Deformable TDNN with adaptive receptive fields for speech recognition

Keyu An, Yi Zhang, Zhijian Ou[†]

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China

aky19@mails.tsinghua.edu.cn, zhangyi18@mails.tsinghua.edu.cn, ozj@tsinghua.edu.cn

Abstract

Time Delay Neural Networks (TDNNs) are widely used in both DNN-HMM based hybrid speech recognition systems and recent end-to-end systems. Nevertheless, the receptive fields of TDNNs are limited and fixed, which is not desirable for tasks like speech recognition, where the temporal dynamics of speech are varied and affected by many factors. In this paper, we propose to use deformable TDNNs for adaptive temporal dynamics modeling in end-to-end speech recognition. Inspired by deformable ConvNets, deformable TDNNs augment the temporal sampling locations with additional offsets and learn the offsets automatically based on the ASR criterion, without additional supervision. Experiments show that deformable TDNNs obtain state-of-the-art results on WSJ benchmarks (1.42%/3.45% WER on WSJ eval92/dev93 respectively), outperforming standard TDNNs significantly. Furthermore, we propose the latency control mechanism for deformable TDNNs, which enables deformable TDNNs to do streaming ASR without accuracy degradation.

Index Terms: speech recognition, deformable convolution, TDNN, adaptive receptive fields, neural architecture.

1. Introduction

Time Delay Neural Networks (TDNNs), also known as one-dimensional Convolutional Neural Networks (1-D CNNs), are widely used in both DNN-HMM based hybrid speech recognition systems [1, 2] and recent end-to-end systems [3, 4]. For speech recognition tasks, TDNNs and their variants have several nice properties. First, due to the pure feed-forward nature, the computation in training TDNNs can be easily paralleled. The training of TDNNs is much faster than recurrent neural networks, especially when using GPUs. Second, it is more convenient to deploy TDNNs for online streaming speech recognition. In TDNNs, future context information, which is essential for accurate acoustic modeling, is provided by temporal convolution over the future frames. The low-latency of TDNNs is determined by the kernel size and the number of layers, which can be easily implemented. In contrast, neural architectures like bidirectional LSTMs, where the future context is provided by the whole-utterance backward LSTMs, and self-attention encoders, which need the entire utterance as input, are not naturally suited to low-latency speech recognition, although there have some modifications and progress [5, 6].

Nevertheless, due to the fixed geometric structures, standard TDNNs are inherently limited in modeling the complex transformations of the input features (MFCC, filter bank, and PLP, etc.) needed for speech recognition. To be specific, the patterns of receptive fields [1] of all neural units in a TDNN layer

---

[†] Corresponding author.

[1] In TDNNs, a window of input features around the current position are used to produce the output for the current neural unit. By convention we call the window of features as the receptive field of the current unit.

are the same, which means that for all neural units in a TDNN layer, input features with **fixed** positions and windows are used to predict linguistic states such as phonemes or characters (Figure 1, left). This is undesirable for tasks like speech recognition, where the temporal windows of speech frames covering different linguistic states are varied and affected by many factors, such as the type of phonemes (vowel v.s. consonant), the gender and the pronunciation habit of the speaker, co-articulation, and so on [7]. To address this, multi-stream models [7, 8] propose to accommodate diverse temporal resolutions in multiple streams to achieve robustness in temporal dynamics modeling. However, in the above-mentioned multi-stream models, the receptive fields for all neural units are pre-designed by human experts, thus can not automatically adapt to the variations in the input. Moreover, multi-stream models significantly increase the number of model parameters.

This paper presents deformable TDNNs (Figure 1, right) as a neural network architecture for adaptive receptive fields in speech recognition. The idea of deformable TDNN is inspired by deformable convolutional networks (deformable ConvNets) [9], which was proposed to enhance CNNs' capability in modeling geometric transformations for sophisticated vision tasks such as object detection and semantic segmentation. Basically, a deformable TDNN can be seen as a 1-D version of deformable ConvNets. Similar to deformable ConvNets, deformable TDNNs augment the temporal sampling locations with additional offsets. The offsets are predicted by an additional TDNN conditioned on the input, thus enabling temporal modeling in a self-adaptive manner. The parameters of the additional TDNN are optimized using the loss for the target task, without additional supervision. Our extensive experiments show that deformable TDNNs improve over standard TDNNs significantly and consistently, without the cost of expensive training time, large overhead over model parameters, and complex hyper parameter designs.
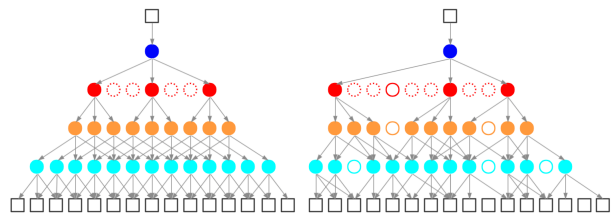

Figure 1: *Illustration of the fixed receptive field in standard TDNNs (left) and the adaptive receptive field in deformable TDNNs (right). Dotted circles denote the dilation between kernel elements.*

It is important to note that similar modules have been used in action segmentation [10] and sign language recognition [11] in videos. The contributions of this work are: 1) We demonstrate that adaptive temporal modeling is crucial for speech recognition by comparing the performances of several neural

architectures with different receptive fields. 2) We show that deformable TDNNs can replace standard TDNNs easily in existing neural architectures, and produce remarkable performance gains, especially when the input is perturbed in the time direction. 3) We introduce the latency control mechanism for deformable TDNNs, which enables deformable TDNNs to do streaming ASR in practical scenarios.

The rest of the paper is organized as follows. Section 2 outlines related work, especially focusing on the comparison of receptive fields used by different neural architectures. Section 3 describes the deformable TDNNs in details. Section 4 describes the experiment setup. Section 5 presents and analyzes the results primarily on the WSJ task. In Section 6, we discuss the receptive fields of deformable TDNNs, the distribution of the offsets, and the latency control mechanism. Section 7 presents the conclusions and the future work. We will release the code in the CAT toolkit [12] upon the acceptance of the paper.

2. Related work

2.1. Receptive fields of neural architectures

The receptive fields of different neural architectures are different by their nature. In convolutional neural networks (e.g. 2-D CNNs and TDNNs), the receptive field is associated with the kernel sizes and dilation rates [7, 8]. The FSMN model [13] is another feedforward architecture. In FSMN, context information is encoded into a fixed-size memory block, and the receptive field is defined by the number of historical items looking back to the past, and the size of the lookahead window into the future. In self-attention layers [14], the output at each time step depends on the entire input feature map, with attention weights conditioned on the input as well. For RNNs, the receptive field can be seen as the input feature at and before the current time step, due to the recurrent manner. Similarly, the output of bidirectional RNNs relies on the entire input feature map. In LSTMs [15] and GRUs [16], two variants of RNN, the use of forget gate allows a dynamically changing contextual window over all of the input history. Extensive results show that LSTMs and GRUs improve significantly over vanilla RNNs.

The effect of neural architecture's receptive field has been explored in many previous works. In [1], the input time steps, at which activations are computed at each layer, are manually designed to model the temporal dependencies. In [4], the hyperparameters, which control the input contexts for each TDNN layer (the kernel size and dilation), are selected by Neural Architecture Search (NAS). The above-mentioned two works show that the selection of input contexts, either by human experts or by NAS, has a remarkable effect on the model performance. Time-restricted self-attention mechanism [17] restricts the input context within a local region, and suggests that too wide or too narrow a context might degrade the results. [18] proposes a novel fully-trainable windowed attention mechanism to restrict attention to a small input time span. In windowed attention, the window size is learned instead of pre-set by rules. For recurrent neural networks, Soft forgetting [19] and Contextualized Soft forgetting [12] propose to unroll the BLSTM network over chunks (without or with overlaps) instead of the whole utterance, thus limiting the dependencies between input and output within the chunks. Results show that such restrictions bring considerable recognition accuracy improvements in addition to realizing streaming, presumably because chunk-based models alleviate overfitting.

2.2. Multi-resolution networks

There have been efforts in considering the varieties of temporal patterns and achieving robust acoustic modeling by accommodating diverse temporal information in multiple streams. In multistream CNNs [8], each stream stacks TDNN-F layers whose kernel has a unique, stream-specific dilation rate. In multi-stride self-attention [20], each group of heads in self-attention process speech frames with a unique stride over neighboring frames. Nevertheless, the hyper-parameters (the dilation rate and the stride) in previous multi-resolution networks are pre-designed by human experts and can not adapt to the input. Moreover, multi-resolution networks typically result in a significant increase in the number of parameters.

2.3. Deformable neural networks

Deformable convolution [9] was initially introduced in computer vision tasks to enhance the transformation modeling capability of CNNs. In action segmentation [10] and sign language recognition [11], where the inputs are organized as a sequence, deformable convolutions were used to model the variations in temporal contexts. Recently, deformable CNNs were applied to the speaker verification task to deal with the irregular structure in feature maps [21]. To the best of our knowledge, there was no related work on applying deformable neural networks in speech recognition.

3. Deformable TDNNs

3.1. Method

A standard TDNN [22] consists of two steps: 1) sampling with fixed 1-D grid $\mathcal{R}$ over the input feature map x; 2) summation of sampled values weighted by w. The 1-D grid defines the receptive field size and dilation. For example, a TDNN layer with kernel size 3 and dilation 2 can be described as:

$$y(t_0) = \sum_{t_n \in \mathcal{R}} w(t_n) * x(t_0 + t_n) \quad (1)$$

where $\mathcal{R} = \{-2, 0, 2\}$.

Similar to deformable convolutions, in a deformable TDNN, the regular 1-D grid $\mathcal{R}$ is augmented with offsets $\{\Delta t_n | n = 1, ..., N\}$ [2], where $N = |\mathcal{R}|$. Eq. 1 becomes

$$y(t_0) = \sum_{t_n \in \mathcal{R}} w(t_n) * x(t_0 + t_n + \Delta t_n) \quad (2)$$

where $t = t_0 + t_n + \Delta t_n$ can be fractional, as Δt_n is typically fractional. Thus, $x(t)$ is calculated via linear interpolation as

$$x(t) = x(\lfloor t \rfloor) * (\lfloor t \rfloor + 1 - t) + x(\lfloor t \rfloor + 1) * (t - \lfloor t \rfloor) \quad (3)$$

where $\lfloor \rfloor$ denotes the floor function, defined as giving the greatest integer less than or equal to the input.

Consider an input sequence of size (C_{in}, T), where C_{in} is the dimension of the feature vector, and T is the length of the feature sequence (e.g. the number of frames). Then, the offsets of the deformable TDNN are of size (N, T) (Figure 2), predicted by an additional 1-D Conv with kernel size N'. The number of additional parameters introduced by the offset predicting TDNN is small ($N * C_{in} * N'$), compared with the parameters of regular 1-D Conv ($C_{out} * C_{in} * N$), as N' is much smaller than C_{out} typically [3].

---

[2] In 1-D convolution, N equals to the kernel size. For notation brevity, we omit the dependence of Δt_n on t_0.

[3] In our experiment, $N' = 5$ and $C_{out} = 640$ by default .

As illustrated in Figure 2, the offsets determine the kernel pattern of the deformable TDNN, and the kernel pattern is optimized at the same time as the kernel weights, without additional supervision. The deformation of the kernel pattern depends on the input features, which enables adaptive adjustment according to the distortions of the input. In this sense, deformable TDNN shares similar high-level spirit with self-attention network [23].
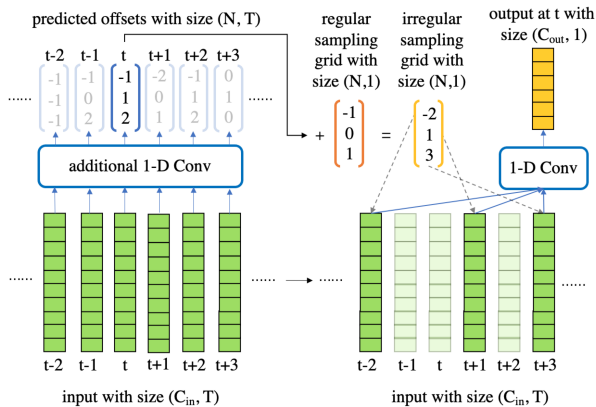

Figure 2: *Illustration of a deformable TDNN with kernel size 3. For the sake of simplicity, we set the predicted offsets to be integers in this figure.*

3.2. Comparison with deformable ConvNets

Given a feature map (the input spectrogram or the hidden layers in a TDNN) with T time steps, we can view it as an image where the time axis is horizontal and the frequency axis is vertical. When computing the output of a deformable TDNN at each time step, all points along the vertical axis share the same offsets. Thus, the spatial resolution of the offset is much smaller than the input feature map, which leads to fewer additional parameters and computing costs. Moreover, the offsets are constrained to have only 1 direction (horizontal). In other words, the offsets along the vertical axis are prohibited, as our motivation is to enhance the model's capability in modeling temporal dynamics. This is different from deformable ConvNets, where the offset fields have the same spatial resolution as the input feature map, and the offsets have 2 directions (e.g., for deformable ConvNets, an input of size (H, W) would yield output offsets of size $(2N, H, W)$).

4. Experiment Setup

For running the experiments, we use the CTC-CRF-based ASR Toolkit, CAT [12]. CAT is based on the recently proposed CTC-CRF [24]. Similar to CTC, CTC-CRF uses a hidden state sequence to obtain the alignment between the label sequence and input feature sequence, and the hidden state sequence is mapped to a unique label sequence by removing consecutive repetitive labels and blanks. In CTC-CRF, the posteriori of the hidden state sequence is defined by a conditional random field (CRF), and the potential function of the CRF is defined as the sum of node potential and edge potential. The definition of node potential is the same as CTC, and the edge potential is realized by an n-gram LM of labels. By incorporating the probability of the sentence into the potential, the conditional independence between the hidden states in CTC is naturally avoided. It has

been shown that CTC-CRF outperforms regular CTC consistently on a wide range of benchmarks, and is on par with other state-of-the-art end-to-end models [4, 12, 24].

We choose the neural architecture proposed in [4] as the baseline acoustic model, which is a 7-layer standard TDNN with hyper-parameters (the kernel size and dilation for each layer) chosen by NAS. The configuration of the baseline model is listed in the Table 1. It is a strong enough baseline (Table 4), and we evaluate the effect of deformable TDNNs by replacing the standard TDNN layers with deformable TDNN layers. In a deformable TDNN layer, the additional offset prediction network is a 1-D Convolution layer. We set the kernel size to 5 by default, and we initialize the parameters of the offset predicting network with 0.

Table 1: *Configuration of the baseline TDNN model obtained by NAS.*

layer index	1	2	3	4	5	6	7
kernel size	5	5	5	3	5	5	5
dilation	1	2	2	1	1	1	2
stride	1	1	1	3	1	1	1

Experiments are conducted on 80-hour WSJ dataset. Input features are 40-dimension fbank with delta and delta-delta features (120 dimensions in total). The features are augmented by 3-fold speed perturbation, and are mean and variance normalized.

5. Experiment

Table 2 shows the effect of deformable TDNNs on WSJ. By replacing standard TDNNs in the last 1, 2, 3 layers with deformable TDNNs, recognition accuracy increases remarkably and consistently on both dev93 and eval92, with only a small overhead over model parameters. This indicates that the significant performance improvement is from the capability of modeling temporal dynamics, rather than increasing model parameters. As the evaluation accuracy stops improving when more than 2 deformable TDNN layers are used, we use deformable TDNNs for the last two layers in the remaining experiments.

Table 2: *WER (%) results of using deformable TDNN in the last 1, 2, 3 TDNN layers. The numbers in parentheses denote the relative performance improvements via the use of deformable TDNN layers.*

usage of deformable TDNN (# layers)	# parameters	dev93	eval92
none (0, baseline) [4]	11.90 M	5.68	2.77
layer 7 (1)	11.92 M	**5.37** (5.5%)	2.64 (4.7%)
layer 6,7 (2)	11.93 M	5.48 (3.5%)	**2.50** (9.7%)
layer 5,6,7 (3)	11.95 M	5.54 (2.5%)	2.55 (7.9%)

To evaluate the robustness of deformable TDNNs, we compare the performance of standard TDNNs and deformable TDNNs when the input spectrum is warped along the time axis during testing. Specifically, we employ the time warping operation proposed in SpecAugment [25], where a random point on the feature map is to be warped either to the left or right by a distance w chosen from a uniform distribution from 0 to W (the time warp parameter) along the time axis. To reduce the randomness, we conduct every comparison experiment 10 times with different random seeds. As shown in table 3, deformable TDNNs perform consistently better at adapting to the variations in the input.

Table 3: *WER (%) results on WSJ eval92 when the input feature is warped either to the left or right by a distance w chosen from a uniform distribution $\mathcal{U}(0, W)$*

Time warp parameter W	80	160	240
Standard TDNN	3.10	6.41	11.05
Deformable TDNN	2.98	5.93	10.31

We compare CTC-CRF with deformable TDNNs and other models in Table 4. From Table 4, we find that: 1) CTC-CRF trained with deformable TDNNs improve significantly over CTC-CRF with other neural architectures, including popular architectures such as BLSTM and self-attention encoders. 2) Two popular methods in speech recognition, SpecAugment [25] and RNNLM lattice rescoring [26] further improve the recognition accuracy. 3) With deformable TDNNs, mono-phone CTC-CRF performs significantly better than other end-to-end/hybrid models. Notably, deformable TDNN based CTC-CRF obtains the lowest word error rate (WER) of 1.42%/3.45% on WSJ eval92/dev93 among all published ASR results, to the best of our knowledge.

Table 4: *WER (%) results of CTC-CRF with deformable TDNNs v.s. other models on WSJ.*

	Model	Unit	dev93	eval92
	LF-MMI [27]	mono-phone	6.0	3.0
	LF-MMI [27]	bi-phone	5.3	2.7
	E2E-LF-MMI [27]	mono-phone	6.3	3.1
	E2E-LF-MMI [27]	bi-phone	6.0	3.0
	ESPRESSO [28]	subword	5.9	3.4
	ESPnet [29] VGGBLSTM AM	char	8.8	5.3
	ESPnet [29] transformer AM	char	6.6	4.6
	ConvAM + ConvLM [30]	char	6.8	3.5
	Jasper + transformerLM [31]	char	9.3	6.9
CTC-CRF	BLSTM [24]	mono-phone	6.23	3.79
	VGGBLSTM [12]	mono-phone	5.7	3.2
	self-attention [1]	mono-phone	6.24	3.12
	TDNN-NAS [4]	mono-phone	5.68	2.77
CTC-CRF	Deformable TDNN	mono-phone	5.48	2.50
	+ SpecAugment	mono-phone	5.38	2.46
	+ RNN LM [2]	mono-phone	3.45	1.42

[1] Obtained by our implementation.
[2] The neural network for RNN LM is a 3-layer TDNN interleaved with LSTM layers.

6. Discussion

6.1. Receptive fields

The receptive fields of standard TDNNs and deformable TDNNs are shown in Figure 3. From Figure 3, we observe that: 1) The receptive fields of deformable TDNNs follow the monotonic order in general, which is desired for speech recognition, where the feature inputs and corresponding outputs generally proceed in the same order. 2) Compared with standard TDNNs, the receptive fields of deformable TDNNs are more flexible. The receptive field size for different outputs varies, which is somewhat similar to the alignment patterns of attention models [32].

6.2. Distribution of the offsets

The distribution of predicted offsets on WSJ eval92 sets is shown in Figure 4. We observe that: 1) Although most of the offsets are close to zeros, a small but significant number of offsets are big in magnitude, which could produce noticeable de-
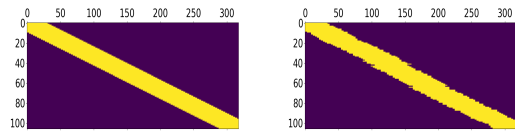

Figure 3: *The receptive fields of standard TDNNs (left) and deformable TDNNs (right). For every grid [i, j], yellow means output[i] depends on input[j], while purple means output[i] is independent of input[j]. The input is three times as long as the output because of the convolution stride. The results are for the utterance (444c040x) in the WSJ eval92 set.*

formations of the kernel patterns. 2) Most predicted offsets are less than 0, indicating that the model tends to pay more "attention" to previous inputs to produce the current prediction.
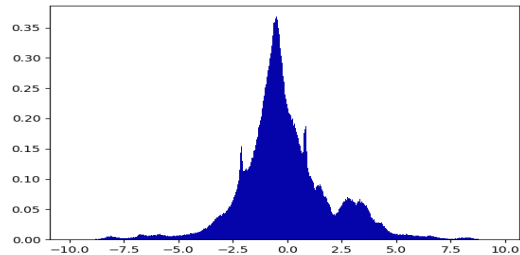

Figure 4: *Distribution of predicted offsets on WSJ eval92 sets.*

6.3. Latency control mechanism

One of the advantages of TDNNs is their predictable and relatively low latency. However, in deformable TDNNs, the additional offsets may lead to unpredictable dependency on future frames, causing uncontrollable latency. A direct solution is to clip the offset during testing, e.g., set the predicted offset to 0 if it is greater than 0. Thus, the usage of deformable TDNN layers will not cause more latency than standard TDNNs. However, in our experiments, this approach brings an intolerable decline in recognition accuracy (Table 5). To address it, we apply the clip mechanism in both training and inference. After applying latency controlled training, deformable TDNNs are able to do streaming speech recognition without the degradation of recognition accuracy.

Table 5: *WER (%) results on WSJ when using different strategies for latency control in training and testing.*

Training	Testing	dev93	eval92
No constrains	No constrains	5.48	2.50
No constrains	latency controlled	7.70	4.38
latency controlled	latency controlled	5.54	2.60

7. Conclusion

This paper introduces deformable TDNNs for end-to-end speech recognition, featured by adaptability to input variations, superior results, and capability of streaming speech recognition. Moreover, deformable TDNNs can replace TDNNs easily in existing neural architectures and can be trained in an end-to-end manner without extra supervision. Future works include evaluating the effectiveness of deformable TDNNs on large-scale datasets and exploring the combination with other neural architectures (e.g., LSTMs, self-attention layers).

8. References

[1] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015, p. 3214–3218.

[2] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proc. INTERSPEECH*, 2018, pp. 3743–3747.

[3] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Proc. INTERSPEECH*, 2018, pp. 12–16.

[4] H. Zheng, K. An, and Z. Ou, "Efficient neural architecture search for end-to-end speech recognition via straight-through gradients," in *Proc. IEEE SLT*, 2021.

[5] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," in *Proc. ICASSP*, 2020, pp. 6084–6088.

[6] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for asr using self-attention network and chunk-hopping," in *Proc. ICASSP*, 2019, pp. 5656–5660.

[7] K. Han, R. Prieto, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," in *Proc. ASRU*, 2019, pp. 54–61.

[8] K. J. Han, J. Pan, V. K. N. Tadala, T. Ma, and D. Povey, "Multistream cnn for robust acoustic modeling," *ArXiv*, vol. abs/2005.10470, 2020.

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[10] P. Lei and S. Todorovic, "Temporal deformable residual networks for action segmentation in videos," in *Proc. CVPR*, 2018, pp. 6742–6751.

[11] K. Papadimitriou and G. Potamianos, "Multimodal Sign Language Recognition via Temporal Deformable Convolutional Sequence Learning," in *Proc. Interspeech*, 2020, pp. 2752–2756.

[12] K. An, H. Xiang, and Z. Ou, "CAT: A ctc-crf based asr toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," in *Proc. INTERSPEECH*, 2020, pp. 566–570.

[13] S. W. Shiliang Zhang, Hui Jiang and L. Dai, "Feedforward sequential memory neural networks without recurrent feedback," *arXiv preprint arXiv:1510.02693, 2015.*, 2015.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[17] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for asr," in *Proc. ICASSP*, 2018, pp. 5874–5878.

[18] S. Zhang, E. Loweimi, P. Bell, and S. Renals, "Windowed attention mechanisms for speech recognition," in *Proc. ICASSP 2019*, 2019, pp. 7100–7104.

[19] K. Audhkhasi, G. Saon, Z. Tüske, B. Kingsbury, and M. Picheny, "Forget a bit to learn better: Soft forgetting for CTC-based automatic speech recognition," 2019, pp. 2618–2622.

[20] K. J. Han, J. Huang, Y. Tang, X. He, and B. Zhou, "Multi-Stride Self-Attention for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2788–2792.

[21] Y. Zhang, H. Yu, and Z. Ma, "Speaker verification system based on deformable cnn and time-frequency attention," in *Proc. APSIPA ASC*, 2020, pp. 1689–1692.

[22] K. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, pp. 23–43, 1990.

[23] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," in *Proc. ICLR*, 2020.

[24] H. Xiang and Z. Ou, "CRF-based single-stage acoustic modeling with CTC topology," in *Proc. ICASSP*, 2019, pp. 5676–5680.

[25] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[26] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," in *Proc. ICASSP*, 2018, pp. 5929–5933.

[27] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "Flat-start single-stage discriminatively trained HMM-based models for ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 1949–1961, 2018.

[28] Y. Wang, T. Chen, H. Xu, S. Ding, H. Lv, Y. Shao, N. Peng, L. Xie, S. Watanabe, and S. Khudanpur, "Espresso: A fast end-to-end neural speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019.

[29] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[30] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.

[31] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. Cohen, H. Nguyen, and R. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," in *Proc. Interspeech*, 2019, pp. 71–75.

[32] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.