

Contents

1 ASR 任务	1
2 发音字典 Lexicon	1
3 Acoustic Model 声学模型	1
3.1 GMM+HMM (Monophone)	2
3.2 GMM+HMM (Triphone)	2
3.3 NN+HMM (Monophone)	3
3.4 TDNN+HMM (Triphone)	3
3.5 CNN+TDNN+HMM(Triphone)	4
3.6 NN+CTC(Triphone)	5
3.7 CNN	6
3.8 RNN / RNN	7
3.9 Attention-Based / Transformer	9
4 Language Model 语言模型	9
4.1 N-Gram	9
4.2 RNN	9

1 ASR 任务

ASR 任务表示为

$$W^* = \operatorname{argmax}_w P(W/X) W^* = \operatorname{argmax}_w P(X/W) P(W)$$

其中 X 为语音提取后的特征向量, W^* 为语句输出, $P(X/W)$ 使用声学模型建模, $P(W)$ 使用语言模型建模。

2 发音字典 Lexicon

考虑对单词/单字建模, 英语与中文常用单词/单字的数目十分庞大, 并且训练集的规模限制导致极易出现 OOV(Out Of Vocabulary) 问题。

考虑对音素建模, 英语中音素总量约 50 个, 这极大的减少了模型数量。

同时, 每个音素的声学特征在随发音过程而变化, 通常将每个音素划分为 3 状态, 即开始, 稳定与结束状态。

问题在于模型如何确定某个单词由哪些音素构成? 事实是模型无法确定, 这种映射需要引入人工发音字典。

3 Acoustic Model 声学模型

输入语音, 输出句子, 只考虑语音与文字的对应关系, 不考虑文字间的概率, 即语义语法等关系。

3.1 GMM+HMM (Monophone)

使用 GMM(高斯混合模型) 与 HMM(隐马尔可夫模型) 构成单音素模型, GMM 对每个状态建模, HMM 对每个音素建模。

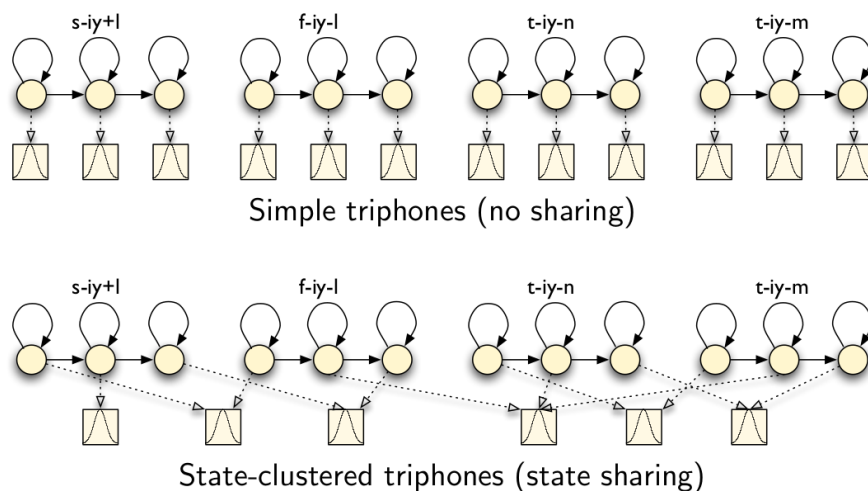
1. 将输入语音切分提取MFCC特征帧
2. GMM模型预测特征帧属于某音素某状态的概率
3. 使用Viterbi算法计算每个单词的HMM生成该序列的概率
4. 选择最大概率的单词作为输出

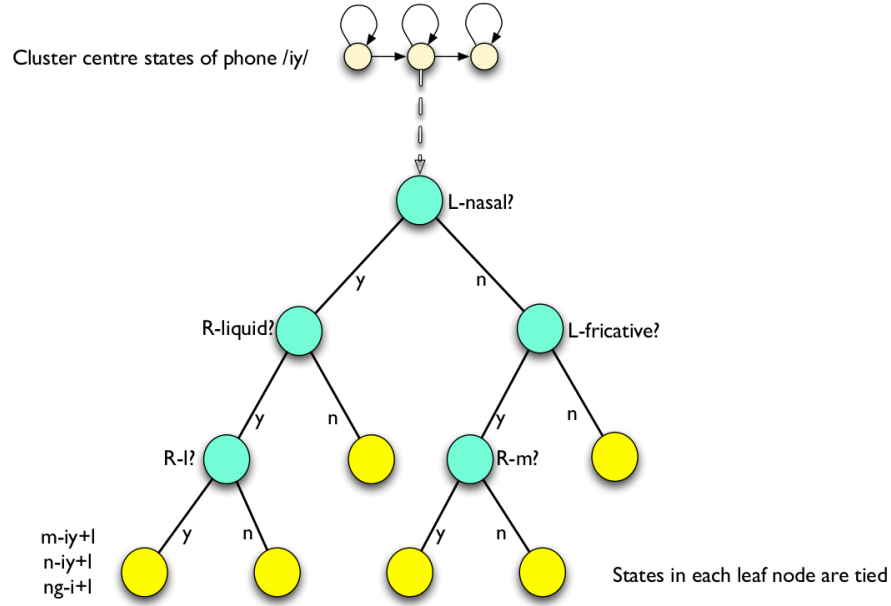
对 N 个单词三状态的语料库, 需要 N 个 HMM 与 $N * 3$ 个 GMM, GMM 分量则需要人工设定。

3.2 GMM+HMM (Triphone)

Triphone 三音素预测考虑了相邻音素对该音素发音的影响, N 个音素的集合理论上可构成 N^3 个 triphone, 假设每个 triphone 有三个状态, 直接建模需要 $N^3 * 3$ 个 GMM, 随着状态划分的增长, 音素规模的增大, GMM 模型的数量极其庞大。

考虑引入状态共享, 即不同音素的状态可以共享同一个 GMM 模型, 这种划分通常使用决策树完成。



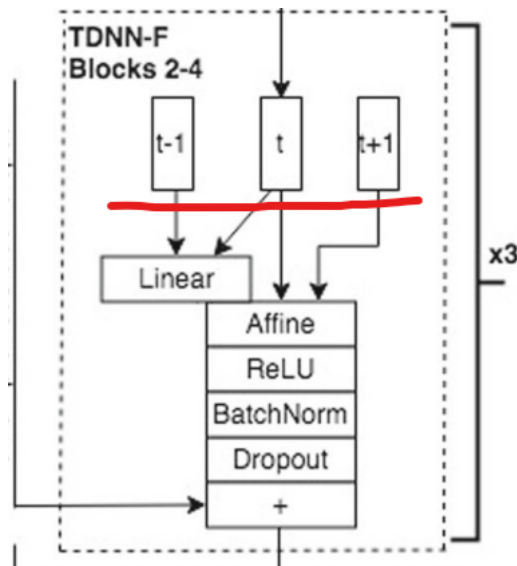


3.3 NN+HMM (Monophone)

使用神经网络代替 GMM 输出观测概率。

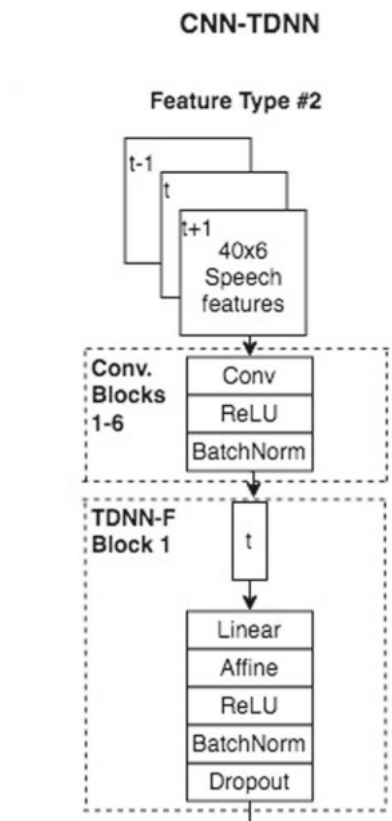
3.4 TDNN+HMM (Triphone)

TDNN 引入时序依赖，与 Triphone 相似，可以建模相邻音素 V_{t-1}, V_{t+1} 之间的影响，同时得益于神经网络权值共享的概念，可以对更长时间依赖的音素 V_{t-3}, V_{t+3} 进行建模。



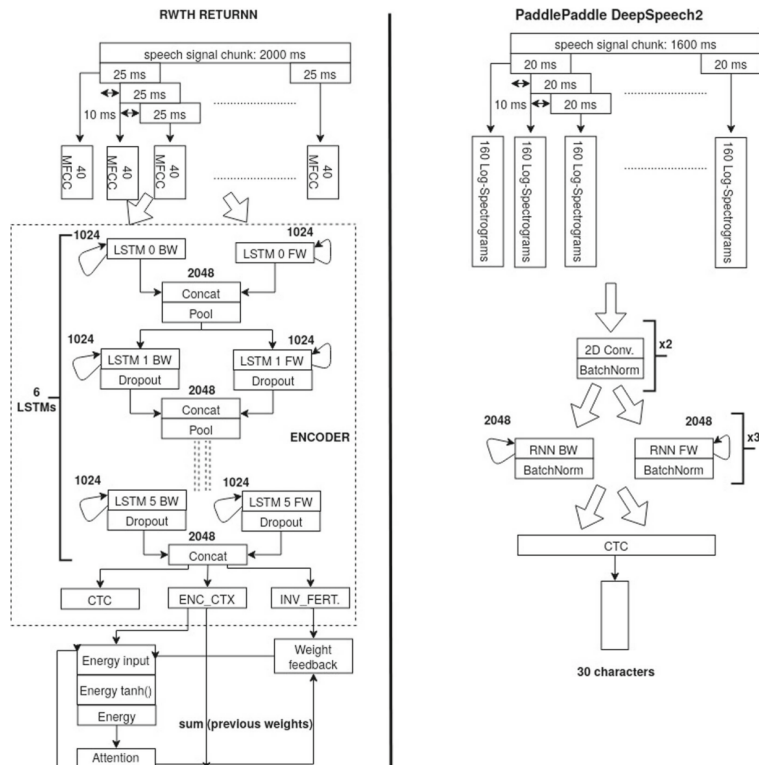
3.5 CNN+TDNN+HMM(Triphone)

使用 CNN 权值共享的特征，针对每个输入特征对 V_{t-1}, V_t, V_{t+1} 进行卷积建模。



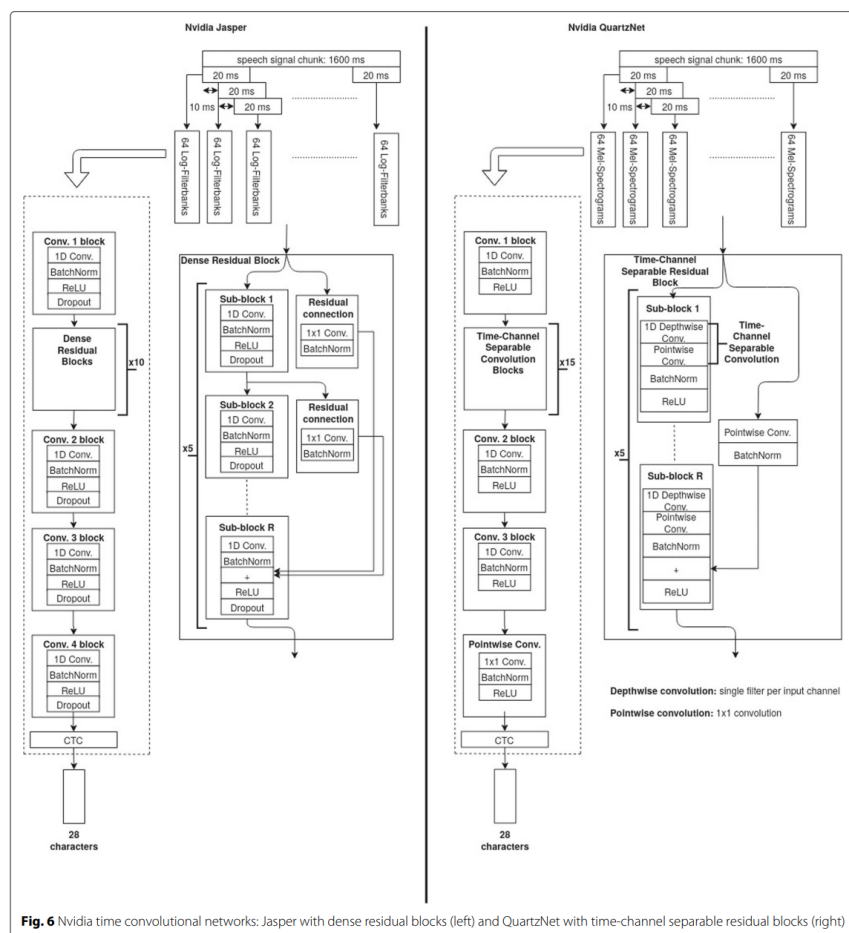
3.6 NN+CTC(Triphone)

这部分模型使用 RNN 替代 TDNN/CNN 等进行更强的时序建模，拓展时序依赖关系，同时将 HMM 替换为了 CTC 模型，来解决变长序列的建模问题。



3.7 CNN

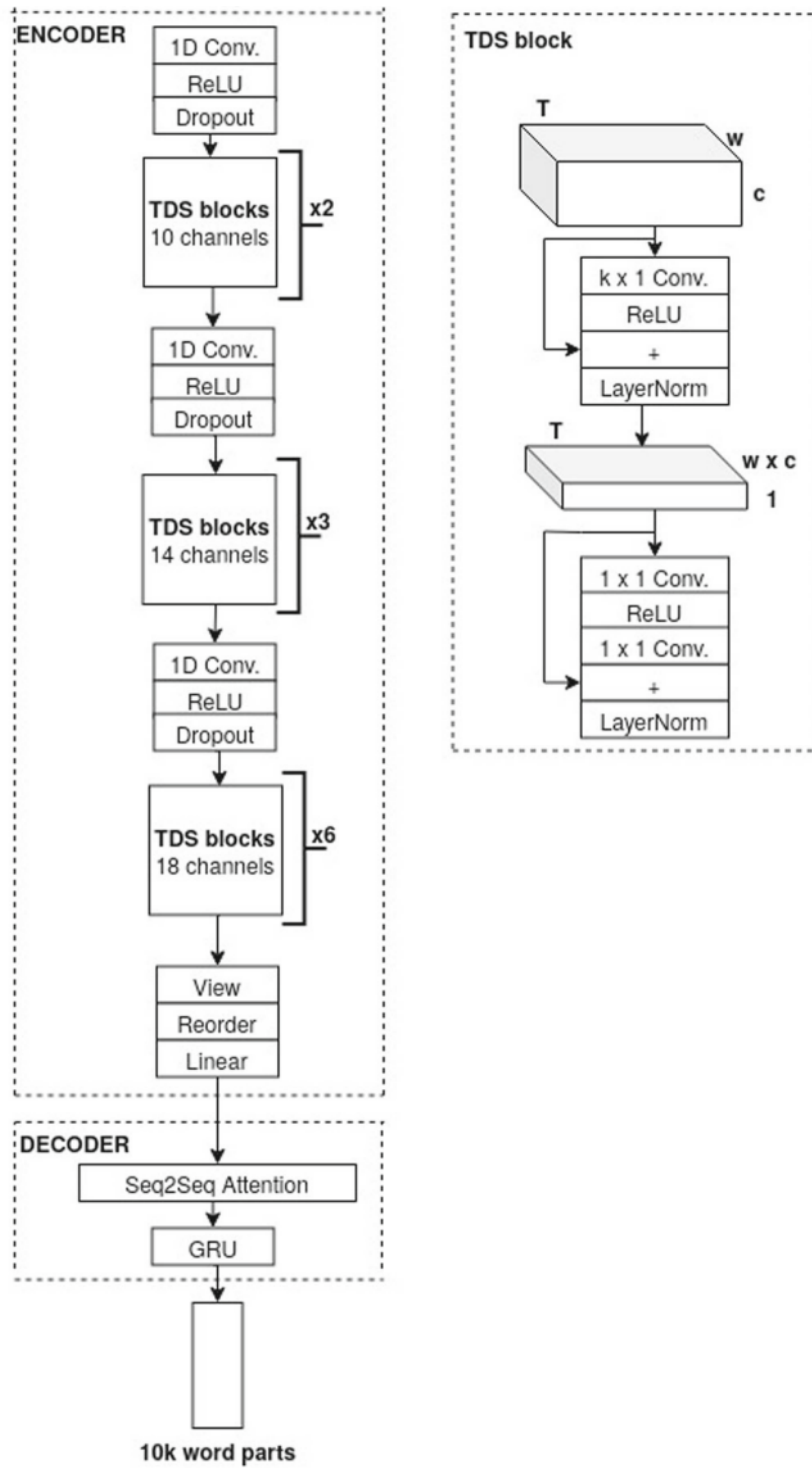
使用纯 CNN 架构, 参数量与计算量均较大。



3.8 RNN / RNN

这部分模型往往使用 Encoder/Decoder 结构，RNN 替代 TDNN/CNN 等拓展时序依赖关系进行更强的时序建模作为 Encoder，Decoder 同样使用 RNN。

输入仍然是 MFCC 特征提取后的特征向量。



3.9 Attention-Based / Transformer

计算量过大，嵌入式可行性存疑。

4 Language Model 语言模型

建模文字间的依赖关系，语言模型与声学模型相对独立。

4.1 N-Gram

引入 N 阶马尔可夫假设，即当前文字先验概率只与前 N 个文字概率相关。

使用 N-Gram 计算当前单词序列/句子 $W = W_1 \dots W_K$ 的先验概率为：

$$P(W_1 \dots W_K) = \prod_{i=1}^K P(W_i / W_{i-1} \dots W_{i-N})$$

N-Gram 基于统计

4.2 RNN

Seq2Seq 方式建模条件概率

$$P(W_1 \dots W_K) = \prod_{j=1}^K \prod_{i=1}^j P(W_i / W_{i+1} \dots W_{i+j})$$

神经网络输出为

$$\{P(W_2/W_1), \dots, P(W_K/W_1 \dots W_{K-1})\}$$