한국음향학회지 제39권 제5호 pp. 476~482 (2020)
The Journal of the Acoustical Society of Korea Vol.39, No.5 (2020)
https://doi.org/10.7776/ASK.2020.39.5.476

pISSN : 1225-4428
eISSN : 2287-3775

Double-attention mechanism of sequence-to-sequence deep neural networks for automatic speech recognition

음성 인식을 위한 sequence-to-sequence 심층 신경망의 이중 attention 기법

Dongsuk Yook,[1†] Dan Lim,[2] and In-Chul Yoo[1]

(육동석,[1†] 임단,[2] 유인철[1])

[1]Artificial Intelligence Laboratory, Department of Computer Science and Engineering, Korea University,
[2]Kakao Corp.

(Received July 31, 2020; accepted September 14, 2020)

ABSTRACT: Sequence-to-sequence deep neural networks with attention mechanisms have shown superior performance across various domains, where the sizes of the input and the output sequences may differ. However, if the input sequences are much longer than the output sequences, and the characteristic of the input sequence changes within a single output token, the conventional attention mechanisms are inappropriate, because only a single context vector is used for each output token. In this paper, we propose a double-attention mechanism to handle this problem by using two context vectors that cover the left and the right parts of the input focus separately. The effectiveness of the proposed method is evaluated using speech recognition experiments on the TIMIT corpus.

Keywords: Attention, Sequence-to-sequence, Deep neural network, Automatic speech recognition

PACS numbers: 43.72.Bs, 43.72.Ne

초　록: 입력열과 출력열의 길이가 다른 경우 attention 기법을 이용한 sequence-to-sequence 심층 신경망이 우수한 성능을 보인다. 그러나, 출력열의 길이에 비해서 입력열의 길이가 너무 긴 경우, 그리고 하나의 출력값에 해당하는 입력열의 특성이 변화하는 경우, 하나의 문맥 벡터(context vector)를 사용하는 기존의 attention 방법은 적당하지 않을 수 있다. 본 논문에서는 이러한 문제를 해결하기 위해서 입력열의 왼쪽 부분과 오른쪽 부분을 각각 개별적으로 처리할 수 있는 두 개의 문맥 벡터를 사용하는 이중 attention 기법을 제안한다. 제안한 방법의 효율성은 TIMIT 데이터를 사용한 음성 인식 실험을 통하여 검증하였다.

핵심용어: Attention, Sequence-to-sequence, 심층 신경망, 음성 인식

I. Introduction

Recently, sequence-to-sequence Deep Neural Networks (DNN) have been widely used in various tasks such as machine translation,[1,2] image captioning,[3] and speech recognition.[4-7] Attention mechanisms[8-10] are critical in the successful application of the sequence-to-sequence models to those tasks, because the models learn the mapping between differently sized input and output sequences by using the attention mechanisms, thus enabling the models to focus on the relevant portion of the input sequence for each output token.

For domains where the size of the input sequence is much larger than that of the output sequence, the attention mechanisms should be capable of handling a large area of

†**Corresponding author:** Dongsuk Yook (yook@korea.ac.kr)
Artificial Intelligence Laboratory, Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea
(Tel: 82-2-3290-3202)

the input focus for each output token. Particularly, in speech recognition, the input sequences (speech signals) are much longer than the corresponding output sequences (phoneme or word labels). Furthermore, the characteristic of the input speech signals often changes within a single phoneme segment due to the coarticulation effect, implying that a single context vector computed as a weighted average of the high-level representation of the input sequence is insufficient to cover the wide range of a varying input sequence.

In this work, we propose a double-attention mechanism that can handle a large area of the input focus with a varying characteristic caused by the left and the right context in the input. It uses two context vectors that cover the left and the right parts of the input focus separately. The experimental results of speech recognition on the TIMIT corpus show that the proposed method is effective in enhancing the speech recognition performance.

II. Related works

2.1 Sequence-to-sequence models

A sequence-to-sequence model uses the input vector sequence $x_{1:T}=(x_1,x_2,\cdots,x_T)$ and produces the output token sequence $y_{1:N}=(y_1,y_2,\cdots,y_N)$, whose length may differ from that of the input sequence. As shown in Fig. 1, a sequence-to-sequence model typically consists of four modules: encoder, decoder, attender, and generator.

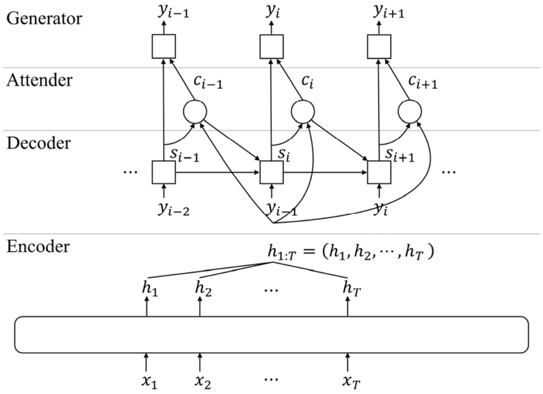

Fig. 1. A sequence-to-sequence deep neural network consisting of encoder, decoder, attender, and generator.

The encoder uses the input vector sequence $x_{1:T}$ and transforms the input into a high-level representation $h_{1:T}=(h_1,h_2,\cdots,h_T)$ as follows:

$$h_{1:T}=\mathrm{Encode}(x_{1:T}). \tag{1}$$

Long Short-Term Memory (LSTM)[11,12] networks or Convolutional Neural Networks (CNNs)[13,14] are typically used for the encoder. Depending on the architecture of the encoder, the length of $h_{1:T}$ may be shorter than that of $x_{1:T}$.

The decoder computes its output s_i at each output token time step i by using the previous decoder's output s_{i-1}, the previous context vector c_{i-1}, and the previous output token y_{i-1} as follows:

$$s_i=\mathrm{Decode}(s_{i-1},c_{i-1},y_{i-1}). \tag{2}$$

In this work, we used LSTM networks for the decoder.

The attender creates the context vector c_i from the encoder's output $h_{1:T}$ and the decoder's output s_i as follows:

$$c_i=\mathrm{Attend}(h_{1:T},s_i). \tag{3}$$

The context vector can be considered as a relevant summary of $h_{1:T}$ that is useful in the prediction of the output token at each time step. Additionally, the attender may utilize the previous alignment vector, which will be explained in the following sections in detail.

The context vector c_i and the decoder's output s_i are fed into the generator to produce the conditional probability distribution of the output token y_i as follows:

$$P(y_i \mid x_{i:T},y_{1:i-1})=\mathrm{Generate}(s_i,c_i). \tag{4}$$

The generator is typically implemented as a MultiLayer Perceptron (MLP) network with a softmax output layer.

2.2 Attention mechanism

The context vector c_i at the output token time step i is defined as a weighted sum of the encoder's outputs h_t's as follows:

$$c_i = \sum_t a_{i,t} h_t, \tag{5}$$

where the alignment vector a_i is the probability distribution of the weights over $h_{1:T}$. That is, $a_{i,t}$ represents the contribution of h_t for the context vector c_i, which is computed as follows:

$$e_{i,t} = \mathrm{Score}(s_i, h_t), \tag{6}$$

$$a_{i,t} = \exp(e_{i,t}) / \sum_{t'} \exp(e_{i,t'}). \tag{7}$$

Typical choice for the score function, $\mathrm{Score}()$, includes an additive score function and a multiplicative score function defined as follows:

$$e_{i,t} = w^T \tanh(\phi(s_i) + \psi(h_t)), \text{(additive)} \tag{8}$$

$$e_{i,t} = \phi(s_i) \bullet \psi(h_t), \text{(multiplicative)} \tag{9}$$

where $\bullet$ is a dot product. $\phi()$ and $\psi()$ are typically implemented using MLPs. These score functions are said to be content-based because each element, $e_{i,t}$, is computed from the content of the decoder's output s_i and the encoder's output h_t.

The hybrid attention mechanism suggested by Chorowski *et al.*[8] extends the content-based attention mechanism by utilizing the previous alignment vector a_{i-1}. At each output token time step i, the location-aware feature f_i is calculated from the previous alignment vector a_{i-1} as follows:

$$f_i = F \otimes a_{i-1}, \tag{10}$$

where F is a learnable convolution matrix, and $\otimes$ is a convolution operator. The location-aware feature is expected to facilitate the sequence-to-sequence model to learn the alignment better, as the current alignment between the speech signal and its corresponding text is dependent on the previous alignment information. It can be fed into the additive score function as follows:

$$e_{i,t} = w^T \tanh(\phi(s_i) + \psi(h_t) + \theta(f_{i,t})), \tag{11}$$

where $\theta()$ is another MLP.

III. Double-attention mechanism

In speech recognition, the size of the encoder's output sequence is much larger than that of its corresponding output token sequence. Furthermore, the left and the right speech contexts may affect the current phoneme to be pronounced, owing to the coarticulation effect as shown in Fig. 2. Therefore, a single context vector may not be sufficient to capture the relevant information in detail, because the single context vector represents the entire relevant information for each output token as a weighted sum of the long encoded sequence. A method to mitigate this problem is to use two separate context vectors computed by two attenders: one for the left context and the other for the right context,[15] as shown in Fig. 3.

The first context vector c_i^1 focusing on the left part of a phoneme is obtained using the attention vector a_i^1 which is computed from s_i, $h_{1:T}$, and f_i^1 as follows:

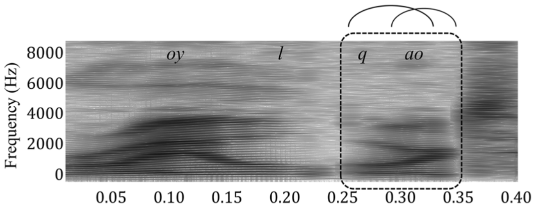

Fig. 2. A sample spectrogram of a waveform. The left and the right portions of a phoneme is affected by the previous and the next speech sounds, respectively.

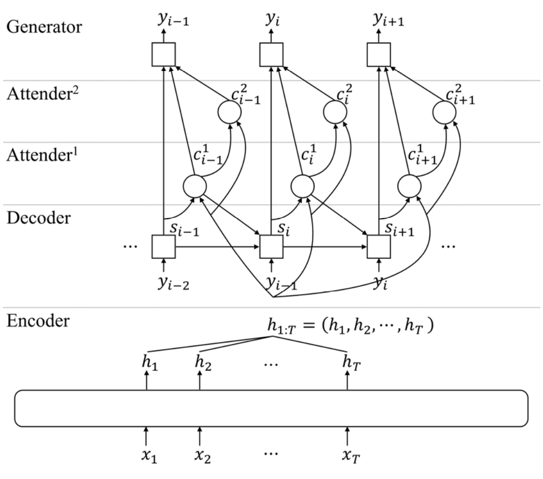

Fig. 3. A sequence-to-sequence model with a double-attention mechanism. It uses two attenders: one for the left context and the other for the right context.

$$f_i^1 = F^1 \otimes a_{i-1}^1, \tag{12}$$

$$e_{i,t}^1 = w^T \tanh(\phi^1(s_i) + \psi^1(h_t) + \theta^1(f_{i,t}^1)), \tag{13}$$

$$a_{i,t}^1 = \exp(e_{i,t}^1) / \sum_{t'} \exp(e_{i,t'}^1), \tag{14}$$

$$c_i^1 = \sum_t a_{i,t}^1 h_t. \tag{15}$$

Similarly, the second attention vector a_i^2 and the second context vector c_i^2 are computed as follows to focus on the right part of the phoneme:

$$f_i^2 = F^2 \otimes a_i^1, \tag{16}$$

$$e_{i,t}^2 = v^T \tanh(\phi^2(c_i^1) + \psi^2(h_t) + \theta^2(f_{i,t}^2)), \tag{17}$$

$$a_{i,t}^2 = \exp(e_{i,t}^2) / \sum_{t'} \exp(e_{i,t'}^2), \tag{18}$$

$$c_i^2 = \sum_t a_{i,t}^2 h_t. \tag{19}$$

It is worth noting that f_i^2 utilizes a_i^1 instead of a_{i-1}^1, and c_i^1 is used instead of s_i in computing $e_{i,t}^2$. Hence, the second attention vector a_i^2 is expected to attend to different parts of the input sequence in contrast to the first attention vector a_i^1, because it considers the first context vector c_i^1 and its alignment information a_i^1.

As the multiplicative score function generally exhibits better performance than the additive score function, further improvements can be achieved by modifying Eqs. (13) and (17) to use the multiplicative score functions that utilize the location-aware feature as follows:

$$e_{i,t}^1 = \phi^1(s_i) \cdot \psi^1(h_t) + w^T \tanh(\theta^1(f_{i,t}^1)), \tag{20}$$

$$e_{i,t}^2 = \phi^2(c_i^1) \cdot \psi^2(h_t) + v^T \tanh(\theta^2(f_{i,t}^2)). \tag{21}$$

Finally, the generator is modified to use both c_i^1 and c_i^2 as well as s_i to compute the posterior probability as follows:

$$P(y_i \mid x_{i:T}, y_{1:i-1}) = \mathrm{Generate}(s_i, c_i^1, c_i^2). \tag{22}$$

The multi-head attention method[10] is similar to the proposed double-attention method in that it inhibits a single averaged context vector. In the multi-head attention method, the input is linearly projected into several subspaces, and the attention is computed in each subspace differently to form multiple attentions. However, it does not care which position to attend to and the order of the multiple attentions. The proposed double-attention mechanism is designed to specifically attend to different parts with an order, i.e., the left and the right parts of a phoneme, to cope with the coarticulation effect. The first context vector is computed using the information available at the previous time (a_{i-1}^1 and s_i), while the second context vector is computed using the information available at the current time (a_i^1 and c_i^1). Therefore, the two context vectors are expected to attend to different parts of the phoneme.

IV. Experiments

To evaluate the effectiveness of the proposed method, speech recognition experiments were conducted using the TIMIT corpus. 40-dimensional Mel-scaled log filter banks with their first and second order temporal derivatives were used as the input feature vectors. The decoder was a single-layer unidirectional LSTM network, whereas the encoder was a deep architecture consisting of seven CNN layers followed by a fully connected layer and three bidirectional LSTM layers as shown in Fig. 4. The first convolutional layer used 256 filters, and the rest of the convolutional layers used 128 filters. The CNN layers used residual connections and batch normalization.[16] The fully connected layer had 1,024 nodes. Each bidirectional LSTM layer had 256 nodes for each direction. To train the model, the Adam algorithm with a learning rate of 10^{-3}, a batch size of 32, and gradient clipping of 1 was used. A dropout with a probability of 0.5 was used across every neural network layer.

Fig. 5 shows example attention vectors for the conventional single-attention mechanism and the proposed double-attention mechanism. As shown in the figure, the area of the single-attention focus (solid lines) is split into two regions (dotted lines for the left context and dashed lines for the right context) using the proposed double-attention mechanism.

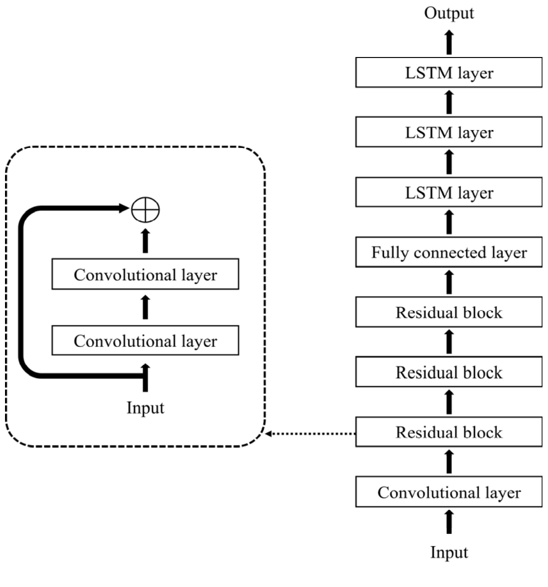

Fig. 4. The structure of the encoder network used in the experiments.

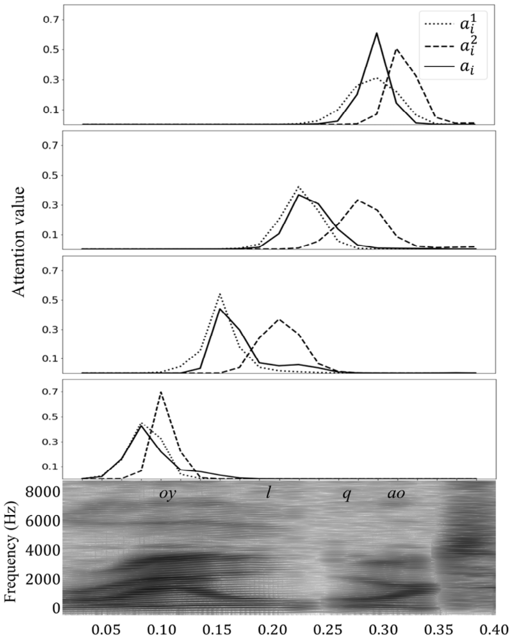

Fig. 5. Alignment examples of four phonemes "oy", "l", "q" (glottal stop), and "ao" in the middle of words "broil or". Solid lines (a_i) represent the alignment vector produced by the conventional single-attention mechanism. Dotted lines (a_i^1) and dashed lines (a_i^2) represent the left and the right alignment vectors produced by the proposed double-attention mechanism.

Table 1 shows the Phone Error Rate (PER) of the speech recognition systems using various attention mechanisms of the sequence-to-sequence models. As shown in the table, the proposed double-attention mechanism with the multiplicative score function reduces the PER by 4 % relatively compared to the baseline system, which uses the conventional single-attention mechanism. When the

Table 1. Performances of various attention mechanisms and score functions for the TIMIT phone recognition task.

Attention	Score function	PER (%)
Single-attention	Eq. (11)	16.8
Double-attention	Eqs. (13) and (17)	16.5
Double-attention	Eqs. (20) and (21)	16.1

attention mechanism is replaced with the multi-head attention method,[10] the PER rises to 17.3 %.

V. Conclusions

We herein proposed a double-attention mechanism for the sequence-to-sequence deep neural networks to handle large input focuses with changing characteristics. Furthermore, the multiplicative score function with the location-aware feature was proposed to better utilize the left and the right contexts of the input. The experimental results of the speech recognition task on the TIMIT corpus verified that the proposed double-attention mechanism indeed attended to the left and the right contexts of the input focus and reduced the speech recognition error rates.

The conventional speech recognition systems with Hidden Markov Models (HMM) typically uses three states for a phone model. This corresponds to a triple-attention mechanism in the sequence-to-sequence deep neural networks, which will be our future research direction.

Acknowledgements

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2017R1E1A1 A01078157). Also, it was partly supported by the MSIT (Ministry of Science and ICT) under the ITRC (Information Technology Research Center) support program (IITP-2018-0-01405) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and IITP grant funded by the Korean government (MSIT) (No. 2018-0-00269, A research on safe and convenient big data processing methods).

References

1. I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” Proc. Int. Conf. NIPS. 3104-3112 (2014).
2. D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv:1409.0473 (2014).
3. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: neural image caption generation with visual attention,” Proc. ICML. 2048-2057 (2015).
4. S. Watanabe, T. Hori, S. Kim, J. Hershey, and T. Hayashi, “Hybrid CTC/attention architecture for end-to-end speech recognition,” IEEE J. Selected Topics in Signal Processing, **11**, 1240-1253 (2017).
5. H. Soltau, H. Liao, and H. Sak, “Neural speech recognizer: acoustic-to-word LSTM model for large vocabulary speech recognition,” Proc. Interspeech, 3707-3711 (2017).
6. K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, “Building competitive direct acoustics-to-word models for English conversational speech recognition,” Proc. IEEE ICASSP. 4759-4763 (2018).
7. C. Chiu, T. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” Proc. IEEE ICASSP. 4774-4778 (2018).
8. J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” Proc. Int. Conf. NIPS. 577-585 (2015).
9. W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: a neural network for large vocabulary conversational speech recognition,” Proc. IEEE ICASSP. 4960-4964 (2016).
10. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaizer, and I. Polosukhin, “Attention is all you need,” Proc. Int. Conf. NIPS. 5998-6008 (2017).
11. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, **9**, 1735-1780 (1997).
12. K. Greff, R. Srivastava, J. Koutnik, B. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” IEEE Trans. on Neural Networks and Learning Systems, **28**, 2222-2232 (2017).
13. Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time-series,” *in Handbook of Brain Theory and Neural Networks*, edited by M. A. Arbib (MIT Press, 1995).
14. O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” IEEE/ACM Trans. on Audio, Speech, and Language Processing, **22**, 1533-1545

(2014).
15. D. Lim, *Improving seq2seq by revising attention mechanism for speech recognition*, (Dissertation, Korea University, 2018).
16. Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," Proc. IEEE ICASSP. 4845-4849 (2017).

Profile

▸ Dongsuk Yook (육동석)

Dongsuk Yook received his B.S. and M.S. degrees in computer science from Korea University, Seoul, Korea, in 1990 and 1993, respectively, and Ph.D. degree in computer science from Rutgers University, New Jersey, USA, in 1999. He worked on speech recognition at the IBM T.J. Watson Research Center, New York, USA, from 1999 to 2001. Currently, he is a professor in the Department of Computer Science and Engineering, Korea University, Seoul, Korea. His research interests include machine learning and speech processing.

▸ Dan Lim (임단)

Dan Lim graduated with a B.E. and M.S. in electrical engineering and computer science from Korea University in 2016 and 2018, respectively. He is now a researcher in the R&D Center of Kakao Corp. His research interest includes machine learning, speech recognition, and speech synthesis.

▸ In-Chul Yoo (유인철)

In-Chul Yoo received his B.S., M.S., and Ph.D. degrees in computer science from Korea University, Seoul, Korea, in 2006, 2008, and 2015, respectively. Currently, he is a research professor in the Artificial Intelligence Laboratory at Korea University. His research interests include robust speech recognition and speaker recognition.