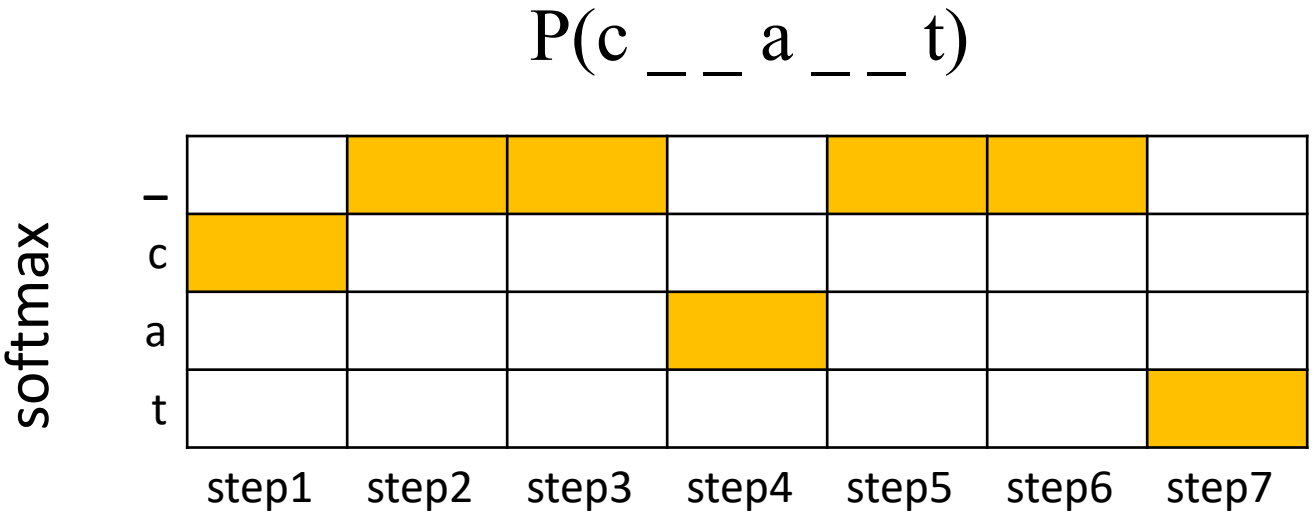
_ c _ a _ _ t

…



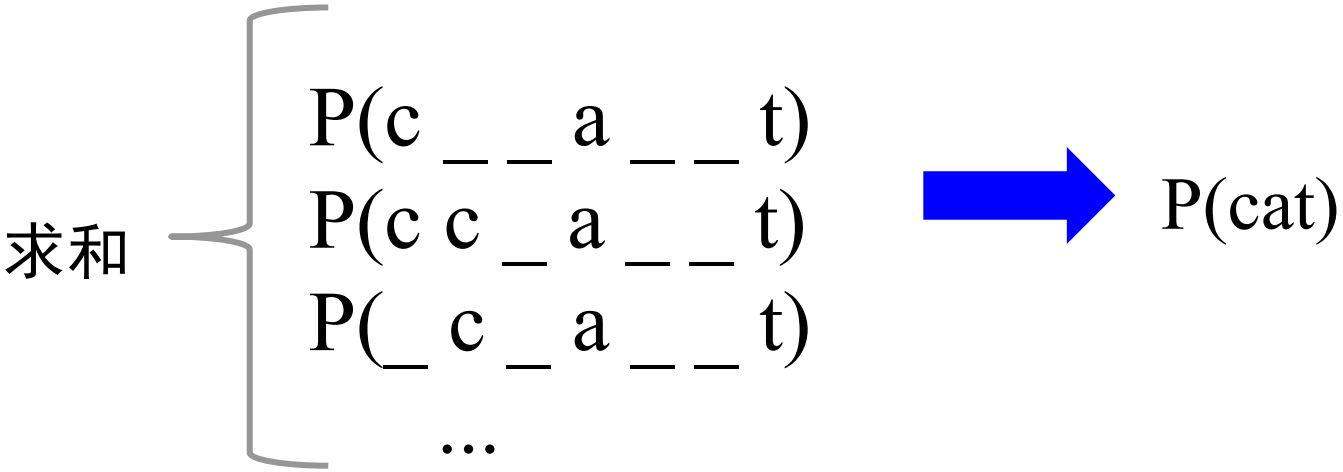
联结主义时序分类（CTC）

- 在CTC框架下，Softmax输出多一个blank的位置。
- 假设神经网络每一步输出的概率是独立的，那么CTC标注序列（包含blank符号）的概率为对应位置连乘。



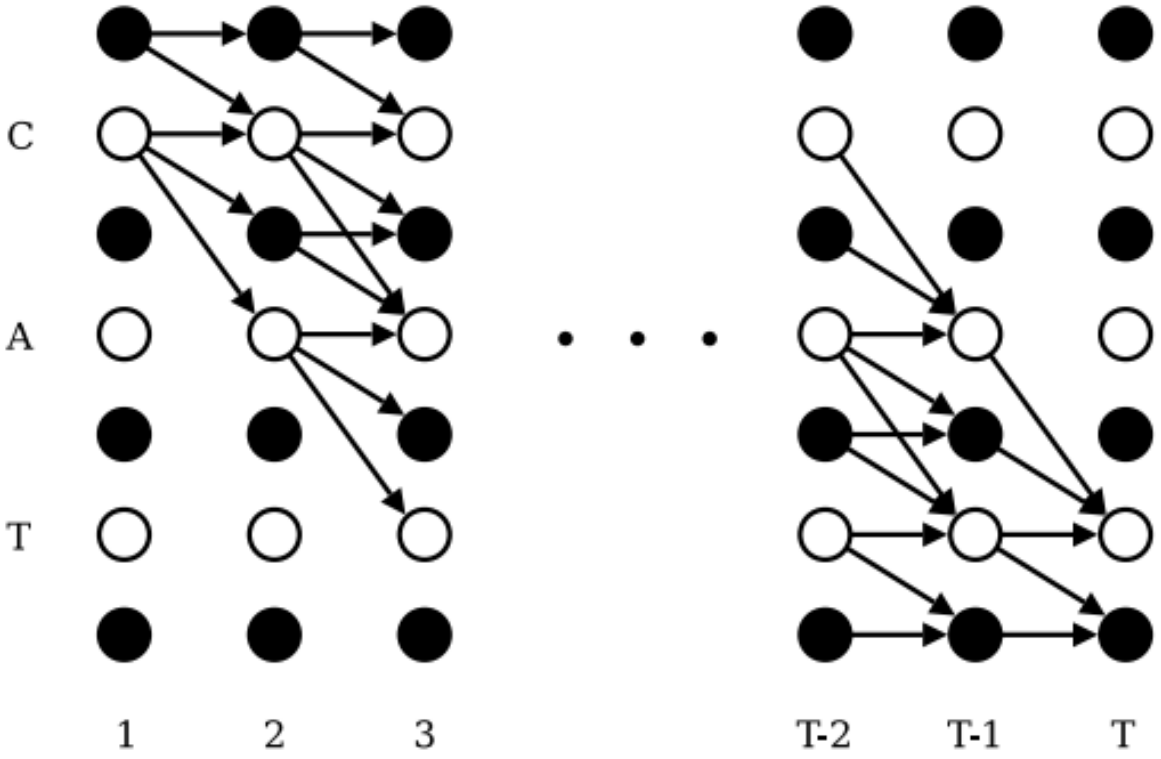
联结主义时序分类（CTC）

- 将所有可能的CTC序列的概率求和，得到标注序列的概率。



联结时序分类模型（CTC）

- 枚举方法：概率路径图（黑点表示blank）
- 用和HMM类似的前后向算法高效求和

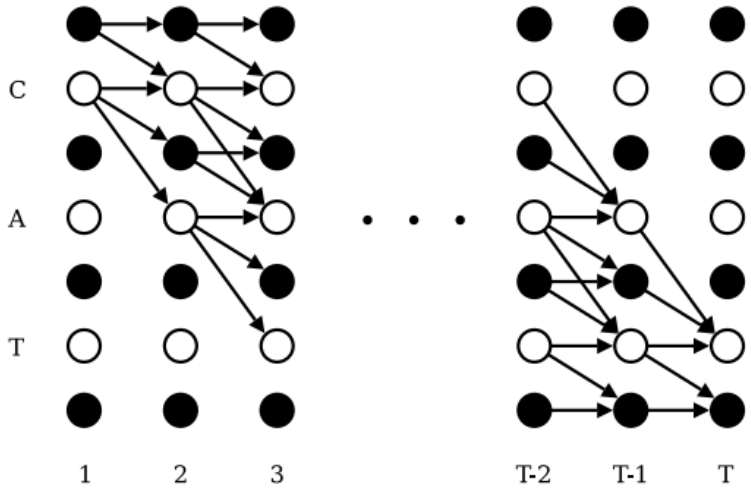


联结时序分类模型（CTC）

- 前向算法及递归形式

$$\alpha(t,u) = \sum_{\pi \in V(t,u)} \prod_{i=1}^{t} y_{\pi_i}^{i}$$

$$\alpha(t,u) = y_{l'_u}^{t} \sum_{i=f(u)}^{u} \alpha(t-1,i)$$



- 后向算法及递归形式

$$\beta(t,u) = \sum_{\pi \in W(t,u)} \prod_{i=1}^{T-t} y_{\pi_i}^{t+i}$$

$$\beta(t,u) = \sum_{i=u} \beta(t+1,i) y_{l'_i}^{t+1}$$

联结时序分类模型（CTC）

- 令x表示声学序列，z表示标注序列，y_k^t表示第t时刻softmax第k个输出
- 损失计算

$$p(\mathbf{z}|\mathbf{x}) = \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta(t,u)$$

$$\mathcal{L}(\mathbf{x},\mathbf{z}) = -\ln \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta(t,u)$$

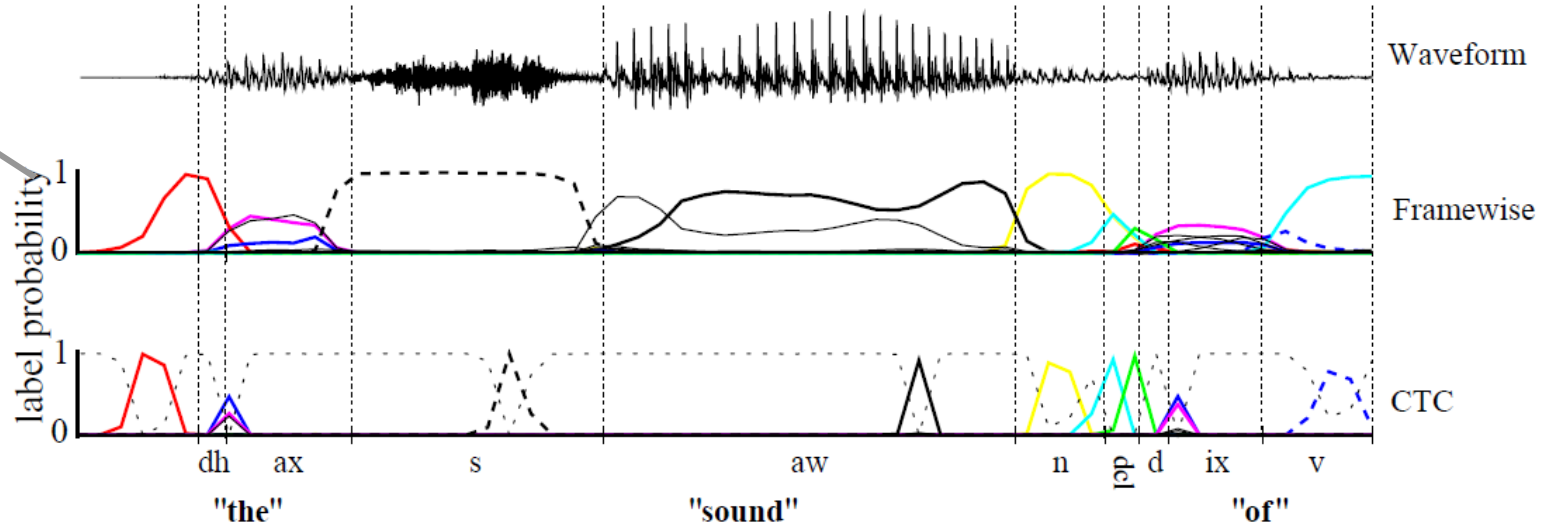
- 负对数损失

$$\frac{\partial \mathcal{L}(\mathbf{x},\mathbf{z})}{\partial y_k^t} = -\frac{\partial \ln p(\mathbf{z}|\mathbf{x})}{\partial y_k^t} = -\frac{1}{p(\mathbf{z}|\mathbf{x})}\frac{\partial p(\mathbf{z}|\mathbf{x})}{\partial y_k^t}$$

联结时序分类模型（CTC）

- 使用CTC准则训练的神经网络会出现尖峰特性。
- DNN-HMM输出的结果则为平台。

传统深度学习混合模型

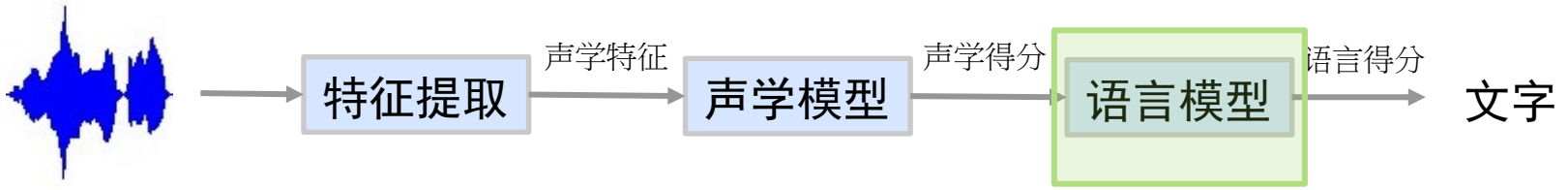


联结时序分类模型（CTC）

CTC模型特点

- CTC中，神经网络不对HMM状态建模，而是直接对发音单元，甚至是字母，汉字建模。
- 不需要训练GMM-HMM生成逐帧标签，直接训练神经网络
- 模型预测90%以上都是空格，在解码的时候可以跳过，加快了解码速度。

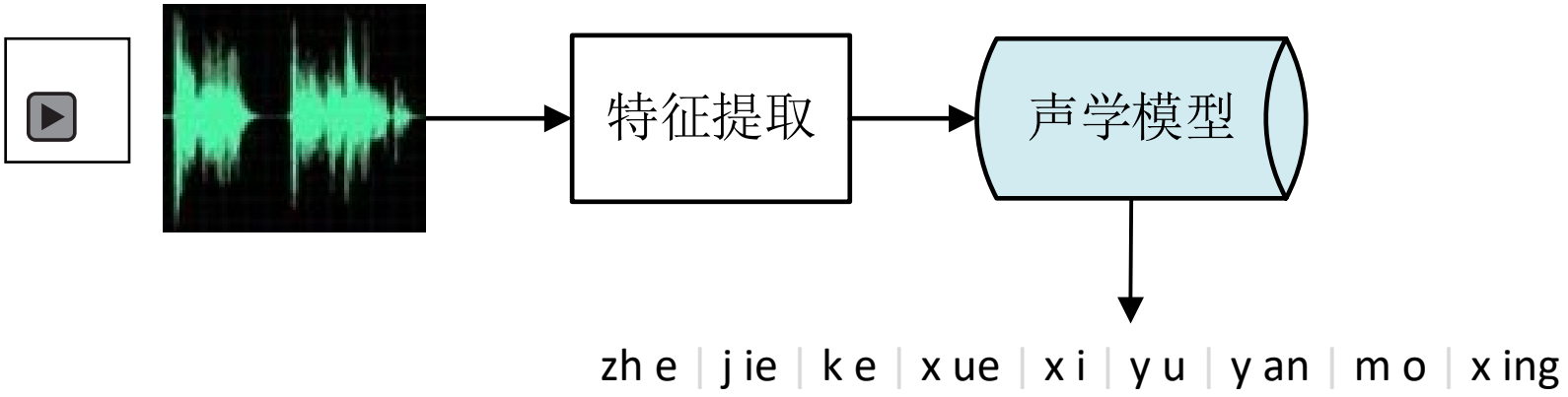
经典语音识别系统结构



提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

声学模型



语言模型

zh e	j ie	k e	x ue	x i	y u	y an	m o	x ing
这	节	可	学	洗	与	演	墨	行
着	接	克	雪	吸	雨	燕	莫	星
者	姐	课	靴	喜	于	言	模	型
折	界	客	薛	习	预	盐	磨	醒
遮	结	科	穴	西	语	彦	魔	形
……								

语言模型

zh e	j ie	k e	x ue	x i	y u	y an	m o	x ing
这	节	可	学	洗	与	演	墨	行
着	接	克	雪	吸	雨	燕	莫	星
者	姐	课	靴	喜	于	言	模	型
折	界	客	薛	习	预	盐	磨	醒
遮	结	科	穴	西	语	彦	魔	形

.....

P(这节课学习语言模型) 最大

马尔可夫性

- 马尔可夫性假设一个单词出现的概率只依赖于前面出现的有限个单词，而不是所有单词。

$$P(w_1, w_2, \cdots, w_T) = P(w_1)P(w_2|w_1)\cdots P(w_T|w_{T-N+1}, \cdots, w_{T-1})$$

- 如上式，分解的时候只考虑前面的N个单词。
- 马尔可夫性大大压缩了统计的求解空间，使概率的估计成为可能。
- 事实上，马尔可夫性质最早的应用，就是马尔可夫构建语言模型统计俄罗斯文学中韵母出现的概率。

安德烈·马尔可夫

N元语法

- 条件概率 $P(w_t|w_{t-N+1},\cdots,w_{t-1})$ 的估计很简单，采用极大似然估计准则，对语料进行计数即可。

- 统计N元语法 $w_{t-N+1},\cdots,w_t$ N-1元语法 $w_{t-N+1},\cdots,w_{t-1}$ 的数目，然后作商，即可得到估计值

$$\widehat{P}(w_t|w_{t-N+1},\cdots,w_{t-1}) = \frac{c(w_{t-N+1},\cdots,w_t)}{c(w_{t-N+1},\cdots,w_{t-1})}$$

- 这里还假设N元语法的概率与其发生的位置无关，即N元语法在开头、结尾、还是文档中任意一个位置概率都是一样的。

- 当N=1时，则假设每一个词和其它词都是独立的，此时分母为文档的总词数，称为一元文法（Unigram）；当N=2时，则假设每一个词和其前一个词有关，称为二元文法（Bigram）。

例子：2元文法

- 一元文法频率统计

假设语料库总词数为13,748词

我	3437	0.25
想	1215	0.088376
去	3256	0.236834
国科大	938	0.068228
怀柔	213	0.015493
校区	1506	0.109543
图书馆	459	0.033387

例子：2元文法

- 2元文法频率统计

	我	想	去	国科大	怀柔	校区	图书馆
我	8	1087	0	13	0	0	0
想	3	0	786	0	6	8	6
去	3	0	10	860	3	0	12
国科大	0	0	2	0	19	2	52
怀柔	2	0	0	0	0	120	1
校区	19	0	17	0	0	0	0
图书馆	4	0	0	0	0	1	0

例子：2元文法

- P(我 想 去 国科大 怀柔 校区)

=P(我)*P(想|我)*P(去|想)*P(国科大|去)*P(怀柔|国科大)*P(校区|怀柔)

=0.25*(1087/3437)*(786/1215)*(860/3256)*(19/938)*(120/213)

=0.000154171

平滑

- 虽然有了马尔可夫性，但数据的稀疏问题依然存在。直接进行计数可能会导致大量的N元语法概率为0，这显然不符合实际。
- 利用平滑算法，提升零概率N元语法的概率，提高模型的泛化性能。
- 经典的平滑算法有，加一法，古德-图灵估计，回退法等。

加一法

- 加一法又称为拉普拉斯平滑(Laplace Smoothing)。
- 加一法对语料中所有N元语法的计数都加一，这样就避免了零概率。

$$\widehat{P}\left(w_t|w_{t-N+1},\cdots,w_{t-1}\right)=\frac{1+c\left(w_{t-N+1},\cdots,w_t\right)}{|V|+c\left(w_{t-N+1},\cdots,w_{t-1}\right)}$$

- 其中,$|V|$是词表的大小。经过加一平滑以后，所有零概率的N元语法都已经变得大于0，并且估计的概率值依然符合概率的归一化性质，即所有概率求和等于1。
- 显然这是一种最简单的平滑方式，现代语言模型通常不采用这种方式。

古德-图灵估计

- 古德-图灵估计是很多平滑算法的基础。
- 基本思想是将统计得出的计数值折扣一部分，然后分配到零概率N元语法上去。
- 令 n_r 表示训练语料中出现 r 次的N元语法的个数（N元语法频数的频数），则令古德-图灵平滑计数为

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

- 对古德-图灵平滑计数进行归一化，即可得到平滑后频数为r的N元语法的概率

$$p_r = \frac{r^*}{\sum_{r=0}^{\infty} n_r r^*}$$

回退法

- 回退法的基本思想是：如果N元语法不存在，就去考虑其对应的N-1元语法，以此类推，最终考虑一元语法（一元语法就是语料的词频）。

- 一般地，回退算法可以表示为如下形式

$$\hat{P}(w|h) = \begin{cases} \tilde{P}(w|h) & if\ c(hw) > 0 \\ \beta(h)\tilde{P}(w|h') & if\ c(hw) = 0\ and\ c(h) > 0 \\ \beta(h')\tilde{P}(w|h'') & if\ c(hw) = 0\ and\ c(h) = 0\ and\ c(h') > 0 \\ \cdots & \end{cases}$$

- 回退法使用低阶概率去估计高阶，来缓解数据稀疏问题。

- 回退法是现代N元语法最常用的平滑方法。大多数常用的语言模型工具，如SRILM，KenLM，IRSTLM等，都使用回退法构建语言模型。

ARPA格式

- 语言模型的标准存储格式是ARPA格式。SRILM，KenLM，IRSTLM等常用的语言模型工具都使用ARPA格式来存储语言模型。
- 文件开头表示N元语法的数量，接下来是第k阶的N元语法。
- 左边一列表示概率值，右边一列表示回退系数，这二者的数值都取以10为底的对数。

```
\data\
ngram 1=7
ngram 2=8
ngram 3=5

\1-grams:
-0.6690068    </s>
-99  <s>  -0.1091445
-0.8450981    a    -0.2340832
-0.8450981    b    -0.1962946

……

\2-grams:
-0.60206  <s> a
-0.60206  <s> c
……

……

\end\
```

提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

神经网络语言模型

- 神经网络语言模型将语言模型利用神经网络建立为一个分类任务：已知上文信息，预测下一个词出现的概率。

- 由于神经网络在连续空间建模，所以不需要显式地进行平滑。

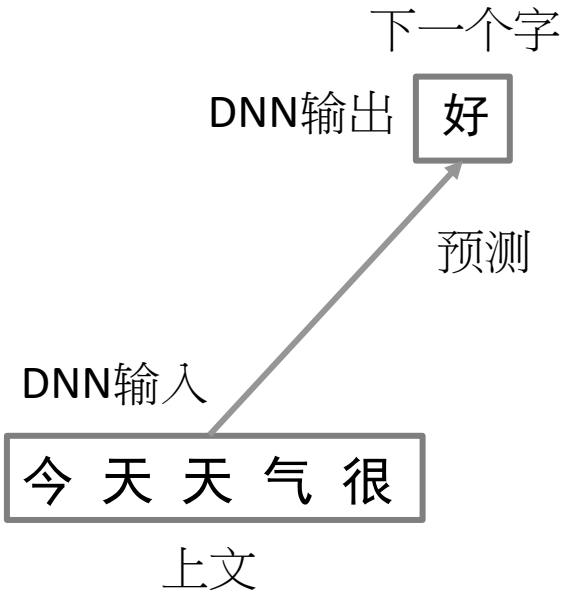
- 特别是由于循环神经网络利用隐状态编码上文所有历史信息，近年来已被证明其性能远远超过N元语法语言模型。

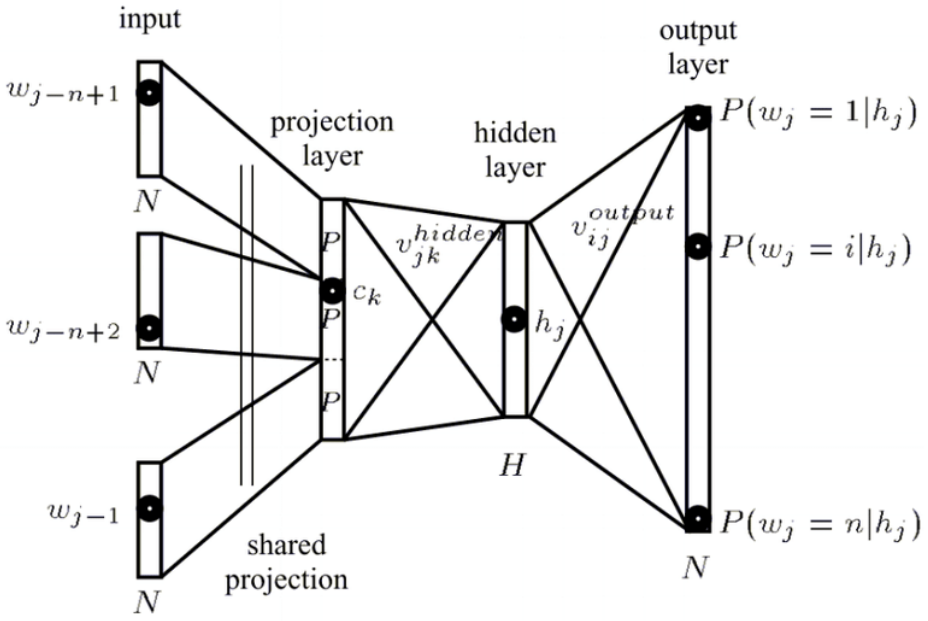
前馈神经网络语言模型

- 前馈神经网络语言模型类似于N元语法，根据前N-1个词，预测下一个词发生的概率。
- 早期的前馈神经网络语言模型直接将每一个词进行one-hot编码，然后输入到前馈神经网络，预测下一个词。
- 网络一般为3~4层，最后一层为Softmax函数，这样网络的输出为词表上的概率分布。
- 训练时，将语料中所有N元语法构成样本，然后以前N-1个词作为输入，第N个词作为标准答案进行训练。训练准则采用交叉熵。

前馈神经网络语言模型

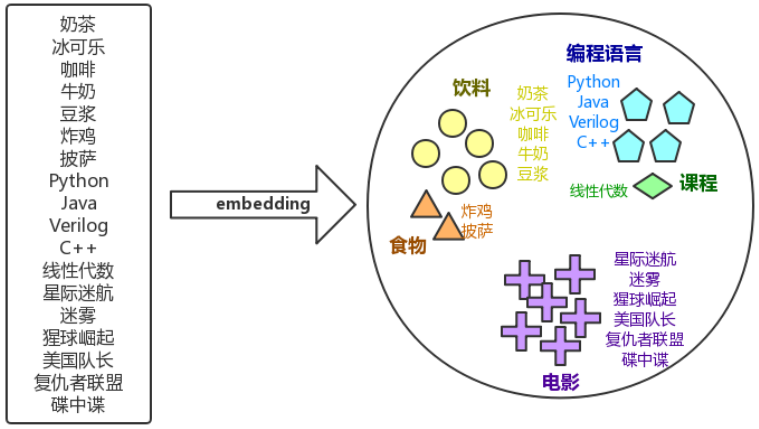
- 如图所示，将前N个词的one-hot编码拼接起来，输入到神经网络中，输出层为下一个位置在词表的概率分布。
- 然而One-hot编码只是作为符号，其本身不能表达词语的关系，即one hot编码没有语义信息。





词嵌入

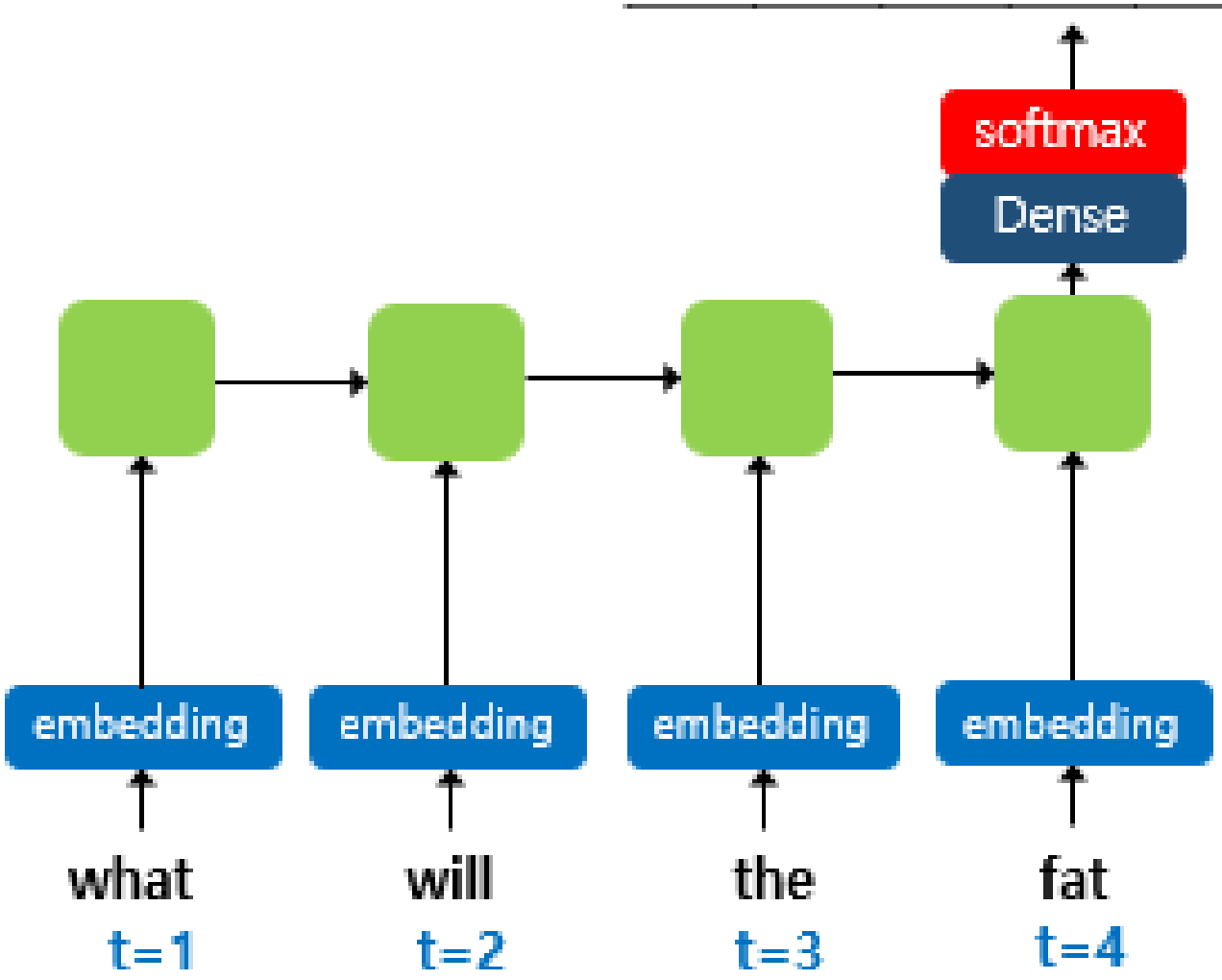
- 词嵌入（word embedding）是指用一个实数向量去表示一个词，也称为词的分布式表示（distributional representation）。
- 一般，词嵌入通过训练语言模型获得。开始时随机初始化，随着网络的训练，词向量会表现出“语义”。具体来说，词义相近的词，词嵌入的欧氏距离也会相近。
- 训练得到的词嵌入，可以进一步地用于下游任务，比如文本分类，词性标注等。
- 也有专门训练词嵌入的方法，如CBOW，Skip-Gram等。



循环神经网络语言模型

- 基于前馈神经网络的语言模型根据固定的前N-1个词去预测下一个词，这样上文信息总是一个固定长度的。
- 然而，真实的句子中，词之间往往存在长距离依赖，此时前馈网络就难以对长距离依赖进行建模。
- 循环神经网络将历史上下文编码为隐状态，将其与当前词同时输入神经网络，来预测下一个词。
- 理论上可以记忆无限长的上下文。

循环神经网络语言模型


softmax
Dense
embedding
embedding
embedding
embedding
what
t=1
will
t=2
the
t=3
fat
t=4

循环神经网络语言模型

- 形式化地，循环神经网络语言模型可以表示为如下形式

$$P(x_{t+1}|x_t, c_t) = f(x_t, c_t)$$

$$c_{t+1} = g(x_t, c_t)$$

- 其中，x 表示输入的词，c 表示历史信息的编码，t 表示时刻。历史信息通过函数 g 来进行更新，通过当前输入 x_t 和历史信息 c_t 来预测下一时刻词发生的概率分布。

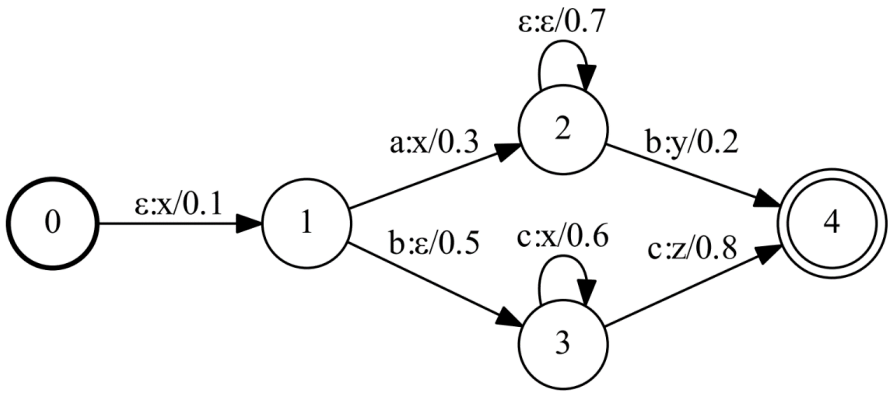
- 循环神经网络的隐层单元可以采用前文提到的长短时记忆网络(Long-short term memory, LSTM)或门控循环单元(Gated Recurrent Unit, GRU)来实现。

提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

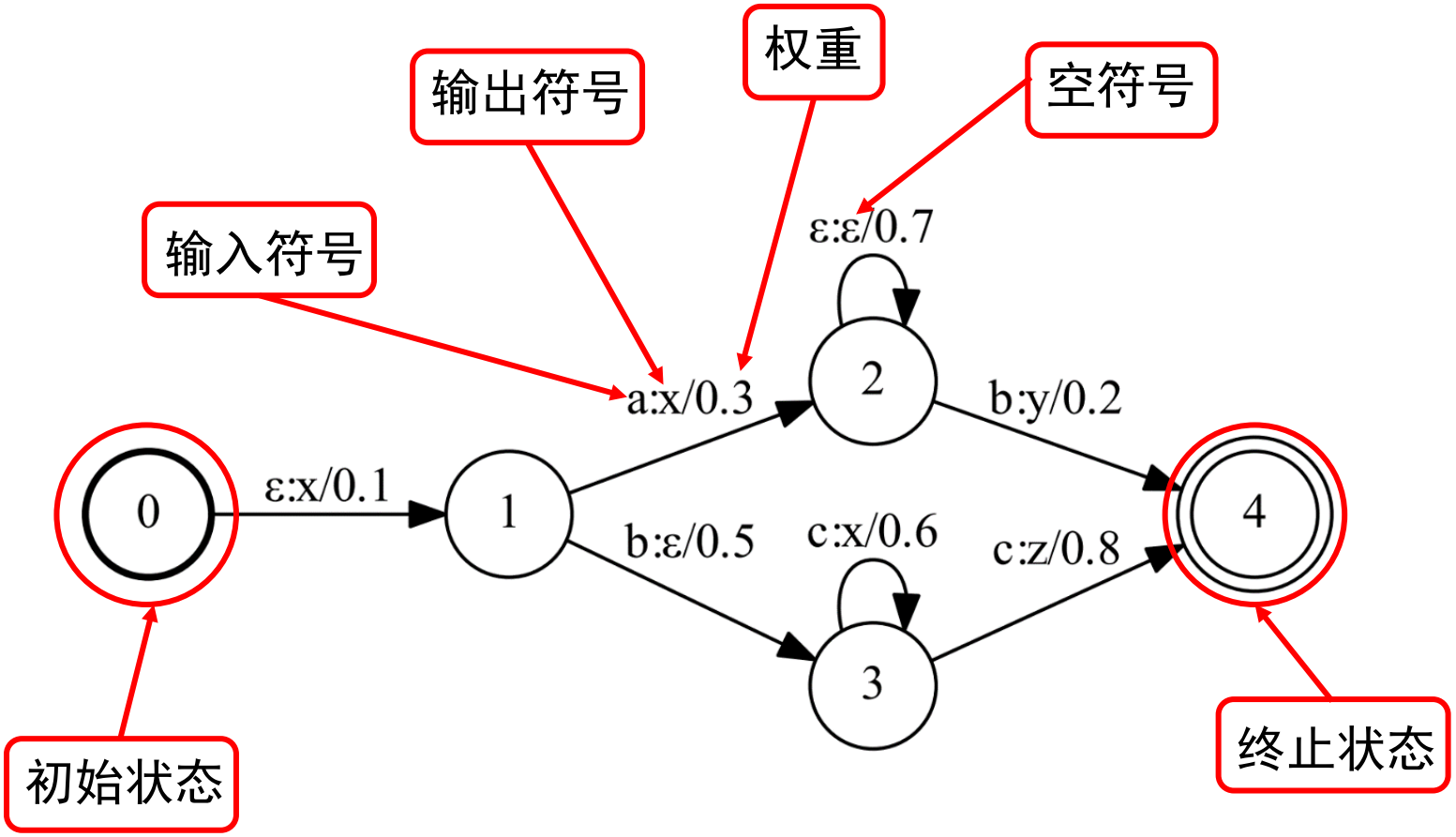
加权有限状态转换器

- 加权有限状态转换器是一种图(Graph)。



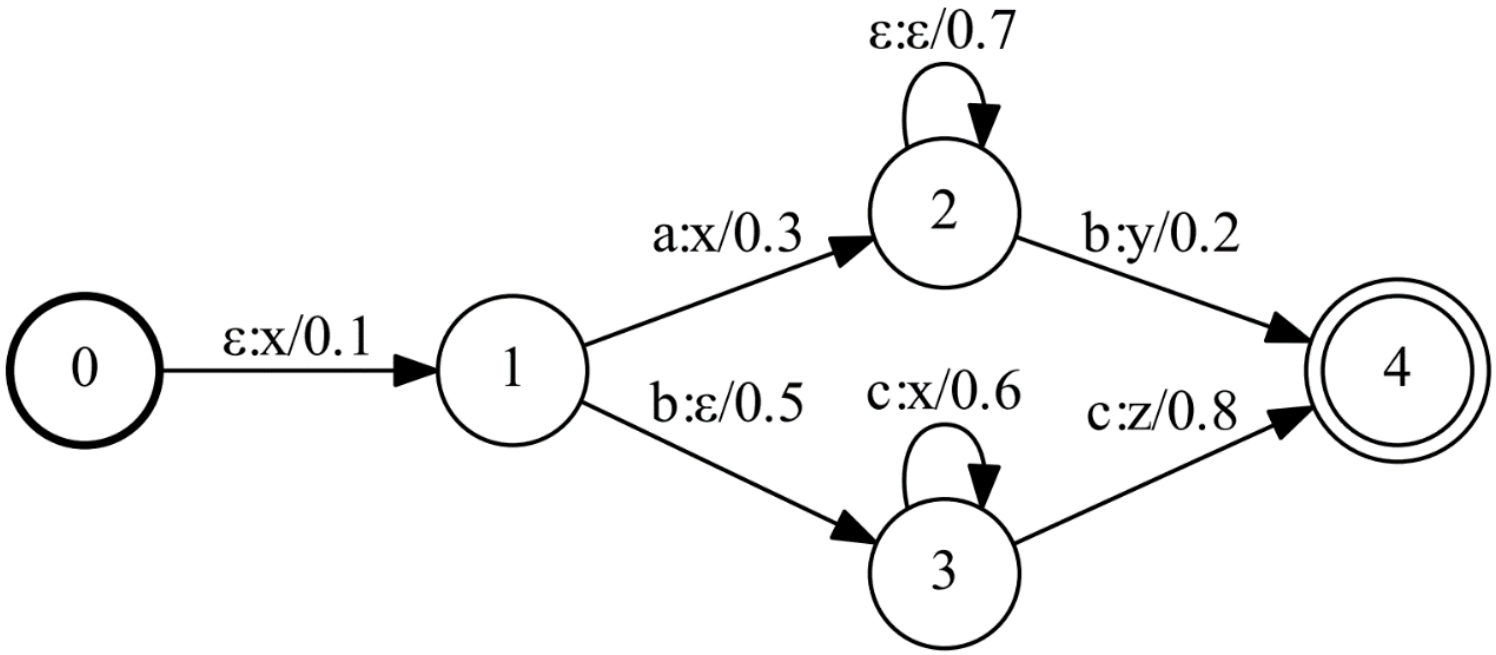
- 图的节点(node)表示状态(state)，边表示输入符号，输出符号，以及这条边的权重。
- 加权有限状态转换器即是通过这种方式，来表示输入符号序列到输出符号序列的转换。
- 加权有限状态转换器是有限状态自动机的推广。

加权有限状态转换器


输出符号
权重
空符号
输入符号
ε:ε/0.7
a:x/0.3
2
b:y/0.2
0
ε:x/0.1
1
4
b:ε/0.5
c:x/0.6
c:z/0.8
3
初始状态
终止状态

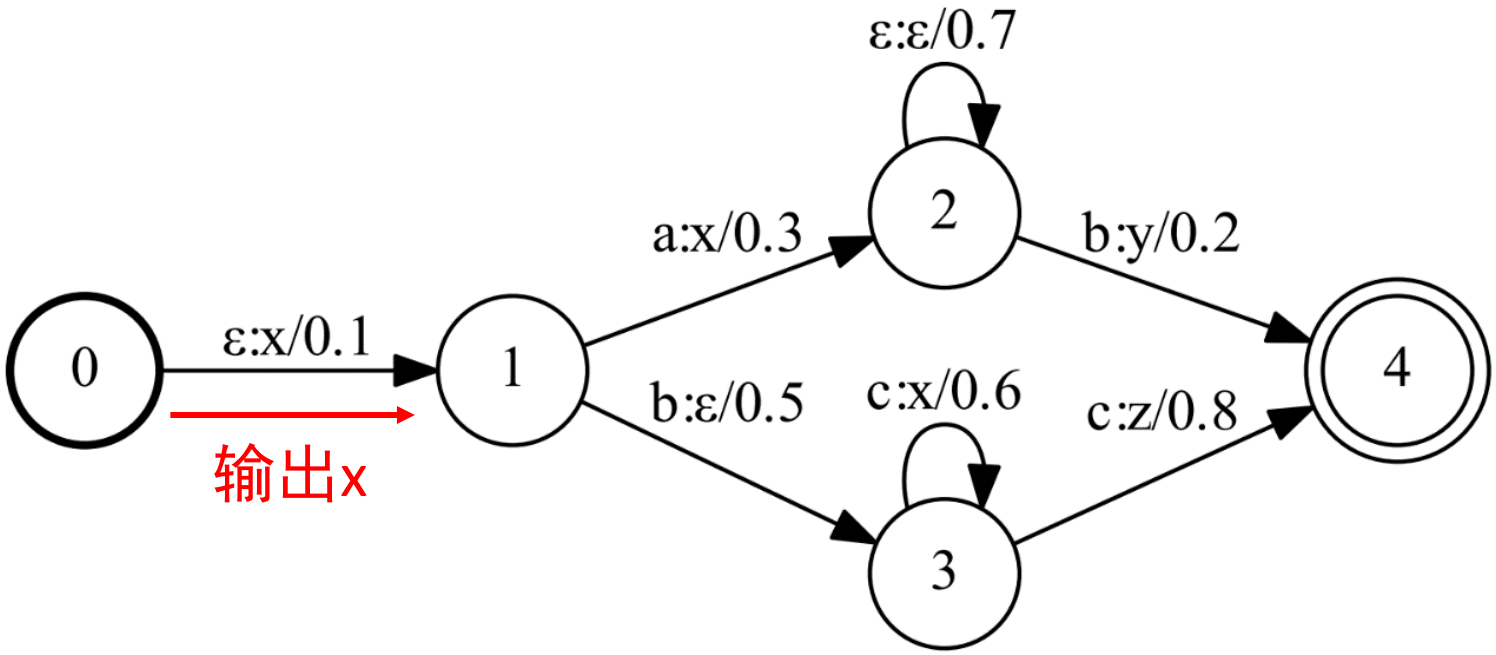
例子

- 输入序列 b c c



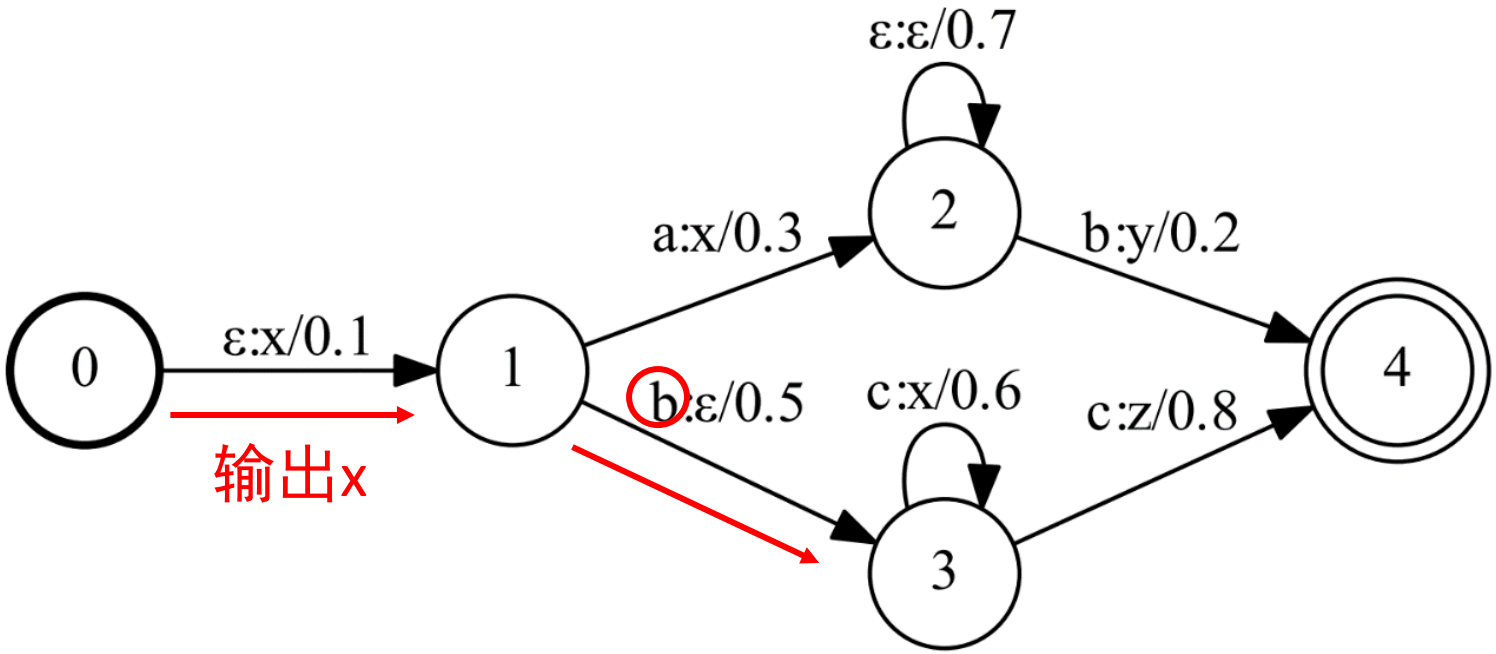
例子

- 输入序列 b c c



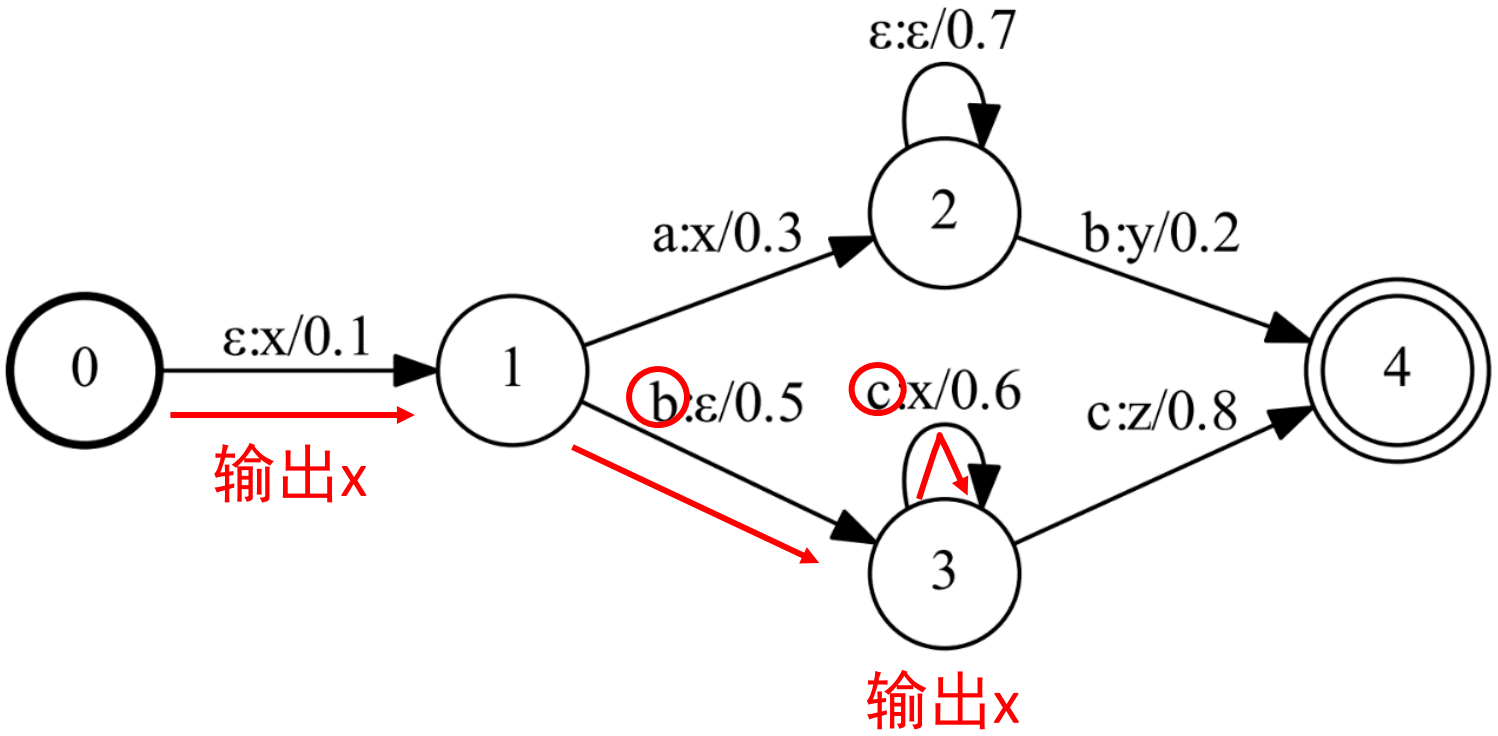
例子

- 输入序列 b c c



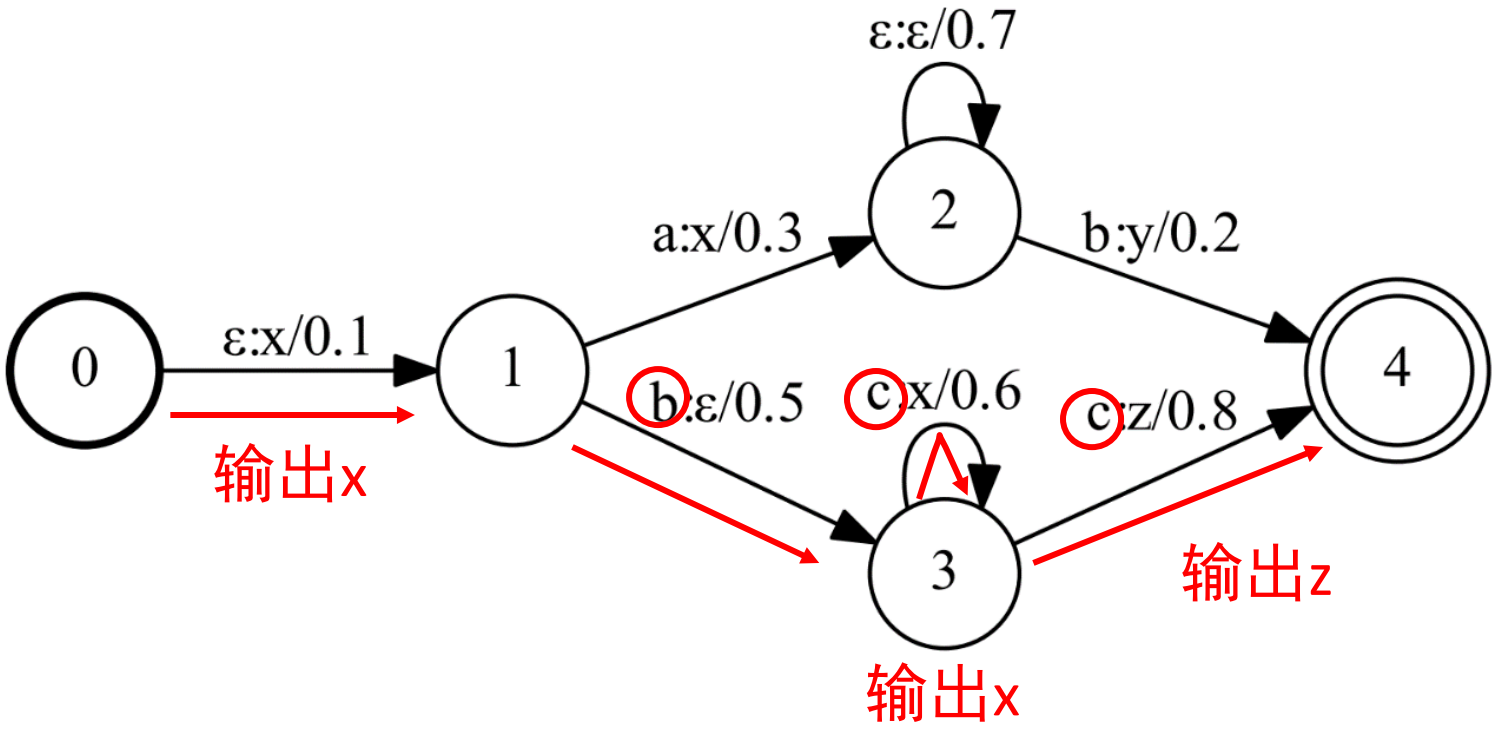
例子

- 输入序列 b c c



例子

- 输入序列 b c c
- 抵达终止状态，得到输出序列x x z



加权有限状态转换器

- 形式化地，加权有限状态转换器是一个八元组(8-tuple)

$$T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$$

  - Σ 表示有限的输入符号集合；
  - Δ 表示有限的输出符号集合；
  - Q 表示有限的状态集合；
  - $I \subseteq Q$ 表示初始状态集合；
  - $F \subseteq Q$ 表示终止状态集合；
  - E 表示转移集合；
  - λ 表示初始权重函数，即各个初始状态的权重分配；
  - ρ 表示终止权重函数，即各个终止状态的权重分配。

加权有限状态转换器

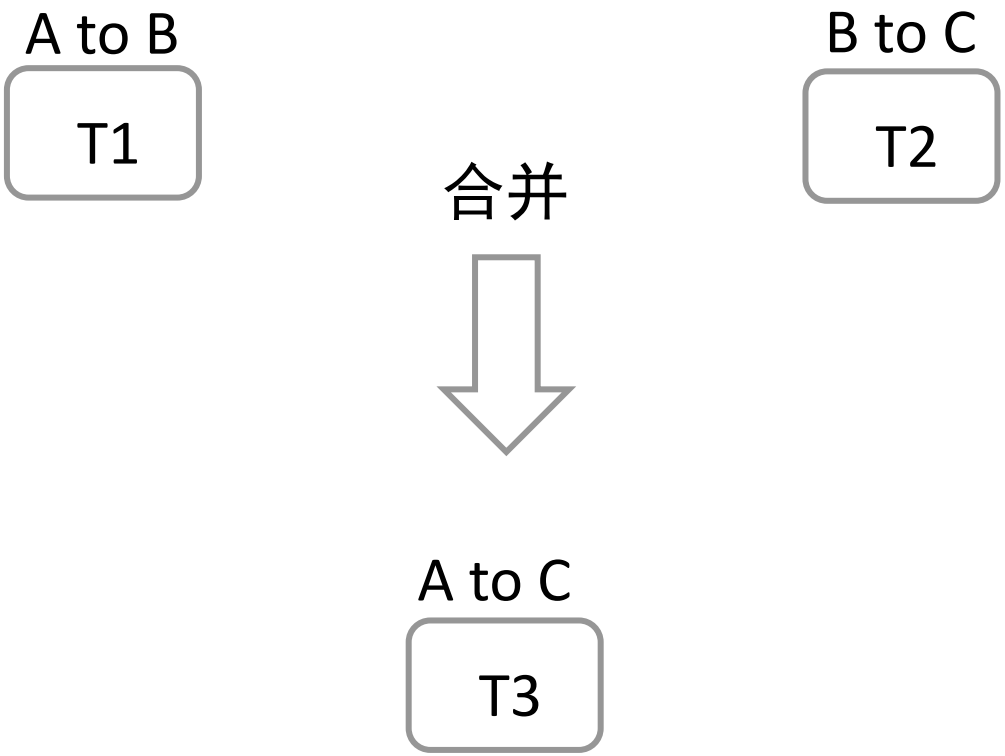
- 加权有限状态转换器可以方便地表示两个符号串之间的转换。
- 更重要的是，加权有限状态转换器的三个重要操作很适用于不同层级知识的融合
  - 合并操作
  - 确定化操作
  - 最小化操作
- 利用加权有限状态转换器可以很方便地把语言模型，发音词典，三音素，以及HMM结合起来。

提纲

- GMM-HMM 声学模型
- 基于深度学习的声学模型
  - DNN-HMM混合声学模型
  - 端到端声学模型
- 语言模型
  - N元语法
  - 神经网络语言模型
- 解码基础
  - 加权有限状态转换器的概念
  - 加权有限状态转换器的操作

合并操作

- 合并操作是加权有限状态转换器最重要的操作。



合并操作

- 考虑3个加权有限状态转换器

$$T_1 = (A, B, Q_1, I_1, F_1, E_1, \lambda_1, \rho_1)$$

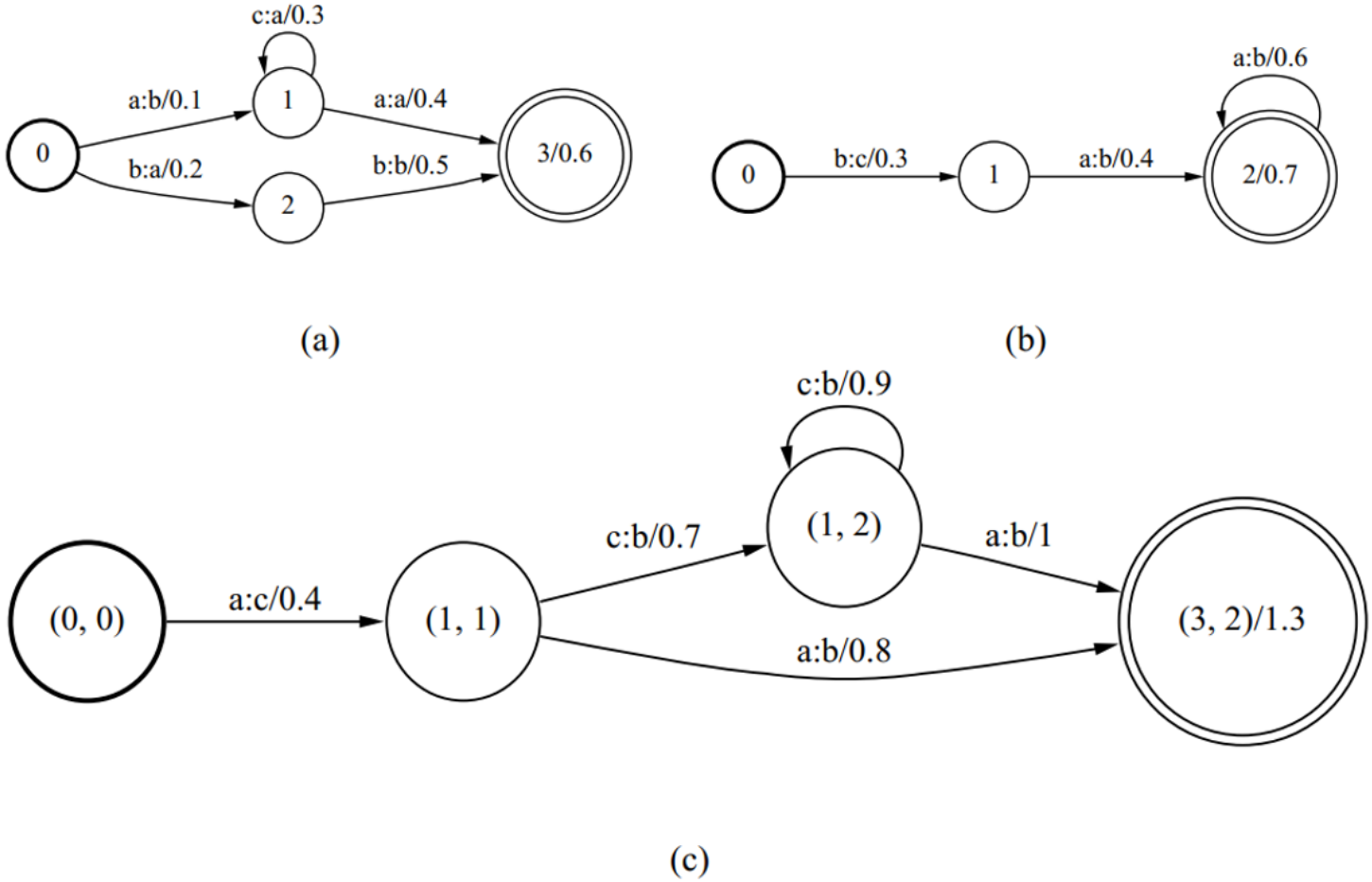
$$T_2 = (B, C, Q_2, I_2, F_2, E_2, \lambda_2, \rho_2)$$

$$T = (A, C, Q, I, F, E, \lambda, \rho)$$

- 任取3个符号串 $x \in A^*$，$z \in B^*$，$y \in C^*$。
- T_1 首先将 x 转换为 z，T_2 又将 z 转换为 y。
- T 直接将 x 转换为 y。
- 非正式地说，如果二者的结果一样，那么就可以说 T 是 T_1 与 T_2 组合得来的。

合并操作

- 合并操作的一个例子

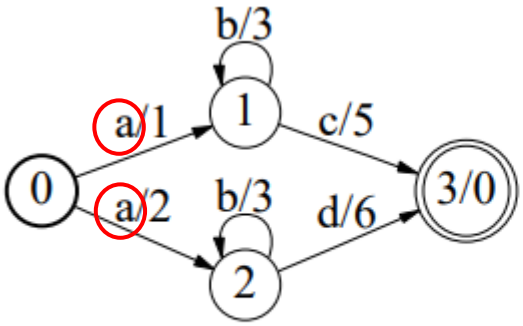


确定化

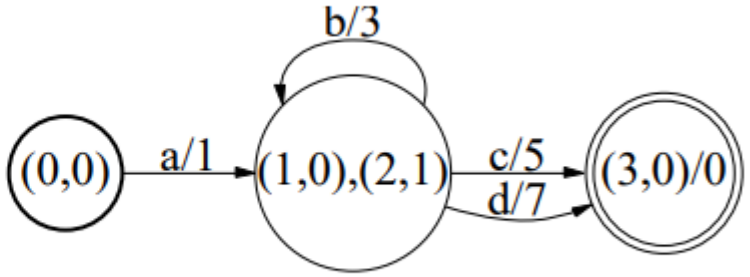
- 如果对任意一个输入序列，转换器输出唯一的一个输出序列，则称这个转换器是函数化的（functional）。
- 对于两个函数化的加权有限状态转换器，如果对任意的输入符号串，二者都以相同的权重输出相同的输出符号序列，那么这两个加权有限状态转换器是等价的。
- 如果一个加权有限状态转换器只有一个初始状态，并且同一状态的任意两个转移，输入符号都不同，那么这个转换器是确定的。
- 确定化操作就是将一个（函数化的）非确定有限状态转换器转换为一个等价的确定有限状态转换器。

确定化

- 确定化的例子
- 为了简单起见，这里的例子是一个自动机，没有输出符号。



非确定的 (a)



等价的确定的 (b)

最小化

- 一般来说，加权有限状态转换器可以有不同的形式。
- 其中存在一种形式，其状态数和转移数是最小的。
- 找到加权有限状态转换器的最小形式，可以大大压缩其体积。
- 最小化操作就是将一个转换器转换为等价的转换器中状态数和转移数最小者。

最小化

算法 2.3 FSA最小化算法

输入：确定化的$A = (Q, \Sigma, E, i, F)$

输出：最小化的WFST：A'

1.找到Q中两两等价的状态，把他们按照等价的传递性划分成等价的状态集合

2.以等价状态集合来构造最小化的WFST，方法如下：

设A的状态映射函数为δ，A'的为γ

若一个状态q属于X类等价集合，且$\delta(p, a) = q$为Y类等价集合

可以在A'上建立一个映射$\gamma(X, a) = Y$

A'中的状态，可以用等价的状态集合替换

另外还有如下事实：

1).A'的初始状态是包含A的初始状态i的等价集合

2).A'的终止状态是包含A的终止状态$f \in F$的等价集合

加权有限状态转换器的工具

- 加权有限状态转换器包括其数据结构以及对应的一批算法。
- Mohri等人提供了开源的加权有限状态转换器工具
  http://www.openfst.org/twiki/bin/view/FST/WebHome

解码分类

- 动态解码 (Prefix Tree)
  - 动态加载模型
  - 需要搜索模型的信息
  - 内存消耗少、构建快
  - 解码速度慢
- 静态解码(WFST)
  - 采用加权有限状态转换器构建解码空间
  - 一次加载完模型
  - 不需要搜索模型的信息
  - 内存消耗多、构建慢
  - 解码速度快

小结

- 本节课介绍了语音识别中的GMM-HMM，DNN-HMM，语言模型，以及加权有限状态转换器。
- GMM-HMM和在其基础上发展的DNN-HMM是重要的语音识别声学建模技术，通过HMM来建模语音动态特性，用GMM或DNN来建模观测概率。后期出现了不显式进行HMM建模，而直接使用神经网络建模的CTC训练技术。这些技术大大提升了语音识别的性能。
- 语言模型是建模句子语法的方法。统计语言模型计算一句话的概率。语言模型可以帮助识别出符合人类语法的词序列。
- 加权有限状态转换器一种图的表示方法，是构建解码空间的基础构件。

谢谢！