FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search

Xiangxiang Chu, Bo Zhang, Ruijun Xu, Jixiang Li
Xiaomi AI Lab
3 Jul 2019
arXiv:1907.01845

Reporter：黃顯堯

Outline

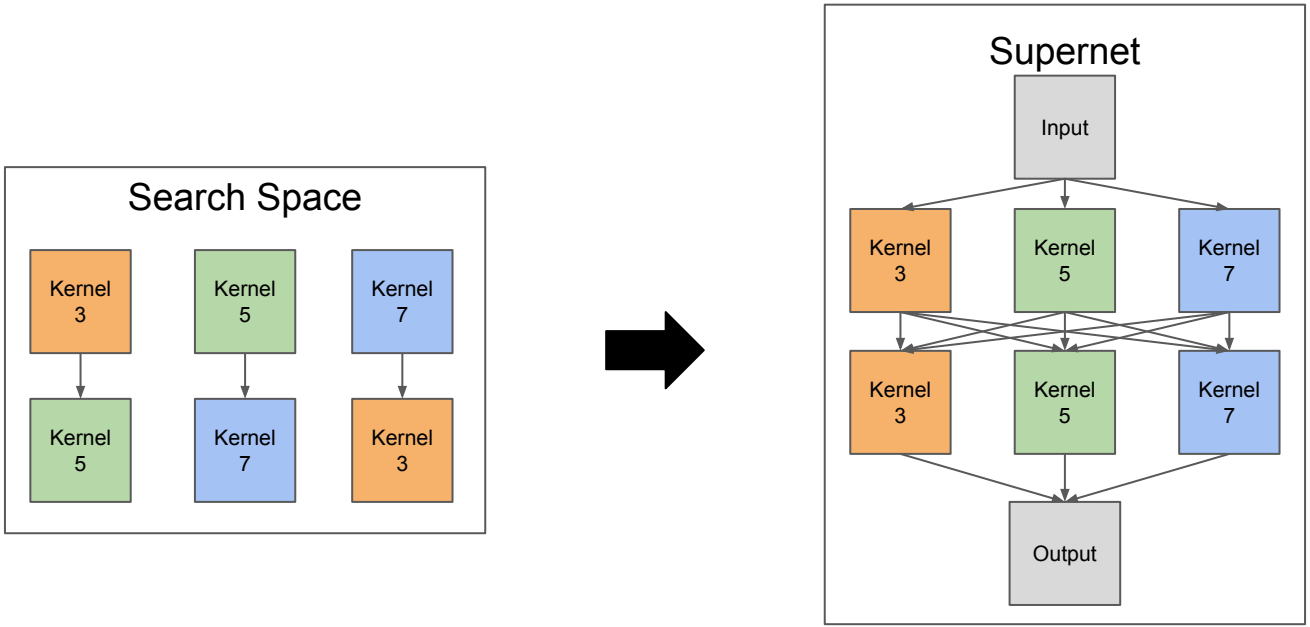
- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Introduction

- It is difficult to design a good neural network manually
  - Requires much domain knowledge
  - Model training and tuning is time consuming
- Neural Architecture Search(NAS)
  - A research automate design of deep learning models

Introduction

- Weight sharing NAS
  - All involve training a supernet that incorporates many candidate subnetworks



Introduction

- Weight sharing NAS
  - Roughly classified into two categories
    - Couple searching and training within **one stage**
    - Decouple them into **two stages**, where the trained supernet is treated as an evaluator for final searching

Introduction

- There are several fundamental issues about Weight sharing NAS
  - A large gap between the accuracies of supernet and the accuracy by stand-alone training from scratch
  - How to build a good evaluator that neither overestimates nor underestimates subnetworks?
  - Why does the weight-sharing mechanism work, if under some conditions?

Introduction

- In this paper, attempt to answer the above three questions
- Present Fair Neural Architecture Search (FairNAS)
  - Fairly train the supernet as an evaluator
  - Turn narrows the accuracy gap

Introduction

- The contribution of Fair Neural Architecture Search (FairNAS)
  - Prove it is due to ***unfair bias*** that the supernet misjudges submodels' performance.

Introduction

- The contribution of Fair Neural Architecture Search (FairNAS)
  - Prove it is due to ***unfair bias*** that the supernet misjudges submodels' performance.
  - Propose two levels of fairness constraints:
    - Expectation Fairness (EF)
    - Strict Fairness (SF)
    - They are enforced to alleviate supernet bias and to boost evaluation capacity

Introduction

- The contribution of Fair Neural Architecture Search (FairNAS)
  - Prove it is due to ***unfair bias*** that the supernet misjudges submodels' performance.
  - Propose two levels of fairness constraints:
    - Expectation Fairness (EF)
    - Strict Fairness (SF)
    - They are enforced to alleviate supernet bias and to boost evaluation capacity
  - Unveil the root cause of the validity of single-path supernet training under our fairness perspective
    - Different choice blocks of the same layer learn similar feature maps on the corresponding channel

Introduction

- The contribution of Fair Neural Architecture Search (FairNAS)
  - Prove it is due to ***unfair bias*** that the supernet misjudges submodels' performance.
  - Propose two levels of fairness constraints:
    - Expectation Fairness (EF)
    - Strict Fairness (SF)
    - They are enforced to alleviate supernet bias and to boost evaluation capacity
  - Unveil the root cause of the validity of single-path supernet training under our fairness perspective
    - Different choice blocks of the same layer learn similar feature maps on the corresponding channel
  - Incorporate fair supernet with an EA-based multi-objective searching framework
    - Obtain three state-of-the-art networks achieved 75.34% top-1 validation accuracy.

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Fairness Taxonomy of Weight-sharing NAS

- To remove the training difference between a supernet and its submodels
  - We scheme an equality principle on training modality
- Equality Principle
  - ***Training a supernet satisfies the Equality Principle if and only if it is in the same way how a submodel is trained.***
  - A supernet that consists of L layers, each with m choice blocks
  - The weights are updated for n times in total.
  - The training process is $P(m, n, L)$

Fairness Taxonomy of Weight-sharing NAS

- The Expectation Fairness
  - Guarantee all choices blocks have equal expectations after n steps
  - For $P(m, n, L)$, $E(Y_{l_1}) = E(Y_{l_2}) = \ldots = E(Y_{l_m})$
    - Let Y_{l_i} be the number of times that the outcome l_i is updated over n trials

Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - Check previous Single-Path which uses uniform sampling
  - Selecting a block from layer l is subject to the categorical distribution
  - Each basic event occurs with an equal probability $p(X = l_i) = \frac{1}{m}$
  - For n steps, the expectation and variance of Y_{l_i} can be written as :

$$E(Y_{l_i}) = n * p_{l_i} = n/m$$

$$Var(Y_{l_i}) = n * p_{l_i}(1 - p_{l_i}) = \frac{n(m-1)}{m^2}$$

Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - Uniform sampling ***meets Expectation Fairness***
  - However, Expectation Fairness is not enough
  - For example,
    - Randomly sample each model and keep it training for k times, then switch to another

Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - Even with uniform sampling where $k = 1$, there is a latent ***ordering issue***
  - For example,
    - For a sequence of choices (M_1, M_2, M_3), it implies an inherent training order
      - It implies an inherent training order $M_1 \rightarrow M_2 \rightarrow M_3$

Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - For $P(m, n, L)$
    - If we adopt uniform sampling, as n goes infinite, ***it is impossible for m choices to be sampled for an exactly equal number of times***

Lemma 1. *Regarding* $P(m, n, L)$, $\forall\ n \in \{x : x\%m = 0, x \in N_+\}$, $\lim_{n \to +\infty} p(Y_{l1} = Y_{l2} = ... = Y_{lm}) = 0.$

Proof. Let $f(m, n) = p(Y_{l1} = Y_{l2} = ... = Y_{lm})$.

$$f(m,n) = C_n^{\frac{n}{m}} C_{\frac{n(m-1)}{m}}^{\frac{n}{m}} ... C_{\frac{n}{m}}^{\frac{n}{m}} \frac{1}{m^n} = \frac{n!}{(\frac{n}{m}!)^m} \frac{1}{m^n} \quad \text{(a)}$$

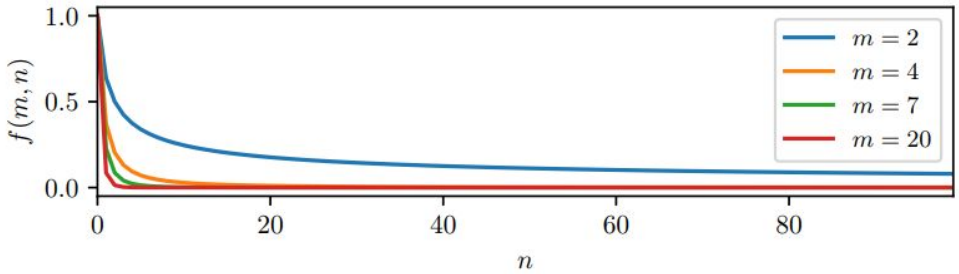
Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - For $P(m, n, L)$
    - If we adopt uniform sampling, as n goes infinite, it is impossible for m choices to be sampled for an exactly equal number of times

$$
\begin{aligned}
\lim_{n \to +\infty} f(m, n) &= \lim_{n \to +\infty} \frac{n!}{(\frac{n}{m}!)^m \times m^n} \\
&= \lim_{n \to +\infty} \frac{\sqrt{2\pi n}(\frac{n}{e})^n}{\sqrt{2\pi \frac{n}{m}}^m (\frac{n}{e})^n} \\
&= \lim_{n \to +\infty} \frac{\sqrt{m}}{\frac{2\pi n}{m}^{\frac{m-1}{2}}} \\
&= 0
\end{aligned}
$$

Fairness Taxonomy of Weight-sharing NAS

- Is Uniform Sampling Fair Enough?
  - For $P(m, n, L)$
    - If we adopt uniform sampling, as n goes infinite, it is impossible for m choices to be sampled for an exactly equal number of times



Fairness Taxonomy of Weight-sharing NAS

- A Meticulous Overhaul: Strict Fairness
  - ***Ensures the parameter of every choice block be updated the same amount of times at any stage***
  - Called strict fairness
    - $P(Y_{l_1} = Y_{l_2} = \ldots = Y_{l_m}) = 1$ holds at any time
    - It seems subtle but it will be later proved to be crucial

Definition 3. ***Strict Fairness.*** *Regarding P(m, n, L),* $\forall\ n \in \{x : x\%m = 0, x \in N_+\}$, $Y_{l_1} = Y_{l_2} = \ldots = Y_{l_m}$ *holds.*

Fairness Taxonomy of Weight-sharing NAS

- A Fairness Taxonomy

NAS Methods	M	C_t	C_s	EF	SF
SMASH [3]	SN	-	-	✗	✗
One-Shot [2]	SN	$16^{\ddagger}$	3.3	✗	✗
DARTS [15]	SN	$4^{\dagger}$	0	✗	✗
FBNet [33]	SP	20	0	✗	✗
ProxylessNAS [4]	TP	15	0	✗	✗
Single Path One-Shot [7]	SP	12	<1	✓	✗
Single-Path NAS [27]	SP	$1.25^{\ddagger}$	0	✓	✗
FairNAS (Ours)	SP	10	2	✓	✓

Table 1. Comparison of state-of-the-art weight-sharing NAS methods as per cost and fairness basis. M: Memory cost at a single path (SP), two paths (TP), and a whole supernet (SN). C_t, C_s: train and search cost measured in GPU days. EF: Expectation Fairness, SF: Strict Fairness. †: searched on CIFAR-10, ‡: TPU.

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. SMASH: One-Shot Model Architecture Search through HyperNetworks. In International Conference on Learning Representations, 2018. 1, 2, 4, 7
[2] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and Simplifying One-Shot Architecture Search. In International Conference on Machine Learning, pages 549–558, 2018. 1, 2, 3, 4, 6, 7
[15] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. In International Conference on Learning Representations, 2019. 1, 2, 3, 4, 5, 6
[33] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 1, 4, 6
[4] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In International Conference on Learning Representations, 2019. 1, 3, 4, 5, 6
[7] 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. arXiv preprint. arXiv:1904.00420, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 11
[27] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-Path NAS: Designing HardwareEfficient ConvNets in less than 4 Hours. arXiv preprint. arXiv:1904.02877, 2019. 1, 4, 6

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Fair Neural Architecture Search

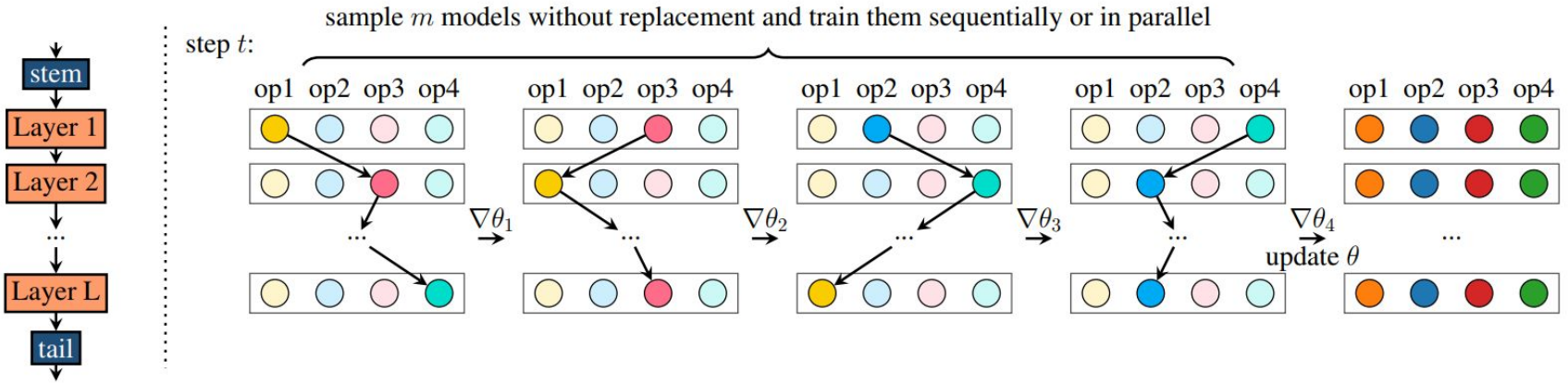
- Following One-shot method, use the supernet to evaluate the performance of a multitude of models
  - Divide our fair neural architecture search into two stages:
    - Training the supernet
    - Searching for competitive models.

Fair Neural Architecture Search

- Stage One: Supernet with Strict Fairness
  - ***Uniform sampling without replacement and sample m models at step t***
  - Do not perform back-propagation and update parameters immediately for each model
    - Define one ***supernet step*** as several backpropagation operations (BP) accompanied by ***a single parameter update***
    - Gradients are then accumulated across the selected m models
      - Supernet's parameters get updated only when all m BPs are done.

Fair Neural Architecture Search

- Stage One: Supernet with Strict Fairness
  - Define one supernet step as several backpropagation operations (BP) accompanied by ***a single parameter update***



Fair Neural Architecture Search

- Strict Fairness Analysis
  - Each choice block is activated only once during a parameter update step
    - $Y'_{l_i} = Y'_{l_2} = \ldots = Y'_{l_m} = n/m$ holds
  - Assures fairness at every step

$$E(Y'_{l_i}) = n/m$$
$$Var(Y'_{l_i}) = 0$$

  - The obvious difference lies in the variance
    - For the previous approach with uniform sampling
      - The variance spread along with n, which gradually increases the bias.

Fair Neural Architecture Search

- Stage Two: Supernet as an Evaluator
  - Utilize so-trained supernet to accurately evaluate each submodel’s performance
  - Integrate MoreMNAS [5] by replacing its incomplete-training evaluator with fairly trained supernet.
    - Speed-up in terms of GPU days by two orders of magnitudes
  - Use Proximal Policy Optimization as the default reinforcing algorithm

[5] Xiangxiang Chu, Bo Zhang, Ruijun Xu, and Hailong Ma. ulti-Objective Reinforced Evolution in Mobile Neural Architecture Search. arXiv preprint. arXiv:1901.01074, 2019. 5, 11

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Experiments

- Search Space
  - Adopt two types of search spaces
    - One for ranking analysis
      - Designed based on MobileNetV2’s inverted bottleneck blocks
      - The same amount of layers with standard MobileNetV2
      - Kernels size in (3, 5, 7)
      - Expansion rates of (3, 6)
    - The other for comparison with other NAS methods
      - Use the same search space of 19 layers as ProxylessNAS[4]

[4] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In International Conference on Learning Representations, 2019. 1, 3, 4, 5, 6

Experiments

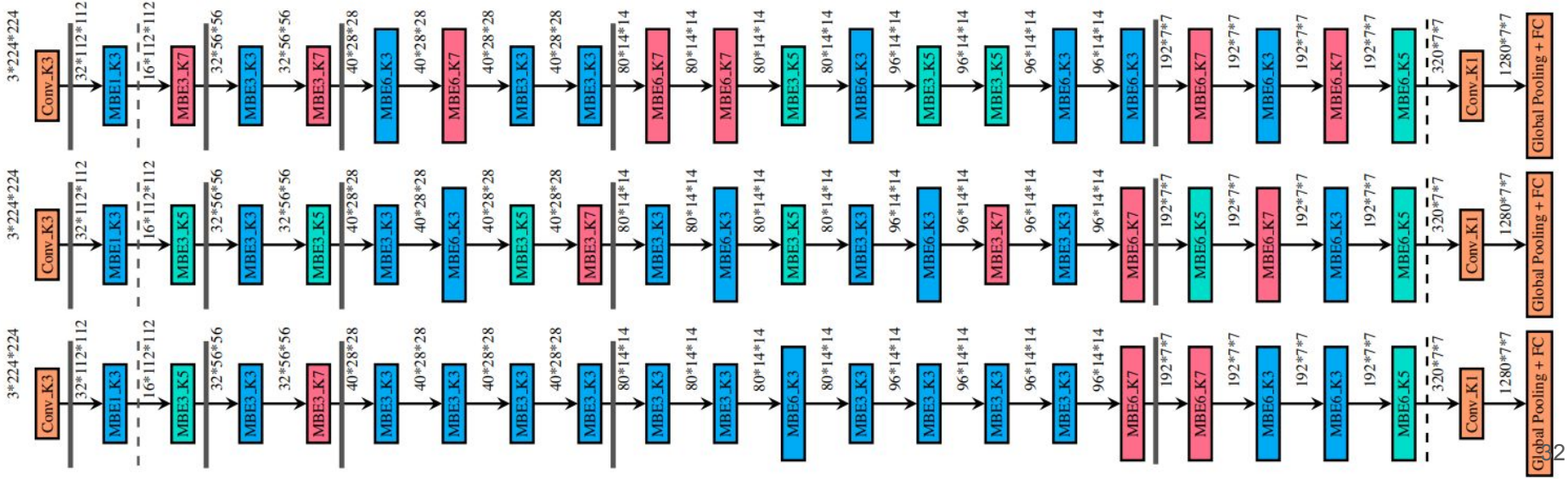
- Comparisons with State-of-the-art Methods
  - Search Cost: 12 GPU days

Methods	×+ (M)	P (M)	Lat (ms)	μ_1 (%)	μ_5 (%)
MobileNetV2 [23]	300	3.4	78	72.0	91.0
NASNet-A [38]	564	5.3	183	74.0	91.6
MnasNet [30]	317	4.2	76	74.0	91.8
MnasNet-92 [30]	388	3.9	92	74.79	92.1
DARTS [15]	574	4.7	-	73.3	91.3
FBNet-B [33]	295	4.5	-	74.1	-
Proxyless-R [4]	320†	4.0	78	74.6	92.2
Proxyless GPU [4]	465†	7.1	124	75.1	-
Single Path One-Shot [7]	323	3.5	-	74.4	91.0
Single-Path NAS [27]	365	4.3	79	74.96	92.2
FairNAS-A (Ours)	388	4.6	104	**75.34**	**92.4**
FairNAS-B (Ours)	345	4.5	90	75.10	92.3
FairNAS-C (Ours)	321	4.4	83	74.69	92.1

[23] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018
[38] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, 2018. 1, 6
[30] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V Le. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 1, 6, 7
[15] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. In International Conference on Learning Representations, 2019. 1, 2, 3, 4, 5, 6
[33] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 1, 4, 6
[4] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In International Conference on Learning Representations, 2019. 1, 3, 4, 5, 6
[7] 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. arXiv preprint. arXiv:1904.00420, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 11
[27] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-Path NAS: Designing HardwareEfficient ConvNets in less than 4 Hours. arXiv preprint. arXiv:1904.02877, 2019. 1, 4, 6

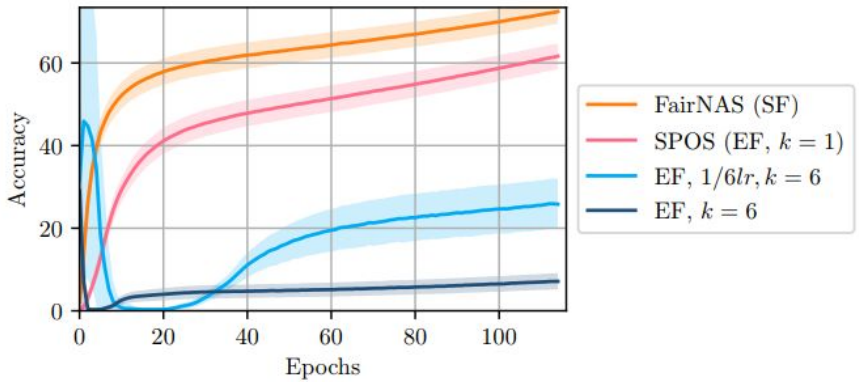
Experiments

- Comparisons with State-of-the-art Methods
  - Three models seem to agree with high expansion rates and large kernels at the tail end
  - Adopts lots of blocks with a small kernel 3 × 3



Experiments - Ablation Study

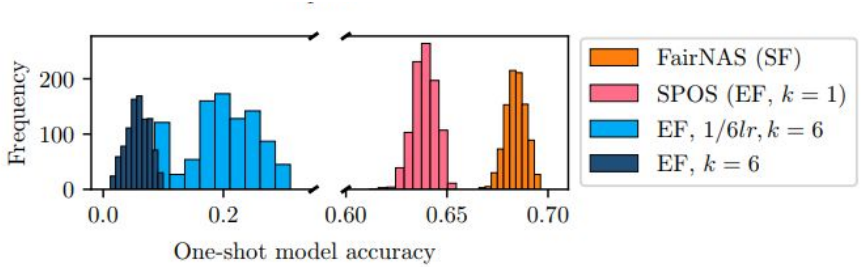
- Expectation Fairness vs. Strict Fairness
  - EF $k = 6$, uniformly sampling one path and k times, followed by parameter update
  - EF $k = 6, 1/6lr$, same as the first one except that the learning rate is scaled by $\frac{1}{k}$
  - EF $k = 1$, as Single-Path One-Shot [7]



[7] 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. arXiv preprint. arXiv:1904.00420, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 11

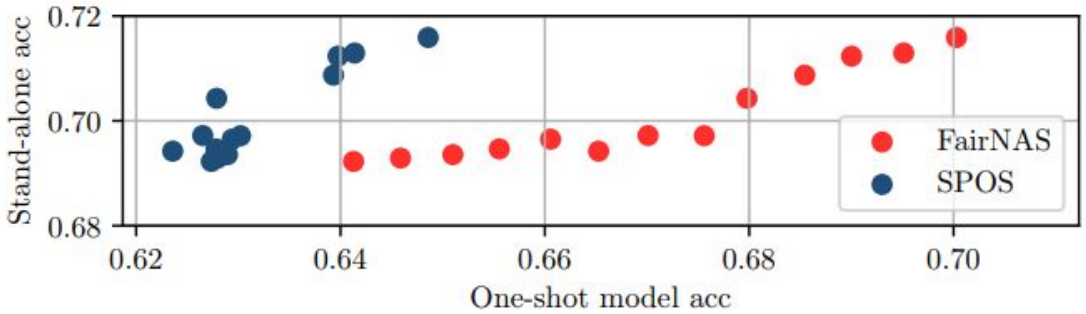
Experiments - Ablation Study

- Expectation Fairness vs. Strict Fairness
  - The Supernet Accuracy gap



Experiments - Ablation Study

- Relation Analysis
  - FairNAS has higher relation



Experiments - Ablation Study

- Kendall Rank Analysis
  - Kendall Tau (τ) measures the ranking relation between one-shot models and fully trained ones
    - The range of τ is from -1 to 1, meaning the rankings are totally reversed or completely preserved
      - whereas 0 means there is no correlation at all

Methods	Fairness	τ
One-Shot [2]†	None	0
Uniform ($k = 6$, baseline)	EF	0.4871
Uniform ($k = 1$, $1/6lr$)	EF	0.4871
SPOS [7] ($k = 1$)†	EF	0.6153
FairNAS‡	SF	**0.9487**

[2] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and Simplifying One-Shot Architecture Search. In International Conference on Machine Learning, pages 549–558, 2018. 1, 2, 3, 4, 6, 7
[7] 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. arXiv preprint. arXiv:1904.00420, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 11

Experiments - Ablation Study

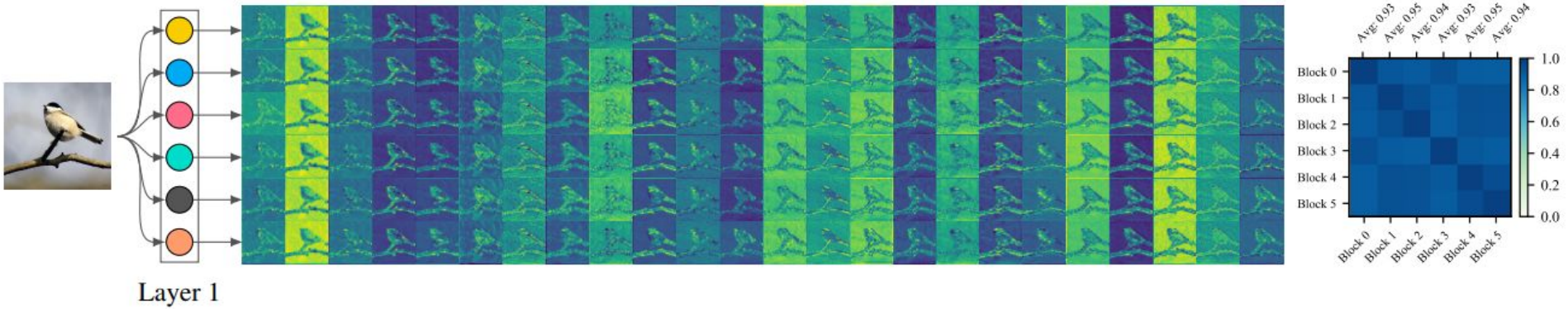
- Is recalibration for batch normalization a must?
  - The statistics for Batch Normalization must be recalibrated in other approaches to boost their model ranking performances
  - Compare the ranking of FairNAS with and without this extra process
    - Both have the same Kendall Tau (τ)
  - Evaluate the sampled architectures on the fly without the time consuming recalibration

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

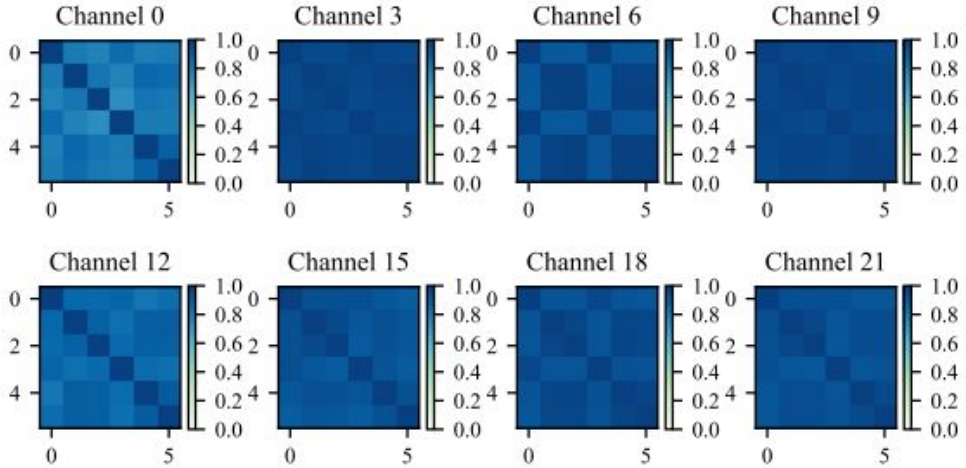
Why Does Single Path Training Work?

- The choice blocks of the first layer yield similar feature maps on the same channel



Why Does Single Path Training Work?

- The choice blocks of the first layer yield similar feature maps on the same channel



Why Does Single Path Training Work?

- The channel-wise feature maps generated by our supernet come with high similarities.
  - Important characteristic significantly stabilizes the whole training process
  - For layer $l+1$, its input are randomly from choice blocks in previous layer l
    - The random sampling constructs a mechanism mimicking ***feature augmentation***
      - Boost the stable training of the supernet
    - The augmentation doesn't alter the feature distribution too much
      - Therefore, recalibrating batch normalization is no longer required for evaluation

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Conclusion

- Scrutinize neural architecture search with a fairness perspective, especially for weight-sharing approaches
- Prove that unfairness inevitably incurs a severely biased evaluation of one-shot model performance
- Propose Strict Fairness to train supernet
  - Fair supernet can incorporated in any NAS pipeline
- The first to give a theoretical analysis of why single-path training is more beneficial

Outline

- Introduction
- Fairness Taxonomy of Weight-sharing NAS
- Fair Neural Architecture Search
- Why Does Single Path Training Work?
- Experiments
- Progress Report

Progress Report

- Architecture searching is computationally expensive
  - RL-based NAS require about 40K GPU hours
  - Weight Sharing based proposed to reduce search efforts
    - Differentiable NAS
    - One-shot NAS

Progress Report

- Weight Sharing
  - Differentiable NAS
    - Learning architecture distribution by architecture parameter
    - Integrated cross-entropy loss (for image classification) with the hardware constraint (in terms of latency) to form the total loss.
    - Previous Differentiable NAS need 216 GPU hours for a specific hardware constraint
      - N different networks specific to N different constraints, 216×NGPU hours are still needed for search

Progress Report

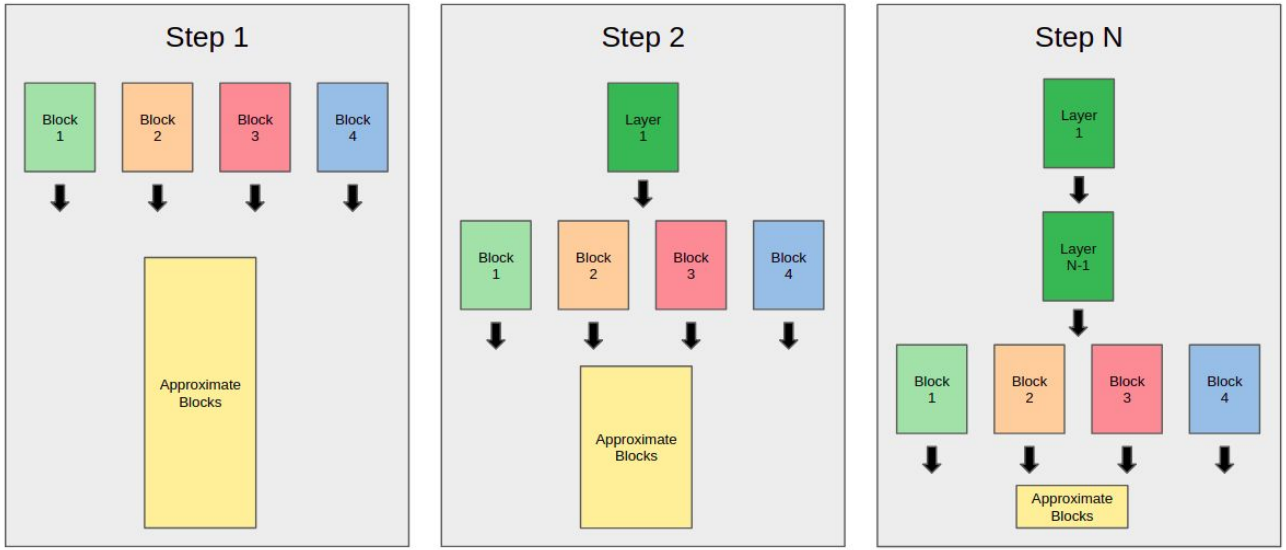
- Weight Sharing
  - One-shot NAS
    - Decouple model training from architecture search
    - Finished training Supernet
      - Sampled different sub-networks from supernet by the accuracy predictor or evolutionary algorithm
    - Evolutionary algorithm
      - Need a lot of validation inference time to execute evolutionary algorithm
    - Accuracy predictor
      - Need 40 GPU hours to training accuracy predictor

Progress Report

- Two problems in previous NAS
  - High cost of supernet training
    - Training a supernet still needs a lot of computation resource and search time.
  - High cost of sub-network specialization
    - No matter DNAS or one-shot NAS,considerable time is needed to specialize sub-networks from the supernet.

Progress Report

- Propose progressive one-shot neural architecture search(PONAS)
  - Combines the advantages of progressive NAS and the one-shot method
  - Search the best layer in the network progressively

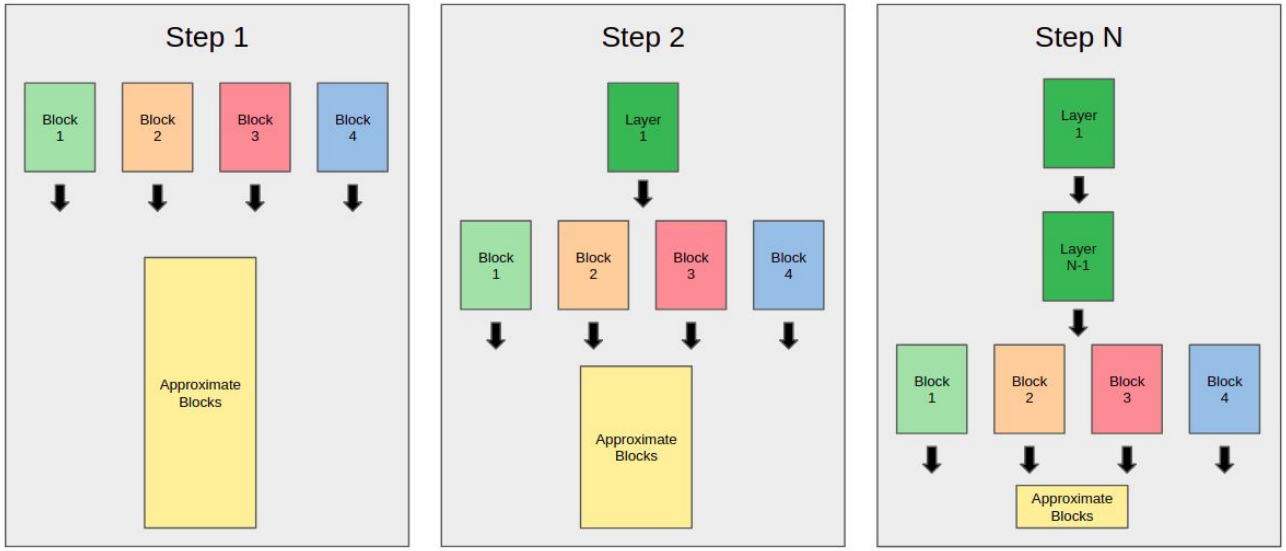


Progress Report

- Progressive NAS
  - Search the best structure of the cell is searched by progressively expanding blocks (operations)
  - Stack multiple best cells to form the final CNN

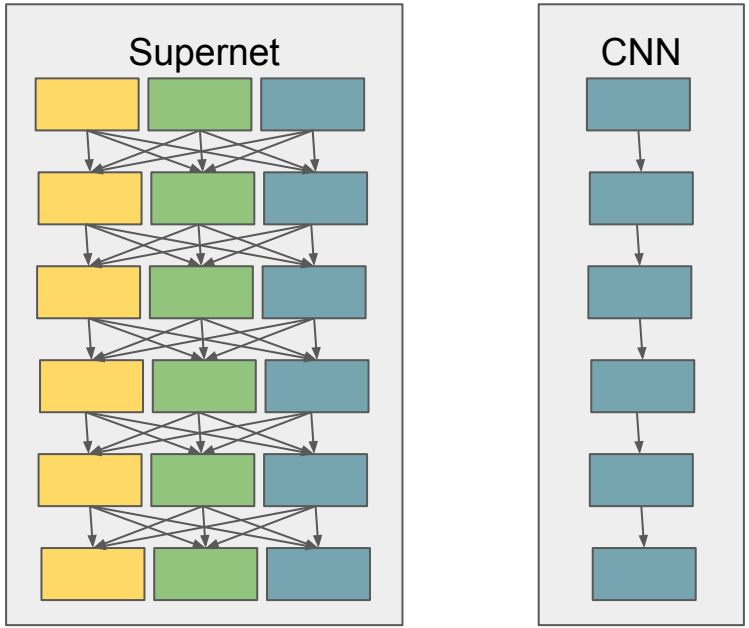
Progress Report

- Propose progressive one-shot neural architecture search(PONAS)
  - Search the best layer in the network progressively
    - The network constructed by the best block in each layer called ***startpoint network***
  - Record the validation accuracy of each candidate block to construct the ***candidate table***



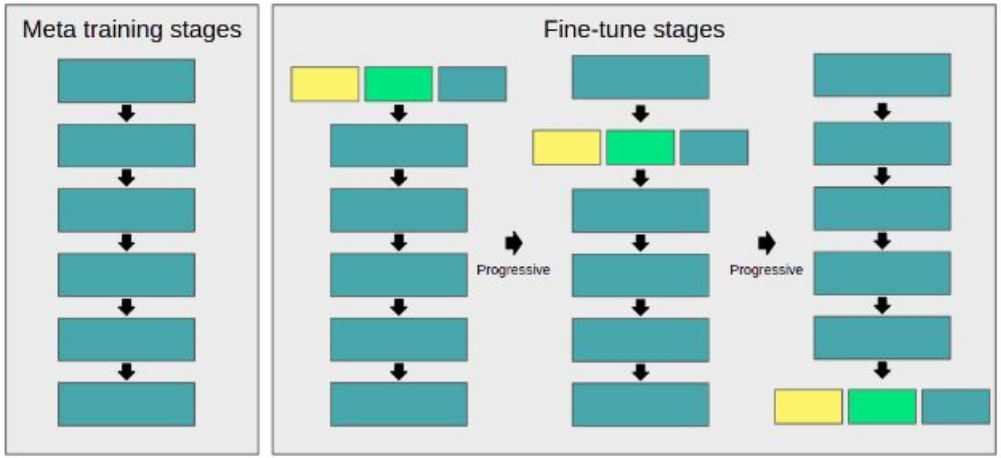
Progress Report

- The main reason about the first problem
  - Previos NAS encode entire search space into a supernet
  - It is hard to make supernet converge because of the complex structure of supernet



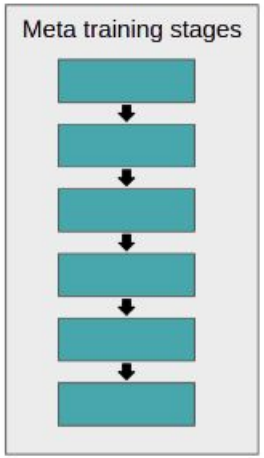
Progress Report

- To address the first problem, propose the two-stage training
  - The first stage called Meta-training stage and the second stage called Fine-tuning stage



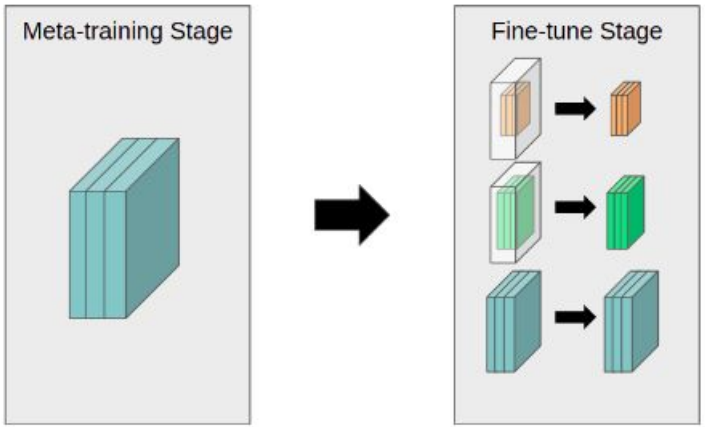
Progress Report

- In Meta-training stage
  - Construct the ***meta network*** which is the biggest network in search space
  - The main different between meta network and supernet
    - Meta network is single path
    - Supernet is multiple path
  - It is more efficient to train meta network than entire supernet



Progress Report

- In Fine-tune stage
  - Construct the supernet inherited weight from meta network to all candidate blocks
  - Fine-tune stage give a better initialization to all candidate blocks than random initialization
    - This allow all candidate blocks only need few epoch to make all blocks converge



Progress Report

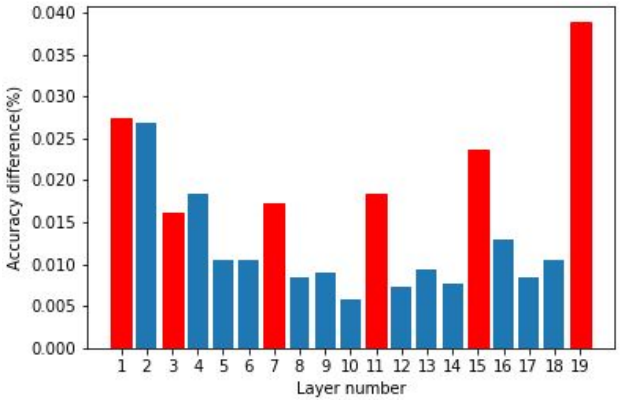
- The main reason about the second problem
- Previous One-Shot NAS
  - Flexibly support different hardware constraint and only training supernet once
  - Based on evolutionary algorithm or a pre-trained accuracy predictor to sample specific network
    - Still need lots of validation inference time to train accuracy predictor or execute evolutionary algorithm

Progress Report

- Propose Once-for-all algorithm to incorporate with PONAS
  - ***Does not need any time to specialize a specific network from startpoint network***

Progress Report

- Propose Once-for-all algorithm to incorporate with PONAS
  - Inspired by the analysis of the similarities between different block in the same layer
    - Make a hypothesis that the ***accuracy loss*** between the block can quantify importance of the candidate blocks
  - Visualize the ***max accuracy loss*** in each layer
    - The layer expanded channel size has the more important than other layers

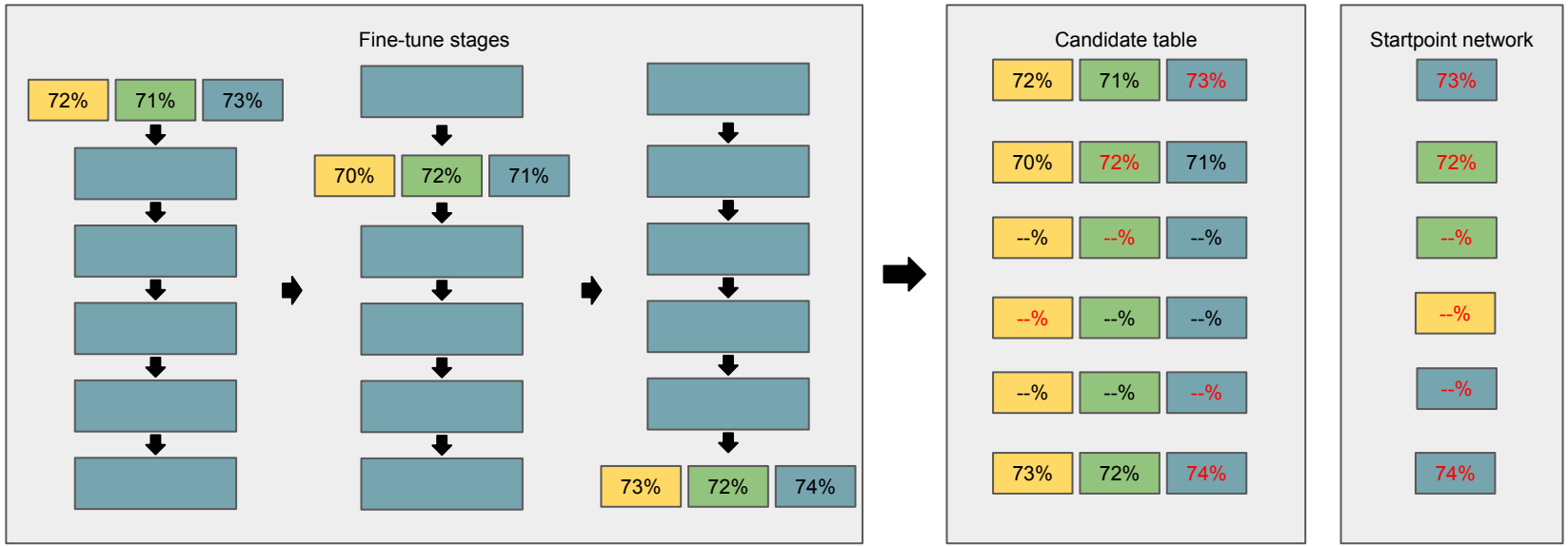


Progress Report

- Once-for-all algorithm
  - ***Choosing the block that minimize the accuracy loss from startpoint network while under the hardware constraint***
  - During Progressive One-Shot Neural Architecture
    - Construct a ***candidate table*** which record the validation accuracy of each candidate block in each layer
  - After PONAS
    - Constructed the ***startpoint network*** by the best block in each layer

Progress Report

- Once-for-all algorithm



Progress Report

- Once-for-all algorithm
  -

Algorithm 1 Once-for-all algorithm

Require: T_{cand}: candidate table; Net^*: best network architecture; $Cost$: hardware constraint;

initial $Net = Net^*$;
while $Cost(Net) > Cost$ **do**
    $T_{diff} = get\_cand\_diff\_table(T_{cand}, Net)$;
    $Net_{new} = replace\_minimize\_accuracy\_loss(Net, T_{diff})$;
    $Net = Net_{new}$;
end while;
return Net;

Progress Report

- Result

Model	Search method	Search space	Search Dataset	Search GPU hours	Deploy GPU hours	Params	FLOPs	Top-1 acc(%)	Top-5 acc(%)
MobileNetV2[16]	manual	-	-	-	-	3.4	300	72.0	91.0
MobileNetV2(1.4X)	manual	-	-	-	-	6.9	585	74.7	92.5
ShuffleNetV2(1.5X)[22]	manual	-	-	-	-	3.5	299	72.6	-
PNASNet[10]	SMBO	Cell	CIFAR-10			5.1	588	74.2	91.9
DPP-Net-Panacea[11]	SMBO	Cell	CIFAR-10			4.8	523	74.02	91.8
DARTS[12]	gradient	Cell	CIFAR-10			4.7	574	73.3	91.3
MnasNet-A1[4]	RL	stage	ImageNet			3.9	312	75.2	92.5
MnasNet-A2	RL	stage	ImageNet			4.8	340	75.6	92.7
ProxylessNAS-R[5]	RL	layer	ImageNet			4.1	320	74.6	92.2
FBNet-A[6]	gradient	layer	ImageNet			4.3	249	73.0	-
FBNet-B	gradient	layer	ImageNet			4.5	295	74.1	-
FBNet-C	gradient	layer	ImageNet			5.5	375	74.9	-
MobileNetV3-Large[23]	RL	stage	ImageNet			5.4	219	75.2	-
MobileNetV3-Small	RL	stage	ImageNet			2.9	66	67.4	-
SinglePathNAS[7]	One-Shot	layer	ImageNet			-	328	74.7	-
OFA w/ PS[8]	One-Shot	layer	ImageNet			-	230	76.00	-
FairNas-A[13]	RL	layer	ImageNet			4.6	388	75.34	92.38
FairNas-B	RL	layer	ImageNet			4.5	345	75.10	92.30
FairNas-C	RL	layer	ImageNet			4.4	321	74.69	92.12
Ours-A		layer	ImageNet			4.0	328	73.5	91.6
Ours-B		layer	ImageNet						
Ours-C		layer	ImageNet						

Progress Report