NPEX 2022 Deep learning for Speech

Day 2 Practice:
DeepSpeech2: CTC-based AM

2022-06-23

DeepSpeech2 (DS2, 2015)

Deep Speech 2: End-to-End Speech Recognition in English and Mandarin

Baidu Research – Silicon Valley AI Lab*

- 이후 모델들의 디자인 가이드라인
- GPU 고속 병렬처리를 기반으로 한 대형 음성인식 모델
  - Batch processing / streaming processing
- (처음으로) 사람보다 높은 음성인식 정확도 (영어 & 중국어)
  - 내부적으로 가지고 있는 영어 11,940시간, 중국어 9,400시간 데이터에서 측정
  - 사람 WER 4.0% / 모델 3.7%
- Batch normalization (BN)과 CTC loss를 적극적으로 도입

CTC Review

Alignment problem

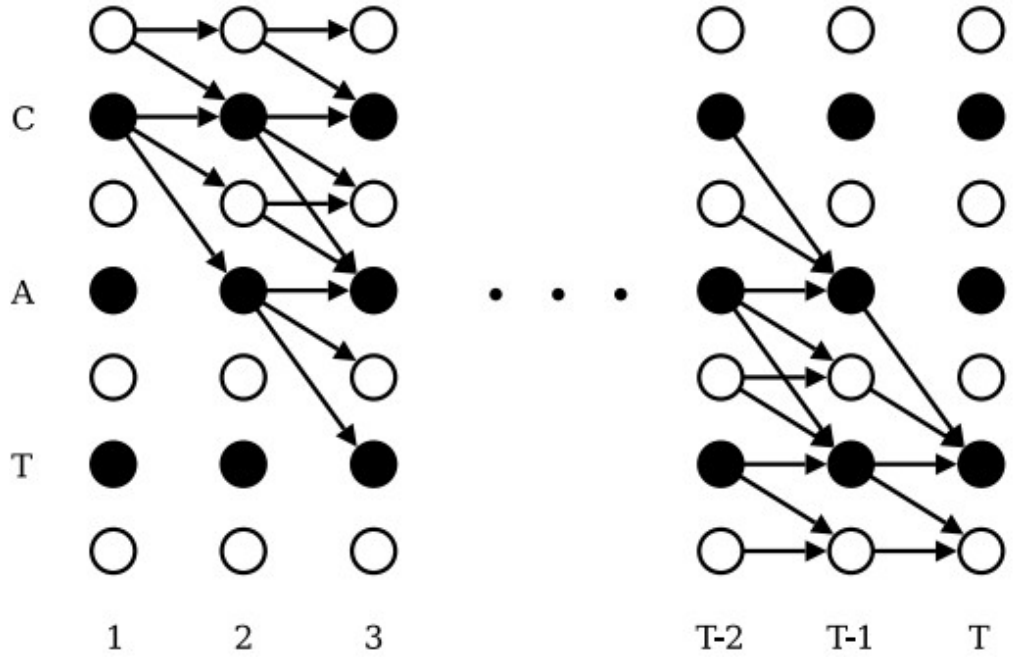
- 음성의 어떤 부분이 텍스트의 어떤 부분인가?
- Frame 개수보다 token 개수가 훨씬 적다 – 1:1 대응이 불가능.

| Waveform | ≫ | Feature | ≫ | Text |

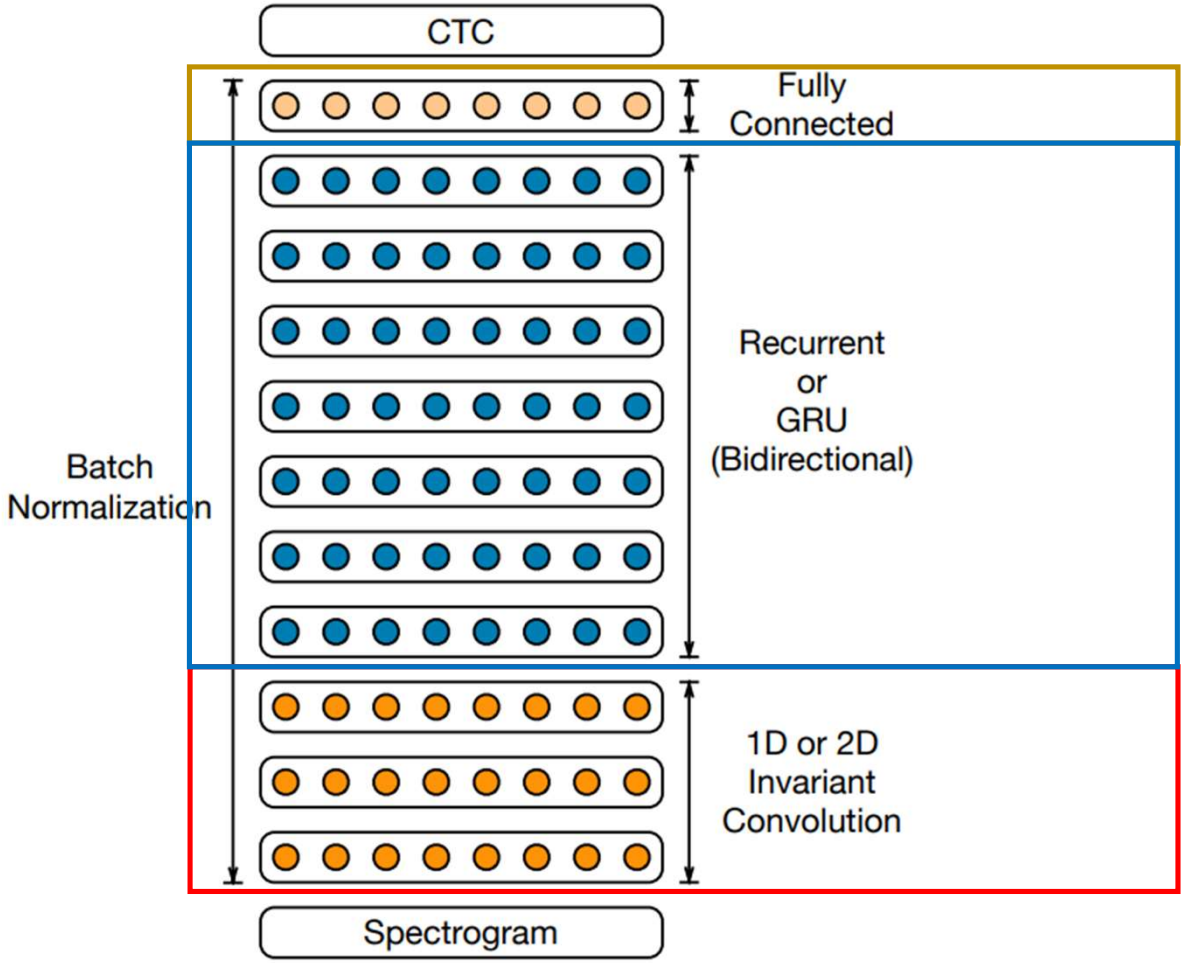
Connectionist Temporal Classification (CTC)

- 각 frame에 맞는 정답 없이도 훈련할 수 있도록
- 음성의 길이(S)에 맞게 텍스트 길이(T) 확장

: Blank token <B>



DS2 Acoustic Model (AM)



CTC Head

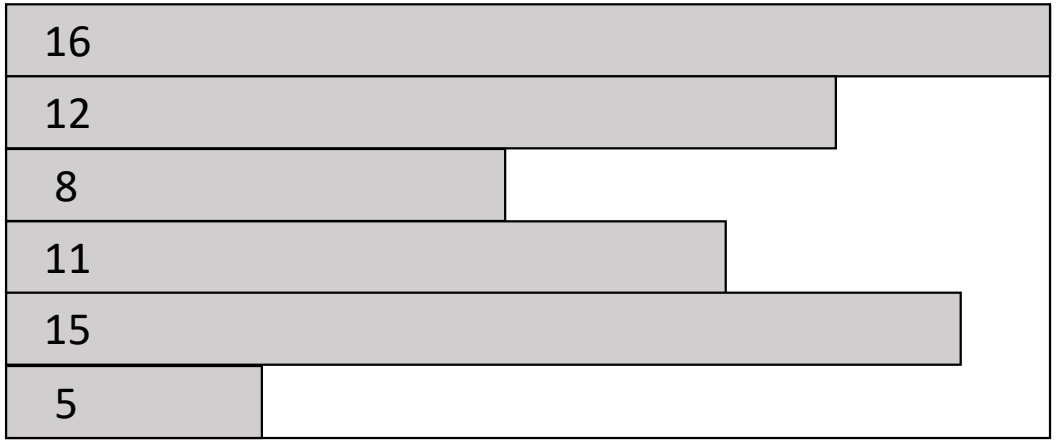
Bi-directional Recurrent Neural Network (GRU / LSTM)

Convolution Subsampling

Batch Processing

Multiple samples (=audio) in one mini-batch

- 음성의 길이가 전부 다름
- 가장 긴 길이에 맞추고 나머지는 0으로 채움
- 매 연산이 끝날 때마다 0으로 잘 채워져 있도록 해야 함
  - Batch 내 다른 음성들의 길이에 영향을 받지 않도록!



  - 빈 칸을 연산해야 하는 비효율을 해결하기 위한 방법이 여러가지 있음
  - Bucketing, sequence packing, sample-by-sample computation, …

Convolution Subsampling

Spectrogram as 2D Image

- Input: (#Batch, 1, #Time, #Freq)
- Output: (#Batch, #Ch, #Time / 4, #Freq / 2)
  - Time 방향으로 2번의 stride x 2 convolution
  - Feature 방향으로 1번의 stride x 1 convolution
  - 연산이 다 끝난 후 frame마다 갖고 있는 정보 = #Ch x #Freq / 2
  - 최근에는 Time과 Feature 방향 모두 2번의 stride를 하는 게 일반적

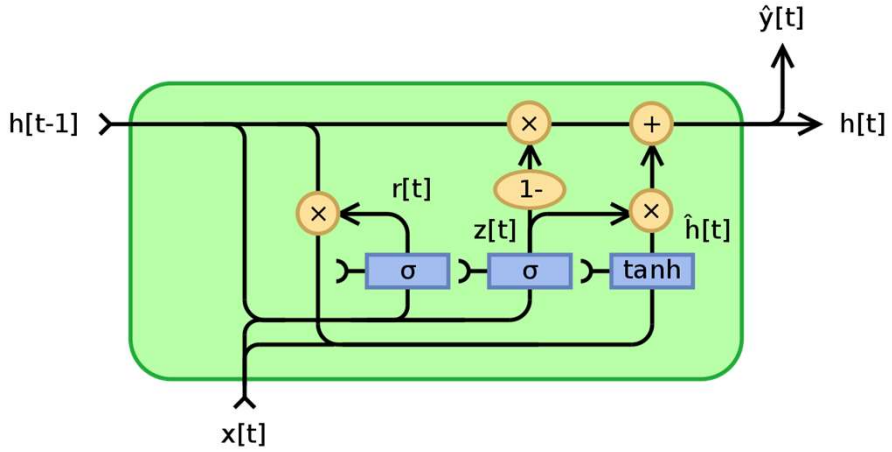
- Conv-BN-ReLU

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

Bidirectional GRU

Gated Recurrent Unit (GRU)

- Bidirectional
  - 음성을 끝까지 다 듣고 텍스트를 만듦
  - 미래 정보가 큰 도움이 됨
  - t번째 output은 [1 ~ t] 정보와 [T ~ t] 정보를 더한 것. (양쪽으로 모든 frame을 보게 됨)
- LSTM과 큰 성능 차이가 없었다고 함.



BN + GRU

- BN으로 신호의 범위를 정규화 시키는 것이 도움이 됨.
  - 최근에는 BN 대신 LN을 거의 언제나 채용
  - LN은 batch-independent 하기 때문
- 이 때, BN의 mean / variance 는 sequence 전체에 대해 계산.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$\tilde{h}_t = f(W_h x_t + r_t \circ U_h h_{t-1} + b_h)$$
$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t$$

Projection head and CTC Loss

CTC Head

- Vocabulary size로 변환 (English character 의 경우 28 + 1 = 29)

  : Blank를 포함해야 함!

- 1-layer FC로 구현

CTC loss

- 가능한 모든 경로의 합을 구하는 작업
  - Forward-backward algorithm (dynamic programming) 으로 최대한 효과적으로 구현해야 함
  - 실제로는 numerical stability를 위해 log-domain에서 작업 (log-softmax)

$$P(Y|X) = \sum_{A \in A_{X,Y}} \left(\prod_{t=1}^{T} p_t(a_t|X) \right)$$

- GPU kernel이 이미 만들어져 있음 (Pytorch, Tensorflow)

Thank You!

NPEX 2022 Deep learning for Speech