Interspeech 2022
18-22 September 2022, Incheon, Korea

NAS-SCAE: Searching Compact Attention-based Encoders For End-to-end Automatic Speech Recognition

Yukun Liu[1,2], *Ta Li*[1,2], *Pengyuan Zhang*[1,2], *Yonghong Yan*[1,2]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustic, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
{liuyukun, lita, zhangpengyuan, yanyonghong}@hccl.ioa.ac.cn

Abstract

Recently plenty of attention-based encoders have been proposed for end-to-end (E2E) automatic speech recognition (ASR). Despite the impressive performance, these encoders usually have a large model size and suffer from expensive memory and computation costs. To obtain more compact encoders for E2E ASR, we propose *searching compact attention-based encoders* using neural architecture search (NAS) in this paper, named NAS-SCAE. NAS-SCAE consists of one search space that contains a set of candidate encoders and one search algorithm responsible for searching the optimal encoder from the search space. On one hand, NAS-SCAE designs a topology-fused search space to integrate different architecture topologies of existing encoders (e.g. Transformer, Conformer) and explore more brand-new architectures. On the other hand, combined with the training pipeline of E2E ASR, NAS-SCAE develops a resource-aware differentiable search algorithm to search compact encoders efficiently and proposes an adjustable search scheme to alleviate the joint optimization problem of the differentiable search algorithm. On four Mandarin and English datasets, NAS-SCAE can effectively reduce the encoder resource consumption with negligible performance drop and achieve at least $2.13\times/2.09\times$ parameters/FLOPs reduction than the human-designed baselines.

Index Terms: speech recognition, end-to-end, neural architecture search

1. Introduction

In recent years, E2E ASR has achieved impressive performance and has been widely used in real-world applications for simply training and decoding pipelines [1–6]. In existing works, the mainstream online and offline E2E ASR frameworks tend to employ an attention-based encoder to extract contextualized representations from the input speeches, e.g. connectionist temporal classification (CTC) [7–9], listen attend and spell (LAS) [10–12] and RNN-Transducer (RNN-T) [13–15].

Conventional attention-based encoders usually are built by stacking identical encoder blocks, and the multi-headed self-attention (MHSA) based Transformer [16] is the most commonly-used block architecture in existing E2E ASR [11, 17, 18], which can capture global contextual information effectively with the MHSA mechanism. Recently convolutional neural network (CNN) also gains increasing attention in building encoders for its advantage in extracting local information, [19] combines the MHSA module and the CNN module in a cascade way, [20] puts the MHSA module parallel with the CNN module and then concatenates the outputs of them, [21] directly replaces the MHSA module with the CNN module on top of Transformer. As shown in Figure 1(a), the block architectures, i.e., the operation selections and architecture topologies of the blocks, are different among these various encoders.

Even though existing works in designing new encoders have improved the performance of E2E ASR effectively, most of them ignore the resource consumption in the encoder design, which leads to expensive memory and computation costs for real-world applications. There have existed several works that design compact encoders for Natural Language Processing (NLP), e.g. [20, 21] employ lightweight CNNs to build encoders, [22] proposes to reduce encoder parameters with NAS. Nevertheless, these works can hardly obtain satisfying performance when directly applied to E2E ASR due to the differences in background knowledge between NLP and ASR in practice. A deep-going exploration of the compact encoder design is necessary but scarce in existing E2E ASR works.

To obtain more compact encoders for E2E ASR, we propose NAS-SCAE to design attention-based encoders with the help of NAS. NAS aims at automating the design of neural network architecture and has been successfully applied in computer vision (CV) and NLP tasks [23–28]. Thus far, there have been several works that employ NAS to design attention-based encoders. [29] employs NAS to search encoders for E2E ASR, [22, 30] introduces NAS into the encoder design of NLP. These works have obtained certain improvements in specific tasks, however, there still exist two non-negligible problems to be addressed: **Inflexible Search Space.** The search spaces in existing works only provide various operation candidates for a fixed topology while paying trivial attention to the diversity of the topology itself, which limits the potential of the search space. **Expensive Search Algorithm.** Most of the existing works adopt the Reinforcement Learning (RL) or Evolutionary Algorithm (EA) based search algorithms, which need to evaluate thousands of architectures and are time-consuming in practical applications.

To address these challenges, NAS-SCAE designs a topology-fused search space and develops a differentiable search algorithm. By a topology-fusion trick, NAS-SCAE integrates different architecture topologies into the topology-fused search space. In this way, the search space can cover many existing human-designed attention-based encoders (e.g. Transformer [16], Conformer [19], LiteTransformer [20], LightConv [21]) and explore more brand-new architectures. Based on the search space, NAS-SCAE develops a resource-aware differentiable search algorithm on top of a popular one in the image classification tasks, i.e. DARTS [23], to search compact encoders for E2E ASR. Compared with the RL-based or EA-

This work is partially supported by the National Key Research and Development Program of China(No. 2020AAA0108002).

10.21437/Interspeech.2022-748

based algorithm, our differentiable search algorithm can provide a more efficient approach to search architectures. To alleviate the joint optimization problem of the differentiable algorithm, we further propose an adjustable search scheme to improve our search algorithm. Moreover, since NAS-SCAE only makes modifications to the architecture, NAS-SCAE is orthogonal to the general compression techniques (e.g. pruning and quantization) and can be easily combined with them.

2. Method

2.1. Differentiable Architecture Search

NAS-SCAE consists of one search space that contains a set of candidate encoders and one search algorithm responsible for searching the optimal encoder from the search space.

In the literature, the search space usually is represented as a directed acyclic graph (DAG) called super-network as shown in Figure 1(b), where each node $x^{(i)}$ is a latent representation and each directed edge (i, j) among node $x^{(i)}$ and node $x^{(j)}$ represents a set of candidate operations [27]. In this paper, we develop a differentiable search algorithm based on DARTS [23] to search encoders efficiently by relaxing the search space into continuous with softmax:

$$\bar{o}^{(i,j)}(x^{(i)}) = \sum_{k=1}^{|\mathcal{O}|} \frac{\exp(\boldsymbol{\alpha}_k^{(i,j)})}{\sum_{k'=1}^{|\mathcal{O}|} \exp(\boldsymbol{\alpha}_{k'}^{(i,j)})} o_k^{(i,j)}(x^{(i)}) \quad (1)$$

where $|\mathcal{O}|$ is the number of candidate operations, $\bar{o}^{(i,j)}$ represents the mixed operation of the directed edge (i, j), $o_k^{(i,j)}$ is the k-th candidate operation in the edge (i, j), and $\boldsymbol{\alpha}_k^{(i,j)}$ is the corresponding learnable architecture parameter. Then the node $x^{(j)}$ can be computed based on all of its predecessors: $x^{(j)} = \sum_{i<j} \bar{o}^{(i,j)}(x^{(i)})$. Different from RL-based or EA-based algorithms in previous works [22, 29, 30], the differentiable search algorithm will jointly optimize the architecture parameter $\boldsymbol{\alpha}$ and the super-network weights $\mathbf{W}$ with gradient descent based on the continuous relaxation, and this can improve search efficiency significantly.

Combined with the training pipeline of E2E ASR, NAS-SCAE will employ two stages to design encoders for E2E ASR: a search stage and a retraining stage. In the search stage, NAS-SCAE jointly optimizes $\boldsymbol{\alpha}$ and $\mathbf{W}$ and derives the searched encoder based on the optimized $\boldsymbol{\alpha}$. Then in the retraining stage, NAS-SCAE retrains the searched encoder in the same way as training the human-design encoders.

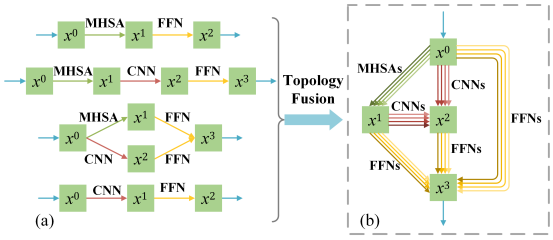


Figure 1: *(a) provides four human-designed attention-based encoder blocks. From top to bottom, they are Transformer, Conformer, LiteTransformer, LightConv, respectively. (b) is an overview of our topology-fused search space.*

2.2. The Topology-fused Search Space

Figure 1(a) illustrates four kinds of attention-based encoder blocks with different architectures in E2E ASR and NLP: Transformer, Conformer, LiteTransformer, LightConv, which consist of the MHSA, CNN and forward-feed network (FFN) modules. In practical applications, Transformer and Conformer can achieve better performance, while LiteTransformer and LightConv can obtain lower resource consumption. To integrate the performance and resource advantages of these different encoder blocks, we propose fusing the architecture topologies of these blocks into a search space, i.e., the topology-fused search space, as shown in Figure 1(b).

We introduce three kinds of edges into the search space, which corresponds to the MHSA, CNN and FFN modules respectively and then arrange these edges on top of the literature methods. Additionally, a `NONE` operation, i.e. $\texttt{NONE}(\mathbf{x}) = \mathbf{0}$, is provided for each edge to allow the abandonment of the current edge. In this way, the architecture topologies of many existing attention-based encoder blocks can be covered in our search space, e.g., Transformer $\{x^0 \to x^1 \to x^3\}$, Conformer $\{x^0 \to x^1 \to x^2 \to x^3\}$, LiteTransformer $\{x^0 \to x^1 \to x^3, x^0 \to x^2 \to x^3\}$, and LightConv $\{x^0 \to x^2 \to x^3\}$. Moreover, apart from these human-designed architectures, more brand-new architecture topologies (e.g. $\{x^0 \to x^1 \to x^2 \to x^3, x^0 \to x^2 \to x^3\}$) also can be obtained in the search space.

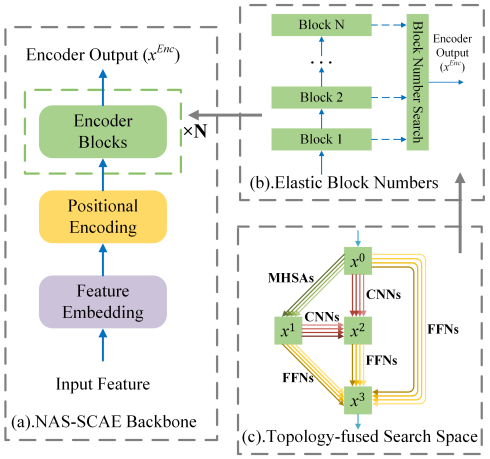


Figure 2: *(a) is the backbone of our NAS-SCAE encoders. (b) illustrates our elastic block number design. NAS-SCAE will determine the block number based on the search result. (c) is the topology-fused search space.*

Furthermore, we provide a set of candidate operations for each edge to allow the potential of different operation combinations. The MHSA operations own flexible head numbers, e.g. 4, 8, 16, denoted as `MHSA4`, `MHSA8`, and `MHSA16`. A set of CNN operations with diverse kernel sizes are provided, e.g. 7, 15, 31, denoted as `CNN7`, `CNN15`, `CNN31`. The FFN operations have various hidden dimensions, e.g. 512, 1024, 2048, denoted as `FFN512`, `FFN1024`, `FFN2048`. Especially, a `NONE` operation is provided for each edge to remove certain edges.

Figure 2(a) illustrates the backbone of our NAS-SCAE encoder. A feature embedding block and a positional encoding block are placed at the top of the encoder, and several NAS-SCAE blocks follow behind. Different from the fixed block number of human-designed encoders, NAS-SCAE makes the block number elastic and determines it based on a block number search module as shown in Figure 2(b), which is achieved on top of the continuous relaxation of block outputs:

$$x^{Enc} = \sum_{n=1}^{N} \frac{\exp(\boldsymbol{\alpha}_n)}{\sum_{n'=1}^{N} \exp(\boldsymbol{\alpha}_{n'})} x^{L_n} \quad (2)$$

where x^{Enc} is the final output of the encoder, N is the max-

Table 1: *Comparisons among the human-designed baselines and NAS-SCAE. We report CER/WER, the encoder parameter numbers and FLOPs, and FLOPs are the average numbers for processing one-second speech. The best results on each dataset are in boldface.*

	Aishell-1				HKUST			LibriSpeech100				Hub5' 00 (SWBD)			
	#Params (M)	FLOPs (M)	dev	test	#Params (M)	FLOPs (M)	test	#Params (M)	FLOPs (M)	test clean	test other	#Params (M)	FLOPs (M)	swbd	callhm
Transformer	19.3	978	4.9	5.4	19.3	978	21.9	19.3	978	9.2	22.1	19.3	978	8.1	16.3
Conformer	19.0	960	**4.8**	5.3	19.0	960	**20.8**	19.0	960	**8.2**	**20.3**	19.0	960	8.0	**16.0**
LiteTransformer	18.6	897	5.4	6.1	18.6	897	21.6	18.6	897	8.7	21.9	18.6	897	9.2	17.6
LightConv	18.7	942	5.6	6.4	18.7	942	23.2	18.7	942	9.5	25.0	18.7	942	9.6	18.5
Random Search	12.2	613	5.3	6.0	12.3	622	22.1	12.6	651	10.9	23.4	12.3	627	8.5	17.2
NAS-SCAE	**8.4**	**424**	**4.8**	**5.2**	**8.9**	**457**	21.0	**8.7**	**447**	8.3	20.5	**8.6**	**439**	**7.9**	16.1

imum number of the encoder block, x^{L_n} refers to the output of the n-th block. And then, for each block, the block architecture will be derived from the topology-fused search space in Figure 2(c) in a block-wise way.

2.3. Resource-aware Architecture Search

To take both the effectiveness and resource consumption into consideration and make our search algorithm resource-aware to search compact encoders for E2E ASR, we propose a resource regularization $\mathcal{L}_{\text{Resource}}$ in NAS-SCAE. Combined with the training pipeline of E2E ASR, the search objective function of NAS-SCAE can be described as:

$$\mathcal{L} = \mathcal{L}_{\text{Performance}} + \eta \mathcal{L}_{\text{Resource}} \tag{3}$$

where $\mathcal{L}_{\text{Performance}}$ refers to the performance-related object, e.g., CTC, CE, or RNN-T, $\mathcal{L}_{\text{Resource}}$ is the resource-related regularization, e.g., the parameter numbers and FLOPs in NAS-SCAE, and η is a scaling factor. We also relax $\mathcal{L}_{\text{Resource}}$ into continuous:

$$\mathcal{L}_{\text{Resource}} = \sum_{n=1}^{N} \sum_{0 \le i < j < |\mathcal{X}|} \sum_{k=1}^{|\mathcal{O}|} \frac{\exp(\boldsymbol{\alpha}_k^{(i,j)_n})}{\sum_{k'=1}^{|\mathcal{O}|} \exp(\boldsymbol{\alpha}_{k'}^{(i,j)_n})} C_k^{(i,j)} \tag{4}$$

where $\boldsymbol{\alpha}_k^{(i,j)_n}$ is the architecture parameter of the n-th block, $|\mathcal{X}|$ is the node number in the search space (for example, $|\mathcal{X}| = 4$ as shown in Figure 1(b)), $C_k^{(i,j)}$ denotes the consumption of the parameter numbers and FLOPs in the k-th candidate operation on edge (i, j), which can be obtained with a table-lookup method. This paper sets η as 0.1 by default.

2.4. Adjustable Search Scheme

In the differentiable search algorithm, the architecture parameter $\boldsymbol{\alpha}$ and the super-network weights $\mathbf{W}$ are jointly optimized by the gradient descent method. However, since the topology-fused search space of NAS-SCAE is more complex and harder to optimize than the CNN-based ones of DARTS in practice, the conventional joint optimization of DARTS can hardly learn $\mathbf{W}$ well and will introduce additional difficulty into our search process.

To alleviate this problem, we develop an adjustable search scheme to strengthen the training of the super-network weights $\mathbf{W}$. On one hand, we first pretrain the super-network with P additional epochs, i.e., only $\mathbf{W}$ will be updated in the first P epochs and $\boldsymbol{\alpha}$ will not be updated until the $P+1$ epoch. On the other hand, different from DARTS updates $\mathbf{W}$ and $\boldsymbol{\alpha}$ alternately in each iteration, we increase update iterations of $\mathbf{W}$ to I, i.e., $\mathbf{W}$ will be updated with I iterations when $\boldsymbol{\alpha}$ is updated with one iteration. This paper sets P as 10 and I as 4 by default.

3. Experiments

3.1. Dataset and Model Implementation

Dataset In this work, we evaluate NAS-SCAE on four Mandarin and English datasets: Aishell-1, HKUST, LibriSpeech100, SWBD, and the performance is evaluated by CER/WER on Mandarin/English datasets, respectively. We extract acoustic features using 80-dimensional Mel-filterbanks with a 25ms window and a 10ms shift. Additionally, we add 3-dimensional pitch features to the extracted features. SpecAugment [31] is used for data augment.

Baselines We employ the Transformer, Conformer, LiteTransformer and LightConv encoders as our human-designed baselines. Transformer, Conformer and LightConv follow the architecture configurations finetuned by the ESPnet developers [32]. As an architecture proposed for NLP, LiteTransformer follows the architecture setting in [20]. For consistency with the other three baselines, Conformer does not employ the Macron style FFN. Further, we employ a random search baseline to compare with NAS-SCAE by randomly sampling 5 candidate architectures based on the proposed search space.

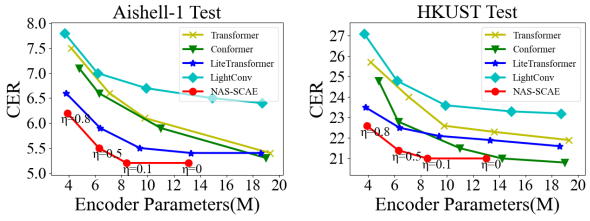
Implementation Details For NAS-SCAE, we set $N = 8$ in Figure 2(a) with the attention dimension as 256 for all MHSA operations. The CNN operations follow the same settings in [19]. The experiments are conducted based on the LAS ASR framework with the toolkit ESPnet [33] on a single Nvidia Titan RTX. A four layers Transformer decoder is adopted in all our models. We retrain the human-designed baselines and the searched architectures with the same hyper-parameters finetuned by the ESPnet developers.

3.2. Performance Comparisons

For a fair comparison, we adjust the block numbers of four human-designed baselines and make them have comparable performance with NAS-SCAE results. Table 1 shows the results of the human-designed baselines and NAS-SCAE[1]. Conformer can obtain the best performance among the human-designed baselines, while NAS-SCAE can obtain comparable results with fewer parameters and FLOPs. To be specific, NAS-SCAE can obtain $2.26\times$, $2.13\times$, $2.18\times$, $2.20\times$ parameters reduction and $2.26\times$, $2.09\times$, $2.14\times$, $2.18\times$ FLOPs reduction than Conformer with negligible performance drops on Aishell-1, HKUST, LibriSpeech100, SWBD, respectively. Compared with the random search that adopts the same search space as NAS-SCAE, the performance gain of NAS-SCAE owns to our differentiable search algorithm. These comparisons demonstrate the effectiveness of NAS-SCAE in designing compact encoders.

To further explore the effectiveness of NAS-SCAE under different model sizes, we conduct more experiments on Aishell-

[1]More detailed results are provided on `https://github.com/liuyukun98/NAS-SCAE/tree/main/Result_Details`



(a) Aishell-1 (b) HKUST

Figure 3: *The performance-parameter trade-off results among NAS-SCAE and four human-designed baselines on Aishell-1 and HKUST test datasets.*

1 and HKUST with $\eta \in \{0, 0.1, 0.5, 0.8\}$, respectively. Meanwhile, we obtain the performance of the human-designed baselines with different parameters by adjusting the encoder block numbers. Figure 3 shows the trade-off results between the performance and encoder parameters.

Among the human-designed baselines, LiteTransformer can obtain the best performance when the parameter number is small (less than 4M), and Conformer will surpass it as the parameter grows. With the topology-fused search space, NAS-SCAE can integrate the strengths of these architectures together. As shown in Figure 3, the NAS-SCAE curve locates at the left bottom, which means NAS-SCAE can consistently obtain smaller parameter numbers than the human-designed baselines with comparable performance. Furthermore, compared with the method that lacks the resource regularization ($\eta = 0$ in NAS-SCAE), a proper resource regularization ($\eta = 0.1$) can reduce parameters effectively without any performance drop, which demonstrates the necessity of the resource regularization.

3.3. Ablation Study

3.3.1. Search Space

To investigate the influence of our topology-fused search space, we conduct ablation studies on reduced search spaces by pruning certain edges or fixing the operation on certain edges. As shown in Table 2, we observe obvious performance drops on the pruned search space *'Only Cascade Topology'* and *'Only Parallel Topology'*. And the performance drop when fixing MHSA or CNN indicates the necessity of diverse candidate operations on these edges. Although there is no performance drop when fixing FFN, the resource consumption becomes larger than the complete search space. These results demonstrate the importance of the topology-fusion design and candidate operation diversity in our search space.

Table 2: *Ablation studies of the search space on Aishell-1. Based on the complete search space in Figure 1(b), 'Only Cascade Topology' refers to the reduced search space that only contains the edges* $\{x^0 \to x^1 \to x^2 \to x^3\}$*, 'Only Parallel Topology' refers to the reduced search space that only contains the edges* $\{x^0 \to x^1 \to x^3, x^0 \to x^2 \to x^3\}$.

Search Space	#Params(M)	FLOPs(M)	dev	test
Only Cascade Topology	8.2	416	5.0	5.4
Only Parallel Topology	8.2	418	5.4	5.9
Fix MHSA as `MHSA4`	8.8	454	5.1	5.6
Fix CNN as `CNN15`	8.9	461	5.2	5.6
Fix FFN as `FFN2048`	13.1	660	**4.8**	**5.2**
Complete Search Space	8.4	424	**4.8**	**5.2**

3.3.2. Adjustable Search Scheme

In Table 3, we conduct more experiments with different P and I on Aishell-1 to explore the influence of the adjustable search scheme, and it should be noted that the conventional search algorithm of DARTS corresponds to $P = 0$ and $I = 1$. With the adjustable search scheme, our improved search algorithm outperforms DARTS both in performance and resource consumption, and we can observe that $P = 10$ and $I = 4$ is the best choice for NAS-SCAE in this paper.

Table 3: *Comparisons among different P and I on Aishell-1. P=0, I=1 refers to the search algorithm of DARTS.*

P	I	#Params(M)	FLOPs(M)	dev	test
0	1	8.9	462	5.3	5.9
0	2	8.6	442	5.0	5.5
0	4	8.5	433	4.9	5.4
0	8	8.3	415	5.0	5.6
0	4	8.5	433	4.9	5.4
5	4	8.5	433	**4.8**	5.3
10	4	8.4	424	**4.8**	**5.2**
20	4	8.6	442	4.9	5.4

3.4. Effectiveness on RNN-T

To explore the effectiveness of NAS-SCAE on different E2E ASR frameworks, we further employ NAS-SCAE to search encoders for the RNN-T framework on Aishell-1, whose decoder configurations also follow the ESPnet recipes. Table 4 illustrates the experiment results on RNN-T frameworks, NAS-SCAE also obtains an effective resource consumption reduction with comparable performance against the human-designed baselines, which demonstrates the effectiveness of NAS-SCAE on RNN-T frameworks. Actually, since $\mathcal{L}_{\text{Performance}}$ in Eq. (3) can be adjusted among CTC, CE, RNN-T or some other objects used in E2E ASR according to the specific frameworks, NAS-SCAE can provide a framework-agnostic approach to design compact attention-based encoders for E2E ASR.

Table 4: *Experiment results of RNN-T on Aishell-1. All the human-designed baselines use identical encoders in the LAS framework, while NAS-SCAE searches the encoder individually on the RNN-T framework.*

Search Space	#Params (M)	FLOPs (M)	dev	test
Transformer	19.3	978	5.6	6.2
Conformer	19.0	960	**5.3**	**5.9**
LiteTransformer	18.6	897	5.4	6.1
LightConv	18.7	942	5.9	6.5
NAS-SCAE	**12.6**	**644**	5.4	6.0

4. Conclusion

In this paper, we propose NAS-SCAE to design more compact attention-based encoders for E2E ASR. NAS-SCAE designs a topology-fused search space to integrate different architecture topologies of existing attention-based encoders and develops a resource-aware differentiable search algorithm to search compact encoders from the search space efficiently. An adjustable search scheme is proposed to alleviate the joint optimization problem of the differentiable search algorithm. Extensive experiments on four Mandarin and English datasets demonstrate the effectiveness of NAS-SCAE in designing compact encoders. Moreover, we observe that NAS-SCAE can work under both the LAS and RNN-T framework, which indicates NAS-SCAE is framework-agnostic for E2E ASR. On top of the NAS-SCAE platform, we plan to explore more advanced search algorithms for future work.

5. References

[1] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[4] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[5] T. Hori, S. Watanabe, and J. R. Hershey, "Joint ctc/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 518–529.

[6] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[8] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7115–7119.

[9] T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," 2019.

[10] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[11] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6084–6088.

[12] J. Drexler and J. Glass, "Learning a subword inventory jointly with end-to-end automatic speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6439–6443.

[13] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[14] M. Jain, K. Schubert, J. Mahadeokar, C.-F. Yeh, K. Kalgaonkar, A. Sriram, C. Fuegen, and M. L. Seltzer, "Rnn-t for latency controlled asr with improved beam search," *arXiv preprint arXiv:1911.01629*, 2019.

[15] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[17] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 8–15.

[18] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[19] 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," *arXiv preprint arXiv:2005.08100*, 2020.

[20] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," *arXiv preprint arXiv:2004.11886*, 2020.

[21] F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," *arXiv preprint arXiv:1901.10430*, 2019.

[22] H. Wang, Z. Wu, Z. Liu, H. Cai, L. Zhu, C. Gan, and S. Han, "Hat: Hardware-aware transformers for efficient natural language processing," *arXiv preprint arXiv:2005.14187*, 2020.

[23] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[24] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4095–4104.

[25] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[26] D. So, Q. Le, and C. Liang, "The evolved transformer," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5877–5886.

[27] H. Zheng, K. An, and Z. Ou, "Efficient neural architecture search for end-to-end speech recognition via straight-through gradients," *arXiv preprint arXiv:2011.05649*, 2020.

[28] S. Hu, X. Xie, S. Liu, M. Cui, M. Geng, X. Liu, and H. Meng, "Neural architecture search for lf-mmi trained time delay neural networks," *arXiv preprint arXiv:2007.08818*, 2020.

[29] J. Kim, J. Wang, S. Kim, and Y. Lee, "Evolved speech-transformer: Applying neural architecture search to end-to-end automatic speech recognition," *Proc. Interspeech 2020*, pp. 1788–1792, 2020.

[30] J. Xu, X. Tan, R. Luo, K. Song, J. Li, T. Qin, and T.-Y. Liu, "Nas-bert: task-agnostic and adaptive-size bert compression with neural architecture search," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1933–1943.

[31] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[32] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5874–5878.

[33] S. Watanabe, T. Hori, S. Karita, T. Hayashi, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech 2018*, 2018.