*INTERSPEECH 2017*
August 20–24, 2017, Stockholm, Sweden

# Gaussian Prediction based Attention for Online End-to-End Speech Recognition

*Junfeng Hou, Shiliang Zhang, Lirong Dai*

National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, P. R. China
{hjf176,zsl2008}@mail.ustc.edu.cn, lrdai@ustc.edu.cn

## Abstract

Recently end-to-end speech recognition has obtained much attention. One of the popular models to achieve end-to-end speech recognition is attention based encoder-decoder model, which usually generating output sequences iteratively by attending the whole representations of the input sequences. However, predicting outputs until receiving the whole input sequence is not practical for online or low time latency speech recognition. In this paper, we present a simple but effective attention mechanism which can make the encoder-decoder model generate outputs without attending the entire input sequence and can apply to online speech recognition. At each prediction step, the attention is assumed to be a time-moving gaussian window with variable size and can be predicted by using previous input and output information instead of the content based computation on the whole input sequence. To further improve the online performance of the model, we employ deep convolutional neural networks as encoder. Experiments show that the gaussian prediction based attention works well and under the help of deep convolutional neural networks the online model achieves 19.5% phoneme error rate in TIMIT ASR task.

**Index Terms**: Automatic Speech Recognition, Encoder-Decoder, Online, Gaussian Prediction based Attention, Deep Convolutional Encoder

## 1. Introduction

End-to-end models are rapidly introduced to many areas as soon as they were proposed. The attention based encoder-decoder model is a popular one and yields competitive or state-of-the-art results in tasks like machine translation (MT)[1][2][3], image caption[4][5], automatic speech recognition (ASR)[6][7][8] and so on. However, most of the attention models need to attend the entire input sequences before producing outputs constrained by the attention mechanism, which is unacceptable for online speech recognition. Apparently, for online or low time latency recognition, the model should generate output symbols conditioned only on part of the input sequence as the input speech stream goes on.

Some works have been reported for encoder-decoder ASR. Bahdanau et al.[6][9] extended the content-based attention mechanism of the basic model[1] to be location-aware by making it take into account the alignment produced at the previous step. They also adopted a moving windowed attention to lower computational complexity, which means one output symbol is only related to part of the speech and the alignments between input speech and output sequence are monotonous. However, the window size is fixed and may be not proper for speech with variable speed. Zhang[8] replaced the shallow gated recurrent unit (GRU) encoder with very deep convolutional neural networks (CNNs) and obtained significant improvements over previous shallow encoder-decoder speech recognition models. Although the models with various structures have been discussed, very few works consider the online end-to-end speech recognition problem. Neural transducer model[10] was proposed recently to compute the next-step output distribution conditioned on the partially observed input sequence and the partially generated sequence. However, the model requires inferring alignments between the input blocks and output chunks during training, which are generated by other algorithms such as dynamic programming and are hard to be trained jointly with the transducer. A sliding window[11] is used to explore the online encoder-decoder models while the window size is pre-defined.

In this paper, we propose a gaussian prediction based attention method for encoder-decoder model, which computing the attention without attending the input sequence representations at each step. The alignment between output and input sequence is assumed to be a time-moving gaussian window with variable size and the location and size of the window are decided by its mean and variance. At each attention step, the mean and variance are predicted by the previous decoder state. To model the monotonous character of the speech, we predict the window's moving forward increment along time from the previous window center, rather than predicting the absolute position of the window. As a bonus, the gaussian prediction based attention method can be trained faster because it is simpler and does not need extra matrix multiplications.

To compensate the representative ability of unidirectional GRU (uniGRU) encoder, we use deep CNNs[8][12][13][14] to replace the GRU encoder. Since CNN only requires limited frames of context in both directions, it is natural to apply CNN to the online encoder-decoder models. Meanwhile CNN explicitly exploits structural locality in the spectral feature space and has better spectral and temporal invariance properties. On the TIMIT phoneme recognition task, we achieve 20.4% phoneme error rate (PER) with gaussian prediction based attention and bring the PER down to 19.5% with deep CNNs encoder.

## 2. Related Work

Recently many sophisticated architectures have been proposed to fit specific tasks based on the basic encoder-decoder models. In ASR, phones and words are only related to a few frames of the speech, which makes the window restriction of the attention area reasonable[6][9]. In MT, a source language phrase is always aligned to a target language phrase uniquely, which means that the model only needs to focus on a fraction of the input sentence and can neglect other confusing words at each time of translation[2]. In [6][9][11], the attention module only attends input frames within a rectangular window whose position is determined by previous alignments and the window size



http://dx.doi.org/10.21437/Interspeech.2017-751

is pre-defined. And in [2], the attention is masked by a gaussian window whose absolute position is predicted by the decoder state and size is also pre-defined.

To resolve the time latency problem of the encoder-decoder model, Jaitly[10] introduced so-called neural transducer model. In [10], by splitting the input and output sequence into blocks, the next-step output distribution is only conditioned on the partially observed input sequence and partially generated sequence. However, the alignments between output symbols and input sequences which are used to infer output chunks are unavailable and should be generated from a different algorithm like dynamic programming. The alignments are approximate and the inferring algorithm is hard to be trained jointly with the model.

Meanwhile, CNNs have also been deployed widely in ASR[8][13][14][15] and deeper CNN architectures often yield better performance. For encoder-decoder model, the recurrent encoders like GRU have also been replaced by CNNs for ASR task[8] and MT task[16][17].

Our work is inspired by the works in [2][6][9][11], with the main difference being that the proposed gaussian based attention is a sliding window with variable size and does not need to attend the whole input sequence as [2] does. Compared with [10], our proposed attention computation is simpler and has no trouble in jointly training the model and the alignment. Furthermore, inspired by [8], CNN based encoder is adopted for its ability of few future context modelling.

# 3. Methods

## 3.1. Basic Architecture

The attention based encoder-decoder model contains two parts : an encoder to process input sequence $X = (x_1, x_2, \ldots, x_L)$, a decoder to generate output sequence $Y = (y_1, y_2, \ldots, y_T)$ with an attention module to align the sequence $X$ and $Y$. The encoder is usually stacked GRU or long short-term memory (LSTM) that converts the input sequence $(x_1, x_2, \ldots, x_L)$ to hidden representations $H = (h_1, h_2, \ldots, h_L)$. The decoder is a recurrent network that attends the input sequence representations and generates outputs based on both the attended representation $c_t$ and the decoder state $s_t$.

When applying encoder-decoder model to ASR, the input $X$ is usually a sequence of feature vectors like filter banks, and the output $Y$ is a sequence of phonemes or words. At the $t$-th step of prediction, the decoder generates an output $y_t$ by attending the encoder representations $H$:

$$
\begin{aligned}
H &= \text{Encoder}(X) && (1)\\
s_t^{'} &= \text{Recurrent}(s_{t-1}, y_{t-1}) && (2)\\
\alpha_t &= \text{Attend}(s_t^{'}, \alpha_{t-1}, H) && (3)\\
c_t &= \sum_{i=1}^{L} \alpha_t^i h_i && (4)\\
s_t &= \text{Recurrent}(s_t^{'}, c_t) && (5)\\
y_t &= \text{Softmax}(f(s_t)) && (6)
\end{aligned}
$$

Where $s_t$ is the state of the decoder at time $t$ and $s_t^{'}$ is the intermediate variable. The attention vector is $\alpha_t \in \mathbb{R}^L$, known as the alignments of input frames and output sequences. The Attend() function in Equation (3) is often modeled by a multiple layer perceptron (MLP) shown below:

$$
\begin{aligned}
e_t^i &= \text{MLP}(s_t^{'}, h_i, [F * \alpha_{t-1}]^i) && (7)\\
\alpha_t^i &= e_t^i \Big/ \sum_{i=1}^{L} e_t^i && (8)
\end{aligned}
$$

## 3.2. Gaussian Prediction based Attention

Based on Equation (3), (7) and (8), the decoder should look at all the hidden representation $H$ outputted from encoder before deciding what to focus on, which is unsuitable for online ASR task. So we modify the attention module by using a gaussian window to represent the alignment. The equations for gaussian prediction based attention are:

$$
\begin{aligned}
\Delta p_t &= S_w \cdot \text{sigmoid}(V_p^T \tanh(W_p s_t^{'})) && (9)\\
p_t &= \Delta p_t + p_{t-1} && (10)\\
\sigma_t &= D_w \cdot \text{sigmoid}(V_\sigma^T \tanh(W_\sigma s_t^{'})) && (11)\\
e_t^i &= \exp(-\frac{(i-p_t)^2}{2\sigma_t^2}), \quad i = 1, 2, \cdots, L && (12)\\
\alpha_t^i &= e_t^i \Big/ \sum_{i=1}^{L} e_t^i && (13)
\end{aligned}
$$

Where $p_t$ and $\sigma_t$ are the predicted mean and variance which specify the location and size of the gaussian window respectively. $S_w$ is the maximum moving forward increment the model should have and $D_w$ is the maximum focusing area the model should attend. Note that the index $i$ in Equation (12) represents the frame number, and the frames with larger gaussian window values are the most possibly focusing areas.

The Equation (9) and (10) mean that the previous window center $p_{t-1}$ moves forward to the current attending center $p_t$ with the moving forward increment $\Delta p_t$ predicted by a MLP given the previous decoder state. What's more, the $\sigma_t$ which decides the window size of the attention is also predicted by the previous decoder state as shown in Equation (11). This attention mechanism helps the encoder-decoder model to move its focus forward explicitly and can handle the online speech recognition naturally even for long input sequences. Meanwhile our experiments also demonstrate that predicting a relative moving forward increment of the attention window is more effective than predicting an absolute window center like [2] for offline speech recognition with encoder-decoder model.

Considering the online requirement and the fact that the gaussian window value becomes very small when the frame number offset from the window center is large(see Equation (12)), the Equation (12) in above proposed attention module is further modified as given in Equation (14). In the equation, $K$ decides the time latency of the model. And all the future inputs beyond the scope $[1, \text{floor}(p_t + K\sigma_t)]$ are not allowed to be accessed at $t$-th prediction step.

$$
e_t^i = \left\{ \begin{array}{ll} \exp(-\frac{(i-p_t)^2}{2\sigma_t^2}), & i \in [1, \text{floor}(p_t + K\sigma_t)] \\ 0, & others \end{array} \right\} \quad (14)
$$

## 3.3. Deep CNNs Encoder

The usually adopted Encoder() in Equation (1) is bi-directional gated recurrent unit (biGRU). However, the encoder of the online encoder-decoder model should process the input sequence

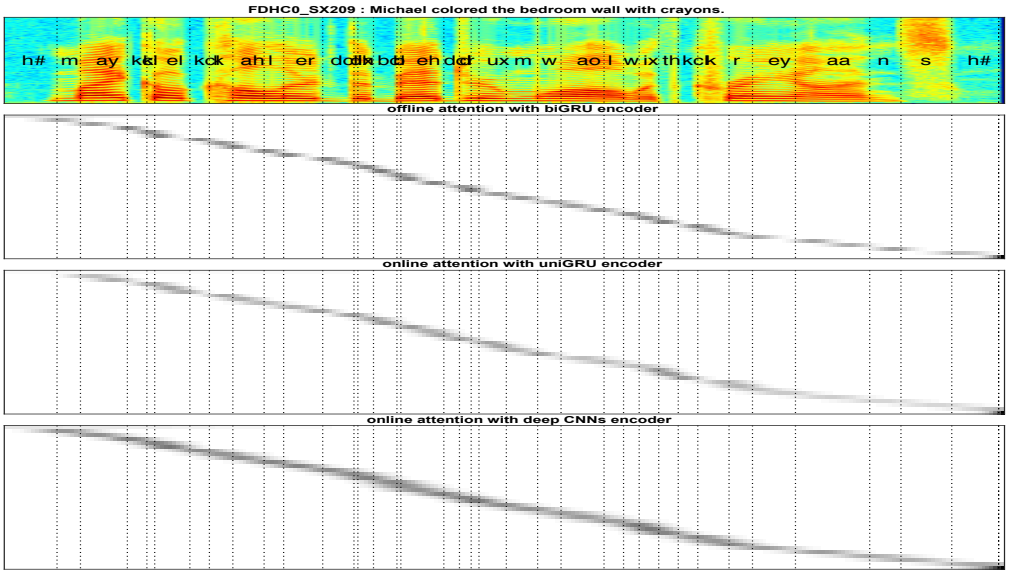


Figure 1: *Offline and Online attention visualization for different encoder-decoder models. The dashed lines indicate ground truth phone segments. Each row of the attention images indicates frames selected by the attention mechanism to emit a phone symbol.*

in real time or low time latency as long as the input arrives, which makes biGRU not applicable. A natural substitution is CNN because of its limited requirement for the future context. Meanwhile, a lot of works[8][13][14][15][16][17][18] have proven the effectiveness of the deep CNNs containing two or more layers for image classification and speech recognition.

In this paper, we use deep CNNs as encoder to strengthen the online encoder-decoder model. The architecture of the deep CNNs is similar to [12], where deep CNNs are a succession of several convolutional blocks and each block contains more than one convolutional layers with Rectified Linear Unit (ReLU)[19] activation function and max pooling.

## 4. Experiments and Analyses

We conduct all experiments in TIMIT dataset. The input features for GRU encoder are 40 mel-scale filter banks together with the energy in each frame, and their first and second temporal differences, yielding in total 123 features per frame. As for CNN encoder, the frequency dimension of the input is 41 and static, first and second temporal differences in each frame are stacked as 3 feature maps. The output is 61-phone set with an extra "end-of-sequence" token. All other experimental settings are similar to [6] if not explained.

We use adadelta optimizing algorithm with $\epsilon = 10^{-8}$ and train the model without regularization first. After lowest development negative log-likelihood is achieved, we continue to train the model with adaptive weight noise[20] and lower the $\epsilon$ to $10^{-9}$ if we do not observe the performance improvement in development PER. Finally we stop training if there is no more gain in development set. Our minibatch size is 5 and $\rho$ is 0.95.

One thing should be mentioned is that when training models with deep CNNs encoder we use SGD with momentum[21] to tune the model after the adadelta training stage as it yields better results. The learning rate of SGD is $2 \times 10^{-5}$ and momentum is 0.5. All the models are implemented based on theano[22].

### 4.1. Effectiveness of Gaussian Prediction based Attention

First of all, we verify the predicted attention module's effectiveness for offline speech recognition task, where encoder is biGRU. The attention employed is shown in Equation (12),(13). For offline task, the encoder is 3 layer stacked biGRU and each layer has 256 nodes for each direction. The decoder is one single layer uniGRU with hidden size 256, similar to [6]. The MLP to predict the mean and variance of the gaussian window also has 256 hidden nodes. We use two kinds of gaussian prediction based attentions for offline recognition. One is shown in Equation (10) named as relatively predicted attention and another is similar to [2] except the variance of the gaussian window is also predicted here, which is called absolutely predicted attention. For relatively predicted attention $S_w$ is set to 30 and for both of them $D_w$ is set to 100. Experimental results are in table 1.

Table 1: *Test set PER for gaussian prediction based attention with biGRU encoder*

| Type | Model | PER |
|---|---|---|
| Offline | baseline[6] | 17.6 |
| | biGRU + relatively predicted attention | **18.2** |
| | biGRU + absolutely predicted attention | 18.5 |

From the results, we can see that the relatively predicted attention is better than absolutely one and is only a little worse than the state-of-the-art encoder-decoder model. In fact the absolutely predicted attention is not only worse in performance but also unsuitable for online recognition because it must know the input sequence length in advance. We also visualize the alignments the relatively predicted attention generates for offline speech recognition and show it in figure 1. As we can see, the alignments the relatively predicted attention generates (the second row with title "offline attention with biGRU encoder") are well corresponding to the spectrogram segments (the first row with dashed lines as ground truth segments). So we can expect that the relatively predicted attention will also work well for online speech recognition with encoder-decoder model.

### 4.2. Online Speech Recognition with Gaussian Prediction based Attention and uniGRU Encoder

We use uniGRU as encoder and the gaussian prediction based attention employed here is given in Equation (14),(13). For online task, the encoder is 3 layer stacked uniGRU and each layer has 256 hidden nodes. The decoder is one single layer uniGRU with hidden size 256. $S_w$ is set to 50 and $D_w$ is set to 100. The hyper-parameter $K$ is used to control the time latency.

We test different K values in this experiment and the results are given in table 2. According to the results in the table 2, we decide to choose $K = 3$ for all the online experiments afterwards because it seems that time latency $3\sigma_t$ or 100 yields similar results. The figure 2 shows the predicted $(\Delta p_t, \sigma_t)$ values for the test set (left) with gaussian prediction based attention and uniGRU encoder. From the image, we can observe that, for most time steps, the attention only attends a few frames in the future which means the time latency is low. And the average values of $(\Delta p_t, \sigma_t)$ are $(8.3, 4.3)$ for uniGRU encoder, which may be explained by that most phone segment lengths are small in TIMIT dataset. We hypothesize that the model may handle speech with different speeds by predicting different moving forward increments flexibly.

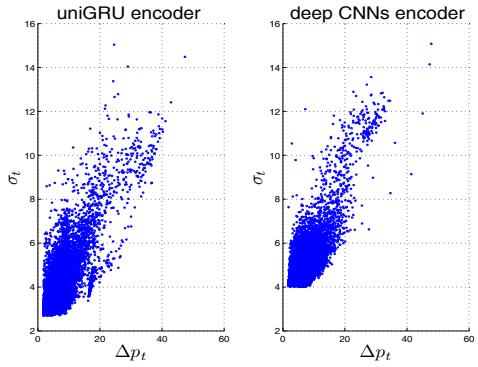


Figure 2: *$(\Delta p_t, \sigma_t)$ for online speech recognition with relatively predicted attention and uniGRU and deep CNNs encoder*

Table 2: *Test set PER of relatively predicted gaussian attention for uniGRU encoder with different time latency*

| K$\sigma_t$ | $\sigma_t$ | $2\sigma_t$ | $3\sigma_t$ | 100 |
|---|---|---|---|---|
| PER | 22.01 | 21.5 | 20.4 | 20.3 |

### 4.3. Online Speech Recognition with Gaussian Prediction based Attention and deep CNNs Encoder

We also employ deep CNNs as encoder to improve the online speech recognition performance. The deep CNNs' architecture is {3-[128-128]-[256-256]-[512-512]} which means the input has 3 feature maps and the deep CNNs encoder contains 3 convolutional blocks. Each block has 2 convolutional layers with ReLU and the feature maps for each block are 128, 256 and 512 respectively. Each block is followed by a max pooling layer in frequency dimension with size 2 and no time pooling for all blocks. We apply zero padding of size 1 at every side before every convolution and the filter size is 3x3 which is similar to [12]. The results are given in table 3.

As we can see, the proposed gaussian prediction based attention with uniGRU works better than [10]. The partial conditioning model with additional GMM-HMM alignment information can get better result but we don't compare with this result for fairness. Deep CNNs encoder works better than uniGRU because it encodes future context for each frame when input sequence is processed. With $K\sigma_t = 100$ the deep CNNs encoder yields better result with PER 19.4% for online recognition. More CNN layers can be used and may yield better results, but we haven't tried that yet in this work.

Table 3: *Test PER for relatively predicted gaussian attention with uniGRU encoder and deep CNNs encoder*

| Type | Model | PER |
|---|---|---|
| Online | uniGRU + partial conditioning[10] | 20.8 |
| | uniGRU + relatively predicted attention | 20.4 |
| | deepCNNs + relatively predicted attention | **19.5** |

We also visualize the alignments generated by our proposed gaussian prediction based attention for online speech recognition with uniGRU encoder (third row) and deep CNNs encoder (last row) in figure 1. In the figure, the online attention with uniGRU is delayed comparing with offline attention with biGRU (second row) because uniGRU can not obtain future context for current frame. And the CNN attention is less sharp than biGRU, which is related to the amount of context information the encoder obtains. Although CNN can compose future context with convolution, the context range is limited which makes the model predict wider but less sharp attention to get more encoded frames for each phone. A much deeper CNNs encoder may relieve that and improve the result further, which is our future work. From figure 2 we can also observe deep CNNs encoder has bigger $\sigma_t$ values than uniGRU. We hypothesize that it may be related to the delayed attention of uniGRU encoder. As the attending center is not in the corresponding phone segment, the attention should not be too wide to involve confusing frames.

One thing should consider is the pre-defined $S_w$. In this work, the model should generate an output symbol for every window increment, which may be not proper for speech with long segment size exceeding the allowed maximum moving forward increment $S_w$. So we try online experiments with larger $S_w$ from 50 to 80 to examine the model's robustness to $S_w$. The accuracy improves a little with bigger $S_w$ so that we can choose a larger $S_w$ in case the long segment occurs. However, long segment phones are unusual in TIMIT so we plan to try this model for dataset with long segment speech in the future.

## 5. Conclusions and Future Work

We introduce a simple but effective attention method to touch the online ASR task with encoder-decoder model. By assuming the alignment between input speech frames and output phone sequences is a gaussian window, we can predict the attention vector at each step without attending the input sequence and get comparable results for offline model. What's more, by this attention mechanism and deep CNNs encoder, the encoder-decoder model yields better results for online speech recognition. In the future we may try more attention prediction methods and verify the performance for variable speed speech.

## 6. Acknowledgements

The authors would like to acknowledge the support of National Natural Science Foundation of China grant No. U1613211.

# 7. References


[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[2] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[3] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[5] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention." in *ICML*, vol. 14, 2015, pp. 77–81.

[6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[7] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[8] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," *arXiv preprint arXiv:1610.03022*, 2016.

[9] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[10] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems*, 2016, pp. 5067–5075.

[11] W. Chan and I. Lane, "On online attention-based speech recognition and joint mandarin character-pinyin training," *Interspeech 2016*, pp. 3404–3408, 2016.

[12] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4955–4959.

[13] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*. IEEE, 2013, pp. 8614–8618.

[14] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Interspeech 2016*, 2016, pp. 410–414.

[15] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[16] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models." in *EMNLP*, vol. 3, no. 39, 2013, p. 413.

[17] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." in *Aistats*, vol. 15, no. 106, 2011, p. 275.

[20] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, 2011, pp. 2348–2356.

[21] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning." *ICML (3)*, vol. 28, pp. 1139–1147, 2013.

[22] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proc. 9th Python in Science Conf*, 2010, pp. 1–7.