Control the number of skip-connects to improve robustness of the NAS algorithm

Bao Feng Zhang1 | Guo Qiang Zhou1,2

1School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
2State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Abstract
Recently, the gradient-based neural architecture search has made remarkable progress with the characteristics of high efficiency and fast convergence. However, two common problems in the gradient-based NAS algorithms are found. First, with the increase in the training time, the NAS algorithm tends to skip-connect operation, leading to performance degradation and instability results. Second, another problem is no reasonable allocation of computing resources on valuable candidate network models. The above two points lead to the difficulty in searching the optimal sub-network and poor stability. To address them, the trick of pre-training the super-net is applied, so that each operation has an equal opportunity to develop its strength, which provides a fair competition condition for the convergence of the architecture parameters. In addition, a skip-controller is proposed to ensure each sampled sub-network with an appropriate number of skip-connects. The experiments were performed on three mainstream datasets CIFAR-10, CIFAR-100 and ImageNet, in which the improved method achieves comparable results with higher accuracy and stronger robustness.

1 | INTRODUCTION

In the field of artificial intelligence, the neural network algorithm has made remarkable achievements in image, voice, and other areas. Zhao [1] proposes a novel super-resolution regularization model based on adaptive sparse representation and self-learning frameworks. Zeng [2] presents a weighted structural local sparse appearance model to further improve the robustness of tracking. Luo [3] proposes a fast and effective pruning method to eliminate the outlier correspondences, especially the sharp ones. However, artificial neural networks require a great deal of specialized knowledge and repetitive experiments.

In recent years, the Neural Architecture Search (NAS) algorithm, which can design high performance networks automatically, has received extensive attention from researchers, and many new network models have evolved [4–6]. NAS algorithms based on reinforcement learning [7], evolutionary algorithm [8, 9], Bayesian optimization [10], and gradient constitute the four development directions of the neural structure search.

ENAS [6] reduced the structure search time to day-level. Gradient-based NAS has become the focus of research due to its advantages of fast search speed and high accuracy. The first gradient-based method [11] made a breakthrough in search time, and follow-up works [12, 13] and used Gumbel-softmax [14] to alleviate the gap problem.

However, one problem named ‘skip aggressive’ is not resolved as the gradient-based NAS method tends to skip-connect operate with the increase in training time. Also, there is another common problem with current methods. NAS algorithm based on cell has a vast search space, but the number of candidate networks sampled is far less than the size of the search space.

In this article, we propose a skip-controller to limit the number of skip-connects in the search process, which not only solve the problem of ‘skip aggressive’, but also avoids the waste of resources on sampling and training meaningless sub-networks with too many skip-connects. In summary, the strengths of our method lie in the following aspects comparing to previous NAS methods:

1. Through limiting the amount of skip-connects to a reasonable range, significantly narrowing the search space and reducing the difficulty of searching the optimal solution.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. IET Computer Vision published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

356 | IET Comput. Vis. 2021;15:356–365.
Liu et al. proposed DARTS (Differentiable Architecture Search) [11], the first gradient-based NAS. Based on DARTS, SNAS [12] and Fbnet [13] proposed to use Gumbel-softmax to reduce the gap between the derived sub-network and the superNet. GDAS [17] further uses Gumbel-softmax to speed up search. However, they don’t address the skip-connect aggressive issue in DARTS.

P-DARTS [18], aiming at the problem of the excessive number of skip-connects, uses path drop-out to limit the unfair advantage of skip-connect in the search process and replaces excessive skip-connects with other operations at the end of the search. This is an Ex-post solution, and more attention should be paid to avoid being misled in the process.

FairDARTS [19] trains the structure parameters of different operations independently to avoid interference between various operations, to make a fair competition between different operations, alleviating excessive skip-connects. However, the above methods ignore the vast search space of the cell-based NAS algorithm, and the number of candidate networks sampled in the whole search process is far less than the size of the search space.

DARTS+ [20] believes that performance degradation will occur when the number of skips exceeds two during the search process. It uses the early-stop strategy to avoid performance degradation. Similar to DARTS+, we also introduce a prior experience that the accuracy of models with more than four skip-connects is almost never more than 97%. Unlike DARTS+’s early stop strategy, that only gets a structure with two skips, we reduce the search space into a reasonable range to search a network structure. In addition, our method avoids sampling and training meaningless sub-networks, which greatly improves the search efficiency.

2 | RELATED WORK

2.1 Preliminary: gradient-based NAS

For convolutional neural networks, DARTS searches for a normal cell and a reduced cell to construct the final network model. Each cell can be defined as a directed acyclic graph, which contains N nodes, and each node represents a feature map. In the experiment, the default setting is $N = 7$, which contains two input nodes, four intermediate nodes, and one output node. The edge $e_{ij}$ from node $i$ to $j$ operates on the input feature $x_i$ and its output is denoted as $o_{ij}(x_i)$. Each edge has an optional set of operations $L = \{o_{ij}, o_{ij}^{\prime}, \ldots, o_{ij}^{\prime\prime}\}$. DARTS uses softmax to select the operations to be a mixed operation. So the corresponding mixed operation of edge $e_{ij}$ is:

$$
\pi_{ij}(x) = \frac{\exp(\alpha_{oi})}{\sum_{o \in O} \exp(\alpha_{oi})} o(x),
$$

where $\alpha_{oi} = \{\alpha_{oi}^1, \alpha_{oi}^2, \alpha_{oi}^3, \ldots, \alpha_{oi}^M\}$ is the probability vectors corresponding to the operations on edge $e_{ij}$.

SNAS and Fbnet use Gumbel-softmax instead of softmax to select the operations so that the structure parameters converge into a softened one-hot variables. The corresponding mixed operation of edge $e_{ij}$ is:

$$
Z_{ij}^k = f_{a_{ij}}(G_{ij}^k)
= \frac{\exp((log(a_{ij}^k) + G_{ij}^k) / \lambda)}{\sum_{l=0}^{n} \exp((log(a_{ij}^l) + G_{ij}^l) / \lambda)}
$$

$$
\pi_{ij}(x) = Z_{ij}^0 \ast a_{ij}(x)
$$

where $Z_{ij}$ is the softened one-hot random variable for operation selection at edge $e_{ij}$, $G_{ij}^k = -log(-log(U_{ij}^k))$ is the kth Gumbel random variable, $U_{ij}^k$ is a uniform random variable.

GDAS no longer use the softened one-hot random variables, which need $M$ times computational memory and time. Instead, parameters are sampled to be strictly one-hot vectors to reduce computation time and memory.

$$
\bar{\pi}_{ij}(x) = f(Z_{ij}^0) \ast a_{ij}(x)
$$

where $f$ is a function that changes soften one-hot variables to one-hot variables.

Each intermediate node $x_j$ sums the inputs of all income $x_i$:

$$
x_j = \sum_{i < j} a_{ij}(x_i)
$$

The output node of the cell cats all the intermediate nodes:

$$
Out = cat(x_2, x_3, \ldots x_{M-1})
$$

The neural network search problem is to optimize the architecture parameters $a_{ij}$ and the network operation parameters $w^*$ to minimize the validation loss.
At present, all the gradient learning methods, except SNAS, use bi-level optimization. In this article, we also use bi-level optimization and the same search space as DARTS. However, we no longer use softmax for operation selection but use Gumbel-softmax to get one-hot architecture parameters.

3.2 | Existing problems

- **Problem 1**: The unfair advantage of skip-connect leads to performance degradation. Specifically, skip-connect has an unfair advantage that is easy to be sampled, and this advantage is constantly amplified in iterations. As a result, too many skip-connects are eventually produced. This advantage leads to an increase in skip-connect, which in turn leads to Performance decline to be more precise, the accuracy of networks with the skip-connect value of more than four hardly exceeds 97%. Consider the cell of DARTS as an example. Each cell contains 14 edges need to select operations, which means the number of skip-connects can be 0–14, in other words, most of the networks sampled are meaningless. That is, there is no reasonable allocation of computing time and resources.

- **Problem 2**: A cell consists of seven nodes that include four intermediate nodes, so a cell contains 14 paths that are required to select the operations, while each edge includes eight operations. When the sampled architecture parameter is a one-hot tensor, that is, activating only one operation for each path, so the size of the search space is $8^{14}$. In the previous gradient-based search methods, with the settings of $\text{batch size} = 64$, $\text{epochs} = 200$, $\text{train dataset: validate dataset} = 1:1$, the number of sampled networks is only: $200 \times 25,000/64 = 78,125$, which is far less than the amount of search space $8^{14}$. This means that the previous search strategy is not sufficient to explore the search space, and it is hard to cover the optimal network sub-network.

4 | METHOD

4.1 | Pre-training superNet similar to swarm up

For Problem 1, to mitigate the unfair advantage of skip-connection, an idea similar to the swarm up is proposed, known as the ‘pre-train the superNet’. In the early stage of the neural search process, sampling was completely random and only the operating parameters were trained, so each candidate sub-network has the same training opportunity. After pre-training the superNet, different operations have the same degree of convergence, providing a fair and reasonable competition platform for architecture convergence. Specifically, training the superNet for $K$ epochs firstly and then using bi-level optimization to train architecture parameters and network parameters in the remaining 200-k epochs. The pseudo code is shown in Algorithm 1.

| Algorithm 1 Pre-train the SuperNet |
|------------------------------------|
| **Input**: superNet; train dataset; evaluate dataset. |
| **Parameter**: K |
| **Output**: searched cell |
| 1: Let epoch = 0 |
| 2: while epoch < $K$ do |
| 3: for each batch size train data do |
| 4: Sample a sub-network and train its operating parameters only; |
| 5: epoch = epoch + 1 |
| 6: end for |
| 7: end while |
| 8: while epoch < 200 do |
| 9: for each batch size eval data do |
| 10: Sample a sub-network |
| 11: train and update the architecture parameters; |
| 12: Sample a sub-network |
| 13: train and update its operating parameters only; |
| 14: epoch = epoch + 1 |
| 15: end for |
| 16: end while |
| 17: return searched cell |

4.2 | Search space exploration strategy

For Problem 2, the search space is enormous, but the exploration of the search space is insufficient. Also, as the prior experience in Problem 1, in the NAS process, a large number of meaningless sub-networks with more than $S$ skip-connects in each cell are sampled which waste many computational resources.

To fully and reasonably explore the search space, we propose two methods to reduce the search space and avoid sampling meaningless sub-networks, and further increase the number of candidate network samples.

First, a skip-controller is proposed that only samples meaningful sub-networks within the $S$ skip-connects. The analysis of the parameter $S$ is detailed in Section 5.3. On the one hand, the search space is greatly reduced from the initial $8^{14}$ to $8^{14} \times (C_{14}^0 + C_{14}^1 + C_{14}^2 + C_{14}^3 + C_{14}^4)/2^{14} \approx 8^{14} \times 0.21$, which reduces the difficulty of searching the optimal network. On the other hand, since we will only train networks within the $S$ skip-connects, the final searched network must be one of the narrowed search spaces, which solve the problem of the excessive number of skip-connects mentioned in Problem 1, which also helps the robustness of the search results. The pseudo code of skip-controller is shown in Algorithm 2.
Algorithm 2 Skip-controller
Input: architecture parameters $\alpha$
Parameter: $S$
Output: Operations_weights
1: Operations_weights = Gumbel_softmax_sample ($\alpha$).
2: if skip_num of Operations_weights < $S$ then
3: return Operations_weights
4: else
5: while skip_num of Operations_weights > $S$ do
6: Operations_weights = Gumbel_softmax_sample($\alpha$)
7: end while
8: return Operations_weights
9: end if

Second, increase the number of samples. It is easy to find from Algorithm 1 that the number of samples is related to epochs, batch_size, and the size of the training data set.

$$samples = epochs \times \text{train\_dataset\_size}/\text{batch\_size} \quad (7)$$

So we adjust the data set ratio from 1:1 to 4:1, increasing the number of sampled sub-networks by 60% without increasing search time. Specifically, we can express it based on pseudocode 1 as Algorithm 3.

It is easy to find that through these two tricks, not only greatly reduces the search space, but also increases the scope of exploration. Therefore, our method is more conducive to find network structures with excellent performance. Our method uses the same search space as DARTS but analyse the problem from the perspective of the candidate models. During the training, only meaningful candidate sub-networks has been sampled, which contain the optimal solution with a high probability. Furthermore, only sub-networks sampled will be trained, so their loss is lower than the network model which is not sampled. The optimization of structural parameters will be updated tending to meaningful sub-networks.

Algorithm 3 Pre-training superNet with skip-controller
Input: superNet, train dataset, evaluate dataset.
Parameter: $K, S$
Output: searched cell
1: Let epoch = 0
2: while epoch < $K$ do.
3: for each batch size train data do.
4: Sample a sub-network within $S$ skip -connects;
5: train and update the sub-network;
6: epoch = epoch + 1
7: end for
8: end while
9: while epoch < 200 do
10: for each batch size eval data do
11: sample a sub-network within $S$ skip-connects;
12: train and update the sub-network;
13: sample a sub-network randomly
14: update the architecture parameters
15: epoch = epoch + 1
16: end for
17: end while
18: return searched cell

4.3 Resource constraint

Although limiting the number of skip-connects in the search space can bring stability to the search results and reduce the difficulty of searching the optimal sub-network, the advantages of the structure with fewer parameters are offset, which lead to the network that tends to include the operation with parameters, resulting in a model with many parameters. To solve this problem, we refer to the method of Morphnet [21], which restricts the network model from three perspectives during the search process: model parameter size, the number of float-point operations (FLOPs), and the memory access cost (MAC) [12]. We define $H, W$ and $f, k$ as the output spatial dimensions and the filter dimensions, respectively, and use $I, O$ for the number of input and output channels, while $g$ represents the number of group convolution. According to Morphnet, for a convolutional layer, the three can be calculated in the following formula:

$$\text{Parameter size} = \frac{fkI}{g}$$
$$\text{FLOPs} = HWfkI/g$$
$$\text{MAC} = HW(I + O) + fkI/g \quad (8)$$

According to SNAS, to distinguish pooling operations and skip-connect operation. They should be defined separately. FLOPs of pooling:

$$\text{FLOPs} = HWfkI \quad (9)$$

FLOPs of skip-connection:

$$\text{FLOPs} = 0 \quad (10)$$

MAC of pooling and skip-connection:

$$\text{MAC} = HW(I + O) \quad (11)$$

Therefore, the final NAS can be redefined as an optimization problem:
\[
\begin{align*}
\min \ L_{\text{val}}(w^*(\alpha), \alpha) + \eta * C(\alpha) \\
\text{s.t.} \ w^*(\alpha) = \arg\min_\omega \ L_{\text{train}}(w, \omega) \\
C(\alpha) = \text{Parameter size} + \text{FLOPs} + \text{MAC}
\end{align*}
\]

In which \(C(\alpha)\) is the cost for the sub-network associated with the random variables \(\alpha\) that decides which operation be sampled for training \(\eta\) is the coefficient of \(C(\alpha)\).

Since the cost of each operation is determined, \(C(\alpha)\) is linear in terms of the one-hot vector \(\alpha\). In other words, the cost of each sampled subnet is determined by the one-hot type structure parameter Multiplied by the cost of each operation. Therefore, it can be added to the loss function in the form of a regular term, so that the training process tends to be a network structure with a low computational cost.

4.4 Data enhancement adjustment

Through experiments, we found that the verification data set should not adopt a data enhancement strategy like the training data set. We process the validation dataset in the same way as the test dataset to achieve a higher accuracy. This may be explained that architecture parameters are few and do not require overly complex data samples. Besides, data enhancement is more suitable for training operation parameters to prevent over-fitting. The architecture parameters are mainly used to distinguish the performance differences between the operations, while data enhancement may confuse in identifying performance differences.

5 EXPERIMENT

5.1 Experiment setups

Searching setup: Our method is based on the Gumbel-softmax to sample a sub-network, instead of SNAS and Fbnet to train the whole network model, so taking GDAS as baseline. The whole search process consists of two stages, which contains cell searching and training the final network. Different from GDAS, the 50k images were divided into 40k training dataset and 10k validation dataset. Following DARTS, the number of initial channels in the first convolutional layer is 16, and the number of computational nodes for each cell is 4. Besides, the number of layers in one block \(M\) is set as 2, and the total training epochs are 200. The candidate operations contain eight different functions: (1) identity, (2) zero, (3) 3 x 3 depth-wise separate conv (4) 3 x 3 dilated depth-wise separate conv, (5) 5 x 5 depth-wise separate conv, (6) 5 x 5 dilated depth-wise separate conv, (7) 3 x 3 average pooling, (8) 3 x 3 max pooling. We set \(K = 50, S = 4, \) and \(\eta = 10^{-6}\) in the experiment, getting three version models, pre_trained(Pre), pre_trained + skip_controller(Pre + skip), and pre_trained + skip_controller + resource_constra int (Pre + skip + res). Meanwhile, the SGD is applied to optimize the network weights with a learning rate of 0.025 at the beginning and the anneal it down to \(10^{-3}\) following a cosine schedule. The weight decay is set to be \(10^{-3}\), and the momentum is set to be 0.9. The temperature of Gumbel soft-max is initialized as 10 and is linearly reduced to 0.3. Adam optimization with the weight decay of \(10^{-3}\) and the learning rate of \(3 \times 10^{-4}\).

5.2 Experimental results

Results on CIFAR: After finishing the search procedure, we stack the cell with the best performance to form a CNN with cell number set to be 20, block number set to be 6, and node number set to be 4. When the complete network is built, the learning rate is initialized to be 0.025 and anneal it to zero with the cosine learning rate schedule. Similar to GDAS, we train the network by 600 epochs in total. Following existing works [5], additional enhancements consist of cutout, probability of path drop-out being 0.2 and auxiliary towers [22] with weight 0.4. Also, other standard pre-processing and data augmentation such as horizontal flipping and normalization are applied to the network learning. The experimental results of the three versions and other state-of-the-art results are listed in Table 1. The accuracies of the three versions are higher than that of the other Gumbel-based gradient methods. From the Table 2, it is easy to find when adding skip_controller, the stability of the results is significantly enhanced. Finally, using the resource constraint trick for model compression can reduce the parameter size significantly with little effect on accuracy.

Results on ImageNet: The ILSVRC2012 is used to test the cells discovered on CIFAR-10. Following GDAS, we train the model with 14 cells and 50 initial channels from scratch for 250 epochs with batch size 512 on four Nvidia Titan XP GPUs, which takes 6 days with 5.0Mb parameters. The network parameters are optimized using an SGD optimizer with an initial learning rate of 0.2 (decayed linearly after each epoch), a momentum of 0.9, and a weight decay of \(3 \times 10^{-5}\). Additional enhancements including label smoothing and auxiliary loss tower are used during training. Because large batch size and learning rates are adopted, learning rate warmup are also used for the first five epochs. In the end, we take the architecture searched using pre_GDAS and pre_GDAS + skip_controller to evaluate on ImageNet which achieves accuracies of 74.6\% and 74.9\%, which surpasses the previous Gumbel-based algorithms. On ImageNet, the accuracy of pre_skip(pre_trained + skip_controller) again exceeds that of pre_GDAS(pre-train the superNet-based on GDAS), indicating its high migration capability. The evaluation results and other previous state-of-the-art methods are compared in Table 3. It is easy to find that our method outperforms the baseline GDAS by about 1.5\%, which proves the success of our improvement. Furthermore, when transforming the architecture searched on CIFAR-100 to ImageNet, its accuracy 75.6\% is further improved.
Ablation study

5.3.1 | Skip_controller is friendly for accuracy

In order to prove the effectiveness of skip_controller, we take GDAS as the baseline and only add skip_controller without changing other operations, and the model is named as GDAS_skip. As shown in the Table 4, the accuracy of the GDAS_skip was higher than the original GDAS, even GDAS (FRC), which is obtained by using GDAS based on artificial design reduce cell. It shows that skip_controller can indeed reduce the search space apart from reducing the difficulty of searching high-quality network models.

5.3.2 | Pre-train the superNet to get higher accuracy

To illustrate that the pre-training superNet helps level the playing field between operations and improves accuracy, we pre-train the superNet based on GDAS and name it as Pre_GDAS. Different from GDAS, we believe that when a one-hot vector was used to sample a sub-network model, the sub-network models should be sampled as many as possible to increase the probability of covering the best sub-network. Therefore, setting the ratio between the training data set and the verification data set to 4:1. In our experiment, the whole experiment is conducted with different random seeds for five times. The results are listed in Table 5. It is easy to find that our method outperforms the baseline GDAS.

### Table 1

classification errors and previous state-the-art methods on CIFAR. EV, RL, and the G mean evolution method, reinforce learning method, and gradient-based method respectively, while Gumbel is also a gradient method based on Gumbel-softmax. It is worth noting that our average error rate of 2.68% with a low variance of 0.0031% is much lower than other Gumbel-SoftMax methods, and our best result on CIFAR-10 is 2.57%. In particular, our algorithm improves the accuracy by 1% on CIFAR-100 compared to GDAS

| Architecture | Test Error on CIFAR (%) |
|--------------|-------------------------|
| DenseNet-BC  | 3.46 17.18 25.6 – Manual |
| NASNet-A + cutout [5] | 2.65 – 3.3 1800 RL |
| AmoebaNet-A + cutout [9] | 3.34 – 3.2 3150 EV |
| AmoebaNet-B + cutout | 2.55 – 2.8 3150 EV |
| Hierarchical Evolution [23] | 3.75 – 15.7 300 EV |
| PNAS [24] | 3.41 – 3.2 225 SMBO |
| ENAS + cutout [6] | 2.89 – 4.6 0.5 RL |
| DARTS (first order) + cutout [11] | 3.00 17.76 3.3 1.5 G |
| DARTS (second order) + cutout | 2.76 17.54 3.3 4.0 G |
| DARTS+ [20] | 2.50 16.28 3.7 1.5 G |
| P-DARTS [18] | 2.50 16.55 3.4 1.5 G |
| ProxylessNAS + cutout [25] | 2.08 – 5.7 4.0 G |
| FairDARTS [19] | 2.59 – 3.27 0.42 G |
| SNAS + mild + cutout [12] | 2.98 – 2.9 1.5 Gumbel |
| SNAS + moderate + cutout | 2.85 – 2.8 1.5 Gumbel |
| SNAS + aggressive + cutout | 3.10 – 2.3 1.5 Gumbel |
| GDAS + cutout [17] | 2.93 18.38 3.4 0.21 Gumbel |
| GDAS(FRC) + cutout | 2.82 18.13 2.5 0.17 Gumbel |
| Pre(CIFAR-10) | 2.75 18.22 3.6 0.21 Gumbel |
| Pre(CIFAR-100) | 2.87 17.60 3.8 0.21 Gumbel |
| Pre + skip | 2.68 17.28 4.1 0.21 Gumbel |
| Pre + skip + res | 2.69 17.19 3.3 0.21 Gumbel |

### Table 2

Pre_GDAS + skip_controller with an average accuracy of 2.68% and variance of 0.07. Both accuracy and stability of the Pre_GDAS + skip_controller are better than the Pre_GDAS, which proves the effect of skip-controller in improving accuracy and robustness.

| Experiment No. | Error on CIFAR-10 |
|----------------|-------------------|
| Pre_GDAS + skip_controller | 2.57% |
| Pre_GDAS + skip_controller | 2.62% |
| Pre_GDAS + skip_controller | 2.72% |
| Pre_GDAS + skip_controller | 2.73% |
| Pre_GDAS + skip_controller | 2.76% |

5.3 | Ablation study

5.3.1 | Skip_controller is friendly for accuracy

In order to prove the effectiveness of skip_controller, we take GDAS as the baseline and only add skip_controller without changing other operations, and the model is named as GDAS_skip. As shown in the Table 4, the accuracy of the GDAS_skip was higher than the original GDAS, even GDAS (FRC), which is obtained by using GDAS based on artificial design reduce cell. It shows that skip_controller can indeed reduce the search space apart from reducing the difficulty of searching high-quality network models.

To illustrate that the pre-training superNet helps level the playing field between operations and improves accuracy, we pre-train the superNet based on GDAS and name it as Pre_GDAS. Different from GDAS, we believe that when a one-hot vector was used to sample a sub-network model, the sub-network models should be sampled as many as possible to increase the probability of covering the best sub-network. Therefore, setting the ratio between the training data set and the verification data set to 4:1. In our experiment, the whole experiment is conducted with different random seeds for five times. The results are listed in Table 5. It is easy to find that our method outperforms the baseline GDAS.
5.3.3 | Skip-controller contributes to further improvement of accuracy and robustness

As mentioned above, DARTS is inherently unstable, so it is Pre_GDAS. As shown in Table 5, although the results of Pre_GDAS are higher than that of the GDAS, the result variance is large, which means it is not robustness. Our robustness is reflected in two aspects, first, the model searched by our method will not have too many skip-connections. Second, because we use a skip-controller to make the search space in a smaller range, the variance of the accuracy of the search model will also decrease. To further illustrate that skip-controller has the effect of enhancing stability and improving accuracy, adding skip_controller based on Pre_GDAS to obtain the version Pre_GDAS + skip_controller with the highest accuracy. Different random seeds are applied to repeat the experiment from scratch for five times. Comparing Tables 5 and 2, it is easy to find that with the help of the skip-controller, the accuracy rate is improved, and the variance is

### Table 3
Classification errors of our method and the previous state-of-the-art methods on ImageNet + x indicates the number of multiply-add operations. It is easy to see that our approach is superior to our baseline GDAS, and even comparable to the latest state of the art methods FairDARTS and DARTS+

| Architecture          | Top1 | Top5 | Params (M) | GPU-days |
|-----------------------|------|------|------------|----------|
| Inception-v1 [22]     | 30.2 | 10.1 | 6.6        | 1448     |
| ShuffleNet [26]       | 26.4 | 10.2 | 5          | 524      |
| NASNet-A [5]          | 26.0 | 8.4  | 5.3        | 564      |
| NASNet-B              | 27.2 | 8.7  | 5.3        | 488      |
| NASNet-C              | 27.5 | 9.0  | 4.9        | 558      |
| AmoebaNet-A [9]       | 25.5 | 8.0  | 5.1        | 555      |
| AmoebaNet-B           | 26.0 | 8.5  | 5.3        | 555      |
| AmoebaNet-C           | 24.3 | 7.6  | 6.4        | 570      |
| PNAS [24]             | 25.8 | 8.1  | 5.1        | 588      |
| MnasNet [27]          | 25.2 | 8.0  | 4.4        | 388      |
| DARTS (second order)  | 26.7 | 8.7  | 4.7        | 574      |
| DARTS + [20]          | 23.7 | 7.2  | 5.1        | 591      |
| P-DARTS [18]          | 24.4 | 7.4  | 4.9        | 557      |
| ProxylessNAS (GPU)    | 24.7 | 7.5  | 7.1        | 465      |
| FairDARTS [19]        | 24.9 | 7.5  | 4.8        | 541      |
| SNAS (mild constraint) | 27.3 | 9.2  | 4.3        | 522      |
| Fbnet [13]            | 25.1 | –    | 5.5        | 375      |
| GDAS [C = 50] [17]    | 26.0 | 8.5  | 5.3        | 581      |
| GDAS (FRC) [C = 52]   | 27.5 | 9.1  | 4.4        | 497      |
| Pre(C10)              | 25.4 | 7.86 | 4.5        | 493      |
| Pre(C100)             | 24.4 | 7.44 | 5.5        | 603      |
| Pre + skip(C10)       | 25.1 | 7.8  | 5.0        | 547      |

### Table 4
Skip_controller helps improve accuracy

| Experiment No. | Error on CIFAR-10 |
|----------------|-------------------|
| GDAS           | 2.93%             |
| GDAS(FRC)      | 2.82%             |
| GDAS + skip-controller | 2.71%           |
| GDAS + skip-controller | 2.73%           |

### Table 5
Pre-training superNet before training the structure parameters can help improve accuracy. We use different random factors for five experiments, and get an average accuracy of 97.25% better than the baseline and a variance of 0.14

| Experiment No. | Error on CIFAR-10 |
|----------------|-------------------|
| GDAS(Baseline) | 2.93%             |
| Pre_GDAS1      | 2.57%             |
| Pre_GDAS2      | 2.65%             |
| Pre_GDAS3      | 2.73%             |
| Pre_GDAS4      | 2.86%             |
| Pre_GDAS5      | 2.95%             |
greatly reduced to half of the original. So our skip-controller can indeed improve the accuracy and robustness of the algorithm.

To compare the migration capability, we select the best models from them and transfer them to CIFAR-100 and ImageNet, respectively. Pre_skip still achieved better results than Pre_GDAS in Table 6, which proves that Pre_skip has better migration capability.

5.3.4 | Analysis of parameter $S$ in skip-controller

We set up skip-controller mainly to avoid training meaningless sub-networks and constrain the search space within the range of skip-connects less than four to avoid skip aggressive problems. We randomly sampled sub-networks with skip-connects greater than four in the superNet and tested the accuracy. We found that skip-connects over four are unlikely to exceed 97%. Therefore, it is not necessary to sample and train a sub-network structure with skip-connects greater than 4. Constraining the number of skip-connects can not only avoid the waste of computing resources but also avoid the instability and performance degradation caused by the excessive number of skip-connects.

For each $S$, we perform four repetitive experiments, and the results are shown in the Table 7. When $S$ is large, the skip-controller is almost useless, and the final search model has more skip-connects and the network structure is too sparse.

| Experiment No.                           | Test Error(%) |
|------------------------------------------|---------------|
| Pre_GDAS                                 | 18.22(CIFAR-100) |
| Pre_GDAS + skip_controller               | 17.28(CIFAR-100) |
| Pre_GDAS                                 | 25.38(ImageNet) |
| Pre_GDAS + skip_controller               | 25.08(ImageNet) |

When $S$ is too small, the candidate sub-networks in the search space have more parameters. However, more parameters do not mean higher accuracy, and a huge model with redundant connections is not a good result. Therefore, $S$ should be of an appropriate value, which can avoid skip aggressive problems and avoid too large search models.

Setting $S$ to 2 does not mean that the accuracy must be higher. It just means sampling in a search space with skip numbers of 0, 1, and 2. We found that if this restriction is too strong, the search process will tend to form sub-networks with more parameters (with few or no skips). This is because the sub-network samples have more parameters and lack a sparse network structure. Just like the general machine learning algorithm, when there are too many positive samples for certain features and lack of negative samples, the learning algorithm will tend to it, forming a kind of over-fitting. When the skip-controller constraint is too strong, the samples (candidate sub-networks) are used to train the structural parameters to have the characteristics of several parameters, but the lack of candidate sub-networks with sparse structure, makes the structural parameters tend to be redundant with more parameters during the convergence process, bringing another kind of over-fitting. This may lead to the worst result: the final network structure is too redundant.

When $S$ is small, it tends to produce a network with huge parameters, which produces over-fitting. When $S$ is larger, it tends to produce a sub-network structure with too many skips. Therefore, the constraint on skip needs an appropriate value. Of course, $S$ can also set to be 2, and it is still possible to search for sub-network structures with good accuracy. But the probability of getting too many parameters is greater. Therefore, setting four is a reasonable choice that can eliminate meaningless sub-networks and make the search space reasonable, and can obtain a small number of parameters and high accuracy.

DARTS+ is carried out in the entire search space. Sub-networks with too many skips are included, which is prone to skip aggressive problems. To avoid too many skip-connects

| $S$ | Number of Skip in Models | Model 1 | Model 2 | Model 3 | Model 4 |
|-----|--------------------------|---------|---------|---------|---------|
| 1   | 0                        | 0       | 0       | 0       | 1       |
| 2   | 0                        | 1       | 1       | 2       |         |
| 3   | 0                        | 1       | 2       | 2       |         |
| 4   | 2                        | 2       | 3       | 4       |         |
| 5   | 3                        | 5       | 5       | 6       |         |
| 6   | 4                        | 5       | 5       | 6       |         |
| 7   | 4                        | 5       | 5       | 6       |         |

| Experiment No.                           | Model Size | Error on CIFAR-10 |
|------------------------------------------|------------|-------------------|
| Pre_GDAS + skip_controller + resource    | 3.58 MB    | 2.75%             |
| Pre_GDAS + skip_controller + resource    | 3.60 MB    | 2.60%             |
| Pre_GDAS + skip_controller + resource    | 2.95 MB    | 2.73%             |

| Experiment No.                           | Model Size | Error on CIFAR-10 |
|------------------------------------------|------------|-------------------|
| Pre_GDAS                                 | 3.58 MB    | 2.75%             |
| Pre_GDAS                                 | 3.60 MB    | 2.60%             |
| Pre_GDAS                                 | 2.95 MB    | 2.73%             |
During the NAS process, it stops when \( \text{skip} = 2 \). Unlike the early stop, we search in the skip-restricted search space and solve the skip aggressive problem from the perspective of the search space.

Our method limits the search space to a reasonable range based on excluding meaningless candidate sub-networks. In other words, the final network structure must be one of this search spaces. Therefore, this can avoid the skip aggressive problem at the root from the perspective of the search space. So even if other operations in the cell change, it can still avoid the harm caused by excessive skips.

5.3.5 Resource constraint

Unfortunately, \( \text{Pre} \text{skip} \) results in a large model size around 4.1 MB. So resource constraint is applied for the model compression. We find that the parameter size decreases significantly with little influence on accuracy, in Table 8.

6 CONCLUSION

A more robust NAS algorithm with a skip-controller is proposed. We pre-train the superNet to mitigate aggressive skip-connects. Furthermore, the skip-controller is applied to limit the number of skips and avoid computing on meaningless sub-networks, bringing about the stability of the NAS algorithm. Finally, the resource constraint is used to effectively compress the model.

Through experiments on CIFAR and ImageNet, our method is superior to other Gumbel-softmax-based baseline methods and is comparable to the state-of-the-art methods, with the error rates of 2.57% and 17.19% on CIFAR-10 and CIFAR-100, respectively. When transferred to the ImageNet, it reaches the test error of 24.4%.

In the future, we will try to search on the ImageNet directly, since our algorithm does not require a huge memory consumption. Also, we will test the validity of the skip-controller on other tasks.

ACKNOWLEDGMENT

This work was supported by the NSF of China 2017–2020 (grants 61,671,260).

REFERENCES

1. Zhao, J.W., Hu, H.P., Cao, F.L.: Image super-resolution via adaptive sparse representation. IET Comput. Vis. 12(5), 753–761 (2018)
2. Zeng, X., et al.: Dual-scale weighted structural local sparse appearance model for object tracking. IET Comput. Vis. 13(2), 146–156 (2019)
3. Luo, N., Wang, Q.: Effective outlier matches pruning algorithm for rigid pairwise point cloud registration using distance disparity matrix. IET Comput. Vis. 12(2), 220–232 (2018)
4. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. In Proc. Int. ICLR. Artificial Intelligence, San Juan, Puerto Rico (May 2016)
5. Zoph, B., et al.: Learning transferable architectures for scalable image recognition. In Proc. Int. CVPR, Salt Lake City, Utah, pp. 8697–8710 (June 2018)
6. Pham, H., et al.: Efficient neural architecture search via parameter sharing. In Proc. Int. ICML, Stockholm, Sweden (July 2018)
7. Baker, B., et al.: Designing neural network architectures using reinforcement learning. In Proc. Int. ICLR, San Juan, Puerto Rico (May 2016)
8. Real, E., et al.: Large-scale evolution of image classifiers. In Proc. Int. ICML, Sydney, NSW, Australia, pp. 2902–2911 (August 2017)
9. Real, E., et al.: Regularized evolution for image classifier architecture search. In Proc. Int. AAAI. artificial intelligence, Honolulu, Hawaii, USA, pp. 4780–4789 (February 2019)
10. Kundasamy, K., et al.: Neural architecture search with Bayesian optimization and optimal transport. In Proc. Int. NIPS, Palais des Congres de Montreal, Montreal, pp. 2016–2025 (December 2018)
11. Liu, H., Simonyan, K., Yang, Y.: Darts: differentiable architecture search. In Proc. Int. ICLR, Vancouver, BC, Canada (April 2018)
12. Xie, S., et al.: Snas: stochastic neural architecture search. In Proc. Int. ICLR, Vancouver, BC, Canada (April 2018)
13. Wu, B., et al.: Fbnet: hardware-aware efficient converter design via differentiable neural architecture search. In Proc. Int. CVPR, Long Beach, CA, pp. 10734–10742 (June 2019)
14. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: a continuous relaxation of discrete random variables. In Proc. Int. ICLR, San Juan, Puerto Rico (May 2016)
15. Krizhevsky, A., Geoffrey, H.: Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto (June 2009)
16. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In Proc. Int. CVPR, pp. 248–255 (June 2009)
17. Dong, X., Yang, Y.: Searching for a robust neural architecture in four GPU hours. In Proc. Int. CVPR, Long Beach, CA, pp. 1761–1770 (June 2019)
18. Chen, X., et al.: Progressive differentiable architecture search: bridging the depth gap between search and evaluation. In Proc. Int. ICCV, Seoul, Korea (Oct 2019)
19. Chu, X., et al.: Fair darts: eliminating unfair advantages in differentiable architecture search. In Proc. Int. arXiv preprint arXiv:190909656
20. Liang, H., et al.: Darts+: improved differentiable architecture search with early stopping. In Proc. Int. arXiv preprint arXiv:190906035
21. Gordon, A., et al.: Morphnet: fast & simple resource-constrained structure learning of deep networks. In Proc. Int. CVPR, Salt Lake City, Utah, pp. 1586–1595 (June 2018)
22. Szegedy, C., et al.: Going deeper with convolutions. In Proc. Int. CVPR, Boston, MA, pp. 1–9 (June 2015)
23. Liu, H., et al.: Hierarchical representations for efficient architecture search. In Proc. Int. ICLR, Toulon, France (April 2017)
24. Liu, C., et al.: Progressive neural architecture search. In Proc. Int. ECCV, Munich, Germany, pp. 19–34 (September 2018)
25. Cai, H., et al.: Direct neural architecture search on target task and hardware. In Proc. Int. ICLR, Vancouver, BC, Canada (April 2018)
26. Zhang, X., et al.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In Proc. Int. CVPR, Salt Lake City, Utah, pp. 6848–6856 (June 2018)
27. Tan, M., et al.: Mnasnet: platform-aware neural architecture search for mobile. In Proc. Int. CVPR, Long Beach, CA, pp. 2820–2828 (June 2019)

How to cite this article: Zhang, B.F., Zhou, G.Q.: Control the number of skip-connects to improve robustness of the NAS algorithm. IET Comput. Vis. 15 (5), 356–365 (2021). https://doi.org/10.1049/cvi2.12036
APPENDICES

THE UNFAIR ADVANTAGE OF SKIP-CONNECT

The loss optimizes the structure parameters, and the structure parameters are sent to softmax or Gumbel-softmax to generate the structure vector of the sampling sub-network. The competition between operations is generated during softmax or Gumbel-softmax. The external manifestation is that the structure of the sub-network with more skip-connects has less loss. When minimizing the loss, the structural parameters corresponding to the skip-connection are optimized, and the structural parameters affect the sampling of the sub-network, resulting in a greater weight on skip-connection. Therefore, in order to avoid this advantage of skip, we propose to limit the number of skip-connects when sampling, named skip-controller.

WHY SKIP-CONNECT INCREASED

Too many skip-connects are not caused by using argmax pruning after the search is over. So it has nothing to do with whether we use Gumbel-softmax or softmax. Because whether it is the one-hot vector of Gumbel-softmax or the softened one-hot vector of softmax, it is generated according to the structural parameters. Therefore, there is still a competitive relationship between different operations on the same side. Gumbel-softmax only performs one-hot operation on the sampled structure vector of softmax. At the end of the experiment, we also use argmax to determine the structural parameters like darts. The one-hot of Gumbel-softmax is just a strategy to reduce the amount of calculation. Therefore, the number of skip-connects does not have much to do with whether Gumbel-softmax is used.

Finally, the increase in the number of skip-connects is caused by this unfair advantage of skip-connect. Although easy optimization is indeed not a problem, the unfair advantage brought about by easy optimization has become a problem in competition between operations [19]. In other words, after the same training time, a small loss model does not mean that its potential is greater, because this is only in the middle stage of training, and the model has not fully converged. The training time required for a network structure with more parameters to fully converge may be much longer than that for a network structure with fewer parameters. Therefore, it cannot be said that a network structure with fewer parameters has more potential due to its easy convergence. Therefore, this unfair advantage is the reason for the increase in skip-connection, which is also proven by FairDARTS [19]. And this unfair advantage will continue to be amplified in the iterative process, resulting in too many skips in the end. The skip-controller we proposed is to avoid excessive skip-connect during the sampling phase.