CSLT TECHNICAL REPORT-20160016 [Thursday 5th January, 2017]

多任务joint training的训练方法

Hang Luo

Correspondence:
luohang@cslt.riit.tsinghua.edu.cn
Full list of author information is available at the end of the article

Abstract

近年来深度学习技术在语音识别领域得到广泛应用，然而目前的神经网络模型主要是处理某一单一任务，例如自动语音识别 [1]或者说话人识别 [2]。这种处理单一任务的模式，显然不符合人类大脑的工作方式，也不利于任务之间知识的共享和交互。基于此， [3, 4] 提出了joint training方法，此方法通过训练一个统一模型来完成多个任务的识别，且在训练过程中通过子任务识别模型的信息互相交互，使得子任务识别的准确率互相提高。本文将主要关注自动语音识别和语言识别的统一训练，介绍 [3] 中joint training的具体实现。同时，本文假设读者已基本了解语音识别流程，故将不再详细叙述语音识别过程中的诸多细节。读者阅读后也可以实现其他任务的统一训练。

Keywords: joint training; 语音识别; 语言识别

1 简介

图1展示了语音识别和语言识别做joint training时的总体框架。图中可以看到语言识别（ASR）中声学模型和语言识别模型（LRE）共享提取特征的过程，且在训练时互相交互信息。为了实现以上系统，需要在WSJ nnet3 recipe上做出4个方面的修改，分别对应图1中的：数据准备，语言识别模型，语言模型，声学模型四个部分。本文使用中文thchs30和英文aurora4数据集，且假设已分别按照nnet3 recipe单独运行（对应 [3]中的baseline1实验）。接下来将详细介绍每个部分的修改。

2 数据准备

数据的准备包括两部分。一部分与基本数据有关，其中主要包括录音文件，标准文本等和“原始数据”相关的文件，另一部分包括发音字典，音素集合等和“字典”相关的文件。由于我们已假设单独运行过thchs30和aurora4实验，故以上文件均已得到，我们只需对其进行融合即可。

其中，“原始数据”相关的文件如下， [3]中使用的特征为40维的fbank特征：

cmnv.scp	feats.scp	spk2utt	text	utt2spk	wav.scp

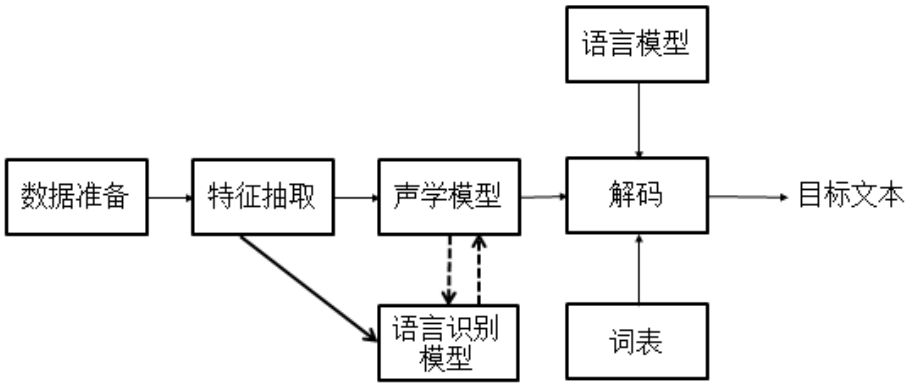

Figure 1: 语音识别和语言识别的joint training框架

对其可使用utils/combine_data.sh进行融合,使用方法如下：

```
utils/combine_data.sh  dest-data-dir  src-data-dir1  src-data-dir2
```

“字典”相关的文件则如下：

```
extra_questions.txt  lexiconp.txt  lexicon.txt  nonsilence_phones.txt
optional_silence.txt  silence_phones.txt
```

对其直接合并且去掉重复项即可，其中稍微需要注意的是，静音在英文lexicon.txt中标注为！SIL，在中文lexicon.txt中为SIL。融合时统一保留为！SIL。在特征提取阶段，kaldi根据融合的文件生成L.fst。

3 语言识别模型

在图1中可看到声学模型与语言识别模型共享相同的特征提取模块，其中声学模型所学习的目标由GMM模型跑完后标注。而语言识别模型的目标则需要人工进行标注。如果不考虑静音的影响，可将每帧的语言标识标注为0和1，分别代表英文和中文。用feat-to-len命令可得到每句话帧的数目，之后根据当前句子的语言类别对其进行标注即可。feats-to-len使用方法如下：

```
feats-to-len  in-rspecifier  out-wspecifier
```

4 语言模型

经过以上准备，中英文混合训练集可以使用一个统一的GMM模型进行训练，但是其使用的语言模型（LM）还是分离的，如图2所示。为了统一处理多种语言，

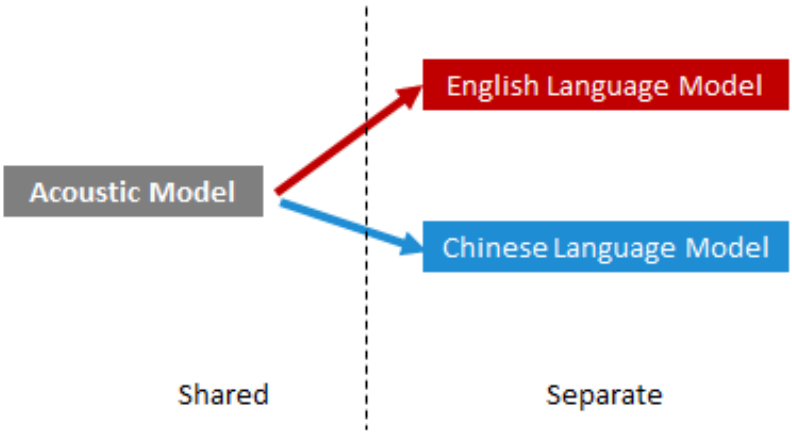

Figure 2: 分开的语言模型

我们需融合语言模型，使用一个语言模型进行解码，如图3所示。此时即对应 [3] 中的baseline2 实验。

由于在此之前单独运行过aurora4和thchs30，因此假设已经有两个数据集的语言模型。融合两种语言模型时，可使用SRILM工具 [5]。使用n-gram命令融合语言模型，n-gram模型融合时使用方式如下：

ngram -lm $mainlm - order$num -mix-lm
$mixlm - lambda - write - lm$mergelm

其中每个参数的含义为：

-lm	这个参数代表被融合的模型
-order	使用最大ngram长度
-mix-lm	代表加入的模型
-lambda	代表加入的模型在融合时的权重
-write-lm	最终的融合模型

前面提到，在特征提取阶段，kaldi根据utils/prepare_lang.sh将融合的lexicon文件生成L.fst，在解码时用于寻找phone对应的最佳word的路径。同样，在此阶段，utils/format_lm.sh 也会将融合的模型处理为G.fst 格式，用于在解码时寻找word词组对应的最佳sentence的路径。加上HMM相关内容及音素的上下文相关信息，utils/mkgraph.sh会生成HCLG.fst文件，用于将声学模型的输出解码成最后结果。

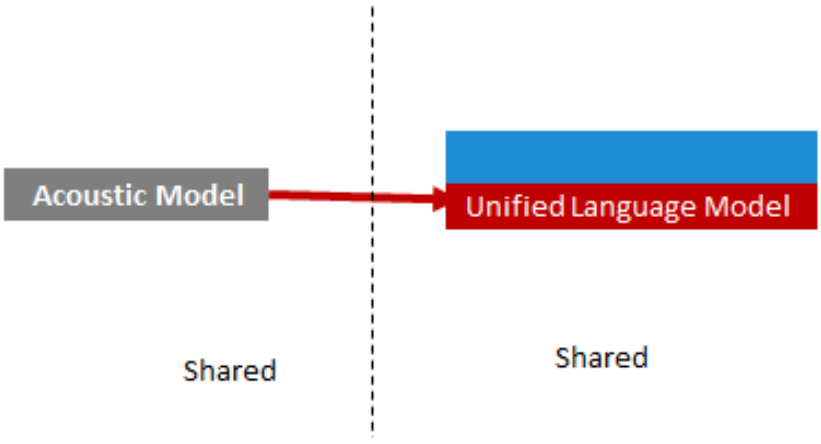

Figure 3: 统一的语言模型

5 声学模型

在单任务的语音识别任务中，声学模型中使用的神经网络仅对应一个输出，即各pdf的概率。但是，由于语言识别模型的加入，融合模型最后有两个输出。因此，一是需要改变神经网络中用于对齐标注的egs数据，使得每个输出对应两个输出的标注数据。二是需要改变原神经网络的结构，使得声学模型和语言识别模型之间能够互相交互也统一训练。对于第一个修改，改写nnet3-get-egs为nnet3-get-egs-double-targets，同时相应改写原recipe中的get_egs.sh和训练神经网络脚本。另外，生成了融合的egs 后，可以在每次跑不同网络时设置common_egs_dir，这样可以避免每次训练时重复生成egs。修改后文件可参考：

/work3/tzy/github/kaldi_master_20160126_interspeech16_cuda7.0/src/nnet3bin/nnet3-get-egs-double-targets.cc

---

对于第二个修改，则通过修改神经网络的配置文件实现。由于存在模型之间的交互，所以一般都事先定义好配置文件，并跳过配置文件自动生成的步骤。以图4为例，图中声学模型和语言识别模型将各自模型的信息，通过g门进行交互。在配置文件中定义两个LSTM，在g门的输入中加入另一个模型的recurrent和non-recurrent信息，具体如下：

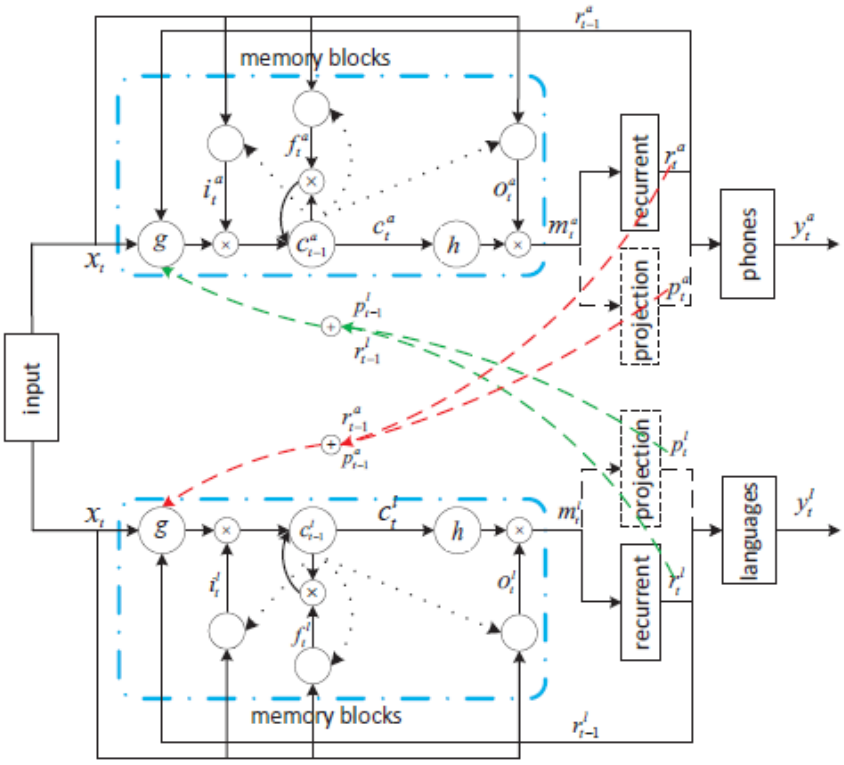

Figure 4: 语音识别和语言识别的joint training框架

在声学模型中，给g门输入加入语言识别模型信息:
component-node name=Lstm1_g1 component=Lstm1_W_c-xr input=Append(L0_lda, IfDefined(Offset(Lstm1_r_t, -1)), IfDefined(Offset(lan_Lstm1_rp_t, -1)))

在语言识别模型中，给g门输入加入声学模型信息:
component-node name=lan_Lstm1_g1 component=lan_Lstm1_W_c-xr input=Append(L0_lda, IfDefined(Offset(lan_Lstm1_r_t, -1)), IfDefined(Offset(Lstm1_rp_t, -1)))

同理，也可改动配置文件使得其他门接收信息。对于此配置文件的完整版本，可参考：

/work3/luohang/speech_lan/bilingual/exp/nnet3/lstm_joint_g_s_2_p/configs/all.config

6 总结

本文主要介绍kaldi中自动语音识别和语言识别joint training的实现流程，对于想要复现及修改论文 [3]中模型的读者有一定的参考意义，同时读者也可参考此文档实现其他任务间的joint training。由于文章主要面向有一定kaldi基础的读者，文章在许多实现细节处不免都介绍地相对简略。此外，文章中使用的数据库每句话均只由一种语言构成，有兴趣的读者也可以将此方法用于解决mixlingual的问题。

7 推荐读物

http://kaldi-asr.org/doc/
https://www.inf.ed.ac.uk/teaching/courses/asr/
http://deeplearning.net/reading-list/
Automatic Speech Recognition A Deep Learning Approcach By Dong Yu and Li Deng
Deep Learning by Yoshua Bengio,Ian Goodfellow,Aaron Courville

References

1. Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82-97.
2. Lei Y, Scheffer N, Ferrer L, et al. A novel scheme for speaker recognition using a phonetically-aware deep neural network[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 1695-1699.
3. Multi-task Recurrent Model for True Multilingual Speech Recognition, Zhiyuan Tang, Lantian Li and Dong Wang
4. Multi-task Recurrent Model for Speech and Speaker Recognition, Zhiyuan Tang, Lantian Li and Dong Wang
5. SRILM –AN EXTENSIBLE LANGUAGE MODELING TOOLKIT, Andreas Stolcke