Interspeech 2022
18-22 September 2022, Incheon, Korea

Regularizing Transformer-based Acoustic Models by Penalizing Attention Weights for Robust Speech Recognition

Mun-Hak Lee[1], *Sang-Eon Lee*[1], *Ju-Seok Seong*[1]
Joon-Hyuk Chang[1*], *Haeyoung Kwon*[2], *Chanhee Park*[2]

[1]Department of Electronic Engineering Hanyang University
[2]Hyundai Motor Company, Seoul, Republic of Korea

lullaby0804@hanyang.ac.kr, leese94@hanyang.ac.kr, as2835510@hanyang.ac.kr,
jchang@hanyang.ac.kr, hykwon@hyundai.com, chanhee.park@hyundai.com

Abstract

The application of deep learning has significantly advanced the performance of automatic speech recognition (ASR) systems. Various components make up an ASR system, such as the acoustic model (AM), language model (LM), and lexicon. Generally, the AM has benefited the most from deep learning. Numerous types of neural network-based AMs have been studied, but the structure that has received the most attention in recent years is the Transformer [1]. In this study, we demonstrate that the Transformer model is more vulnerable to input sparsity compared to the convolutional neural network (CNN) and analyze the cause of performance degradation through structural characteristics of the Transformer. Moreover, we also propose a novel regularization method that makes the transformer model robust against input sparsity. The proposed sparsity regularization method directly regulates attention weights using silence label information in forced-alignment and has the advantage of not requiring additional module training and excessive computation. We tested the proposed method on five benchmarks and observed an average relative error rate reduction (RERR) of 4.7%.

Index Terms: Speech Recognition, HMM based hybrid ASR, Acoustic Model, Transformer, Sparse Feature

1. Introduction

The development of deep learning has contributed significantly to improving the performance of ASR systems. The AM is an integral component of an ASR system to which deep learning technology is applied and serves to classify phoneme information in a given speech signal. Over the past several decades, various neural network structures have been proposed and studied to enhance the performance of the AM. Recurrent neural network (RNN)-based models have a recursive structure and are known to be suitable for processing the time-series data. However, RNNs have several limitations, such as gradient vanishing and difficulty in parallelization. RNN model structures, such as long short-term memory [2, 3] and gated recurrent units [4, 5], have been proposed to alleviate these problems, but they still exhibit low efficiency. Another neural network structure most commonly used in modern ASR systems is CNN. The CNN has high data efficiency by using parameter sharing and a local receptive field, and many AMs with the highest recognition performance adopt the CNN structure. Some representative CNN-based AMs include a time-delay neural network [6], which is a modified version of the 1-D CNN, Conformer that mixes CNN and Transformer [7], multi-stream CNN, and ContextNet , which diversifies the receptive field of the CNN [8, 9].

*corresponding author

Table 1: *Complexity and maximum path length for various layer types. We denote the dimension of the input feature as D, and the CNN kernel size as $\boldsymbol{K}$.*

	Complexity per layer	Maximum path length
Self-attention	$\mathcal{O}(T^2 \cdot D)$	$\mathcal{O}(1)$
FNN	$\mathcal{O}(T^2 \cdot D^2)$	$\mathcal{O}(1)$
CNN	$\mathcal{O}(\boldsymbol{K} \cdot T \cdot D^2)$	$\mathcal{O}(\log_{\boldsymbol{K}} T)$
RNN	$\mathcal{O}(T \cdot D^2)$	$\mathcal{O}(T)$

In addition to RNNs and CNNs, self-attention is the most popular type of neural network in recent years. The Transformer, a representative structure to which self-attention is applied, is a deep learning module that stacks multi-head self-attention, a fully connected neural network (FNN), and a layer norm [10]. It has demonstrated remarkable performance in the fields of natural language processing and computer vision [11, 12]. In the field of ASR, [13] proposed a Transformer-based AM structure, and [14] exhibited the superiority by relative evaluation of the RNN and Transformer in an end-to-end ASR framework. As another example, a structure that combines the CNN and Transformer exhibits excellent recognition performance [7, 15].

The most representative characteristic of a self-attention network that differentiates it from other neural network structures is that it considers the global context without the gradient vanishing problem [1]. Table 1 lists the maximum path length [16] of various neural networks for processing time-series data of fixed length T. The maximum path length refers to the number of operations passed until the information of a specific time step affects another time step T away. A small maximum path length prevents gradient vanishing and thus allows better modeling of long-range dependencies. Self-attention has a short maximum path length, which overcomes the vanishing gradient problem, and has a higher computational efficiency than FNN in modeling T-length sequences (Table 1). Furthermore, although one FNN layer considers only the correlation between fixed-length inputs, self-attention enables the processing of variable-length inputs. These characteristics are the primary reasons why the Transformer has achieved remarkable success in numerous fields [16].

Although the Transformer has replaced the existing state-of-the-art records in many fields, it also has several weaknesses. For instance, problems with low data efficiency compared with

10.21437/Interspeech.2022-362

Table 2: *Correlation between silent interval length and WER (%). We show that the Transformer network is significantly affected by the ratio of the silent interval in the utterance compared to the 1-D CNN network. We evaluated the recognition performance using WSJ (test_eval92, test_eval92_5k, test_eval93, test_eval93_5k, train_short-500, test_short-500) and Car-env (Car-env_test) datasets.*

	CNN			Transformer		
	WSJ (sil:24.88%)	WSJ-trim (sil:8.50%)	RERR (%)	WSJ (sil:24.88%)	WSJ-trim (sil:8.50%)	RERR (%)
test_eval92	5.42	5.33	1.66(▲)	6.31	6.17	2.22(▲)
test_eval92_5k	1.57	1.60	-1.91(▼)	1.89	1.79	5.29(▲)
test_eval93	6.67	6.65	0.30(▲)	8.53	8.12	4.81(▲)
test_eval93_5k	2.83	2.84	-0.35(▼)	4.31	4.13	4.18(▲)
train_short-utt	6.54	4.88	25.38(▲)	5.22	3.43	34.29(▲)
test_short-utt	14.45	11.21	22.42(▲)	19.72	13.18	33.16(▲)
	Car-env (sil:61.54%)	Car-env-trim (sil:15.47%)	RERR (%)	Car-env (sil:61.54%)	Car-env-trim (sil:15.47%)	RERR (%)
Car-env-test	3.92	3.99	-1.79(▼)	5.12	4.91	4.10(▲)

CNN networks [12, 17, 18] and problems with the excessive computational complexity of T^2 when modeling long time-series data have been pointed out as weaknesses [19]. In this study, we intend to address the vulnerability of the Transformer network, which has not been observed in previous studies. The contributions of our study are summarized as follows:

1. We assumed a situation in which the speech signal in the input audio was sparsely distributed. In this situation, it is shown that the Transformer network suffers a fatal performance degradation compared to CNN
2. We present a regularization method to alleviate this sparsity problem. The proposed regulation method uses information about silent intervals in forced-alignment used for training the AM and regulates attention weights to train the Transformer network to distinguish silent intervals from speech intervals.

2. Sparse feature problem

In this study, we discuss the problem of sparse features, in which most values are located near zero in the feature space. These sparse features cause the following problems in machine learning tasks. 1. The time and space complexity increase. 2. Makes the model easily overfit to noise. 3. Slowing down model training. The most common solution to these problems is to increase the feature density using principal component analysis (PCA), or to make the model itself robust to sparse features [20]. In this section, we show that reducing the sparsity in speech features improves the performance of the Transformer-based AM and analyze the causes of the performance enhancement.

The target dataset to which we want to train the Transformer AM is a keyword speech dataset recorded in a car environment (Car-env). The driver transmits a short voice command while pressing the record button in the vehicle, and as this operation method is adopted, the speech dataset we collected includes a long waiting period in which vehicle noise is mixed before and after utterance. Therefore, the Car-env dataset has a characteristic in which utterances are distributed sparsely compared to the audio dataset recorded in the studio (the silent interval of the general benchmark is approximately 20%, whereas the Car-env dataset has silent intervals of approximately 60%).

In this study, we regard these silent intervals as redundancies that promote audio signal sparsity and analyze the correlation between speech data sparsity and recognition performance by adjusting the length of the silent intervals. Accordingly, we created datasets (WSJ-trim, Car-env-trim) in which the silent intervals at both ends of speech utterances of the Wall Street Journal (WSJ) and Car-env datasets were cut off. Subsequently, we compared the word error rate (WER) of the ASR system trained and evaluated on the original and truncated data, respectively. In this experiment, we use a 1-dimensional CNN model as a comparative model of the Transformer. The Transformer model has a global dependency, whereas the CNN considers only a limited level of local dependency according to the kernel size, number of layers, and dilation rate. We analyze the effects of the contrasting characteristics of the two models on the correlation between speech sparsity and recognition performance.

The experimental results are listed in Table 2. We evaluated the WER for six subsets in the WSJ. The four evaluation sets are provided by the WSJ, and train_short-utt and test_short-utt are composed of 500 selected each in the order of the shortest sentences among the WSJ trainset and testsets. The silent intervals of the original WSJ dataset occupy approximately 24.88% of the total audio length, whereas the truncated dataset has a silent interval ratio of 8.50%. For the Car-env dataset, the original file has a silent interval ratio of 61.54%, whereas the truncated dataset has a silent interval ratio of only 15.47%. As a result of the experiment, the CNN-based AM was almost unaffected by the silent intervals at both ends of the utterance, while the Transformer-based AM exhibited a consistent improvement in RERR by approximately 2$\sim$34% in a dataset with small sparsity. The experimental results reveal the vulnerability of the Transformer, which is significantly affected by the sparsity of the input features. For utterances composed of short sentences (train/test-short-utt), this performance difference was more pronounced. Accordingly, we interpret the performance difference to be caused by the structural characteristics (global dependency and short maximum path length) of the self-attention module. Based on this analysis, in the next section, we demonstrate that restricting the global dependency of the self-attention module based on the silence label information in the forced-alignment advances the generalization performance of the Transformer-based AM.

3. Proposed methods

Our ASR system follows the training method of the hybrid ASR system that classifies phoneme information for each frame using a deep neural network-hidden Markov model (DNN-HMM)

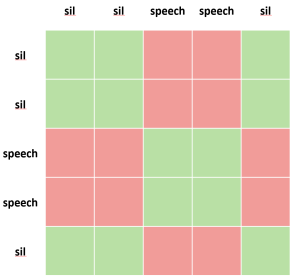

Figure 1: *We visualize the attention weights (A) of the Transformer network. The attention weights express the correlation between frames of the input sequence, and the proposed sparsity regularization method regulates the cross-attend area (red area) to have a low value.*

based AM [21]. In such a training method, phoneme labels for each frame of a speech signal (forced-alignment) are generated using a Gaussian mixture-hidden Markov model (GMM-HMM) based ASR model, and a DNN-based AM is trained in a supervised learning fashion using the generated forced-alignment. The proposed idea aims to directly regulate the attention weights (A) of the self-attention network using the silent area information in the forced-alignment so that the self-attention module operates more robustly against feature sparsity. The Transformer creates frame-wise features that consider global feature information through a scaled dot-product self-attention mechanism, as illustrated below:

$$Attention(Q, K, V) = softmax(\frac{QK^{\top}}{\sqrt{d}})V = AV, \quad (1)$$

where $A \in \mathbb{R}^{T\times T}$ denotes the attention weight, $Q \in \mathbb{R}^{T\times d}$ denotes the query, $K \in \mathbb{R}^{T\times d}$ denotes the key, $V \in \mathbb{R}^{T\times d}$ denotes the value, d denotes the hidden space dimension, and T denotes the length of the utterance. In this case, A plays a role in calculating the correlation between input frames, and the self-attention module adds up the information between frames with high correlation and transmits it to the next layer. We divide the attention weights into two areas using silence label information in the forced-alignment:

1. The case of inferring the silent area by attending the silent area and inferring the speech area by attending the speech area (green area in Figure 1)
2. The case of inferring the speech area by attending the silent area (or vice versa) (red area in Figure 1)

Among these two cases, we intend to impose a penalty on the cross-attend case (i.e., the red area in Figure 1) that attends the area with different information. We design two types of loss functions to meet this objective and modified the objective function by summing it with the general cross-entropy loss. The first method applies L1 regularization to the cross-attend area of the attention weights.

$$\mathcal{L}_{L1}(A, \hat{Y}) = \sum_{i=1}^{T}\sum_{j=1}^{T} \mathbb{I}(\hat{Y}_i, \hat{Y}_j)|a_{ij}|$$

$$\mathbb{I}(\hat{Y}_i, \hat{Y}_j) = \begin{cases} 1, & \text{if } \hat{Y}_i \neq \hat{Y}_j \\ 0, & \text{if } \hat{Y}_i = \hat{Y}_j \end{cases} \quad (2)$$

$$\hat{Y}_t = \begin{cases} 1, & \text{if } Y_t \text{ is speech} \\ 0, & \text{if } Y_t \text{ is silence} \end{cases}$$

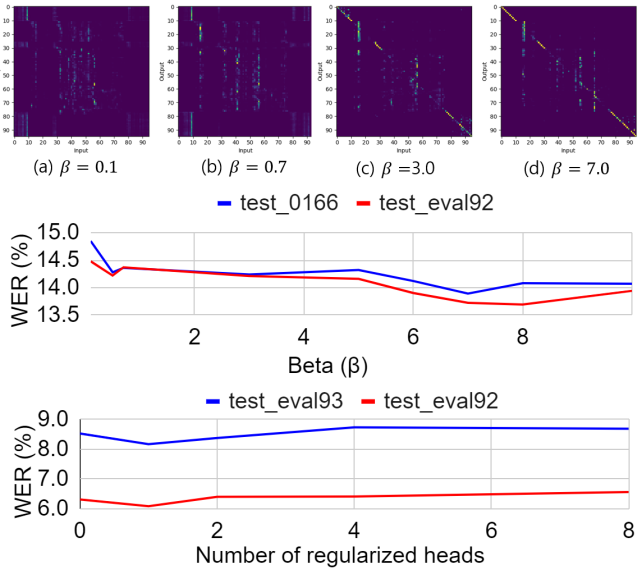

Figure 2: *We apply the sparsity regularization to the AURORA4 (top, middle) and WSJ (bottom). We visualized the attention weights of the last Transformer layer trained with different β (top). And we measured WER (%) by varying β (middle) and the number of regularized attention heads (bottom).*

where $Y \in \{1, ..., k\}^T$ denotes the target vector, a_{ij} denotes the i-th row and j-th column element of A, and k denotes the number of classes. The second method uses the modified version of the triplet ranking loss [22], which forces the cross-attend case to maintain a lower similarity value than the other case.

$$\mathcal{L}_{rank}(A, \dot{Y}) = \sum_{i=1}^{T}(1 - \dot{Y}_{ij}\dot{Y}_{ik})\max(m, \dot{Y}_{ij}a_{ij} + \dot{Y}_{ik}a_{ik}) \quad (3)$$

$$\dot{Y}_{ij} = \begin{cases} -1, & \text{if } \hat{Y}_i = \hat{Y}_j \\ +1, & \text{if } \hat{Y}_i \neq \hat{Y}_j \end{cases} \quad (4)$$

where the margin ($m \geq 0$) corresponds to a tuning factor and j, k are randomly sampled pairs in $\{1, \ldots, T\}$. Although the two different loss terms aforementioned play a similar roles, we observed that the performance of $\mathcal{L}_{rank}$ was consistently high in most experiments, and unless noted otherwise, we used $\mathcal{L}_{rank}$ in all experiments. Moreover, we updated the parameters of the Transformer using an objective function that combines the proposed regularization term and cross-entropy loss for supervised learning. Additionally, we use $\hat{\beta}$ to control the regularization strength, and total loss function ($\mathcal{L}_{total}$) is given as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \hat{\beta}\mathcal{L}_{rank} \quad (5)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss. We increase $\hat{\beta}$ linearly from 0 to β for every iteration, and $\beta > 0$ is a tuning factor. The proposed sparsity regularization method is not applied in the decoding process. Therefore, additional computation is not required in the decoding process, and there is no need for an external voice activity detection module or pre-generated forced-alignment.

Table 3: *Recognition performance of two types of AMs in 5 types of datasets.*

	Silence ratio (%)	Evaluation set	1D-CNN (WER, %)	Transformer (WER, %)	Transformer +sparsity-reg (WER, %)	RERR (%)
Car-env	60.63	test	3.92	5.12	4.61	9.96(▲)
		train	2.35	1.30	1.24	6.92(▲)
WSJ	24.88	test-eval92	5.42	6.31	6.08	3.65(▲)
		test-eval93	6.67	8.53	8.17	4.22(▲)
		train	8.39	6.34	6.29	0.79(▲)
CHIME3	18.85	dt05-real-noisy	13.04	11.32	11.21	0.97(▲)
		dt05-simu-noisy	11.50	11.03	10.69	3.08(▲)
AURORA4	18.30	test-0166	17.53	14.17	13.89	1.98(▲)
		test-eval92	16.85	14.47	13.72	5.18(▲)
Google-SC	9.31	test set	4.20	2.98	2.72	8.72(▲)

4. Experiments

We utilized four benchmarks (WSJ, CHIME3, Google Speech Command (Google-SC, [23]), and AURORA4) and one vehicle environment dataset (Car-env) for the experiments. Car-env is a 100-hour Korean dataset recorded in a vehicle consisting of short commands having an average of 1.6 words each. For Car-env and Google-SC, 10% of the data randomly sampled from the entire dataset was used as the test set (not used for training, speaker-independent). We reported the performance of other official benchmarks in a pre-divided test set. Moreover, we utilized an 83-dimensional f-bank feature extracted from a sound source with a 16k sampling rate as an input to the AM (with a 25ms window size and 10ms overlap).

We used Kaldi [24] for training the GMM-HMM AM and PyTorch [25, 26] for training the Transformer AM. We utilized a tri-gram-based LM in all the experiments. For the official benchmarks, we used the lexicon provided with the dataset, and when no lexicon was provided (Google-SC, Car-env), words were decomposed into characters and used as recognition units. In the case of the 1-D CNN, we used the same settings as in [6].

In the case of the Transformer AM, 12 Transformer layers with eight attention heads were stacked after two 2-D CNN subsampling layers (with a subsampling rate of three), and positional encoding was not applied. Accordingly, we used spec augmentation (only time masking and frequency masking) to train the Transformer network and did not apply any data augmentation methods to the 1-D CNN [13, 27]. In all the experiments, we used 0.0 as the margin (m) value and 8.0 for β.

5. Results

Regularization methods of machine learning models enhance the generalization performance at the expense of model expressiveness. Hence, it is necessary to adjust the trade-off between expressive power and generalization performance. Accordingly, we adjusted the trade-off by changing the number of attention heads to which the regulation was applied as well as the regularization strength (β).

First, we measured the change in recognition performance by changing the β value from 0.1 to 10 and visualized the attention weights of the last layer of the network trained using various β values, as shown in Figure 2. For most beta values, the sparsity regularization method exhibited significant performance improvement (Figure 2, middle). We visualized the attention weights that change according to the β value in Figure 2 (top). When the attention weights were more strongly regulated by assigning a high β value, we observed a progressive decrease in the value in the cross-attend region, such that the speech intervals (middle) and silent intervals (both ends) were more clearly distinguished.

Second, we conducted experiments by diversifying the number of regulated attention heads. The self-attention module used in the experiment has eight attention heads, and we demonstrated the recognition performance for each number of regulated heads in Figure 2 (bottom). The highest performance improvement was obtained when only one head was regulated out of the eight attention heads, and the recognition performance deteriorated when two or more heads were regulated. Based on these experimental results, the number of regulated heads was fixed to one and β to 7.5 in all experiments.

Finally, we applied the proposed sparsity regularization method to five datasets and obtained an average RERR of 4.7%. The experimental results are summarized in Table 3. The performance improvement was higher for datasets with longer silent intervals that were recorded in a noisy environment. Moreover, it is evident from Table 3 (Car-env and WSJ) that the Transformer network suffers from relatively severe overfitting compared to the 1D-CNN. This overfitting phenomenon was consistent with the previous research results in which the Transformer model exhibits lower generalization performance than the CNN when only a small dataset is provided [12, 17, 18], and it was shown that the proposed regularization method has a positive effect on alleviating this overfitting.

6. Conclusion

We experimentally demonstrated that the Transformer AM trained using speech data with long silent intervals undergoes extreme overfitting. To alleviate this overfitting problem, we proposed a new regulation method that directly regulates the attention weights of the self-attention module. We tested the proposed method on four public benchmarks and one real-world environment dataset and obtained an average RERR of 4.7%.

7. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

8. References

[1] 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*, vol. 30, 2017.

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

[3] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, 2014.

[4] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Improving speech recognition by revising gated recurrent units," in *Proc. INTERSPEECH 2017*, pp. 1308–1312.

[5] G. Cheng, D. Povey, L. Huang, J. Xu, S. Khudanpur, and Y. Yan, "Output-gate projected gated recurrent unit for speech recognition." in *Proc. INTERSPEECH 2018*, pp. 1793–1797.

[6] 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.

[7] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. INTERSPEECH 2020*, pp. 5036–5040.

[8] K. J. Han, J. Pan, V. K. N. Tadala, T. Ma, and D. Povey, "Multistream CNN for robust acoustic modeling," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6873–6877.

[9] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *arXiv preprint arXiv:2005.03191*, 2020.

[10] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[13] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6874–6878.

[14] S. Karita, N. Chen *et al.*, "A comparative study on transformer vs rnn in speech applications," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 449–456.

[15] K. J. Han, R. Prieto, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 54–61.

[16] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on visual transformer," *arXiv e-prints*, pp. arXiv–2012, 2020.

[17] J. Guo, K. Han, H. Wu, C. Xu, Y. Tang, C. Xu, and Y. Wang, "CMT: Convolutional neural networks meet vision transformers," *arXiv preprint arXiv:2107.06263*, 2021.

[18] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CVT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 22–31.

[19] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for asr," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5874–5878.

[20] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 8, pp. 1026–1041, 2007.

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 815–823.

[23] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. Neural Information Processing Systems (NIPS) 2017 Workshop on Autodiff*.

[26] 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*, 2017.

[27] 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*, pp. 2613–2617.