The Korean Journal of Applied Statistics (2019) DOI: https://doi.org/10.5351/KJAS.2019.32.2.213
32(2), 213–227

Korean speech recognition using deep learning

Suji Lee[a] · Seokjin Han[a] · Sewon Park[a,1] · Kyeongwon Lee[a] · Jaeyong Lee[a]

[a]Department of Statistics, Seoul National University

(Received December 21, 2018; Revised February 13, 2019; Accepted February 14, 2019)

Abstract

In this paper, we propose an end-to-end deep learning model combining Bayesian neural network with Korean speech recognition. In the past, Korean speech recognition was a complicated task due to the excessive parameters of many intermediate steps and needs for Korean expertise knowledge. Fortunately, Korean speech recognition becomes manageable with the aid of recent breakthroughs in "End-to-end" model. The end-to-end model decodes mel-frequency cepstral coefficients directly as text without any intermediate processes. Especially, Connectionist Temporal Classification loss and Attention based model are a kind of the end-to-end. In addition, we combine Bayesian neural network to implement the end-to-end model and obtain Monte Carlo estimates. Finally, we carry out our experiments on the "WorimalSam" online dictionary dataset. We obtain 4.58% Word Error Rate showing improved results compared to Google and Naver API.

Keywords: Korean speech recognition, end to end deep learning, Connectionist temporal classification, Attention, Bayesian deep learning

1. 서론

음성인식은 음성 신호(speech signal)의 특징을 추출하여 분석하고, 이를 단어나 문장으로 변환하는 일련의 처리 과정을 의미한다. 음성인식 기반의 인터페이스는 텍스트 혹은 그래픽 기반의 인터페이스보다 의사 교환이 자연스럽고 시각적 제약으로부터 자유로워 접근성이 우수하다는 장점이 있다. 기존의 인간-기계의 인터페이스 방식이 텍스트, 그래픽 기반이었다면 최근 향상된 음성인식 기술은 그 방식을 음성 기반으로 확장시켰다. 이러한 장점에 의해 현재 여러 산업에서 음성 기반의 인터페이스를 활용하고 있다. 예컨대 구글의 'Google Assistant', 애플의 'Siri', 삼성의 'Bixby'와 같은 음성인식 비서 플랫폼들이 출시되었으며, 카카오의 '미니'나 SKT의 '누구' 등 생활 속 다양한 방면에 음성인식 기술을 접목한 상품들이 개발되고 있다. 이와 같은 흐름에서 더 나은 성능을 갖는 음성인식 기술은 필요성이 높아지고 있다.

기존의 음성인식 모형은 주로 음소 단위의 음향모형(acoustic model)과 언어모형(language model)으로 구성된다. 여기서 음향모형은 음성 신호를 음소나 유사 음소로 변환하는 작업을 의미하며 변환된 음소들은 렉시콘(lexicon)과 언어모형에 의해 문법, 문맥에 맞는 언어로 재구성된다. 렉시콘이란 음소와

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1A2A3074973).
[1]Corresponding author: Department of Statistics, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea. E-mail: swpark0413@gmail.com

이에 대응하는 단어를 기록한 사전으로서 음소 단위로 훈련되는 음향모형의 후처리를 위해 사용된다. 예를 들어, 음소에서 문장으로의 언어적 후처리 과정에서 N-gram과 같은 별도의 모형을 필요로 하고 (Jelinek, 1997), 은닉 마르코프(hidden markov) 모형을 기반으로 한 음향 모형은 각 음성에 대응하는 음소 단위의 학습 과정을 요구하였다 (Gales와 Young, 2008). 이처럼 기존의 음성인식 방식은 음성신호에서 음소, 단어, 문장 순의 복잡한 변환 과정이 요구되며 음소를 비롯한 언어적 특성에 대해 많은 사전 지식을 알아야 하는 번거로움이 있었다.

더욱이 한국어는 교착어라는 특성 때문에 다른 언어에 비해 음성 인식 모형 구현이 더 어렵다. 예컨대 '쓰-'라는 같은 어근을 갖더라도 접사에 의해 '쓰다', '씁니다', '썼습니다'의 여러 고유 단어가 존재하고, 음소의 표기도 sseu, sseub, ssoss 등으로 다양하다. 이처럼 고유 단어 수가 무수히 많은 교착어의 경우 단어 단위로 모형이 잘 훈련되지 않아 Kwon과 Park (2003)과 같이 형태소 단위로 모형을 훈련하는 방법이 제안되었다. 형태소란 뜻을 가진 가장 작은 말의 단위로서, 형태소 단위로 훈련할 경우 필요로 하는 단어의 개수가 줄어 효과적이다. 그러나 이러한 접근은 상당한 수준의 언어학적 지식이 요구되며 음소 발음과 형태소라는 중간 매개가 필요하다는 어려움이 존재한다.

음성인식에서 최근 주목받고 있는 딥러닝(deep learning)은 자료(입력)에서 목표한 결과(출력)을 별도의 중간 매개 없이 학습하는 종단 간(end-to-end) 학습을 가능케 했다. 종단 간 학습을 이용한 음성인식에서는 주어진 음성을 음소 및 형태소를 거치지 않고 바로 단어나 문장으로 변환할 수 있다. 음성인식의 여러 중간 단계들을 생략하면 음소 단위로 훈련을 할 필요 없고, 음소를 매개로 하는 중간 단계와 렉시콘 사전이 생략할 수 있어 과정이 간소화된다. 대표적인 음성인식의 종단 간 모형으로는 Graves 등 (2006)이 제안한 연결성 시계열 분류기(connectionist temporal classification; CTC) 모형과 Chan 등 (2015)가 제안한 listen, attend, and spell (LAS) 모형이 있다. LAS는 Seq2Seq 모형의 변형인 주의 기제 모형을 음성인식에 적용한 대표적인 모형이다. Kim 등 (2017)은 앞선 CTC 모형과 주의 기제(attention mechanism) 모형을 결합한 모형을 제안하였다.

본 논문에서는 기존보다 우수한 한국어 음성 인식 모형을 구현하고자 기존의 종단 간 모형에 베이즈 딥러닝을 결합한 새로운 모형을 제안하였다. 베이즈 딥러닝을 모형에 도입함으로써 몬테카를로(Monte Carlo) 추정치를 도출하고, 점 추정치로 예측하던 기존 모형의 한계점을 보완하여 확률적 예측이 가능한 음성 인식 모형을 구현하였다. 또한, 제안한 모형이 기존의 Application Programming Interface (API)와 비교했을 때 유의미한 성능 개선을 보이는지 확인하였다. 한국어 음성인식에 베이즈 딥러닝 모형을 적용한 것은 본 논문이 처음이다.

논문의 구성은 다음과 같다. 2장에서는 음성인식에서 주로 사용되는 딥러닝 모형을 살펴본다. 3장에서는 기존의 종단 간 학습 방식을 변형하여 베이즈 딥러닝을 적용한 음성인식 모형을 제안한다. 4장에서는 실제 한국어 음성자료를 이용한 모의실험을 다루며, 5장에 그 결과를 정리한다. 6장에서는 본 논문의 결론과 한계점을 논의한다.

2. 음성인식에서의 딥러닝

이 장에서는 음성인식에 주로 사용되는 딥러닝 모형을 살펴본다. 2.1절에서는 CTC 모형을 살펴보고, 2.2절에서는 Seq2Seq 모형과 그 변형 중 하나인 주의 기제를 살펴본다. 뒤이어 CTC 모형과 주의 기제를 결합한 형태에 대해 2.3절에서 간단히 알아본다. 마지막으로 모형의 학습과 별개로 음성 자료를 문자열로 바꾸는 디코딩에 이용되는 빔 탐색(beam search) 및 언어 모형(language model)에 관해 2.4절에서 살펴본다.

최신 음성인식 모형들은 순환신경망(recurrent neural network; RNN)을 기반으로 개발되고 있다.

순환신경망은 은닉층이 재귀적인 구조를 갖는 인공신경망이며 음성인식, 번역, 이미지 주석 생성 등 순차적인 자료 처리에 적합하다. 그러나 기존 순환 신경망은 시간이 길어질수록 경사도 소실 문제(vanishing gradient problem)가 발생하기 때문에 장기 의존성(long-term dependency)을 학습시킬 수 없는 한계점이 존재한다 (Bengio 등, 1994). 이러한 장기 의존성 문제를 해결하기 위해 등장한 모형이 long short-term memory (LSTM)이다 (Hochreiter와 Schmidhuber, 1997). LSTM은 기존의 뉴런을 3개의 게이트(입력/출력/망각)를 도입한 LSTM 블록으로 대체하여 장기 의존성을 학습 가능하게 해준다. 또한, Cho 등 (2014)은 LSTM의 변형인 gate recurrent unit (GRU)를 제안하였다. GRU는 일부 게이트를 생략하고 단순하게 리셋 게이트와 갱신 게이트 만으로 구성된 모형으로 LSTM보다 나은 성능을 보이며 계산시간이 적게 걸리는 장점이 있다 (Chung 등, 2014). 순환신경망과 LSTM의 보다 자세한 내용은 Goodfellow 등 (2016)을 참고하길 바란다.

딥러닝을 기반으로 한 음성 인식 모형은 음성 자료를 잠재 변수로 변환하는 인코더(encoder)와 잠재 변수로부터 문자열을 얻어내는 디코더(decoder)로 구성되어 있다. 일반적으로 음성 자료를 10–20ms 단위의 프레임(frame)으로 나누는데, 하나의 음성 자료가 가지는 프레임의 개수를 T라고 표기하자. 이 때 각 프레임이 속하는 공간을 $\mathcal{X} = \mathbb{R}^d$라 하면 음성 자료는 $\mathbf{x} = (x_1, \ldots, x_T) \in \mathcal{X}^T$로 나타낼 수 있다.

음성 자료에 대응되는 문자열을 $\mathbf{y} = (y_1, \ldots, y_L) \in \mathcal{Y}^L$로 나타내는데, 여기서 L은 문자열의 길이이며 $\mathcal{Y}$는 문자 집합, 혹은 알파벳 집합으로 $|\mathcal{Y}| < \infty$를 만족한다. 추가로 한글에 포함되지 않는 특수문자들을 포함하는 문자 집합 $\mathcal{Y}_* = \mathcal{Y} \cup \{\langle \text{sos} \rangle, \langle \text{eos} \rangle, \text{“_”}, \ldots\}$을 정의한다. 특수문자의 예로는 문장의 시작과 끝을 의미하는 sos (start of sentence), eos (end of sentence)나 아무 의미를 가지지 않는 문자인 _ 등이 있다.

모형에서 사용되는 잠재 변수는 프레임에 대응되는 $\mathbf{z} = (z_1, \ldots, z_T) \in \mathcal{Z}^T$ ($\mathcal{Z} = \mathbb{R}^r$)와 프레임에 대응되지 않는 $h \in \mathcal{H}$ ($\mathcal{H} = \mathbb{R}^s$)가 있다. 집합열 $\{A^i\}$에 대해, $A^{\leq I} = \bigcup_{i=1}^{I} A^i$와 $A^{\infty} = \bigcup_{i=1}^{\infty} A^i$로 정의하고 $\mathfrak{M}(A)$는 A 위에 정의되는 모든 확률측도의 집합으로 정의한다. 또한 모든 수열 $\mathbf{v} = (v_1, \ldots, v_I)$에 대해 $\mathbf{v}_i = (v_1, \ldots, v_i)$ ($i \leq I$)와 $\text{softmax}(\mathbf{v}) = (e^{v_1}/\sum_{i=1}^{I} e^{v_i}, \ldots, e^{v_I}/\sum_{i=1}^{I} e^{v_i})$를 정의하자. 모형의 모수는 $\theta \in \Theta$로 표기한다.

2.1. 연결성 시계열 분류기

음성인식의 어려움 중 하나는 $\mathbf{x}$의 길이 T와 $\mathbf{y}$의 길이 L이 자료마다 일정하지 않고 다르다는 것이다. 또한 x_i가 y_j에 대응된다고 할 때 i와 j의 관계 역시 자료마다 달라지며 심지어 명확히 주어지지 않는다. Graves 등 (2006)이 제시한 CTC 모형은 문자 “_”를 포함하며 길이가 T로 고정된 중간 단계 문자열 $\mathbf{y}_* = (y_{*1}, \ldots, y_{*T}) \in \mathcal{Y}_*^T$의 확률을 계산하는 인코더를 만들고, $\mathbf{y}_*$로부터 실제 문자열 $\mathbf{y} \in \mathcal{Y}^{\leq T}$를 추출하는 디코더를 만들어 이를 해결하였다.

$\mathcal{Y}_* = \mathcal{Y} \cup \{\text{_}\}$로 하고 $r = |\mathcal{Y}_*|$로 놓자. 우선 인코더 모형 $\text{Enc}_\eta : \mathcal{X}^T \to \mathcal{Z}^T$를 만들고 $\mathbf{z} = \text{Enc}_\eta(\mathbf{x})$로 하자. 여기서 z_t는 t번째 음성 프레임 x_t가 주어졌을 때 대응되는 길이 $|\mathcal{Y}_*|$의 벡터로 각 원소는 y_{*t} 문자들의 가능성을 의미하며, η는 인코더를 구성하고 있는 모든 모수들을 의미한다. t 시점에서 문자의 확률은 $q_t := \text{softmax}(z_t) \in \mathfrak{M}(\mathcal{Y}_*)$로 추정한다. 이를 이용하여 $\mathbf{x}$가 주어졌을 때 중간 문자열 $\mathbf{y}_*$의 확률은 다음과 같이 추정한다:

$$q(\mathbf{y}_*|\mathbf{x}, \eta) := \prod_{t=1}^{T} q_t(y_{*t}), \quad \mathbf{y}_* \in \mathcal{Y}_*^T.$$

이제 함수 $\gamma : \mathcal{Y}_*^T \to \mathcal{Y}^{\leq T}$를 서로 이웃한 같은 문자열을 합친 후 모든 “_”문자를 지우는 함수로 정의하자. 예를 들어, $HHH_E__LL_LO___ \to H_E_L_LO_ \to HELLO$이므로 $\gamma(HHH\_E\_\_LL\_LO\_\_\_) =$

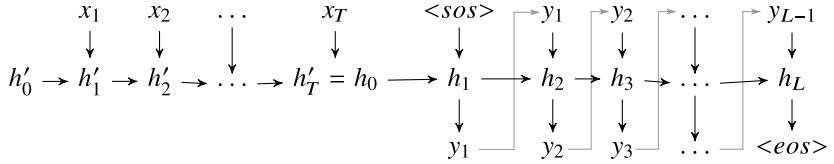

Figure 2.1. Sequence to sequence model.

$HELLO$이다. 위 모형에서 $\mathbf{x}$가 주어졌을 때 $\mathbf{y}$의 예측값 $\hat{\mathbf{y}}$는 다음과 같이 구한다:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\mathcal{Y}^{\le T}}{\arg\max}\, p(\mathbf{y}|\mathbf{x},\eta) = \underset{\mathbf{y}\in\mathcal{Y}^{\le T}}{\arg\min}\left[-\log p(\mathbf{y}|\mathbf{x},\eta)\right], \quad \text{여기서 } p(\mathbf{y}|\mathbf{x},\eta) := \sum_{\mathbf{y}_*\in\gamma^{-1}(\mathbf{y})} q(\mathbf{y}_*|\mathbf{x},\eta).$$

2.2. Seq2Seq 및 주의 기제

Cho 등 (2014)가 제시한 Seq2Seq 모형은 주어진 배열을 다른 배열로 변환하는 딥러닝 모형 중 하나로, 기계번역에 주로 사용된다. $\mathcal{Y}_* = \mathcal{Y}\cup\{\langle\text{sos}\rangle, \langle\text{eos}\rangle\}$일 때, Seq2Seq 모형은 입력을 받아서 잠재변수를 생성하는 인코더 $\text{Enc}_\eta : \mathcal{X}^T \to \mathcal{Z}^T \times \mathcal{H}$, 잠재변수와 모수 ϕ로부터 문자열의 확률 측도를 생성하는 디코더 $\text{Dec}_\phi : \mathcal{Z}^T \times \mathcal{H} \to \mathfrak{M}(\mathcal{Y}_*^\infty)$로 구성된다. 인코더와 디코더는 모두 순환신경망을 기반으로 한다.

$\text{Enc}_\eta(\mathbf{x}) = (\mathbf{z}, h'_T)$에서 $\mathbf{z}$와 h'_T는 다음과 같이 재귀적으로 구한다:

$$h'_0 = \mathbf{0}, \quad h'_t = f'(x_t, h'_{t-1}|\eta_1), \quad (t = 1,\dots,T),$$
$$z_t = g'(h'_t|\eta_2), \quad (t = 1,\dots,T).$$

z_t는 시점 t에서 순환신경망의 출력값이고, h'_t는 시점 t에서의 잠재 변수를 뜻한다. h'_t는 $x_1,\dots,x_t$에 모두 의존하며 t 시점까지의 모든 정보를 담은 잠재 변수가 된다. η_1과 η_2는 각 순환신경망을 구성하고 있는 모든 모수들을 의미하며 $f'(\cdot\,|\eta_1)$과 $g'(\cdot\,|\eta_2)$는 사용하는 순환신경망 모형에 의해 결정된다.

인코더로 구한 $\mathbf{z}$와 h'_T로부터 $\text{Dec}_\phi(\mathbf{z}, h'_T) = p(\cdot\,|\mathbf{x},\theta) \in \mathfrak{M}(\mathcal{Y}_*^\infty)$를 구한다. h_l는 시점 l에서의 잠재 변수이고 $p(\mathbf{y}_*|\mathbf{x},\theta)$는 $\mathbf{x}$와 모수 $\theta = (\eta,\phi)$가 주어졌을 때 $\mathbf{y}_* \in \mathcal{Y}_*^\infty$의 확률을 뜻한다. 인코더와 비슷하게 h_l은 $y_{*1},\dots,y_{*(l-1)}$에 모두 의존하며, 디코더는 $\langle\text{eos}\rangle$가 나오면 디코딩을 종료한다. 이 과정을 식으로 나타내면 다음과 같다:

$$h_0 = h'_T, \quad y_{*0} = \langle\text{sos}\rangle, \quad h_l = f(y_{*(l-1)}, h_{l-1}|\phi) \quad (l = 1,\dots,L),$$
$$p(\cdot\,|\mathbf{x},\theta,\mathbf{y}_{*(l-1)}) = \text{softmax}(g(h_l|\phi)) \quad (l = 1,\dots,L),$$
$$p(\mathbf{y}_*|\mathbf{x},\theta) = \prod_{l=1}^{L} p(y_{*l}|\mathbf{x},\theta,\mathbf{y}_{*(l-1)}).$$

여기서 문자열의 길이, L은 디코딩이 종료된 시점을 의미하며 f, g는 사용하는 순환신경망 모형에 의해 결정된다. 이제 Seq2Seq 모형에서 $\mathbf{y}$의 예측값 $\hat{\mathbf{y}}$은 다음과 같이 구할 수 있다:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\mathcal{Y}^\infty}{\arg\max}\, p((\mathbf{y},\langle\text{eos}\rangle)|\mathbf{x},\theta) = \underset{\mathbf{y}\in\mathcal{Y}^\infty}{\arg\min}\left[-\sum_{l=1}^{L}\log p(y_l|\mathbf{x},\theta,\mathbf{y}_{l-1}) - \log p(\langle\text{eos}\rangle|\mathbf{x},\theta,\mathbf{y})\right].$$

Figure 2.1은 앞서 설명한 seq2seq 모형을 도식화한 것이다.

Seq2Seq 모형의 대표적인 변형 형태 중 하나로 주의 기제 모형이 있다 (Bahdanau 등, 2014; Luong 등, 2015). 기존 Seq2Seq 모형은 $l+1$ 시점에서 문자의 확률을 계산할 때 바로 이전 문자인 y_l과 l 시점에서의 잠재 상태 h_l만 고려한다. 여기서 h_l는 $x_1, \ldots, x_T$의 정보를 모두 포함하는데, 이는 모형이 문자를 예측할 때 모든 음성 자료를 이용한다는 뜻이다. 즉, "통계학"이라는 음성을 한글로 바꾸는 과정에서 "학"이라는 글자를 예측하는데 "ㅌ"이나 "ㄴ" 같이 직접적으로 관련 없는 음성 정보도 이용한다. 주의 기제 모형은 "학"이라는 글자를 예측할 때 "ㅎ", "ㅏ", "ㄱ"과 같이 관련 있는 음성 자료를 모형이 사용하도록 하여 효율을 높이고자 하는데 있다.

이를 위해 주의 기제는 맥락(context) $c_l = \sum_{t=1}^{T} \alpha_{lt} z_t$를 디코더에 추가하였다. 가중치 α_{lt}는 디코더가 l번째 문자를 예측할 때 t 시점의 음성 자료를 어느 정도의 비중으로 고려할 것인가를 의미한다. 가중치는 모두 ξ에 대한 에너지 함수 모형 $\text{Energy}_\xi : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$을 학습시켜서 얻는다. 모수 $\theta = (\eta, \phi, \xi)$에 대하여 주의 기제 모형을 식으로 표현하면 다음과 같다.

$$
\begin{aligned}
& e_{lt} = \text{Energy}_\xi(h_l, z_t) \quad (l = 1, \ldots, L,\ t = 1, \ldots, T), \\
& \alpha_{lt} = \frac{\exp(e_{lt})}{\sum_{t=1}^{T} \exp(e_{lt})} \quad (l = 1, \ldots, L,\ t = 1, \ldots, T), \\
& c_l = \sum_{t=1}^{T} \alpha_{lt} z_t \quad (l = 1, \ldots, L), \\
& h_0 = h'_T, \quad y_{*0} = \langle \text{sos} \rangle, \quad h_l = f(y_{*(l-1)}, h_{l-1}, c_{l-1} | \phi), \quad (l = 1, \ldots, L), \\
& p(\cdot\,|\mathbf{x}, \theta, \mathbf{y}_{*(l-1)}) = \text{softmax}(g(h_l, c_l|\phi)), \\
& p(\mathbf{y}_*|\mathbf{x}, \theta) = \prod_{l=1}^{L} p(y_{*l}|\mathbf{x}, \theta, \mathbf{y}_{*(l-1)}).
\end{aligned}
$$

에너지 함수 모형 Energy_ξ로는 다음과 같은 모형이 제안되었다:

$$
\text{Energy}_\xi(h, z) = \begin{cases} v_\xi^\top \tanh(W_\xi h + W'_\xi z), & \text{(Bahdanau 등, 2014)}, \\ h^\top W_\xi z, & \text{(Luong 등, 2015)}. \end{cases}
$$

여기서 $\xi = (v_\xi, W_\xi, W'_\xi)$ 또는 $\xi = W_\xi$는 가중치 행렬(weight matrix)를 의미한다. Figure 2.2는 주의 기제 모형을 구체적으로 나타낸 것이다.

2.3. Connectionist Temporal Classification과 주의 기제의 결합

주의 기제 모형에서 z_t는 r차원 잠재 변수로 해석된다. Kim 등 (2017)은 z_t를 CTC 모형에서처럼 음성 프레임 x_t이 주어졌을 때 y_{*t} 문자들의 가능성으로 해석을 하여, 디코더를 이용한 예측값과 별개로 새로운 CTC 예측값을 얻도록 서로 다른 두 모형들을 결합하였다. 결합된 모형의 손실함수로는 CTC와 주의 기제의 손실함수의 가중평균 $\mathcal{L}_{\text{CTCAtt}} = \lambda \mathcal{L}_{\text{CTC}} + (1-\lambda)\mathcal{L}_{\text{Att}}$ $(0 < \lambda < 1)$를 이용하는 아이디어를 제안하였다.

2.4. 디코딩에서 빔 탐색과 언어 모형의 적용

일반적으로 디코딩을 할 때는 $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \theta)$를 찾아야 하고, 이를 위해서는 $\mathcal{Y}^{\le T}$ 또는 $\mathcal{Y}^\infty$에 속하는 모든 문자열이 가지는 확률을 계산하여야 한다. 그러나 시간이나 문자열의 길이에 따라서 가능한 문자의 개수가 지수적으로 증가하므로 비효율적이다. 따라서 모든 가능한 문자열을 탐색하는 대신

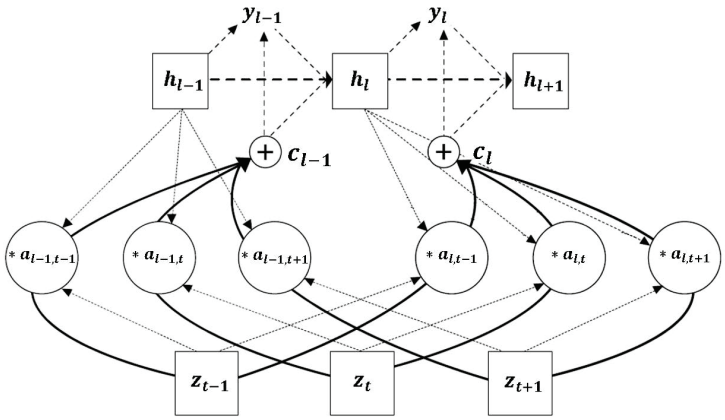

Figure 2.2. Attention model (Bahdanau *et al.*, 2014).

빔 탐색(beam search)을 이용하여 기억해야 하는 노드를 제한하는 기법을 사용하였다. 빔 탐색은 주어진 문자 다음에 선택될 수 있는 문자의 모든 가능한 경우의 수를 계산한 후, 미리 정한 상위 B개 확률의 문자 조합만을 취하여 목표 문자열를 찾을때까지 반복하는 경험적 탐색 알고리듬이다. 여기서, B는 빔 너비(beam width)라고 부른다.

추가로 디코딩 시에 따로 훈련된 언어 모형을 이용하여 정확도를 높일 수 있다. 일반적으로 음성인식 모형에 입력으로 들어오는 음성은 아무 의미 없는 발음들이 아니라 언어적 맥락이나 문법과 같이 음성과 독립적인 언어적 특성이 존재하는 문장이다. 따라서, 이러한 정보를 따로 학습한 뒤에 모형에 추가하여 인식의 정확도를 높일 수 있다.

3. 베이즈 딥러닝

기존의 딥러닝 모형은 점 추정치만 구하기 때문에 모수가 갖는 불확실성에 대한 정보를 얻을 수 없으며 과적합에 취약하다는 단점을 가지고 있다. 이러한 단점을 베이즈 통계 관점으로 극복하고자 한 것이 베이즈 딥러닝 모형이다. 현대적인 베이즈 딥러닝 모형은 Blundell 등 (2015); Gal과 Ghahramani (2016a) 등이 제안했으며, 자료의 개수와 모수의 차원이 모두 크기 때문에 계산 방법으로 변분 베이즈 (variational Bayes; VB)을 쓴다.

변분 베이즈는 비교적 쉽게 다룰 수 있는 확률밀도함수의 모임 $\{q_\omega : \omega \in \Omega\}$로부터 실제 사후분포인 $p(\theta|\mathbf{x}, \mathbf{y})$의 근사치를 뽑아내고, 근사된 확률밀도함수를 사후분포 대신 추론에 사용하는 기법이다. 일반적으로 근사치 q_ω는 쿨백-라이블러(Kullback-Leibler; KL) 발산 $\mathrm{KL}(q_\omega \| p(\cdot|\mathbf{x}, \mathbf{y}))$를 최소화하는, 또는 evident lower bound (ELBO) $\mathbb{E}_{q_\omega}[\log p(\mathbf{y}|\mathbf{x}, \theta)] - \mathrm{KL}(q_\omega \| p)$를 최대화하는 모수 ω를 찾는다.

최적화는 그래디언트 방법을 이용하는데, $\nabla_\omega \int q_\omega(\theta) \log p(\mathbf{y}|\mathbf{x}, \theta)\, d\theta - \nabla_\omega \mathrm{KL}(q_\omega \| p)$의 앞부분을 재매개화한 뒤 몬테카를로 추정치를 이용하면 효율적으로 구할 수 있다. 만일 어떤 함수 f_ω와 ω에 의존하지 않는 분포 q가 존재하여 $f_\omega(Z)$ $(Z \sim q)$의 분포와 q_ω가 같다면 다음이 성립한다:

$$\nabla_\omega \int q_\omega(\theta) \log p(\mathbf{y}|\mathbf{x}, \theta)\, d\theta \simeq \frac{1}{k} \sum_{j=1}^{k} \frac{\partial \log p(\mathbf{y}|\mathbf{x}, f_\omega(Z_{(j)}))}{\partial \omega}, \quad Z_{(j)} \overset{\text{iid}}{\sim} q,\ j = 1, \ldots, k.$$

한편 위에서 제시한 변분 베이즈는 특정 조건에서 드롭아웃(dropout)과 같은 특성을 갖는다. 드롭아웃은 딥러닝 모형에서 자주 사용되는 제약 방법(regularization method)의 하나로, 학습 과정에서 단계마다 모수를 전부 학습시키는 대신 일부분만 선택하여 학습시키는 방식이다. 계산에 요구되는 자원이 적으며 모형의 형태나 학습 알고리즘에 상관없이 적용할 수 있다.

변분 베이즈에서

$$q_\omega(\theta) = \prod_j q_{\omega_j}(\theta_j),$$
$$q_{\omega_j}(\theta_j) = \pi \mathbb{I}(\theta_j = \omega_j) + (1-\pi)\mathbb{I}(\theta_j = 0) \tag{3.1}$$

로 정의하자 ($0 < \pi < 1$). 이 때, Gal과 Ghahramani (2016a)는 q_ω를 사용하는 변분 베이즈가 드롭아웃과 정확히 같음을 보였으며, 이를 이용하면 드롭아웃을 통해 예측 분포의 평균 등을 몬테카를로 방법으로 구할 수 있다. 또한, 이렇게 베이즈 관점으로 드롭아웃을 해석하는 경우에 기존 순환신경망에서 사용되던 드롭아웃 기법과 다른 새로운 기법을 유도할 수 있다 (Gal과 Ghahramani, 2016b).

4. 모의실험 환경

앞서 언급하였던 모든 모형을 이용하여 한국어 음성인식 모형을 구현하였으며, 이를 이용하여 모의실험들을 진행하고 그 결과를 비교하였다. 모의실험에 사용된 소스 코드는 https://github.com/dltnwl/KoSR에 공개되어 있다.

4.1. 자료 수집

국립국어원에서 운영하는 우리말샘 온라인 사전을 크롤링(crawling)하여 음성 자료를 수집하였다. 그중 임의로 56,990개의 단어(총 23.1시간)를 골라 학습에 사용하였고 또 다른 500개의 단어(총 12.4분)를 이용하여 모형의 성능을 측정하였다.

4.2. 전처리

4.2.1. Mel-frequency cepstral coefficients Mel-frequency cepstral coefficients (MFCC)는 대부분의 음성인식 모형에서 사용되는 전처리 기법이다 (Huang 등, 2001). MFCC를 구하기 위해서는 우선 오디오 파일을 약 10–20ms 간격으로 나누어 총 T개의 프레임 $\mathbf{f}_1, \ldots, \mathbf{f}_T \in \mathbb{R}^N$를 만든다. 각 프레임 $\mathbf{f}_t$에는 일정 시간 간격동안 음성의 파형이 저장되어 있다. 이제 각 $\mathbf{f}_t$마다 이산푸리에변환(discrete Fourier transform)을 취해 $\mathbf{F}_t = (F_{t1}, \ldots, F_{tN}) \in \mathbb{R}^N$

$$F_{tn} = \sum_{n'=1}^{N} f_{tn'} e^{-\frac{2\pi i}{N}(n-1)(n'-1)}, \quad 1 \le n \le N$$

을 계산한다. M개의 필터를 가지는 멜 필터뱅크(mel filterbank) $\{H_m; m = 0, 1, \ldots, M-1\}$을 정의하자. 우선 함수 $\mathrm{Mel}(f) = 1125\log(1 + f/700)$을 정의하고

$$b_m = \frac{N}{f_{\text{sample}}} \mathrm{Mel}^{-1}\left(\mathrm{Mel}(f_{\text{low}}) + m\frac{\mathrm{Mel}(f_{\text{high}}) - \mathrm{Mel}(f_{\text{low}})}{M+1}\right), \quad 0 \le m \le M$$

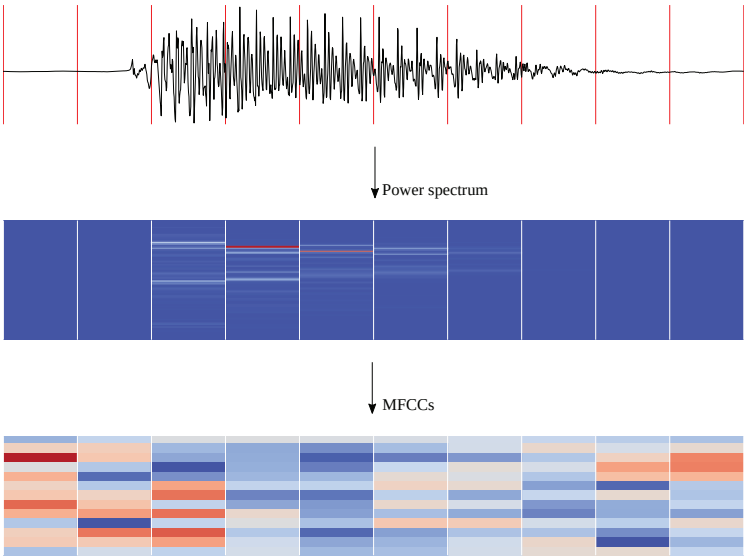

Figure 4.1. Mel-frequency cepstral coefficients.

로 한다. 여기서 f_{sample}은 오디오 파일의 샘플링 주파수이고, f_{low}와 f_{high}는 사용하고자 하는 필터들의 최소 주파수와 최대 주파수이다. 이제 H_m을 다음과 같이 정의하자:

$$H_m(n) = \begin{cases} \dfrac{2(n - b_{m-1})}{(b_{m+1} - b_{m-1})(b_m - b_{m-1})}, & b_{m-1} \le n < b_m, \\ \dfrac{2(b_{m+1} - n)}{(b_{m+1} - b_{m-1})(b_{m+1} - b_m)}, & b_m \le n < b_{m+1}, \\ 0, & \text{이 외의 경우.} \end{cases}$$

이제 멜 필터뱅크를 적용한뒤 로그를 취해 이산코사인변환(discrete cosine transform)을 적용하면 MFCC $\mathbf{c}_t = (c_{t1}, \ldots, c_{tM}) \in \mathbb{R}^M$

$$c_{tm} = \sum_{m'=0}^{M-1} \log \left(\sum_{n=1}^{N} |F_{tn}|^2 \, H_{m'}(n-1) \right) \cos \left(\frac{m' + 1/2}{M} (m-1)\pi \right), \quad 1 \le m \le M.$$

를 얻는다. Figure 4.1은 이 과정을 도식화한 것이다.

본 논문에서는 $M = 13$으로 설정하였으며, 또한 MFCC로부터 델타 계수(delta coefficients) $\Delta\mathbf{c}_t = \mathbf{c}_{t+2} - \mathbf{c}_{t-2}$ 및 $\Delta\Delta\mathbf{c}_t = \Delta\mathbf{c}_{t+1} - \Delta\mathbf{c}_{t-1}$를 추가로 계산하여 $x_t = (\mathbf{c}_t, \Delta\mathbf{c}_t, \Delta\Delta\mathbf{c}_t)$로 사용하였다.

4.2.2. 자모 분리 간단한 실험 결과, 종단 간 한국어 음성 인식 모형에서 문자열 집합 $\mathcal{Y}$를 한글 글자로 사용하는 대신 이를 초성(onset), 중성(nucleus) 및 종성(coda)로 나눌 때에 계산 시간 및 정확도 면에서 더 좋은 결과를 주는 것이 확인되었다. 요컨대, "통계학"이라고 말하는 음성 자료가 있을 때, 이를 "ㅌㅗㅇㄱㅖㅎㅏㄱ"으로 디코딩한 뒤 다시 한글로 변환하는 방식이 "통계학" 글자로 바로 디코딩하는 것보다 효율적이다. 이는 직관적으로도 쉽게 받아들일 수 있는 현상인데, $\mathcal{Y}$를 한글 글자의 집합으로 하면 $|\mathcal{Y}| = 11172$지만 반면 $\mathcal{Y}$를 한글 자모로 하는 경우 $|\mathcal{Y}| = 40$로 학습 및 디코딩 시간을 크게 줄일

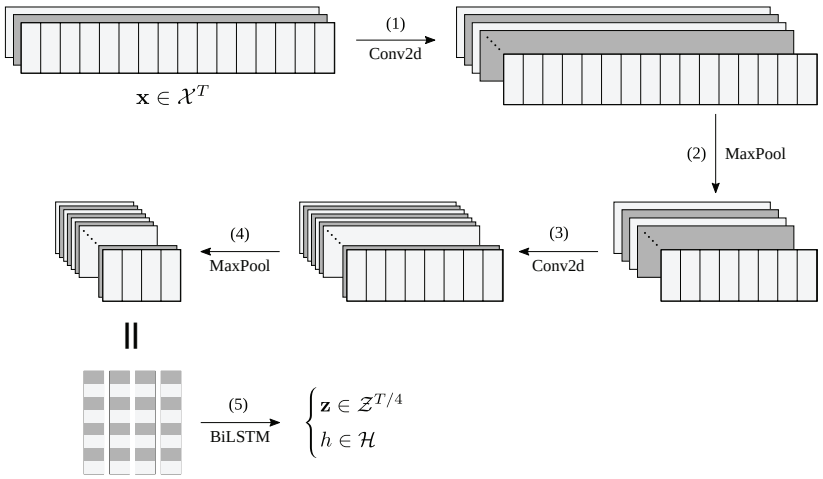

Figure 4.2. The structure of the encoder.

수 있다. 또한, 그대로 한글 글자를 사용한다면, "간"과 "궁"의 경우와 같이 초성은 같은 발음을 가지지만 따로 학습하는 문제가 생기는데 자모로 분리하여 학습하는 경우는 이러한 문제가 발생하지 않는다.

4.3. 모형 세부 사항

앞서 제시하였던 모든 모형들을 구현하였으며, 각각은 모두 동일한 인코더 구조를 가지며 그 구성은 다음과 같다.

(1) 64개 필터를 가지는 3×3 합성곱 계층(convolutional layer) 2개

(2) 2×1 최대 풀링 계층(max pooling)

(3) 128개 필터를 가지는 3×3 합성곱 계층 2개

(4) 2×1 최대 풀링 계층

(5) 1024차원 메모리를 가지는 양방향 LSTM

이때, MFCC로 전처리된 결과 $\mathbf{x}$를 이미지 자료처럼 다루었으며 딥러닝을 이용한 이미지 분석에 주로 사용되는 합성곱 계층 및 최대 풀링 계층을 사용했다. Figure 4.2는 인코더 구조를 도식화한 것으로, 그림의 번호는 인코더 구조 번호에 대응된다.

주의 기제를 쓰는 경우, 디코더로는 16차원 임베딩 계층(embeding layer)과 128차원 메모리를 가지는 GRU를 사용하였다. CTC와 주의 기제의 결합 모형에서, 손실 함수의 가중치 $\lambda = 0.2$로 설정하였다. 모든 모형에서 드롭아웃은 인코더의 양방향 LSTM에만 적용하였으며, 드롭아웃 확률 $\pi = 0.5$로 고정하였다. 드롭아웃은 일반 드롭아웃과 Gal과 Ghahramani (2016b)가 제시한 베이지안 드롭아웃을 사용하여 성능을 비교하였다. 즉, 실제 사후분포의 근사치를 쉽게 다룰 수 있는 확률밀도함수의 모임으로 식 (3.1)를 사용하였다.

4.4. 유한 오토마타 기반의 언어모형

앞서 속도 및 정확성을 위해 자모를 분리하였는데, 이 경우 이후 얻어진 결과물들 또한 초성/중성/종성의 순서가 맞아야 하며 그렇지 않은 경우는 다시 원래 한국어 문자열로 변환할 때 문제가 발생한다. 이

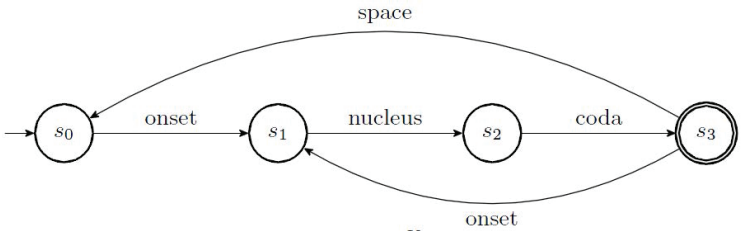

Figure 4.3. A finite automata that searches for correct Korean strings.

문제는 언어모형을 적용하여 제대로 된 한국어 문자열로 변환이 가능한 경우들만 탐색하도록 설정하는 것으로 해결할 수 있다. 본 논문에서는 유한한 개수의 상태 유한 오토마타(finite automata; FA)에 기반한 언어모형을 사용하였다. 유한 오토마타는 다음과 같은 요소들의 쌍(tuple)으로 정의된다:

- 문자 집합 $\mathcal{Y}$.
- 상태 집합(state set) $\mathcal{S} = \{s_0, s_1, \ldots, s_K\}$.
- 시작 상태(start state) $s_0 \in \mathcal{S}$.
- 받아들이는 상태(accepting state) $\mathcal{A} \subset \mathcal{S}$.
- 전이 함수(transition function) $\delta : \mathcal{S} \times \mathcal{Y} \to \mathcal{S}$.

유한 오토마타는 다음의 과정으로 문자열을 판단한다.

(1) 문자열 $\mathbf{y} = (y_1, \ldots, y_L) \in \mathcal{Y}^\infty$가 주어진 상태에서, 현재 상태 s를 시작 상태 s_0로 초기화한다 ($s \leftarrow s_0$).

(2) 문자를 계속 읽어가면서 $s \leftarrow \delta(s, y_i)$로 상태를 계속 갱신하되, 중간에 $\delta(s, y_i)$가 정의되지 않는 경우는 입력 $\mathbf{y}$가 원하는 문자열이 아니라고 판단한다.

(3) 위 과정을 L번 반복한 뒤 최종적으로 얻어진 상태 s가 $\mathcal{A}$에 속하면 입력 $\mathbf{y}$가 원하는 문자열이라 판단하고, 그렇지 않은 경우는 아니라고 판단한다.

Figure 4.3은 한글 자모 입력으로부터 올바른 한국어 문자열을 찾는 유한 오토마타로, 이 오토마타가 받아들이는 문자열만 정상적인 한국어 문자열로 변환할 수 있다. 여기서 $\mathcal{Y}$는 모든 한글 자모와 띄어쓰기의 집합이며 $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$, $\mathcal{A} = \{s_3\}$이고 δ는 다음과 같다:

$$\delta(s, y) = \begin{cases} s_1, & s = s_0,\ y \in \text{onset}, \\ s_2, & s = s_1,\ y \in \text{necleus}, \\ s_3, & s = s_2,\ y \in \text{coda}, \\ s_1, & s = s_0,\ y \in \text{onset}, \\ s_0, & s = s_0,\ y = \text{space}, \\ \text{reject}, & \text{otherwise}. \end{cases}$$

모형의 성능은 두 개의 문자열이 같아지기 위해 필요한 추가(add), 편집(edit), 삭제(delete)의 개수인 편집 거리(edit distance)를 바탕으로 한 값들로 측정하였다. 추정치와 실제 문자열의 편집 거리를 실제 문자열의 길이로 나눈 값들을 사용하였으며, 단어 단위로 계산한 word error rate (WER)과 자모 단위로 계산한 label error rate (LER)을 사용하였다.

Table 5.1. Performance comparison between end-to-end deep learning models

	BW	CTC		Attention		CTC + Attention	
		LER (%)	WER (%)	LER (%)	WER (%)	LER (%)	WER (%)
Base model	1	5.092	33.00	5.750	30.40	4.311	25.40
	8	5.087	32.80	5.717	30.20	**4.278**	**25.20**
	32	5.087	32.80	5.717	30.20	**4.278**	**25.20**
VB-Dropout	1	5.328	34.00	4.694	26.60	5.912	29.00
	8	5.356	34.00	4.583	26.40	5.964	28.80
	32	5.356	34.00	4.583	26.40	5.964	28.80
VB-Dropout-MC	1	5.422	34.40	4.840	28.20	6.703	30.40
	8	5.442	34.00	5.096	27.80	6.703	30.40
	32	5.561	35.20	4.770	26.40	6.428	29.60

BW = beam width; CTC = connectionist temporal classification; LER = label error rate; WER = word error rate; VB = variational Bayes; MC = Monte Carlo.

4.5. 모형 학습

모든 모형의 학습은 확률적 경사 하강법(stochastic gradient descent) 방법 중 하나인 Adam을 이용하였다 (Kingma와 Ba, 2014). 학습률(learning rate)은 10^{-4}로 설정하였으며, 총 40번의 에폭(epoch)을 학습하였다. 또한 그래디언트의 노름(norm)이 너무 커져서 학습의 악영향을 끼치는 것을 방지하기 위해, 노름이 일정 값을 넘지 못하게 하는 그래디언트 클리핑(gradient clipping)을 사용하였다.

5. 모의실험 결과

본 논문에서는 한국어 음성인식의 성능 개선을 위해 베이즈 신경망을 적용한 종단 간 학습 모형들을 제안하였다. 제안한 모형들을 검증하기 위해 한국어 단어 사전을 사용하여 각 모형의 정확도를 비교하였으며 나아가 현재 상용되는 한국어 음성인식 API들과도 성능을 비교했다. 5.1절에는 제안한 종단 간 모형들의 성능을, 5.2절에는 제안한 모형 중 가장 우수한 모형과 실제 상용되는 API의 성능을 비교하였다.

5.1. End-to-end, VB/end-to-end 비교(단어)

본 논문에서 제안한 베이즈 딥러닝 결합의 종단 간 학습 모형(VB-Dropout, VB-Dropout-MC)을 한국어 사전 음성에 적용하여 Table 5.1에 정리하였다. 또한 빔 너비가 음성 인식 결과에 미치는 영향을 확인하기 위해 빔 너비를 1, 8, 32로 변화시켰을 때의 결과를 함께 정리하였다. Base model은 일반적인 종단 간 학습 모형들을, VB-Dropout은 베이즈 딥러닝을 적용한 종단 간 학습 모형의 점 추정치를 뜻한다. VB-Dropout-MC는 베이즈 딥러닝을 적용한 여러 모형의 예측 값들 평균치를 사용한 몬테카를로 추정치이다.

실험 결과 일반적인 종단 간 학습 모형에서는 CTCAtt 결합(CTC + Attention) 모형, CTC, 주의 기제(Attention) 모형 순으로 좋은 성능을 보였으며, 변분 베이즈를 적용한 모형에서는 주의 기제 모형, CTC, CTCAtt 결합 모형 순으로 좋은 성능을 보였다. 일반적인 종단 간 학습 모형과 베이즈 딥러닝을 적용한 모형의 성능을 비교해보았을 땐 주의 기제 모형에서 베이즈 딥러닝을 적용한 모형이 더 나은 성능을 보인 반면 CTC나 CTCAtt 결합 모형에서는 일반적인 모형의 성능이 더 좋았다. 빔 너비를 변화시켰을 때의 결과에서는 변분 베이즈 모형의 몬테카를로 추정치를 제외하고는 너비를 8 이상 늘리는 것이 성능에 영향을 주지 않았다.

Table 5.2. Performance comparison when adding a finite automata language model

	BW	CTC		Attention		CTC + Attention	
		LER (%)	WER (%)	**LER (%)**	WER (%)	LER (%)	WER (%)
Base model	1	4.672	27.00	5.390	29.40	4.261	25.20
	8	**4.102**	25.60	5.390	29.20	4.178	**24.80**
	32	**4.102**	25.60	5.390	29.20	4.178	**24.80**
VB-Dropout	1	4.986	28.60	4.628	26.60	5.796	28.60
	8	4.474	27.00	4.583	26.40	5.848	28.40
	32	4.474	27.00	4.583	26.40	5.848	28.40
VB-Dropout-MC	1	4.578	27.40	4.816	26.80	6.729	30.60
	8	4.772	27.80	4.906	27.40	6.704	30.60
	32	4.617	28.20	4.779	26.80	6.351	28.60

BW = beam width; CTC = connectionist temporal classification; LER = label error rate; WER = word error rate; VB = variational Bayes; MC = Monte Carlo.

Table 5.3. Performance comparison with commercial API

Model	LER (%)	WER (%)
Attention + FA(VB-Dropout)	4.58	26.4
Google API	29.88	76.0
Naver Clova	21.65	35.0

API = Application Programming Interface; FA = finite automata; LER = label error rate; WER = word error rate; VB = variational Bayes.

Table 5.2은 종단 간 학습 모형들에 유한 오토마타 언어 모형을 적용했을 때의 결과를 나타낸 것이다. 언어 모형의 적용 결과, 대부분 모형에서 성능 향상을 보였으며 CTC에서 가장 높은 성능 향상을 보였다. 최소의 LER과 WER을 갖는 모형은 각각 언어모형이 적용된 빔 너비가 8 이상인 일반적인 CTC, CTCAtt 결합 모형이었다.

5.2. 기존 API와의 비교(단어)

Table 5.3은 제안한 모형 중 가장 우수한 모형과 실제 상용되는 API의 성능을 비교한 것이다. 이때, 성능 비교를 위한 검증 데이터로 사전 발음 자료 중 100개의 단어를 사용하였다. 성능을 비교하기 위한 모형은 5.1절에서 가장 우수한 정확도를 보인 CTCAtt 결합 모형과 베이즈 딥러닝 음성 인식 모형, 상용 API인 Google API와 네이버 클로바(Clova)를 이용하였다. 본 논문에서 제안한 모형이 상용 API보다 우수한 성능을 보임을 확인할 수 있다. 특히 제안된 모형의 WER와 LER는 각각 26.4%와 4.58%로서, 76%의 WER 값과 29.88 %의 LER 값을 보인 Google API보다 월등히 개선된 성능을 보였다.

6. 논의

본 논문에서는 여러 가지 종단 간 모형을 제안하고 한국어 음성인식에 적용하였다. 종단 간 모형으로 CTC, 주의 기제, CTC 주의 기제 결합 모형을 제안하였고, 종단 간 모형의 결과($\mathcal{Y}^{\leq T}$ 또는 $\mathcal{Y}^{\infty}$)로부터 최적의 문자열을 찾기 위해 빔 탐색을 이용하였다. 추가적으로 유한 오토 마타를 이용하여 한글의 초중종성 순서를 조정하였다. 또한 변분 베이즈를 종단 간 학습 모형에 적용하여 몬테카를로 추정치도 구하였으며 이를 비롯한 여러 모형들의 정확도를 비교하였다. 그 결과 논문에서 제안한 모형의 WER는 26.4%의 WER를 보여 각각 76%, 35%의 WER를 보인 기존의 API 보다 월등히 우수한 성능을 보였

다.

한편 본 논문에는 다음의 두 가지 한계점이 있다. 먼저, 유한 오토마타 언어 모형을 사용했을 때 성능 향상의 폭이 미미했다는 점이 있다. 오토마타 언어모형은 한글의 초중종성의 순서만 조정하며 한국어의 언어적 특성에 대한 후처리는 하지 않는다. 한국어의 언어적 특성을 고려한 후처리를 배제할 경우, 한글의 자음 동화, 구개음화, 된소리 거센소리 등을 반영하지 못하여 음성인식의 결괏값이 실제 문자언어와 괴리가 생기게 된다. 모형의 성능을 개선하기 위해서는 문법, 문맥상의 정확도를 높이는 추가적인 후처리를 적용해야 한다. 다음으로, 연구를 위한 음성 자료가 부족했다는 점이 있다. 5.2절의 결과에서 본 논문에서 제안된 모형은 상용 API보다 우수한 성능을 보였으나, 실험에 쓰인 검증 데이터가 제안된 모형의 훈련 데이터와 유사한 성질을 지녔기에 편향적인 결과가 나올 수 있음을 고려해야 한다.

종단 간 분석을 적용한 한국어 음성인식은 기존 모형이 가졌던 복잡성과 비효율성을 해결하면서 높은 성능을 보였다는 점에 의의가 있다. 기존에는 한국어의 언어적 특성 등으로 인해서 매우 많은 양의 정제된 데이터가 필요했으며 복잡한 중간 과정을 거쳐야 했다. 이 과정에서 형태소를 비롯한 한국어에 대한 전문적인 지식이 결부되어야만 한국어 음성인식 모형을 만들 수 있었으나, 5장에서 확인할 수 있듯이 간소화된 과정만으로도 높은 성능의 한국어 음성인식을 구현할 수 있었다. 또한, 베이즈 딥러닝 모형을 적용하여 결과의 불확실성에 대한 정보를 함께 제안할 수 있다는 점에도 의의가 있다. 베이즈 딥러닝 적용한 모형은 그렇지 않은 모형과 비교했을 때, 그 분포를 함께 제안할 수 있다는 장점이 있다. 이 점을 활용하면 부족한 훈련 데이터에서 기인하는 한계점을 보완할 수 있을 것이라 기대한다.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint* arXiv:1409.0473

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, **5**, 157–166.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks, *arXiv preprint*, arXiv:1505.05424

Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O. (2015). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4960–4964. IEEE, 2016.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-Decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *arXiv preprint*, arXiv:1406.1078

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint*, arXiv:1412.3555

Gal, Y. and Ghahramani, Z. (2016a). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 1050–1059.

Gal, Y. and Ghahramani, Z. (2016b). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, 1019–1027.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning* (Vol. 1), MIT press, Cambridge.

Graves, A., Fernandez, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376. ACM.

Gales, M., and Young, S. (2008). The application of hidden Markov models in speech recognition, *Foundations and Trendsμl kPa in Signal Processing*, **1**, 195–304.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, **9**, 1735–1780.

Huang, X., Acero, A., and Hon, H. (2001). Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice hall PTR, New Jersey.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*, MIT press, Cambridge.

Kim, S., Hori, T., and Watanabe, S. (2017). Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 4835–4839. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Kwon, O. W. and Park, J. (2003). Korean large vocabulary continuous speech recognition with morphemebased recognition units, *Speech Communication*, **39**, 287–300.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint* arXiv:1508.04025

딥러닝 모형을 사용한 한국어 음성인식

이수지[a] · 한석진[a] · 박세원[a,1] · 이경원[a] · 이재용[a]

[a]서울대학교 통계학과

(2018년 12월 21일 접수, 2019년 2월 13일 수정, 2019년 2월 14일 채택)

요 약

본 논문에서는 베이즈 신경망을 결합한 종단 간 딥러닝 모형을 한국어 음성인식에 적용하였다. 논문에서는 종단 간 학습 모형으로 연결성 시계열 분류기(connectionist temporal classification), 주의 기제, 그리고 주의 기제에 연결성 시계열 분류기를 결합한 모형을 사용하였으며. 각 모형은 순환신경망(recurrent neural network) 혹은 합성곱신경망(convolutional neural network)을 기반으로 하였다. 추가적으로 디코딩 과정에서 빔 탐색과 유한 상태 오토마타를 활용하여 자모음 순서를 조정한 최적의 문자열을 도출하였다. 또한 베이즈 신경망을 각 종단 간 모형에 적용하여 일반적인 점 추정치와 몬테카를로 추정치를 구하였으며 이를 기존 종단 간 모형의 결괏값과 비교하였다. 최종적으로 본 논문에 제안된 모형 중에 가장 성능이 우수한 모형을 선택하여 현재 상용되고 있는 Application Programming Interface (API)들과 성능을 비교하였다. 우리말샘 온라인 사전 훈련 데이터에 한하여 비교한 결과, 제안된 모형의 word error rate (WER)와 label error rate (LER)는 각각 26.4%와 4.58%로서 76%의 WER와 29.88%의 LER 값을 보인 Google API보다 월등히 개선된 성능을 보였다.

주요용어: 한국어 음성인식, 종단 간 딥러닝, 연결성 시계열 분류기, 주의 기제, 베이즈 딥러닝

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2018R1A2A3074973).

[1]교신저자: (108826) 서울시 관악구 관악로 1, 서울대학교 통계학과. E-mail: swpark0413@gmail.com