LEARNED TRANSFERABLE ARCHITECTURES CAN SURPASS HAND-DESIGNED ARCHITECTURES FOR LARGE SCALE SPEECH RECOGNITION

Liqiang He¹, Dan Su¹, Dong Yu²

¹Tencent AI Lab, Shenzhen, China
²Tencent AI Lab, Bellevue WA, USA

ABSTRACT

In this paper, we explore the neural architecture search (NAS) for automatic speech recognition (ASR) systems. We conduct the architecture search on the small proxy dataset, and then evaluate the network, constructed from the searched architecture, on the large dataset. Specially, we propose a revised search space that theoretically facilitates the search algorithm to explore the architectures with low complexity. Extensive experiments show that: (i) the architecture learned in the revised search space can greatly reduce the computational overhead and GPU memory usage with mild performance degradation. (ii) the searched architecture can achieve more than 15% (average on the four test sets) relative improvements on the large dataset, compared with our best hand-designed DFSMN-SAN architecture. To the best of our knowledge, this is the first report of NAS results with a large scale dataset (up to 10K hours), indicating the promising application of NAS to industrial ASR systems.

Index Terms— neural architecture search, speech recognition, transferable architecture

1. INTRODUCTION

The performance of ASR systems has been largely boosted by deep learning [1]. The core part of deep learning is to design and optimize deep neural networks. Various types of neural network architectures have been employed in ASR systems, such as convolutional neural networks (CNNs) [2], long short-term memory (LSTM) [3], gated recurrent unit[4], time-delayed neural network [5], feedforward sequential memory networks (FSMN) [6], etc. Some combinations of different architectures are also proposed to take advantage of their complementary property, such as CLDNN [7]. Recently, transformer architecture which has achieved its success in the natural language process (NLP) tasks has also been widely used in ASR systems [8, 9], demonstrating its superior performance compared with the state-of-the-art models. Our previous work also proposed a variant of model architecture which combined DFSMN with self-attention networks (SAN), and further applied the memory augmenting method on the self-attention layer [10]. In summary, the performance improvement of ASR systems owes much to the dedicated hand-designed model architectures.

However, designing state-of-the-art neural network architectures requires a lot of expert knowledge and takes ample time. Therefore, there has been a growing interest in developing algorithmic solutions to discover powerful neural network architectures automatically. The network architectures automatically searched in [11, 12, 13] have achieved highly competitive performance in computer vision tasks, such as image classification and object detection. But, the heuristic search methods with evolution and reinforcement learning technique require massive computational overheads (3150 GPU days of evolution [13] and 2000 GPU days of reinforcement learning [12]). Several approaches focusing on the efficient architecture search have been proposed. Among them, DARTS [14] introduced a differentiable NAS framework to relax the discrete search space into a continuous one by weighting candidate operations with architectural parameters, which achieved comparable performance and remarkable efficiency improvement compared to previous approaches. As a progressive version of DARTS, P-DARTS [15] was proposed to bridge the depth gap between the network depth of architecture search and architecture evaluation.

Despite the rapid advance of NAS techniques in the computer vision communities, there has been very limited research on the application of NAS to ASR systems. Compared with computer vision tasks such as image classification, a major hindrance is that speech recognition is a more complex task in terms of the dimension of the input and output. What’s more, for the big data era of speech recognition, a typical amount of training data can be over 10K hours, which amounts to more than 10 million samples.

In this work, we explore the feasibility of NAS for speech recognition systems.
recognition tasks on the large dataset. Considering the search cost, we make the architecture search with multiple stages following P-DARTS on the small proxy dataset, and then the evaluation network, constructed from the searched architecture, is evaluated on the large dataset. We propose a revised search space for speech recognition tasks which theoretically facilitates the search algorithm to explore the architectures with low complexity, compared with the DARTS-based search space. Experimental results show that the architecture, discovered in the revised search space on the AISHELL-1 dataset, can achieve more than 20% and 15% (average on the four test sets) relative improvements respectively on the AISHELL-2 dataset and the large (10k hours) dataset, compared with our best hand-designed DFSMN-SAN architecture.

Our contributions can be summarized as follows: (i) We show that the architectures searched on the small proxy dataset has good transferability to the large (10k hours) dataset for the speech recognition tasks. (ii) We propose a revised search space, from which the searched architecture achieves a better balance between model complexity and recognition performance. (iii) We show that the searched architecture achieves significant performance improvements on the large dataset, compared with our best hand-designed model architecture.

2. NEURAL ARCHITECTURE SEARCH

2.1. DARTS [14]

Different from conventional methods applying evolution or reinforcement learning over a discrete search space, a differentiable network architecture search based on bilevel optimization is introduced in DARTS, which achieves remarkable efficiency improvement by several orders of magnitude. The categorical choice of one operation is relaxed to learning a set of continuous variables \( \alpha = \{\alpha^{(i,j)}\} \), normalized with the Softmax function.

\[
\hat{\sigma}^{(i,j)}(x) = \frac{\exp(\alpha^{(i,j)}_i) - o(x)}{\sum_{o' \in \mathcal{O}} \exp(\alpha^{(i,j)}_{o'})}
\]

(1)

A bilevel optimization is proposed which jointly optimizes the architecture \( \alpha \) as the upper-level variable and the network weights \( \omega \) as the lower-level variable:

\[
\min_{\alpha} \mathcal{L}_{\text{val}}(\omega^*(\alpha), \alpha) \quad \text{s.t.} \quad \omega^*(\alpha) = \arg\min_{\omega} \mathcal{L}_{\text{train}}(\omega, \alpha)
\]

(2)

(3)

where \( \mathcal{L}_{\text{train}} \) and \( \mathcal{L}_{\text{val}} \) denote the training and the validation loss, respectively. Both losses are determined not only by the architecture \( \alpha \), but also the network weights \( \omega \).

2.2. P-DARTS [15]

Although good transferability of the searched architecture has been observed in DARTS, more attention has been paid to the discrepancies between the super-network (the continuous architecture encoding) and the evaluation network constructed from the optimal sub-network (the derived discrete architecture) for the specific task. One of the discrepancies is the depth gap between the depth of the super-network and the evaluation network, which has been proven to cause performance deterioration. As a progressive version of DARTS, search space approximation is proposed in P-DARTS to alleviate the problem of the depth gap by dividing the search process into multiple stages. With each stage forward, the depth of the super-network becomes deeper, while at the same time the number of operations in the search space becomes smaller, which makes the search process with a deeper super-network possible with the limited computation and memory budget. Additionally, search space regularization is introduced to address the “over-fitting” problem brought by the skip-connect operation.

3. SEARCH SPACE REVISION

In this section, we summarize the characteristics and the improvements of the search algorithm when applying NAS to ASR systems. Based on the architecture in DARTS, there are two modifications specifically made for the speech recognition tasks. For large scale speech recognition, we propose a revised search space that theoretically facilitates the search algorithm to explore the architectures with low computational and memory overhead. Besides, the regularization method for the architecture search is discussed.

3.1. Architecture for speech recognition

Following DARTS, we search for the convolutional cells as the building blocks and then stack the learned cells to form the final network architecture. As shown in Figure 1 (left), each cell connects to the previous two cells or stem convolutions located at the beginning of the network. Cells located at the 1/3 and 2/3 of the network are reduction cells (totally two), which is different from the other normal cells. Two modifications for speech recognition are made as shown in the red dotted boxes of Figure 1 (right). In the modification at the beginning of the network, acoustic features and the first-order and the second-order derivatives are separately assigned to the independent channels. In the modification at the ending of the network, the average pooling operation is replaced by several fully connected layers, following with the softmax layer to compute the posteriors. Moreover, there are two reduction cells in the network, each of which reduces the resolution of the feature maps from the previous cells by half, and this architecture is also adopted for the speech recognition task, as the lower frame rate technique proposed by [16] has shown its benefit.

3.2. Search Space Revision and Regularization

The search space for the speech recognition task is represented in the form of the convolutional cells, and each cell is denoted as a directed acyclic graph consisting of \( N \) nodes and their corresponding edges. Each directed edge between two nodes is associated with the candidate operations. In DARTS, The candidate set in the convolutional cells includes the following operations: [zero, identity, 3x3 max pooling, 3x3 average pooling, 3x3 separable convolution, 5x5 separable convolution, 3x3 dilated separable convolution, 5x5 dilated separable convolution].

Considering the depth gap between the network depth applied in the search and the evaluation for the speech recognition tasks, the search process of our work adopts the search space approximation method proposed by P-DARTS. Based on the DARTS-based operation space, the preliminary architecture searches are carried out on the proxy dataset. One of the searched architecture is shown in Figure 2 (left).

We explore three issues related to the search space when applying P-DARTS for speech recognition tasks. First, the search process of the computer vision tasks tends to generate architectures with many skip-connect operations, especially on the small proxy dataset, and the derived architectures for evaluation often suffer from
the performance degradation. But, after searching on a small proxy
dataset, we find that the skip-connect operation rarely appears in
the final searched architecture for the speech recognition tasks. This
is arguably due to that speech recognition is a more complex task
in terms of both the dimension of the input and output, compared
with computer vision tasks such as image classification. Second,
the cell architectures, learned in the DARTS-based search space,
with better performance are prone to have many separable convolu-
tion operations, and this operation applies the module list twice, the
module list that consists of sequential modules with a RelU-Conv-
BN order. However, such learned architectures applied for large
scale speech recognition consumes too much computation resources
and GPU memory, which can be prohibitive due to the limitation of
GPU hardware. Last, to eliminate the influence of randomness, the
search process in DARTS and P-DARTS should be repeated several
times with different seeds for the final architecture with better per-
formance, and this method is still applicable for the search process
of the speech recognition. Notably, the search process of the speech
recognition tends to generate architectures with many average pool-
ing and dilated convolution operations, and obvious performance
fluctuations have been observed based on the evaluation networks
derived by stacking the learned normal cells for more times.

Concerning the first two issues, we revise the operation space
by replacing skip-connect operation with 1x7 then 7x1 convolution
[13]. The skip-connect operation has lower priority in the relaxed
search space for the absence in the final architectures most of the
time, so the stability of the search process can be improved by re-
moving this operation. The newly added convolution operation has
the following advantages. First, the convolution with the larger con-
volution kernel increases the receptive field to capture the latent rep-
resentation of acoustic features and meanwhile limits the number of
model parameters as smaller as possible. Second, the convolution
with fewer sequential modules improves computing efficiency and
memory overhead. With the revised operation space, the architecture
searches for the speech recognition tasks are carried out on the small
proxy dataset, and one of the searched architecture is shown in Fig-
ure 2 (right). The evaluation network, constructed from the searched
architecture, achieves a better trade-off between model complexity
and recognition performance.

Fig. 2: (a) and (c) are the cells (denoted as ASRNET-A) learned in the original search space. (b) and (d) are the cells (denoted as ASRNET-B)
learned in the revised search space proposed by our work.

separable convolution, 1x7 then 7x1 convolution].

As for the last issue mentioned above, we adopt the search space
regularization proposed by [15] to alleviate the problem of obvious
performance fluctuations caused by the randomness of the search
process. First, the operation-level dropout is inserted after each di-
lated separable convolution and average pooling operation to facil-
itate the algorithm to explore other operations. Second, the reg-
ularization rule of architecture refinement restricts the number of
preserved average pooling operations in the final architecture to be
a constant. average pooling operation has a significant impact on
recognition accuracy at the evaluation stage, especially on the proxy
dataset with complex scenarios.

4. EXPERIMENTS

4.1. Datasets

We use AISHELL-1 [17] as the small proxy dataset for the archi-
tecture search. The dataset contains 178 hours of Chinese Mandarin
speech from 400 speakers, and the 10 hours test set is used for the
architecture evaluation. Two bigger corpora are used to verify the
transferability of the searched architecture. First is the AISHELL-2
[18] dataset which contains 1000 hours of speech data from 1991
speakers. Second is a 10K hours multi-domain dataset [10]. We
also augment the AISHELL-1 and AISHELL-2 training data with
2-fold speed perturbation [19] in the experiments. To evaluate the
performance of the searched architecture, we report performance on
3 types of test sets which consist of hand-transcribed anonymized
utterances extracted from reading speech (1001 utterances), conver-
sation speech (1538 utterances), and spontaneous speech (2952 ut-
terances). We refer them as Read, Chat, and Spon respectively. Be-
sides, to provide a public benchmark, we also use the AISHELL-2
development set (2500 utterances, short for DEV) recorded by high
fidelity microphone as the test set.

4.2. Training setup

We use 40-dimensional log Mel-filterbank features with the first-order
and the second-order derivatives. Training utterances are filter-
ted by a maximum frame length of 1024, and the length of each
utterance is padded to be 4-frames-aligned to be fit for the two re-
duction layers. All the experiments are based on the CTC learning
Table 1: Token accuracies (Acc) and evaluation costs (Cost) of the small evaluation networks on the AISHELL-1 dataset. Abbreviations: L is the number of cells, C is the initial number of channels.

| Small | Params (M) | Acc (%) | Cost (hours) |
|-------|------------|---------|--------------|
| ASRNET-A (L=17,C=32) | 6.6 | 93.00 | 35.2 |
| ASRNET-B (L=17,C=24) | 6.8 | 92.24 | 17.8 |

Table 2: Comparison with DFSMN-SAN architecture on the AISHELL-2 dataset.

| Medium | Params (M) | CER (%) | Rel Imp (%) |
|--------|------------|---------|-------------|
| DFSMN-SAN | 14.4 | 7.36 | - |
| ASRNET-A (L=32,C=38) | 12.1 | 3.65 | 23.2 |
| ASRNET-B (L=32,C=30) | 14.7 | 5.79 | 21.3 |

framework and trained with multiple GPUs using BMUF [20] optimization. We use CI-syllable-based acoustic modeling units which include 1394 Mandarin syllables, 39 English phones, and a blank. First-pass decoding with a pruned 5-gram language model is performed with a beam search algorithm by using the weighted finite-state transducers (WFSTs). Character error rate (CER) results are measured on the test sets. Rel Imp refers to Relative Improvement in Table 2 and Table 3.

4.3. Architecture Search

The training set of the AISHELL-1 dataset is randomly split into two equal subsets, one for learning network parameters and the other for tuning the architectural parameters. The search process following [15] is divided into three stages. For each stage, the super-network is trained for 15 epochs, with batch size 4 (for both the training and validation sets) and the initial number of channels 16. Only network parameters are trained in the first 6 epochs, and both network and architecture parameters are alternately optimized in the rest 9 epochs. The momentum SGD optimizer with initial learning rate 0.01 (annealed down to zero following a cosine schedule without restart), momentum 0.9, weight decay 0.0003, is adopted to optimize the network parameters. The dropout probability on dilated separable convolution and average pooling is decayed exponentially and the initial values are set to be 0.05, 0.05, 0.05 for stage 1, 2 and 3, respectively. The final discovered normal cells are restricted to keep at most 2 average pooling. We run the search processes separately in the original (DARTS-based) search space and the revised search space proposed by our work. Concerning the influence of randomness, the search process is repeated 3 times with different seeds, respectively for both the DARTS-based search space and the revised search space. The search process takes around 89 hours on 8 Tesla P40 GPUs.

Table 3: Comparison with DFSMN-SAN architecture on the 10k hours dataset. CERS are measured on the four test sets.

| Large | Params (M) | Read (%) | Chat (%) | Spon (%) | DEV (%) |
|-------|------------|----------|----------|---------|--------|
| DFSMN-SAN | 36.1 | 1.95 | 22.92 | 25.41 | 4.42 |
| ASRNET-B (L=32,C=50) | 36.7 | 1.61 | 19.99 | 20.83 | 3.86 |

Rel Imp - 17.38 12.79 18.05 12.58

numbers of channels are 32, 24 for ASRNET-A and ASRNET-B respectively. The token accuracies are computed on the test set of the AISHELL-1 dataset. As seen in Table 1, the performance of the evaluation network constructed from ASRNET-A is slightly better than the one constructed from ASRNET-B, but the training time of the former almost takes almost twice as long as the latter. The training tasks are performed on 8 Tesla P40 GPUs.

To test the transferability of the searched architectures, the medium-sized evaluation networks stacked with 32 cells are trained from scratch for 15 epochs with batch size 4, on the AISHELL-2 dataset. The initial numbers of channels are 38, 30 for ASRNET-A and ASRNET-B respectively. Other training configurations are the same as the ones used for the small evaluation network. Character error rate results are computed on the test set of the AISHELL-2 dataset. As shown in Table 2, the medium-sized networks have achieved more than 20% relative improvements, compared with DFSMN-SAN [10] consisting of 10 DFSMN components and 2 multi-head self-attention sub-layers. Concerning computational overhead and GPU memory usage, the evaluation network constructed from ASRNET-A is almost twice as much as the one constructed from ASRNET-B, so the latter achieves a better trade-off between model complexity and recognition performance. The training processes are accelerated by applying 24 Tesla P40 GPUs.

To further validate the transferability of the searched architecture ASRNET-B, the large evaluation network stacked with 32 cells is trained from scratch for 6 epochs with batch size 4 and the initial number of channels 50, on the 10K hours large dataset. The momentum SGD optimizer with initial learning rate 0.0002, momentum 0.9, weight decay 0.0003, is adopted to optimize the network parameters. As shown in Table 3, compared with DFSMN-SAN consisting of 30 DFSMN components and 3 multi-head self-attention sub-layers, the large network has achieved more than 15% (average on the four test sets) relative improvements. The significant performance improvements imply that the searched architecture has good transferability. The training process takes around 7 days on 24 Tesla V100 GPUs.

4.4. Architecture Evaluation

On the AISHELL-1 dataset, the small evaluation networks stacked [12] with 17 cells are trained from scratch for 20 epochs with batch size 4. Other hyper-parameters remain the same as the ones used for the architecture search. Based on the recognition performance on the test set of AISHELL-1 dataset, the final architecture (denoted as ASRNET-A) discovered in the original search space is shown in Figure 2 (left) and the final one (denoted as ASRNET-B) discovered in the revised search space is shown in Figure 2 (right). The initial

5. CONCLUSIONS

In this paper, we empirically show that not only is the application of NAS for large scale acoustic modeling in speech recognition possible, but it also allows for very strong performance. Specifically, we perform the architecture search on 150 hours small dataset and then transfer the searched architecture to a large dataset for evaluation. On the 1000 hours AIShell-2 and 10K hours multi-domain datasets, the searched architecture achieves more than 20% and 15% (average on the four test sets) relative improvements respectively compared with our best hand-designed model architecture. The study of this work may unleash the potentials of NAS application for ASR systems. Future work includes adding latency control constraints into NAS to perform the architecture search for streaming ASR scenarios.
6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara 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.

[2] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, “Deep convolutional neural networks for lvcsr,” in 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013, pp. 8614–8618.

[3] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, “Hybrid speech recognition with deep bidirectional lstm,” in 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013, pp. 273–278.

[4] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio, “Light gated recurrent units for speech recognition,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 2, pp. 92–102, 2018.

[5] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[6] Shiliang Zhang, Ming Lei, Zhijie Yan, and Lirong Dai, “Deep-fsmn for large vocabulary continuous speech recognition,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5869–5873.

[7] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Hasım Säk, “Convolutional, long short-term memory, fully connected deep neural networks,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015, pp. 4580–4584.

[8] Linhao Dong, Shuang Xu, and Bo 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.

[9] Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Müller, Sebastian Stüker, and Alexander Waibel, “Very deep self-attention networks for end-to-end speech recognition,” in Proc. Interspeech 2019. IEEE, 2019, pp. 66–70.

[10] Zhao You, Dan Su, Jie Chen, Chao Weng, and Dong Yu, “Dfsmn-san with persistent memory model for automatic speech recognition,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7704–7708.

[11] Barret Zoph and Quoc V Le, “Neural architecture search with reinforcement learning,” arXiv preprint arXiv:1611.01578, 2016.

[12] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le, “Learning transferable architectures for scalable image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[13] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le, “Regularized evolution for image classifier architecture search,” in Proceedings of the aaai conference on artificial intelligence, 2019, vol. 33, pp. 4780–4789.

[14] Hanxiao Liu, Karen Simonyan, and Yiming Yang, “Darts: Differentiable architecture search,” arXiv preprint arXiv:1806.09055, 2018.

[15] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. “Progressive differentiable architecture search: Bridging the depth gap between search and evaluation,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1294–1303.

[16] G. Pundak and T. Sainath, “Lower frame rate neural network acoustic models,” in INTERSPEECH. IEEE, 2016.

[17] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA). IEEE, 2017, pp. 1–5.

[18] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu, “Aishell-2: transforming mandarin asr research into industrial scale,” arXiv preprint arXiv:1808.10583, 2018.

[19] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[20] Kai Chen and Qiang Huo, “Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering,” in ICASSP-2016, March 2016.