# DNAS:A Decoupled Global Neural Architecture Search Method

Kepeng Xu[1] Gang He[12✉]
[1] Xidian University
[2] Kuaishou Technology
kepengxu11@gmail.com

## Abstract

*Neural Architecture Search (NAS) can automatically design model architecture with better performance. Current researchers have searched for local architecture similar to* ***block****, then stacked to construct entire models, or searched the entire model based on a manually designed benchmark module. There is no method to directly search the architecture of the* ***global(entire) model*** *at the* ***operation level****. The purpose of this article is to search the entire model directly in the operation level search space. We analyzed the search space of past methods which searching for local architectures, then a working mode for global model architecture search named* ***CAM*** *is proposed. Proposed* ***CAM*** *decouples the architectural parameters of the entire model which can complete the entire model architecture search with few architecture parameters. In the experiment, the test error 2.68 % in CIFAR-10 is obtained by the proposed method at the* ***global architecture*** *level, which can compare with the stage-of-art local architecture search methods.*

## 1. Introduction

In recent years, deep neural network technology has achieved great success in the tasks of structured data representation such as vision and language. However, there are still many difficulties that prevent the application of this technology, like that require huge amounts of computing cost and expert costs for manually designing models. Neural architecture search is an efficient method to solve these problems.

The purpose of the NAS(neural architecture search) is to search high efficient model architectures through optimization algorithms in a given search space. Early NAS methods [19] used intelligent heuristic algorithm and reinforcement learning to complete the search for the optimal architecture unit. Recently, a large number of gradient-based one-shot methods [5,6,15,16] have appeared, which greatly

✉ Corresponding author

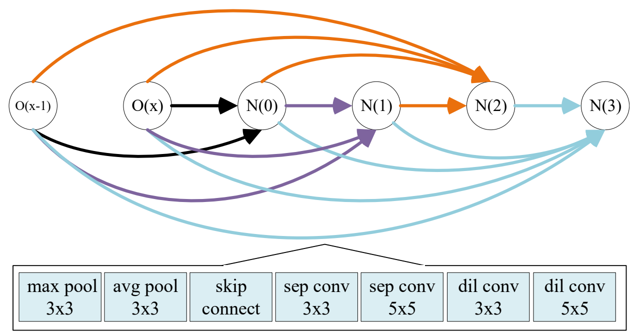


Figure 1. DARTS search the architecture of single cell and finally stack many cells to form the final model. The figure is the architecture of a cell. Each cell takes the output of the previous two cells($O(x-1), O(x)$) as input, there are four nodes($N(0,1,2,3)$) in the Cell, each node is connected to the previous node, there are 14 connections, each connection has 7 candidate operations, so the search space size is $14 \times 7$.

reduce the search time and the cost of computing, making the technology has a certain application value. Current NAS methods search local architecture (blocks) and then continuously stack the searched blocks to construct the entire model architecture or using the artificially designed module as the basic search unit to search the global neural network model. The above method without searching the entire model architecture directly on the basis of operations, so have some obvious limitations:

- Block search methods [6] are a trade-off made by reducing the size of the search space to reduce the difficulty of the search this mode greatly reduces the architectural diversity of the model.
- In the one-shot methods like DARTS [6] search process, construct a proxy model that use a small number of layers (such as 8) to find the optimal block and then stack block 20 times construct the final model. So there has a performance gap between proxy model and the final model.

Current local block search methods and the methods based on manual design module can get outstanding perfor-

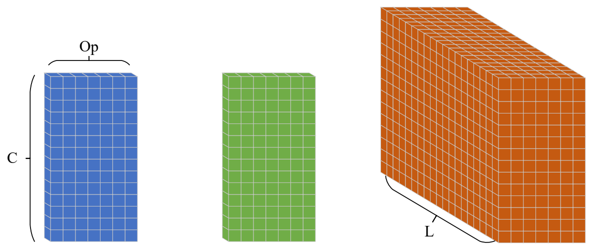


Figure 2. In figure, part 1 and 2 are the parameter matrices of two different types of Block (Normal and Reduce) defined in the DARTS, and part 3 represents the global model architecture searched with the DARTS series methods. The architecture representation parameters required to search the model architecture in the full space will increase significantly.

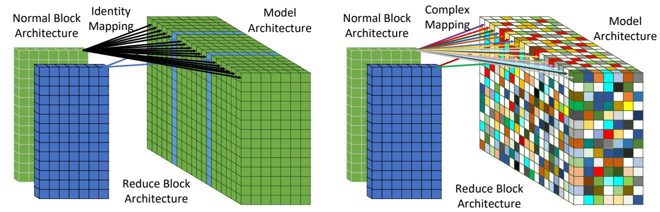


Figure 3. The classic operation-based method simplifies the model search into two types of block structure search and then stacks these two blocks to form the final model, which is called **IAM** here, as shown in part A. Proposed **CAM** mode allows the extension to generate a more complex model architecture, as shown in colored lines of part B.

mance, but the potential performance of global search methods which based operation can not be ignored. In this paper, our core purpose is to **directly search the entire model architecture**. A method that searching the entire model architecture from the global search space is proposed. NAS methods can be divided into three categories according to search algorithms, based on RL,based on evolutionary algorithm, and based on gradient. The proposed method uses gradient to search, which can greatly reduce the search time.

Gradient-based methods relax the discrete combinatorial optimization problem as a differentiable weight optimization problem. As can be seen in Fig .1, the first gradient-based method DARTS [6], the entire model is composed of two types blocks (normal and reduced),block is defined as a DAG(directed acyclic graph). The search space size of each block is $C \times Op$, $C$ refers to the number of connections in each block, and $Op$ refers to the size of the candidate operation set in each connection. In the DARTS [6] series methods, $C = 14$ and $Op = 7$. As mentioned in above 1, the existing method simply stacks block to construct the model, which makes the structure of each layer of the model the same, in fact the function of each layer of the neural network is not same, so the architecture obtained in this method not optimal.

The method [3] directly searching the entire neural network architecture generally uses artificial design modules such as MobileNetV2 [11] as the basic constituent unit. With the artificially designed block structure as the basic unit, this method performs very well. The search problem of NAS method can be regarded as a multi-dimensional function optimization problem. Compared with directly searching the entire neural network architecture, the search method with the block structure as the target is simpler, but the search space is greatly limited (a trade off between complexity and speed).

We call the above method of searching for the Block architecture and stack to the entire model as **IAM(Identity Architecture Mapping)** here(as the Fig.3 shows). Reduces the complexity of the final model structure and uses proxy-model, in exchange for rapid convergence of the search process. The goal of this paper is to search for the global neural network model more efficiently. A method called **CAM(Complex Architecture Mapping)** is proposed. Such a method can obtain a diversified neural network model architecture of inter-layer Block without a significant increase in architectural parameters, So as to quickly and effectively search the entire model architecture.

Here, we propose a parameter decoupling search method to specifically describe the **CAM** process and optimize the architecture search in the global architecture parameter space, so as to obtain the hierarchical differentiated neural network model architecture. Through the method of parameter decoupling, the number of architecture parameters is greatly reduced, thus reducing the difficulty of optimization. The experimental results show that in the global search state, the proposed method has a significant improvement in accuracy compared with the benchmark method. The performance of the fine-grained model architecture obtained by the proposed global search method can even compete with the state-of-art block search method.Our contribution is divided into the following two points.

- Analyzing the related methods of searching the block structure based on the proxy-model in the past and propose a mode named **CAM**, which can extend the past methods to the global neural architecture search task.

- The parameter decoupling method is proposed to specifically implement the **CAM** mode, which reduces the number of architecture parameters required for global architecture search, thereby reducing the difficulty of global architecture optimization.

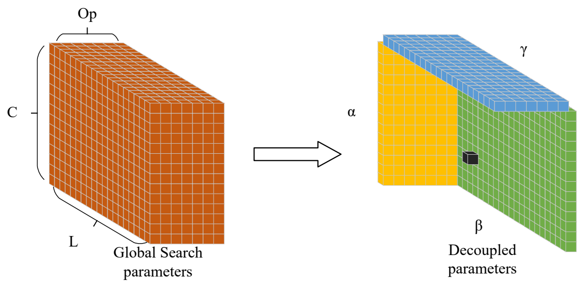


Figure 4. The right side of the diagram is the parameter structure of the global model architecture search under the regular setting, and the left side is visualization of decoupled method.The proposed method decomposes the architecture parameters into three parts $\alpha$, $\beta$, and $\gamma$. The black square in figure refers to $\Psi[k, i, j]$. In this way, the number of parameters that was originally $(L \times C \times Op)$ was reduced to $(L \times C + L \times Op + C \times Op)$, thereby reducing the difficulty of search optimization.

## 2. Related work

NAS has attracted many researcher's attention recently. The NAS methods can be divided into two parts according to the optimized method: **1.** Define NAS's operation selection problem as a combinatorial optimization problem, and use the RL(reinforcement learning) and EA(evolutionary algorithms) to search for the optimal model. **2.** Relaxing the NAS's operation selection problem as the optimization problem of the corresponding weight value of different operations, which can be directly optimized by gradient descent.

NAS-Net [19] is the method based on RL neural architecture search. On Cifar10 and Imagenet, better performance is achieved in terms of parameters and accuracy than manual design models. After that, methods such as [10] using EA to solve search problems. [6] uses gradient optimization to complete the block search of the neural network, reducing the computing cost of the NAS by three orders of magnitude. Later, it appeared that [16] reduced the memory consumption during the search process through 'partial channel connection', [2] reduced the memory by selecting active edge optimization, and [1] used Sigmoid instead of Softmax to enhance the search stability. [4] proposes an early stopping mechanism get stable convergence result. Ensemble Gumbel-Softmax improves the optimization quality of the parameter matrix from the perspective of gradient estimation. NASIB [13] proposed a method that can complete the search under more candidate operations. BANANAS [14] proposed a Bayesian neural structure search method, which predicts the final performance of the selected architecture early. The above methods accelerates the convergence process of the stable search method from the perspectives of memory and calculation to obtain a better result. Other methods such as [3], using [12] or [8] as the basic block to search the entire neural network model. The proposed method search the entire neural network architecture at the operational level.

## 3. Proposed method

### 3.1. Definition of search space

The proposed method aims to search for model architecture in the global search space base on the operations set level. So as to compare with methods such as DARTS [6], the proposed method has the same search space at the block level, the set of candidate operations for each connection is the same, and the number of nodes in the different block is the same.

At first, we explain DARTS's [6] method. In the DARTS, the NAS problem is simplified to search two types of blocks (normal and reduce, in Fig .2). In order to reduce the computation and memory requirements, this method stack two simple block structures $L_p$ times to construct a proxy-model in the search phase (the number of blocks in the search proxy-model is defined as $L_p$). The NAS problem can be defined as a bi-level optimization problem, and then the entire optimization task is completed by alternately optimizing architecture parameters and model parameters. After searching finished, two blocks obtained are stacked $L$ times to become the final model. The architecture of the block has been illustrated in Fig .1. If searching the entire model architecture directly in DARTS, then the entire architectural parameter number will become $(L \times C \times Op)$ as shown in Fig .2. the number of layers, connections in a single block, candidate operations in each connection are defined as $L$,$C$,$Op$. As can be seen from Fig .2, when searching the entire neural network model from the operation level, the DARTS series methods require a large number of architectural parameters, which leads to an increase in the dimension of the independent variable and make the optimization process of multi-dimensional functions very difficult. This is one of the main difficulties of searching neural network model architectures in the global search space.

The method proposed decomposes the model into a stack of $L$ blocks. Unlike the previous method, the proposed method allows each block architecture to be different. Each block can be regarded as a DAG(Directed Acyclic Graph) which have four intermediate nodes $N_i, i \in [1 \cdots 4]$, It receives the output of the previous two blocks' output($O(x)$ and $O(x-1)$) as input. Each node $N_i$ uses the previous node $N_j$ $(j < i)$ as an input then constitutes a connection.Each node $N_i$ in the block uses the previous node $N_j$ $(j < i)$ as a precursor node to build a connection.

Next, the mathematical definition of the connection between any two nodes in the block is given in Formula (1). The set of operations,the architecture weight of the $j$th operation in the $i$th connection in the $k$th block,the activation function are defined as $O$,$\psi(k, i, j)$ and $\sigma$.

$$X = \sum_{j=0}^{Op} \left( \frac{\sigma^{\psi(k,i,j)}}{\sum_{j=0}^{Op} \sigma^{\psi(k,i,j)}} \cdot O^j(x) \right) \tag{1}$$

### 3.2. CAM Mode

In this section, we will introduce the relationship between global neural structure search and block search. At present, the widely used proxy-model and simplify the architectural diversity of the final model through **IAM**(Identity Architecture Mapping). The model obtained by these methods has a small parameter space and low optimization difficulty so that the model with higher performance and lower complexity can be obtained in a limited time.

For the global neural architecture, the required architectural parameter expression form is shown on the right side of the Fig.2. The determination of the entire neural network requires a large number of architecture parameters. Methods such as DARTS use Block as the search target and stack the same Block construct model. Actually, it uses fewer Block structure parameters to represent the global neural architecture with the help of **IAM**. The **IAM** is shown in part A in Fig.3.

Numerous methods have proved that feasible solutions in a broad sense can be obtained in such a simplified search space, which proves that the past methods are very effective. The motivation of this paper is to directly search the entire model architecture base on the operation level. In order to effectively search for the global architecture, this paper designs a **CAM(Complex Architecture Mapping)**(as shown as in part B of Fig.3) instead of the **IAM**(DARTS, etc..). CAM allows us to use a small number of architecture parameters to represent the entire model architecture, thereby reducing the complexity of the search process. In Fig.3 that in the DARTS block blue and black line represents the identity mapping, and the colored lines on the right part represent the proposed **CAM** mode.

We analyze how to expand the architecture parameter space from a single block to entire neural network architecture. Thinking from another perspective, the architecture parameters of a single block essentially refer to the weight of each candidate operation on each connection. The IAM mode simply stacks blocks, which can be understood as copying the architectural parameters of a single block many times. In the global neural architecture parameter representation, we can model the mapping from block architecture parameters to the global architecture with the participation of the 'layer-connection' parameter $\beta$ and the 'layer-operation' parameter $\gamma$. (Using CAM, allowing each layer to be different)

Then, we propose the Formula 2, which define the general form of Block expansion to the global architecture.(This is a formal definition, the specific method used in this article will be introduced in the next section.)

$$\psi = F_1 \odot F_2 \odot \alpha \tag{2}$$

Here, the mapping function of $\beta$ parameter is defined as $F_1$, the mapping process of $\gamma$ parameter is defined as $F_2$, the architecture parameter of a single Block is defined as $\alpha$, and $\odot$ refers to the mapping operation.

In the process of mapping from a single Block architecture to the global neural network architecture, the $\alpha$ parameters are mapped under the joint action of $\beta$ and $\gamma$ to produce the entire neural network model architecture parameters. In the next subsection, the specific implementation process of parameter mapping will be introduced in detail.

### 3.3. Decoupling of architectural weight parameters

As we indicated in Fig.2, using the gradient-based method directly search in the full space. The numerous architecture parameters and high memory consumption will increase the difficulty of optimization.

In order to achieve **CAM** and reduce the size of parameter space, this paper considers the correlation between global architecture parameters in different dimensions, and finally decomposes the architecture parameters into three components,in Fig.4. namely:

• Connection-Operation weight $\boldsymbol{\alpha}$: weight distribution of different operations corresponding to each connection.

• Layer-Connection weight $\boldsymbol{\beta}$: the weight distribution of different connections corresponding to each layer block.

• Layer-Operation weight $\boldsymbol{\gamma}$: the weight distribution of different operations corresponding to each layer block.

The decomposition of the parameter space will be introduced in here. The three dimensions of the parameter space have been defined earlier: $L, C, Op$ refer to the number of layers, the number of candidate connections for a single block, and the number of candidate operations for a single connection. This paper decomposes the huge parameter space into three parameter matrices.

In the global architecture search process, every element of $\psi$ needs to be optimized. We consider the composition of $\psi[k, i, j]$ from the perspective of decoupling, which can be understood as $\alpha[i, j]$ at the kth layer on the map,$\beta[k, i]$ on the jth operation, and $\gamma[k, j]$ on the ith connection, as shown in Fig.4. From this we can see that in the past, the local search method based on the cell is a simplified version of this method, that is, the mapping of $\alpha[i, j]$ on the kth layer is an identity mapping.A formal definition of $\psi$ decoupling is given in Formula (3),the combination operation method of $\alpha$, $\beta$, and $\gamma$ is defined as $\otimes$.

$$\psi = \alpha \otimes \beta \otimes \gamma \tag{3}$$

Next, define the combination of $\alpha$, $\beta$, and $\gamma$ is further defined in Formula(4)

$$\psi[k,i,j] = \sigma_\alpha(\alpha[i,j]) \cdot \sigma_\beta(\beta[k,i]) \cdot \sigma_\gamma(\gamma[k,j]) \quad (4)$$

In Formula(4), $k \in [0,L), i \in [0,C), j \in [0,Op).\psi[i,j,k]$ refers to the weights of $k$th layer, the $i$th connection, and the $j$th candidate operation, $\sigma$ are activation function. In the search phase, $\sigma_\alpha$, $\sigma_\beta$, and $\sigma_\gamma$ are sigmoid, softmax, and sigmoid, respectively. Under this definition, the final form of the proposed method is defined as Formula (5).

$$X = \sum_{j=0}^{Op} \left(\sigma_\alpha(\alpha[i,j]) \cdot \sigma_\beta(\beta[k,i]) \cdot \sigma_\gamma(\gamma[k,j]) \cdot O^j(x)\right) \quad (5)$$

Formula(5) show the calculation of different operation weights in each connection, so that the weight values can be updated by gradient optimization, and the architecture of the neural network model can be optimized.

### 3.4. Relationship to Prior Work

Previous researchers have focused on how to more efficiently search for block-level model structures. For example, PC-DARTS speeds up the stable search process through partial channel connections and Edge Normalization. FairDARTS uses Sigmoid instead of Softmax and adds a zero-one loss function To increase the competitive advantage of reducing search-training gaps and skips, the method proposed in this paper follow part connections and zero-one loss in order to accelerate the global search speed.

## 4. Experiment and Results

### 4.1. Dataset detail

The number of categories of CIFAR10, CIFAR100, TinyImagenet are 10,100,200. The dataset setting and search space setting follow DARTS.

### 4.2. Results on CIFAR-10

In the search phase, we set the number of channel equal to 16, use the same data augmentation strategy and hyper-parameter setting as the DARTS method(Set skip connection drop rate to 0.2), the search time on single RTX2080Ti is 0.4 day, and then train the obtained model architecture. After comprehensive experiments, the final average error is 2.68%(Detailed results are shown in Table.1). The visual diagram of the model architecture obtained by the search is shown in Fig.5 .

The Fig.5 shows the architecture visualization results in detail. It can be seen that Separable convolutions, Dilated convolutions, and other operations with feature encoding capabilities appear more frequently in the shallow layer of the model. None-Weights operations such as skip-connection occur more frequently at deeper levels of the model. This is consistent with the researchers' past cognition that the lower-level operations of the neural network model are responsible for feature extraction. The skip-connection operation can destroy the singularity of the feature layer of the model so that the gradient backpropagation can effectively update the model parameters in the lower layer.

Table 1. Comparison with state-of-the-art network architectures on CIFAR-10. † represent model architecture searched from global search space

| Architecture | Test Err. (%) | Params (M) | Search Cost (GPU-days) | Search Level |
|---|---|---|---|---|
| NASNet-A + cutout [19] | 2.65 | 3.3 | 1800 | Block |
| PNAS [5] | 3.41 | 3.2 | 225 | Block |
| ENAS + cutout [9] | 2.89 | 4.6 | 0.5 | Block |
| DARTS (1st order) + cutout [7] | 3.00 | 3.3 | 0.4 | Block |
| DARTS (2nd order) + cutout [7] | 2.76 | 3.3 | 1 | Block |
| SNAS (moderate) + cutout [15] | 2.85 | 2.8 | 1.5 | Block |
| BayesNAS + cutout [18] | 2.81 | 3.4 | 0.2 | Block |
| PC-DARTS + cutout [16] | 2.57 | 3.6 | 0.1 | Block |
| FairDARTS + cutout [1] | 2.59 | 3.27 | - | Block |
| BANANAS + cutout [14] | 2.64 | - | 11.8 | Block |
| NASIB + cutout [13] | 3.57 | 6.71 | 1.5 | Block |
| NASP + cutout [17] | 2.83 | 3.3 | 0.1 | Block |
| PC-DARTS + cutout [16] | 3.69 | 3.44 | 0.4 | **Global**† |
| DNAS + cutout | **2.68** | **3.35** | 0.4 | **Global**† |

### 4.3. Results on CIFAR-100

During the CIFAR-100 model search process, we set the number of blocks, channels, and batch size equal to 14, 16, 96. As can be seen from Table 2, the proposed method has achieved a competitive performance in global architecture search.

Table 2. Comparison with state-of-the-art network architectures on CIFAR-100.

| Architecture | Test Err. (%) | Params (M) | Search Cost (GPU-days) | Search Level |
|---|---|---|---|---|
| DenseNet-BC | 19.64 | 15.3 | - | - |
| VGG-16 | 27.07 | 34.0 | - | - |
| NASNet [19] | 22.71 | 5.7 | 1800 | Block |
| PC-DARTS + cutout [16] | 17.64 | 4.36 | 0.1 | Block |
| PC-DARTS + cutout | 19.8 | 4.27 | - | **Global**† |
| DNAS | **17.47** | 4.60 | 0.6 | **Global**† |

### 4.4. Transferring to Tiny-Imagenet

In order to prove the effectiveness of the obtained architecture, we transfer the obtained model to Tiny-Imagenet. In the training process, we adopted random rotation and flipping augmentation strategies. The detailed experimental settings follow the DARTS. From Table 3, the model searched on CIFAR-10 performs well on the TinyImagenet

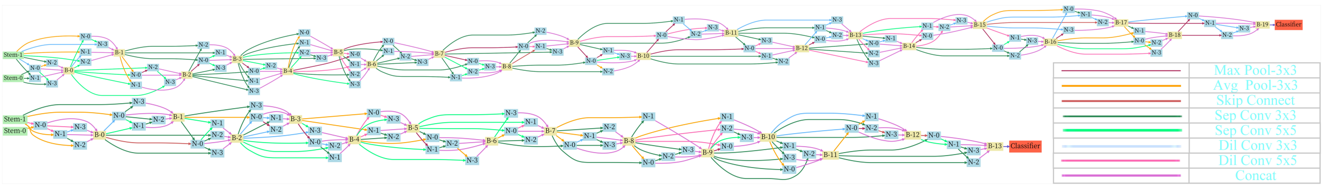


Figure 5. The model architecture diagram searched on CIFAR-10/CIFAR-100. The lines in different colors represent different types of operation connections in this diagram, the upper part is the model architecture searched on CIFAR-10, and the lower part is the model architecture searched on CIFAR-100.

Table 3. Comparison with state-of-the-art network architectures on TinyImagenet.

| **Architecture** | **Test Err. (%)** | **Params (M)** | **Search Cost (GPU-days)** | **Search Level** |
|---|---|---|---|---|
| NASNet-A [19] | 58.99 | 4.8 | 1800 | Block |
| ENAS [9] | 57.81 | 4.6 | 0.5 | Block |
| DARTS [6] | 57.42 | 3.9 | 4 | Block |
| SNAS [15] | 57.81 | 3.3 | 1.5 | Block |
| NASP [17] | 58.32 | 8.9 | 0.2 | Block |
| DNAS(CIFAR-10) | **65.30** | 5.9 | 0.4 | **Global**† |

dataset, which proves that the architecture obtained by the proposed method has strong transformation ability.

## 5. Conclusions

In the past, architecture search methods searched the local architecture, and then stacked the local architecture to form the entire model. Unlike these methods, the core goal of this paper is to search for model architectures at the operation level in the global space. The working mode of local architecture search is analyzed and a model construction mode of **CAM** is proposed, then a method of **decoupling architecture parameters** to reduce redundant architecture parameters is proposed, thereby improving the efficiency of the global architecture search. From the experimental results in CIFAR-10, CIFAR-100 and TinyImagenet, the method proposed in this paper has achieved performance comparable to the most advanced local architecture search methods on the basis of global architecture search. In a word, the competitiveness of the **global neural architecture search** method is improved.

## References


[1] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair darts: Eliminating unfair advantages in differentiable architecture search, 2019. 3, 5

[2] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours, 2019. 3

[3] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. *CoRR*, abs/1904.00420, 2019. 2, 3

[4] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping, 2019. 3

[5] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, 2018. 1, 5

[6] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1, 2, 3, 6

[7] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 5

[8] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 3

[9] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018. 5, 6

[10] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. Large-scale evolution of image classifiers, 2017. 3

[11] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. 2

[12] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. 3

[13] Abhishek Singh, Anubhav Garg, Jinan Zhou, Shiv Ram Dubey, and Debo Dutta. Nasib: Neural architecture search within budget. 10 2019. 3, 5

[14] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. 10 2019. 3, 5

[15] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: Stochastic neural architecture search. In *ICLR*, 2019. 1, 5, 6

[16] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient differentiable architecture search, 2019. 1, 3, 5

[17] Quanming Yao, Ju Xu, Wei-Wei Tu, and Zhanxing Zhu. Differentiable neural architecture search via proximal iterations. *CoRR*, abs/1905.13577, 2019. 5, 6

[18] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. BayesNAS: A Bayesian approach for neural architecture search. In *ICML*, 2019. 5

[19] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 1, 3, 5, 6