Attention-Based Models for Speech Recognition

Winter Vacation Capstone Study

TEAM Kai.Lib

발표자 : 김수환

2020.02.17 (MON)

Introduction

- Introduction

Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

Reference 1,000회 이상…
LAS가 250회…

본 논문에서는 어텐션이 음성 인식 분야에서도 어느 정도의 성능을 보였지만, NMT를 위해 도입된 개념인만큼,
음성 인식이란 특성을 충분히 반영하지는 못했다고 주장한다.
NMT와 같이 단어 단위로 들어가는 상대적으로 짧은 시퀀스 길이에 비해, 20~40ms로 프레임을 짤라서
수백 ~ 수천의 시퀀스 길이를 가지는 음성 인식에 맞는 어텐션을 제안한다.

Introduction

- Introduction

Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

Listen, Attend and Spell

William Chan
Carnegie Mellon University
williamchan@cmu.edu

Navdeep Jaitly, Quoc V. Le, Oriol Vinyals
Google Brain
{ndjaitly,qvl,vinyals}@google.com

당시 두 논문은 음성 인식 분야에서 "혁명"이라고 불릴 정도의 큰 파장이였음.

기존 CTC 방식에서 End-to-End로 넘어가는 "기준"이 된 2 논문

General Framework

- **어텐션의 개념**

$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1}, h)$$

$$g_i = \sum_{j=1}^{L} \alpha_{i,j} h_j$$

$$y_i \sim Generate(s_{i-1}, g_i),$$

기본적인 어텐션에 대한 큰 그림이다.

일반적으로 α는 alignment, g는 glimpse라고 부른다.

α는 기존의 어텐션 분포, g는 각각의 어텐션 스코어와 인코더의 아웃풋을 곱한 어텐션 값이다.

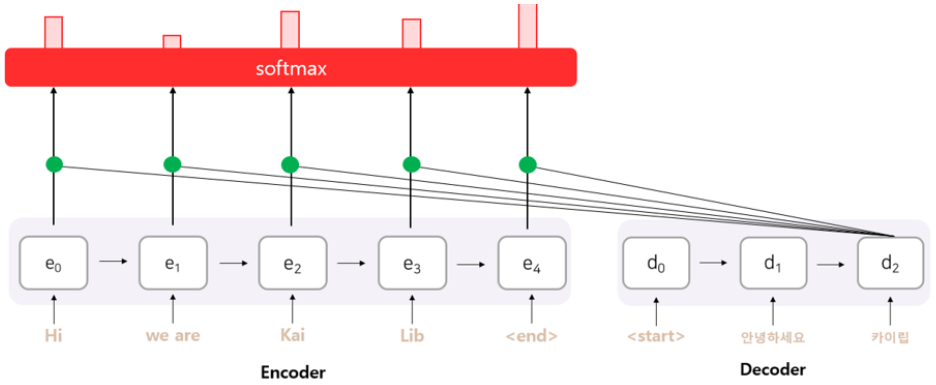
그렇게 구한 g와 s_{i-1} (디코더의 아웃풋) 을 기반으로 예측을 진행한다.

General Framework

- 어텐션의 개념

$$e_{i,j} = Score(s_{i-1}, h_j),$$

$$\alpha_{i,j} = \exp(e_{i,j}) \Big/ \sum_{j=1}^{L} \exp(e_{i,j}) \ .$$



이제 alignment를 어떻게 구하는지를 수식화 한 것이다.

특정 방식으로 Score를 구하고, 해당 스코어를 Softmax 함수에 넣어 alignment를 구한다.

General Framework

- **어텐션의 종류**

이름	스코어 함수
dot	$score(s_t,\ h_i) = s_t^T h_i$
$scaled\ dot$	$score(s_t,\ h_i) = \frac{s_t^T h_i}{\sqrt{n}}$
$general$	$score(s_t,\ h_i) = s_t^T W_a h_i$ // 단, W_a는 학습 가능한 가중치 행렬
$concat$	$score(s_t,\ h_i) = v_a^T\ tanh(W_a[s_t; h_i])$
$location - base$	$\alpha_t = softmax(W_a s_t)$ // α_t 산출 시에 s_t만 사용하는 방법.

이전 스터디인 'Attention Mechanism'에서도 살펴봤지만, 어텐션의 종류는 이 "Score"를 어떻게 구하느냐에 따른 차이이다.

General Framework

- **Content-Based attention**

Content-Based attention

$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + b).$$

e : attention score

s_{i-1} : decoder output

h : encoder outputs

b : bias

W, V, w : weight

본 논문에서는 위의 2가지 어텐션 방법에 주목했다.
Content-Based 방법은 현재 스텝의 디코더의 출력과 모든 인코더의 아웃풋에 웨이트와 편향을 주고,
tanh와 해당 결과에 다시 웨이트를 주는 방법이다.

Dot-Product 방법보다 조금 더 복잡한 수식으로 인코더와 디코더의 아웃풋을 고려해주었다고 볼 수 있다.
하지만, Content-Based의 문제는 시퀀스에서의 자신의 위치에 상관없이 스코어링을 한다는 점이다.
이를 "similar speech fragments" 문제라고 한다.

General Framework

- **Location-Based attention**

Location-Based attention

$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1})$$

α_{i-1} : last alignment

s_{i-1} : decoder output

이러한 문제의 대안으로 location-based 방식이 있다.
alignment 계산시, 디코더의 아웃풋과, 이전 alignment를 고려해줌으로써,
현재 시퀀스에서 어느 위치인지를 알 수 있게끔 해주는 방식이다.

하지만 이 방식은 인코더의 아웃풋을 고려하지 않은 채,
디코더의 아웃풋만을 가지고 예측하기 때문에 분명한 한계점이 존재한다.

General Framework

- **새로운 방식의 어텐션**

Content-Based attention

$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + b).$$

+

Location-Based attention

$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1})$$

본 논문은 이러한 2 방식의 어텐션을 적절히 결합해서
음성 인식용 어텐션을 제안한다.

해당 어텐션을 **Hybrid**, **Location-Aware**, **Location-Sensitive** 등 여러 이름으로 불린다.
(본 논문에서 Hybrid, Location-Aware라는 2 용어 모두 사용)

General Framework

- **Hybrid attention**

Hybrid attention

$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + U f_{i,j} + b)$$

$$f_i = F * \alpha_{i-1}.$$

기존 content-based 방식에서 이전 스텝의 alignment를 고려해준다.

이때, 이전 alignment에 Convolution으로 1xC => KxC의 행렬 형태로 바꿔준다.

(C : classification number)

그리고 해당 행렬에 웨이트를 주어서 content + location 방식을 완성한다.

Three Potentioal Issue

- **Hybrid attention**

$$\alpha_{i,j} = \exp(e_{i,j}) \Big/ \sum_{j=1}^{L} \exp(e_{i,j})$$

앞에서 살펴봤던 위의 수식에는 .3가지 이슈가 있다.

1. 인풋 시퀀스가 길다면, glimpse에는 노이즈가 섞여있을 가능성이 크다.

2. 시간 복잡도가 크다.

3. Softmax 함수는 Single Vector에만 집중하는 경향이 있다

Three Potentioal Issue

- **Issue 1**

1. 인풋 시퀀스가 길다면, glimpse에는 노이즈가 섞여있을 가능성이 크다.

만약 인풋 시퀀스가 길다면, 어떤 시점 t에서 멀리 떨어져 있는 t + k라는 시점에서의 음성과는 서로 관련이 없을 것이다. 하지만 Softmax 함수 특성상, 모든 인풋들에 값을 부여한다. 이러한 Softmax의 특성에 의해 많은 관련없는(irrelevant) 인코더의 출력들이 고려될 것이다. 이는 Noise로 작용된다.

Three Potentioal Issue

- **Issue 2**

2. 시간 복잡도가 크다.

O(LT)

인풋 시퀀스의 길이가 L이라고 할 때, 디코더는 매 타임 스텝마다 이 L개의 frame을 고려해주어야 한다. 그리고, 디코딩 길이를 T라 할 때, 위의 과정을 T만큼 반복하게 된다. 이는 O(LT) 라는 높은 시간 복잡도를 만들게 된다.

Three Potentioal Issue

- **Issue 3**

3. Softmax 함수는 Single Vector에만 집중 (focus) 하는 경향이 있다.

이러한 경향은 top-score를 받은 여러 프레임을 고려할 수 없게 한다.

Three Potentioal Issue

Sharpening & Windowing

본 논문은 위의 문제를 간단하게 해결하기 위해 "Sharpening"이라는 개념의 제안했다. Softmax 수식을 약간 수정하는 것이다.

$$a_{i,j} = \exp(\beta e_{i,j}) \Big/ \sum_{j=1}^{L} \exp(\beta e_{i,j})$$

when, β > 1

본 논문에서는 inverse temperature를 걸어준다고 표현했다.
위의 수식이 왜 1번 문제를 해결해 주는지에 대해서는 아직 이해를 하지 못하였다.

그리고 본 논문은 위의 방식이거나, top-k개의 프레임만을 뽑아서 re-normalization을 해주는 방식으로도 해결 가능하다고 말한다.
하지만, 위의 2 방식 모두 2번째 시간복잡도의 문제는 해결하지 못했으며, 2번째 방법의 경우는 오히려 시간 복잡도를 더 늘리게 된다.

그리고 Windowing이라는 방법이 나오게 되는데, 이전 alignment의 중간값(median)을 기준으로 윈도우 크기 만큼만 고려해주는 방식이다. 해당 방법은 O(L+T)로 시간 복잡도를 낮춰준다.

Three Potentioal Issue

Sharpening은 long-utterance (긴 발화)에서의 퍼포먼스는 개선했지만, 전체적인 퍼포먼스면에서는 좋지 못한 결과로 이어졌다.
(짧은 발화에서는 퍼포먼스가 별로였다)
하지만 해당 실험은 최상위 점수를 받은 프레임들을 선택하여 집계하는 방식이 좋을 것이라는 가정을 하도록 만들었다고 한다.

General Framework

Smoothing

그래서 나오게 된 방법이 Smoothing 방법이다.

$$a_{i,j} = \sigma(e_{i,j}) \Big/ \sum_{j=1}^{L} \sigma(e_{i,j})$$

위의 식처럼 기존 Softmax 식에 Sigmoid를 추가해준 방식이다.
이러한 방식은 다양성을 가져온다고 본 논문은 말한다.

General Framework

- **Result**

Model	Dev	Test
Baseline Model	15.9%	18.7%
Baseline + Conv. Features	16.1%	18.0%
Baseline + Conv. Features + Smooth Focus	15.8%	**17.6%**

Bseline Model : Content-Based attention

Baseline + Conv. Features : Hybrid attention

Baseline + Conv. Features : Smooth Focus : Hybrid + Smoothing

Reference

Paper Review

https://github.com/sh951011/Paper-Review/blob/master/Attention-Based%20Models%20for%20Speech%20Recognition.md

Implementation

https://github.com/sh951011/Korean-Speech-Recognition/blob/master/models/attention.py