AI Academy 음성/언어 심화과정
- conformer 기반 E2E 음성인식기 실습

서강대학교 청각지능 연구실
김지환

서강대학교
SOGANG UNIVERSITY

Table of contents

개요

영어 낭독체 기반 end-to-end 음성인식기 실습

- 100시간의 무료 낭독체 데이터셋을 통한 end-to-end 기반의 영어 음성인식기 개발
- Convolution-augmented transformer (Conformer) 기반 character 단위 모델 실습
- Tensorflow 2.X 환경에서 Google Colab을 통해 실습 진행

Reference

- Tensorflow 2.X (https://www.tensorflow.org/)
- LibriSpeech (http://openslr.org/12/)

개요

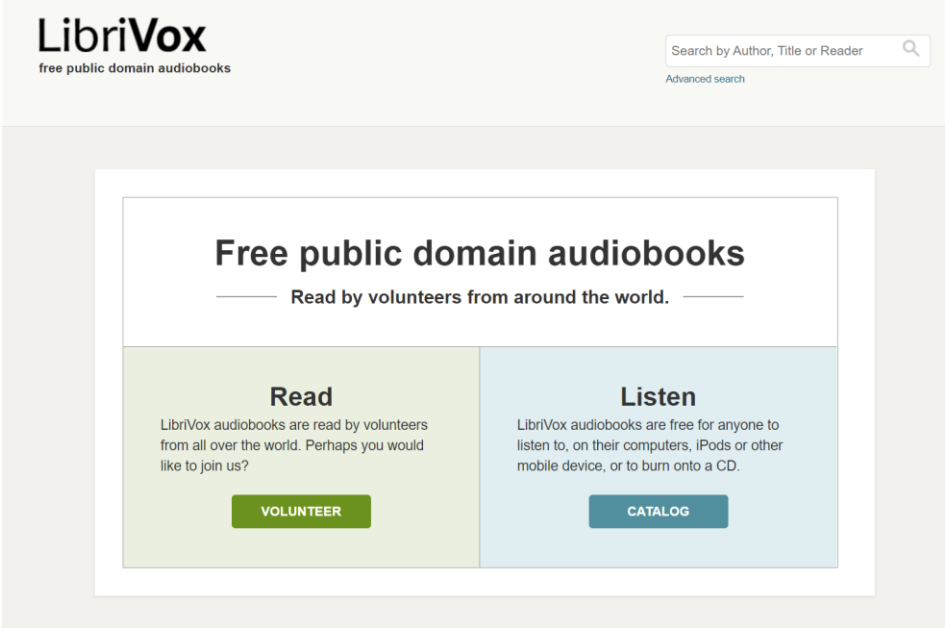
TensorFlow

- Google Brain에서 개발한 Machine learning을 위한 end-to-end 오픈소스 플랫폼
- 가장 널리 사용되는 deep learning toolkit 중 하나임
- 다양한 딥 러닝 모델(FFNN, RNN, CNN 등)에 대한 라이브러리 제공
- 다양한 도메인(vision, NLP, OCR, 음성 등)에 대한 전처리 및 후처리 라이브러리 및 데이터 제공

개요

Dataset 소개 - LibriSpeech

- 사용자 참여형 오디오북 프로젝트인 LibriVox Project의 결과물 (https://librivox.org/)
- 16kHz로 샘플링된 약 1,000시간 분량의 녹음된 오디오북 데이터
- 음성인식 연구에서 가장 널리 사용되는 대규모 영어 음성 데이터
- Open Speech and Language Resources (openSLR, https://openslr.org/resource/12)에서 무료로 다운로드 받아 사용할 수 있음
- 본 실습에서는 자원이 제한된 Google colab에서 동작 가능하도록 전체 데이터의 일부인 100시간 분량만 사용함



실습 환경 소개 – Google Colaboratory

Google colaboratory(colab)

- Jupyter notebook을 기반으로 웹 브라우저 상에서 python 프로그래밍을 수행할 수 있는 클라우드 기반 서비스
- 개인이 서버 등의 하드웨어나 Pycharm 등의 통합 개발 환경(IDE) 소프트웨어를 별도 세팅할 필요 없이 웹 기반으로 수행 가능함

특징

- 하나의 환경에 대해 복수의 인원이 동시 작업 가능
- GPU, TPU 등 고성능 연산 장치의 사용이 가능함

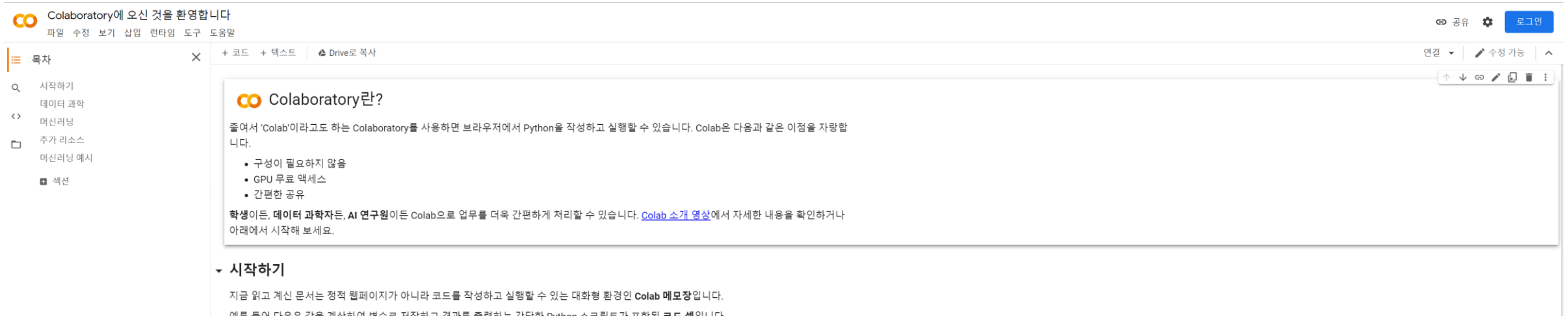
접근 방법

- https://colab.research.google.com 을 통해 사용 가능 (Google 계정 필요)

실습 환경 소개 – Google Colaboratory

초기 화면 및 로그인

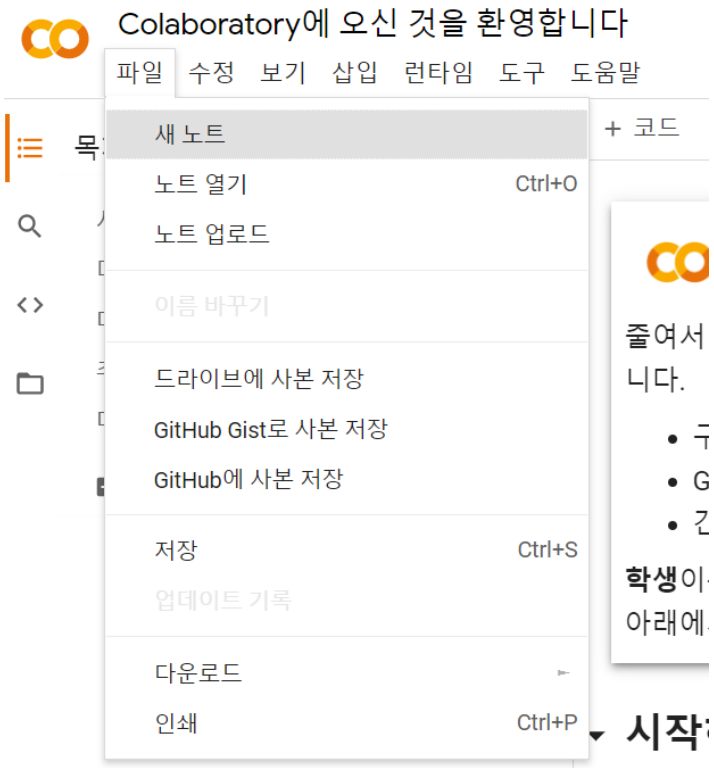
- https://colab.research.google.com 에 접속

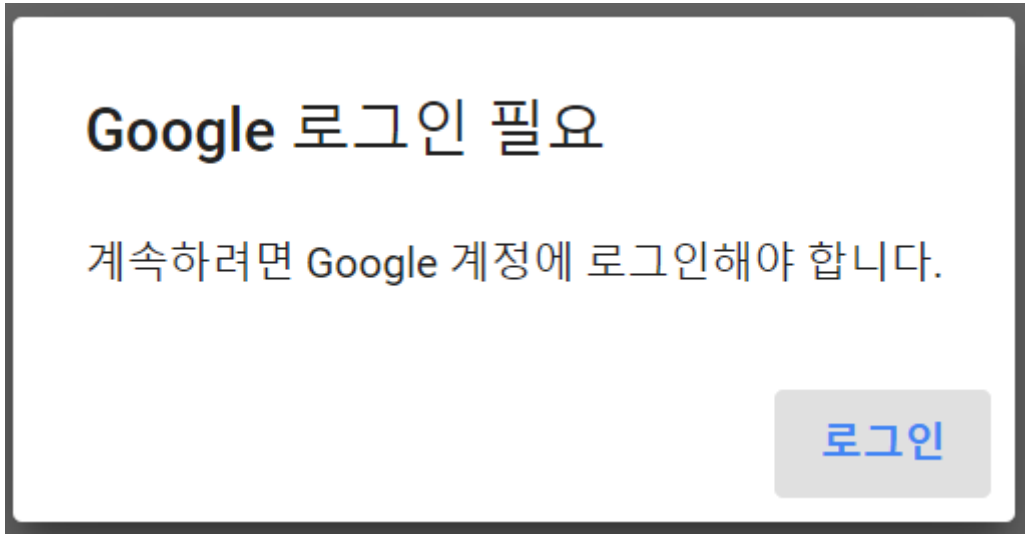


실습 환경 소개 – Google Colaboratory

초기 화면 및 로그인

- 파일 – 새 노트를 클릭하여 새로운 노트를 생성함
- 로그인이 되어 있지 않을 시 로그인을 요청함



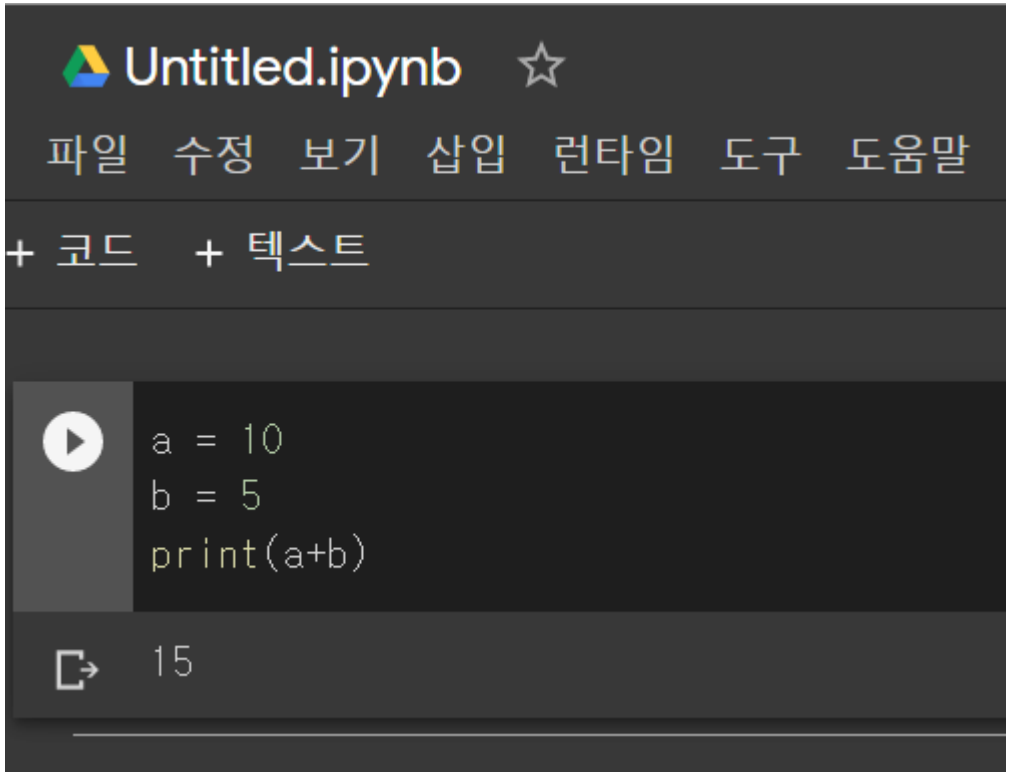


실습 환경 소개 – Google Colaboratory

Cell

- Google colab에서 작업을 위한 cell들의 리스트로 이루어져 있음
- 2가지 타입의 cell이 존재함

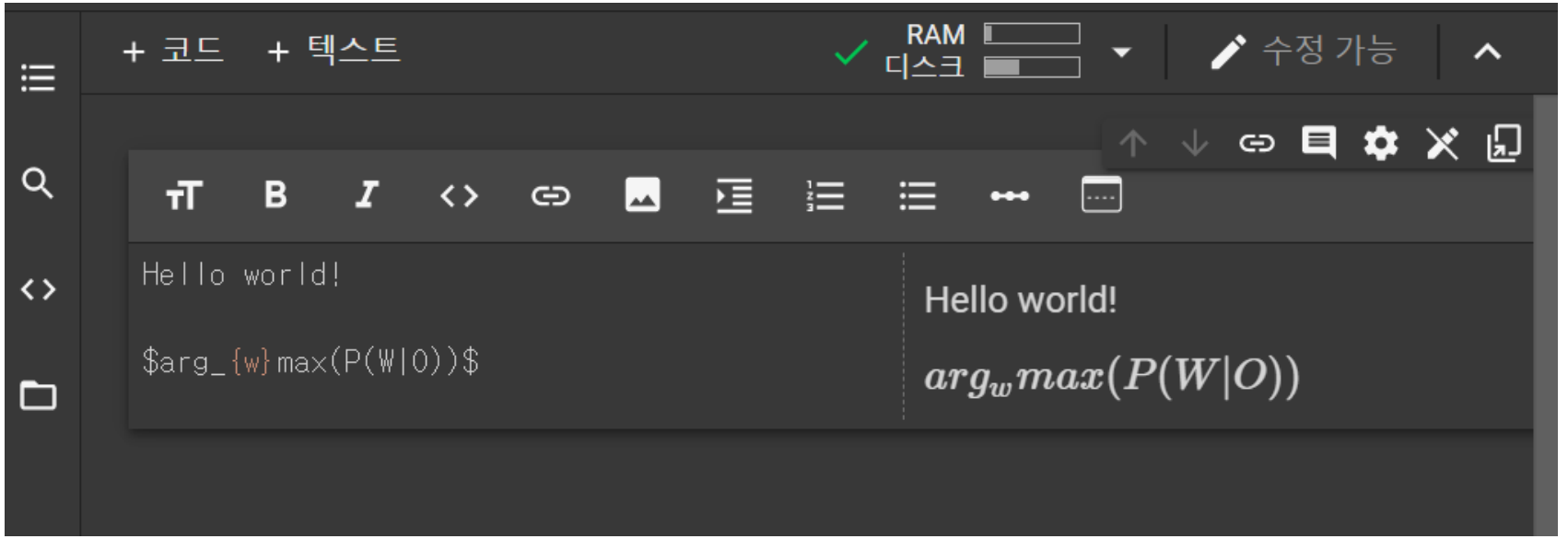
- Code cell
  - 코드의 기본 실행 단위로, 하나의 code cell에서 작성된 코드는 한 번에 실행됨



실습 환경 소개 – Google Colaboratory

Cell

- Text cell
  - Markdown syntax를 사용하고 latex 형식을 지원함



실습 환경 소개 – Google Colaboratory

Cell

- 실행 완료된 cell은 실행 순서대로 숫자가 매겨짐
- 같은 셀을 재시작하는 경우, 가장 최근에 실행된 셀의 다음 번호로 덮어씌워짐

```
[1]  print("Hello World!")

     Hello World!

[2]  a = 3
     b = 4

[3]  print("a: ", a)
     print("b: ", b)

     a:  3
     b:  4

[4]  c = a + b

[6]  print("c: ", c)

     c:  7

[7]  def func(a,b):
       print(a+b)

[8]  func(3,4)

     7
```

실습 환경 소개 – Google Colaboratory

Cell

- 실행하지 않은 code cell을 참조하려 할 경우, 아래와 같이 오류가 발생함

실습 architecture

Material

- TensorflowASR 기반 실습용 코드(https://github.com/indra622/tiny_sgspeech)
- LibriSpeech data 다운로드 및 모델 학습, 테스트를 진행할 수 있음
- Convolution-augmented transformer + RNN-Transducer 모델 지원

Reference

- TensorflowASR (https://github.com/TensorSpeech/TensorFlowASR)

실습 architecture

코드 기본 구조

- Dataset
- Featurizer
- Model
- Loss
- Optimizer
- Runner
- Utils

실습 pipeline

- Raw data(corpus) -> featurizer 정의 -> dataset 가공
- Model 정의 -> model build -> optimizer 정의 -> training
- Test

실습 architecture

Dataset

- Tab separated values(tsv)형식으로 정의된 음원 파일(flac)의 링크와 전사(transcript)를 입력 받아, tensorflow에서 지원하는 data loader object 형태로 변환해주는 역할을 함

```
1  /home/CORPUS/LibriSpeech/train-clean-100/374/180298/374-180298-0000.flac    14.53   chapter sixteen i might have told you of the beginning of this liaison in a few lines but i wanted you to see every step by which we came i to agree to whatever marguerite wished
2  /home/CORPUS/LibriSpeech/train-clean-100/374/180298/374-180298-0001.flac    16.09   marguerite to be unable to live apart from me it was the day after the evening when she came to see me that i sent her manon lescaut from that time seeing that i could not change my mistress's life i changed my own
3  /home/CORPUS/LibriSpeech/train-clean-100/374/180298/374-180298-0002.flac    13.29   i wished above all not to leave myself time to think over the position i had accepted for in spite of myself it was a great distress to me thus my life generally so calm
```

Featurizer

- Speech와 text 각각의 feature를 출력하는 역할을 담당함.
- Speech featurizer: log Mel-spectrogram, Mel-frequency cepstrum coefficients (MFCCs) 등의 특징 추출을 제공.
- Text featurizer: 출력 단위를 index로 변환함

실습 architecture

Model

- 학습 대상 모델을 정의함
- 본 실습에서는 convolution-augmented transformer (Conformer) 의 small size를 사용함

Runner

- 정의된 모델을 학습하거나, 학습한 모델을 테스트하기 위한 코드
- Trainer: loss function과 optimizer를 사용해서 모델을 학습하는 역할을 수행함
- Tester: 모델과 utils에 있는 성능 metric을 불러와서 성능을 측정하는 역할을 수행함

실습 architecture

Optimizer

- 모델 학습에 사용되는 optimizer와 scheduler를 정의함
- 본 실습에서는 Tensorflow에서 지원하는 adam optimizer 및, conformer 논문에서 사용하는 scheduler를 사용함

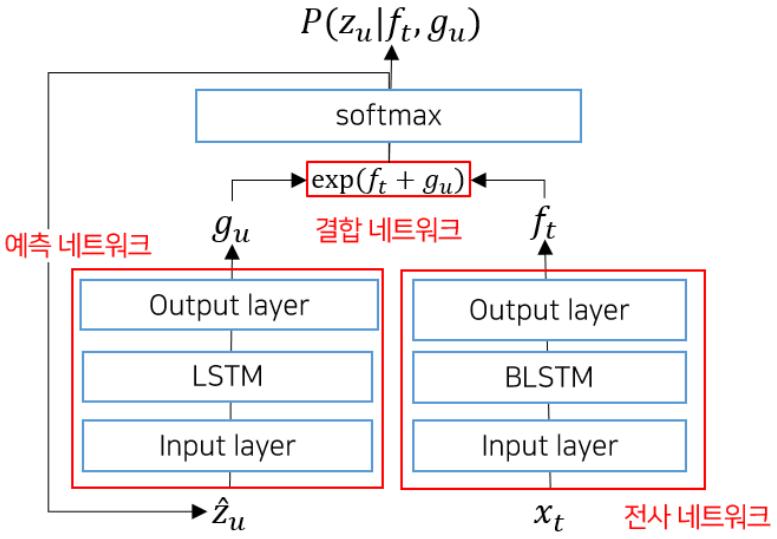
Loss

- 학습에 사용되는 loss function을 정의함
- Conformer는 RNN-T 구조를 사용하고 있기 때문에, Tensorflow에서 지원하는 RNN-T loss function를 사용함

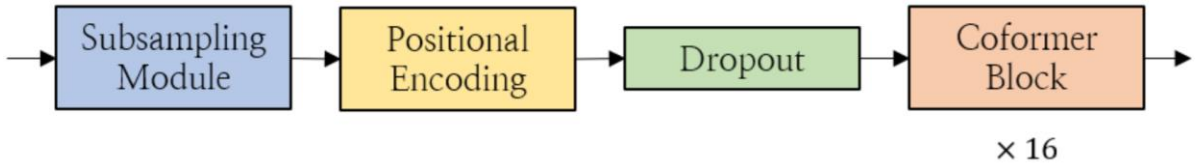
실습 대상 end-to-end 모델

Conformer

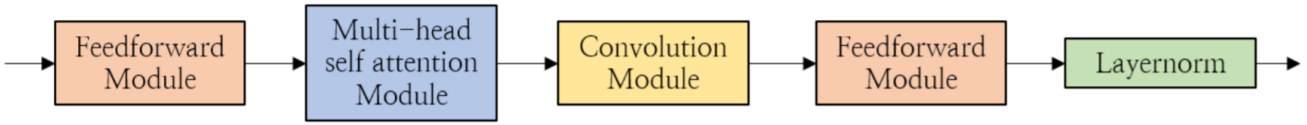
- Convolution block으로 구성된 Transformer encoder를 RNN-T의 transcription network로 사용한 모델
- 2021년 기준, 1,000시간의 LibriSpeech 학습 자료 기준 end-to-end 음성인식에서 가장 높은 성능을 보이고 있음
- (https://paperswithcode.com/sota/speech-recognition-on-librispeech-test-clean)
- 본 실습에서는 convolution-augmented transformer (Conformer) 의 경량화 버전을 사용함



[Vanila RNN-T 모델 구조]



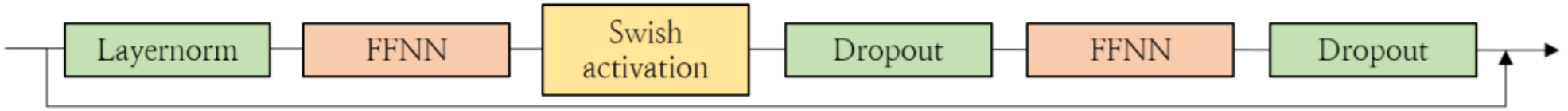
[Conformer encoder 구조]



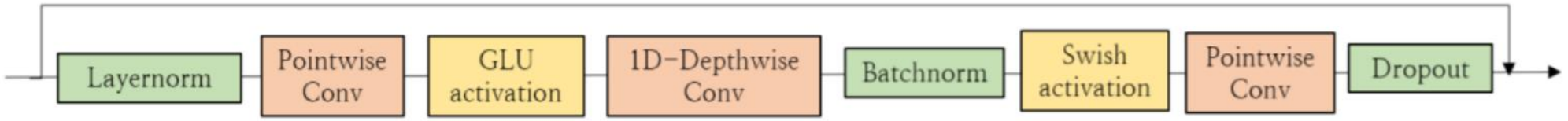
[Conformer block 구조]

실습 대상 end-to-end 모델

- **Conformer**



[Feed forward module 구조]

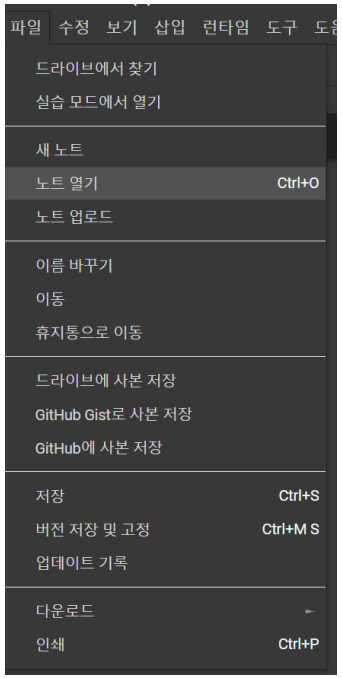


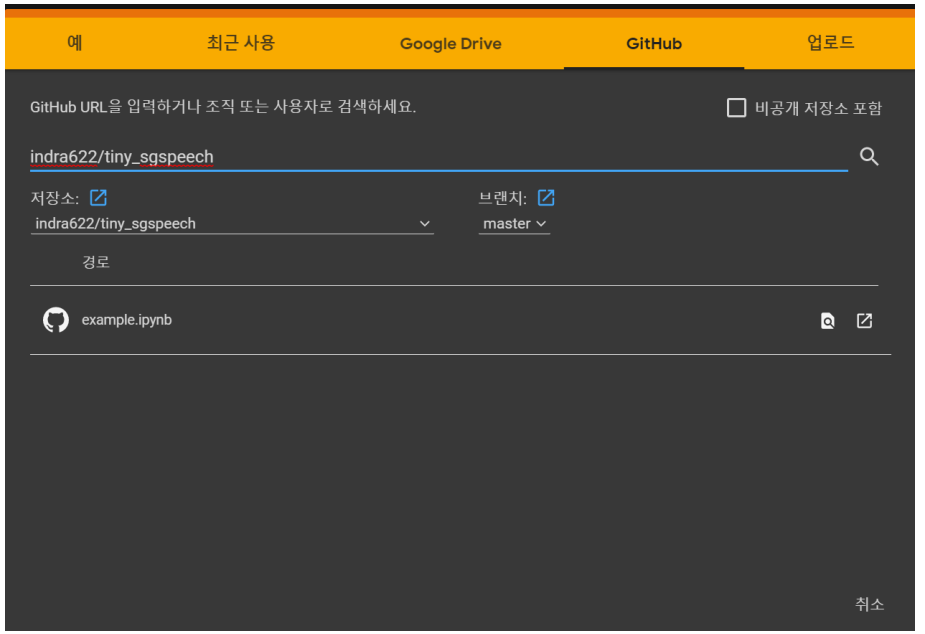
[Convolution module 구조]

Cheatsheet

◆ 노트 열기

- https://colab.research.google.com 에 접속 후 로그인
- 파일 -> 노트 열기 후 GitHub 탭에서 ‘indra622/tiny_sgspeech’ 입력 후 example.ipynb 클릭





Cheatsheet

◆ 실습 코드 및 데이터 준비

- git clone 명령어를 통해 실습 repository를 복사

```
!git clone https://github.com/indra622/tiny_sgspeech
```

```
Cloning into 'tiny_sgspeech'...
remote: Enumerating objects: 116, done.
remote: Counting objects: 100% (116/116), done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 116 (delta 45), reused 70 (delta 16), pack-reused 0
Receiving objects: 100% (116/116), 63.36 KiB | 12.67 MiB/s, done.
Resolving deltas: 100% (45/45), done.
```

Cheatsheet

◆ 실습 코드 및 데이터 준비

- wget 명령어를 통해 LibriSpeech 데이터 일부를 다운로드
- 본 실습에서는 총 1,000시간의 데이터 중, 100시간의 학습 데이터와 dev-clean 데이터를 사용함

```
!wget https://www.openslr.org/resources/12/train-clean-100.tar.gz
!wget https://www.openslr.org/resources/12/dev-clean.tar.gz
```

```
--2021-07-01 09:40:44--  https://www.openslr.org/resources/12/train-clean-100.tar.gz
Resolving www.openslr.org (www.openslr.org)... 46.101.158.64
Connecting to www.openslr.org (www.openslr.org)|46.101.158.64|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6387309499 (5.9G) [application/x-gzip]
Saving to: 'train-clean-100.tar.gz'

train-clean-100.tar 100%[===================>]   5.95G  32.9MB/s    in 3m 8s

2021-07-01 09:43:53 (32.3 MB/s) - 'train-clean-100.tar.gz' saved [6387309499/6387309499]
```

Cheatsheet

◆ 실습 코드 및 데이터 준비

- tar 명령어를 통해 데이터 압축 해제 후, 압축 파일 삭제

```
!tar -xvzf train-clean-100.tar.gz && rm train-clean-100.tar.gz
!tar -xvzf dev-clean.tar.gz && rm dev-clean.tar.gz
```

```
LibriSpeech/dev-clean/3853/163249/
LibriSpeech/dev-clean/3853/163249/3853-163249-0029.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0049.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0009.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0006.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0026.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0035.flac
LibriSpeech/dev-clean/3853/163249/3853-163249-0056.flac
```

Cheatsheet

◆ 실습 코드 및 데이터 준비

- create_librispeech_trans.py 코드를 사용하여 tsv 파일 형식으로 dataset을 가공

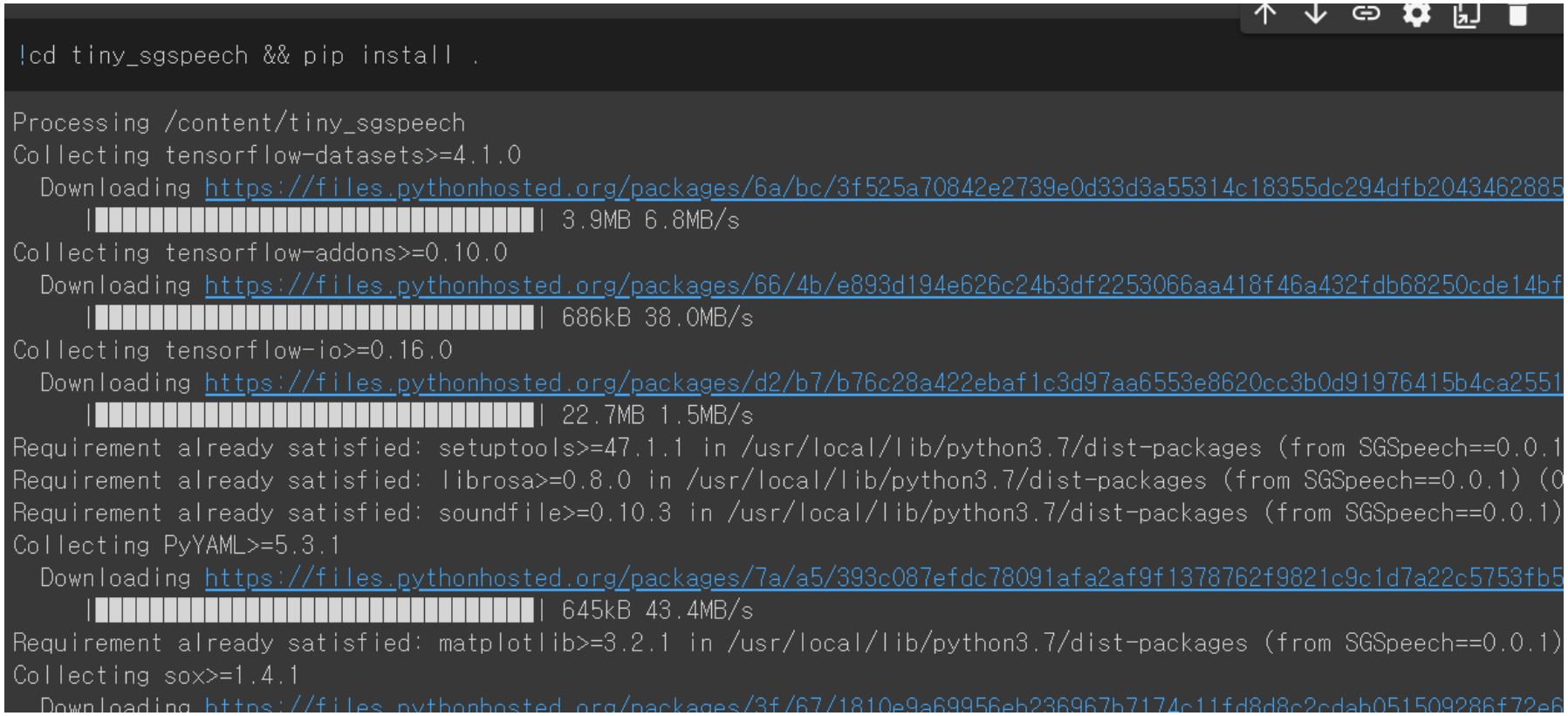
```
!python tiny_sgspeech/create_librispeech_trans.py --dir /content/LibriSpeech/train-clean-100 /content/LibriSpeech/train-clean-100/transcripts.tsv
!python tiny_sgspeech/create_librispeech_trans.py --dir /content/LibriSpeech/dev-clean /content/LibriSpeech/dev-clean/transcripts.tsv

2021-07-01 09:47:43.880301: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
[Loading]: 100% 585/585 [02:41<00:00,  3.62it/s]
[Writing]: 100% 28539/28539 [00:00<00:00, 915035.18it/s]
2021-07-01 09:50:29.846528: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
```

- 학습 자료에 대한 .tsv 파일

Cheatsheet

◆ 외부 라이브러리 설치 및 패키지 설치

- pip install 명령어를 통해 패키지 설치 (setup.py)



Cheatsheet

- **GPU 환경 설정**

```
[6]  from tiny_sgspeech.sgspeech.utils import setup_environment, setup_strategy

     setup_environment()
     strategy = setup_strategy([0])
```

```
     Run on 1 Physical GPUs
```

Cheatsheet

- **Data preparation**
  - Featurizer 및 dataset 불러오기
  - 임의의 training set에 대해 audio 내용 확인

```
from tiny_sgspeech.sgspeech.configs.config import Config
from tiny_sgspeech.sgspeech.featurizers.speech_featurizer import NumpySpeechFeaturizer
from tiny_sgspeech.sgspeech.featurizers.text_featurizer import CharFeaturizer

config = Config('/content/tiny_sgspeech/config.yml')
speech_featurizer = NumpySpeechFeaturizer(config.speech_config)
text_featurizer = CharFeaturizer(config.decoder_config)
```

```
from tiny_sgspeech.sgspeech.datasets.speech_dataset import SpeechSliceDataset
from tiny_sgspeech.sgspeech.featurizers.text_featurizer import CharFeaturizer

train_dataset = SpeechSliceDataset(
    speech_featurizer=speech_featurizer, text_featurizer=text_featurizer,
    **vars(config.learning_config.train_dataset_config)
)
eval_dataset = SpeechSliceDataset(
    speech_featurizer=speech_featurizer, text_featurizer=text_featurizer,
    **vars(config.learning_config.eval_dataset_config)
)
```

```
td = next(iter(train_dataset.create(1)))
speech_link = td[0]
speech_feature = td[1]
speech_duration = td[2]
transcription = td[3]
transcription_length = td[4]
```

```
Read files
```

```
import IPython.display as ipd
import tensorflow as tf

speech_path = speech_link[0].numpy().decode('utf-8')
ipd.Audio(speech_path)
```

서강대학교 SOGANG UNIVERSITY

Cheatsheet

- **Data preparation**
  - Transcription에 대한 index 및 내용 확인
  - 오디오 원음과 feature 형태 확인

```
transcription
```

```
<tf.Tensor: shape=(1, 77), dtype=int32, numpy=
array([[15, 16, 21,  1,  7, 19, 16, 14,  1,  4,  9, 16, 10,  4,  6,  1,
         9, 16, 24,  6, 23,  6, 19,  1,  2,  1, 17,  2, 19, 21,  1, 16,
         7,  1,  9,  6, 19,  1, 19,  6, 13, 10,  8, 10, 16, 15,  1, 20,
         9,  6,  1,  5,  6, 19, 10, 23,  6,  5,  1,  7, 19, 16, 14,  1,
        21,  9,  2, 21,  1, 22,  8, 13, 10, 15,  6, 20, 20]], dtype=int32)>
```

```
text_featurizer.iextract(transcription)
```

```
<tf.Tensor: shape=(1,), dtype=string, numpy=
array([b'not from choice however a part of her religion she derived from that ugliness'],
      dtype=object)>
```

```
import librosa
import os

raw_audio, _ = librosa.load(os.path.expanduser(speech_path), sr=16000, mono=True)
```

```
print(raw_audio.shape)
print(speech_feature.shape)
```

```
(73360,)
(1, 456, 80, 1)
```

Cheatsheet

모델 정의 및 학습

- Trainer 및 conformer 모델 형태 선언
- 모델 topology 확인

```
from tiny_sgspeech.sgspeech.runners.transducer_runners import TransducerTrainer


conformer_trainer = TransducerTrainer(
    config=config.learning_config.running_config,
    text_featurizer=text_featurizer, strategy=strategy
)
```

```
Use RNNT loss in TensorFlow
```

```
from tiny_sgspeech.sgspeech.models.conformer import Conformer

cf = Conformer(**config.model_config, vocabulary_size=text_featurizer.num_classes)
```

```
cf._build(speech_featurizer.shape)
```

```
cf.summary(line_length=150)
```

```
Model: "conformer_encoder"
_____________________________________________________________________________________________________________________________________________
Layer (type)                                                                Output Shape                                                    Param #
=============================================================================================================================================
conformer_encoder_subsampling (Conv2dSubsampling)                           multiple                                                        188208
_____________________________________________________________________________________________________________________________________________
conformer_encoder_pe (PositionalEncodingConcat)                             multiple                                                        0
_____________________________________________________________________________________________________________________________________________
conformer_encoder_linear (Dense)                                            multiple                                                        414864
_____________________________________________________________________________________________________________________________________________
conformer_encoder_dropout (Dropout)                                         multiple                                                        0
_____________________________________________________________________________________________________________________________________________
conformer_encoder_block_0 (ConformerBlock)                                  multiple                                                        506736
_____________________________________________________________________________________________________________________________________________
conformer_encoder_block_1 (ConformerBlock)                                  multiple                                                        506736
_____________________________________________________________________________________________________________________________________________
conformer_encoder_block_2 (ConformerBlock)                                  multiple                                                        506736
_____________________________________________________________________________________________________________________________________________
conformer_encoder_block_3 (ConformerBlock)                                  multiple                                                        506736
_____________________________________________________________________________________________________________________________________________
conformer_encoder_block_4 (ConformerBlock)                                  multiple                                                        506736
_____________________________________________________________________________________________________________________________________________
```

Cheatsheet

모델 정의 및 학습

- Model build 및 optimizer 정의
- Model compile 및 학습

```
from tiny_sgspeech.sgspeech.optimizers.schedules import TransformerSchedule
import tensorflow as tf
import math

with conformer_trainer.strategy.scope():
    # build model
    conformer = Conformer(**config.model_config, vocabulary_size=text_featurizer.num_classes)
    conformer._build(speech_featurizer.shape)

    optimizer_config = config.learning_config.optimizer_config
    optimizer = tf.keras.optimizers.Adam(
        TransformerSchedule(
            d_model=conformer.dmodel,
            warmup_steps=optimizer_config["warmup_steps"],
            max_lr=(0.05 / math.sqrt(conformer.dmodel))
        ),
        beta_1=optimizer_config["beta1"],
        beta_2=optimizer_config["beta2"],
        epsilon=optimizer_config["epsilon"]
    )
```

```
conformer_trainer.compile(model=conformer, optimizer=optimizer,
                          max_to_keep=10)

conformer_trainer.fit(train_dataset, eval_dataset, train_bs=2, eval_bs=1)
```

Cheatsheet

◆ 모델 평가

- Model 형태 정의 및 pretrained 모델 불러오기

```
test_cf = Conformer(**config.model_config, vocabulary_size=text_featurizer.num_classes)
test_cf._build(speech_featurizer.shape)
test_cf.load_weights('/content/conformer.h5')
test_cf.summary(line_length=150)
test_cf.add_featurizers(speech_featurizer, text_featurizer)
```

```
Model: "conformer_encoder"
_____________________________________________________________________________________________________________
Layer (type)                                                Output Shape                                Param #
=============================================================================================================
conformer_encoder_subsampling (Conv2dSubsampling)           multiple                                    188208
_____________________________________________________________________________________________________________
conformer_encoder_pe (PositionalEncodingConcat)             multiple                                    0
_____________________________________________________________________________________________________________
conformer_encoder_linear (Dense)                            multiple                                    414864
_____________________________________________________________________________________________________________
conformer_encoder_dropout (Dropout)                         multiple                                    0
```

Cheatsheet

모델 평가

- 결과 확인
- Label: 정답으로 주어진 transcription
- Prediction: 인식기 결과

```
from tiny_sgspeech.sgspeech.runners.base_runners import BaseTester

conformer_tester = BaseTester(
    config=config.learning_config.running_config,
    output_name='result'
)
conformer_tester.compile(test_cf)
conformer_tester.run(eval_dataset, batch_size=1)
```

```
Label:['anybody before the father came in']
Prediction:['anybody before that came in']
[Test]:  91%|          | 2461/2703 [15:04<01:07,  3.60batch/s]
Label:['yes miss clarke the middle aged lady with the parrishes']
Prediction:['yes misclark the midle aged lady with therishes']
[Test]:  91%|          | 2462/2703 [15:04<01:09,  3.45batch/s]
Label:['i suppose she has been carefully questioned very i should say']
Prediction:['i suppose she has been carefully questioned very i should say']
```