Introduction to Neural Architecture Search for Computer Vision

ECCV 2020 Tutorial on

From HPO to NAS: Automatic Deep Learning

Linjie Yang, ByteDance Inc.

Overview

- Background
- Search algorithms
- Search spaces
- Other directions
- Summarization & Discussions

Designing neural networks

Manually

AlexNet, GoogLeNet, VGG, ResNet

MobileNet V1/V2, ShuffleNet

Neural Architecture Search

Design a search space and a search algorithm,
search for structures automatically

ImageNet classification leaderboard

- 9 out of 10 top performing models are from NAS algorithms

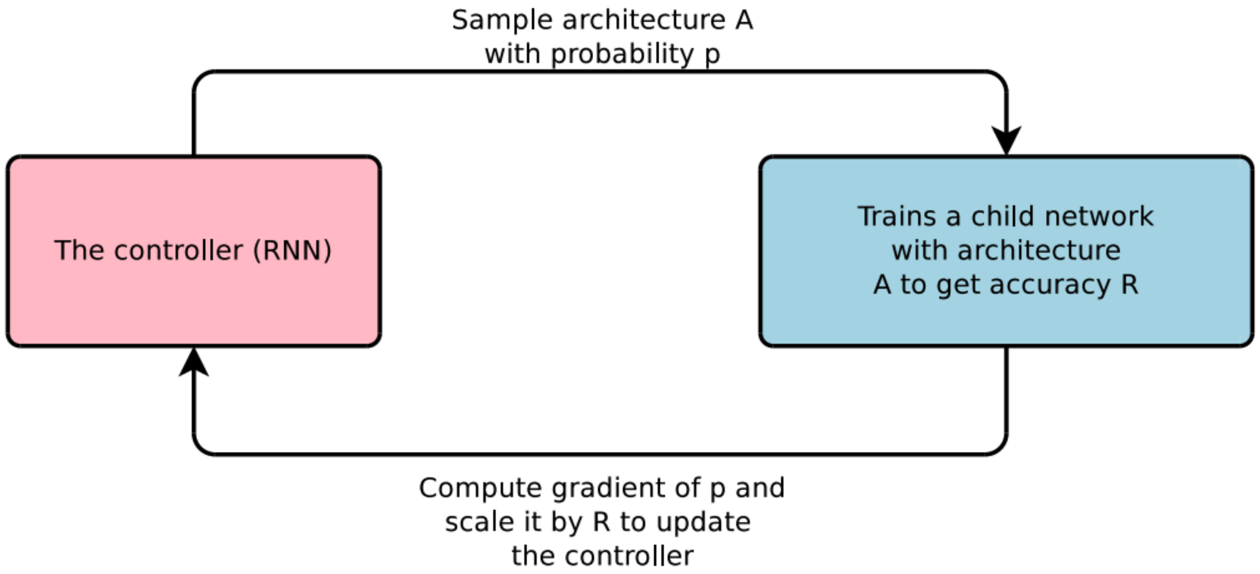
Model	Detail	Input size	Top-1 Acc	Top-5 Acc	Param(M)	Mult-Adds	FLOPS(G)
EfficientNet-B7	(2.0, 3.1, 600, 0.5)	600x600	84.4	97.1	66		37000
GPipe-AmoebaNet-B	(N=6, F=512)	480x480	84.3	97	557		
EfficientNet-B6	(1.8, 2.6, 528, 0.5)	528x528	84	96.9	43		19000
AmoebaNet-A	(N=6, F=448)	331x331	83.9	96.6	469	104B	
EfficientNet-B5	(1.6, 2.2, 456, 0.4)	456x456	83.3	96.7	30		9900
AmoebaNet-B	(N=6, F=228)	331x331	83.1	96.3	155.3	41.1B	
PNASNet-5_Large_331	(N=4, F=216)	331x331	82.9	96.2	86.1	25.0B	25.169
Oct-ResNet-152+SE	α=0.125, test:331	224x224	82.9	96.3	66.8		22.2
AmoebaNet-B	(N=6, F=190)	331x331	82.8	96.1	86.7	23.1B	

https://kobiso.github.io/Computer-Vision-Leaderboard/imagenet.html

Search algorithms

- Reinforcement Learning
- Evolution algorithms
- Differentiable search

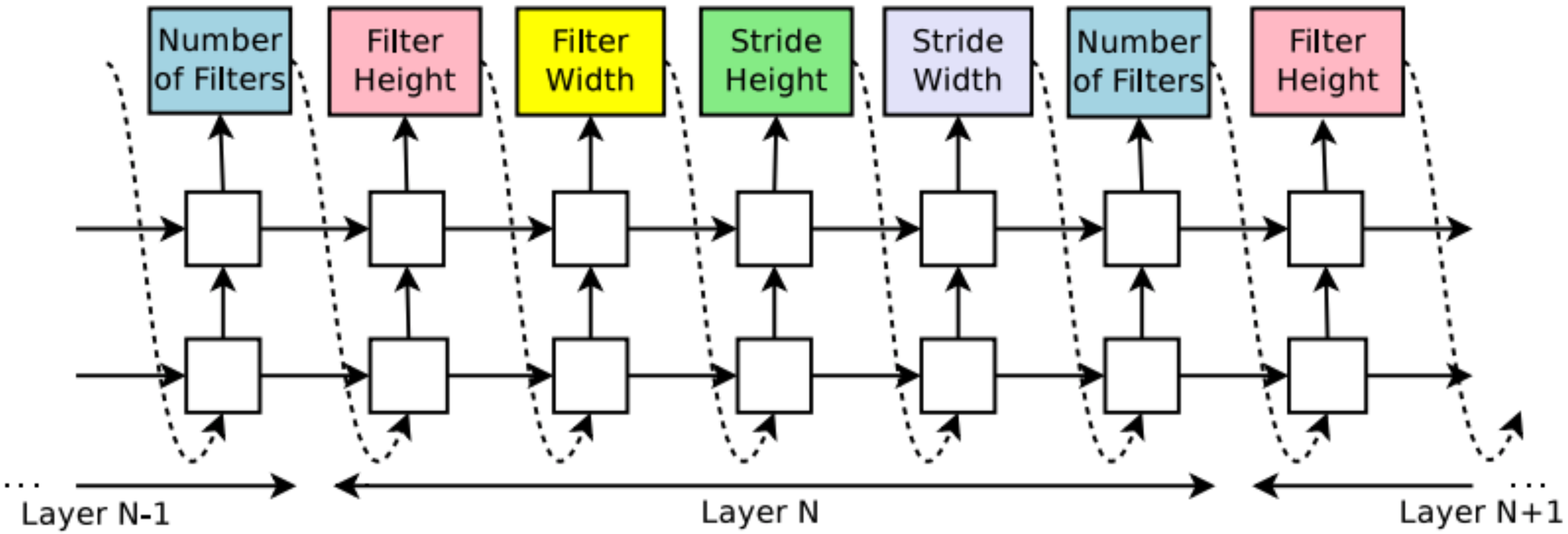
Reinforcement Learning



Neural Architecture Search with Reinforcement Learning. Zoph and Le. ICLR 2017.

Reinforcement Learning

- Generate layer parameters sequentially with an RNN controller

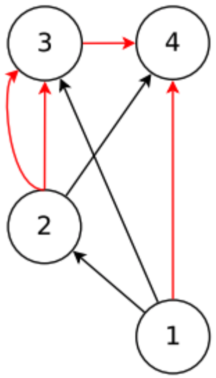


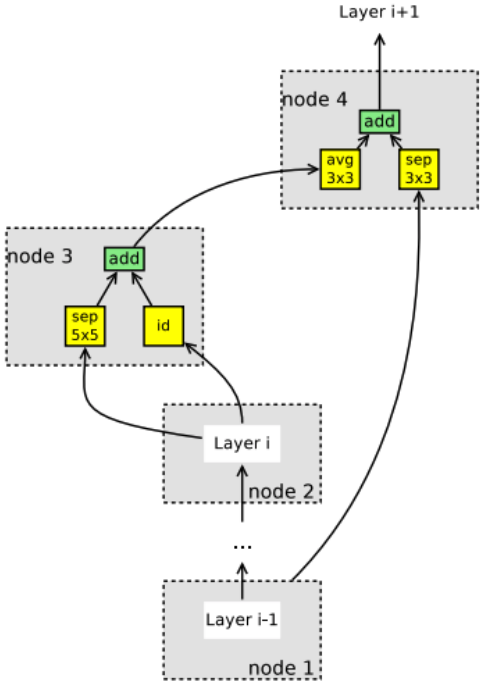
Neural Architecture Search with Reinforcement Learning. Zoph and Le. ICLR 2017.

Reinforcement Learning – parameter sharing

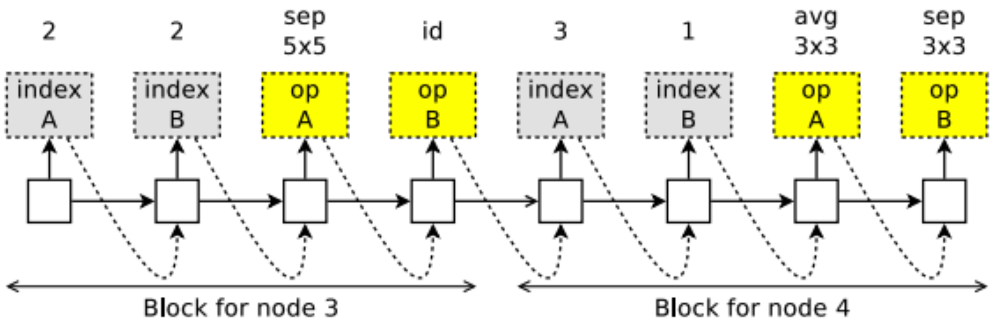
- ENAS: Train every sampled model from scratch is too slow – share weights to speed up training

Use a predefined network graph. Only search connections from the graph. For each connection, search an operation.





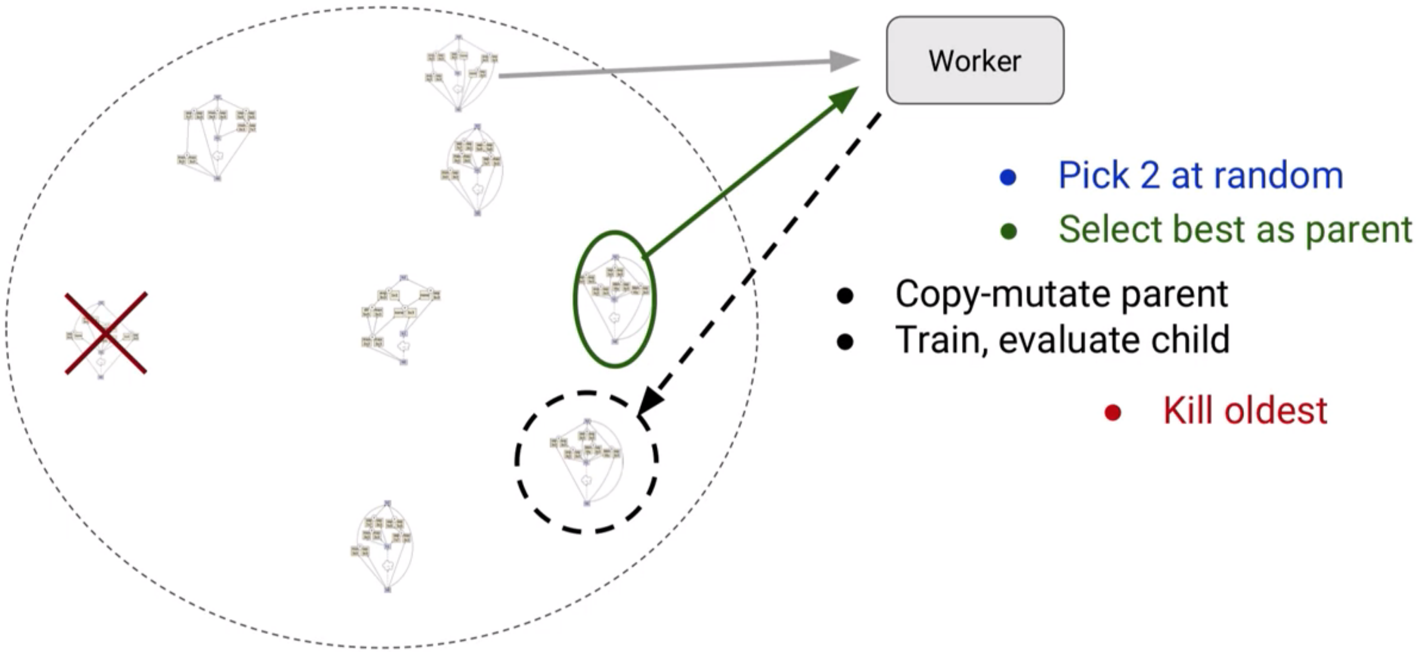
Use RNN to predict connection and operation.

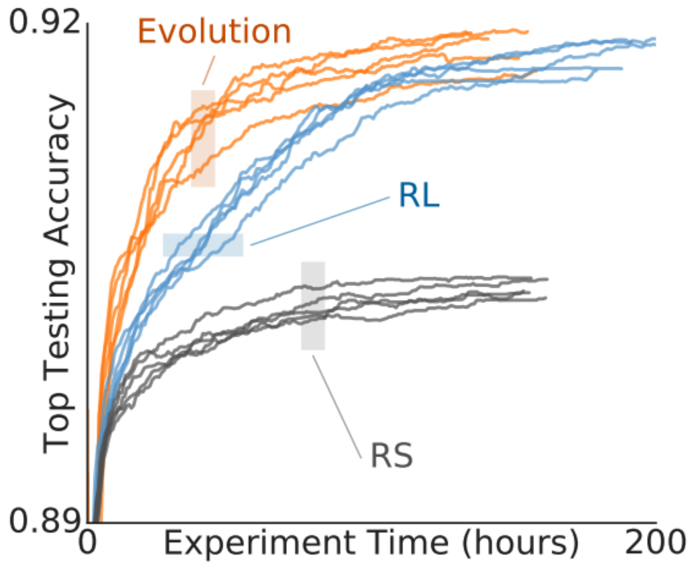


Efficient Neural Architecture Search via Parameter Sharing. Pham et al. ICML 2018.

Evolution algorithms

- AmoebaNet

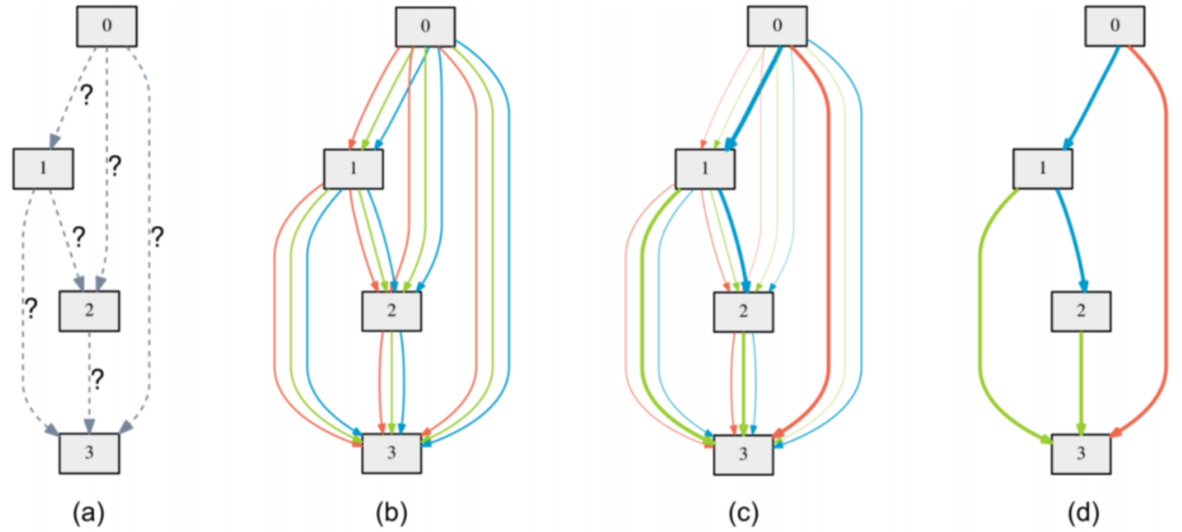




Regularized Evolution for Image Classifier Architecture Search. Real et al. AAAI 2019.
Image Credit: Esteban Real

Differentiable search - DARTS

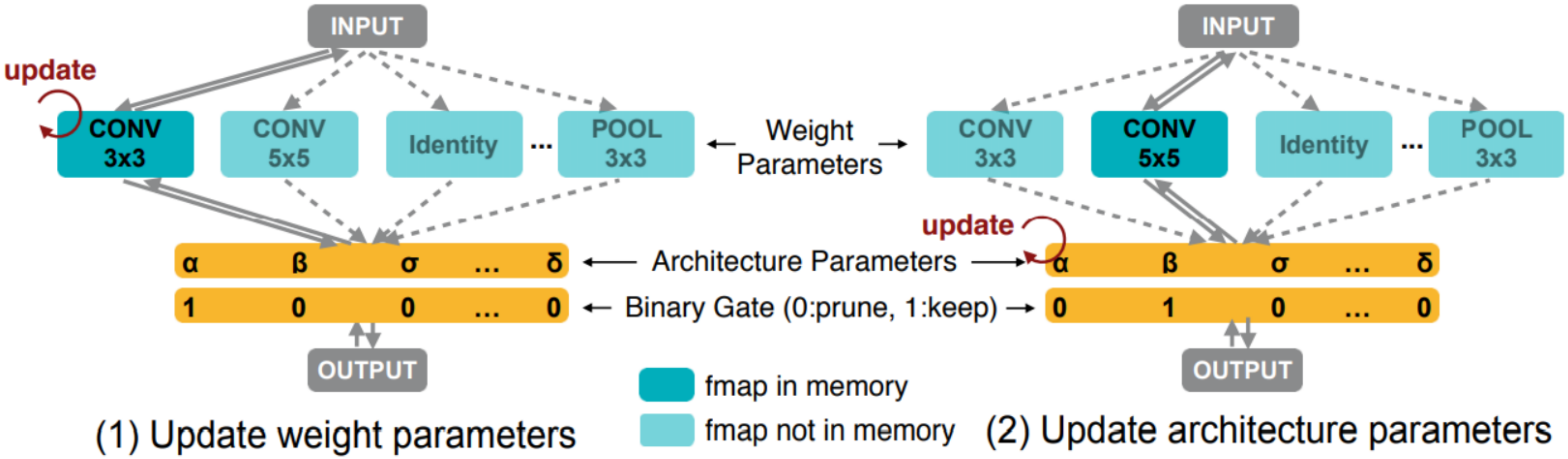
- Search for a subnetwork in a super-network.
  - Assign a learnable importance weight for each edge which is optimized jointly with the other network weights.
  - Weights shared across different subnetworks.
  - Prune the final model according to the importance weights.
  - Retrain the model.



DARTS: Differentiable Architecture Search. Liu et al. ICLR 2019.

Differentiable search – ProxylessNAS

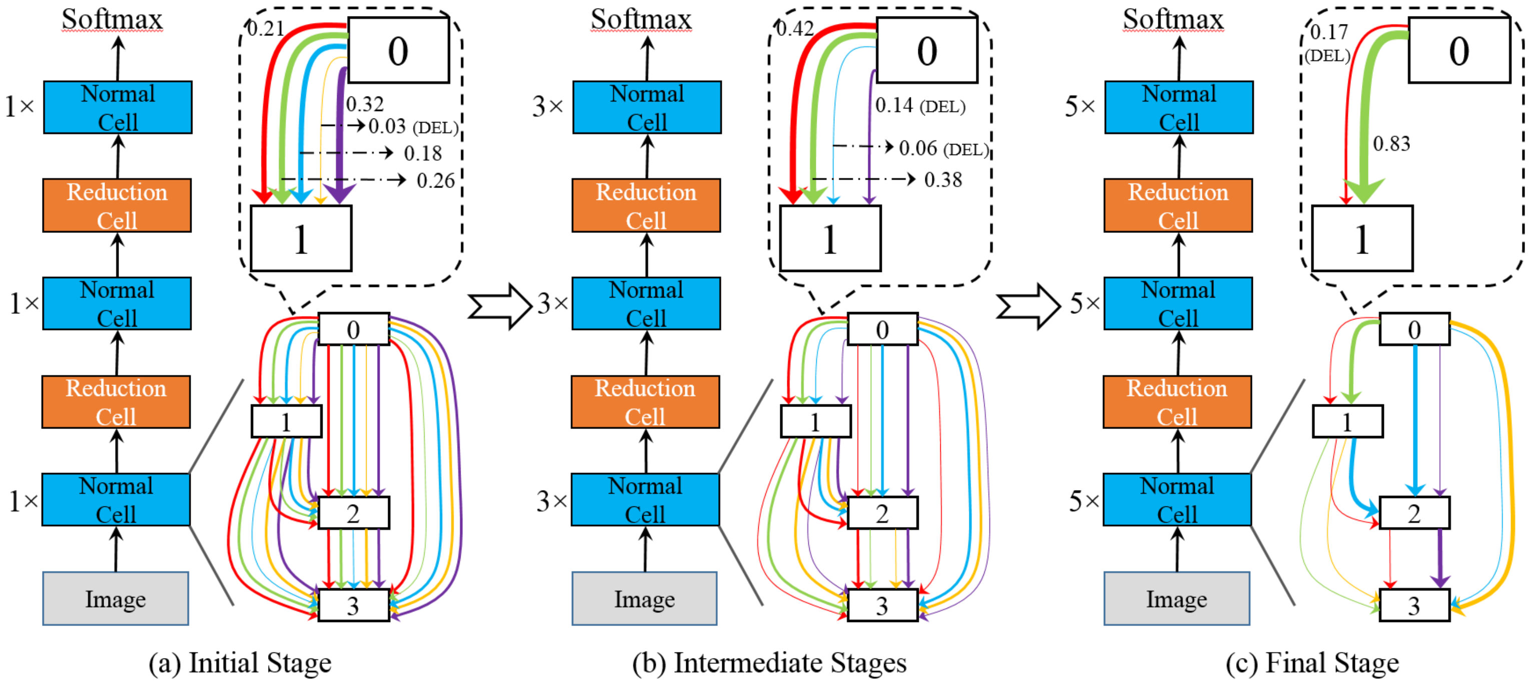
- All previous method can only search for structures on small scale dataset such as CIFAR10 and then transfer to large scale dataset such as ImageNet with a larger and deeper model.
- ProxylessNAS is the first to directly search on ImageNet dataset. It only loads a sampled subnetwork into GPU at each iteration to avoid memory overflow.
- Can search structure under predefined resource constraints.



ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. Cai et al. ICLR 2019

Differentiable search – P-DARTS

- For DARTS, how to bridge the gap between search and final model?
- P-DARTS: A multi-stage search progress which gradually increases the search depth



Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. Chen et al. ICCV 2019

Differentiable search – Results

Architecture	Test Err. (%)		Params	×+	Search Cost	Search Method
	top-1	top-5	(M)	(M)	(GPU-days)	
Inception-v1 [29]	30.2	10.1	6.6	1448	-	manual
MobileNet [9]	29.4	10.5	4.2	569	-	manual
ShuffleNet 2× (v1) [34]	26.4	10.2	∼5	524	-	manual
ShuffleNet 2× (v2) [19]	25.1	-	∼5	591	-	manual
NASNet-A [37]	26.0	8.4	5.3	564	1800	RL
NASNet-B [37]	27.2	8.7	5.3	488	1800	RL
NASNet-C [37]	27.5	9.0	4.9	558	1800	RL
AmoebaNet-A [22]	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B [22]	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C [22]	24.3	7.6	6.4	570	3150	evolution
PNAS [16]	25.8	8.1	5.1	588	225	SMBO
MnasNet-92 [31]	25.2	8.0	4.4	388	-	RL
DARTS (second order) [18]	26.7	8.7	4.7	574	4.0	gradient-based
SNAS (mild constraint) [33]	27.3	9.2	4.3	522	1.5	gradient-based
ProxylessNAS (GPU) [2]	24.9	7.5	7.1	465	8.3	gradient-based
P-DARTS (searched on CIFAR10)	24.4	7.4	4.9	557	0.3	gradient-based
P-DARTS (searched on CIFAR100)	24.7	7.5	5.1	577	0.3	gradient-based

Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. Chen et al. ICCV 2019

Search algorithms - Comparison

	Reinforcement Learning	Evolution Algorithm	Differentiable Search
Computation cost	High	High	Low
Search space	Large	Large	Restricted

Pros and Cons of weight sharing

- Differentiable search all use weight sharing to facilitate joint optimization of different model candidates.
- Pros:
  - Weight sharing speed up the convergence of different candidate models.
- Cons:
  - Weight sharing entangles different candidate models and prevents good convergence of each candidate.
  - The ranking of the candidate models is not guaranteed to be preserved. The selected final model could be suboptimal.

Evaluating the search phase of neural architecture search. Yu et al. ICLR 2020
NAS-Bench-1Shot1: Benchmarking and Dissecting One-shot Neural Architecture Search. Zela et al. ICLR 2020

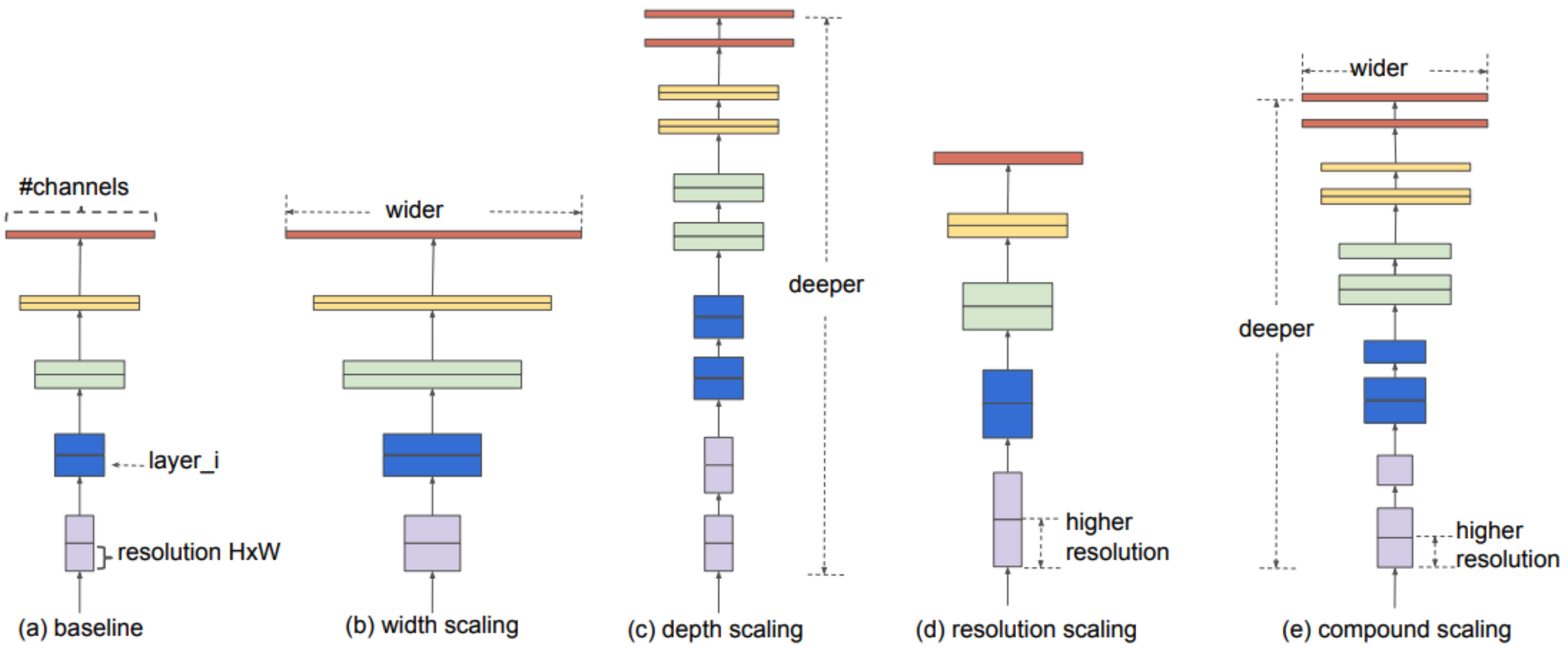
Search spaces

- General DAG with a set of operators
  - Reinforcement Learning (no weight sharing)
  - Evolution algorithms
    - AmoebaNet

- Subgraph of a supergraph
  - Reinforcement Learning (weight sharing)
    - ENAS
  - Differentiable search
    - DARTS, ProxylessNAS, P-DARTS
- New horizon: Model Scaling
  - Input resolution, depth, width etc.
    - EfficientNet, Once-For-All

Model Scaling - EfficientNet

- Scale width, depth, resolution for a base network.

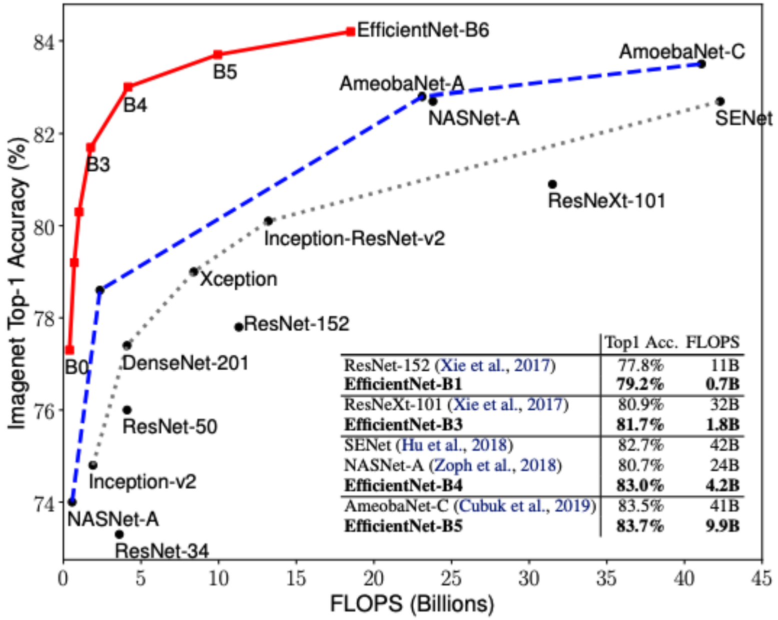


EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Tan et al. ICML 2019

Model Scaling - EfficientNet

- Use a compound coefficient ϕ to uniformly scales network width, depth, and resolution
- Outperform all previous NAS algorithms under same flops

$$
\begin{aligned}
\text{depth: } & d = \alpha^{\phi} \\
\text{width: } & w = \beta^{\phi} \\
\text{resolution: } & r = \gamma^{\phi} \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}
$$



	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**79.2%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.7%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**83.0%**	**4.2B**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.7%**	**9.9B**

Model Scaling - EfficientNet

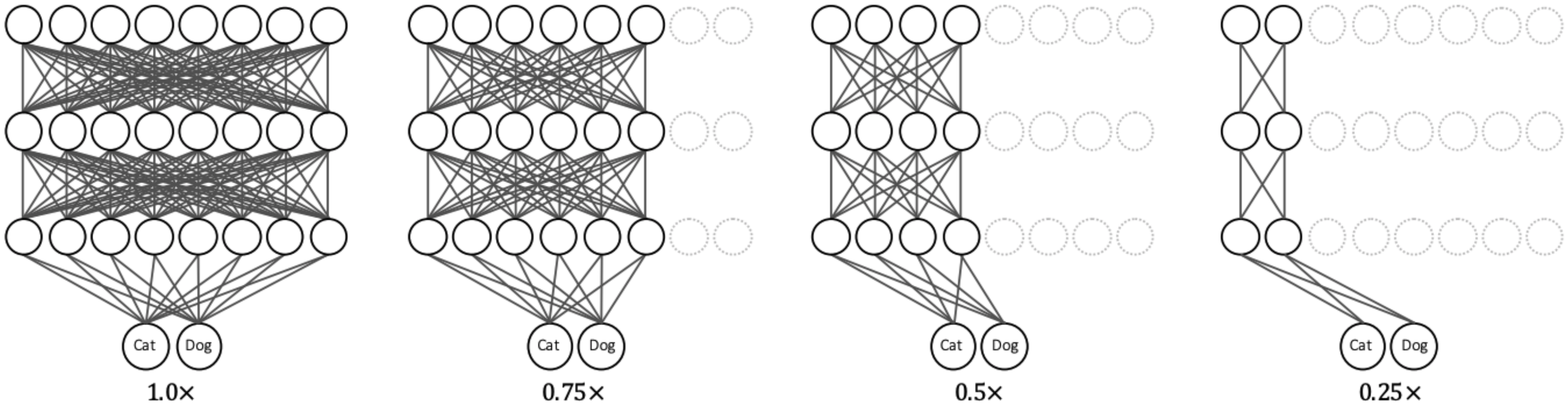
- Why such a simple method work?
  - Limited exploration on the dimension of input resolution, depth and width.
  - Traditional NAS algorithm usually search with operations in fixed width and fixed input size.
  - Scaling up on these dimensions increases computation cost, but also increases model performance effectively.

Model Scaling + weight sharing

- Are weight sharing possible among different input resolution, depth, width?
- Does weight sharing across these axes also degenerate model performance?

Width scaling + weight sharing

- Slimmable neural networks



Slimmable neural networks. Yu et al. ICLR 2019

Width scaling + weight sharing

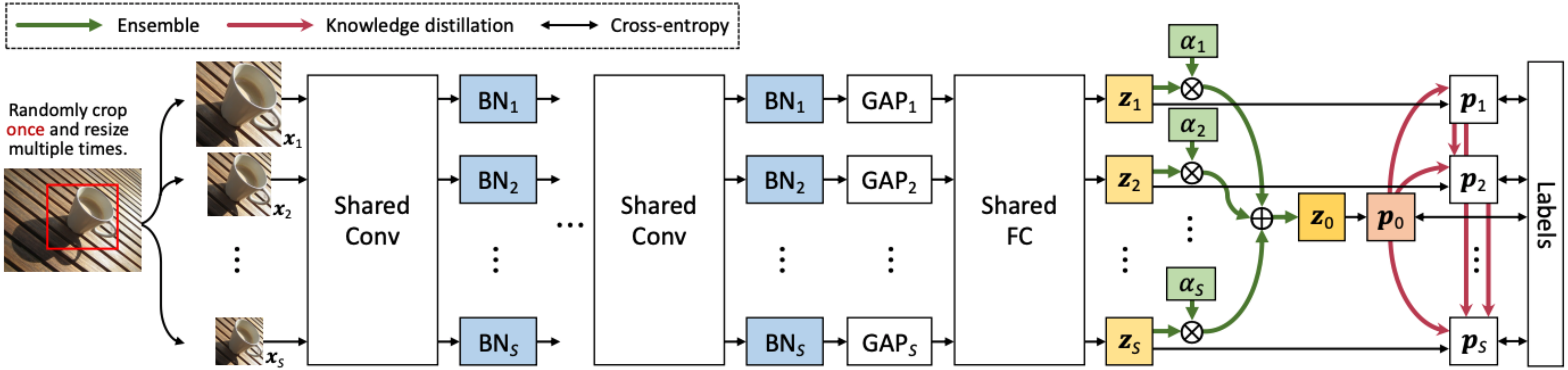
- Performance does not degenerate with weight sharing
- A single model can be deployed and executed with multiple width settings.

Individual Networks			Slimmable Networks			FLOPs
Name	Params	Top-1 Err.	Name	Params	Top-1 Err.	
MobileNet v1 1.0×	4.2M	29.1	S-MobileNet v1 [0.25, 0.5, 0.75, 1.0]×	4.3M	28.5 (0.6)	569M
MobileNet v1 0.75×	2.6M	31.6			30.5 (1.1)	317M
MobileNet v1 0.5×	1.3M	36.7			35.2 (1.5)	150M
MobileNet v1 0.25×	0.5M	50.2			46.9 (3.3)	41M
MobileNet v2 1.0×	3.5M	28.2	S-MobileNet v2 [0.35, 0.5, 0.75, 1.0]×	3.6M	29.5 (-1.3)	301M
MobileNet v2 0.75×	2.6M	30.2			31.1 (-0.9)	209M
MobileNet v2 0.5×	2.0M	34.6			35.6 (-1.0)	97M
MobileNet v2 0.35×	1.7M	39.7			40.3 (-0.6)	59M
ShuffleNet 2.0×	5.4M	26.3	S-ShuffleNet [0.5, 1.0, 2.0]×	5.5M	28.7 (-2.4)	524M
ShuffleNet 1.0×	1.8M	32.6			34.5 (-0.9)	138M
ShuffleNet 0.5×	0.7M	43.2			42.7 (0.5)	38M
ResNet-50 1.0×	25.5M	23.9	S-ResNet-50 [0.25, 0.5, 0.75, 1.0]×	25.6M	24.0 (-0.1)	4.1G
ResNet-50 0.75×†	14.7M	25.3			25.1 (0.2)	2.3G
ResNet-50 0.5×†	6.9M	28.0			27.9 (0.1)	1.1G
ResNet-50 0.25×†	2.0M	36.2			35.0 (1.2)	278M

Slimmable neural networks. Yu et al. ICLR 2019

Resolution scaling + weight sharing

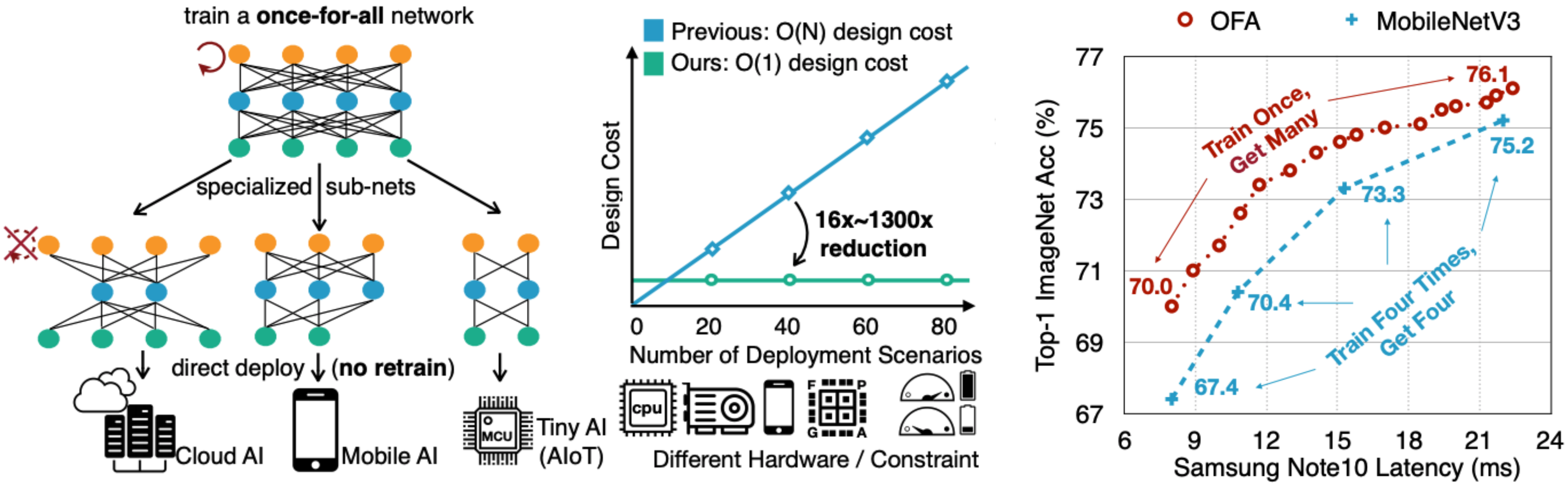
- Performance also does not drop with weight sharing
- A single model can be deployed and executed with multiple input resolutions.



Resolution Switchable Networks for Runtime Efficient Image Recognition. Wang et al. ECCV 2020.

Multiple scaling axes + weight sharing

- A search space with multiple axes: width, depth, kernel size, resolution.
  - Due to too many coupled candidate models, jointly train them all will have accuracy drop.
  - Use a progressive shrinking technique to reduce search space gradually



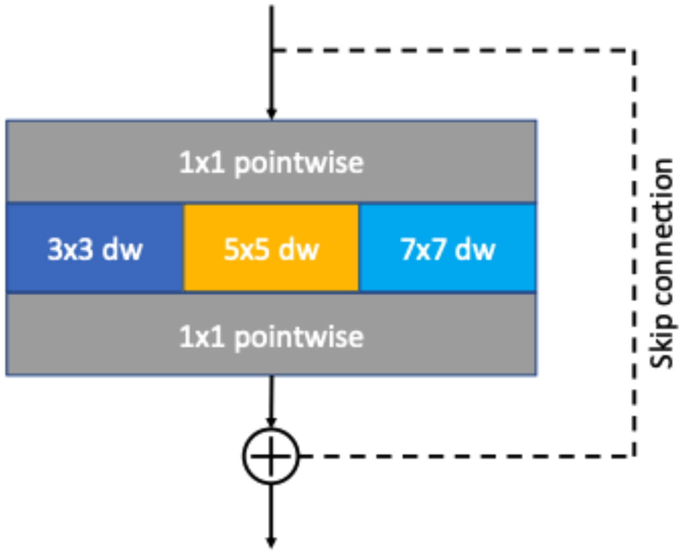
Once-for-All: Train One Network and Specialize it for Efficient Deployment. Cai et al. ICLR 2020.

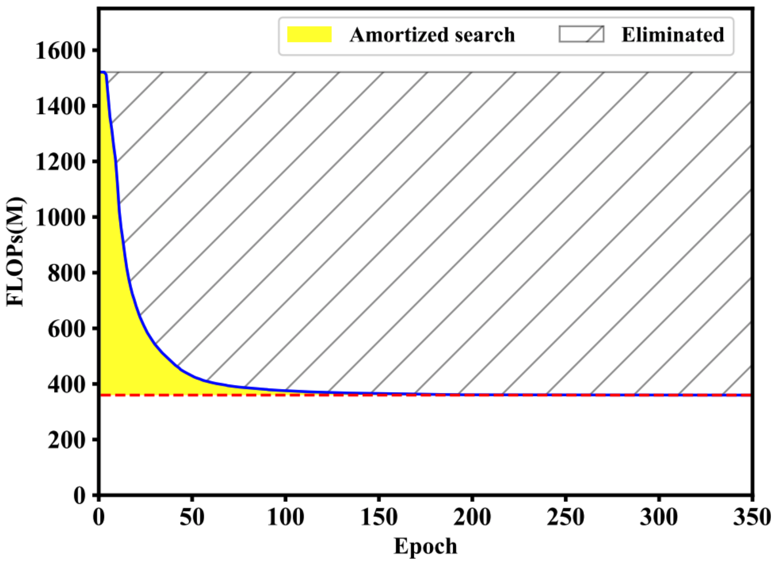
Neural architecture search – other directions

- Fine-grained search space
- New operators
- On other vision tasks: detection, segmentation etc.

Fine-grained search space

- AtomNAS: Fine-grained channel numbers + operations
- Gradually reduce low importance channels in the search stage to reduce computation cost

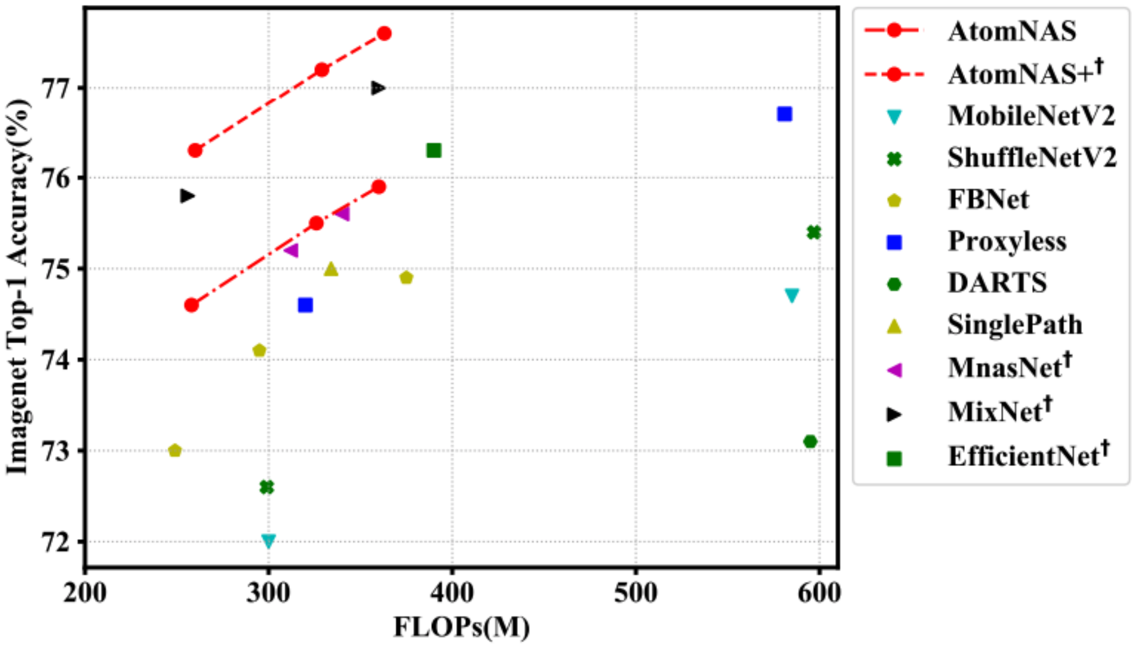




AtomNAS: Fine-Grained End-to-End Neural Architecture Search. Mei et al. ICLR 2020.

Fine-grained search space

- Outperform EfficientNet on mobile settings.



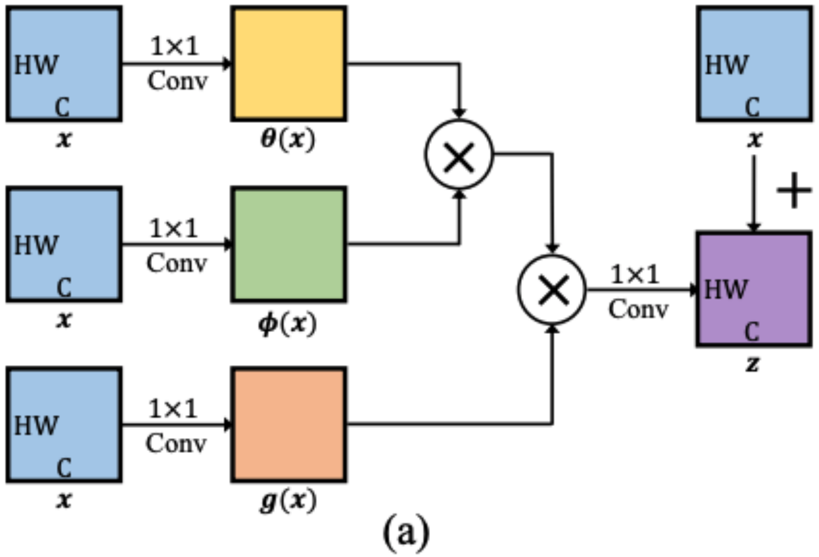
AtomNAS: Fine-Grained End-to-End Neural Architecture Search. Mei et al. ICLR 2020.

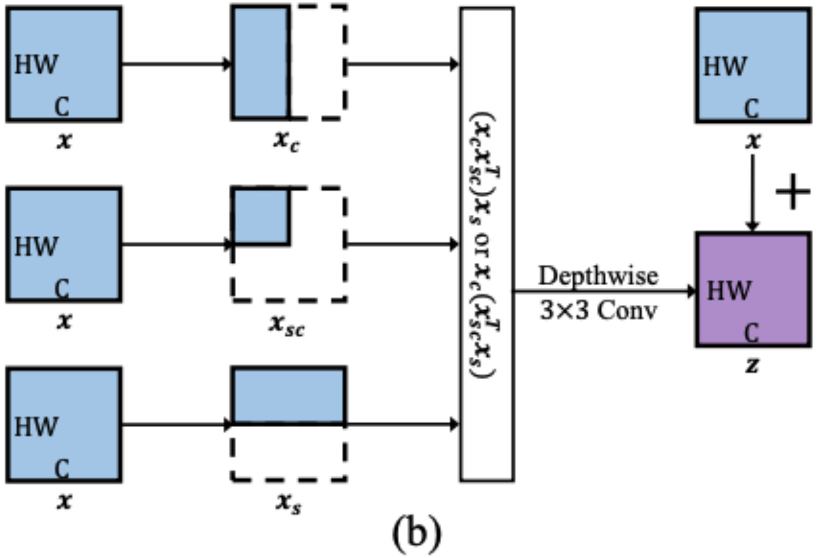
New operators

- Traditional NAS algorithms only use a small range of operators: convolution, linear, ReLU, pooling, skip-connection etc.
- New operators can be added to enlarge the search space and enable more powerful models.

New operators

- Non-local operator





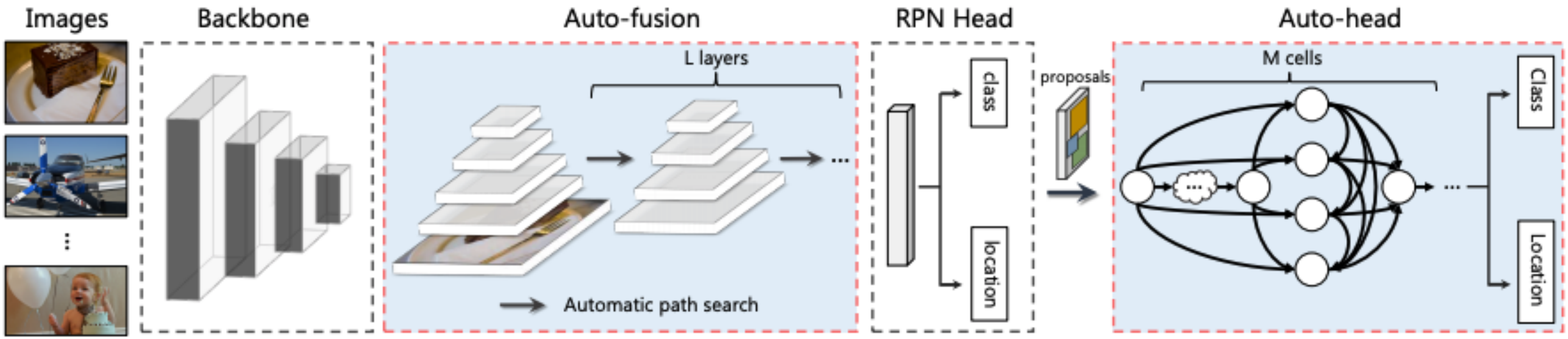
Original non-local block

Lightweight non-local block

Neural Architecture Search for Lightweight Non-Local Networks. Li et al. CVPR 2020.
Non-local neural networks. Wang et al. CVPR 2018.

NAS on detection

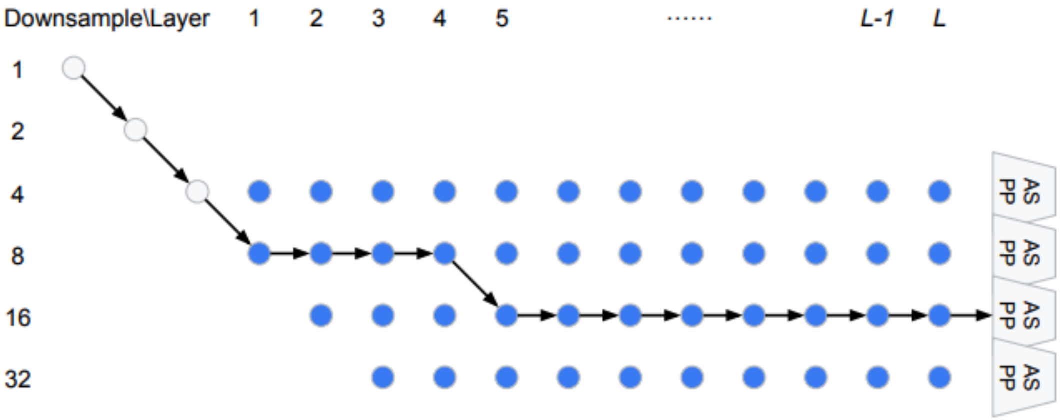
- NAS-FPN: search for a feature pyramid network using reinforcement learning
- Auto-FPN: search for a feature fusion module for RPN, and a class/location prediction head using differentiable search.



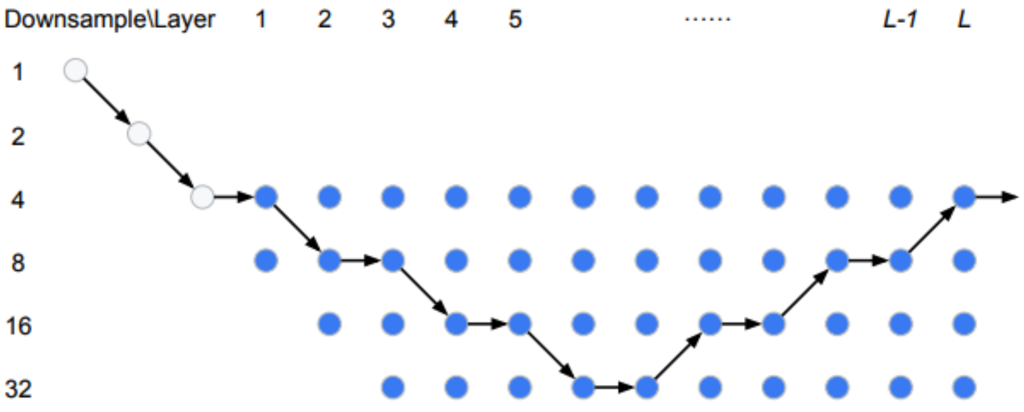
NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. Ghiasi et al. CVPR 2019.
Auto-FPN: Automatic Network Architecture Adaptation for Object Detection Beyond Classification. Xu et al. ICCV 2019

NAS on segmentation

- Auto-DeepLab: search for a downsampling-upsampling path in a 2-D search space.



(a) Network level architecture used in DeepLabv3 [9].



(b) Network level architecture used in Conv-Deconv [56].

Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. Liu et al. CVPR 2019.

Summarization

- NAS-based models outperforms hand crafted models in a wide-range of tasks.
- Neural architecture search is not fully automatic. Designing search space and search algorithm still need a lot of manual effort.
- Differentiable search is more popular than other search algorithms in recent publications due to its efficiency and good performance.
- Different tasks may need different search spaces.

Pitfalls of NAS

- Comparing different NAS algorithms is hard due to the different search spaces and different settings. NAS algorithms can also have different rankings on different datasets.
- The performance gain of NAS algorithms mainly comes from a well-designed search space rather than the search algorithms.
- NAS algorithms do not guarantee to find optimal solutions. Weight sharing speeds up the algorithm but disrupts the true ranking of the candidate models.
- A recent trend of research is to compare different NAS algorithms on the same benchmark.

NAS-Bench-101: Towards Reproducible Neural Architecture Search. Ying et al. ICML 2019.
NAS-Bench-1Shot1: Benchmarking and Dissecting One-shot Neural Architecture Search. Zela et al. ICLR 2020
NAS evaluation is frustratingly hard. Yang et al. ICLR 2020.

Useful materials

- AutoML: A Survey of the State-of-the-Art. He et al. ArXiv 2020.
- A compiled list of NAS literature: https://www.automl.org/automl/literature-on-neural-architecture-search/
- A curated list of automated deep learning related resources: https://github.com/D-X-Y/Awesome-AutoDL

Thank you for attending this session!

Enjoy ECCV 2020!

References

- Neural Architecture Search with Reinforcement Learning. Zoph and Le. ICLR 2017.
- Efficient Neural Architecture Search via Parameter Sharing. Pham et al. ICML 2018.
- Regularized Evolution for Image Classifier Architecture Search. Real et al. AAAI 2019.
- DARTS: Differentiable Architecture Search. Liu et al. ICLR 2019.
- ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. Cai et al. ICLR 2019
- Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. Chen et al. ICCV 2019
- Evaluating the search phase of neural architecture search. Yu et al. ICLR 2020
- NAS-Bench-1Shot1: Benchmarking and Dissecting One-shot Neural Architecture Search. Zela et al. ICLR 2020
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Tan et al. ICML 2019
- Slimmable neural networks. Yu et al. ICLR 2019
- Resolution Switchable Networks for Runtime Efficient Image Recognition. Wang et al. ECCV 2020.
- Once-for-All: Train One Network and Specialize it for Efficient Deployment. Cai et al. ICLR 2020.

References (cont.)

- AtomNAS: Fine-Grained End-to-End Neural Architecture Search. Mei et al. ICLR 2020.
- Neural Architecture Search for Lightweight Non-Local Networks. Li et al. CVPR 2020.
- Non-local neural networks. Wang et al. CVPR 2018.
- NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. Ghiasi et al. CVPR 2019.
- Auto-FPN: Automatic Network Architecture Adaptation for Object Detection Beyond Classification. Xu et al. ICCV 2019
- Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. Liu et al. CVPR 2019.
- NAS-Bench-101: Towards Reproducible Neural Architecture Search. Ying et al. ICML 2019.
- NAS evaluation is frustratingly hard. Yang et al. ICLR 2020.