**声学模型训练文档**

版本：<1.1.0>

**中国科学院声学研究所**

**2019年6月**

改版记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 发布日期 | 描述 | 作者 | 复审 | 批准 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1 机器环境 4](#_Toc11862619)

[2 环境说明 4](#_Toc11862620)

[3 源码环境 4](#_Toc11862621)

[3.1 依赖的第三方软件 4](#_Toc11862622)

[3.2 编译流程 5](#_Toc11862623)

[4 训练环境 5](#_Toc11862624)

[4.1 训练数据 6](#_Toc11862625)

[4.2 利用种子模型进行ali 6](#_Toc11862626)

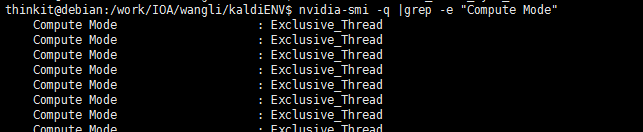
[4.3 声学模型的retrain 7](#_Toc11862627)

[4.4 声学模型转换 9](#_Toc11862628)

机器环境

声学模型的训练需要在64位装有Debian8.5的GPU机器上进行。

在模型训练之前必须运行：nvidia-smi -c 1 ，修改gpu工作模式为：独占模式。查看gpu工作模式指令：nvidia-smi -q |grep -e "Compute Mode"）。



环境说明

声学模型的训练依赖两个环境：训练环境和训练源码，训练环境为：kaldi\_env\_20180420scripts\_tongyong，通过调用源码环境中的可执行文件进行声学模型的训练。

训练源码为：kaldi-master-20180420。

源码环境

* 1. 依赖的第三方软件

训练源码依赖的第三方软件如下：

1、gcc/g++ 4.9 （必须是这个版本）

2、automake 包含aclocal

3、libtool 包含libtoolize、 glibtoolize

6、zlib1g-dev 包含zlib.h

7、libatlas-dev

8、libatlas-base-dev

9、gawk 支持脚本使用的awk版本（gawk）相关

10、subversion

11、git

12、gfortran

13、bzip2

14、python2.7

15、cuda 8.0（依赖于具体的机器类型）

16、NVIDIA-Linux-x86\_64-352.99（可以通过cuda安装配套版本）

* 1. 编译流程

源码环境中有对应的安装说明INSTALL，可按照INSTALL进行安装，或者按照如下步骤安装。

1、进入 tools目录

1）检查依赖软件是否齐全及版本是否符合要求，运行：

bash extras/check\_dependencies.sh

2）编译第三方软件运行： make

2、进入 src目录

1）./configure --shared

2）make depend

3）make

训练环境

训练环境的目录结构包含以下内容

1、path.sh：用于设置源码环境路径，以便于调用可执行文件。

2、cmd.sh：用于设置并行环境，一般不需要动

3、conf：训练过程中涉及到的参数文件，不需要修改

4、data：训练过程中涉及到的数据资源

data/lang：编译后的字典信息目录，一般不需要修改

data/train：训练数据目录，存放：text、wav.scp、utt2spk、spk2utt文件

data/local：字典及音素文件，一般不需要修改

5、exp：模型输出目录

6、steps、utils及scripts中是一些包装好的脚本，不需要修改

7、local：最顶层脚本目录，需要运行的脚本一般放于此目录。

8、run\_blstm\_align.sh及run\_blstm\_extend.retrain.sh是retrain需要的脚本

* 1. 训练数据

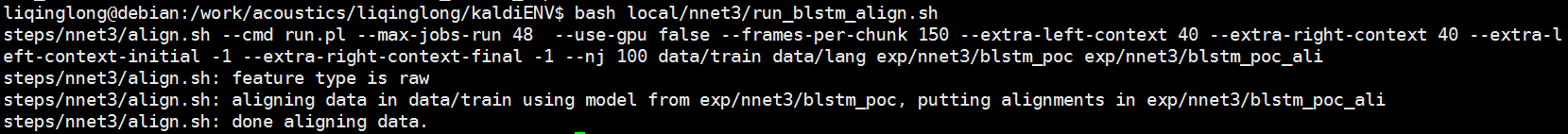
将数据预处理中trainDir目录下产生的训练数据文件（text、wav.scp、utt2spk、spk2utt）拷贝到data/train目录下。

* 1. 利用种子模型进行ali

这一步是利用准备好的数据，以及基线模型，产生状态级的时间信息。具体操作如下：

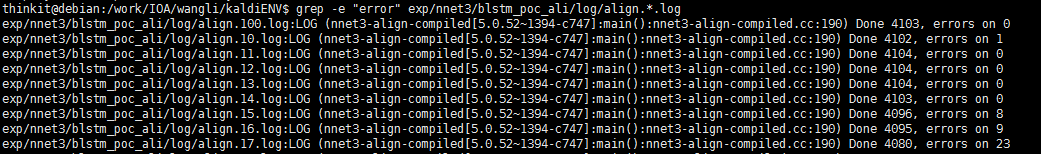
1. 进入data/train目录
   1. 此目录下面的有效文件为text、utt2spk、spk2utt、feats.scp
   2. 查看用于训练的训练数据：
      1. 进入训练目录kaldiENV
      2. 运行 . path.sh
      3. 进入data/train
      4. 运行：feats-to-len scp:feats.scp ，会在屏幕上打印出用于训练的总帧数N，则总时间为：N/360000 （小时）
2. 进训练环境
3. 运行脚本：run\_blstm\_align.sh，脚本中的参数说明：
   1. srcdir=设置用来ali的模型目录（即基线模型的目录）
   2. train\_data\_dir=设置训练数据目录（一般是：data/train）
   3. alidir=设置输出目录
   4. –nj 100：job数（一般不超过CPU核数）
   5. frames\_per\_chunk=150（根据基线模型进行设置）
   6. extra\_left\_context=40（根据基线模型进行设置）
   7. extra\_right\_context=40（根据基线模型进行设置）
   8. extra\_left\_context\_initial=-1（根据基线模型进行设置）
   9. extra\_right\_context\_final=-1（根据基线模型进行设置）

ali完成后屏幕上打印的信息：



1. alidir/log/align.\*.log中记录了详细的ali信息，如果有什么错误，可以到此log中查找原因。
2. ali完成后，可以执行：

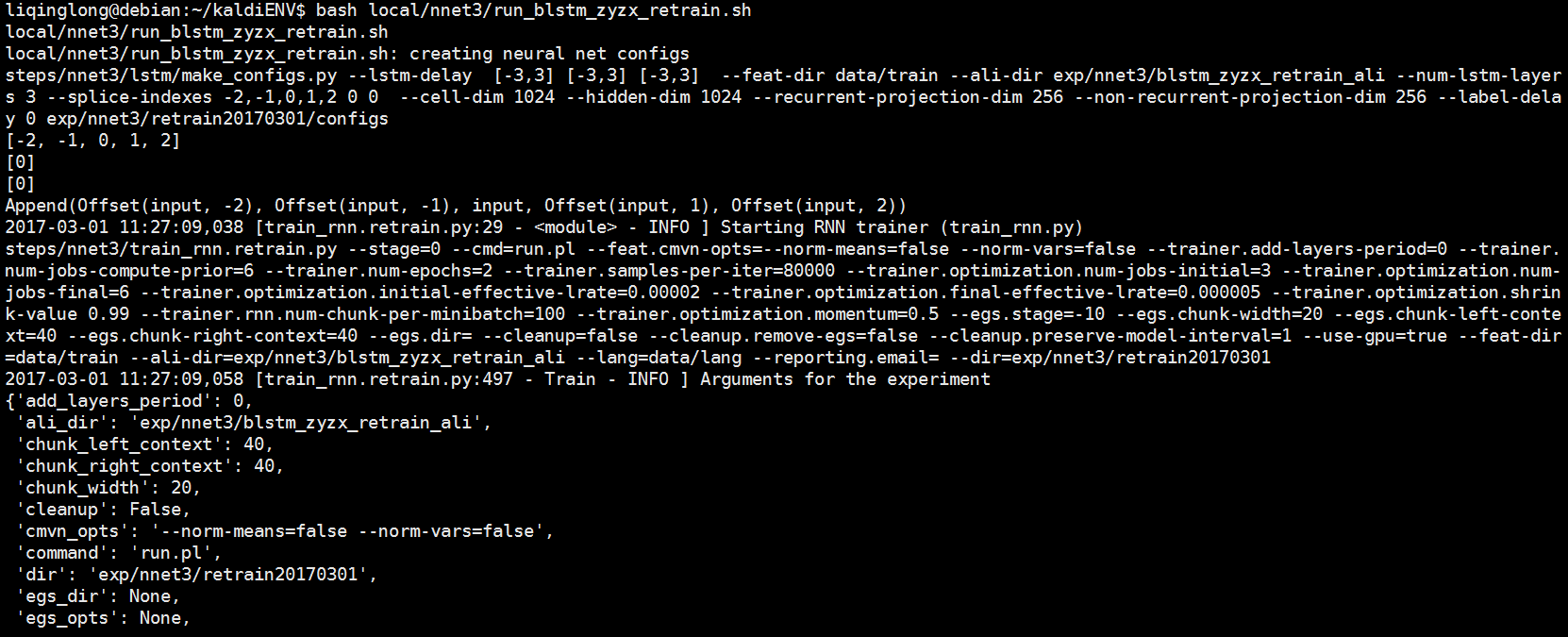
grep -e  "error"  alidir/log/align.\*.log 查看ali失败的文件个数，如果文件数比较多，说明ali有问题。

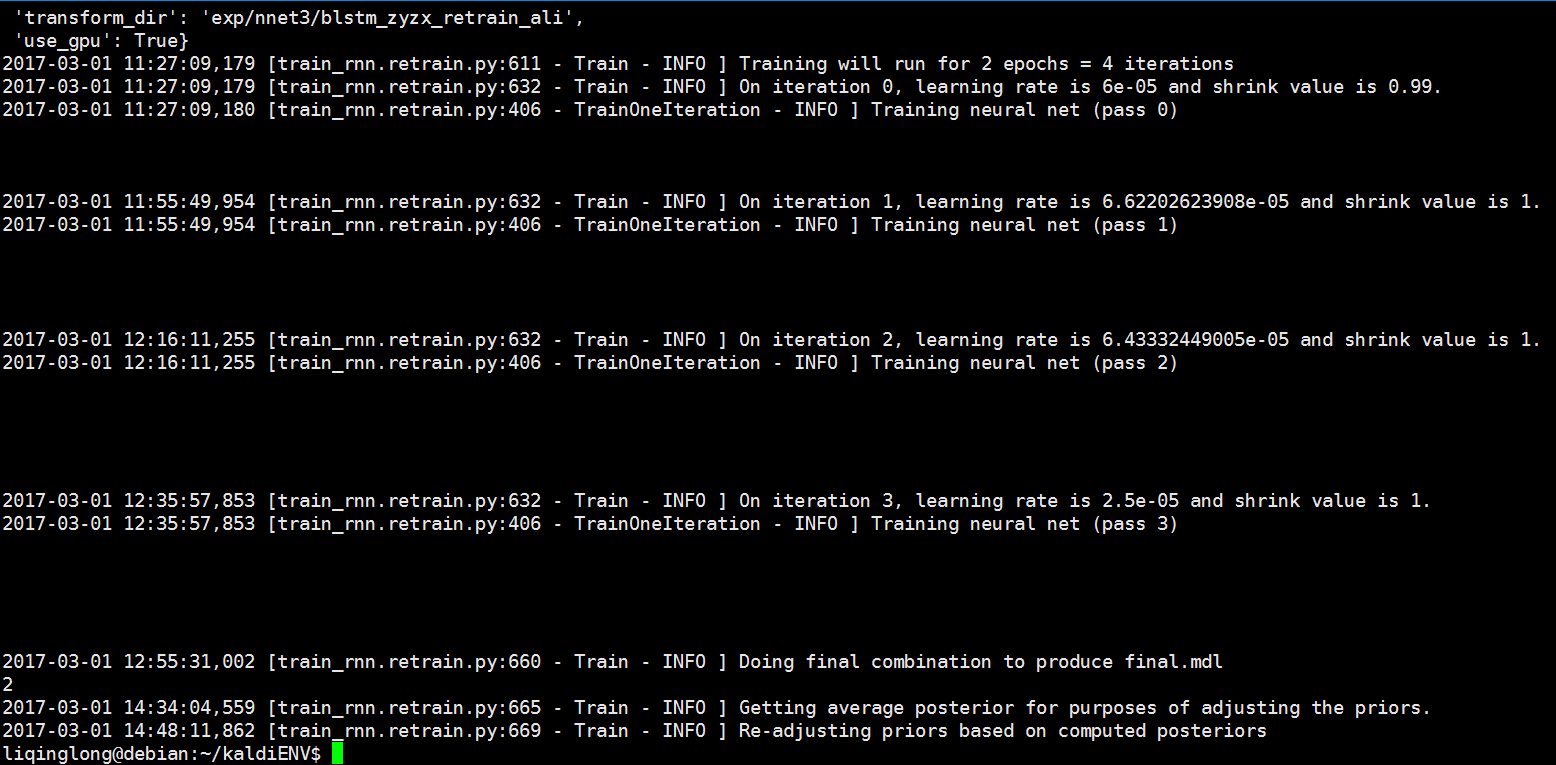


* 1. 声学模型的retrain

这一步是利用产生的ali结果以及准备的训练数据，对模型进行更新，具体操作如下：

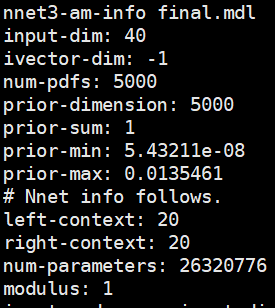
1. 进入训练环境
2. 修改run\_blstm\_extend.retrain.sh脚本中的参数
3. 开始运行时，stage=-10，train\_stage=-10，如果运行过程中断了，则需要通过这两个参数控制，从中断的地方继续运行。
4. dir=设置模型输出目录
5. ali\_dir=设置上一步产生的ali结果目录
6. data\_dir=设置训练数据目录
7. 运行：bash run\_blstm\_extend.retrain.sh





如中间有错误会有中断，调试后在run\_blstm\_extend.retrain.sh中修改train\_stage为对应的中断点后，重新执行脚本。

1. 模型查看工具
   1. 进入训练目录
   2. 运行 source ./path.sh
2. 进入模型目录
   1. 运行：nnet3-am-info final.mdl



input-dim：40 是输入的特征维度

num-pdf：5000是状态数

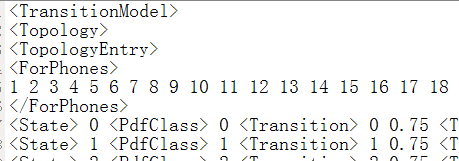
* 1. 声学模型转换

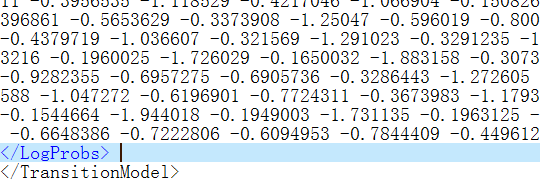
这一步是将训练的模型转换为可利用的TIT格式，具体操作如下：

1. 进入训练环境
2. 将产生的final.mdl模型文件转换为文本格式：
3. 进入训练环境
4. 运行： source path.sh
5. 进入模型目录
6. 运行：nnet3-am-copy –binary=false final.mdl findl.mdl.txt
7. 将findl.mdl.txt文件中的<TransitionModel>\*</TransitionModel>这部分的内容删除，可使用sed命令进行删除：

sed -i '/<TransitionModel>/,/<\/TransitionModel>/d' final.mdl.txt

删除掉的内容为：





1. 将处理好的findl.mdl.txt放入模型转换运行环境下的model\_ori下，按照转换的文档进行转换。需要申请授权，license\_2.txt放到运行环境目录下。