# 인공신경망 기반 한국어 음성인식의 방법 비교

Comparison of Korean Speech Recognition System based on Artificial Neural Network


저자
(Authors)

이루카스, 성원용
Lee Lukas, Sung Wonyong



출처
(Source)

한국통신학회 학술대회논문집 , 2018.6, 847-848 (2 pages)
Proceedings of Symposium of the Korean Institute of communications and Information Sciences , 2018.6, 847-848 (2 pages)

발행처
(Publisher)

한국통신학회
Korea Institute Of Communication Sciences

URL

http://www.dbpia.co.kr/Article/NODE07512764

APA Style

이루카스, 성원용 (2018). 인공신경망 기반 한국어 음성인식의 방법 비교. 한국통신학회 학술대회논문집, 847-848.

# 인공신경망 기반 한국어 음성인식의 방법 비교

이루카스, 성원용
서울대학교

proboscis@snu.ac.kr, wysung@snu.ac.kr

# Comparison of Korean Speech Recognition System based on Artificial Neural Network

Lee Lukas, Sung Wonyong
Seoul National Univ.

## 요 약

본 논문은 CTC, TDNN, EESEN 모델로 한국어 음성 인식 시스템을 구현하였다. 또 단어 오류율 (WER)을 중심으로 한국어 음성인식 시스템의 성능을 비교하였다. 성능 비교 결과 단어 오류율은 CTC 와 Kaldi 기반의 TDNN 모델이 유사하게 나타났다. EESEN은 다른 모델에 비해 정확도가 떨어졌다.

## Ⅰ. 서 론

인공신경망 기반 음성 인식 시스템에서는 음향 모델 훈련 방식, 디코딩 방식의 등의 차이 등에 따라 CTC(Connectionist Temporal Classification) 또는 DNN-HMM 기반의 WFST등이 활용되고 있다. 각각의 모델은 음성 인식 시스템 구현에 있어서 장단점이 있고 영어의 경우 특정 음성 데이터셋을 기준으로 성능이 비교되었다. 그러나 한국어의 경우 기준이 되는 음성 데이터셋이 없고, 이에 따라 공개적인 성능 비교도 없는 실정이다.

따라서 본 논문에서는 각 음성인식시스템에 대해 한국어 모델을 적용하여 구현하고 정확도를 중심으로 성능을 비교한다. 구체적으로 바이두(Baidu)의 딥 스피치2(Deep Speech2)[1] 기반의 CTC, 칼디(Kaldi) 기반의 HMM-DNN, CTC와 WFST 디코딩을 결합한 EESEN으로 한국어 음성인식시스템을 구현하였다. 또, 각각 음성 인식 시스템의 성능을 비교하였다.

## Ⅱ. 본 론

### 1. CTC

CTC[2]는 입력의 길이와 정답의 길이가 다른 순서 데이터(Sequence data)를 대응시키는 지도학습 방법이다. CTC를 통한 음향모델의 훈련 과정은 다음과 같다. 우선, 입력과 출력 텍스트의 길이가 다르기 때문에 다르기 때문에 <blank> 기호를 추가하여 입력과 출력의 길이를 맞추도록 한다. 이 경우 정답에 대응되는 다양한 경로가 생기고 정답을 든 경로를 발생시키는 확률 값을 모두 합산한다. 이 때 정확한 라벨의 가능도(likelihood)를 최대화하도록 목적함수를 정하고 순환신경망을 이용해 훈련한다. 이러한 CTC는 음향 정보간 출력 텍스트 간 강제 정렬 과정이 필요가 없고, HMM(Hidden Markov Model)-GMM(Gaussian Mixture Model)의 훈련 과정이 없는 장점이 있다.

본 논문에서는 음향 모델을 CTC로 훈련 할 때 딥 스피치2의 CTC 목적함수를 활용하였다. 신경망의 구조 컨볼루션 신경망 두 층과 LSTM 6층을 쌓아서 만들었다. 또 배치 정규화(Batch normalization)을 활용하였고 마지막 신경망 층에는 Fully Connected 층을 추가했다.

### 2. TDNN 모델

TDNN(TIME-delay deep neural network)[3]은 시간적으로 넓은 범위의 시계열 자료를 보면서 음향 모델을 학습하는데 이용된다. TDNN의 낮은 층에서는 좁은 범위 프레임의 특징 정보를 학습한다. 높은 층으로 갈수록 더 넓은 범위의 프레임 정보를 보고 학습할 수 있게 된다.

이 때 일반적인 TDNN의 계산량을 줄이기 위해 네트워크를 subsampling하고 이렇게 뽑힌 부분 네크워크들을 연결한다. Subsampling 과정을 통해 순전파(forwardpass)와 역전파(backpropagation) 과정에서 계산량을 줄일 수 있다.

본 논문에서는 음성인식 툴킷인 칼디를 활용해 TDNN을 활용한 음향모델의 훈련을 진행하였다. 칼디의 Nnet3는 HMM-DNN을 이용한 음향 모델 훈련 방식 중 TDNN 훈련 방식을 제공한다. 또 WFST[4]를 이용해 디코딩한다.

### 3. ESSEN

ESSEN[5] 프레임워크는 WFST 디코딩을 활용하여 CTC 훈련을 활용한 음향 모델과 사전, 언어모델을 결합한다. 음향 모델의 경우 LSTM과 CTC 목적함수를 이용해 훈련한다. 음성인식시스템에 필요한 각 요소는 WFST 그래프로 대응된다. n-gram 언어 모델은 문법(G)요소로 나타내어 한 문장에서 단어 간 확률 관계를 나타낼 수 있다. 또, 각 단어와 그 단어를 쪼갠 단위를 연결 하는 단어 사전은 렉시콘(L) 요소로 나타낼 수 있다. 또, 공백 라벨 (blank)를 포함한 여러 라벨을 이를 토큰(T)요소로 나타낼 수 있다. ESSEN에서는 이 각 요소를 합성하여 하나의 통합된 WFST로 나타낸 후 최소화(minimization), 결정(determinzation) 등을 거쳐 최적화 한다

본 논문에서는 음향모델로 훈련하는 문장 라벨을 음소(phone) 단위로 바꿔 토큰 요소로 활용하였다. 또 CTC훈련의 경우 LSTM 4 층을 쌓아 진행하였다.

4. 구현 방법

음향모델 훈련 데이터로는 제로스[6]의 데이터 50시간을 활용하였다. 테스트 데이터로는 제로스의 테스트 셋을 활용하였다. 주어진 것은 한국어 음성 자료와 받아쓰기 자료였다. 받아쓰기 자료를 한국어 발음 규칙에 따라 각 단어의 한국어의 발음을 49개의 음소(phone)으로 바꾸었다. 또, 음성 데이터에서 25ms의 프레임 윈도우를 10ms씩 이동해 가면서 MFCC 특징 정보를 추출하여 훈련에 이용하였다.

CTC의 경우 한글 받아쓰기 자료를 형태소 별로 분할한 후, 이를 음소 단위의 영어 글자 조합으로 바꿔서 총 49개의 라벨로 활용하였다. 예를 들면 “안녕하세요” 를 “an ny eo ng ha se ye oo”로 변환 한 후 이를 라벨로 활용하였다. EESEN의 경우도 같은 방법으로 훈련 라벨을 만들었다. CTC와 EESEN은 HMM가 아닌 CTC기반이기 때문에 모노폰(monophone) 단위로 훈련하였다. TDNN-모델도 같은 방법으로 모노폰 단위를 만들었다. 그러나 TDNN 모델은 여기에 추가하여, 문맥의존적 트라이폰(context dependent triphone) 단위로 HMM을 만들었다. 또, TDNN 모델의 경우 각 음성과 받아쓰기 자료의 시계열 정렬이 자동으로 이루어지지 않는다. 따라서 GMM을 이용해 입출력간 강제 정렬(Forced alignment) 후 훈련을 진행하였다.

TDNN 모델과 Eesen에서 언어모델로 n-gram모델이 이용되었다. 웹 크롤링하여 각 단어의 빈도수와 단어 간 빈도 관계를 칼디 언어모델 툴킷을 사용해 훈련한 후 알파(Arpa) 파일을 생성했다. 이를 제로스의 n-gram 모델과 합쳐서 이용했다. 또한 이를 언어모델 재점수화(rescoring)에 활용하였다. 또 이 알파파일의 구성 단어들을 모아 단어 사전(Lexicon)의 구성요소로 이용하였다. 단어 사전에서는 각 단어와 그 단어의 발음에 대응하는 음소의 조합을 대응시켰다. 공통적으로 EESEN과 TDNN에서 활용한 단어 사전의 단어 개수는 약 58만개이다.

한국어의 경우 교착어이므로 어간이 활용되어 다양한 어미와 결합할 수 있다. 따라서 단순히 띄어쓰기 단위로만 언어모델을 만들 경우 나올 수 있는 단어의 종류가 무한정 많아진다. 또 코퍼스에서 단어가 나오는 실질적인 빈도수 파악도 어려워진다. 따라서 Zeroth Project에서는 단어 사전의 단위로 형태소를 사용한다. 다만 형태소는 띄어쓰기로 구분되어 있지 않으므로 Morfessor[7] 툴킷을 활용하여 형용사 단위를 지도학습으로 훈련하여 사용하였다. 이 방법은 제로스에서 시도된 방법을 동일하게 적용하였다.

디코딩의 경우, CTC에서는 빔 검색(Prefix-beam search)으로 가장 높은 확률을 가진 음소 후보를 찾았다. 빔의 개수는 총 64개를 활용하였다. EESEN과 TDNN모델은 WFST 디코딩을 활용했고 WFST에서는 두 모델 모두 언어모델을 WFST에 결합하여 활용했다.

5. 실험 결과

방식으로 음향 모델을 훈련하고 언어모델을 적용한 결과 , 단어 에러율(WER, Word error rate)은 표 2 과 같았다.

표 2. 단어 에러율(WER) 비교

| | CTC | TDNN-Chain | Eesen |
|---|---|---|---|
| WER | 15.37 % | 14.03 % | 27.53 % |

단어오류율(WER)을 비교하여 보면 CTC와 TDNN 모델의 경우 유사하게 나타났다. EESEN은 이 둘에 비해 단어 오류율이 매우 떨어졌다.

음향 모델의 훈련 과정에서 목적함수로 공통적으로 CTC을 썼음에도 EESEN이 본 논문의 CTC 구현보다 성능이 떨어진다. 신경망 모델의 차이를 그 이유로 추측해볼 수 있다. EESEN의 경우는 CNN을 사용하지 않는다. 따라서, CNN을 EESEN 내부에서 구현하여 신경망을 좀 더 깊게 쌓을 경우 성능 향상을 꾀할 수 있을 것으로 보인다.

추후로 할 수 있는 연구는 다음과 같다. CTC훈련을 진행할 때 정답 라벨을 영어 형태의 음소(phone)로 변환하지 않고, 한글 자음 모음을 CTC의 입출력으로 활용할 수 있다. 이 경우 중간 과정을 제거할 수 있다. 다만 이 경우 한글의 모아 쓰기를 어떻게 진행할지에 대한 연구가 필요하다. 또, 보다 큰 데이터에서 음향 모델을 훈련하여 단어 에러율을 낮추는 것도 과제로 남아있다.

## Ⅲ. 결론

본 논문에서는 한국어 음성인식시스템을 CTC, TDNN모델, EESEN으로 구현하였다. 또 Ngram 언어모델로 재점수화를 진행하였다. 이렇게 구현한 음성인식 시스템의 성능 비교를 진행하였다. 비교 결과 TDNN으로 훈련한 모델이 단어 오류율이 가장 낮았고 CTC 모델이 뒤를 이었다. EESEN 한국어 모델의 경우 성능이 가장 떨어졌다.

## 참 고 문 헌

[1] Amodei, Dario, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." International Conference on Machine Learning. 2016.

[2] Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." International Conference on Machine Learning. 2014.

[3] Peddinti, Vijayaditya, Daniel Povey, and Sanjeev Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[4] Mohri, Mehryar, Fernando Pereira, and Michael Riley. "Weighted finite-state transducers in speech recognition." Departmental Papers (CIS) (2001): 11.

[5] Miao, Yajie, Mohammad Gowayyed, and Florian Metze. "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding." Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015.

[6] 제로스 https://github.com/goodatlas/zeroth

[7] Virpioja, Sami, et al. "Morfessor 2.0: Python implementation and extensions for Morfessor Baseline." (2013).