《语音识别与合成》期末实验项目文档

1120200563 肖良寿 *

January 3, 2023

摘要

本实验共由两部分组成：基于kaldi软件thchs30例程的中文语音识别模型实验和基于MindSpore与Ascend 310的Tacotron2模型实验，本文档较详细地阐述实验的原理和功能，以及实验过程中的细节。

第一部分的实验参照kaldi官方给出的例程进行模型训练与评估，并在该基础之上制作了一个小型的用于演示测试效果的在线识别demo。第二部分的实验使用Tacotron2语音识别模型，参照MindSpore框架官方给出的模型以及实验说明，将LJ Speech作为数据集完成模型训练与推理。框架给出的官方模型本身很方便，而实践中，MindSpore框架的安装较为艰难。

1 kaldi实验

本部分阐述运用kaldi软件进行语音识别的实验。实验采用的示例为EGS下thchs30，进行普通话语音识别。实验采用的数据集为清华大学语音与语言技术研究中心(CSLT)发布的一个开放的中文语音数据库THCHS30[1]。并仿照EGS目录下$vox forge$示例制作在线识别模型进行演示，具体效果参见提交的演示视频。

1.1 实验环境及工具

- 华为云GPU加速型弹性云服务器，8核32G，Ubuntu20.04 64bit
- MobaXterm远程连接工具
- Python 3.8、PyTorch1.17

1.2 kaldi软件包的安装与配置

Kaldi[1]是一款十分流行的开源语音识别工具，使用WFST来实现解码算法，主要代码使用C++编写，在此之上使用bash和python脚本设计了一些工具。

Kaldi架构如图1所示，最上层是外部的工具，包括用于线性代数库BLAS/LAPACK和开源的开源的WFST库OpenFst。中间层是Kaldi的库，包括HMM和GMM等代码，下面是编译出来的可执行程序，最下面一层则是一下脚本，用于实现语音识别的不同步骤(比如特征提取，比如训练单因子模型等等)。

kaldi安装过程大致经过如下几步：自官网[2]下

*Email: xiaoliangshou.bit@gmail.com

[1]地址:http://www.openslr.org/18/

[2]https://github.com/kaldi-asr/kaldi.git

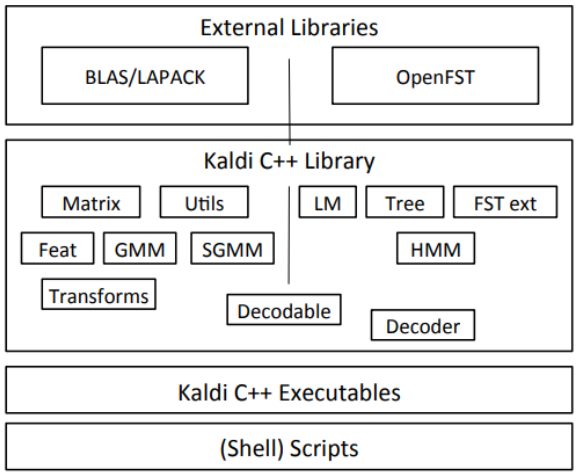

图 1: Kaldi架构

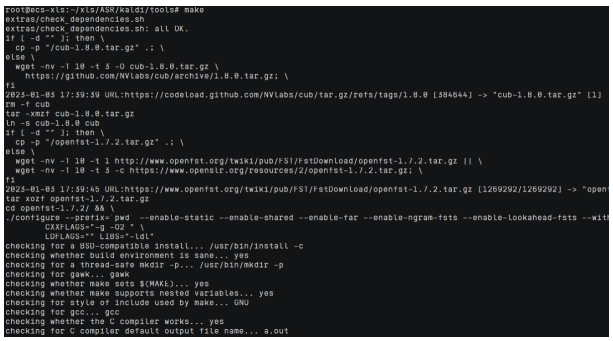

图 3: Tools目录编译过程

载源代码、进入源码的Tools目录执行编译、进入源码的src目录执行编译、检验是否安装成功。值得注意的是，根据kaldi给出的安装提示，两个目录的编译顺序不能颠倒。安装提示如图2所示。 编

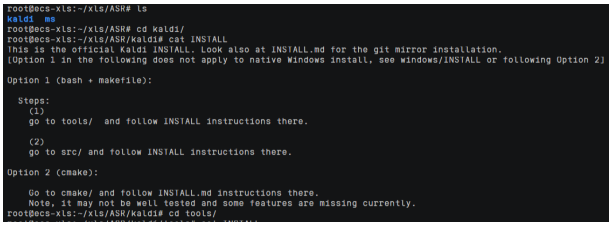

图 2: Kaldi安装提示

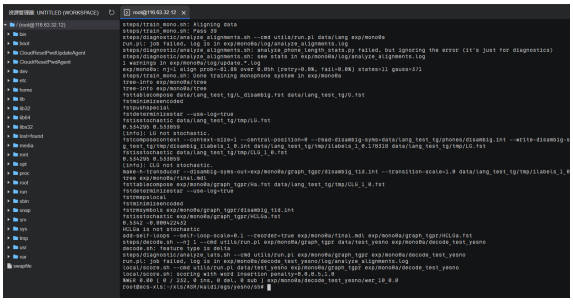

图 4: Kaldi安装成功

译过程由make命令执行，历时较长，尤其src目录下。Tools目录的编译过程如图3所示。 编译结束之后，执行EGS下简单的*yesno*例程进行检验。执行成功时如图4所示。

1.3 thchs30例程原理简介

kaldi主要用脚本来驱动，每个recipe下会有很多脚本。执行thchs30例程需切换至kaldi/egs/thchs30/s5目录下。经过训练后的该目录结构如图5所示。在s5路径下，有三个脚本：run.sh、cmd.sh和path.sh。其中run.sh是驱动整个例程的主脚本，在该脚本中组织执行了例程其他的全部脚本，启动例程训练也只需要执行这一脚本。run.sh中首先调用cmd.sh脚本，该脚本的具体作用为配置例程的运行环境，也是在启动项目之前需要提前修改的。具体来说，cmd.sh中需要根据运行环境是单机或GridEngine集群修改环境变量配置，单机时选择"run.pl"，多机器集群则选用"queue.pl"，一般情况下在单机多线程环境下运行，采用前者，本实验即是如此。path.sh脚本在执行后会修改一些环境变量，包括export LC_ALL=C，因为很多kaldi的工具需要数据是排序的，而排序需要用C语言字符串顺序。

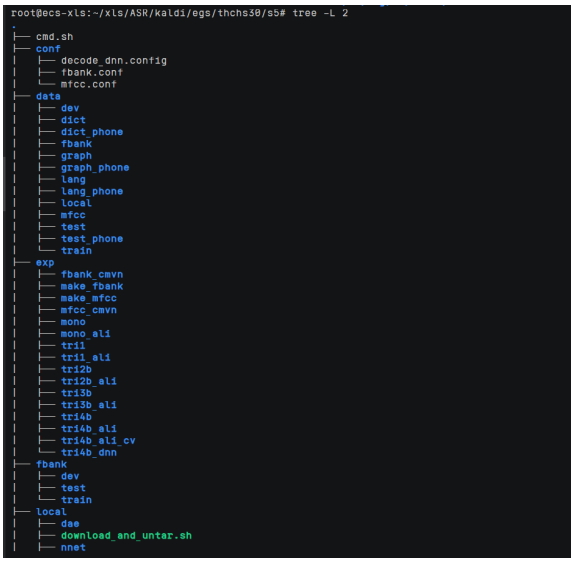

图 5: 训练后thchs30的目录结构

kaldi的一般训练过程如图6所示[3]，各个阶段涉及的相应脚本及功能见表1。thchs30例程中的一些脚本来自于wsj例程，如align_fmllr.sh、train_quick.sh等，由steps目录存放并导入，该目录是训练的步骤，被认为是最重要的脚本。

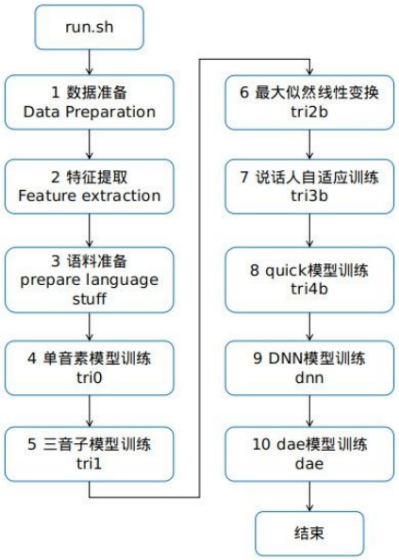

图 6: 一般的训练过程

1.4 数据集以及训练过程

本实验采用的数据集为清华大学语音与语言技术中心(CSLT)发布的开放中文语音数据库THCHS30。[2]该数据最早的录音是2002年发布的"TCMSD"，之后于2015年由王东博士发起，并由新疆大学和清华人工智能云研究中心合作出版了THCHS30，打破数据垄断，保护科研人员个人对自动语音识别(ASR)这一研究方向的初始利益。该数据集提供了包括词汇、语言模型和训练recipe在内的全套资源，让新的从业者可以使用

[3]来自于课程PPT

表 1: 脚本目录及其功能清单

功能模块	脚本名	作用
项目启动	run.sh	启动并控制整个模型的训练，执行其他脚本
	cmd.sh	配置运行环境，根据模型运行在单机或GridEngine集群上选择不同的脚本
	path.sh	环境变量相关的脚本
数据准备	thchs-30_data_prep.sh	生成text, wav.scp, utt2pk, spk2utt格式的文件
提取特性	make_mfcc.sh	提取Mel频率倒谱系数MFCC
	compute_cmvn_stats.sh	计算倒谱均值方差的归一化cmvn
语料准备	prepare_lang.sh	准备data/lang目录下的数据
	format_lm.sh	把arpa格式的语言模型转化成WFST
单音素模型训练	train_mono.sh	输入为之前准备的数据目录data/mfcc/train和lang目录data/lang，输出目录exp/mono
	thchs-30_decode.sh	monophone训练完成之后执行一次评估
三音素模型训练	train_deltas.sh	训练triphone模型
	thchs-30_decode.sh	triphone训练完成之后执行一次评估
	align_si.sh	数据对齐
最大似然线性变换	train_lda_mllt.sh	
	thchs-30_decode.sh	评估，同上述
	align_si.sh	同上述
说话人自适应训练	train_sat.sh	说话人自适应训练
	thchs-30_decode.sh	评估，同上述
	align_fmllr.sh	对齐，同上述
quick模型训练	train_quick.sh	quick模型训练
	thchs-30_decode.sh	评估，同上述
	align_fmllr.sh	计算训练的alignments
DNN模型训练	run_dnn.sh	DNN模型训练
dea模型训练	run_dae.sh	dea模型训练

这些资源来建立一个基本的大词汇量连续汉语语音识别系统。

由于云服务器自身的网络等原因，实验中先从数据集网站将其下载到本地，再通

过MobaXterm工具上传至服务器，节省数据获取的时间以及提高成功率。上传数据的过程如图7所示。

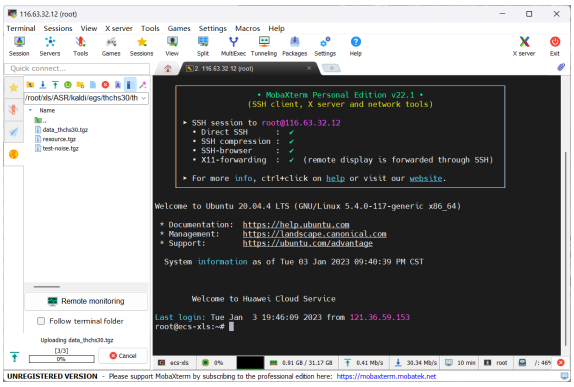

图 7: 数据上传过程

1.5 在线识别

参照kaldi给出的voxforge例程，本实验在原有模型训练的基础上制作了一个小型的demo，演示如何使用在线解码二进制文件，而二进制文件是可选的，从而需要进入Tools目录下通过执行install_portaudio.sh脚本来安装跨平台的音频I/O库portaudio，并进入kaldi软件根目录下src路径中执行命令"make ext"来编译ext。测试模式有两种：模拟在线解码模式，其中音频来自预先录制的音频片段；实时解码模式，使用语言和声学模型来识别他的语音。实验演示的测试模式为前者。

完成了工具部分后还需建立目录结构并同步识别模型。具体为，从voxforge例程目录下把online_demo拷贝到thchs30下，与s5目录同级，online_demo路径下建online-data和work两个文件夹，online-data下建audio和models。audio存放要识别的wav，models建立tri1文件夹，将s5下/exp/下的tri1下的final.mdl和35.mdl拷贝到其中，同时将s5下的exp下的tri1下的graph_word里面的words.txt和HCLG.fst也拷到其中。之后对脚本中的模型和识别代码作出修改，即将ac_model_type设置为tri1。最后把要识别的音频文件复制到online-data/audio/路径下，执行run.sh 脚本即可进行识别。具体演示详见于演示视频，项目最终的目录结构如图8所示。

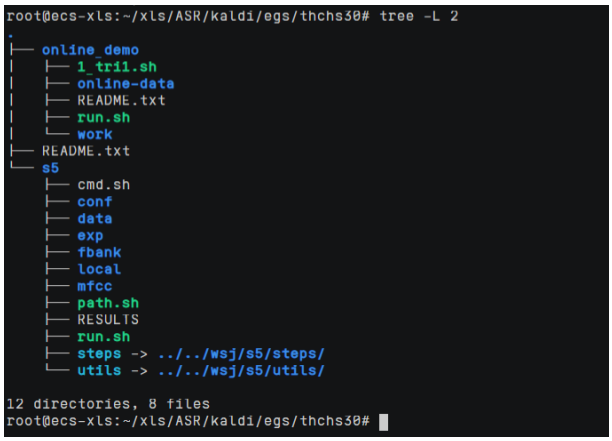

图 8: kaldi项目完整目录结构

2 MindSpore实验部分

本部分阐述基于MindSpore框架与Ascend平台进行Tacotron2语音合成模型实验。

2.1 实验环境

- 华为云AI加速型弹性云服务器，32核64G，CentOS7.6 64bit
- MobaXterm远程连接工具

- Python 3.8、MindSpore1.9.0

2.2 实验原理简介

Tacotron2[3]是由Google Brain在2017年提出来的一个End-to-End语音合成框架。模型结构如图9所示，模型从下到上可以看作由两部分组成：

- 声谱预测网络：一个Encoder-Attention-Decoder网络，用于将输入的字符序列预测为梅尔频谱的帧序列
- 声码器（vocoder）：一个WaveNet的修订版，用于将预测的梅尔频谱帧序列产生时域波形

第一部分中，Encoder输入的数据维度为$[batch_size, char_seq_length]$，使用512维的Character Embedding，把每个character映射为512维的向量，输出维度为$[batch_size, char_seq_length, 512]$。之后使用3个一维卷积，每个卷积包括512个kernel，每个kernel的大小是5*1（即每次看5个characters）。每做完一次卷积，进行一次BatchNorm、ReLU以及Dropout。输出维度为$[batch_size, char_seq_length, 512]$（为了保证每次卷积的维度不变，因此使用了padding）。最后把上面得到的输出，输送给一个单层BiLSTM，隐藏层维度是256，由于这是双向的LSTM，所以最终输出维度是$[batch_size, char_seq_length, 512]$。

Decoder则是一个自回归结构，它从编码的输入序列预测出声谱图，一次预测一帧。上一步预测出的频谱首先传入一个PreNet，它包含两层神经网络，PreNet作为一个信息瓶颈层（bottleneck），这对于Attention是必要的。PreNet的输出和Attention Context向量拼接在一起，传给一个含有1024个单元的两层LSTM。LSTM的输出再次和Attention Context向量拼接在一起，然后经过一个线性投影来预测目标频谱。最后，目标频谱帧经过一个5层卷积的PostNet（后处理网络），再将该输出和Linear Projection的输出相加（残差连接）作为最终的输出，另一边，LSTM的输出和Attention Context向量拼接在一起，投影成标量后传给sigmoid激活函数，来预测输出序列是否已完成预测。

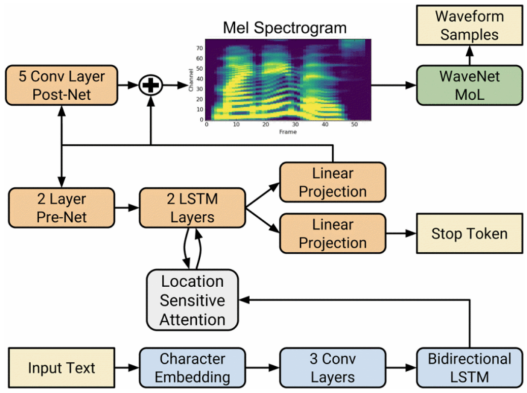

图 9: Tactron2模型结构

2.3 环境配置过程

本部分实验基于MindSpore框架与Ascend平台，借助MindSpore官方[4]给出的Tacotron2模型完成。而MindSpore框架的安装较复杂，安装失败可能性大，从而其构成了实验中不可缺少的至关重要的部分。本部分将就实验环境配置以及MindSpore框架的安装历程作出阐述。

2.3.1 ModelArts

据以往经验以及同学先行尝试的经验，

[4]https://www.mindspore.cn/install

ModelArts训练模型坑多且不稳定，极不友好，未做进一步尝试。

2.3.2 Ascend AI加速服务器Ubuntu 18.04

申请搭载Ascend 310芯片的云服务器，操作系统镜像仅有Ubuntu 18.04版本可选。配置完成服务器系统后进入MindSpore框架安装。根据官方网站给出的操作说明，需依次安装如下软件：昇腾AI处理器配套软件包、GCC、CMake、MindSpore框架，之后配置环境变量，最后检验是否安装成功。在最后一步中，需用到CMake工具加以验证，如图10。然而出现报错，称CMake版本过旧导致失败，更新CMake时，信息又显示：当前的的CMake版本为Ubuntu18.04下最新版本。如前所述，Ubuntu只有18.04可供选择，遂无奈弃之。

图 10: MindSpore检验是否成功安装命令

2.3.3 GPU加速服务器 Ubuntu 20.04

继上一种配置安装方式失败之后，返回用于kaldi实验的服务器，尝试安装GPU加速版的MindSpore框架。该服务器上原装有CUDA11.4，由MindSpore官网可知，其仅支持CUDA11.1与CUDA10.1，如图11所示。故而卸载了原装的cuda，历时数个小时重新下载并安装CUDA与cuDNN。然而进行到最终检验MindSpore是否正常安装时，显示GPU不可用，遂卸载上述已安装的软件，并通过官方给出的自动安装的脚本进行安装，脚本运行如图12所示，然而执行到检验是否成功安装时，依旧显示GPU不可用，新申请GPU版ECS依旧如此，遂又弃之。

图 11: MindSpore支持的cuda版本

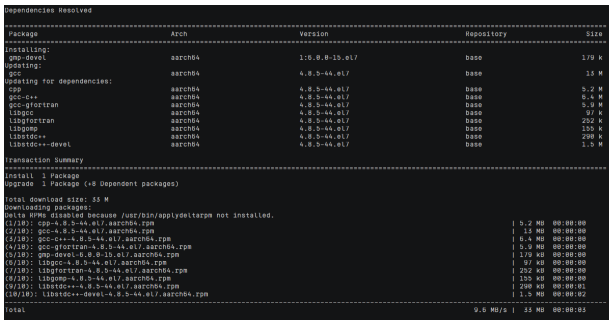

图 12: MindSpore自动安装脚本

2.3.4 Ascend AI加速服务器CentOS7.6

之后重新配置了搭载Ascend 310芯片、CenOS7.6的32核64G云服务器重新安装MindSpore框架。同前述，将下载与本地的

昇腾AI处理器配套软件包上传至服务器，驱动以及工具包的安装如图13及14所示。

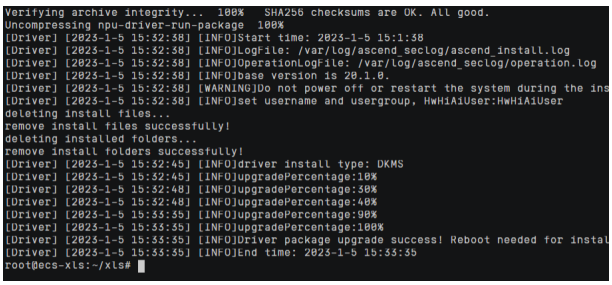

图 13: Ascend驱动安装

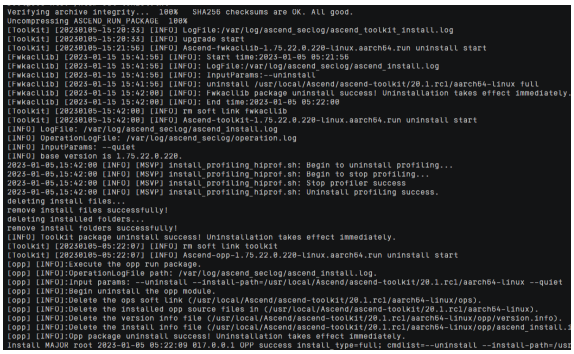

图 14: Ascend工具包安装

此后如上述执行自动安装脚本。排除若干问题后，将官网给的示例代码写入test.py，运行结果如图15所示，此时表明MindSpore框架安装成功，进入Tacotron2模型的训练阶段。

2.4 Tacotron2模型训练

在经历了宛如渡劫的框架安装之后，遵循官方[5]给出的实验步骤进行Tactron2模型的训练。数

[5]https://gitee.com/mindspore/models/tree/master/official/audio/Tacotron2

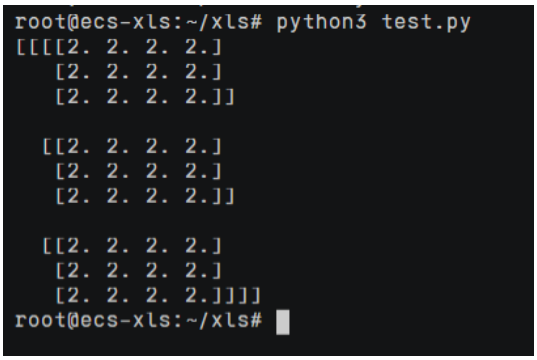

图 15: MindSpore安装成功

据集采用LJSpeech[4]，下载及上传过程同kaldi部分所述。首先运行generate_hdf5.py程序，生成对应的.hdf5文件，启动训练后数据集处理以及模型结构目录分别如图16、17所示。

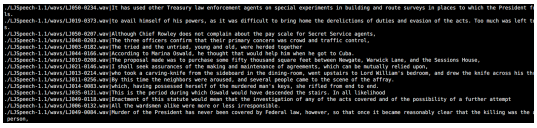

图 16: LJSpeech数据集处理中

之后由于云服务器CloudShell连接方式中途断开，且云计算资源耗尽，训练结果尚未知，将checkpoint保存目录设置与数据同级目录下，该目录的结构如图18所示。

3 总结

作为课程的结课项目，本实验存在一定的瑕疵，具体体现在MindSpore框架实验的部分，受限于计算机业务能力，在前期准备工作阶段浪费了大量宝贵的时间与计算资源。此外，除数据上传等操作外，弹性云服务器利用提供方给出的CloudShell连接方式，在系统配置以及最重

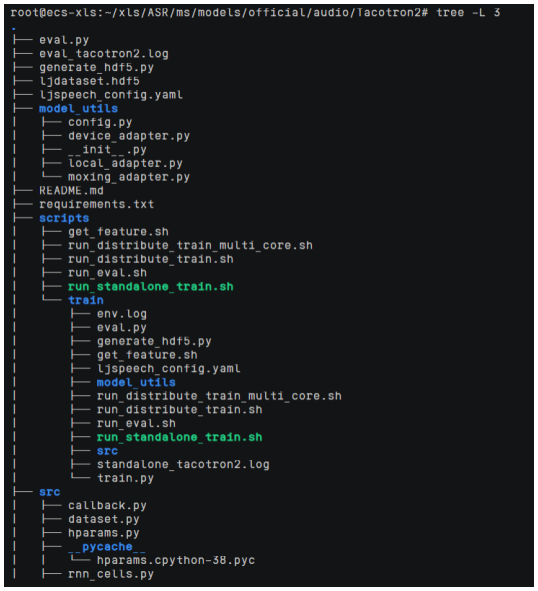

图 17: 训练后Tactron2目录结构

图 18: ECS连接自动断开

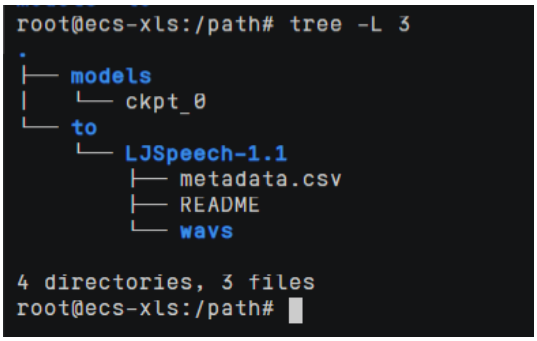

图 19: 数据集与模型目录

要的代码训练阶段时常出现连接断开的情况，致使过程丢失、重复训练等问题。但整体上看，项目完成较为顺利，实现了kaldi在THCHS30上的语音识别过程，以及启动了基于MindSpore框架的Tacotron2模型训练。此外，对于云计算资源的使用有了更深刻的体会，对个人能力的提升有很大的裨益。

真诚感谢老师的辛勤指导！同时感谢华为技术有限公司提供的计算资源支持！

参考文献

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011. 1.2

[2] Z. Z. Dong Wang, Xuewei Zhang, "Thchs-30 : A free chinese speech corpus," 2015. [Online]. Available: http://arxiv.org/abs/1512.01882 1.4

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783. 2.2

[4] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017. 2.4