

AI 深度学习语音识别及 工具使用



班 级： 16 计科 C2

专 业： 计算机科学与技术

姓 名： 宋远迪

学 号： 20161112726

指导教师： 李斌

日 期： 年 月 日

目录

1	设计总览.....	4
2	相关工作.....	5
3	参考资料.....	6
4	开发与使用环境.....	8
5	模型训练过程.....	9
5.1	数据准备	9
5.1.1	生成 manifest 文件.....	9
5.1.2	计算均值和标准差用于归一化	9
5.1.3	建立词表	9
5.1.4	数据增强	9
5.2	训练模型	11
5.3	语音到文本推断	11
5.4	模型评价	12
6	实现过程.....	13
6.1	数据处理	13
6.2	训练过程	13
6.3	语音到文本推断	15
6.4	模型评价	15
6.5	部署服务	16

6.6	用客户端与服务端进行通信	16
7	其他问题.....	17
8	设计体会.....	18
	参考文献	19

1 设计总览

本系统为基于深度学习的语音输入系统，运用 AI 模式识别（深度学习），基于 python 开发，易于使用与部署，识别率较高。无特定工作环境，但对录入环境要求一定程度的安静。

2 相关工作

本项目为上学期李斌老师带领的其中一小组的接续工作。他们小组对离线语音识别方向进行了一定程度的研究，并且实现了利用现有 API 的在线语音识别，并为其制作了界面。我们在他们的基础上对界面进行了修改，并添加了一个对现有声音文件格式化并识别的小功能。如图 2.1 所示。



图 2.2.1 改进后的界面与新功能

3 参考资料

参考自百度关于语音识别的论文^[1]，文中对现有语音识别方式加以改进，设计了新的端到端语音识别模型，结构如图 3.1。

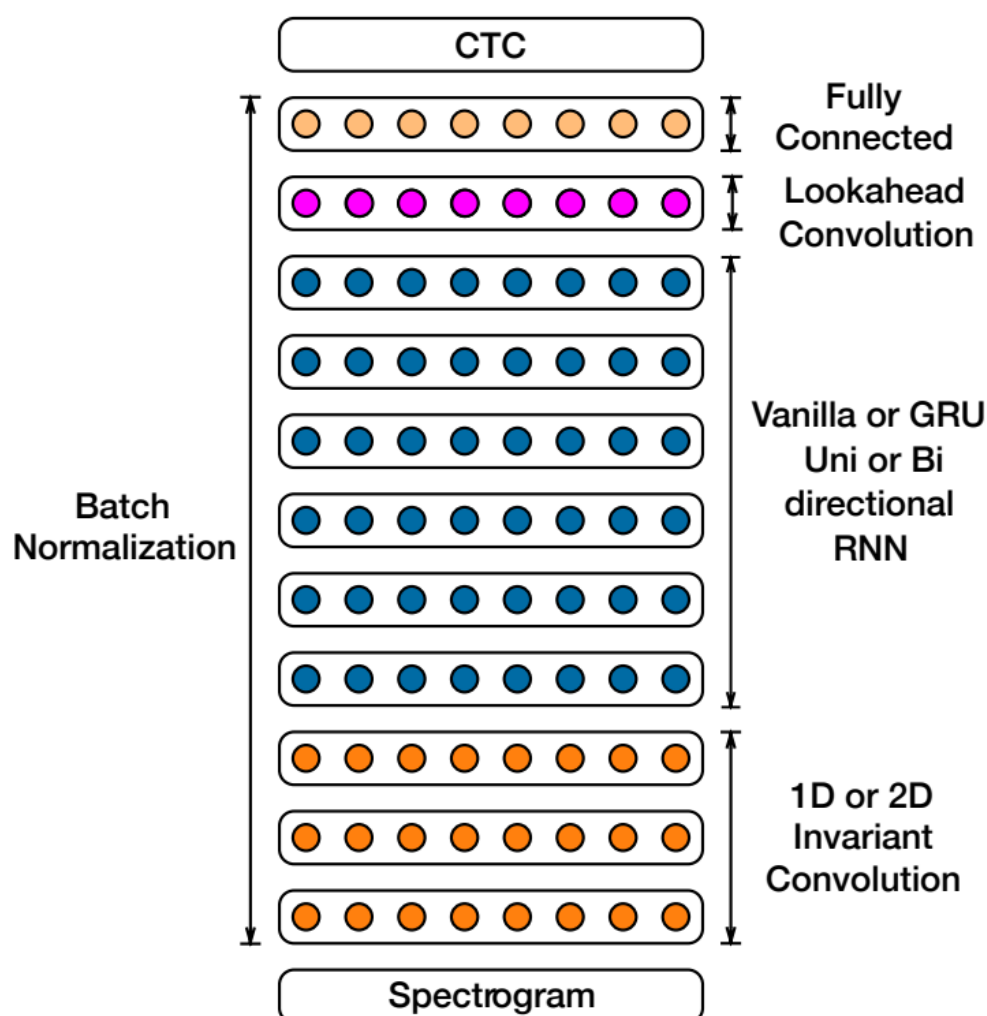


图 3.3.1 DeepSpeech2 模型架构

端到端的模型旨在一步直接实现语音的输入与解码识别，从而不需要繁杂的对齐工作与发音词典制作工作，具有了可以节省大量的前期准备时间的优势，真正的做到数据拿来就可用。

端到端的模型的另一个优点是，更换识别语言体系时可以利用

相同的框架结构直接训练。例如同样的网络结构可以训练包含 26 个字符的英文模型，也可以训练包含 3000 个常用汉字的中文模型，甚至可以将中英文的词典直接合在一起，训练一个混合模型。

其中对 CTC 模型中的转录 y 的取得给出了特别的集束搜索(beam search)方式：

$$Q(y) = \log(p_{RNN}(y|x)) + \alpha \log(p_{LM}(y)) + \beta Wc(y)$$

其中 $Wc(y)$ 是转录 y 中英文或者中文的数量， α 控制着语言模型和 CTC 网络的相对贡献。权重 β 鼓励转录出更多的字。在其中也设计了如下的新技术。

- SortaGrad

在引入 BN 层后，CTC 的训练仍会变得不稳定，在同一个 epoch 内，将每一个 batch 中最长音频长度按递增顺序进行训练，将提升稳定性。

- Frequency Convolutions

时序卷积，为了针对可变长度的语音语句的时序翻译不变性建模。

- Lookahead Convolution and Unidirectional Models

添加一个新的层，通过线性结合每个神经元的 T 时间步内的激发状态来控制所需上下文的数量，尽可能实现实时翻译。

除此之外还有许多关于对齐的优化，限于篇幅在此不再提及。

4 开发与使用环境

Python2.7<https://www.python.org/download/releases/2.7/>

Paddlepaddle 深度学习框架 <http://www.paddlepaddle.org/>

CUDA 运算平台 <https://developer.nvidia.com/cuda-toolkit>

Cudnn<https://developer.nvidia.com/cudnn>

Intel-mkl<https://software.intel.com/en-us/mkl>

可以在 Docker 上运行来避免环境部署产生问题

Docker<https://www.docker.com/>

Nvidia-Docker<https://github.com/NVIDIA/nvidia-docker>

Python 所需的相关库：(可以使用 `pip install+包名` 进行安装)

服务器端：

`scipy==0.13.1`

`resampy==0.1.5`

`SoundFile==0.9.0.post1`

`python_speech_features`

客户端：

`portaudio`

`pyaudio`

`pynput`

5 模型训练过程

5.1 数据准备

5.1.1 生成 manifest 文件

DeepSpeech2 以 Manifest 文件作为数据接口。Manifest 文件为 json 格式的数据元音频，具体格式为：

```
{"audio_filepath": "/home/work/.cache/paddle/Libri/134686/1089-134686-0001.flac", "duration": 3.275,
"text": "stuff it into you his belly counselled him"}
{"audio_filepath": "/home/work/.cache/paddle/Libri/134686/1089-134686-0007.flac", "duration": 4.275,
"text": "a cold lucid indifference reigned in his soul"}
```

接受 flac 与 wav 格式音频输入。

需要注意的是如果输入词条内容为中文，需要在接下来的每一次传递时指定解码方式为'utf-8'。

5.1.2 计算均值和标准差用于归一化

为了对音频特征进行 z-score 归一化（零均值，单位标准差），我们必须预估一些训练样本特征的均值和标准差。将会参与 BN 层的构建。运行过程中由根目录下/tools/compute_mean_std.py 实现。

5.1.3 建立词表

转换录音为索引用于训练，解码，再将一系列索引转换为文本等操作需要一个可能会出现字符集合的词表。并不对词汇进行截断。运行过程中由根目录下/tools/build_vocab.py 实现。

5.1.4 数据增强

DeepSpeech2 提供了工具来对已有语音数据进行增强。

可对数据进行如下操作：

- 音量扰动
- 速度扰动
- 移动扰动
- 在线贝叶斯归一化
- 噪声干扰（需要背景噪音的音频文件）
- 脉冲响应（需要脉冲音频文件）

设置好操作项目、顺序、强度与概率后可合成新的音频片段加入数据集。

利用增强的数据可以使训练的模型更加完备，提升识别率，但会使减缓训练速度。如果增强数据添加过多，可能会引起模型对于添加了过多噪点的数据集的过拟合。大幅度影响模型识别精度。

5.2 训练模型

训练模型的过程如上文所提到，因为是端到端的深度学习，并不必在意训练其中的各个状态。可使用 CPU 进行训练，也可以利用 CUDA 使用一个或多个 GPU 使训练更快进行。

需要注意的是，系统内存与 GPU 内存对 batch 造成了限制，由于利用了 SortaGrad 技术，将在第一个 epoch 中令一些长度极大的音频进入同一个 batch，所以需要根据实际情况调整 batch 大小。当其顺利运行第一个 epoch 的最差情况，即可认为可以顺利运行后面的迭代过程。

运行过程中由根目录下/train.py 实现。

5.3 语音到文本推断

可以用来推断，解码，以及给一些给定音频剪辑进行可视化语音到文本的结果。这有助于对 ASR 模型的性能进行直观和定性的评估。此处利用了前文提到的 CTC 波束搜索原理制作解码器，需要有预训练的模型支持这一过程。在学习工具使用的过程中，也可以下载由同一个语音库经完备训练出的模型与自己训练出的模型进行对比。

预训练的模型可以用根目录下/models/lm 中根据所需语言获取。

运行过程中由根目录下/infer.py 实现。

5.4 模型评价

可利用测试集进行测试，得到模型的误字率，可以用其对之前进行推断的解码器超参数进行调整以提升识别率。在学习工具使用的过程中，也可以下载由同一个语音库经完备训练出的模型与自己训练出的模型进行对比。

预训练的模型可以用根目录下/models/lm 中根据所需语言获取。

运行过程中由根目录下/test.py 实现。

6 实现过程

6.1 数据处理

如 5.1 部分所述，具体过程如图 6.1。

```
----- Configuration Arguments -----  
count_threshold: 0  
manifest_paths: ['data/tiny/manifest.dev-clean']  
vocab_path: data/tiny/vocab.txt  
-----  
----- Configuration Arguments -----  
manifest_path: data/tiny/manifest.tiny  
num_samples: 64  
output_path: data/tiny/mean_std.npz  
specgram_type: linear  
-----  
Tiny data preparation done.
```

图 6.1 数据处理过程

6.2 训练过程

如 5.2 部分所述对数据进行训练，如图 6.2。但需要注意其中提到的问题，在训练较大数据集时因为 SortaGrid 技术可能引起内存溢出，如图 6.3。可以用二分法寻找合适的 batch 大小加以解决，如图 6.4。

```

----- Time: 9 sec, Pass: 8, ValidationCost: 250.938652039
...
----- Time: 8 sec, Pass: 9, ValidationCost: 247.160766602
...
----- Time: 9 sec, Pass: 10, ValidationCost: 240.842609406
...
----- Time: 8 sec, Pass: 11, ValidationCost: 234.789764404
...
----- Time: 9 sec, Pass: 12, ValidationCost: 230.392539978
...
----- Time: 9 sec, Pass: 13, ValidationCost: 227.257221222
...
----- Time: 9 sec, Pass: 14, ValidationCost: 224.620121002
...
----- Time: 9 sec, Pass: 15, ValidationCost: 222.276668549
...
----- Time: 9 sec, Pass: 16, ValidationCost: 219.805831909
...
----- Time: 8 sec, Pass: 17, ValidationCost: 216.946613312
...
----- Time: 8 sec, Pass: 18, ValidationCost: 213.953937531
...
----- Time: 8 sec, Pass: 19, ValidationCost: 211.192001343
root@e1570ab24c45:/DeepSpeech/examples/tiny#

```

图 6.2 对小数据集进行训练

```

.....F0604 11:32:07.585005 09 hl_cuda_device.cc:273] Check failed: cudaSuccess == cudaStat (0 vs. 2) Cuda Error: out of memory
*** Check failure stack trace: ***
@ 0x7f9c7ac8004d google::LogMessage::Fail()
@ 0x7f9c7ac8339b google::LogMessage::SendToLog()
@ 0x7f9c7ac7fb5b google::LogMessage::Flush()
@ 0x7f9c7ac8326e google::LogMessageFatal::~LogMessageFatal()
@ 0x7f9c7ac28a9f hl_malloc_device()
@ 0x7f9c7aa7e7c7 paddle::GpuAllocator::alloc()
@ 0x7f9c7aa69458 paddle::PoolAllocator::alloc()
@ 0x7f9c7aa68e33 paddle::GpuMemoryHandle::GpuMemoryHandle()
@ 0x7f9c7aa3d5b4 paddle::GpuMatrix::resize()
@ 0x7f9c7aa59cf2 paddle::Matrix::resizeOrCreate()
@ 0x7f9c7a776548 paddle::SequenceToBatch::copyFromSeq()
@ 0x7f9c7a7aa20c paddle::RecurrentLayer::forwardBatch()
@ 0x7f9c7a7ad72c paddle::RecurrentLayer::forward()
@ 0x7f9c7a90d9ff paddle::NeuralNetwork::forward()
@ 0x7f9c7ac4fb80 GradientMachine::forwardBackward()
@ 0x7f9c7a7085f4 _wrap GradientMachine_forwardBackward
@ 0x4cb45e PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4cabd1 PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4ca099 PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4ca099 PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4ca099 PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4cabd1 PyEval_EvalFrameEx
@ 0x4c2765 PyEval_EvalCodeEx
@ 0x4c2509 PyEval_EvalCode
@ 0x4f1def (unknown)
Aborted (core dumped)
Failed in training!

```

图 6.3 可能引起的内存溢出问题

```

Pass: 0, Batch: 6700, TrainCost: 23.074532
.....
Pass: 0, Batch: 6800, TrainCost: 22.953000
.....
Pass: 0, Batch: 6900, TrainCost: 23.062089
.....
Pass: 0, Batch: 7000, TrainCost: 22.582598
.....
Pass: 0, Batch: 7100, TrainCost: 22.915757
.....
Pass: 0, Batch: 7200, TrainCost: 23.568645
.....
Pass: 0, Batch: 7300, TrainCost: 22.878085
.....
Pass: 0, Batch: 7400, TrainCost: 23.901534
.....
Pass: 0, Batch: 7500, TrainCost: 25.737494
.....
----- Time: 6229 sec, Pass: 0, ValidationCost: 16.3942381382
.....bat.....

```

图 6.4 缩小 batch size 后成功训练

6.3 语音到文本推断

见 5.3，过程如图 6.5。

```
Current error rate [wer] = 1.000000

Target Transcription: on the general principles of art mister quilter writes with equal lucidity
Output Transcription: eeeeeeee
Current error rate [wer] = 1.000000

Target Transcription: painting he tells us is of a different quality to mathematics and finish in art is adding more fact
Output Transcription: telesales artist
Current error rate [wer] = 1.000000

Target Transcription: as for etchings they are of two kinds british and foreign
Output Transcription: instead
Current error rate [wer] = 1.000000

Target Transcription: he laments most bitterly the divorce that has been made between decorative art and what we usually call pictures makes the customary appeal to the last judgment and reminds us that in the great days of art michael angel was the furnishing upholsterer
Output Transcription: estatetransportationopinioncalendararchives
Current error rate [wer] = 1.000000
[INFO 2019-06-03 04:22:07,660 infer.py:124] finish inference
```

图 6.5 推断过程

6.4 模型评价

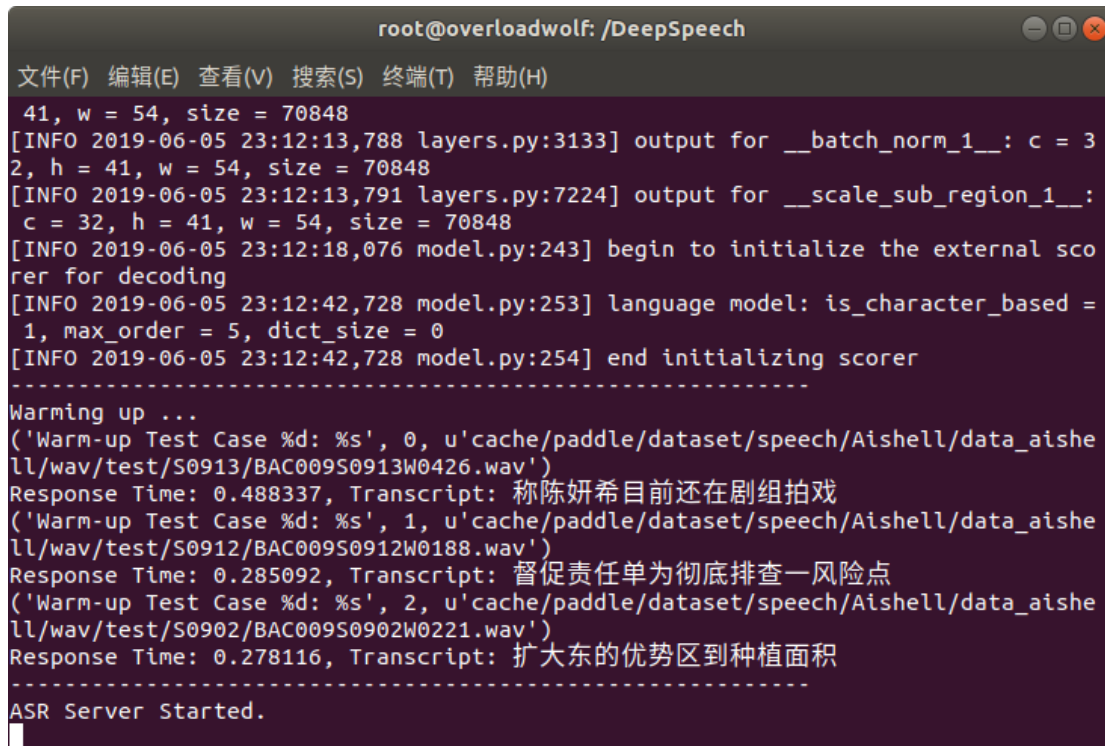
见 5.4，过程如图 6.6。

```
c = 32, h = 81, w = 54, size = 139968
[INFO 2019-06-03 04:39:41,474 layers.py:2606] output for __conv_1__: c = 32, h = 41, w = 54, size = 70848
[INFO 2019-06-03 04:39:41,475 layers.py:3133] output for __batch_norm_1__: c = 32, h = 41, w = 54, size = 70848
[INFO 2019-06-03 04:39:41,475 layers.py:7224] output for __scale_sub_region_1__: c = 32, h = 41, w = 54, size = 70848
[INFO 2019-06-03 04:39:44,745 model.py:243] begin to initialize the external scorer for decoding
[INFO 2019-06-03 04:40:06,192 model.py:253] language model: is_character_based = 0, max_order = 5, dict_size = 400000
[INFO 2019-06-03 04:40:06,370 model.py:254] end initializing scorer
[INFO 2019-06-03 04:40:06,370 test.py:98] start evaluation ...
Error rate [wer] (8/?) = 0.990000
Error rate [wer] (16/?) = 0.994460
Error rate [wer] (24/?) = 0.995968
Error rate [wer] (32/?) = 0.996479
Error rate [wer] (40/?) = 0.997050
Error rate [wer] (48/?) = 0.996241
Error rate [wer] (56/?) = 0.994536
Error rate [wer] (64/?) = 0.994100
Final error rate [wer] (64/64) = 0.994100
[INFO 2019-06-03 04:40:14,992 test.py:130] finish evaluation
root@893657d23dee:/DeepSpeech/examples/tiny#
```

图 6.6 模型评价过程

6.5 部署服务

利用 docker 端口映射并在端口上开启服务，如图 6.7。

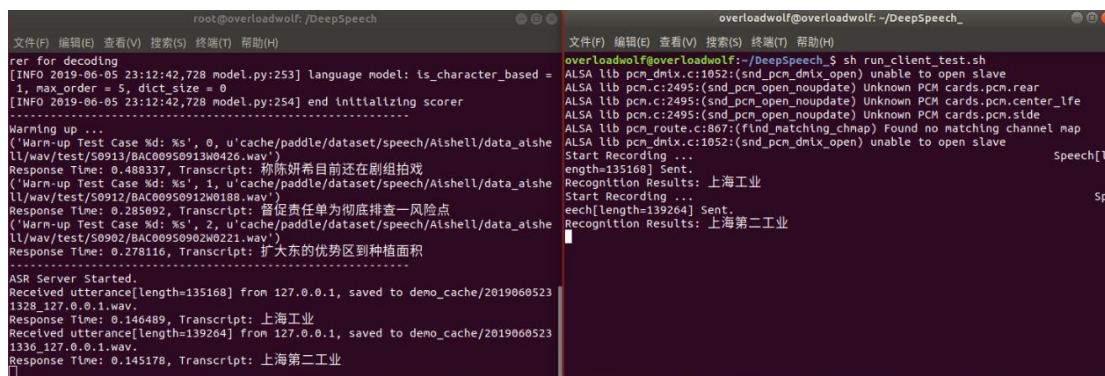


```
root@overloadwolf: /DeepSpeech
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
41, w = 54, size = 70848
[INFO 2019-06-05 23:12:13,788 layers.py:3133] output for __batch_norm_1__: c = 3
2, h = 41, w = 54, size = 70848
[INFO 2019-06-05 23:12:13,791 layers.py:7224] output for __scale_sub_region_1__:
c = 32, h = 41, w = 54, size = 70848
[INFO 2019-06-05 23:12:18,076 model.py:243] begin to initialize the external sco
rer for decoding
[INFO 2019-06-05 23:12:42,728 model.py:253] language model: is_character_based =
1, max_order = 5, dict_size = 0
[INFO 2019-06-05 23:12:42,728 model.py:254] end initializing scorer
-----
Warming up ...
('Warm-up Test Case %d: %s', 0, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0913/BAC009S0913W0426.wav')
Response Time: 0.488337, Transcript: 称陈妍希目前还在剧组拍戏
('Warm-up Test Case %d: %s', 1, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0912/BAC009S0912W0188.wav')
Response Time: 0.285092, Transcript: 督促责任单为彻底排查一风险点
('Warm-up Test Case %d: %s', 2, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0902/BAC009S0902W0221.wav')
Response Time: 0.278116, Transcript: 扩大东的优势区到种植面积
-----
ASR Server Started.
```

图 6.7 利用 docker 开启服务

6.6 用客户端与服务端进行通信

在此时通过本机端口实现的，如图 6.8。以此原理将服务放在公网上使用。



```
root@overloadwolf: /DeepSpeech
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
er for decoding
[INFO 2019-06-05 23:12:42,728 model.py:253] language model: is_character_based =
1, max_order = 5, dict_size = 0
[INFO 2019-06-05 23:12:42,728 model.py:254] end initializing scorer
-----
Warming up ...
('Warm-up Test Case %d: %s', 0, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0913/BAC009S0913W0426.wav')
Response Time: 0.488337, Transcript: 称陈妍希目前还在剧组拍戏
('Warm-up Test Case %d: %s', 1, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0912/BAC009S0912W0188.wav')
Response Time: 0.285092, Transcript: 督促责任单为彻底排查一风险点
('Warm-up Test Case %d: %s', 2, u'cache/paddle/dataset/speech/Aishell/data_aish
ell/wav/test/S0902/BAC009S0902W0221.wav')
Response Time: 0.278116, Transcript: 扩大东的优势区到种植面积
-----
ASR Server Started.
Received utterance[length=135168] from 127.0.0.1, saved to demo_cache/2019060523
1328_127.0.0.1.wav.
Response Time: 0.146489, Transcript: 上海工业
Received utterance[length=139264] from 127.0.0.1, saved to demo_cache/2019060523
1336_127.0.0.1.wav.
Response Time: 0.145178, Transcript: 上海第二工业

overloadwolf@overloadwolf: ~/DeepSpeech_
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
overloadwolf$ sh run_client_test.sh
ALSA lib pcm_dmix.c:1052:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_dmix.c:1052:(snd_pcm_dmix_open) unable to open slave
Start Recording ...
length=135168] Sent.
Recognition Results: 上海工业
Start Recording ...
length=139264] Sent.
Recognition Results: 上海第二工业
```

图 6.8 客户端和服务端通信

7 其他问题

在训练较大数据集时（aishell, 15.6G, 共 178 小时语料），按照当前所使用设备配置（Nvidia1060+机械硬盘）进行训练，进行一次迭代需要的时间在 1.5 小时到 2.5 小时之间。参考常用的配置，这个规模的数据集大概需要进行 50 次迭代才可使用，但按这个速度大概需要不间断进行一周左右。若有更好的设备或资金更多，就可以选择使用租用 gpu 计算服务器来进行这个过程，会加速很多。

8 设计体会

本项目真正需要代码实现的内容并不多，但是比之前纯用代码实现的项目困难了许多。主要问题还是在于开发经验的缺失。最初想要利用 `winbash` 进行开发，过程中发现 `winbash` 并不能调用任何硬件资源，纯 CPU 训练又太过缓慢，遂转至双系统进行开发。因为不熟悉 `docker` 的使用，最初决定手动部署环境，但确实因为库依赖，驱动依赖等问题失败了无数次。之后重装系统（因为装 `cuda` 的过程中搞乱了系统的显卡驱动版本），选择使用 `Nvidia docker` 进行环境配置顺利进行，训练过程中又因为作者并没有及时更新，语音库云存储地址变化耗费许多时间。不过手动解决上述问题之后训练的过程还是成功了。但在换用更大中文语音库的时候遇到了配置跟不上的问题，虽然最后找到了一个别人训练好的模型拿来演示，但这也是这次项目的不足之处了。有一定体量的深度学习项目确实不适合用个人电脑进行训练，需要更加专业的配置或者特定用途的服务器，下次进行这种项目的话，在开始之前就应该对此有所准备，并申请资金或找寻合适的设备，避免再度出现这样的悲剧。但总的来说这个项目还是挺成功的（虽然识别率依然堪忧），本次项目中熟悉了 `docker` 以及部分深度学习平台的使用方法，了解了服务部署和服务端客户端通信的方法，相信对以后的工作会有益处。

参考文献

- [1] Amodei D , Anubhai R , Battenberg E , et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin[J]. Computer Science, 2015. <http://proceedings.mlr.press/v48/amodei16.pdf>