Self-Adaptive Scaling Approach for Learnable Residual Structure

Fenglin Liu[1], Meng Gao[3], Yuanxin Liu[4,5] and Kai Lei[2]*
[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]ICNLAB, School of ECE, Peking University, Shenzhen, China
[3]School of ICE, Beijing University of Posts and Telecommunications, Beijing, China
[4]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[5]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
fenglinliu98@pku.edu.cn, gaomeng@bupt.edu.cn
liuyuanxin@iie.ac.cn, leik@pkusz.edu.cn

Abstract

Residual has been widely applied to build deep neural networks with enhanced feature propagation and improved accuracy. In the literature, multiple variants of residual structure are proposed. However, most of them are manually designed for particular tasks and datasets and the combination of existing residual structures has not been well studied. In this work, we propose the Self-Adaptive Scaling (*SAS*) approach that automatically learns the design of residual structure from data. The proposed approach makes the best of various residual structures, resulting in a general architecture covering several existing ones. In this manner, we construct a learnable residual structure which can be easily integrated into a wide range of residual-based models. We evaluate our approach on various tasks concerning different modalities, including machine translation (IWSLT-2015 EN-VI and WMT-2014 EN-DE, EN-FR), image classification (CIFAR-10 and CIFAR-100), and image captioning (MSCOCO). Empirical results show that the proposed approach consistently improves the residual-based models and exhibits desirable generalization ability. In particular, by incorporating the proposed approach to the Transformer model, we establish new state-of-the-arts on the IWSLT-2015 EN-VI low-resource machine translation dataset.

1 Introduction

Recently, residual learning attracts considerable attention in training deep neural networks, and many efforts have been devoted to study the utilization of residual structure in tasks across a broad span of fields, including but not limited to computer vision (He et al., 2016a; Huang et al., 2017; He et al., 2016b; Szegedy et al., 2017) and natural language processing (Vaswani et al., 2017; Devlin et al., 2019). Residual structure, which alleviates the so-called gradient exploding or vanishing problem in optimization (He et al., 2016a), enables the training of neural networks with great depth by building skip connections between layers.

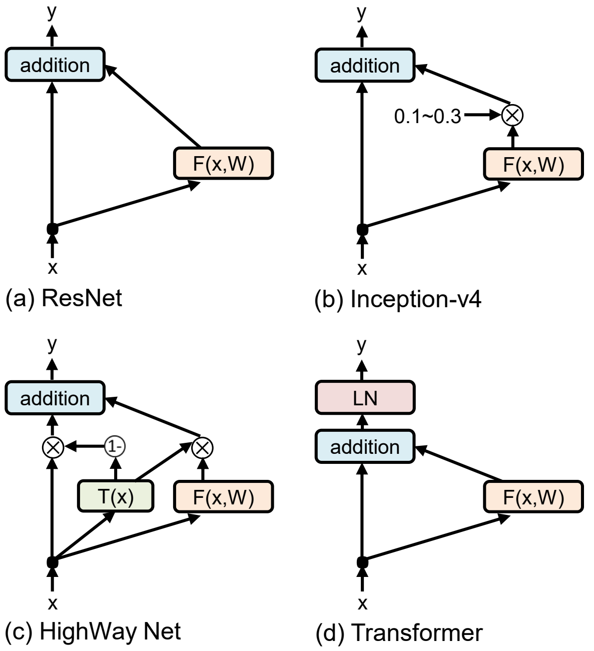


Figure 1: Various types of residual structures: (a) ResNet (He et al., 2016a); (b) Inception Net (Szegedy et al., 2017); (c) Highway Net (Srivastava et al., 2015); (d) Transformer (Vaswani et al., 2017), where LN represents layer normalization (Ba et al., 2016).

Generally, the residual structures (as illustrated in Figure 1) can be formulated as:

$$\boldsymbol{y} = \mathcal{G}(\alpha \cdot \boldsymbol{x} + \beta \cdot \mathcal{F}(\boldsymbol{x}, \mathcal{W})) \tag{1}$$

where $\boldsymbol{x}$ denotes the input (i.e., the skip connection), $\mathcal{F}$ denotes the residual function (i.e., residual branch) parameterized by W, and $\boldsymbol{y}$ is the output of the residual block. The balance between $\boldsymbol{x}$ and $\mathcal{F}$ is governed by the weights α and β, followed by $\mathcal{G}$, which could be either identity mapping or

*Corresponding Author

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 862–870
Hong Kong, China, November 3-4, 2019. ©2019 Association for Computational Linguistics

normalization.

Previous works on residual structure designing, which differ in the way that the information flows are regulated, mainly concern two elements, namely the mapping formulation (weight assignment) and the normalization mechanism. As shown in Figure 1, ResNet (He et al., 2016a), Inception-v4 (Szegedy et al., 2017) and Highway Net (Srivastava et al., 2015) explored the question on how should the residual connection be incorporated into the existing neural network structures so that the best improvements can be achieved. Recently, Transformer (Vaswani et al., 2017) applied the layer normalization (Ba et al., 2016) to help the optimization of the non-linear transformation (i.e., the $\mathcal{F}$) to some extent. At the same time, He et al. (2016b) observed considerably worse results when they utilized batch normalization (Ioffe and Szegedy, 2015) after the residual connection, the reason for which batch normalization is less employed in the residual structure.

Despite their respective advantages and success in certain fields, we argue that the structures are only particular cases of a more general one, which necessitates further insights into possible combinations. However, the determination of an effective combination may require prior knowledge of the data distribution, which is not always available, or extensive hyper-parameter exploration, which is inefficient.

In this paper, we aim at constructing a comprehensive and flexible residual structure. To this end, we propose the Self-Adaptive Scaling approach. In the residual structure, the proposed approach automatically computes scaling factors to adjust the mapping formulation and the normalization mechanism, respectively. By assigning different importance to the skip connection, the residual branch and a normalized result, the scaling factors adaptively controls the topology of the residual building blocks.

As a result, the structure learned by our proposed approach can be easily generalized to various kinds of tasks and data, dispensing with the time-consuming architecture search, to some extent. The proposed learnable residual structure can be easily integrated into existing residual-based models. We evaluate the proposed approach on representative residual models for various tasks. The experiment results and analyses attest to our argument and the effectiveness of the proposal.

Overall, the contributions are summarized as followed:

- We proposed a novel self-adaptive scaling (*SAS*) approach to acquire a learnable residual structure, which allows deep neural models to automatically learn the residual structure and can cover different types of existing ones.
- The proposed approach is simple and can be easily applied to a wide range of existing residual-based models. According to our empirical studies, the *SAS* can enable existing models to achieve consistent performance gains, demonstrating its generalization ability to a wide range of existing systems.
- The experimental results on the IWSLT-2015 EN-VI show that *SAS* helps the Transformer-Base model to perform even better than the Transformer-Big model and, encouragingly, we establish a new state-of-the-art on this low-resource machine translation dataset.

2 Related Work

In recent years, the application of residual structure to deep neural networks has become an active research topic (He et al., 2016a; Srivastava et al., 2015; He et al., 2016b; Vaswani et al., 2017; Szegedy et al., 2017). In the studies on residual architectures, there are two problems of interest. The first is how should the information from the skip connection and the residual branch be well balanced so that the best improvements can be achieved. The second is how should the neural network with residual connections be optimized so that its representation capability could be fully mined. These two types of problems are mainly addressed by designing appropriate mapping formulation and normalization mechanism, respectively, and we refer to them as *On the Connection Problem* and *On the Optimization Problem*.

On the connection problem. There are roughly three lines of methods to control the balance in residual connections: identity mapping, constant scaling ratio and adjusted scaling ratio. He et al. (2016b) designed five types of shortcut connections and discussed the possible residual connections in detail. Based on their theory and experiments, they argued that “keeping a ‘clean’ information path is helpful for easing optimization”. The reason is that with scaling, the gradient of the residual suffers from the gradient exploding or vanishing problem, which hinders the deep neural network from efficient optimization. Szegedy et al. (2017) adopted constant scaling to govern the residual balance in deep inception networks, which, despite its decent performance, is relatively inflexible. Highway network (Srivastava et al., 2015) is among the very first endeavors to implant residual structures into deep neural networks. It built a highway connection from the input to the output, where a transform gate was proposed to control the balance of the skip connection $\boldsymbol{x}$ and the residual branch $\mathcal{F}(\boldsymbol{x}, \mathcal{W})$, as opposed to the identity mapping.

On the optimization problem. In the realm of computer vision, PreAct-ResNet (He et al., 2016b) demonstrated that it is helpful to apply batch normalization to $\boldsymbol{x}$, instead of $\mathcal{F}(\boldsymbol{x}, \mathcal{W})$. In other words, the batch normalization acts on the output of the previous block. For natural language processing, the popular Transformer (Vaswani et al., 2017) makes use of residual connection in conjunction with layer normalization to build the model architecture and achieves record-setting performance. Layer normalization is widely believed to be helpful for stabilizing training and facilitating convergence. According to our experiments and analyses, the layer normalization can indeed facilitate optimization and therefore improve the overall performance of the model.

Different from existing work, we summarize the combination of normalization and residual connection in existing works with a general form $\boldsymbol{y} = \alpha * \boldsymbol{x} + \beta * \mathcal{F} + \gamma * \text{LN}(\boldsymbol{x} + \mathcal{F})$, where the mapping formulation and the normalization mechanism are both taken into account. By changing the scaling factors α, β and γ, the topology of the residual block can be adaptively adjusted, resulting in a learnable residual structure. The learned architecture distinguishes itself from the previous ones with generality and flexibility.

Our work is also related to the line of research on neural architecture search (Zoph and Le, 2017), where the network structure is also automatically learned by algorithm. However, neural architecture search requires sampling architecture descriptions based on predicted probability from a controller network that is optimized via reinforcement learning, which is time-consuming. But our proposed approach can be trained directly with the loss functions of different tasks.

No.	α	β	γ	Architecture
1	0	0	1	$\text{LN}(\boldsymbol{x} + \mathcal{F})$ (Vaswani et al., 2017)
2	1	1	0	$\boldsymbol{x} + \mathcal{F}$ (He et al., 2016a)
3	1	β	0	$\boldsymbol{x} + \beta * \mathcal{F}$ (Szegedy et al., 2017)
4	α	1	0	$\alpha * \boldsymbol{x} + \mathcal{F}$ (He et al., 2016b)

Table 1: Four particular cases in formula (3), which cover four representative residual structures, i.e., Transformer (Vaswani et al., 2017), ResNet (He et al., 2016a), Inception-v4 (Szegedy et al., 2017) and shortcut-only gating proposed in He et al. (2016b).

3 Architecture

In this section, we first briefly introduce the Scaling Gate in Section 3.1, which is used to predict the scaling factors for the mapping formulation and the normalization mechanism. Then, based on the scaling factors, in Section 3.2, we describe how to adaptively make the best of different types of residual structure to build a learnable residual structure.

3.1 Scaling Gate

The Scaling Gate should be able to predict reasonable scaling factors. Our motivation stems from the superior performance of Feed-Forward Network used in Vaswani et al. (2017). When selecting the activation function, since we expect the final predicted value of the scaling factors to cover the range of 0$\sim$1, we applied the Sigmoid activation function to the original outputs of the scaling gate. As a result, the scaling gate takes the $\boldsymbol{x} \in \mathbb{R}^h$ and the $\mathcal{F}(\boldsymbol{x}, \mathcal{W}) \in \mathbb{R}^h$ as input and computes the output through two linear transformations with a Tanh activation in between:

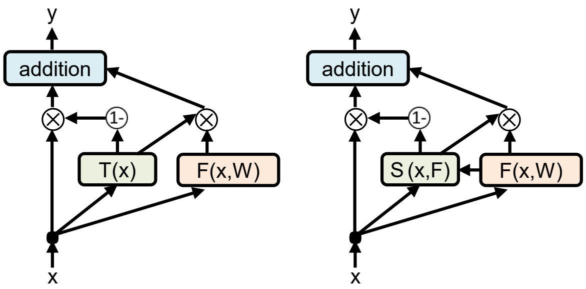
$$\mathcal{S}(\boldsymbol{x}, \mathcal{F}) = \text{Tanh}([\boldsymbol{x}; \mathcal{F}]W_f + b_f)W_{ff} + b_{ff} \qquad (2)$$

where [;] denotes concatenation operation, $W_f \in \mathbb{R}^{2h \times h}$ and $W_{ff} \in \mathbb{R}^{h \times 1}$ are the parameters to be learned, and $\mathcal{S}(\boldsymbol{x}, \mathcal{F})$ is followed by a Sigmoid activation function.

In HighWay Net (Srivastava et al., 2015), the inputs of the Transform Gate only involve the information of $\boldsymbol{x}$ and the structure only contains one layer of linear transformation, i.e., $\boldsymbol{x}W + b$. In contrary, we integrate the information from $\boldsymbol{x}$ and $\mathcal{F}$ and build the structure with two layers of linear transformation, which strengthens the scaling gate's expressive power. The difference is illustrated in Figure 2, and as illustrated in Table 6, our scaling gate is experimentally found to perform better.

No.	α	β	Architecture	Remarks
1	0	0	$\text{LN}(\boldsymbol{x}+\mathcal{F})$	The residual structure of Transformer (Vaswani et al., 2017).
2	0	1	$\mathcal{F}$	The well-trained $\mathcal{F}$ is sufficient in representation ability.
3	1	0	$\boldsymbol{x}$	$\mathcal{F}$ is poor-trained or $\boldsymbol{x}$ is sufficient in representation ability.
4	1	1	$\boldsymbol{x}+\mathcal{F}$	The residual structure of ResNet (He et al., 2016a).
5	1	β	$\boldsymbol{x}+\beta*\mathcal{F}$	The residual structure of Inception-v4 (Szegedy et al., 2017).
6	α	1	$\alpha*\boldsymbol{x}+\mathcal{F}$	The residual structure of shortcut-only gating (He et al., 2016b).
7	1-β	β	$(1-\beta)*\boldsymbol{x}+\beta*\mathcal{F}+$ $\beta(1-\beta)*\text{LN}(\boldsymbol{x}+\mathcal{F})$	The combination of Highway Net (He et al., 2016b) and Transformer.

Table 2: Seven special cases in formula (4), which include four representative structures, i.e., Transformer (Vaswani et al., 2017), ResNet (He et al., 2016a), Inception-v4 (Szegedy et al., 2017) and Highway Net (Srivastava et al., 2015).



(a) Transform Gate (b) Scaling Gate

Figure 2: The difference between the Transform Gate in Highway Net (Srivastava et al., 2015) and the proposed Scaling Gate.

3.2 Self-Adaptive Scaling Approach

It is intuitive to combine different types of residual structure through adjustable scaling factors. Therefore, we reformulate the residual block as follows:

$$\boldsymbol{y}=\alpha*\boldsymbol{x}+\beta*\mathcal{F}+\gamma*\text{LN}(\boldsymbol{x}+\mathcal{F}) \quad (3)$$

where α, β and γ can be predicted by scaling gates with different parameters, and LN stands for layer normalization (Ba et al., 2016).

By choosing certain values for α, β and γ, we can get several special cases, as shown in Table 1. However, from the summarized cases, we can easily find that if we set $\gamma=(1-\alpha)(1-\beta)$, the approach can also cover these four baselines. Especially, it is essential to decrease the parameters to achieve the same purpose[1]. Thus, the final self-adaptive scaling approach can be defined by the following formula:

$$\boldsymbol{y}=\alpha*\boldsymbol{x}+\beta*\mathcal{F}+(1-\alpha)(1-\beta)*\text{LN}(\boldsymbol{x}+\mathcal{F}) \quad (4)$$

where α and β act as the scaling factors predicted as aforementioned. From the above formula, we can obtain seven special cases in Table 2. In all, our proposed self-adaptive scaling approach is not only the general form of several existing structures, which is able to encourage the model to take full advantage of different types of residual structures, but also gives rise to a learnable residual structure, which can be automatically learned by deep neural models from the data.

4 Experiment

In this section, we evaluate the proposed approach on three representative tasks in the natural language processing field, computer vision field, and cross-modal scenario, that is, image classification, machine translation and image captioning. We first briefly introduce the baseline models for comparison, the datasets, the metrics and implementation details, followed by the discussions about the experimental results. Since our major concern is the combination of different components in residual units, we keep the internal structure (i.e., the residual function $\mathcal{F}(\boldsymbol{x},\mathcal{W})$) of each component unaffected. The training and inference strategies also remain the same as the original models. For more details, please refer to the cited publications.

4.1 Machine Translation

Baselines, Datasets, Metrics and Settings. For the task of machine translation, we adopt the popular Transformer (Vaswani et al., 2017), which is a strong baseline. The model is implemented with the code from tensor2tensor (Vaswani et al., 2018). Transformer follows the encoder-decoder paradigm, but it replaces the self-recursive operation in RNNs with the self-attention that summarizes all context. In each Transformer block, the self-attended results are post-processed by residual connection and layer normalization.

There are 133K, 4.5M, and 36M training

[1] The empirical results also show that $(1-\alpha)(1-\beta)$ performs better than γ (94.05 accuracy vs. 93.95 accuracy in CIFAR-10).

Method	EN-VI	EN-DE	EN-FR
Transformer-Base (Vaswani et al., 2017)			
Baseline	30.9	27.5	38.2
+ Proposal	**32.0**	**27.6**	**38.4**
Transformer-Big (Vaswani et al., 2017)			
Baseline	31.6	28.5	41.0
+ Proposal	**32.2**	**28.7**	**41.3**

Table 3: Results (BLEU) on the machine translation task. Higher means better. The proposal brings consistent and substantial improvements.

Dataset	Baseline	+ Proposal	Improvements
PreAct-ResNet (Vaswani et al., 2017)			
CIFAR-10	93.66	**94.05**	+0.39
CIFAR-100	71.29	**72.91**	+1.62

Table 4: Results (Accuracy (%)) on the image classification task, averaging over 5 runs. Higher is better. The proposal consistently outperforms the baselines as in machine translation. Especially, better improvements are achieved for models on CIFAR-100.

pairs in the IWSLT-2015 English-Vietnamese (EN-VI) (Cettolo et al., 2015), WMT-2014 English-German (EN-DE) and English-French (EN-FR), respectively. `tst2012` and `tst2013` are selected as the development and test sets, respectively, for EN-VI. For EN-DE, we use `newstest2013` and `newstest2014`; and for EN-FR, `newstest2012+2013` and `newstest2014` are selected. For experiments on the two WMT datasets, we follow the implementation settings in Vaswani et al. (2017). For experiments on the IWSLT EN-VI dataset, we set the batch size equal to 4096 and train on single GPU, as it is relatively small. For all datasets, we use a single model by averaging the last 10 checkpoints to produce the results with beam search of 4 and length penalty of 0.6.

Results. The results of machine translation are presented in Table 3. Under both the Base and the Big configuration, our proposal consistently boosts the performance of the Transformer baseline. When equipped with our proposed approach, the Transformer-Base model even transcends the big version, which is three times as large in size, on the EN-VI translation task. It shows that the scaling factors indeed help adjust the residual structure to the data distribution, which is very efficient in exploiting the expressive power of deep residual networks. Encouragingly, an union of the proposal and the Transformer-Big model achieves substantial improvement, and it outperforms the state-of-the-art method (Huang et al., 2018) in the EN-VI low resource dataset.

4.2 Image Classification

Baselines, Datasets, Metrics and Settings. In the computer vision field, we benchmark our proposed learnable residual structure with residual-based image classification systems, i.e., Pre-Activated ResNet (PreAct-ResNet) (He et al., 2016b). ResNet-110 consists of 54 double-layer residual blocks, which makes it non-trivial to optimize. To demonstrate that our *SAS* is applicable to such deep framework, we select ResNet-110 in the experiments. We retain most of the hyper-parameters in He et al. (2016b), with the exception of the weight decay rate, which is set to 0.0002, so as to guarantee more stable training. Both CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) are comprised of colored images for classification. CIFAR-100, which contains 100 classes, appears to be more difficult as compared to CIFAR-10, where there are only 10 classes. Following common practice (He et al., 2016b; Srivastava et al., 2015), accuracy rate of classification over 5 runs are reported as the evaluation results. In Srivastava et al. (2015) and He et al. (2016b), they found that it may be beneficial to attach more importance to the skip connection x for initialization, i.e., the bias term in the transform gate should be initialized with a negative value. Following this practice, in image classification task, we set the bias b_{ff} for α and β to 3 and -3, respectively, at the start of training.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
GLIED (Liu et al., 2019c)	80.4	-	-	**39.6**	**28.9**	58.8	129.3	**22.6**
Transformer (Vaswani et al., 2017)								
Baseline	80.2	64.9	50.7	39.0	28.4	58.6	126.3	21.7
+ Proposal	**81.2**	**65.4**	**51.2**	39.3	28.7	**59.0**	**129.8**	**22.6**
Improvements	+1.0	+0.5	+0.5	+0.3	+0.3	+0.4	+3.5	+0.9

Table 5: Performance on the MSCOCO Karpathy test split. Higher is better in all columns. The baseline enjoys a comfortable improvement with the proposed approach. Additionally, we report the performance of the recently published state-of-the-art GLIED, as we can see, our approach helps the Transformer captioning model outperforms GLIED substantially in terms of CIDEr, which further demonstrates the effectiveness of our approach.

The remaining weights are initialized in the same way as in (He et al., 2016b).

Results. As can be seen in Table 4, consistent improvements are also obtained over the baseline model, which is much deeper than the six-layer Transformer model. The increase in accuracy is 0.39 and 1.62 on the CIFAR-10 and CIFAR-100 dataset, respectively. This demonstrates that our proposal also works in deeper cases. It comes to our notice that the proposal induces better improvements on the more challenging CIFAR-100 dataset. This is presumably that the learnable structure can make the best of each component in the residual building block, which allows more flexible fitting into the multi-class image distribution, resulting in a larger space for improvement in the 100 classes scenario.

4.3 Image Captioning

Baselines, Datasets, Metrics and Settings. To further demonstrate the generalization ability of our proposed approach, we conduct experiments on the task of image captioning. The experiments are based on the multi-head attention mechanism (Vaswani et al., 2017), which has recently shown great potential and is competitive with the most advanced models (Liu et al., 2019c,b), for the reason of which we choose it as our baseline to examine the performance of our approach on the multidisciplinary task.

There are several datasets that consist of image-sentence pairs. Our reported results are evaluated on the popular Microsoft COCO (MSCOCO) (Chen et al., 2015) dataset, which contains 123,287 images. Each image in the dataset is paired with 5 sentences. The results are reported using the widely-used publicly-available splits in the work of Karpathy and Li (2015). The MSCOCO validation and test set contain 5,000 images each. Following common practice (Liu et al., 2018, 2019a), we replace caption words that occur less than 5 times in the training set with the generic unknown word token UNK, resulting in 9,567 words.

We adopt SPICE, CIDEr, BLEU, METEOR and ROUGE for testing. They are previously used as evaluation methods for image captioning, we report the results using the MSCOCO captioning evaluation toolkit (Chen et al., 2015). Among the metrics, BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) are originally designed to evaluate the performance of machine translation systems. ROUGE is widely used to examine the quality of machine-produced summaries. SPICE and CIDEr are bespoke metrics for image captioning, which measures scene graph and n-gram matching, respectively, and we refer to them as primary indicators of model performance.

Results. The results on Karpathy test split (Karpathy and Li, 2015) are reported in Table 5. By using our proposed learnable residual structure, improvements of 3.5 points and 0.9 points in terms of CIDEr and SPICE respectively can be achieved, further demonstrating the effectiveness and generalization capabilities of our approach to a wide range of tasks. More encouragingly, the proposed approach helps the baseline model achieves 129.8 CIDEr score, an improvement over GLIED (Liu et al., 2019c) by 0.5.

5 Analysis

In this section, we conduct several analyses to give further insights into our proposed approach, which are based on the image classification task and we adopt PreAct-ResNet-110 (He et al., 2016b) as the baseline model.

Analysis on Scaling Gate. Table 6 summarizes the obtained results when applying the Transform

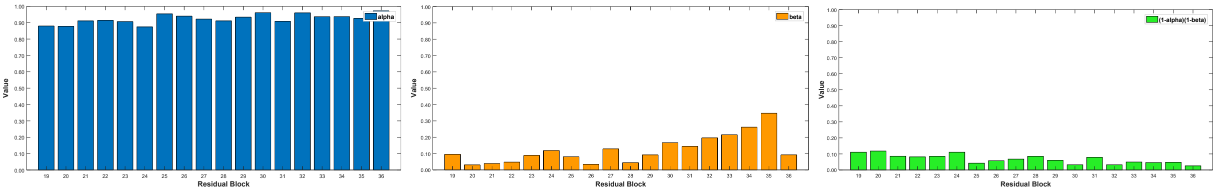


Figure 3: Illustrations of the averaged value of α (left) and β (middle), together with the corresponding $(1-\alpha)(1-\beta)$ (right), which are predicted by the proposed approach for each residual block in pretrained PreAct-ResNet-110 on CIFAR-10.

Methods	CIFAR-10
Baseline (PreAct-ResNet-110)	93.66
+ Transform Gate	92.15
+ Scaling Gate (Single Layer)	92.73
+ Scaling Gate (Full Model)	**93.82**

Table 6: The effects of applying the Transform Gate from Highway Net (Srivastava et al., 2015) and the proposed Scaling Gate to the PreAct-ResNet-110 (He et al., 2016b), where the performance is evaluated by Accuracy(%). The single layer Scaling Gate takes the form $\mathcal{S}(\boldsymbol{x}, \mathcal{F}) = [\boldsymbol{x}; \mathcal{F}]W_f + b_f$.

Architecture	Acc.(%)
$\boldsymbol{x} + \mathcal{F}$ (Baseline) (He et al., 2016b)	93.66
$\alpha * \boldsymbol{x} + \beta * \mathcal{F} + (1-\alpha)(1-\beta) * \mathrm{BN}(\boldsymbol{x} + \mathcal{F})$	93.23
$\alpha * \boldsymbol{x} + \beta * \mathcal{F} + (1-\alpha)(1-\beta) * \mathrm{LN}(\boldsymbol{x} + \mathcal{F})$	**94.05**

Table 7: Results on CIFAR-10 using the PreAct-ResNet-110 with the batch/layer normalization. The batch normalization is less effective than the layer normalization in residual structure.

Gate in Highway Net and our Scaling Gate to PreAct-ResNet-110, as well as the results of the vanilla model. As we can see, when equipped with Transform Gate, the effect is counter-productive on CIFAR-10 dataset. This indicates that the information from $\boldsymbol{x}$ along is not robust and effective enough to predict the scaling factors in the residual structure. The single layer version of our Scaling Gate takes into account the residual branch $\mathcal{F}$, thereby improving over the Transform Gate. It is worth mentioning that compared with the Transform Gate ($\mathcal{T}(\boldsymbol{x}) = \boldsymbol{x}W_f^T + b_f^T$), which has $(h \times h) + h$ learnable parameters, Scaling Gate (Single Layer) only introduces $(2h \times 1) + 1$ learnable parameters, which is much more efficient. By modeling the scaling factor with Scaling Gate (Full Model), a 0.16 points promotion is achieved over the baseline on CIFAR-10, which further demonstrates the advantages and effectiveness of the proposed Scaling Gate.

Analysis on Self-Adaptive Scaling. Averaging over 10,000 experimental examples, we display in Figure 3 the value of α, β and $(1-\alpha)(1-\beta)$ in each residual block of the pretrained PreAct-ResNet-110 on CIFAR-10. The filter sizes for the blocks at the bottom, middle and top of the model are different. We only show the representative blocks at the middle of the model due to space limitation. The first column shows that in almost all cases, α is greater than 0.9, which indicates that identity mapping is very helpful to information transfer and eases the optimization of deep neural networks. This can be attributed to the facilitated backward propagation of error signals by identity mappings. It is shown in the second column that as the number of network layers increases, the value of β grows simultaneously, which indicates that the representation ability of the residual branch $\mathcal{F}$ is stronger when it comes closer to the output of the model. The main reason is that the error signal passed to $\mathcal{F}$ is more adequate in the upper blocks, which is beneficial to optimization. As can be seen from the third column, more importance is assigned to the normalized result when it comes to the lower parts of the entire architecture, which means that the value of $(1-\alpha)(1-\beta)$ is larger. This is because that in the underlying static blocks of deep neural networks, the guidance from error signal is weak and the optimization is unstable, thus making the introduction of layer normalization necessary.

In all, the proposed approach regulates the information from individual components with scaling factors to build the learnable residual structure, which helps make the best of different type of residual structure, resulting in an effective combination for better performance.

Analysis on Using Batch Normalization. The batch normalization is used commonly in the field of computer vision. Therefore, we replace the layer normalization with the batch normalization in the proposed approach to see the difference. As shown

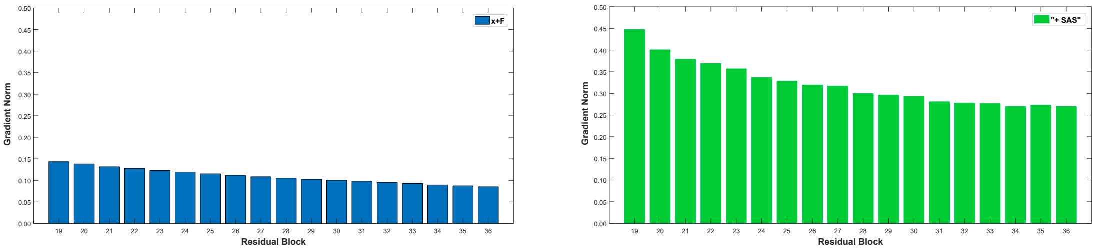


Figure 4: Gradient norm of the output of residual blocks of two different structures that are based on PreAct-ResNet-110. The values are calculated as an average of over 10,000 random examples in the training set of CIFAR-10. We conduct analyses on the blocks from the model's middle part. The *SAS* denotes the self-adaptive scaling approach.

in Table 7, applying batch normalization has a negative effect on the performance. Most importantly, it lags behind the baseline by a noticeable margin, which shows that batch normalization is less effective than layer normalization in the residual structure. We speculate that layer normalization is able to mitigate the training issue in the form of exploding gradient induced by the adjusted ratio provided by the layer normalization's own parameters, while batch normalization could not, which could be intuitively derived by the framework in Hanin and Rolnick (2018). It is probably the reason why He et al. (2016b) observed considerably worse results when they applied batch normalization on the residual structure.

Analysis on Better Optimization Capability. To understand how the proposed approach helps the optimization of deep neural models, we inspect into the gradient norm of the output of each residual block in the pre-trained PreAct-ResNet-110 on CIFAR-10. The gradients are averaged over 10,000 randomly selected training examples. As shown in the left plot of Figure 4, the gradients of the "$x + \mathcal{F}$" structure are basically the same, indicating that all the residual blocks have similar speed for gradient descent and optimization. In contrast, the right plot of Figure 4 reflects that more gradients are allocated to the lower blocks with the help of *SAS*, and the overall gradient values are greater. This phenomenon is interesting and finally gives rise to better results, as shown in Table 4. We conjecture that the layers distant from the model output cannot receive adequate guidance from the error signal, thus requiring more gradient for optimization. Moreover, since the layer normalization is able to stabilize the information flow and accelerate convergence, adaptively incorporating the residual structure with layer normalization can also facilitate optimization. By allocating more importance to layer normalization in the residual blocks via scaling factors, the layers at the bottom of the network can be better optimized, which is in line with the foregoing analysis on Self-Adaptive Scaling approach.

6 Conclusion

In this work, we focus on building a learnable residual structure, which automatically learns the design of residual structure from data, instead of the handy-crafted designs in previous work. We propose the Self-Adaptive Scaling approach to achieve this goal, which combines various residual structures via the predicted scaling factors, resulting in a general residual structure covering several existing models. The proposed approach is simple and can be easily integrated into existing residual-based models. Experiments on the machine translation, image classification and image captioning tasks validate the effectiveness of the proposed method, which successfully promotes the performance of all the strong baselines. This also demonstrates the generalization ability of our method. In particular, when being applied to the recently proposed Transformer model, our approach establishes new state-of-the-arts on the IWSLT EN-VI low resource machine translation task, which further substantiates its efficiency. Detailed analyses prove that the proposed approach can also promote the optimization ability of deep neural networks, and is conducive to exerting the expressive power of existing models.

Acknowledgments

This work was partially supported by the Shenzhen Fundamental Research Project (No. ZDSYS201802051831427). We thank all the anonymous reviewers for their constructive comments and suggestions.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign. In *IWSLT 2015, International Workshop on Spoken Language Translation*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Boris Hanin and David Rolnick. 2018. How to start training: The effect of initialization and architecture. In *NeurIPS*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Identity mappings in deep residual networks. In *ECCV*.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2018. Towards neural phrase-based machine translation. In *ICLR*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137. IEEE Computer Society.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, Computer Science Department, University of Toronto.

Fenglin Liu, Meng Gao, Tianhao Zhang, and Yuexian Zou. 2019a. Exploring semantic relationships for image captioning without parallel data. In *ICDM*.

Fenglin Liu, Yuanxin Liu, Xuancheng Ren, Xiaodong He, Kai Lei, and Xu Sun. 2019b. Aligning visual regions and textual concepts for semantic-grounded image representations. In *NeurIPS*.

Fenglin Liu, Xuancheng Ren, Yuanxin Liu, Kai Lei, and Xu Sun. 2019c. Exploring and distilling cross-modal information for image captioning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5095–5101. ijcai.org.

Fenglin Liu, Xuancheng Ren, Yuanxin Liu, Houfeng Wang, and Xu Sun. 2018. simnet: Stepwise image-topic merging network for generating detailed and comprehensive image captions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 137–149. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318. ACL.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, François Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. In *AMTA*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Barret Zoph and Quoc V. Le. 2017. Neural architecture search with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.