端到端在线声学模型finetune

## 目录结构

|-- eteh-v2-release-online\_export\_202109

| |-- data\_frontend\_v4\_hires\_chinese\_u8\_mfcchirs40\_release 数据处理步骤一，主要完成分词、提特征等

| | |-- bin

| | |-- libs

| | |-- log\_20190619

| | |-- readme.txt

| | |-- run.sh

| | `-- scripts

| |-- env\_alignments 数据处理步骤二，进行align对齐

| | |-- cmd.sh

| | |-- lang

| | |-- model align的基线模型

| | |-- path.sh

| | |-- run\_alignment.sh

| | |-- steps

| | `-- utils

| |-- env\_datajson 数据处理步骤三，准备训练需要的json文件

| | |-- cmd.sh

| | |-- lang 语言层信息

| | |-- path.sh

| | |-- run\_datajson.sh

| | `-- utils

| |-- env\_train 训练环境

| | |-- bin

| | |-- cmd.sh

| | |-- conf 训练配置文件和json数据配置文件

| | |-- exp 基线模型

| | |-- gen\_data\_yaml.pl

| | |-- path.sh

| | `-- run.online\_finetune.sh 训练可执行文件

| |-- eteh 训练依赖的环境

| | |-- CONTRIBUTING.md

| | |-- LICENSE

| | |-- README.md

| | |-- bin

| | |-- eteh

| | `-- example

| |-- model\_convert 模型转换工具

| | |-- README

| | |-- average\_checkpoints.py

| | |-- convert\_model\_mta.py

| | |-- model.eteh.config

| | `-- run.sh

| |-- path.sh

| `-- run\_prepare\_trainData.sh 数据预处理总控脚本

|-- env 依赖环境，docker镜像

`-- gpu-cuda10.1-cudnn7-u18-local-espnet-torch1.7-release.tar

## 训练环境-docker容器

依赖环境位于env目录下

### NVIDIA 驱动要求

要求加载镜像的物理机安装支持cuda10 及以上的NVIDIA驱动，驱动版本要在410.129版本及以上

### Docker镜像加载

docker load -i gpu-cuda10.1-cudnn7-u18-local-espnet-torch1.7-release.tar

### 查看image ID

docker images

### 启动镜像

NV\_GPU='0,1,2,3' nvidia-docker run --ipc=host --name e2e-finetune-env -it -d -v /data/user:/data/user --entrypoint="/bin/bash" 422f8d9340c5

-v 参数指定工作目录，第一个参数为物理机路径，第二个参数为容器工作路径

422f8d9340c5 image ID

--name 执定容器的名字

### 进入容器

docker exec -it e2e-finetune /bin/bash

### 关闭容器及删除

容器启动后会占用部分磁盘空间，为了保证磁盘空间的高效利用，容器使用完成后要及时关闭和删除，命令如下：

docker stop containerID

docker rm containerID

如果后续镜像也不会再使用，也需要进行删除，命令如下：

docker rmi imageID

## 训练

进入到该目录下eteh-v2-release-online\_export\_202109，进行数据预处理及模型训练。

请预先参照Q&A重新编译kaldi环境，防止出现数据预处理异常，kaldi编译需要一些时间。

### 数据预处理

#### stage=0开始处理

进入到该目录下deliverables\_release\_online\_finetuneJH\_202102，修改数据预处理总控脚本run\_prepare\_trainData.sh，可修改参数如下：

**注意：以下路径请使用绝对路径**

wav\_dir=/home/user/e2e\_finetune/test/input 语音路径

output\_dir=/home/user/e2e\_finetune/test/output 中间结果及最终结果输出路径

num\_utt\_percent=50 从训练集中挑选百分之多少作为验证集，取倒数，这里是1/50=2%

输出的可用于训练的json文件路径为：

${output\_dir}/preparejson/dump/train\_dev\_json

#### stage=3开始处理

由于align时间较长，有时候我们需要利用之前align的结果，可将stage设置为3进行处理，需要注意两个地方：

1. run\_prepare\_trainData.sh脚本中:

ali\_ouput\_dir 设置为已有align文件的目录

ali\_nj 设置为已有align文件的个数

一定要注意ali\_nj文件个数的设置，以免部分align数据利用不全

1. 资源文件要一致

align的时候用的资源文件要同env\_datajson/lang文件下的资源文件要一致，以免造成生成json文件脚本异常，可以从原来align环境中找到对应的资源文件，拷贝过来，具体包括以下资源文件：

align\_lexicon.int、phones.txt、words.txt

### 模型训练

进入deliverables\_release\_online\_finetuneJH\_202102/env\_train目录，修改run.online\_finetune.sh脚本，参数说明如下：

#使用该命令nvidia-smi查看可用卡数，进行配置，例如：0,1,2,3

export CUDA\_VISIBLE\_DEVICES=3

#用几块进行训练，与上面配置的CUDA\_VISIBLE\_DEVICES个数相同

ngpu=1

#迭代的epoch个数为40-30=10次

epochs=70 # epochs for online model

#基线模型，无需修改

checkpoint =baseModel/checkpoint.59

#模型配置，无需修改

train\_config=conf/ce\_espnet\_online\_transformer\_nospecaug\_md.yaml

#json训练数据配置

data\_conf=conf/data.yaml

#数据预处理最后输出的json路径，根据数据预处理实际输出进行配置

json\_train\_dir =/home/user/e2e\_finetune/test/output/preparejson/dump/train\_dev\_json

#训练数据json个数，与数据预处理过程最后生成的train的json个数相同

json\_nj=8

#训练模型输出路径，可自定义修改

exp\_dir=/home/user/e2e\_finetune/test/output/exp

主要修改参数为json\_train\_dir、exp\_dir，可以适当调节epochs的个数，修改完成后执行该脚本进行训练。

模型输出路径：${exp\_dir}

### 模型转换

#### bin模型（常用）：

进入eteh-v2-release-online\_export\_202109/model\_convert目录，执行：

./run.sh 模型输出路径 model.epochs.bin 5

参数1为模型输出路径，保存要做平均的model.\*.tar的路径

参数2为二进制模型，可用于实际系统进行测试

参数3 模型平均的个数，一般取5个或10个，会对model.\*.tar进行日期排序，取最后num个，一般情况下日期排序就是模型编号排序，如果model.\*.tar是拷贝过去的，顺序会被打乱，为了确保无误，参数1下面只保留确定要平均的模型

#### Onnx模型：

在线的onnx模型转换需要先进行bin模型转换，然后利用平均后的模型进行onnx模型转换，onnx在线识别引擎需要同时使用bin模型和onnx模型。

进入eteh-v2-release-online\_export\_202109/model\_convert\_onnx目录，将model.last5.avg.best（bin模型转换时生成的中间做参数平均的模型）拷贝到zh\_online目录，然后执行命令：

python torch2onnx\_online\_onnxruntime.py

会在zh\_online目录生成converted.model.last5.avg.best目录，下面有生成好的onnx模型。

## Q&A

### 数据预处理异常，Kaldi命令提示找不到相应的库文件

问题描述：align.1.log

compile-train-graphs: error while loading shared libraries: libkaldi-decoder.so: cannot open shared object file: No such file or directory

nnet3-align-compiled: error while loading shared libraries: libkaldi-nnet3.so: cannot open shared object file: No such file or directory

解决办法：进入到容器后，进入到/kaldi-master/src目录，重新编译kaldi，参照INSTALL说明进行编译即可。

make clean

./configure --shared

make depend -j 8

make -j 8