Lattice: Concepts, Methods and Applications

2021-07-01

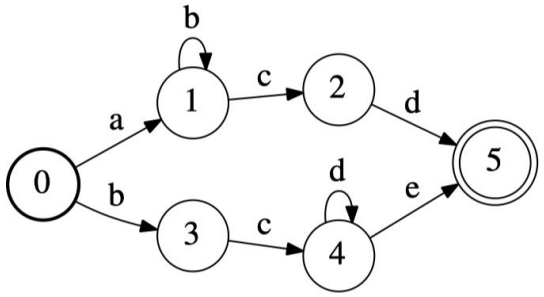
目录

基础概念

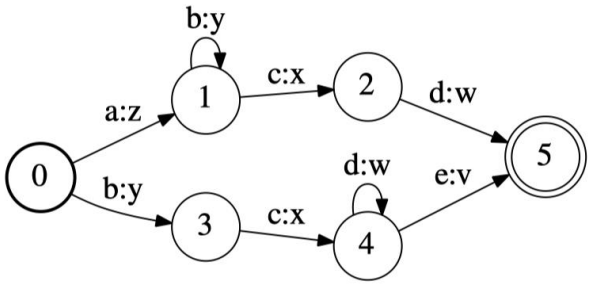
1. 加权有限状态机 WFST
2. 基于 WFST 的语音识别解码 (以 DNN-HMM 为例)
3. Lattice 的基本概念

回顾：加权有限状态机 WFST

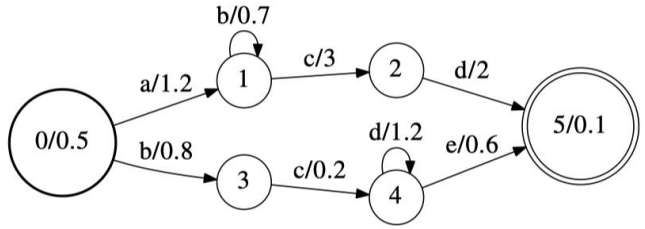
WFST：Weighted Finite-State Transducer



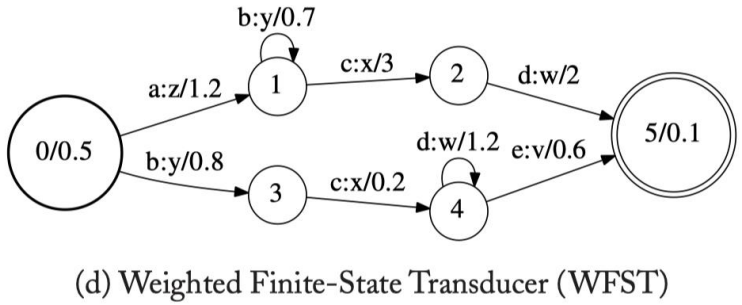
(a) Finite-State Acceptor (FSA)



(b) Finite-State Transducer (FST)



(c) Weighted Finite-State Acceptor (WFSA)



(d) Weighted Finite-State Transducer (WFST)

数学定义

元素	含义
Σ	有限输入标签集合
Δ	有限输出标签集合
Q	有限状态集合
I	初始状态集合
F	终止状态集合
E	有限状态转移弧集合
λ	初始状态权重函数
ρ	终止状态权重函数

WFST 是 $(\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ 定义的 8 元组

回顾：基于 WFST 的语音识别解码器

基于 WFST 的语音识别解码器（以DNN-HMM为例）

功能分解

组成部分	输入标签序列	输出标签序列	功能说明
G.fst	word	word	WFSA，将语言模型的概率引入到解码图中
L.fst	phone	word	发音词典建立起音素序列和词之间的转换关系
C.fst	biphone/triphone	phone	音素与三音素之间的对应关系
H.fst	HMM transition	biphone/triphone	三音子状态的 transition 与三音素之间的关系

构建解码图

$$\mathbf{HCLG = asl\ (\ min\ (\ rds\ (\ det\ (\ H'\ o\ min\ (\ det\ (\ C\ o\ min\ (\ det\ (\ L\ o\ G\ )\ )\ )\ )\ )\ )\ )\ )\ )}$$

asl ==“add-self-loops”and rds==“remove-disambiguation-symbols”, H’ 是没有自环的H.fst

Lattice 的基本概念及特点

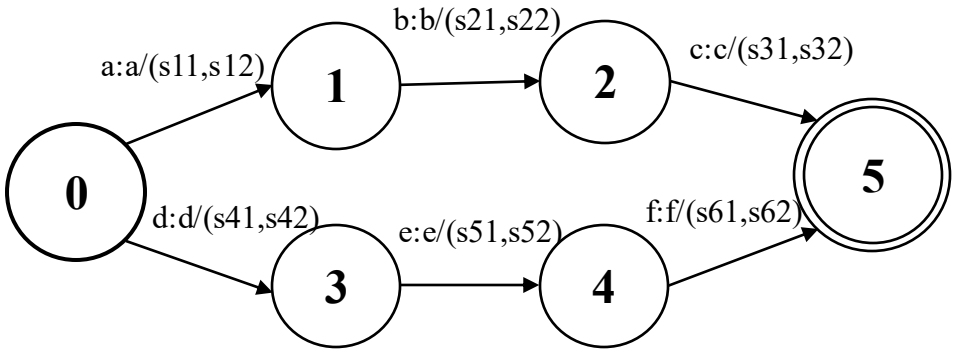
Lattice 的生成过程简述

输入输出

- 从 HCLG.fst 的起始点出发，处理发射弧和非发射弧
- HCLG.fst 的输入标签是 transition_id，经过transition model 得到 pdf_id，从声学模型该帧前向得到的向量中，取出 pdf_id 对应的后验概率/似然分数 (取负对数) 作为 acoustic cost
- 在 HCLG.fst 上解码时每条弧上本身也是有权重 / cost 的，表示 graph cost， transition_id 经过图达到解码结束时路径上从FST上累积的cost

Lattice 的定义

- Lattice 是存储解码中间结果的数据结构
- Lattice 存储帧级别的解码信息，输入标签是 transition_id，输出标签是word
- Word Lattice 存储词级别的解码结果，是一种特殊的 WFSA 结构，不仅保留了词表，还有权重 (acoustic cost和graph cost)



数学定义

元素	含义
Σ/Δ	有限(输入/输出)标签集合
Q	有限状态集合
I	初始状态集合
F	终止状态集合
E	有限状态转移弧集合

Lattice 的基本概念及特点

Lattice 示例

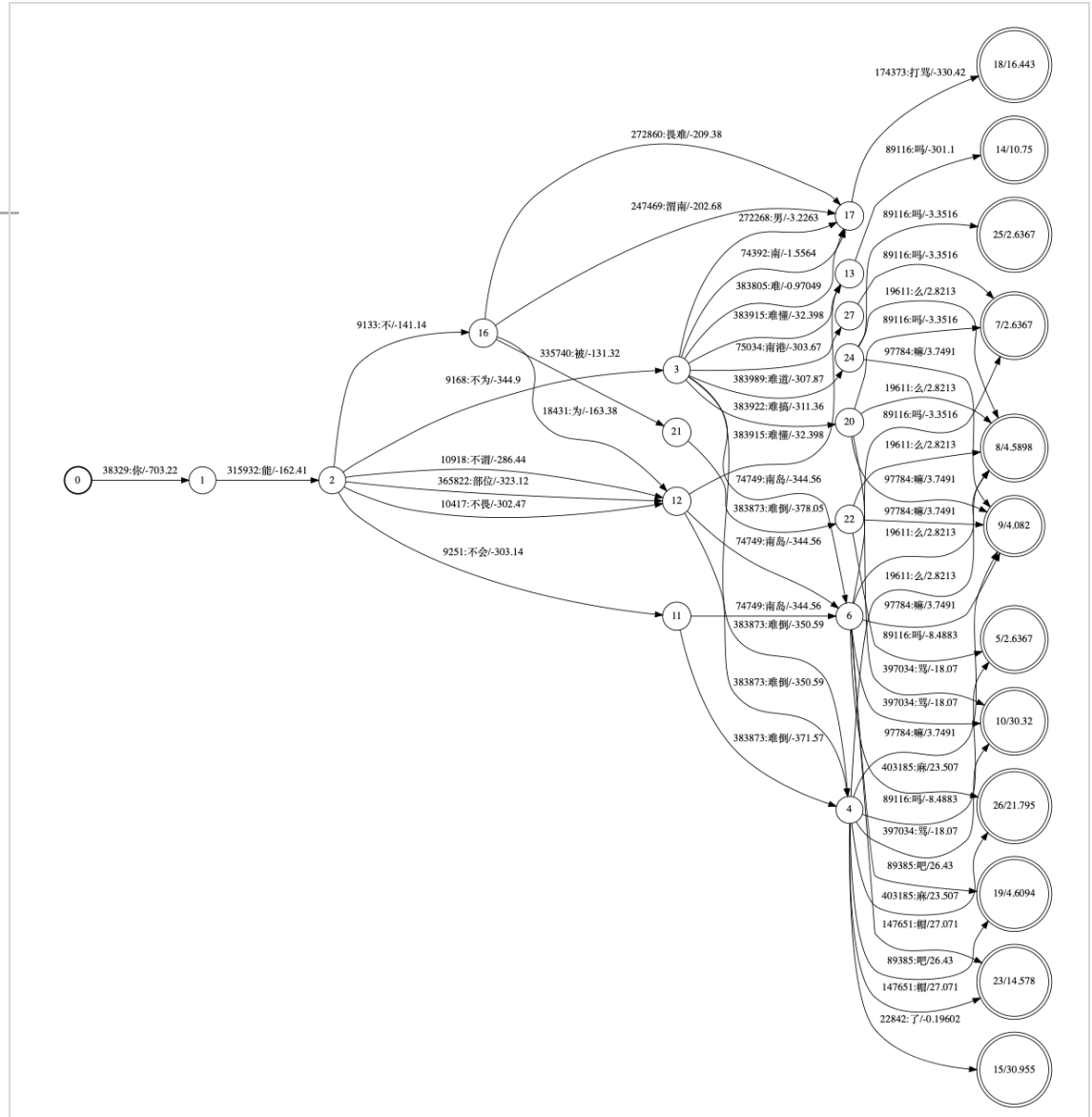
举例

语音实际内容：你能不被难倒吗

ASR 识别结果：你能不为难倒吗

总结：Lattice 的特点

1. Lattice 是一种特殊的 WFST 结构
2. 采用图结构存储解码搜索中的多候选结果
   相比于线性的多候选 N-Best，保留候选结果更多、更 compact
3. 保留了解码过程中的分数信息
   便于模型性能诊断debug；额外的分数信息可以用于后处理或者作为下游任务的补充信息



解码的词级别 Lattice

Lattice在ASR中的应用

声学模型

1. 基于 Lattice 的区分性训练
2. 基于置信分数的半监督学习

语言模型

1. 声学/语言模型重打分
2. 以 Lattice 为输入的二次解码

Lattice 在 ASR 中的应用：声学模型-区分性训练

- 语音识别任务目标：**O(observation)**是输入声学特征观测序列，**W**是字/词序列

$$\hat{W} = \arg\max_{W}\{P(W|O)\}$$

- **贝叶斯定理**

声学模型 语言模型

$$\hat{W} = \arg\max_{W}\{P(W|O)\} = \arg\max_{W}\left\{\frac{P(X|O)\,P(W)}{P(O)}\right\} = \arg\max_{W}\{P(O|W)\,P(W)\}$$

$$\hat{W} = \arg\max_{W}\{P(W|O)\}$$

$$\hat{W} = \arg\max_{W}\{P(O|W)\,P(W)\}$$ 贝叶斯公式

$$P(O|W) = \sum_{L} P(O|L)P(L|W)$$ 发音词典建模

$$P(O|L) = \sum_{C} P(O|C)P(C|L)$$ 三音子模型

$$P(O|C) = \sum_{S} \boldsymbol{P(O|S)}P(S|C)$$ 决策树状态聚类

$$\hat{W} = \arg\max_{W}\{\sum_{L}\sum_{C}\sum_{S} \boldsymbol{P(O|S)}P(S|C)P(C|L)P(L|W)\,P(W)\}$$

final.mdl H C L G

Lattice 在 ASR 中的应用：声学模型-区分性训练

$$\hat{W} = \arg\max_{W} \{ \sum_{L} \sum_{C} \sum_{S} \underbrace{\boldsymbol{P(O|S)}}_{\text{final.mdl}} \underbrace{P(S|C)}_{\text{H}} \underbrace{P(C|L)}_{\text{C}} \underbrace{P(L|W)}_{\text{L}} \underbrace{P(W)}_{\text{G}} \}$$

$$P(O|S) = a_{s_o s_1} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} b(o_t|s_t)$$

- 各状态之间的转移概率a_{ij} → transition model
- 发射概率/观测概率 $b(o_t|s_t)$

$$b(o_t|s_t) = \frac{P(s_t|o_t)P(o_t)}{P(s_t)}$$

- $P(s_t)$：不同状态的先验概率，从训练数据的对齐结果中统计得到
- $\boldsymbol{P(s_t|o_t)}$：给定声学特征下对应三音子状态的概率

交叉熵损失函数 $\mathcal{F}_{CE} = -\sum_{u=1}^{U} \sum_{t=1}^{T_u} \log y_{ut}(s_{ut})$

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- Maximum Mutual Information (MMI)

$$\theta_{ML} = \arg\max_{\theta} \{ P_\theta(O|W) \}$$

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \sum_u \log P_\theta(O_u|W_u)$$

$$\theta_{MMI} = \arg\max_{\theta} \{ P_\theta(W|O) \}$$

$$\theta_{\mathrm{MMI}} = \arg\max_{\theta} \sum_u \log P_\theta(W_u|O_u)$$

$$= \arg\max_{\theta} \sum_u \log \frac{P_\theta(O_u|W_u)P(W_u)}{P(O_u)}$$

$$= \arg\max_{\theta} \sum_u \log \frac{P_\theta(O_u|W_u)P(W_u)}{\sum_W P_\theta(O_u|W)\,P(W)}$$

$$\mathcal{F}_{MMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)}$$

κ是acoustic scaling factor

Lattice-based MMI

- 分母：不仅与正确标注有关，还和所有可能的词序列W有关
- 在已有的声学模型的基础上，修改目标函数，利用解码结果基于新的目标函数训练模型参数
- 相当于最大化正确标注的条件对数似然/概率，最小化其他错误序列

核心思想：
用解码生成的Lattice，来**近似**语音可能对应的所有词序列 (分母部分)

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- MMI → bMMI (boosted MMI)

$$\mathcal{F}_{MMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)}$$

$$\mathcal{F}_{BMMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W) e^{-b\,A(W,W_u)}}$$

$A(W, W_u)$ 表示两个序列间的相似性，或者W与W_u相比的准确率
b是增强系数，相当于给准确率低的W更大的权重

- MPE (Minimum Phone Error) / sMBR (state-level Minimum Bayes Risk)

MPE： phone-level
sMBR： state-level

$$\mathcal{F}_{MBR} = \sum_u \frac{\sum_W p(\mathbf{O}_u|S)^\kappa P(W) A(W, W_u)}{\sum_{W'} p(\mathbf{O}_u|S)^\kappa P(W')}$$

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- 实验结果

Table 3: Results (% WER) of the DNNs trained on the full 300 hour training set using different criteria.

System	Hub5 '00			Hub5 '01			
	SWB	CHE	Total	SWB	SWB2P3	SWB-Cell	Total
GMM BMMI	18.6	33.0	25.8	18.9	24.5	30.1	24.6
DNN CE	14.2	25.7	20.0	14.5	19.0	25.3	19.8
DNN MMI	12.9	24.6	18.8	13.3	17.8	23.7	18.4
DNN sMBR	12.6	24.1	18.4	13.0	17.7	22.9	18.0
DNN MPE	12.9	24.1	18.5	13.2	17.7	23.4	18.2
DNN BMMI	12.9	24.5	18.7	13.2	17.8	23.5	18.3

问题：
需要预先训练一个 DNN-CE 声学模型，还需要对训练集解码，能不能直接建模不训练 CE 模型？

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- Lattice Free MMI (LF-MMI / chain)

$$\mathcal{F}_{MMI} = \sum_u \log \frac{p(\mathbf{O}_u|S_u)^\kappa P(W_u)}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)}$$

- 分母部分：
  - MMI 是真正的解码，使用1-gram/2-gram的词语言模型
  - LF-MMI 将分母解码使用的语言模型从词级别转换为音素级别
  - 使用训练集**的音素对齐**结果训练4-gram音素LM，不做平滑和裁剪
  - compose成音素级别的解码图HCP (den.fst/分母图)
  - 所有句子使用同一个分母图，普通MMI每个句子各自解码得到分母图

- 分子部分：
  - 每个句子对齐得到音素级的lattice，包含时间信息
  - 音素级 lattice 转换成fst，称为分子图
  - 分子图输出标签是 pdf_id，生成训练会用的 egs
  - 根据时间信息切成片段（1-1.5s），能够加速训练

- 训练目标函数是 CE + MMI 的双 loss，CE 权重通常为0.1，对模型收敛有帮助

- Lattice-Free，是指：分母部分 (denominator) 的词级别 Lattice 的 Free
- 分子部分仍然使用**音素级别的 Lattice**
- 完全不依赖对齐 Lattice 结果的端到端训练，可以参考 end-to-end LF-MMI 的工作，本质是一种类似于 CTC 但是更复杂的序列 Loss，比 CTC Loss 更优

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- LF-MMI / LF-bMMI / LF-sMBR

Table 4: Performance of LF-MMI on various LVCSR tasks with different amount of training data, using TDNN acoustic models

Database	Size	WER		
		CE	CE → sMBR	LF-MMI
AMI-IHM	80 hrs	25.1	23.8	22.4†
AMI-SDM	80 hrs	50.9	48.9	46.1†
TED-LIUM	118 hrs	12.1	11.3	11.2*
Switchboard	300 hrs	18.2	16.9	15.5
Librispeech	1000 hrs	4.97	4.56	4.28
Fisher + SWBD	2100 hrs	15.4	14.5	13.3

Table 4. WERs of TDNN-LSTMP LF-MMI baseline and LF-bMMI on Eval2000 using Switchboard+Fisher-2100hrs data

Models	b	WERs(%)		
		SWB	CH	Total
LF-MMI	0.0	8.1	15.5	12.0
LF-bMMI	0.05	8.1	15.2	11.7
LF-bMMI	0.10	**7.7**	**14.7**	**11.3**
LF-bMMI	0.15	7.9	14.9	11.5

Table 6: *WERs (%) for CSJ evaluation set.*

Criterion	E1	E2	E3	avg.
(4gram-LM)				
LF-MMI	8.51	6.94	6.94	7.46
LF-MMI → LF-sMBR	**8.41**	**6.72**	**6.86**	**7.33**
(4gram-LM + RNN-LM rescoring)				
LF-MMI	**7.43**	6.38	6.41	6.74
LF-MMI → LF-sMBR	7.49	**6.22**	**6.16**	**6.62** (*)

(*) Character error rate (CER) was 5.03% (E1: 5.67%, E2: 4.90%, E3: 4.53%)

Table 7: *WERs (%) for LibriSpeech evaluation set.*

Criterion	clean	other
LF-MMI	3.72	8.69
LF-MMI → LF-sMBR	**3.68**	**8.57**

Lattice 在 ASR 中的应用：声学模型-区分性训练

区分性训练 Discriminative Training

- LF-MMI + sMBR

思想：LF-MMI当成预先训练的模型，对全部训练数据解码，再进行基于 Lattice 的区分性训练

Table 2: Comparison of objective functions on Hub5 '00 eval set, using SWBD-300 Hr data

Objective function	Model (Size)	WER	
		Total	SWBD
CE	TDNN-A (16.6 M)	18.2	12.5
CE → sMBR	TDNN-A (16.6 M)	16.9	11.4
LF-MMI	TDNN-A (9.8 M)	16.1	10.7
	TDNN-B (9.9 M)	15.6	10.4
	TDNN-C (11.2 M)	15.5	10.2
LF-MMI → sMBR	TDNN-C (11.2 M)	15.1	10

Table 2. MR-WER (%) results on the submitted MGB-3 systems

	System	dev	test
Unadapted	without sMBR	47.42	-
	with sMBR	47.32	-
Adapted	LF-MMI	35.97	-
	LF-MMI + sMBR	35.15	-
	Primary	33.41	32.78

个人实践结论：
1. LF-MMI 的 chain 模型在 ASR 任务上明显优于 CE 的 nnet3 模型
2. LF-MMI + sMBR 的区分性训练方案耗时很长，在各测试集上无一致性明显提升，大数据量时暂不建议使用

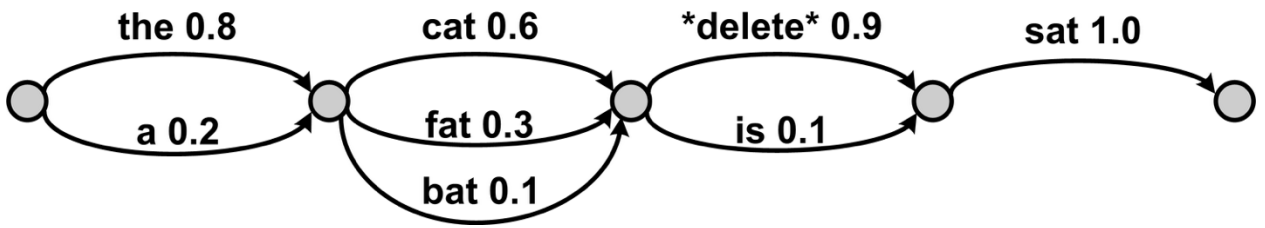
Lattice 在 ASR 中的应用：声学模型-半监督训练

基于置信分数的半监督训练

基于 Lattice 置信分数的**筛选方法**

- 使用已有的模型，对无标注数据生成伪标注文本
- 基于（状态级别/音素级别/词级别）的置信分数筛选数据
  - 状态级别：声学模型对每帧预测的后验概率
  - 音素级别：当前帧某个 phone 的后验概率 = 求和 (声学模型预测的 pdf 概率 × pdf 到这个phone的概率)

- 问题：如何估计词级别的置信分数 → 基于 Lattice 的贝叶斯解码
  - 作用：从 Lattice 中对预测的词进行时间上的对齐和剪枝，形成一种“香肠”结构，称为混淆网络
  - 混淆网络的边上的分数是归一化后具有置信分数的含义，可以设置阈值，筛选置信分数较高的片段



- 缺点：
  - 筛选的阈值/保留的数据量不容易选择，需要反复实验尝试，选择不好模型效果反而变差

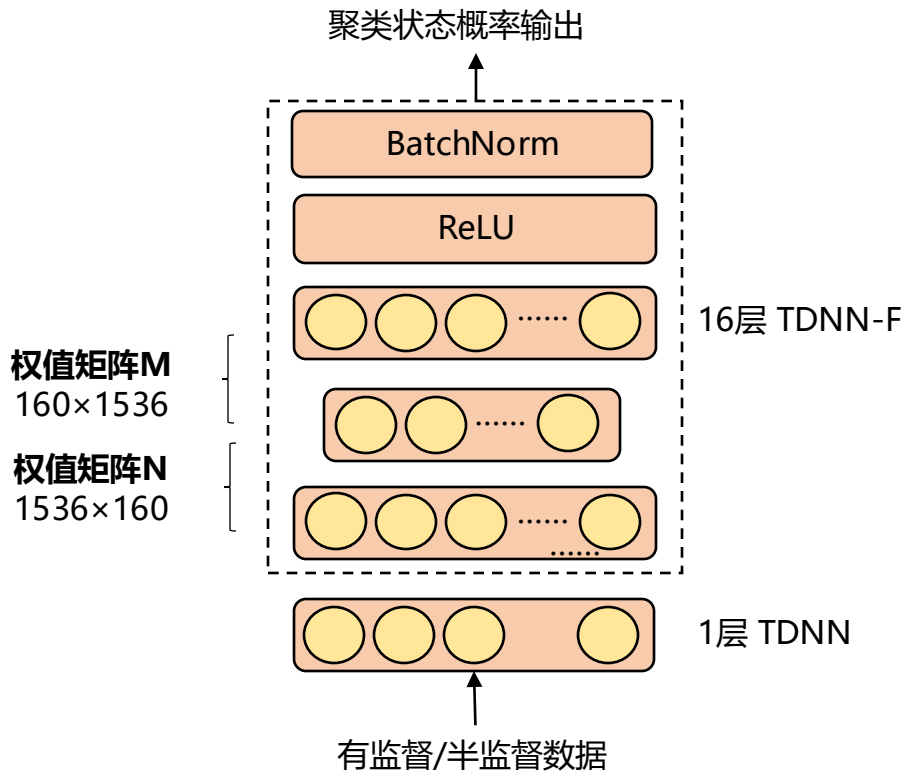
Lattice 在 ASR 中的应用：声学模型-半监督训练

基于置信分数的半监督训练

基于 Lattice 状态级后验概率的**训练**

- 同样是解码获取Lattice，但是不用转成词 Lattice
- 从解码Lattice 中获取最优状态序列作为伪 label，同时使用 lattice 获取对应的后验概率
- 反向传播时，以 Lattice 上计算的后验概率为权重进行参数更新，使得数据有效利用的粒度真正细致到帧级别，而且避免了繁琐的筛选流程。
- 数据加权：
  - 训练 phone-lm 构建 den.fst 时，有监督数据的对齐权重更高
  - 更新 AM 的参数时，无监督数据权重更低

思想：可靠性越高的数据，在参数更新时的重要性越大。
- 数据加权：有监督数据 > 半监督数据
- 帧级别加权：半监督数据中，帧级别的后验概率高的数据，起的作用越大



Lattice 在 ASR 中的应用：声学模型-半监督训练

基于置信分数的半监督训练

实验结果

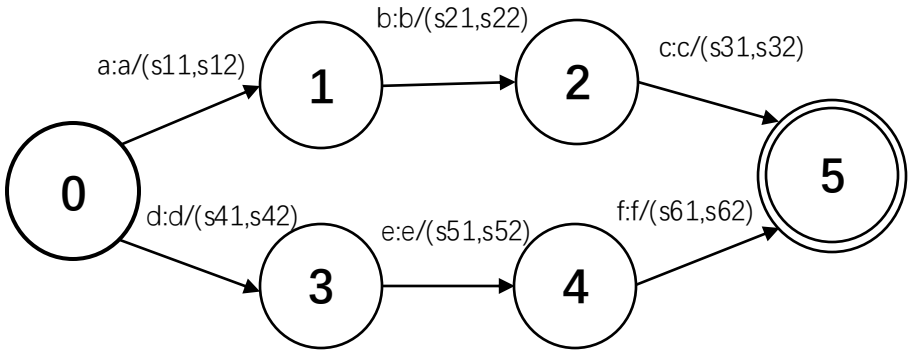
实验组	15h sup + 250h unsup		50h sup + 250h unsup	
	dev	test	dev	test
Baseline	29.41	29.22	22.63	21.97
Best path	23.04	23.23	20.00	19.82
+ frame weight	22.02	21.89	19.60	19.61

个人实践结论：
1. 300h有标注 + 600h半监督，基于 Lattice 的半监督训练方案，CER 有 0.5% 的降低；
2. 使用半监督的伪标注文本，训练弱语言模型与真实标注文本的语言模型插值，CER 继续有0.8%的降低。

Lattice 在 ASR 中的应用：解码/重打分

Lattice 的分数构成

- acoustic_cost：transition_id 对应 pdf_id 在该帧上的输出
- graph_cost：经过 HCLG.fst 解码图的 cost
  - HMM 的 transition 分数
  - L 的 发音分数
  - G 的 语言分数



根据分数的构成，可以分成：

- 声学模型重打分

  用AM-1解码的lattice，可以用AM-2重新计算帧级别的声学分数，再重新在Lattice上搜索最短路径得到one-best结果

- **语言模型重打分**

  在 Lattice 限定的候选结果中，使用更优的语言模型对语言分数重新估计，再从Lattice上重新求One-Best词序列

Lattice 在 ASR 中的应用：解码/重打分

语言模型

语言模型的任务

- 给出一个字/词序列的概率，用于判断语句的合理性大小
- 根据概率的链式法则可以得到：

$$\mathrm{P}(w_1, w_2, \ldots, w_N) = \mathrm{P}(w_1) \prod_{n=2}^{N} P(w_n | w_1^{n-1})$$

- 实际上是研究如何建模：在给定历史序列时预测下一个词的概率

马尔科夫假设 → ngram

- 假设当前词出现的概率只与前n-1个词有关

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-N+1}^{n-1})$$

如何更好的对$P(w_n | w_1^{n-1})$这一条件概率进行建模呢？

Lattice 在 ASR 中的应用：解码/重打分

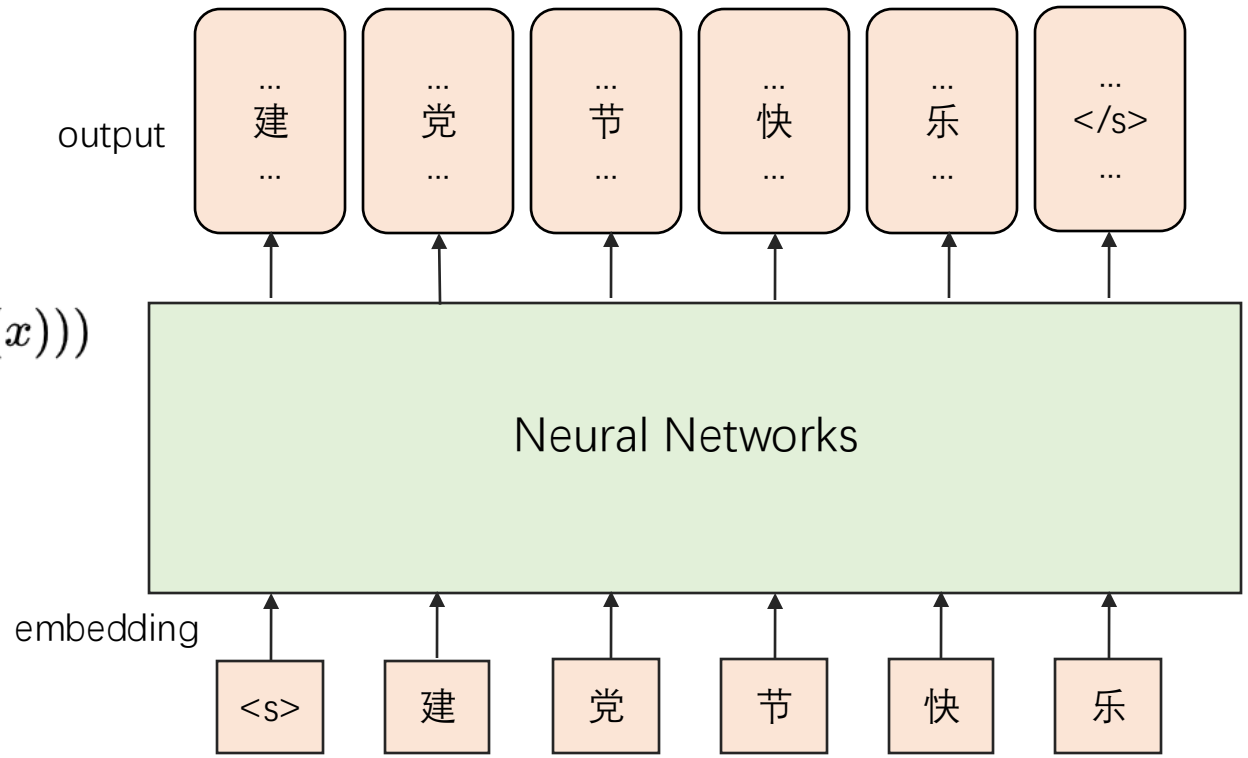
语言模型

NNLM 任务目标

- 交叉熵

$$H(p,q) = -\sum_{x}(p(x)logq(x) + (1-p(x))log(1-q(x)))$$

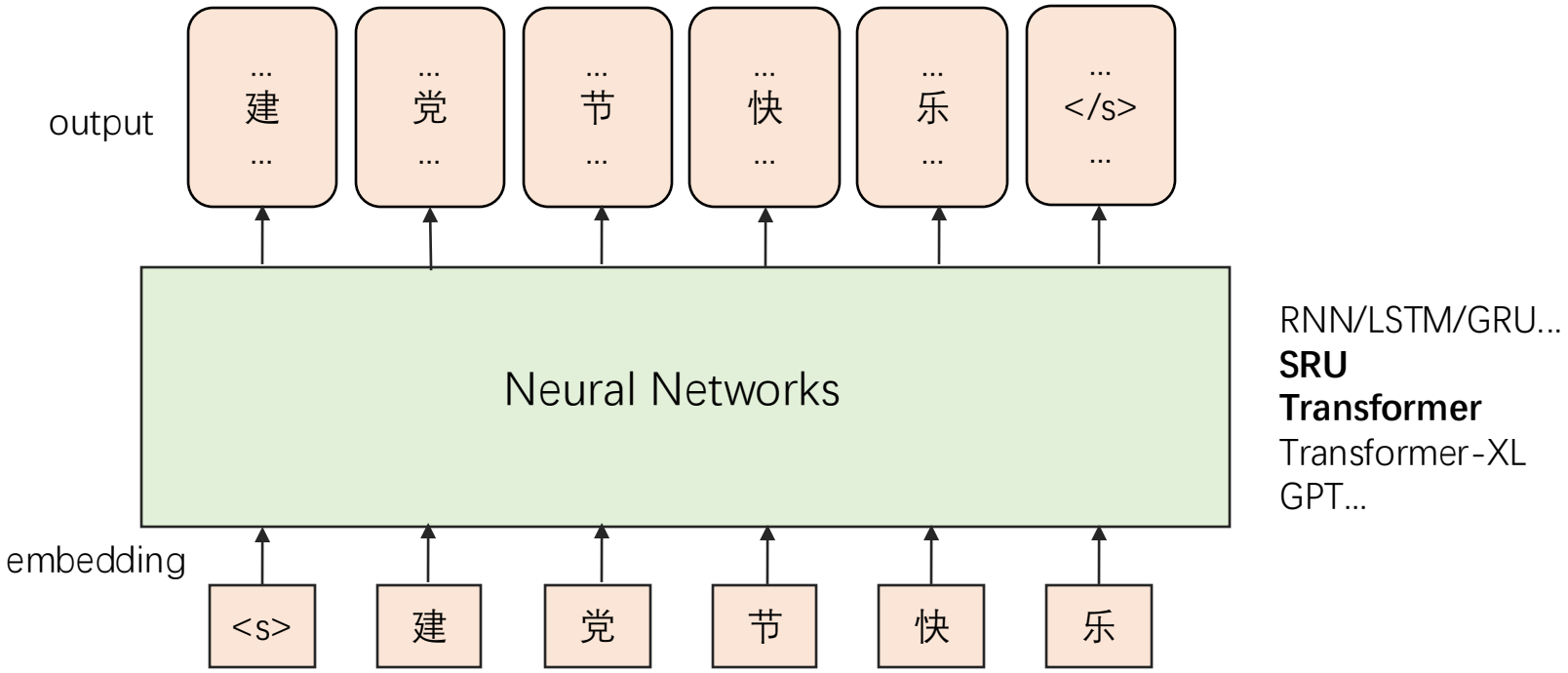
$$H(p,q) = -\sum_{x}(p(x)logq(x)$$



Lattice 在 ASR 中的应用：解码/重打分

语言模型

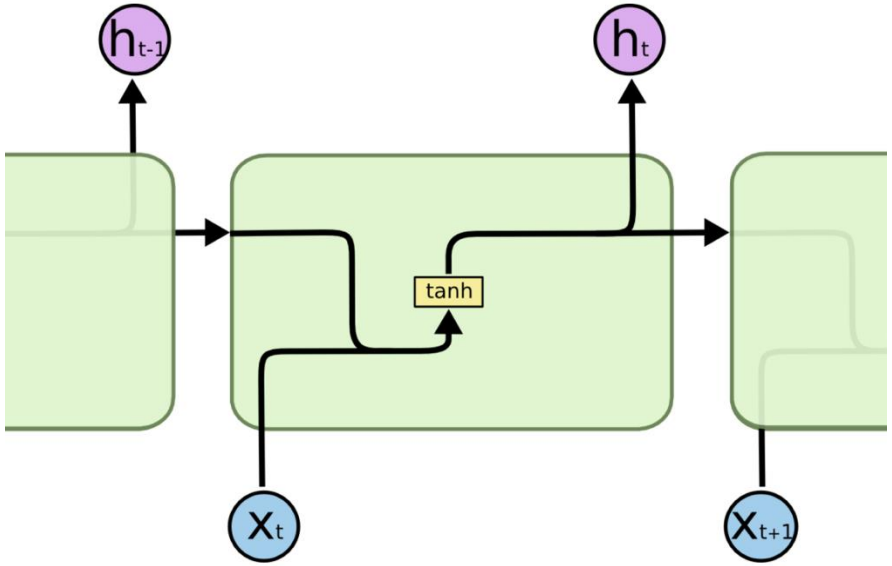
NNLM 任务的抽象化



Lattice 在 ASR 中的应用：解码/重打分

语言模型

RNN/LSTM/GRU 语言模型



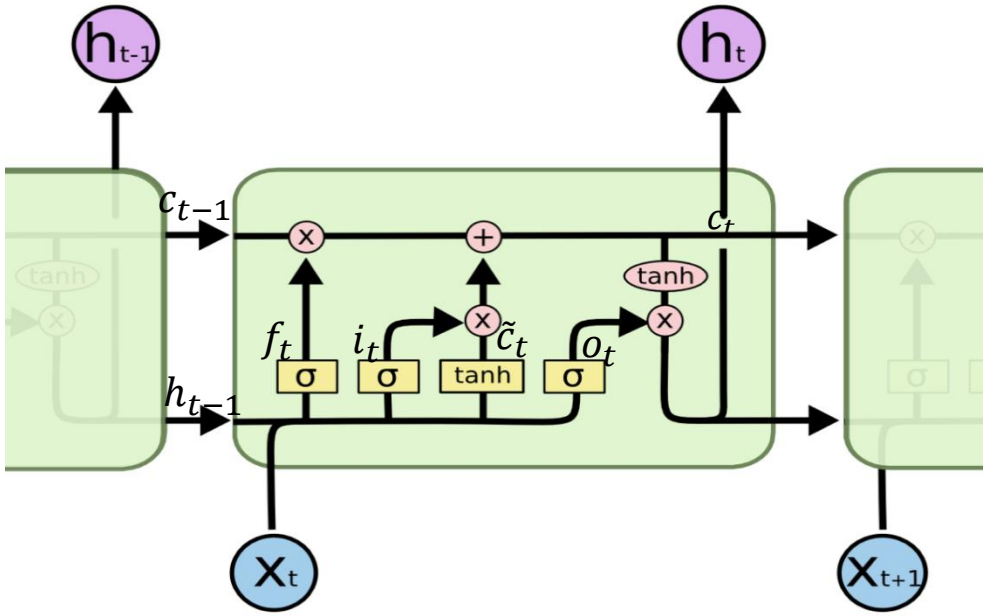
隐含状态　$h_t = \tanh(\mathrm{W}h_{t-1} + \mathrm{U}x_t)$

当前时刻输出　$y_t = \mathrm{W}h_t$

Lattice 在 ASR 中的应用：解码/重打分

语言模型

LSTM 语言模型



输入门：$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1})$

遗忘门：$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1})$

输出门：$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1})$

新记忆单元：$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1})$

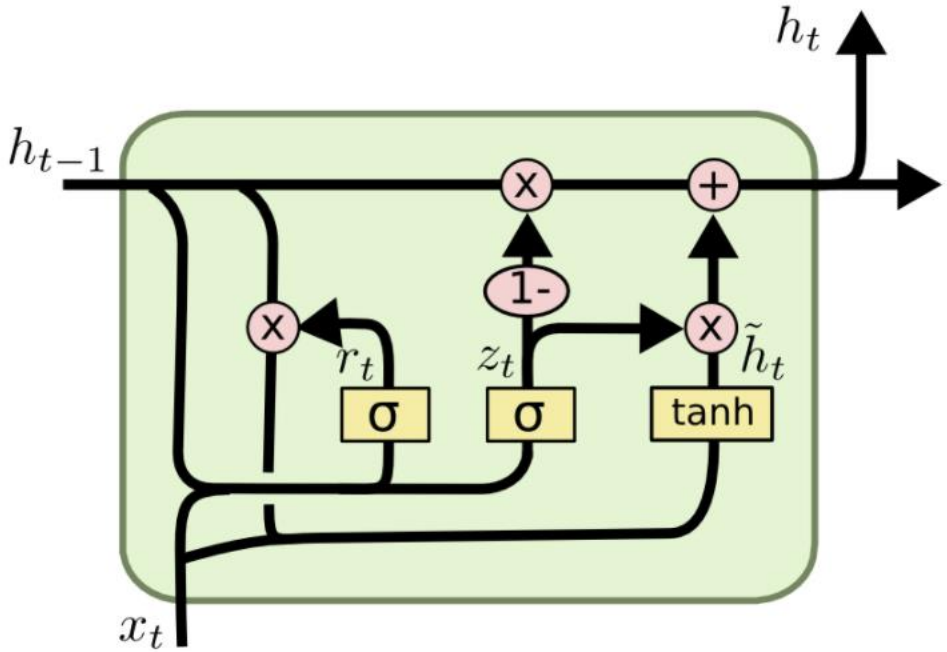
最终记忆单元：$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

下一隐含状态：$h_t = o_t \circ \tanh(c_t)$

Lattice 在 ASR 中的应用：解码/重打分

语言模型

GRU 语言模型

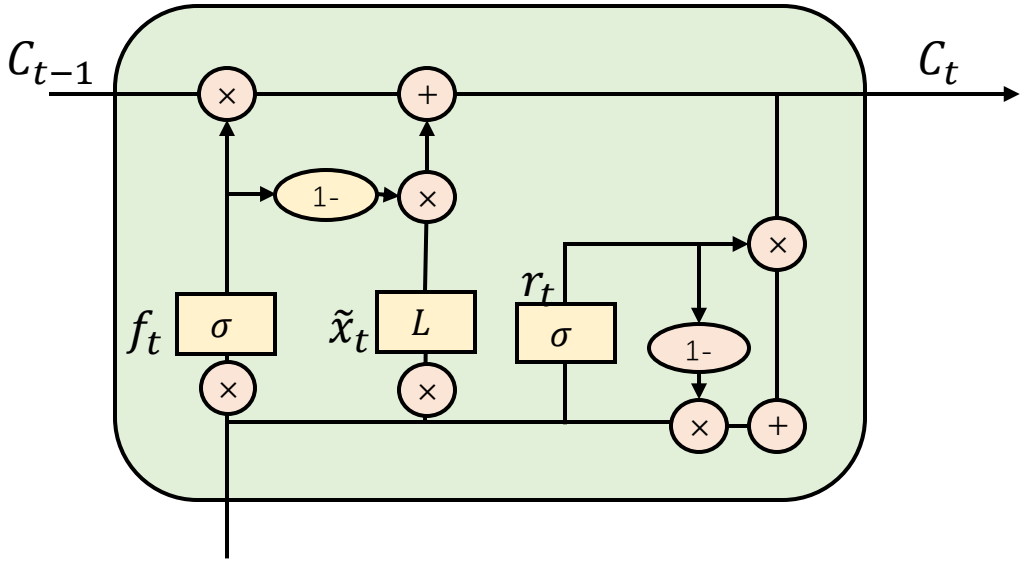


$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$
$$\tilde{h}_t = tanh(W x_t + U(r_t \circ h_{t-1})$$
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t$$

Lattice 在 ASR 中的应用：解码/重打分

语言模型

SRU 语言模型

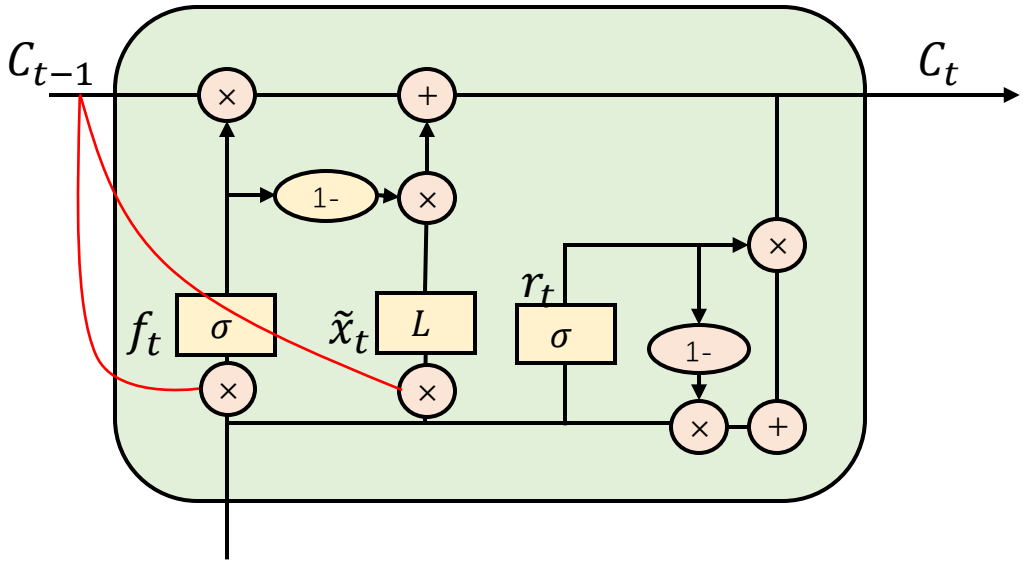


$$\tilde{\mathbf{x}}_t = \mathbf{W}\mathbf{x}_t$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{b}_f)$$
$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{b}_r)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \tilde{\mathbf{x}}_t$$
$$\mathbf{h}_t = \mathbf{r}_t \odot g(\mathbf{c}_t) + (1 - \mathbf{r}_t) \odot \mathbf{x}_t$$

Lattice 在 ASR 中的应用：解码/重打分

语言模型

SRU 语言模型

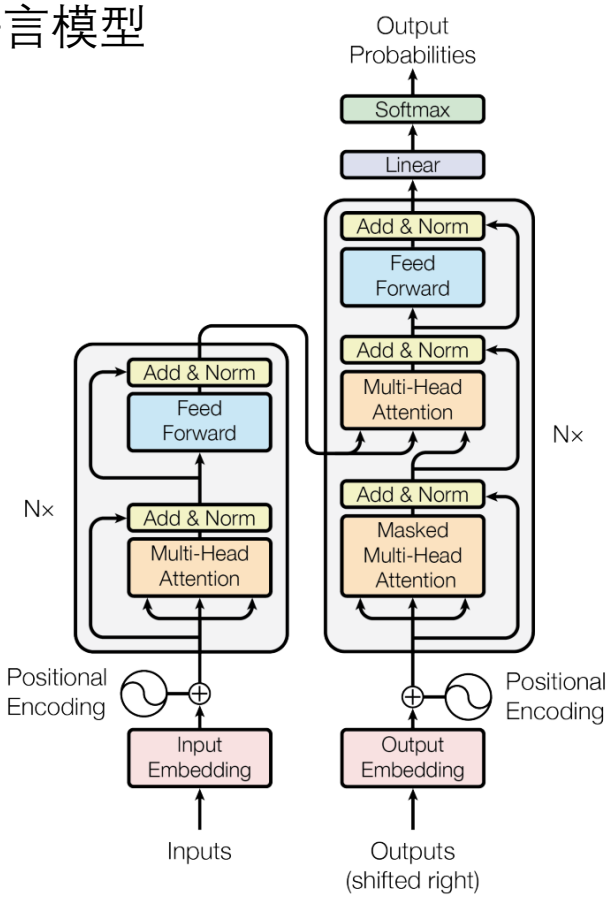


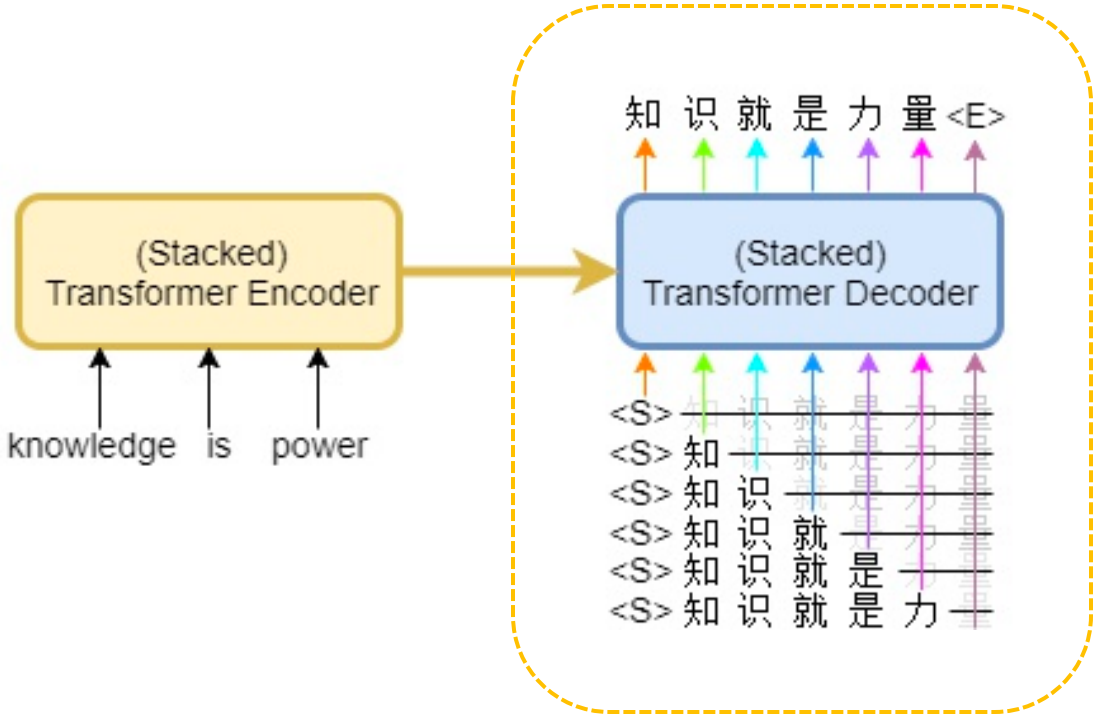
$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{v}_f \odot \mathbf{c}_{t-1} + \mathbf{b}_f\right)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot (\mathbf{W}\mathbf{x}_t)$$
$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r\mathbf{x}_t + \mathbf{v}_r \odot \mathbf{c}_{t-1} + \mathbf{b}_r\right)$$
$$\mathbf{h}_t = \mathbf{r}_t \odot \mathbf{c}_t + (1 - \mathbf{r}_t) \odot \mathbf{x}_t$$

Lattice 在 ASR 中的应用：解码/重打分

语言模型

Transformer 语言模型

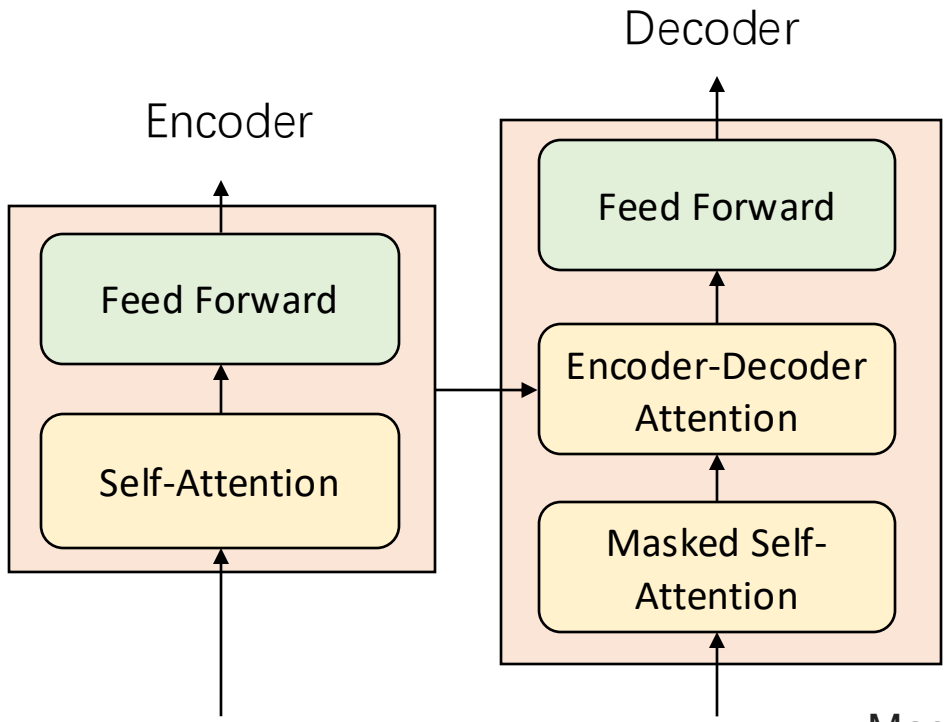




Lattice 在 ASR 中的应用：解码/重打分

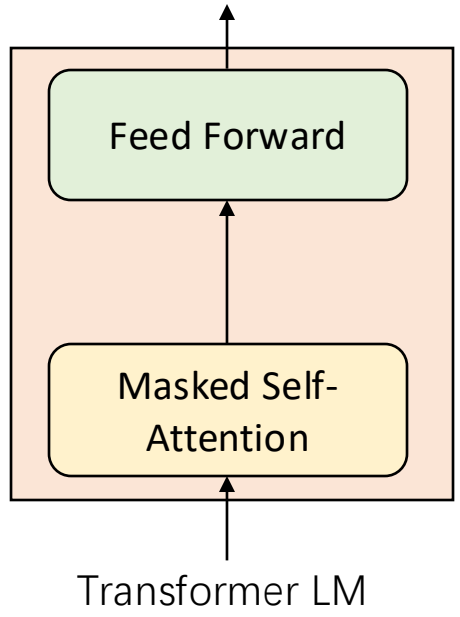
语言模型

Transformer 语言模型



Transformer Decoder 用于语言模型

- 去除 encoder-decoder attention
- 使用 masked self-attention，增加future mask 不用后文的信息



Masked self-attention layer is only allowed to attend to earlier positions in the output sequence. This is done by masking future positions (setting them to -inf) before the softmax step in the self-attention calculation.

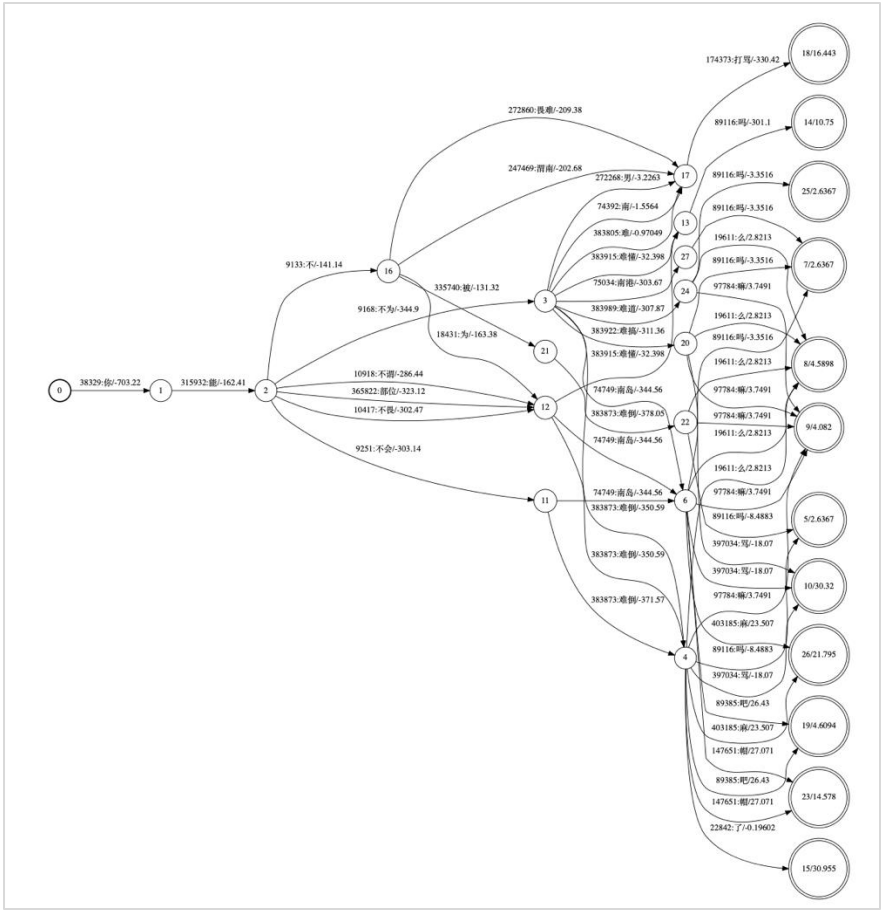
Lattice 在 ASR 中的应用：解码/重打分

语言模型重打分

初级版：N-Best 重打分

将 Lattice 包含的词序列按照 Top-N 展开，每个都是一个线性的词序列转换成单个词序列的语言分

004_mic2_0141.wav-1 你 能 不为 难倒 吗
004_mic2_0141.wav-2 你 能 不为 南岛 吗
004_mic2_0141.wav-3 你 能 不为 难倒 么
004_mic2_0141.wav-4 你 能 不为 难倒 嘛
004_mic2_0141.wav-5 你 能 不为 难倒 骂
004_mic2_0141.wav-6 你 能 不为 南岛 么
004_mic2_0141.wav-7 你 能 不为 南岛 嘛
004_mic2_0141.wav-8 你 能 不会 难倒 吗
004_mic2_0141.wav-9 你 能 部位 难倒 吗
004_mic2_0141.wav-10 你 能 不为 南岛 骂
004_mic2_0141.wav-11 你 能 部位 南岛 吗
004_mic2_0141.wav-12 你 能 不为 难懂 吗
004_mic2_0141.wav-13 你 能 不会 难倒 么
004_mic2_0141.wav-14 你 能 不会 难倒 嘛
004_mic2_0141.wav-15 你 能 部位 难倒 么
004_mic2_0141.wav-16 你 能 部位 难倒 嘛
004_mic2_0141.wav-17 你 能 不为 难倒 了
004_mic2_0141.wav-18 你 能 不 畏难 打骂
004_mic2_0141.wav-19 你 能 不为 难倒 吧
004_mic2_0141.wav-20 你 能 不会 难倒 骂
004_mic2_0141.wav-21 你 能 不为 男 打骂
004_mic2_0141.wav-22 你 能 部位 难倒 骂
004_mic2_0141.wav-23 你 能 不 为 难倒 吗
004_mic2_0141.wav-24 你 能 不为 南 打骂
004_mic2_0141.wav-25 你 能 部位 南岛 么
004_mic2_0141.wav-26 你 能 部位 南岛 嘛
004_mic2_0141.wav-27 你 能 不为 难 打骂
004_mic2_0141.wav-28 你 能 不畏 难倒 吗
004_mic2_0141.wav-29 你 能 不为 南岛 吧
004_mic2_0141.wav-30 你 能 不 渭南 打骂
004_mic2_0141.wav-31 你 能 不为 难搞 吗
004_mic2_0141.wav-32 你 能 不 被 难倒 吗



Lattice 在 ASR 中的应用：解码/重打分

语言模型重打分

初级版：N-Best 重打分

$$\arg\max_{\mathrm{W}} \left[\mathrm{AM}(\mathrm{W}) + \alpha\, LM_{ngram}(\mathrm{W})\right]$$

$$\arg\max_{\mathrm{W}} \left[\mathrm{AM}(\mathrm{W}) + \alpha\, LM_{interp}(\mathrm{W})\right]$$

$$LM_{interp}(\mathrm{W}) = \beta LM_{NN}(\mathrm{W}) + (1-\beta) LM_{ngram}(\mathrm{W})$$

$$LM_{interp}(\mathrm{W}) = \sum_i \beta_i\, LM_i(\mathrm{W}), \sum_i \beta_i = 1$$

Lattice 在 ASR 中的应用：解码/重打分

语言模型重打分

1. 从 Lattice 中 取出候选路径对应的词序列

$$W^* = \underset{W \in \text{lattice}}{\text{argmax}} P(X|W)^a P_1(W)^{l_1} P_2(W)^{l_2}$$

2. 计算词序列中每个词（Lattice中每个弧）的新语言分数

3. 对于同一条弧存在于多个路径中(多个路径共享这条边)的情况：
   - 弧上的分数是所有路径上该弧分数的：平均值
   - 弧上的分数是所有路径上该弧分数的：加权平均，权重是各路径的历史分数进行归一化得到
   - 弧上的分数是经过这条弧的最优的路径中这条弧对应的分数

4. 新语言模型分数和原本的 n-gram 语言分数插值，作为弧的新语言分数

5. 根据声学模型和新语言分数从 Lattice 取出最优路径作为最终识别结果

Lattice 在 ASR 中的应用：解码/重打分

语言模型重打分

实验结果

Setup	Dev		Test	
	clean	other	clean	other
TDNN-F + 4-gram	2.75	8.16	2.93	8.17
Multistream CNN +4-gram	2.62	6.78	2.80	7.06
+TDNN-LSTM LM	2.14	5.82	2.34	6.04
+24-layer SRU	1.56	4.28	1.83	4.57
+Interpolated SRU	1.56	4.25	1.79	4.49

Method	Hub5'00	Swb	Callhm
4-gram KN	12.8	8.6	17.0
N-best (LSTM)	10.9	7.1	14.6
N-best (Transformer)	10.8	7.2	14.4
Non-iterative ($\epsilon = 0.5$)	10.6	6.8	14.3
Non-iterative ($\epsilon = 0.005$)	10.4	6.8	14.0

Lattice 在 ASR 中的应用：解码/重打分

Single-Shot Lattice Rescoring

LT-LM: a novel non-autoregressive language model for single-shot lattice rescoring

N-Best/Lattice Rescoring 的问题
- 很多重复计算，Lattice包含路径太多时展开候选序列过多(建模的目标和语言分数任务存在差异)

Sing-Shot Lattice Rescoring
- 不再需要将 Lattice 进行线性展开
- 输入是 Lattice 而不是线性序列，输出是 LatticeArc 的新语言分，与 Lattice Rescoring 任务的目标更契合

Lattice在下游任务中的应用：思想

1. 为什么比直接用 ASR 的 one-best 结果更好?
one-best 最优结果的原始信息损失过多，ASR 不可靠时 one-best 错误率可能较高，造成误差累积

2. 相比于端到端直接用语音建模的优势?
端到端直接从语音特征建模，目前用于语音翻译等任务时没有级联模型好。
带语音的平行语料稀少，端到端建模需要数据量较大；模型训练可控性差，不易debug

Lattice 方案:
介于纯语音特征和 one-best 解码结果之间，提炼出文本信息的同时，还保留了一些次优的结果和分数，相当于编码了更多的信息供下游任务使用。

总 结

可能的工作

1. 语音识别
    1. 将目前的 LF-MMI 方案替换成 LF-sMBR 或 LF-bMMI
    2. 更大数据量级的半监督声学模型训练/调优
    3. sMBR 用状态级结果进行优化，是否更适合半监督训练？
    4. 更强的语言模型，用于 Lattice 重打分：**SRU / Transformer-XL** 等
    5. 高效的 single-shot Lattice 重打分方案

2. 基于 Lattice 的应用
    1. 关键词检索：基于ASR Lattice 的多候选查询和索引，用于敏感词检测、质检
    2. 口语理解：Lattice 作为下游自然语言处理和理解的输入