语音识别交流

王春惠

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

语音识别现状

- 腾讯：CTC 混合架构
- 百度：流式多级截断注意力模型（SMLTA）
- 阿里： kaldi 进行二次开发具体模型未知
- Apple： kaldi 区分性训练
- JD ：使用kaldi进行二次开发
- 出门问问：基于kaldi进行二次开发
- CVTE：对kaldi进行二次开发

语音识别现状

- 工业上：
  - 大部分公司以kaldi 进行二次开发
  - Ctc
  - 区分性训练
  - Tdnn等
- Research：
  - Transducer
  - Transformer
  - Ctc
  - Teacher-Student
  - 区分性训练

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

Kalid 介绍

- 核心代码简洁、易懂

  一个基于C++的语音识别工具，包含现大量的语音识别基础算法，核心代码编写简洁易懂，方便进行二次开发。

- 周边脚本复杂性高

  Kaldi 周边脚本种类复杂，上手难度大。其中包括shell、perl、python

- 数据处理方式灵活

  采用pipeline的数据处理方式

- 训练方式

  可单机器多卡，多机多卡。

Kaldi 现有目录介绍

- 例子目录：egs
- 工具目录：tools
- 源码目录：src

  xxxbin：可执行文件目录

  xxx　　：算法源码

注：kaldi中所有的执行文件都是在xxxbin中，都是通过环境变量添加到脚本中。

Kaldi 现有算法介绍

- 特征提取：
  MFCC，PLP，Fbank（常用MFCC和Fbank）
- 基础性算法
  GMM、HMM 等
- 说话人适应性训练
  fMLLR 等
- 基于DNN的说话人适应性训练
  UBM-I-vector，X-vector等
- 神经网络：
  DNN，RNN，LSTM，GRU，CNN等

…

Kaldi 二次开发

- 工程：

  如果使用kaldi 进行工程落地，通常进行二次开发的是kaldi的解码器，也就是在src/nnet2bin目录下相对应的代码。

- 算法：

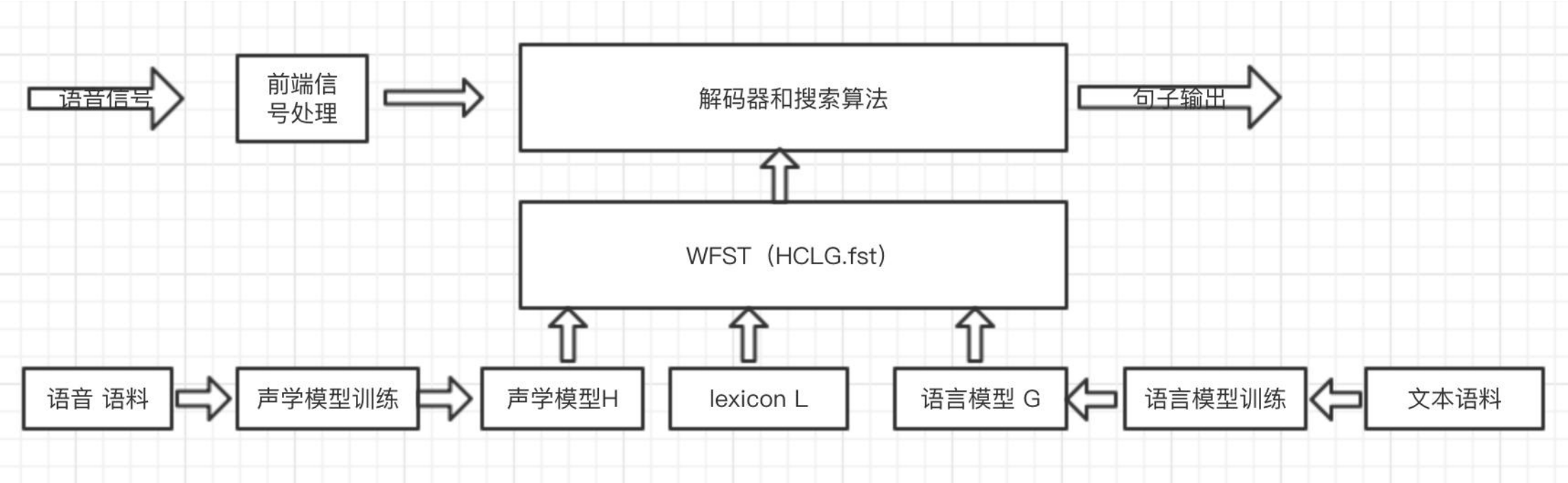
  对于kaldi算法来说，kaldi会提供egs目录，并会不断更新算法，但是如果需要进行新层的添加或者神经网络配置修改，可以直接在对应算法的脚本中进行,但是有些情况是需要自己去修改源代码来进进行添加新算法、以及新的损失函数。

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

名词介绍

- Lattice：给出一段语音对应的可选择的生成句子的打分
- Alignment：通过维特比算法把hmm状态与对应的语音对齐
- acoustic scale：解码器中的声学比例，通常是0.1,但是chain model为1.0，acwt
- G.fst：语言模型转换器
- Pdf-id: HMM 状态
- Transition-id： 对pdf-id的编码

Hybrid System 语音识别流程



Hybrid System 语音识别流程

- 声学前端
  主要围绕在家居和车载环境所必须的麦克风阵列、降噪、去回声、去混响、唤醒等功能；
- 声学模型部分
  主要关注如何将声学信号建模；
- 语言模型
  语言文字本身建模
- WFST 构建
  为解码器构建解码图
- 解码器识别

语音识别公式表达

- 声学特征：$X = x_1, x_2 \ldots, x_n$
- 单词序列：$W = w_1, w_2 \ldots, w_n$
- 公式：

$$W' = argmax P(W|X)$$
$$= argmax \frac{P(W)P(X|W)}{P(X)}$$
$$= argmax P(X|W)^b P(W)^a$$

P(X|W):声学模型； b: 声学比例

P(W):语言模型； a:语言模型权重

作用：平稳声学模型和语音模型

评价标准

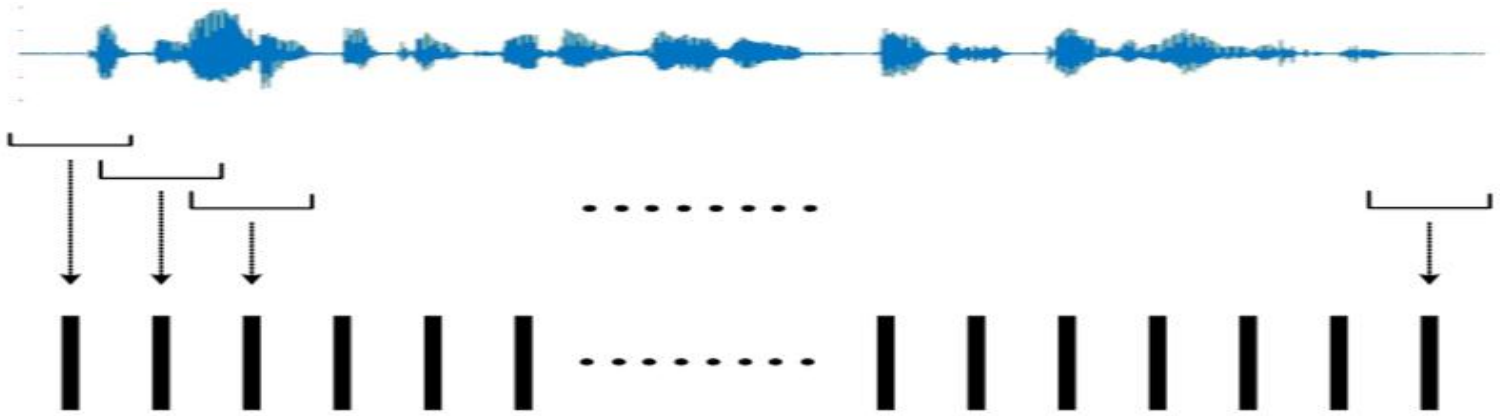
- WER: 词错误率

$$WER = 100 * \frac{Insertions + Substition + Deletions}{Total\ words\ in\ Correct\ Transcript}$$

- 主要算法： Minimum Edit Distance(编辑距离)
- 作用：衡量两个序列相似程度的指标

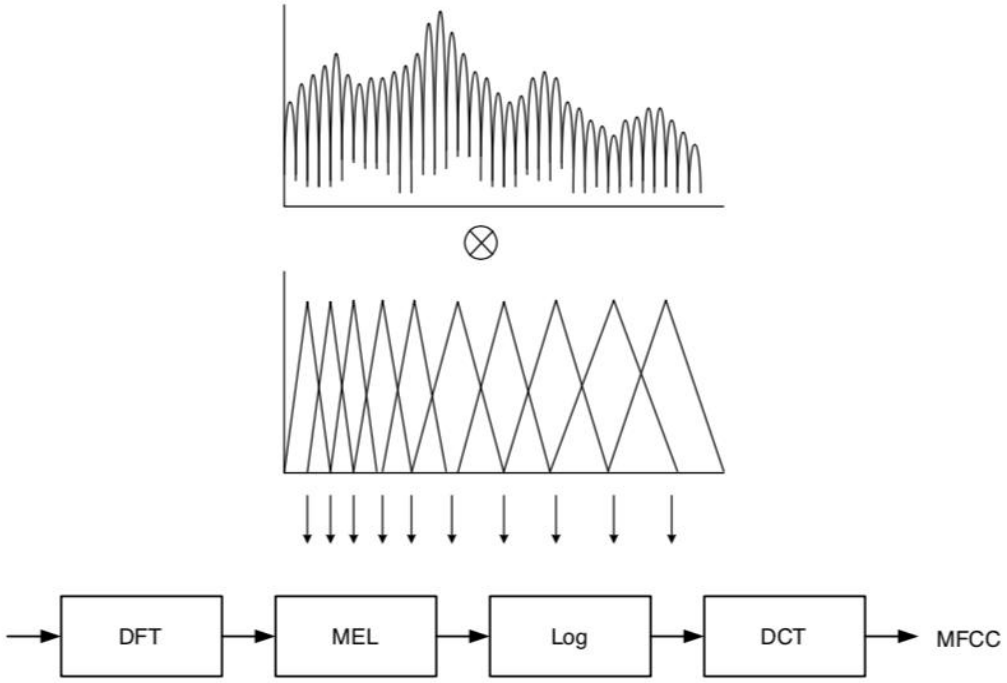
- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

特征提取



- 20ms 的帧窗口、每10ms移动一次
- 常用特征方法
  - MFCC
  - PLP
  - FBank (DNN中广泛使用)

Fbank 和 MFCC



MFCC/Fbank
计算：
MFCC：在Fbank的基础上进行，MFCC计算量大
特征：
Fbank 提取的特征的相关性比较大
MFCC 具有更好的辨别性

Cepstral Mean and Variance Normalization(CMVN)

- 目的：解决channel distortion 问题
- offline情况下的CMVN：
  - 均值： $u_t(i) = \frac{1}{N}\sum_{n=t-N/2}^{t+\frac{N}{2}-1} x_n(i)$
  - 方差： $\sigma_t^2 = \frac{1}{N}\sum_{n=t-N/2}^{t+\frac{N}{2}-1}(x_n(i) - u_t(i))^2$
  - 特征： $x_t(i) = \frac{x_t(i)-u_t(i)}{\sigma_t(i)}$
- Online情况下的CMVN：
  - 不是global，而是默认对一定时间段进行，默认是600帧

声学单元和字典

- 字典

  我 uu uo3
- 内容独立音素

  b c ch e ee ei …
- 内容依赖音素
  - 协音
  - HMM模型构建的每个内容依赖音素
- 其他声学单元
  - 音节、单词等

Context-Dependent Phonemes

- 优势

  能够根据声学背景构建细致的声学模型

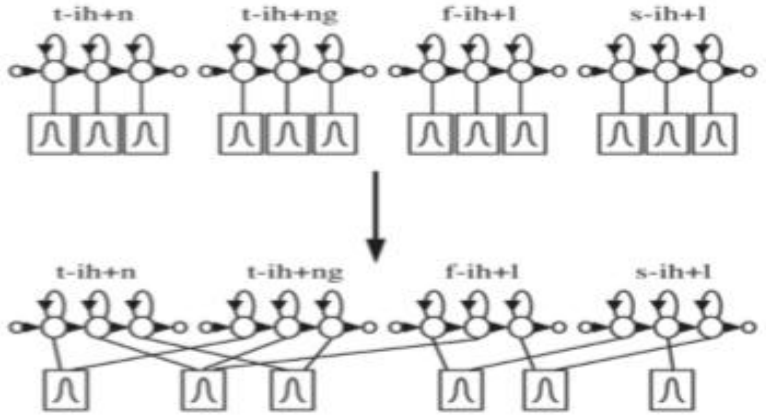
- 缺点

  产生大量的CD音素

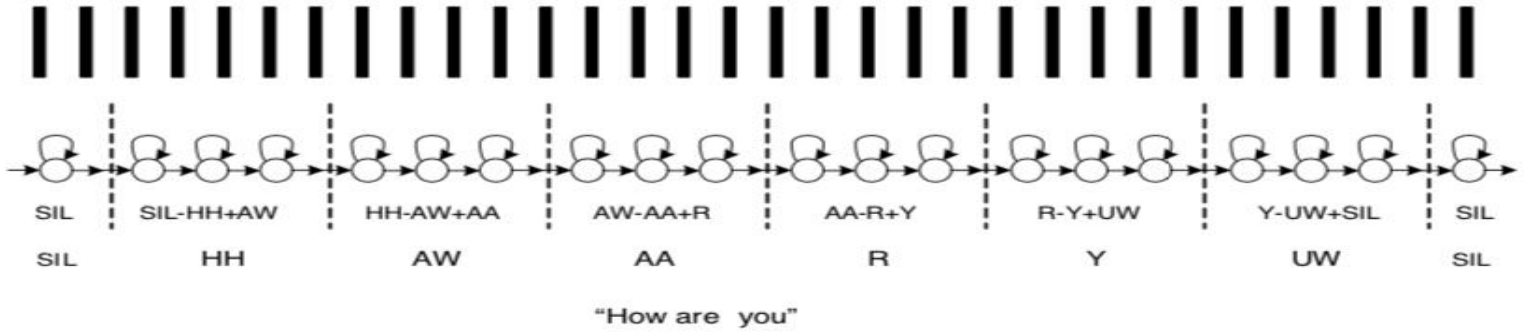
  例如：$45^3 = 91125$， $45^5 = 1.8 \times 10^8$

- 解决方法

  ➢参数绑定



GMM-HMM:声学模型训练



如何计算给出模型的序列似然？

- 隐藏状态序列S = { S_1,... S_T} , 特征序列 O = {o_1,...,o_T}

$$P(O|\lambda) = \sum_S P(O \mid S, \lambda)$$

- 前后向算法

$$P(O|\lambda) = \sum_i \sum_j a_{t-1}(j) a_{ij} b_i(o_t) \beta_t(i)$$

前向变量：$a_t(i) = P(o_1 \dots o_t, s_t = i \mid \lambda)$

后向变量：$\beta_t(i) = P(o_{t+1} \dots o_T | s_t = i, \lambda)$

GMM-HMM 声学模型训练

- 如何通过数据评估模型参数λ？
  - 给出特征序列$O = \{o_1, o_2, \ldots o_T\}$

$$\lambda^* = argmaxP(O\ |\lambda)$$

  - EM 算法 （Baum-Welch 算法）

$$c_{ik} = \frac{\sum_t r_{ik}(t)}{\sum_t r_i(t)}$$

$$u_{ik} = \frac{\sum_t r_{ik}(o) o_t}{\sum_t r_{ik}(t)}\ , \qquad \sum_{ik} = \frac{\sum_t r_{ik}(t)(o_t - u_{ik})(o_t - u_{ik})^T}{\sum_t r_{ik}(t)}$$

  - $r_{ik}(t) = P(s_t = i,\ \varsigma_t = k\ |O, \lambda)$

Maximum likelihood linear regression(MLLR)

- 将一个已经存在的模型通过线性集合转换映射成为一个新的适应性模型，最大化数据的适应性似然。
- 一个鲁棒性非常好的无监督增强适应性模型。
- 公式：

$$u_{ik} = A_c u_{ik} + b_c$$

A_c: 回归矩阵 b_c: 与c个分类相关的偏移向量

- 目的：将均值映射到另一个空间来取消说话人之间的错误匹配

GMM-HMM: 语言模型训练

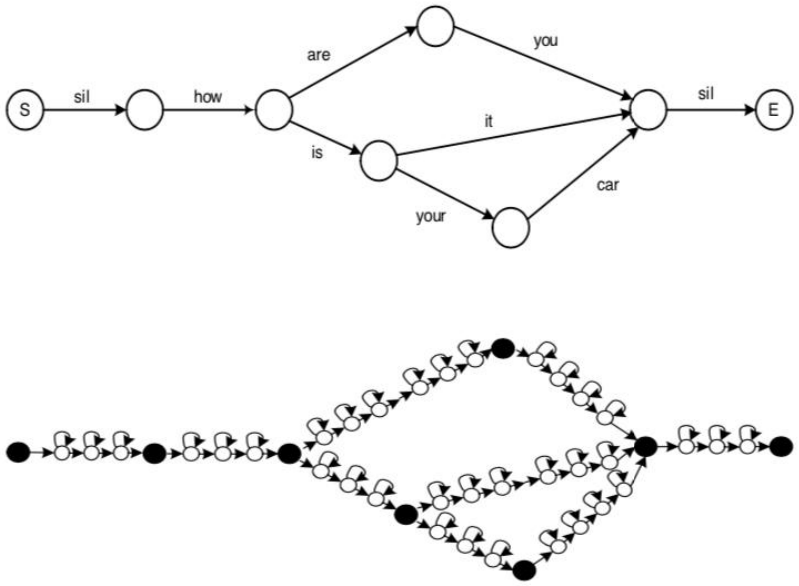
- N-gram:
  - 近似n个历史单词的条件概率

$$P(w_1, w_2, \dots w_m) \approx \prod_{i=1}^{m} P(w_i | w_{i-n+1}, \dots, w_{i-1})$$

  - 使用训练数据计数

$$P(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{C(w_{i-n+1}, \dots, w_{i-1}, w_i)}{C(w_{i-n+1}, \dots w_{i-1})}$$

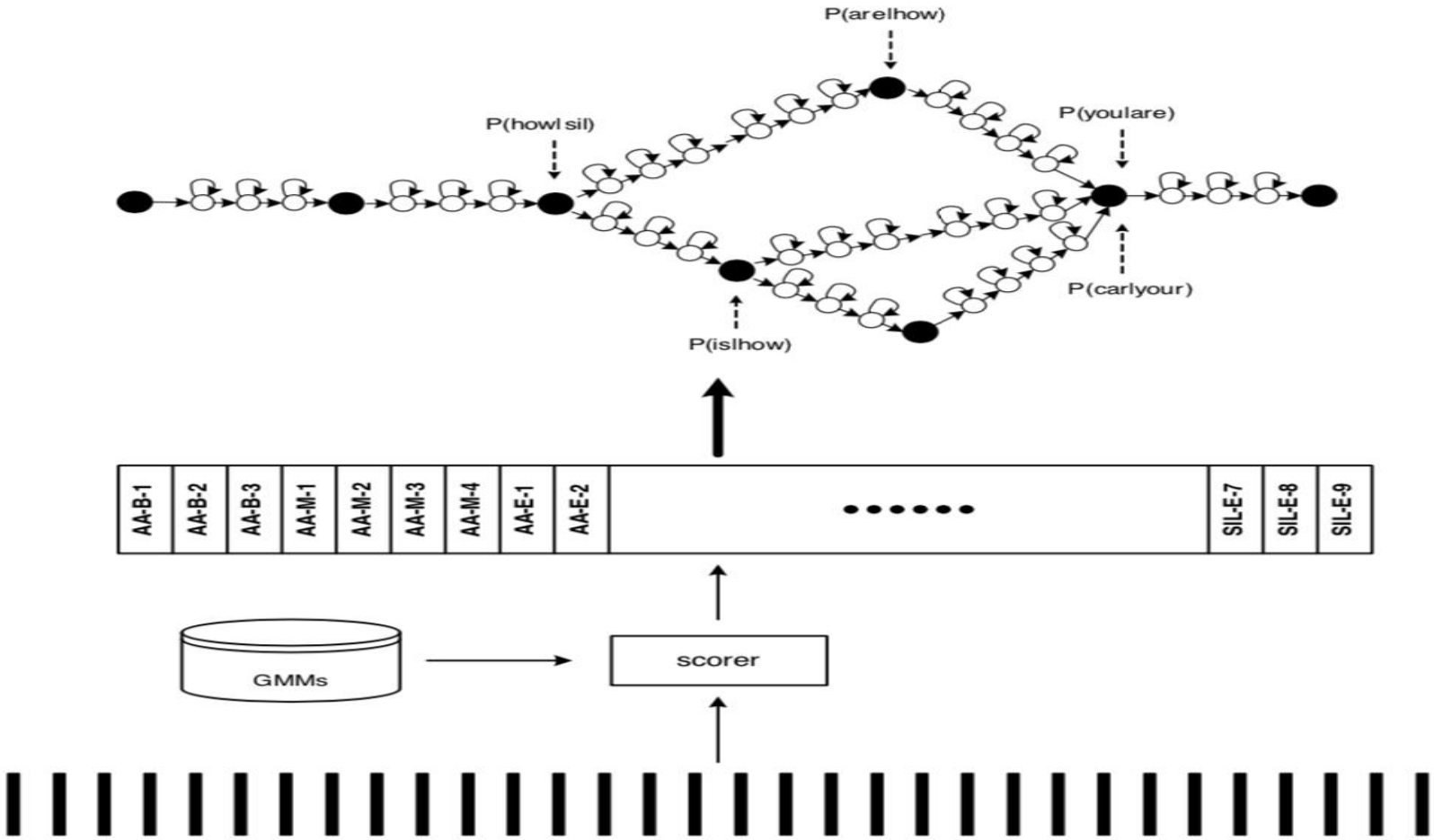
GMM-HMM: 解码



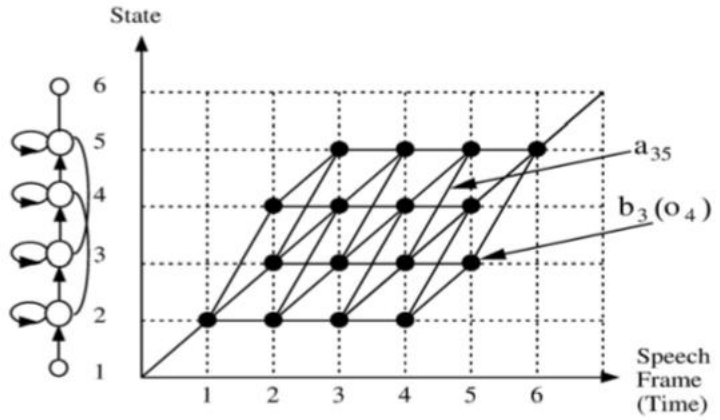
内置语言模型
内置字典
内置具有GMM分布的每个CD-HMM状态的CD-HMM
计算每个特征向量在每个CD-HMM状态中的log 似然

$$P(x|s) = \sum_i c_i \mathcal{N}(x; u_i, \varepsilon)$$

GMM-HMM: 解码


P(arelhow)
P(howlsil)
P(youlare)
P(carlyour)
P(islhow)
AA-B-1
AA-B-2
AA-B-3
AA-M-1
AA-M-2
AA-M-3
AA-M-4
AA-E-1
AA-E-2
SIL-E-7
SIL-E-8
SIL-E-9
GMMs
scorer

GMM-HMM: 解码



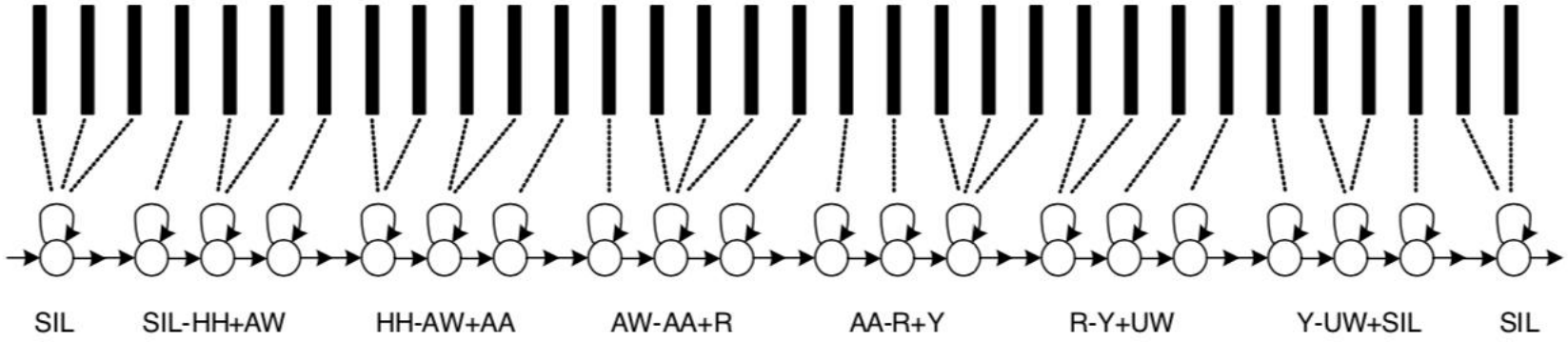
如何通过给出的模型的最大似然来发现一个特征的路径？

- 动态规划（维特比解码）
- $\emptyset_j(t)$ 表示在状态j时间t下状态序列从o_1 到 o_t的最大似然。

$$\emptyset_j(t) = \max_{s_1,s_2,\dots s_{t-1}} P(s_1, s_2, \dots s_{t-1}, s_t = j, o_1, o_2, \dots o_t|\lambda)$$

即： $\emptyset_j(t) = \max_t\{\emptyset_i(t-1)a_{ij}\}b_j(o_t)$

GMM-HMM:强制对齐

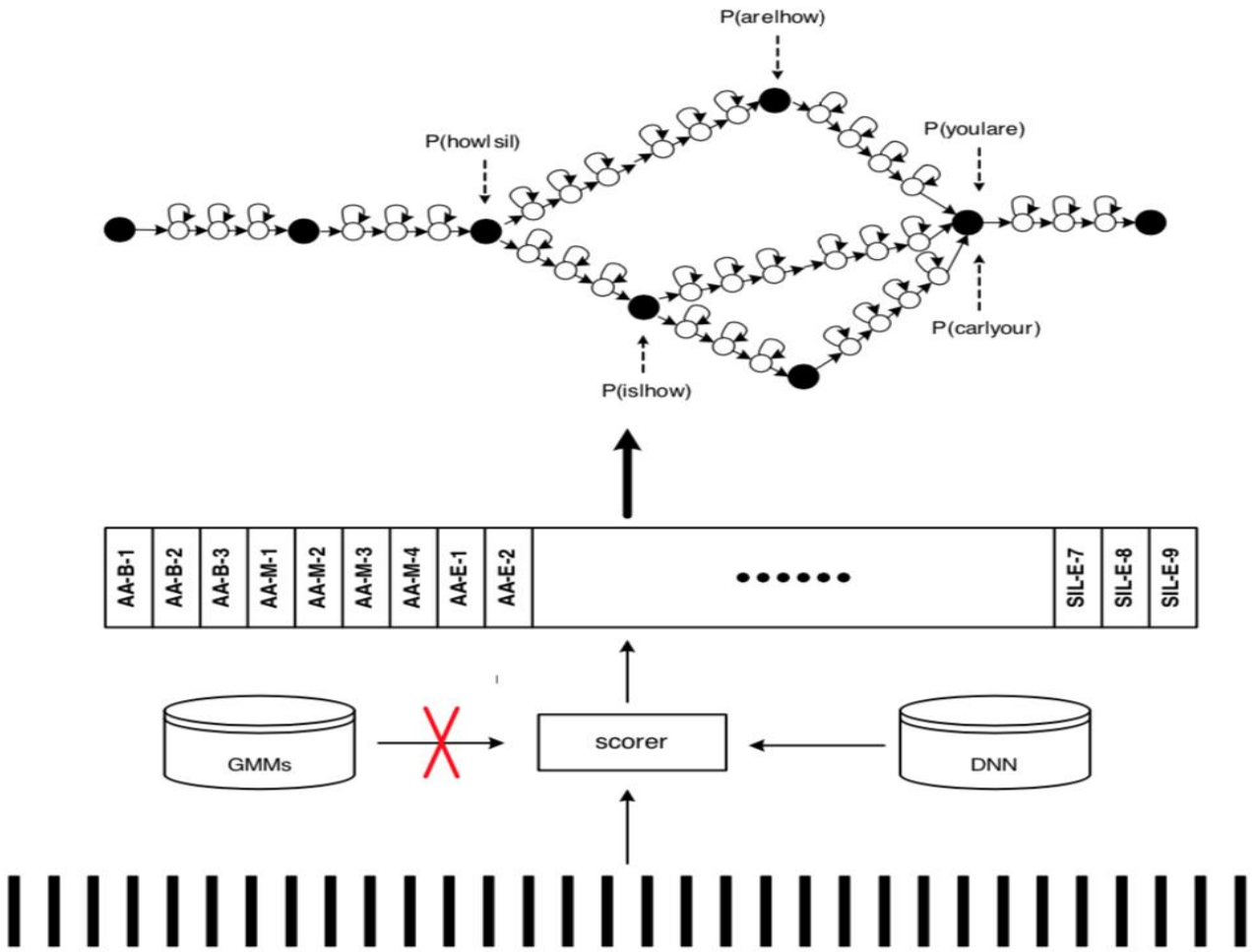


给出文本，如何发现最好的状态序列？

- 除了文本已知外跟解码类似
- 再次使用维特比算法

Kaldi中主要是用的就是维特比强制对齐。

CD-DNNs


P(arelhow)
P(howlsil)
P(youlare)
P(carlyour)
P(islhow)
AA-B-1
AA-B-2
AA-B-3
AA-M-1
AA-M-2
AA-M-3
AA-M-4
AA-E-1
AA-E-2
SIL-E-7
SIL-E-8
SIL-E-9
GMMs
scorer
DNN

Data Augmentation

- 目的：
  - 增加训练数据量，提高模型的泛化能力
  - 增加噪声数据，提高模型的鲁棒性

- 方法：
  - 加减速语音
  - 增加不同形式的噪声到语音中
  - Spec Augment

Data Augmentation(1)

- 方法一：增减速语音

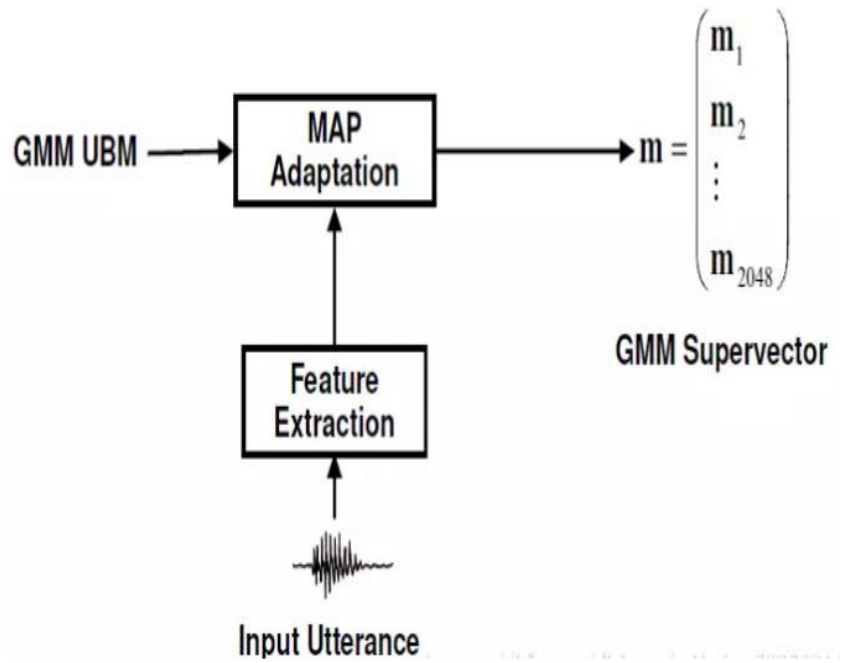
  通常在现有语音数据的基础上，对语音进行适量的加速和减速，产生新的语音，从而达到数据增广的目的。

- 涉及脚本：
  - utils/data/perturb_data_dir_speed_3way.sh
  - 通常默认是提速0.1，减速0.1

Data Augmentation(2)

- 数据增强数据集：musian、RIRS_NOISES
- 目的：构建具有真实环境的屋内回声和噪声环境的数据集
- 合成：在kaldi中使用的脚本是通过合成信噪比为20、15、10和0
- 涉及脚本：
  - steps/data/reverberate_data_dir.py
  - steps/data/make_musan.sh
  - steps/data/augment_data_dir.py

UBM



将说话人GMM模型的每个高斯分量的均值叠加形成一个高纬的超矢量。

i-vector

- Ivector 的主要作用在于说话人拟合。

$$s = m + Tw$$

T:低维空间矩阵

w: i-vector

s : supervector

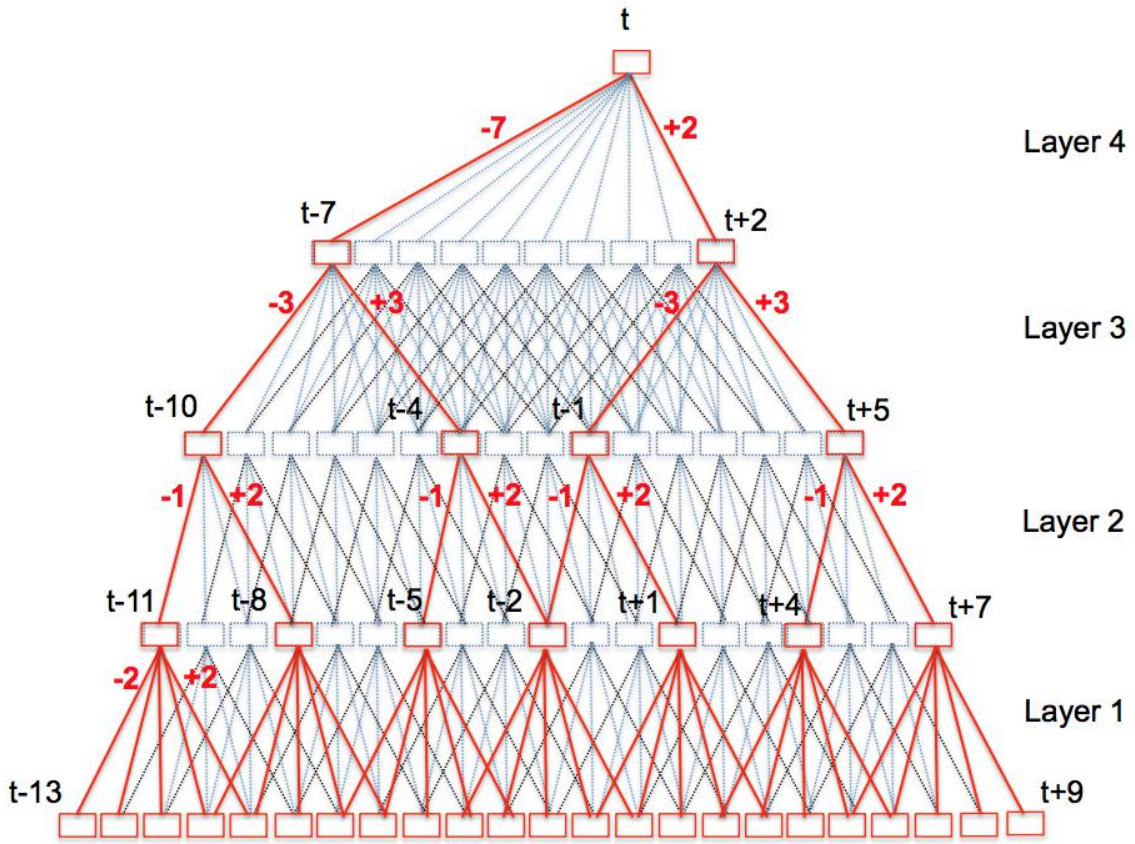
m : ubm's mean supervector

Note: 因为s、m在ubm算法训练的时候已经获取，所需这里需要做的是为了获取w，必须求的T。T的主要求解方式是EM算法

LF-MMI(chain)

- 一种不需要帧级别交叉熵预训练情况下对神经网络声学模型进行序列区分性训练的方法
- 采用LFR，加速训练
- 不需要构建分母词图，从而节约了因为区分性训练构建lattice的时间

TDNN



优势：

速度快：

使用子采样，速度快，涉及到跳帧操作

通常使用TDNN进行训练是非TDNN训练要快5倍

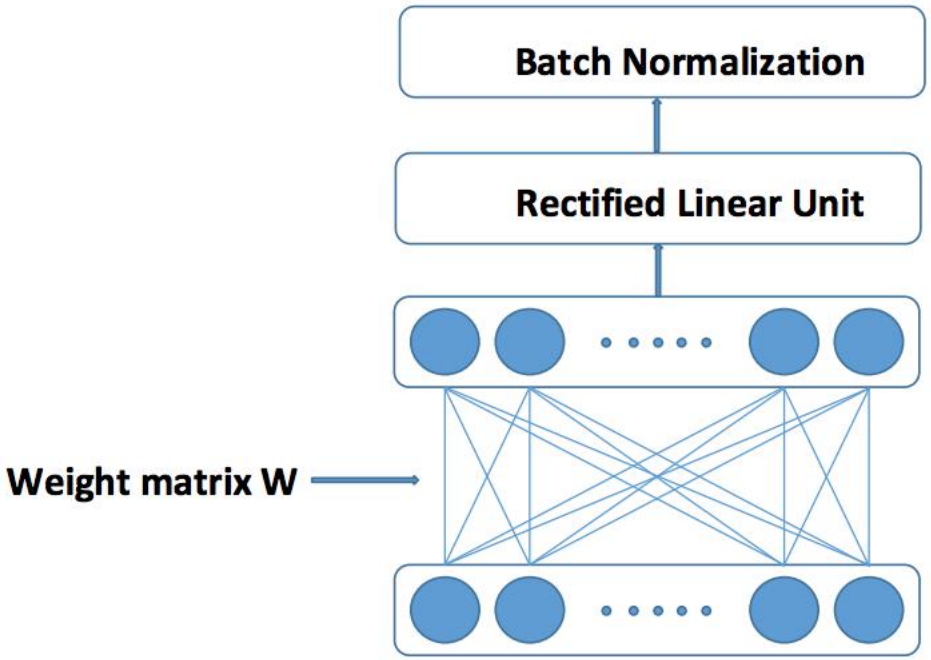
模型构建简单：

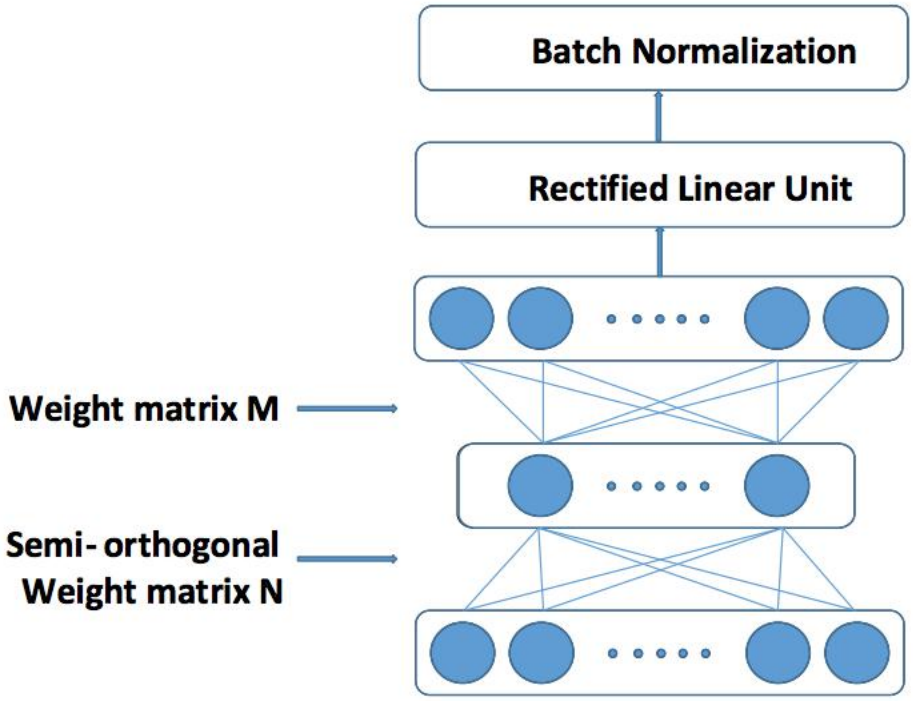
TDNN 可以被当作一维CNN

TDNN-F

- TDNNF：一个基于TDNN架构，但是每一层都使用奇异值分解(SVD)进行压缩
- SVD：降低神经网络训练参数，形成一个线性bottleneck
- 效果：
  - 识别效果优于TDNN

Acoustic Model	Size	Eval2000		RT03	Time [4](s)
		SWBD	Total		
Baseline TDNN (625)	19M	9.5	14.3	17.5	90
+ l2 regularization		9.1	14.0	16.9	96
Baseline TDNN (1536)	80M	9.4	14.6	17.2	211
+ l2 regularization		9.0	13.9	16.6	210
Factorized TDNN (1536-256)	20M	9.7	14.4	17.4	154
+ l2 regularization		9.1	13.9	17.0	155
++ semi-orthogonal		9.2	**13.7**	**16.0**	147

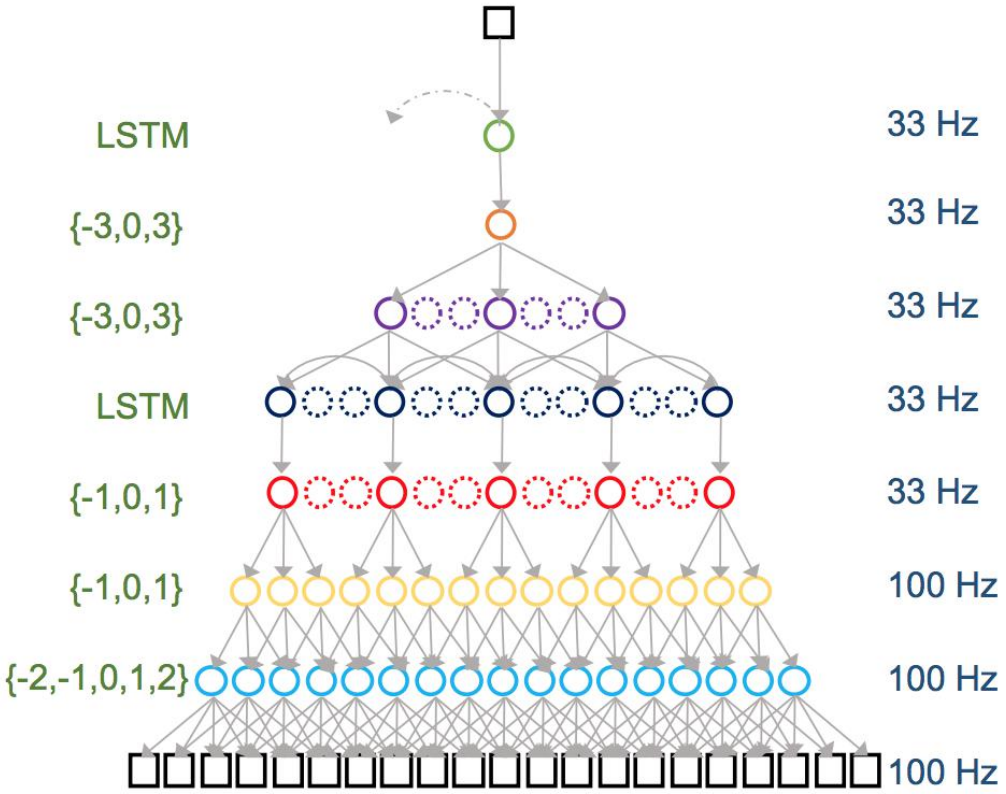




图左边为典型的TDNN结果，图 右边为TDNNF的结构。
TDNNF，主要是进行权重的奇异值分解，使得权重氛围两部分，一部分是基于前一次神经网络cell个数和一个bottleneck的矩阵，另一个矩阵为bottleneck和下一层神经网络cell个数的半正交矩阵。
例如：

[700, 2100] = [700, 250] * [250, 2100]

TDNN-LSTM



LFR: 低帧率，对输入特征进行拼接捆绑，通常情况下3帧拼接的方式能够加快识别速度而不影响识别效果。

Model	Architecture*	Latency (ms)	Matched Inference		
			WER(%)		RTF
			SWBD	Total	
TDNN-D	$T^{100}T^{100}T^{100}TTTT$	150	9.6	14.8	0.8
LFR-LSTM	$L_fL_fL_f$	70	10.1	15.6	2.5
MFR-LSTM	$L_f^{100}L_fL_f$	70	9.9	14.8	2.7
TDNN-LSTM-A	$T^{100}T^{100}T^{100}TTTTL_fL_fL_f$	200	9.5	14.6	2.7
TDNN-LSTM-B	$L_f^{100}L_f^{100}L_f^{100}T^{100}T^{100}T^{100}TTTT$	200	9.4	14.3	4.0
TDNN-LSTM-C	$T^{100}T^{100}T^{100}L_fTTL_fTTL_f$	200	9.2	14.2	3.7
TDNN-LSTM-D	$T^{100}T^{100}T^{100}L_f^{100}TTL_fTTL_f$	200	9.0	13.9	4.8
LFR-BLSTM	$[L_f, L_b], [L_f, L_b], [L_f, L_b]$	2020	9.6	14.5	4.7
MFR-BLSTM	$[L_f^{100}, L_b^{100}][L_f, L_b][L_f, L_b]$	2020	9.0	13.6	6.6
TDNN-BLSTM-A	$T^{100}T^{100}T^{100}[L_fT, L_bT][L_fT, L_bT][L_f, L_b]$	2170	9.2	14.1	4.6
TDNN-BLSTM-B	$T^{100}T^{100}T^{100}[L_f, L_b]TT[L_f, L_b]TT[L_f, L_b]$	2170	9.1	13.8	4.7
TDNN-BLSTM-C	$T^{100}T^{100}T^{100}[L_f^{100}T^{100}, L_b^{100}T^{100}][L_fT, L_bT][L_f, L_b]$	2130	9.0	13.8	9.6

rescore

- Ngram LM rescore
  - 脚本： steps/lmrescore.sh
- RNNLM rescore
  - 脚本： steps/rnnlmrescore.sh

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

offline识别案例

- 背景：客服与客户电话对话数据
- 语音形式： 双声道，8k HZ采样率
- 目标：能够识别销售与客户之间对话数据，并形成通顺的人类可理解的文字版本。
- 数据集：真实场景语音2000小时+公共语音1000小时
- 结果：真实场景WER 6%

Offline-clean dataset

- 数据集：400小时公共语音
- 语言模型：400小时对应的文本
- 结果：wer 70 %
- 使用模型：tdnn，cnn-tdnn-f, tdnn-f, tdnn-lstm
- 模型间结果：四种模型对比类似

语言模型提升(1)

- 背景：400小时公共语音
- 语言模型：搜狐教育5G数据构建的语言模型
- 模型类型：3gram
- 结果：提升效果2%

语言模型提升(2)

- 背景：400小时公共语音
- 语言模型：搜狐教育50G数据构建的语言模型
- 模型类型：3gram
- 结果：结果几乎没变

语言模型提升(3)

- 背景：400小时公共语音
- 语言模型：使用公司20M文本数据集进行新语言模型的构建
- 结果：相同声学模型的情况下，WER 降低了10 个点，到达了60%

语言模型提升(2)

- 背景：400小时公共语音
- 语言模型：使用1G数据构建3gram语言模型
- 结果：WER 降低了3个点

Offline-multi-condition

- 背景：400小时公共语音，1G相关语料 3gram语言模型
- update：multi-condition 即data augmentation
- 结果：整体识别错误率到达 27%

Offline-multi-condition-rescore

- 背景：400小时公共语音，1G相关预料
- 声学模型：multi-condition-model
- 语言模型：3gram 语言模型
- Update：rnnlm 重打分
- 结果：WER 降低 3个点到达24%

模型部署测试

- 背景：400小时公共语音，1G相关预料
- 声学模型：multi-condition-model
- 语言模型：3gram 语言模型
- Update：rnnlm 重打分
- 问题：信道中公司销售识别正常，另一信道客户方识别有部分丢失现象。

标注语音

- 语音数据：1000小时
- 普通语音：600小时+400小时
- 文本预料：15G

Offline-multi-condition-rescore

- 背景：2000小时语音，1000小时公司数据，1000小时公共数据，15G文本数据
- 声学模型：multi-condition-model
- 语言模型：3gram 语言模型
- rescore：rnnlm 重打分
- 结果：识别错误率 12-10%

多阶语言模型

- 背景：2000小时语音，1000小时公司数据，1000小时公共数据，50G文本数据+100G公共新闻教育数据+50G贴吧教育相关数据
- 声学模型：multi-condition-model
- 语言模型：3gram 语言模型
- 2阶：5gram 重打分
- 3阶：rnnlm 重打分
- 结果：识别错误率 11-9%

添加语音数据集

- 背景：2500小时语音，1500小时公司数据，1000小时公共数据，50G文本数据+100G公共新闻教育数据+100G贴吧教育相关数据
- 声学模型：multi-condition-model
- 语言模型：3gram 语言模型
- 2阶：5gram 语言模型
- rescore：rnnlm 重打分
- 结果：识别错误率 9-7%

Fine-tuning

- 背景：1000小时公司数据，50G文本数据+100G公共新闻教育数据+100G贴吧教育相关数据
- 声学模型：multi-condition-model

- Pre-mode：4000 小时声学模型
- 语言模型：3gram 语言模型（单独编译程G.fst可以放进入了）
- 2阶：5gram 语言模型
- 3阶：rnnlm 重打分
- 结果：识别错误率 8-6%

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

机器配置

- 训练机器配置：

  3台，40核、128G内存、4卡，服务器

- 主要训练方式：

  使用Grid Engine 进行集群构建，NFS服务器作为统一文件系统。

- 语音识别现状
- Kaldi介绍
- 语音识别流程
- 语音识别算法
- 案例讲解
- 机器配置
- 问题解决

语音切割

- 有没有更好的切割方法呢?
  - 答案：目前没有更好的切割方法，切割是一个比较难解决的问题
- 根据我们现在的切割方法有个问题，就是一个语音文件往往对应多句话，而看到的aishell等开源语音都是一句话一个语音文件的，哪种更好呢，多句话的可不可以接受？
  - 答案：腾讯目前的切割是很好的，一个语音文件对应多句话可以接受，只要少于半分钟的语音一般都没啥问题
- 切割语音文件完了之后，怎样判断语音文件的质量是否满足训练的需求，需不需要语音去噪，语音增强等技术，会有多大的效果？
  - 答案：语音可以满足训练需求，语音去噪一般是在车间等非常嘈杂的环境下才需要，我们这个场景暂时不需要。可以考虑将语音做加噪声等操作来扩充语料。
- 训练语料预处理，一般要关注哪些点，比如采样率、单声道，还需要关注些什么呢？处理后的录音要达到什么要求？
  - 答案：一般电话的都是8khz，双声道，我们使用单声道训练就可以。

语言模型

- 如何训练一个垂直领域的好的语言模型
  - 答案：就用n-gram模型加上RNN就可以，RNN上线的话不行，速度比较慢
- 如何更好的维护这个语言模型，使得效果越来越好
  - 答案：不停的加入更多正确的语料，纠正之前错误的语料
- kaldi里面的这个语言模型，是需要做二次开发的吧
  - 答案：主要做解码器的开发，会有大量的开发工作

公司实际问题

- 短文本识别效果差，短指的是少于7个字，这个是为什么，怎么解决这个问题
- 答案：短文本大家识别效果都差，因为语言模型没办法产生效果
- 我司使用kaldi做语音识别可行吗，需要什么样的服务器进行训练，在线预测应该要做二次开发吧
- 答案：可行。解码器会有大量的开发工作
- 我们短期目标是离线可以用kaldi，长期目标的话，可能需要在线预测，要求比较高的实时性，可能用户讲完1句话我们在0.5s内就要翻译成文本，kaldi可以做这种在线识别吗？
- 答案：肯定可行，离线和在线都用tdnn就行，这个时效性比较强

语音标注

- 类型
  - 标注（贵） 400-800元
  - 纠正（便宜） 100-200元

- 现状
  - 已经有语音的识别结果和对应的时间 **（腾讯的ASR接口希望可以带上置信度打分功能）**

- 选择类型

  语音纠正 **（使用腾讯的asr文本做一个参考）**

- Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition
- Low latency acoustic modeling using temporal convolution and LSTMs
- Purely sequence-trained neural networks for ASR based on lattice-free MMI
- Cepstral mean and variance normalization in the model domain
- Speaker Verification Using Adapted Gaussian Mixture Models
- Implementation of the Standard I-vector System for the Kaldi Speech Recognition Toolkit
- Notes on the derivative of SVD
- JHU aspire system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMs
- Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks

- SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition

• 其他

- 1.DFSMN效果：
- 阿里公布的效果是在5000多个小时中文上训练，wer可以达到96%多点，但是并没有公布语料还有其中的参数，作者等人对其进行复现，发现复现不出那么好的结果，比较鸡肋。TDNN及其改进算法在kaldi里面一直有人对其改进，但是如果在aishell里面进行训练模型的话，需要改代码，调用神经网络相关算法，同时aishell是针对150个小时数据的，当换更多语料（上千小时）的时候aishell里面代码将不可以拿来训练，需要考虑集群。
- 2.可以试着将自己的语音先加到几千小时，先不做标注，拿来预测试下
- 3.kaldi有可迁移训练模型，需要查找相应资料，避免数据重复训练
- 4.标点符号问题：kaldi预测出来的是没有标点符号的，标点符号需要后期进行处理
- 5. 为什么kaldi预测出来的会有很多识别不出来？确实就是kaldi没有识别出来，可以在语音模型和语言模型上下功夫，多增加数据，如果还是没有结果的话，那就没办法。
- 6.我们的文本存在大量的错误，用这些文本做语言模型靠谱吗？靠谱的，因为当你数据量足够大时，一些错误的识别可以忽略，同时还需要采集大量的交友（婚恋相关领域）文本，拿来一起做语言模型。
- 7. aishell里是有多个算法，这些算法之间有依赖吗？有依赖，dnn依赖hmm的模型
- 8.你们最终的语音模型是什么？chain+tdnn,chain+tdnn-f等模型做对比，发现效果差不多。就是跟aishell训练流程差不多。所谓的multi-condition-model就是做了数据增强的model。 数据增强方法前面有讲