端到端声学模型离线finetune

注：如果离线finetune之前做过在线finetune，且使用的同一批训练数据，则可以利用在线生成好的json训练数据，直接进行训练。

## 目录结构

├── eteh-v2-release-offline\_export\_202109

│   ├── data\_frontend\_v4\_hires\_chinese\_u8\_mfcchirs40\_release数据处理第一步，主要完成数据分词、提特征等

│   │   ├── bin

│   │   ├── libs

│   │   ├── log\_20190619

│   │   ├── readme.txt

│   │   ├── run.sh

│   │   └── scripts

│   ├── env\_train 训练环境目录

│   │   ├── baseModel 基线模型

│   │   ├── bin

│   │   ├── cmd.sh

│   │   ├── conf

│   │   ├── gen\_data\_yaml.pl

│   │   ├── path.sh

│   │   └── run.offline\_finetune.sh 训练启动脚本

│   ├── eteh 训练依赖环境

│   ├── model\_convert 二进制模型转换

│   │   ├── average\_checkpoints.py

│   │   ├── average\_checkpoints.py\_bak

│   │   ├── config\_chn.txt

│   │   ├── convert\_model.py

│   │   └── model\_convert.sh

│   ├── model\_convert\_onnx onnx模型转换

│   │   ├── config.py

│   │   ├── espnet

│   │   ├── \_\_pycache\_\_

│   │   ├── torch2onnx\_onnxruntime.py

│   │   └── ZhModel

│   ├── prepare-json 数据预处理第二步，训练数据json文件准备

│   │   ├── prepjson

│   │   └── prepjson\_lang

│   ├── run\_prepare\_trainData.sh 数据预处理脚本

│   └── utils 工具目录

└── env

   └── gpu-cuda10.1-cudnn7-u18-local-espnet-torch1.7-release.tar 镜像文件

## 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-offline-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-offline\_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eteh-v2-release-offline\_export\_202109，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/split\_train\_dev

### 模型训练

进入eteh-v2-release-offline\_export\_202109/env\_train目录，修改run.offline.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

num\_gpu=1

#迭代的epoch个数为40-30=10次

epochs=40 # epochs for online model

#基线模型，无需修改

checkpoint= baseModel/checkpoint.29

#模型配置，无需修改

train\_config=conf/ce\_espnet\_baseline\_fintune.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=/data/user/AM/dataEnv/test/output/offline/split\_train\_dev

#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=/data/user/AM/dataEnv/test/output/offline/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}目录下

学习率可调：

conf/ce\_espnet\_baseline\_fintune.yaml 文件里factor参数默认为1，可调节至0.1、0.01对模型进行调优。

### 模型转换

#### bin模型（常用）：

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

./run.sh 模型输出路径 37 40

参数1为模型输出路径，保存要做平均的checkpoint.\*的路径

参数2为模型平均的开始epoch数

参数3 为模型平均的结束的epoch数

上述命令对37、38、39epoch模型做平均。

## 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

## 模型训练yaml文件说明

ce\_espnet\_baseline\_fintune.yaml

set\_config:

data\_type: json

load: False

jconfig:

batch\_size: 16 #训练数据batch大小，可根据显存进行调节，16-32

max\_length\_in: 512

max\_length\_out: 150

num\_batches: 0

min\_batch\_size: 1

shortest\_first: True

batch\_sort\_key: "input"

swap\_io: False

count: "seq"

batch\_bins: 100000

batch\_frames\_in: 0

batch\_frames\_out: 0

batch\_frames\_inout: 0

clean\_data: True #是否对数据进行筛选

down\_sample: 2

ilen\_max: 2000 #最大输入utt的帧数

ilen\_min: 17 #最小输入utt的帧数

olen\_max: 100 #最大输出utt的字符个数

trans\_config:

# these three processes are a.k.a. SpecAugument

- type: "time\_warp"

max\_time\_warp: 5

inplace: true

mode: "PIL"

- type: "freq\_mask"

F: 30

n\_mask: 2

inplace: true

replace\_with\_zero: false

- type: "time\_mask"

T: 40

n\_mask: 2

inplace: true

replace\_with\_zero: false

opti\_config:

name: 'eteh.models.pytorch\_backend.optimizer.optimizer:Noam'

factor: 1 #调节学习率大小，缩放因子

warm\_step: 25000

model\_size: 256

criterion\_config:

name: 'eteh.models.pytorch\_backend.criterion.loss:E2E\_Loss'

size: 5720

padding\_idx: -1

smoothing: 0.1

rate: 0.3

model\_config: #模型配置参数

name: 'eteh.models.pytorch\_backend.model.e2e:E2E\_Transformer\_CTC'

idim: 40 #特征输入维度

odim: 5720 #label输出个数

encoder\_attention\_dim: 320 #encoder注意力机制维度

encoder\_attention\_heads: 8 #注意力头个数

encoder\_linear\_units: 2048 #encoder的线性单元个数

encoder\_num\_blocks: 14 #encoder的block个数

encoder\_input\_layer: conv2d #encoder的输入层，二维卷积

encoder\_dropout\_rate: 0.1

encoder\_attention\_dropout\_rate: 0

decoder\_attention\_dim: 320

decoder\_attention\_heads: 4

decoder\_linear\_units: 2048

decoder\_input\_layer: embed

decoder\_num\_block: 7

decoder\_dropout\_rate: 0.1

decoder\_src\_attention\_dropout\_rate: 0

decoder\_self\_attention\_dropout\_rate: 0

ctc\_dropout: 0.1

train\_config: #训练参数

char\_num: 5720 #输出label个数

accum\_grad: 2 #must be 1 when amp is used #梯度累积，2\*batch\_size更新梯度

amp: False

valid\_config: #valid 配置

data\_type: json

jconfig:

batch\_size: 16

max\_length\_in: 512

max\_length\_out: 150

num\_batches: 0

min\_batch\_size: 1

shortest\_first: True

batch\_sort\_key: "input"

swap\_io: False

count: "seq"

batch\_bins: 100000

batch\_frames\_in: 0

batch\_frames\_out: 0

batch\_frames\_inout: 0

clean\_data: True

down\_sample: 2

ilen\_max: 2000

ilen\_min: 17

olen\_max: 100

decode\_config: #解码参数

beam: 10

ctc\_beam: 15

lm\_rate: 0

ctc\_weight: 0.3

char\_num: 5720