离线e2e模型训练开发测试环境

# 硬件要求

服务器建议使用V100（P100也可以），内存128G，存储1T。

# 软件要求

操作系统建议安装CentOS 7.x，gcc/g++ 4.8及以上；

安装docker/nvidia-docker，版本建议20.10.1；

Nvidia驱动在410以上。

# 测试运行

模型训练所依赖的环境统一使用docker镜像部署，训练代码采用挂载方式，这里不再详述，下面将对模型训练环境使用说明进行介绍。

离线模型训练环境的目录结构如下所示：

eteh-v2-release-JXJK2021\_orig\_v2\_release

|--bin 模型训练入口函数

|--eteh 模型训练核心代码模块

|--example 示例

|--hkust\_egs

|--baseModel 基线模型

|--conf 相关配置

|--output 存放输入输出数据

|--run\_train.sh 训练脚本

|--run\_predictor.sh 预测脚本

|--run\_evaluate.sh 评估脚本

|--utils 工具类

## 3.2 模型训练

模型训练脚本位于example/hkust\_egs/run\_train.sh

参数说明：

exp\_dir=output/exp 模型输出路径

train\_config=conf/ce\_espnet\_baseline\_fintune.yaml 模型训练配置

data\_conf=conf/data.yaml 训练数据yaml文件

checkpoint=baseModel/checkpoint.29 基线模型

运行方法：

./run\_train.sh

## 3.2 模型预测

模型预测脚本位于example/hkust\_egs/run\_predictor.sh

参数说明：

exp\_dir=output/exp 模型输出路径

train\_config=conf/ce\_espnet\_baseline\_fintune.yaml 模型训练配置

data\_conf=conf/data.yaml 训练数据yaml文件

char\_list=baseModel/vocab.kefu\_cts.txt 字典文件

checkpoint=baseModel/checkpoint.29 基线模型

运行方法：

./run\_ predictor.sh

## 3.3 模型评估

模型评估脚本位于example/hkust\_egs/run\_evaluate.sh

参数说明：

rec=your result file 结果文件

ref=your ref file 答案文件

运行方法：

./run\_evaluate.sh

执行完成后，会在结果所在路径生成.sys文件，为识别率测试结果，如下：

SPKR | # Snt # Chr | Corr Sub Del Ins Err S.Err

Sum/Avg | 213 6108 | **86.5** 10.5 3.0 2.6 **16.0** 69.0

由上述结果可知，字正确率为86.5%，字错误率为16%。

# 配置参数说明

## 4.1 训练参数配置

##训练数据配置

set\_config:

data\_type: json 训练数据类型

load: False

jconfig:

batch\_size: 16 batch\_size大小

max\_length\_in: 512

max\_length\_out: 150

num\_batches: 0

min\_batch\_size: 1 最小的batch\_size大小

shortest\_first: True 按最短优先对数据进行排序

batch\_sort\_key: "input" 按input数据长度对数据进行排序

swap\_io: False

count: "seq" 采用seq方式进行batch打包，上方参数起作用，下面参数只为“bin”时起作用

batch\_bins: 100000

batch\_frames\_in: 0

batch\_frames\_out: 0

batch\_frames\_inout: 0

clean\_data: True 是否对数据进行清洗

down\_sample: 2

ilen\_max: 2000 最大输入长度，单位为帧

ilen\_min: 17 最小输入长度，单位为帧

olen\_max: 100 最大输出长度

trans\_config: 数据增强的相关配置，暂未启用

# 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

##优化器相关配置

opti\_config:

name: 'eteh.models.pytorch\_backend.optimizer.optimizer:Noam' 优化器类型

factor: 1 学习率影响因子，调到，学习率会放大

warm\_step: 25000

model\_size: 256

##损失函数相关配置

criterion\_config:

name: 'eteh.models.pytorch\_backend.criterion.loss:E2E\_Loss' 定义要用的损失函数

size: 5720 字典大小

padding\_idx: -1 padding值

smoothing: 0.1 标签平滑参数

rate: 0.3 ctc损失函数占比

##模型配置相关

model\_config:

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

idim: 40 输入数据特征维度

odim: 5720 输出维度，字典大小

encoder\_attention\_dim: 320 编码器注意力维度

encoder\_attention\_heads: 8 编码器注意力头个数

encoder\_linear\_units: 2048 编码器线性层维度

encoder\_num\_blocks: 14 编码器block个数

encoder\_input\_layer: conv2d 编码器的输入

encoder\_dropout\_rate: 0.1 编码器的dropout概率

encoder\_attention\_dropout\_rate: 0 编码器attention的dropout概率

decoder\_attention\_dim: 320 解码器注意力维度

decoder\_attention\_heads: 4 解码器注意力头个数

decoder\_linear\_units: 2048 解码器线性层维度

decoder\_input\_layer: embed 解码器输入

decoder\_num\_block: 7 解码器block个数

decoder\_dropout\_rate: 0.1 解码器dropout概率

decoder\_src\_attention\_dropout\_rate: 0

decoder\_self\_attention\_dropout\_rate: 0

ctc\_dropout: 0.1 ctc dropout概率

##训练参数配置

train\_config:

char\_num: 5720 字典大小

accum\_grad: 2 #must be 1 when amp is used 梯度累积

amp: False

##验证集数据配置，这里痛set\_config，这里不再详述

valid\_config:

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

##解码配置

decode\_config:

beam: 10 总的解码beam值

ctc\_beam: 15 ctc解码beam值

lm\_rate: 0 语言模型概率占比

ctc\_weight: 0.3 ctc概率占比

char\_num: 5720 字典大小

## 4.2 训练数据配置

##训练数据

clean\_source:

1:

type: HKUST\_TRAIN

path: /data/test/output/preparejson/split\_train\_dev/data.1.json

2:

type: HKUST\_TRAIN

path: /data/test/output/preparejson/split\_train\_dev/data.2.json

3:

type: HKUST\_TRAIN

path: /data/test/output/preparejson/split\_train\_dev/data.3.json

##验证数据

valid\_source:

1:

type: HKUST\_DEV

path: /data/test/output/preparejson/split\_train\_dev/data.dev.json