DeepSpeech2 End-to-End Speech Recognition in English and Mandarin

Baidu Research – Silicon Valley AI Lab
Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro,
Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel,
Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley,
Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman,
Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang,
Bo Xiao, Dani Yogatama, Jun Zhan, Zhenyao Zhu

Contents

1. Introduction

2. Model Architecture

3. DeepSpeech2

4. Results

5. Conclusion

1. Introduction

Speech Recognition(음성인식)..?

Speech -> Text

인간의 말(speech)을 기계가 글(text)로 바꿔주는 기술

음성인식의 challenge (해결해야할 과제)

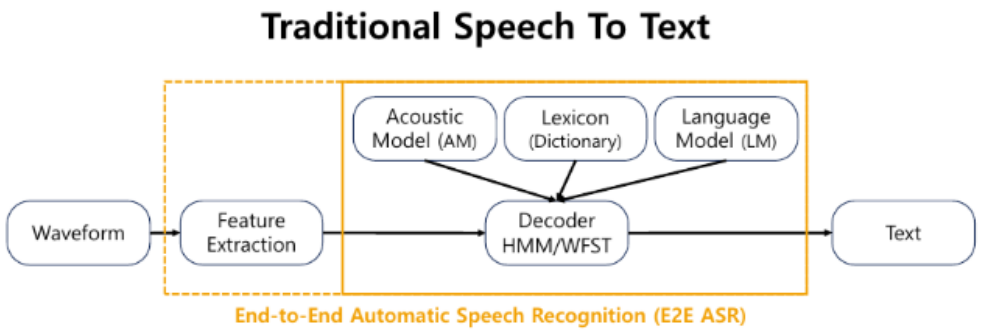
- 음성과 음향의 광범위한 가변성

: 화자의 억양, 어조, 언어, 발음 등이 달라져도 같은 것으로 인식할 수 있어야 함.

- 여러 엔진이 필요함

: 예전에는 Acoustic 모델 등
3개의 모델을 조합하여
사용해야했음.

=> 단일모델로 통합 필요!



Traditional Speech To Text (from 강의 ppt)

1. Introduction

Deep Speech2의 Contribution

① End-to-End 모델

: 3개의 모델을 따로 훈련한 결과를 결합하여 사용해야해서 사용법이 까다로웠던 단점을 극복하고 End-to-End 단일 엔진 모델로 음성인식 딥러닝 모델 개발

② 단순한 모델 구조

: RNN을 활용한 단순한 구조의 딥러닝 모델과 CTC loss를 사용한 간단한 구조

③ 다양한 벤치마크에서 인간 수준 성능 달성

: 월스트리트저널 뉴스 기사(WSJ), LibriSpeech, VoxForge, CHiME

④ 실산업에 배포및 활용 가능

⑤ 대규모 데이터셋 학습 (영어 11940시간, 중국어 9400시간)

⑥ 효율적인 학습 테크닉

1. Introduction

Deep Speech2를 읽으면서 집중한 부분

① input 과 output 은 어떤 데이터인가? 그 모양(shape)은?

② 네트워크는 어떻게 구성되어 있는가?

  ㄴ 그 네트워크를 사용한 이유는?

③ loss function 은 무엇인가?

  ㄴ 그 함수를 사용한 이유는?

④ 성능 향상을 위해 사용한 테크닉은?

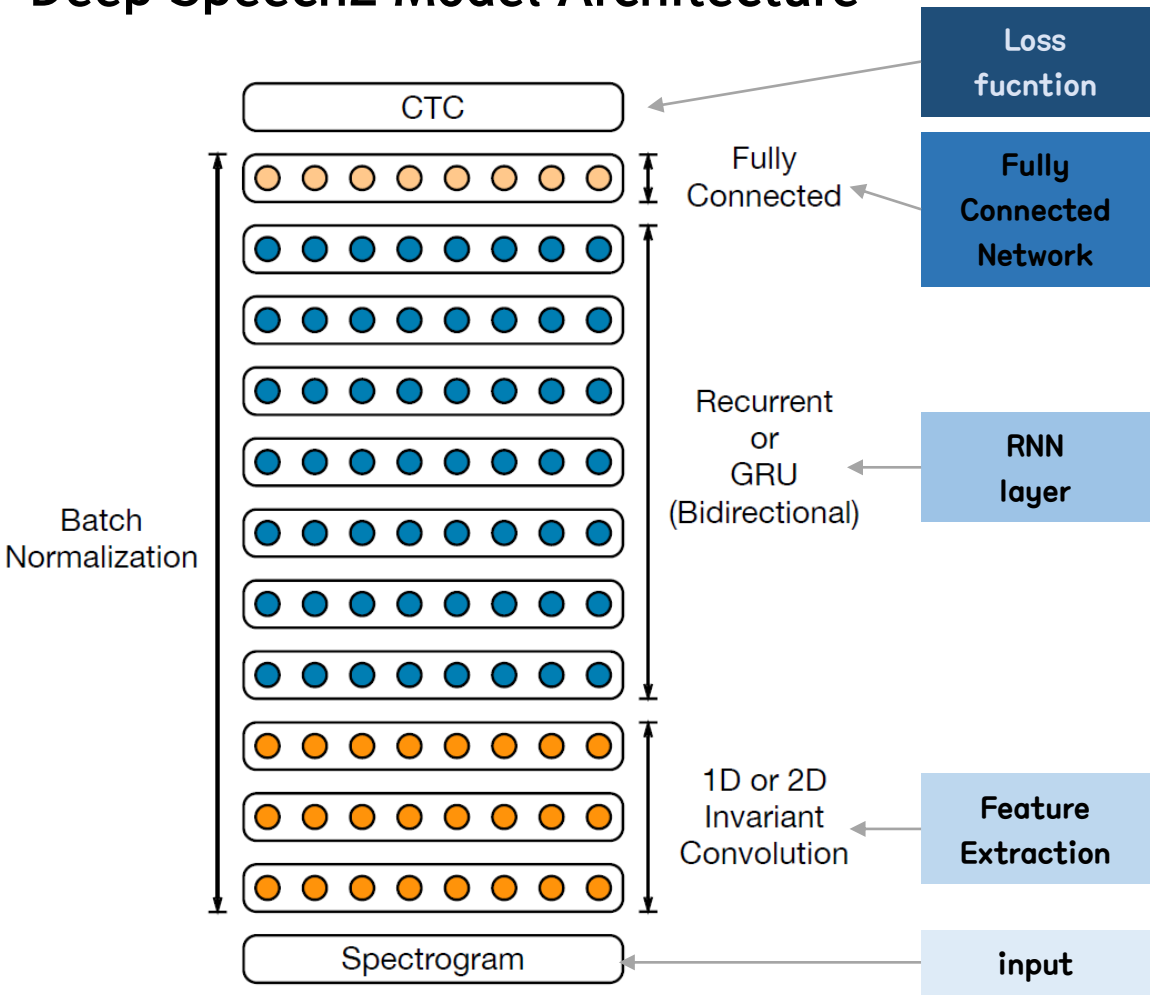
⑤ 벤치마크 성능 측정은 어떻게 하였는가?

  ㄴ 어떤 벤치마크 데이터셋을 사용했는가?

  ㄴ 성능 측정 지표는?

2. Model Architecture

Deep Speech2 Model Architecture



CTC loss 활용.

RNN output 을 CTC로 계산 가능하게 변환

하나 이상의 RNN or GRU 를 사용하여 Hidden layer 구성

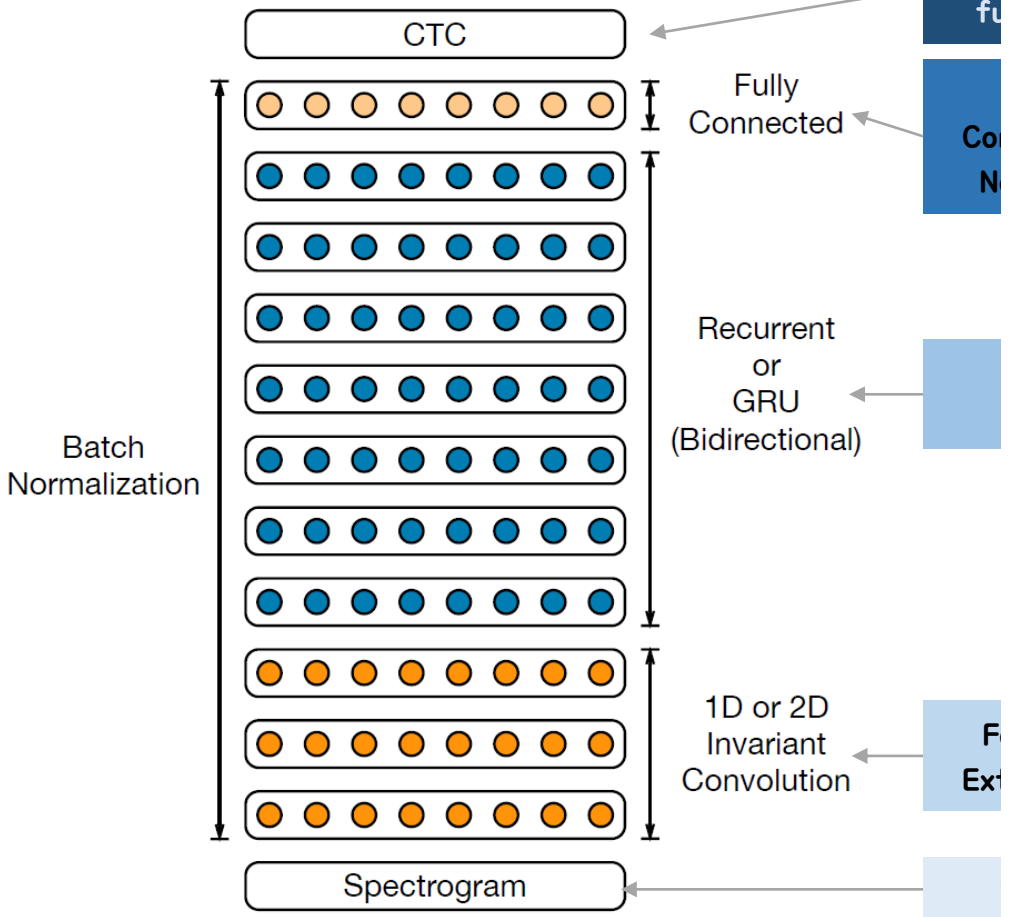
Conv layer 를 통해 spectrogram의 feature 를 추출함

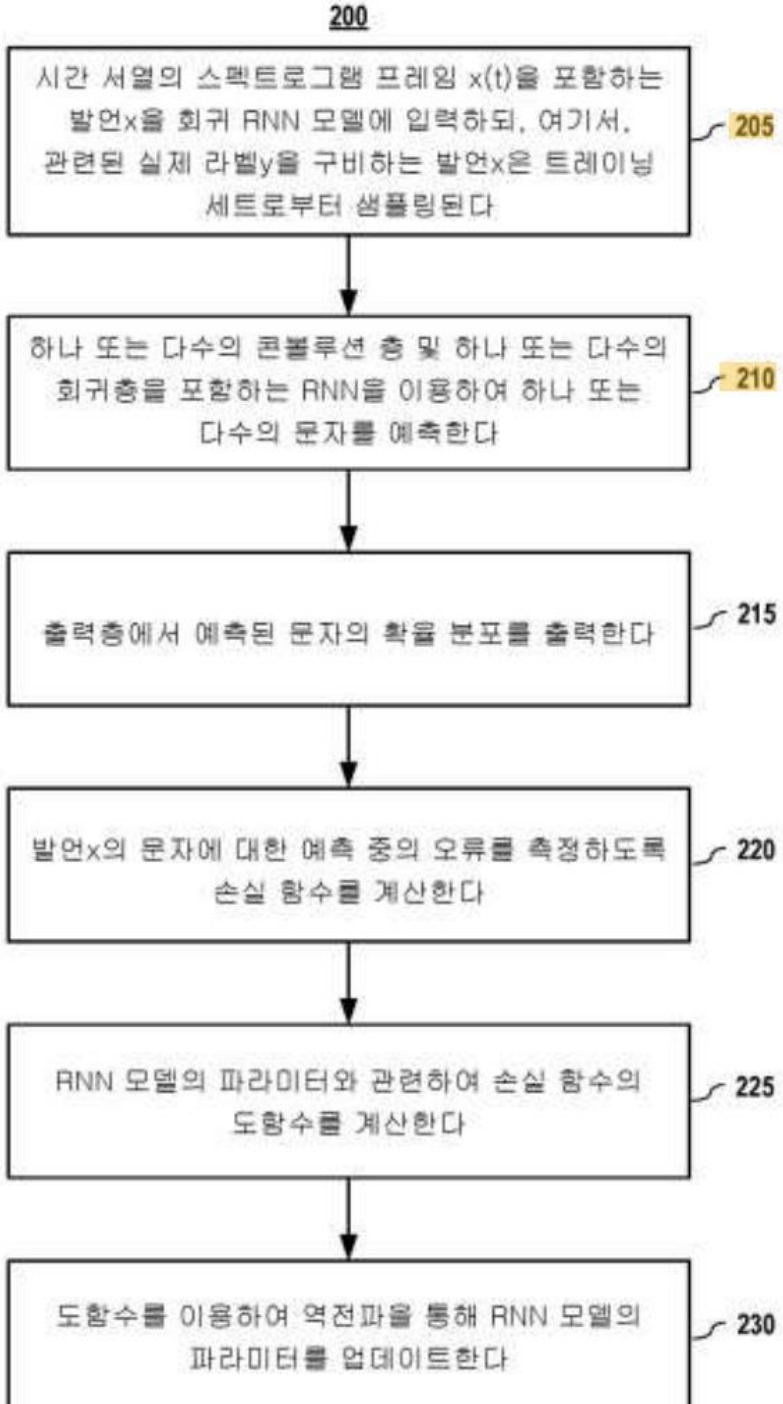
발화의 spectrogram 을 input으로 함.

2. Model Architecture

https://patents.google.com/patent/KR20170107015A/ko

Deep Speech2 Model Architecture





게 변환

여 Hidden layer 구성

l feature 를 추출함

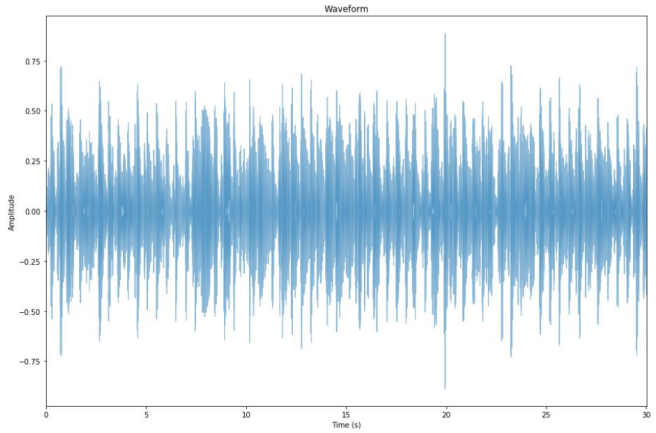
함.

2. Model Architecture

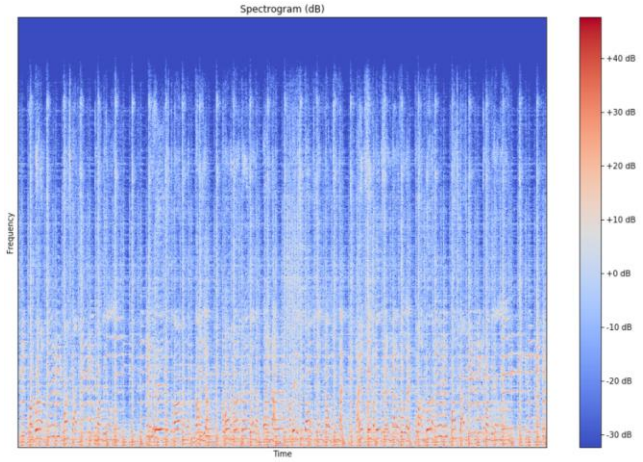
① Dataset

https://hyunlee103.tistory.com/36

input x: 발화데이터



음파 데이터



Spectrogram

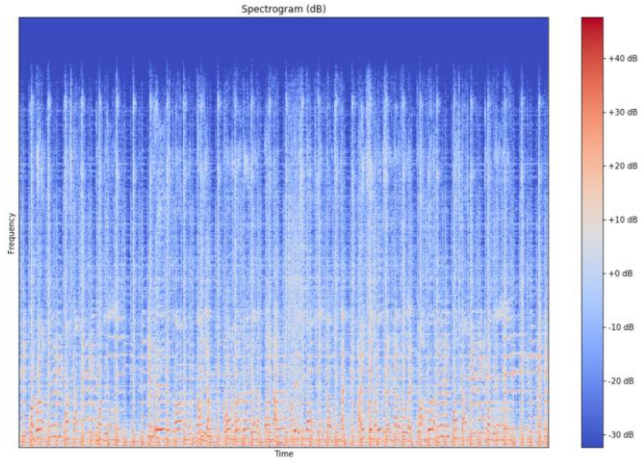
시간에 따른 각 주파수의
음향 에너지를 시각화한 것.
x 축: time
y 축: frequency (음색) 소리의 높낮이
z 축: amplitude(진폭. 강도) 소리의 세기

2. Model Architecture

① Dataset

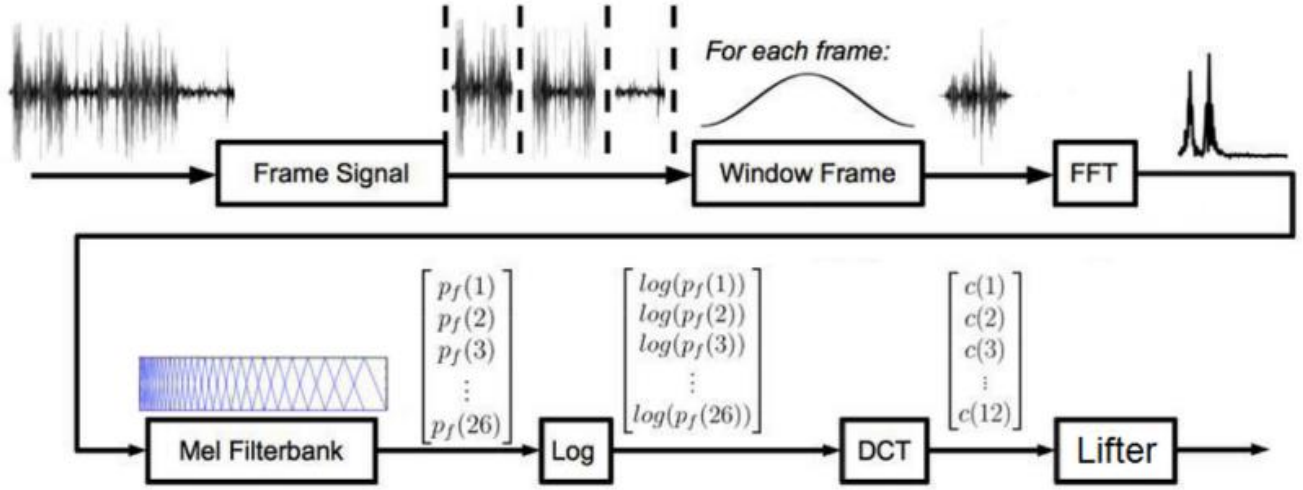
https://youdaeng-com.tistory.com/5

input x: 발화데이터

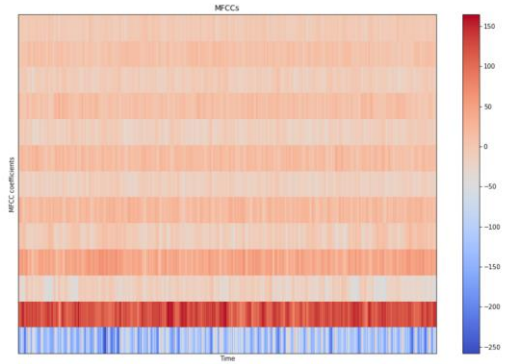


Spectrogram

신호의 성질을 잘 반영하는
feature를 뽑아서 사용



MFCC feature extraction



MFCCs

가장 대표적인
오디오 feature를
뽑는 과정

2. Model Architecture

① Dataset

https://youdaeng-com.tistory.com/5

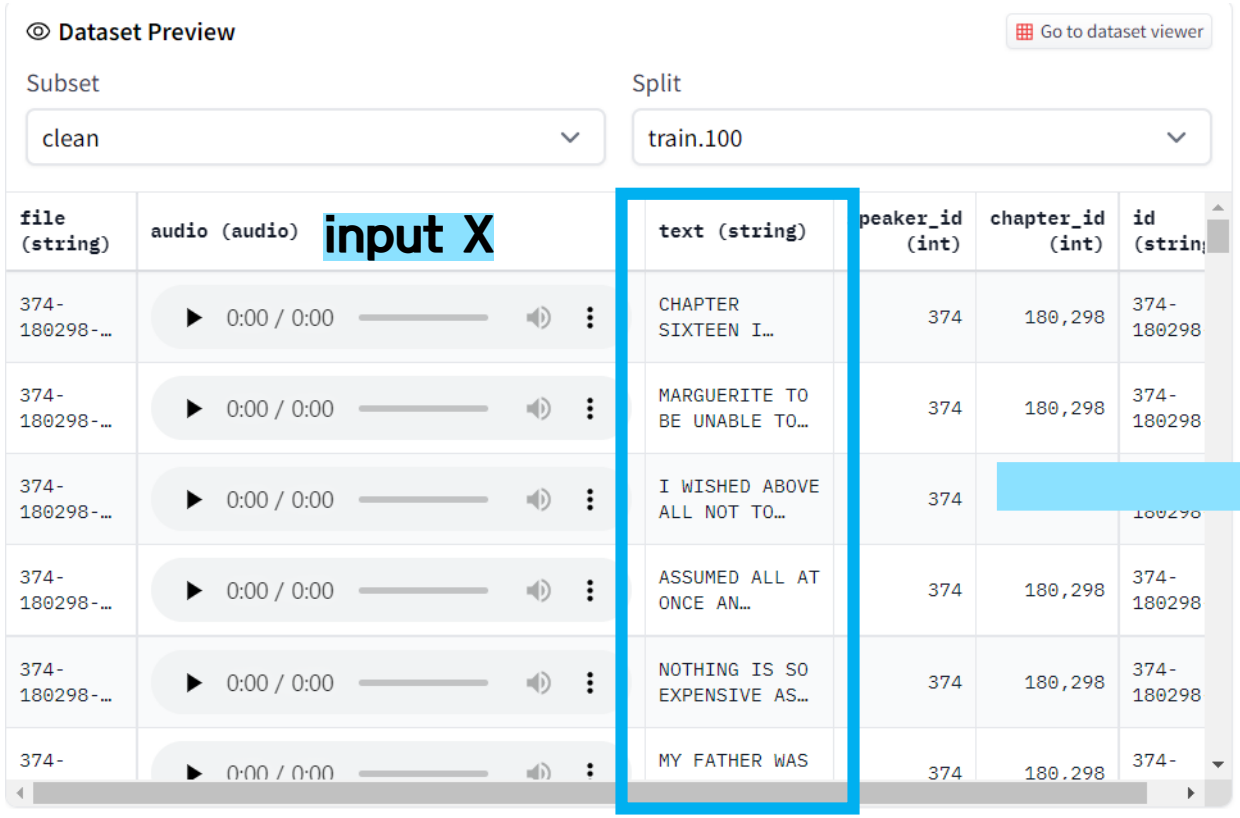
- 몇 분 ~ 몇 시간 정도의 데이터로, 데이터 간의 시간 차이가 크고, 데이터 크기가 커서 데이터 길이를 조정함.
- 평균 7초 길이의 데이터로 조정
- 잘리는 발음이 없도록 audio-transcription 쌍 (x, y)에 대해 전처리 수행

2. Model Architecture

① Dataset

https://huggingface.co/datasets/librispeech_asr

output y: 발화데이터의 label

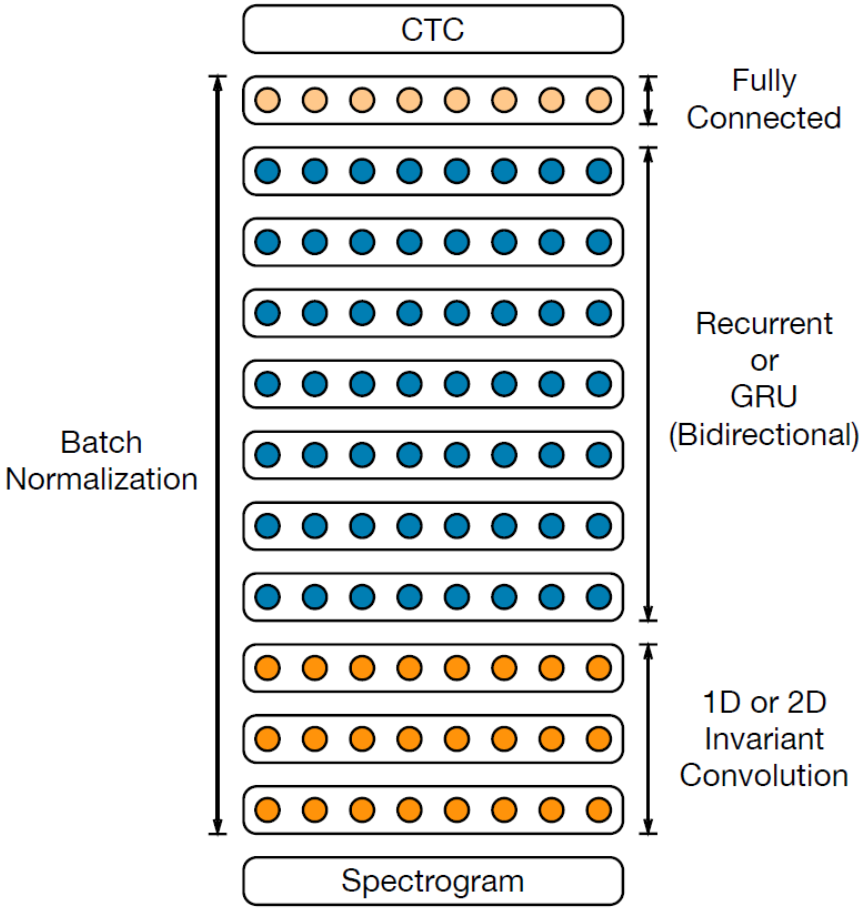


Librispeech Dataset Preview

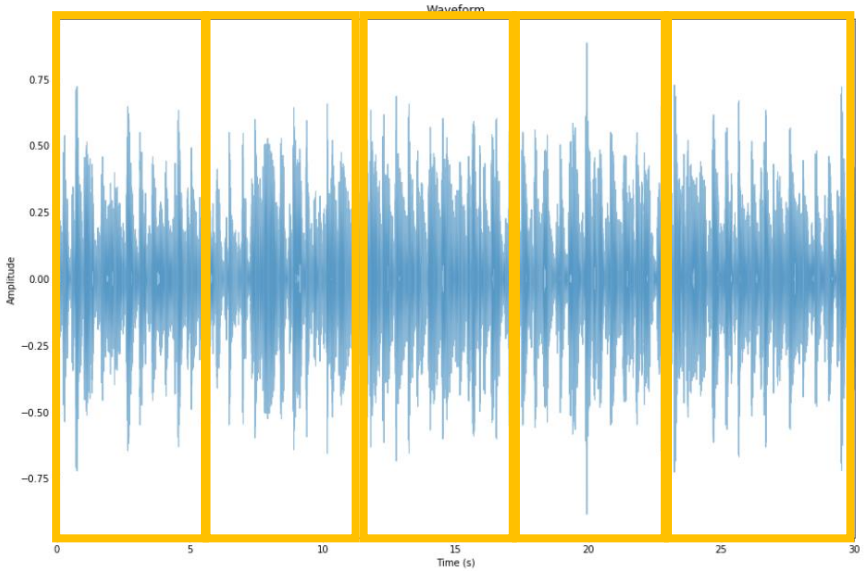
음성 데이터에 대한
글(Text)이
발화데이터 x에 대한
label, y가 됨.

2. Model Architecture

② Feature extraction

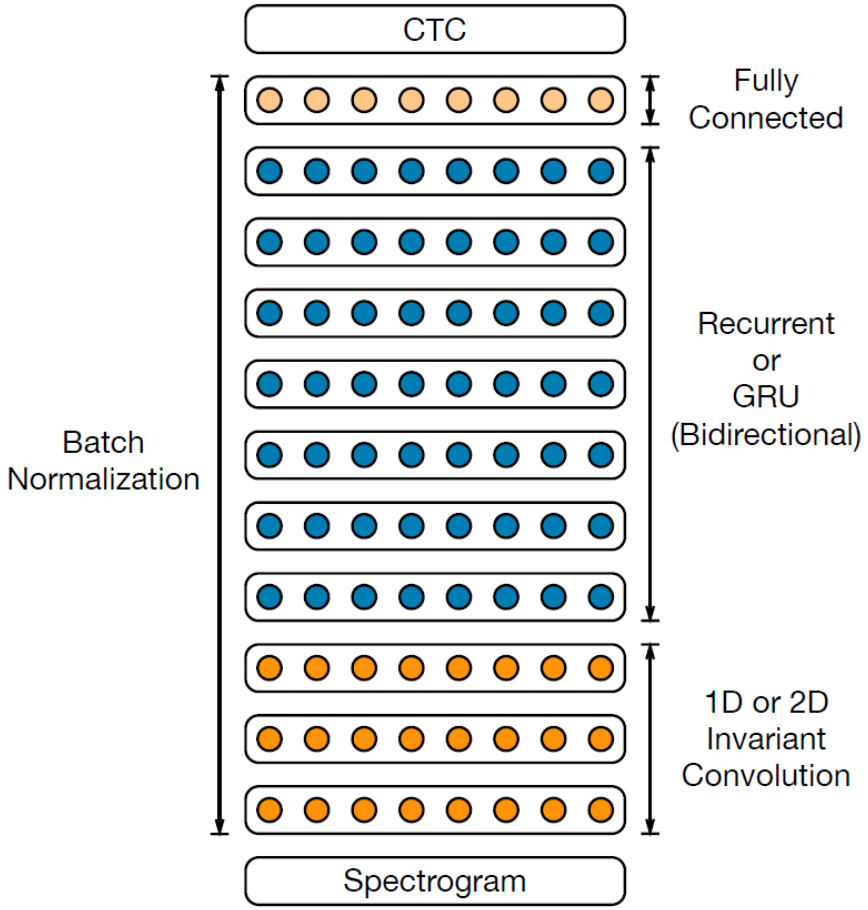


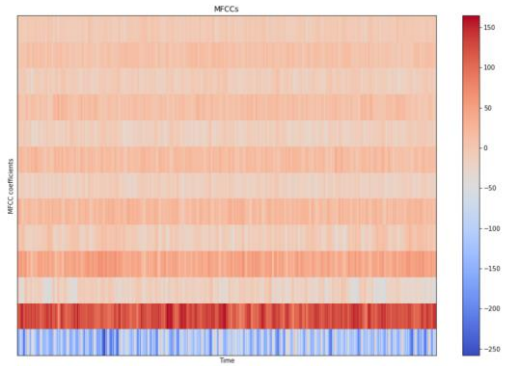
- temporal convolution 은 발화 길이가 다양한 음성 데이터의 특성을 효율적으로 모델링하기 위해 음성 인식에 일반적으로 사용된다.



2. Model Architecture

③ RNN layer



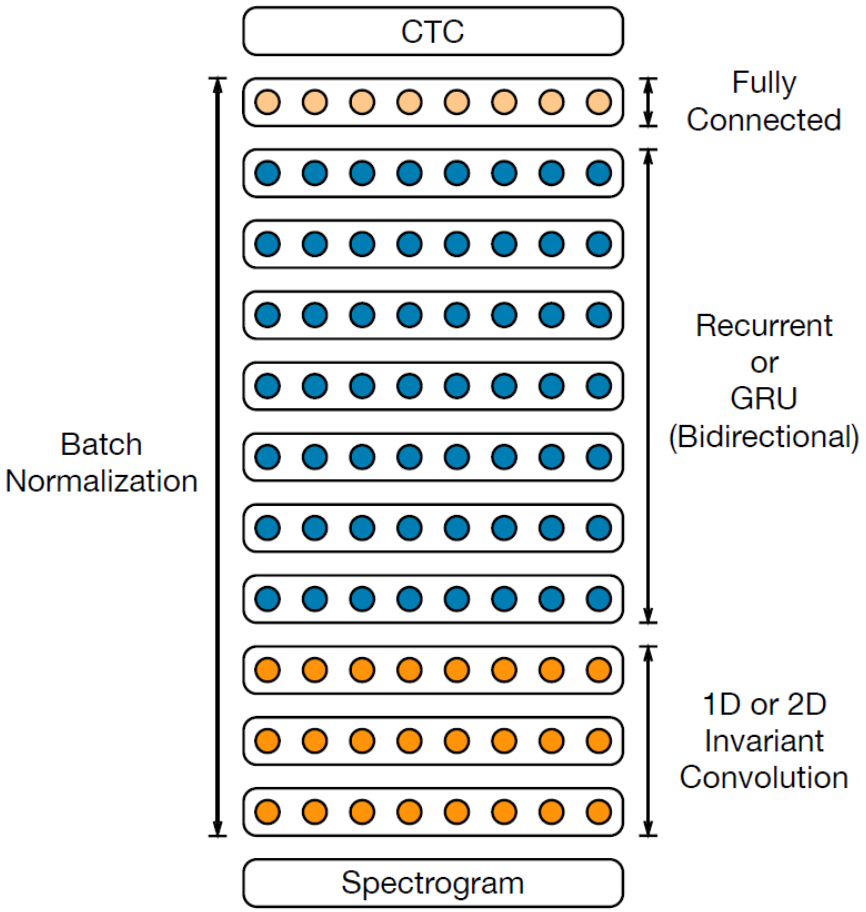


- 음파 데이터는 시간에 따른 소리 데이터 => 시계열 데이터
  => RNN 사용

- (ds1) LSTM은 메모리를 너무 많이 사용하고 연산속도가 느려서 기본 RNN 사용
  - (LSTM은 각 단계에서 여러 개의 게이트 뉴런 응답을 계산하고 저장한다는 단점)

2. Model Architecture

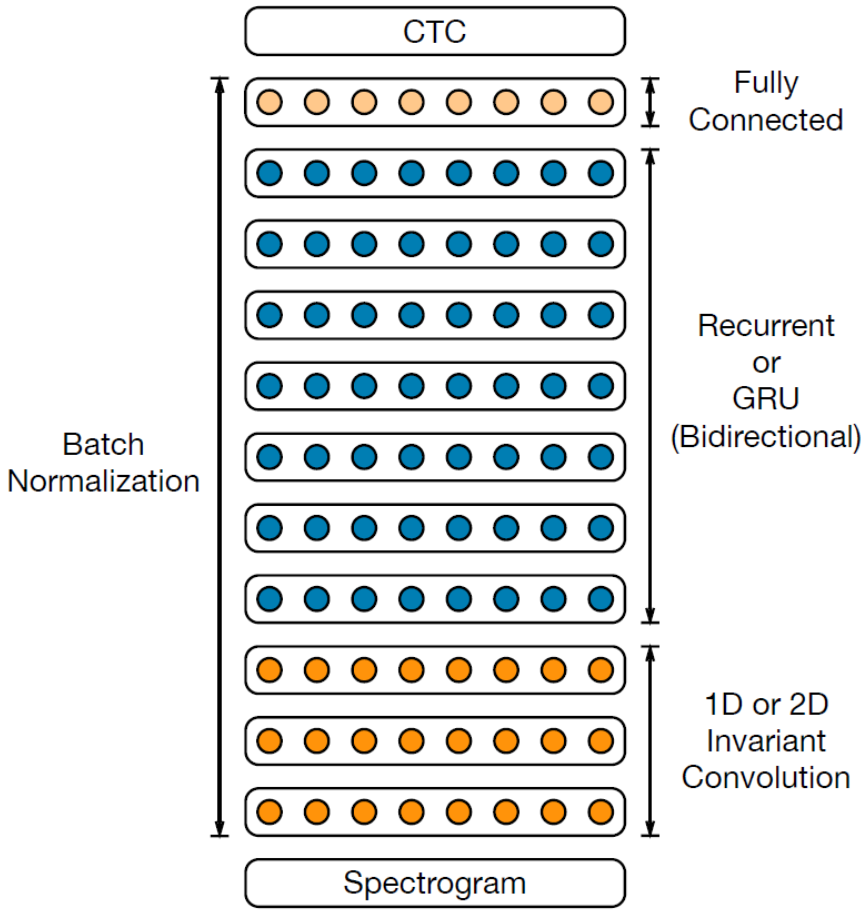
④ Fully Connected Layer



(CTC loss 연산을 위해)
RNN의 output으로 예측된 문자의 확률분포 출력

2. Model Architecture

⑤ CTC Loss function

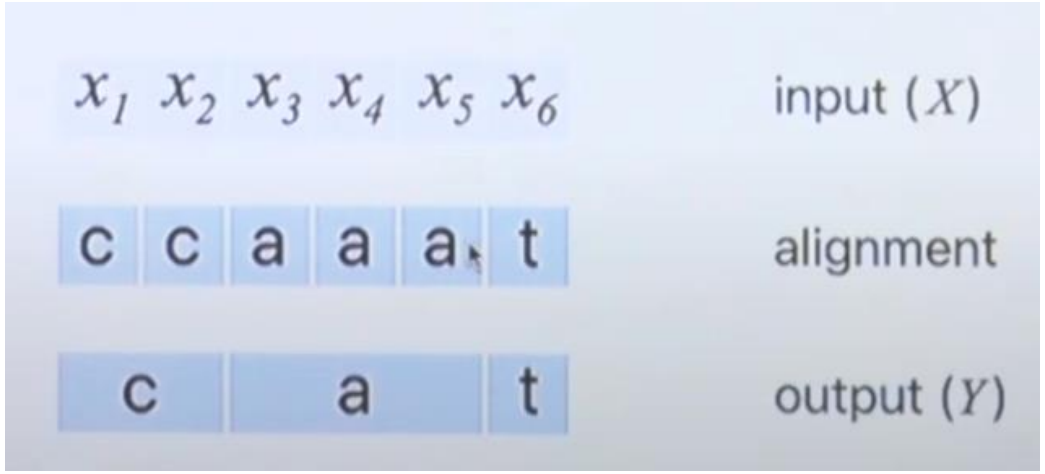


https://www.cs.toronto.edu/~graves/icml_2006.pdf

Connectionist Temporal Classification(CTC)

: input과 output 의 alignment 맞추기

Pytorch 패키지: https://pytorch.org/docs/stable/generated/torch.nn.CTCLoss.html



https://www.youtube.com/watch?v=xQ0kkGb5gLk

x: **발화데이터**

$\hat{y}$: **예측한 텍스트**

y: x**의** label

3. DeepSpeech2

DeepSpeech2 는 DeepSpeech1 과 핵심 구조는 같지만, 성능을 향상시킨 모델

- Batch Normalization
- SortaGrad
- RNNs VS GPUs
- Frequency Convolution (1D vs 2D)

3. DeepSpeech2

① Batch Normalization

https://arxiv.org/pdf/1510.01378.pdf

sequence-wise normalization 적용

: vertical connection 에만 BatchNorm 적용

미니배치 안에서 각각 hidden unit의 전체 길이에 대해 mean, variance 계산

$$\bar{\mathbf{x}}_k = \frac{1}{n}\sum_{i=1}^{m}\sum_{t=1}^{T}\mathbf{x}_{i,t,k},$$

$$\sigma_k^2 = \frac{1}{n}\sum_{i=1}^{m}\sum_{t=1}^{T}(\mathbf{x}_{i,t,k} - \bar{\mathbf{x}}_k)^2,$$

T : 각 시퀀스의 길이

n : 미니 배치에서 unpadded frames의 총 수

- RNN 훈련속도가 빨라짐
- 매우 큰 dataset, 깊은 layer에 적용하면 generalization error 줄이기 가능

test error라고도 함. 관측하지 못한 새로운 input에 대한 error의 기댓값

② SortaGrad

학습 초기에서 불안정한 상황 가끔 발생함.

- 학습 초기 output 과 gradient 가 잘 조정되지 않은 레이어를 많이 통과하므로 학습을 잘 못함.
- CTC loss 사용시 gradients exploding (발산) 종종 발생

=〉 SortaGrad 방식 개발

: 미니배치 구성 방식을 CTC loss에 맞도록 변형하는 데이터셋 구성방식

CTC loss 사용시 음성 시퀀스가 길어질수록 loss와 gradient가 커짐 =〉 학습이 어려움.

- 학습의 어려움(어려운 정도)를 문장의 길이로 가정함.
- 처음에는 짧은 발화부터 발화의 길이 점차 증가하여 학습시킴.
- first training epoch 이후에는 랜덤으로 섞어서 학습함.

=〉 훈련 시 안정성이 높아졌다.

3. DeepSpeech2

③ Comparison of simple RNNs VS GRUs

- 기본 RNN을 사용한 모델이지만, 이 네트워크를 LSTM이나 GRU로 사용하면 성능 향상을 꾀할 수 있을 것.
- 작은 데이터셋으로 LSTM과 GRU 테스트 실험 -> 둘의 정확도는 비슷했으나, GRU가 훈련 속도가 더 빠름.

측정 지표: WER
Word Error Rate

Is there anyone there?
-> Is there neone there?
4개의 word 중에서 1개 틀림
WER = 25%

Architecture	Simple RNN	GRU
5 layers, 1 Recurrent	14.40	10.53
5 layers, 3 Recurrent	10.56	8.00
7 layers, 5 Recurrent	9.78	7.79
9 layers, 7 Recurrent	9.52	8.19

tinyset

Model size	Model type	Regular Dev	Noisy Dev
18×10^6	GRU	10.59	21.38
38×10^6	GRU	9.06	17.07
70×10^6	GRU	8.54	15.98
70×10^6	RNN	8.44	15.09
100×10^6	GRU	7.78	14.17
100×10^6	RNN	7.73	13.06

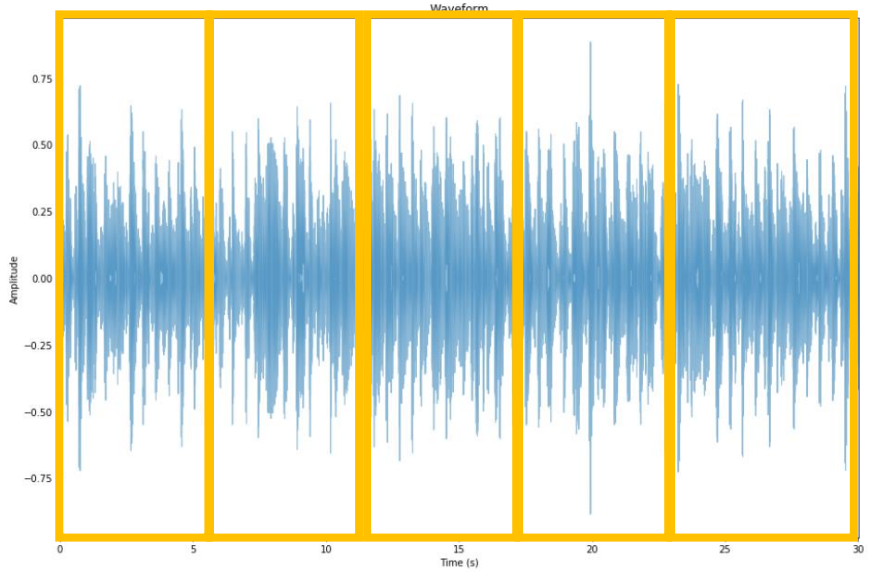
validation set

- GRU는 simple RNN 보다 성능이 우수하지만, 모델 크기를 확장하면 RNN의 성능이 약간 더 좋음.
  따라서 우리 실험에서는 simple RNN 사용.

3. DeepSpeech2

④ Frequency Convolution (1D vs 2D)

- temporal convolution 은 발화 길이가 다양한 음성 데이터의 특성을 효율적으로 모델링하기 위해 음성 인식에 일반적으로 사용된다.



Architecture	Channels	Filter dimension	Stride	Regular Dev	Noisy Dev
1-layer 1D	1280	11	2	9.52	19.36
2-layer 1D	640, 640	5, 5	1, 2	9.67	19.21
3-layer 1D	512, 512, 512	5, 5, 5	1, 1, 2	9.20	20.22
1-layer 2D	32	41x11	2x2	8.94	16.22
2-layer 2D	32, 32	41x11, 21x11	2x2, 2x1	9.06	15.71
3-layer 2D	32, 32, 96	41x11, 21x11, 21x11	2x2, 2x1, 2x1	8.61	14.74

Table 4: Comparison of WER for various arrangements of convolutional layers. In all cases, the convolutions are followed by 7 recurrent layers and 1 fully connected layer. For 2D-invariant convolutions the first dimension is frequency and the second dimension is time. All models have BatchNorm, SortaGrad, and 35 million parameters.

1개의 1D-invariant convolution (1차원의 불변 합성곱 레이어) 보다 3개의 2D-invariant convolution (2차원의 불변 합성곱 레이어) 일 때 성능이 더 좋았다.

4. Results

DATASET scale

Fraction of Data	Hours	Regular Dev	Noisy Dev
1%	120	29.23	50.97
10%	1200	13.80	22.99
20%	2400	11.65	20.41
50%	6000	9.51	15.90
100%	12000	8.46	13.59

데이터셋 크기 증가에 대한 영어데이터셋의 WER 비교
- 데이터셋이 클수록 WER이 낮아진다 == 성능이 좋아진다.

Model Size

Model size	Model type	Regular Dev	Noisy Dev
18×10^6	GRU	10.59	21.38
38×10^6	GRU	9.06	17.07
70×10^6	GRU	8.54	15.98
70×10^6	RNN	8.44	15.09
100×10^6	GRU	7.78	14.17
100×10^6	RNN	7.73	13.06

Model size가 클수록 성능이 좋아지고, 작은 차이지만,
GRU보다 RNN의 성능이 더 좋다.

DeepSpeech1 vs DeepSpeech2

Test set	DS1	DS2
Baidu Test	24.01	13.59

DeepSpeech1 보다 성능을 향상시킴.

4. Results - Benchmarks

- 월스트리트저널(WSJ)의 뉴스 기사 : 뉴스 기사
- LibriSpeech : 책 (오디오북)
- VoxForge : 1명의 화자의 speech 데이터
- CHiME: CHiME-3 세션당 4명의 참가자의 생활 대화

Read Speech			
Test set	DS1	DS2	Human
WSJ eval'92	4.94	3.60	5.03
WSJ eval'93	6.94	4.98	8.08
LibriSpeech test-clean	7.89	5.33	5.83
LibriSpeech test-other	21.74	13.25	12.69

Accented Speech			
Test set	DS1	DS2	Human
VoxForge American-Canadian	15.01	7.55	4.85
VoxForge Commonwealth	28.46	13.56	8.15
VoxForge European	31.20	17.55	12.76
VoxForge Indian	45.35	22.44	22.15

나라별 억양에 따른 성능 비교

Noisy Speech			
Test set	DS1	DS2	Human
CHiME eval clean	6.30	3.34	3.46
CHiME eval real	67.94	21.79	11.84
CHiME eval sim	80.27	45.05	31.33

노이즈 제거

실제 노이즈

노이즈 합성

5. Conclusion

Deep Speech2를 읽으면서 집중한 부분

① input 과 output 은 어떤 데이터인가? 그 모양(shape)은?

input=발화데이터, output=transcription(텍스트), 벡터로 구성됨.

② 네트워크는 어떻게 구성되어 있는가? Frequency CNN -> RNN -> FCN -> CTC loss

ㄴ 그 네트워크를 사용한 이유는? CNN: 음향데이터 처리를 위해. RNN: 음성데이터는 시계열데이터라서.

③ loss function 은 무엇인가? CTC loss

ㄴ 그 함수를 사용한 이유는? input 과 output 의 alignment 를 맞추기 위해서

④ 성능 향상을 위해 사용한 테크닉은? Sequency-wise Normalization, SortaGrad, CNN 변경, RNN 변경

⑤ 벤치마크 성능 측정은 어떻게 하였는가?

ㄴ 어떤 벤치마크 데이터셋을 사용했는가? WSJ, LibriSpeech, VoxForge, CHiME

ㄴ 성능 측정 지표는? WER(Word Error Rate)