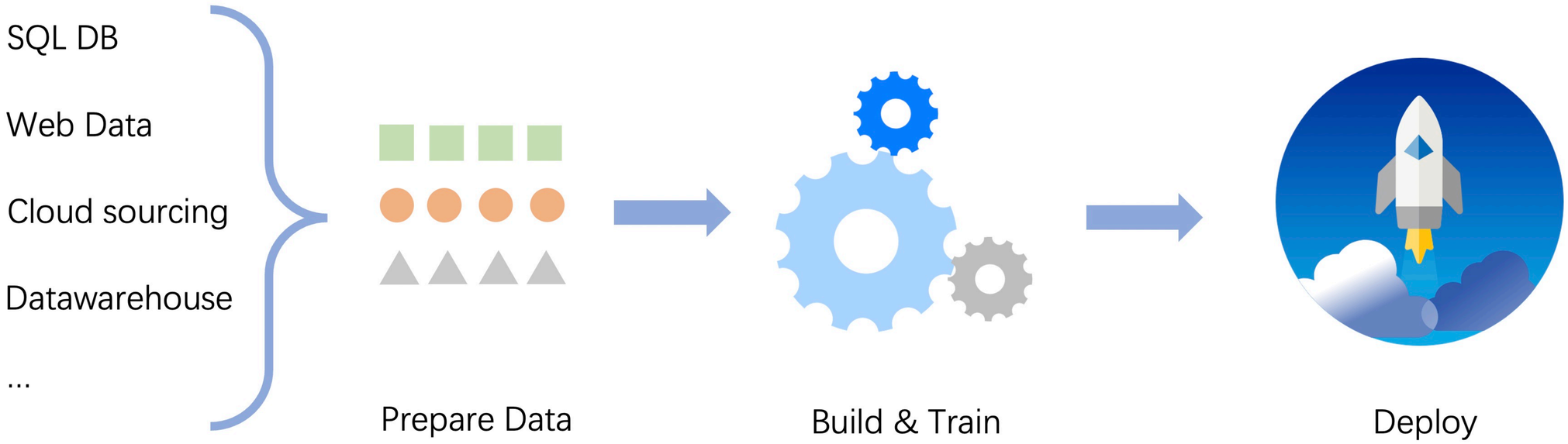
COMP5212: Machine Learning

Lecture 19

Minhao Cheng

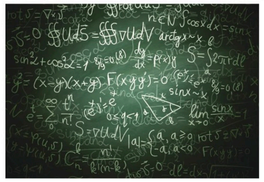
Machine learning pipeline



Machine learning pipeline

The devil is in the details

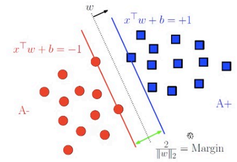
- What feature
  - Constraint/Rule



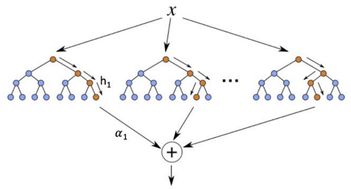
  - Budget

  - Efficiency

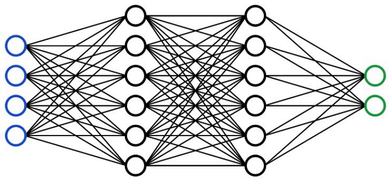
- Which model
  - Linear model



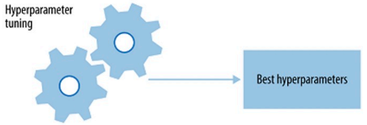
  - Boosting model

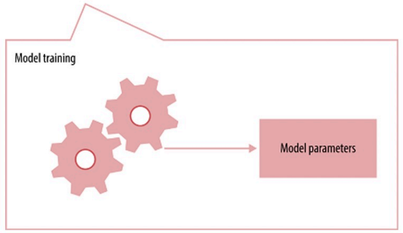


  - Neural network

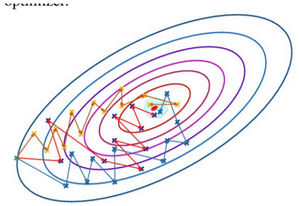


- Which parameter
  - Hyperparameter





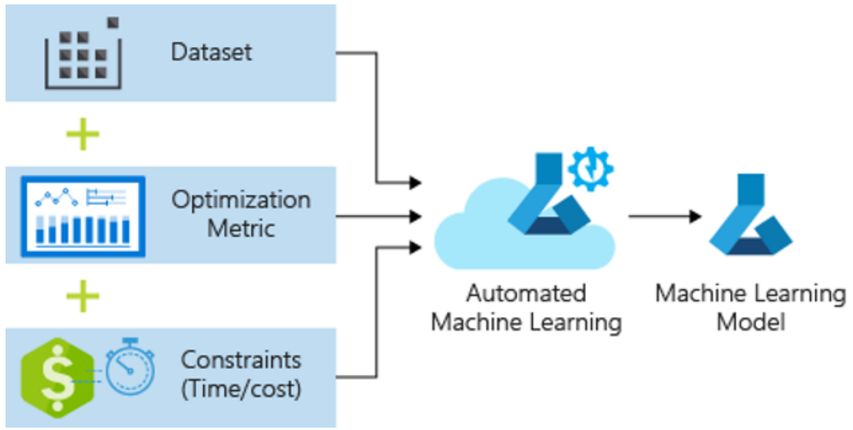
  - Optimizer



Automated Machine learning

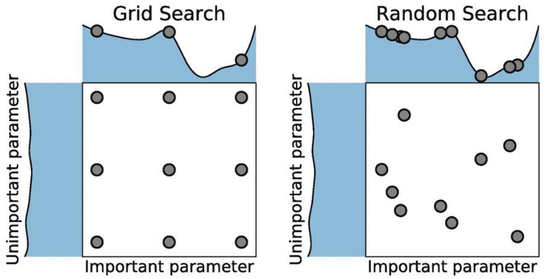
The devil is in the details

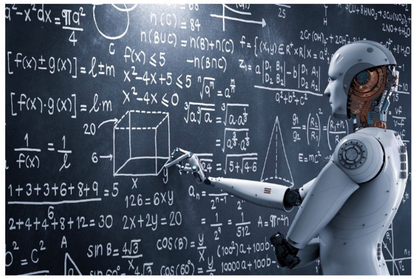
- AutoML simplifies each step in the machine learning process, from handling a raw dataset to deploying a practical machine learning model.

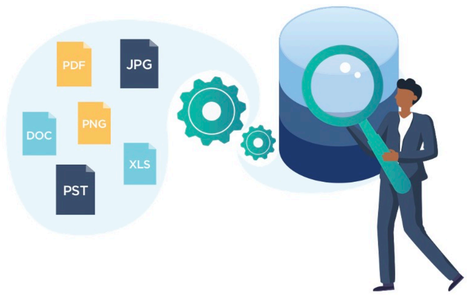


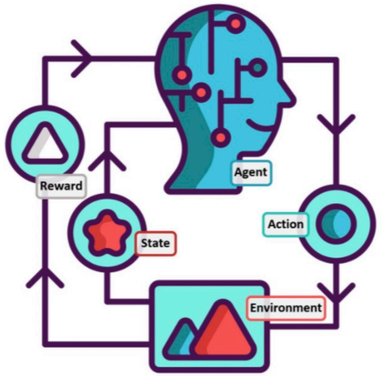
Automated Machine learning

- AutoML:
  - Neural Architecture Search (NAS)
  - Hyperparameter Optimization (HPO)
  - Meta Learning and Learning to learn
  - Automated Reinforcement learning
  - AutoML in Physical World
  - Automated model selection
  - …





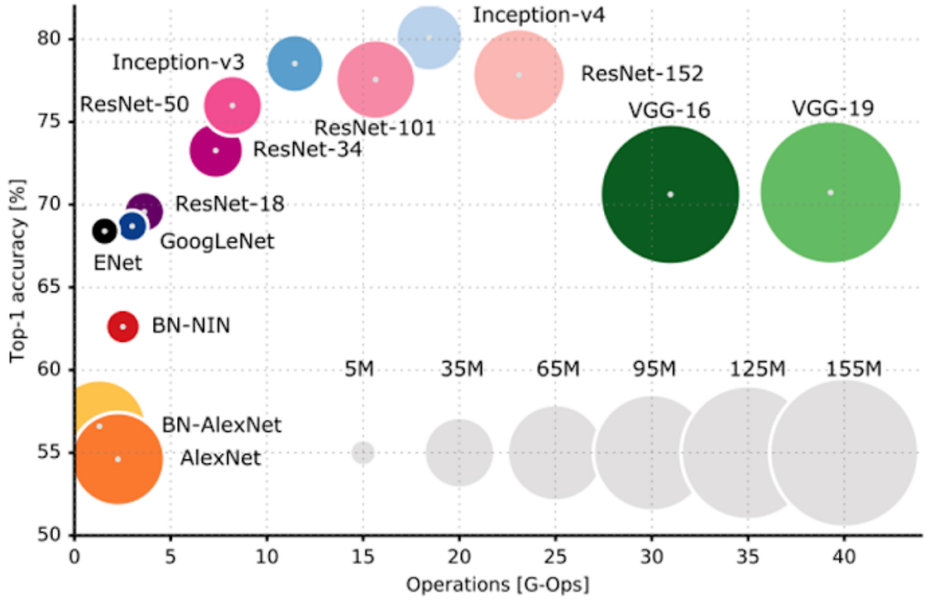


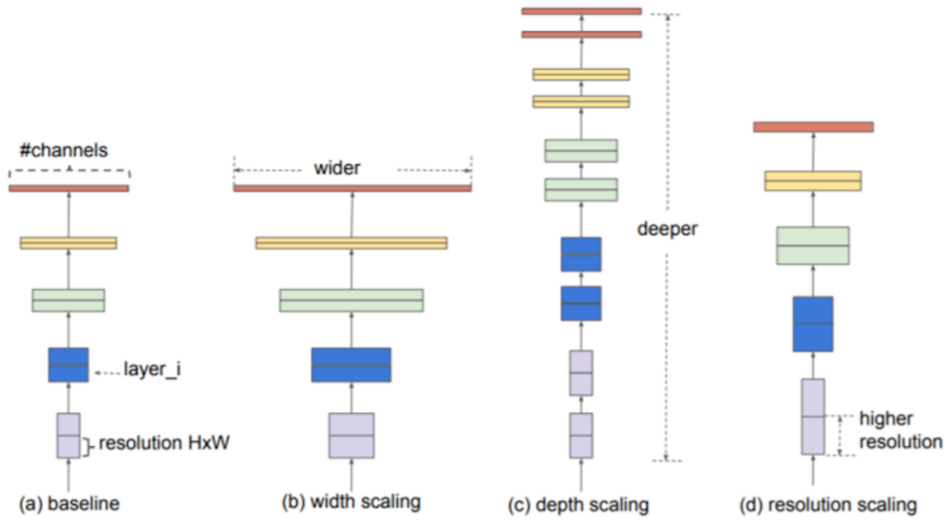


AutoML

Architecture of Neural Network

- Neural network architecture is important for both accuracy and efficiency



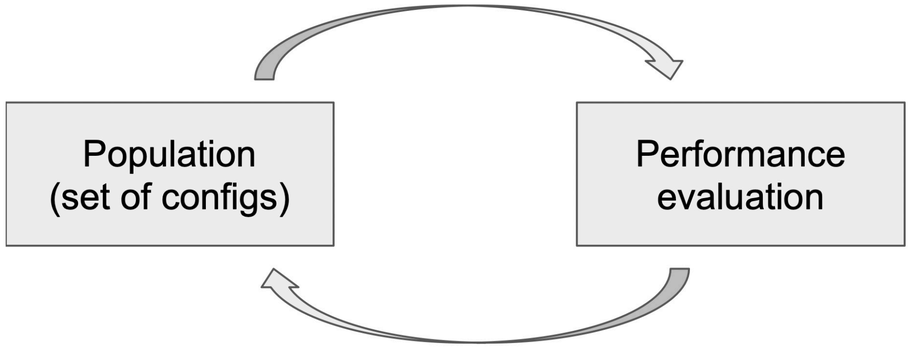


Can we automatically design architecture?

Neural Architecture Search

History of Neural Architecture Search (NAS)

- Early years: only on toy or small-scaled problems
  - Evolutionary algorithms (Miller et al., 89; Schaffer et al., 92; Verbancsics & Harguess, 13)
  - Bayesian optimization (Snoek et al, 12; Domhan et al., 15)



Neural Architecture Search

An early example

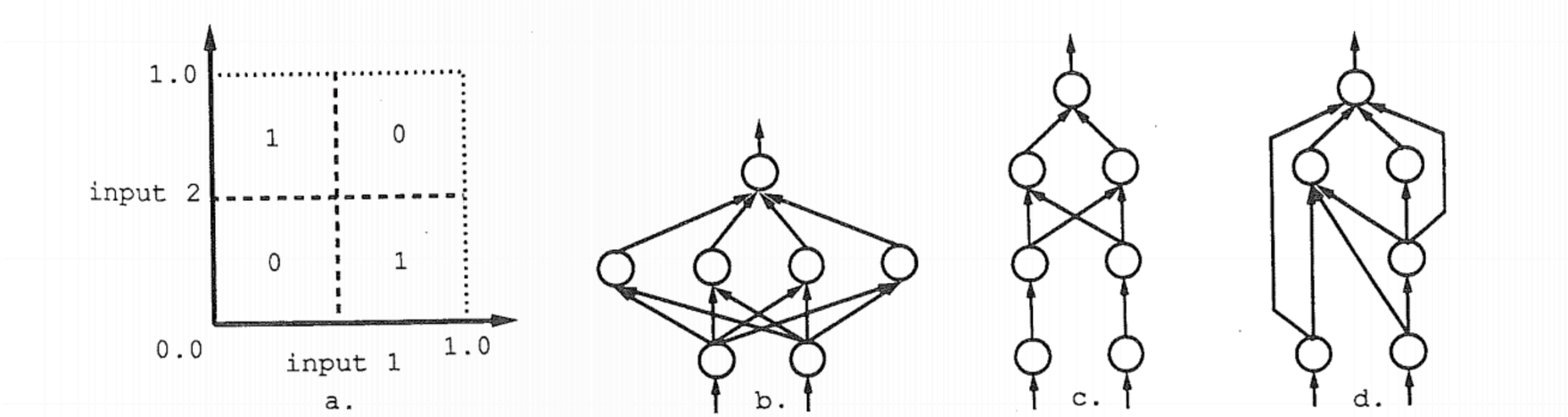


Figure 3. The four-quadrant problem. a. 2-d mapping to be learned. b. Standard 3-layer architectural solution. c. Standard 4-layer architectural solution. d. Typical discovered architectural solution.

Neural Architecture Search

- In 2016, Reinforcement learning (RL) is proposed for NAS
  - A better (structured) representation of search space
  - Learning a controller to generate architectures

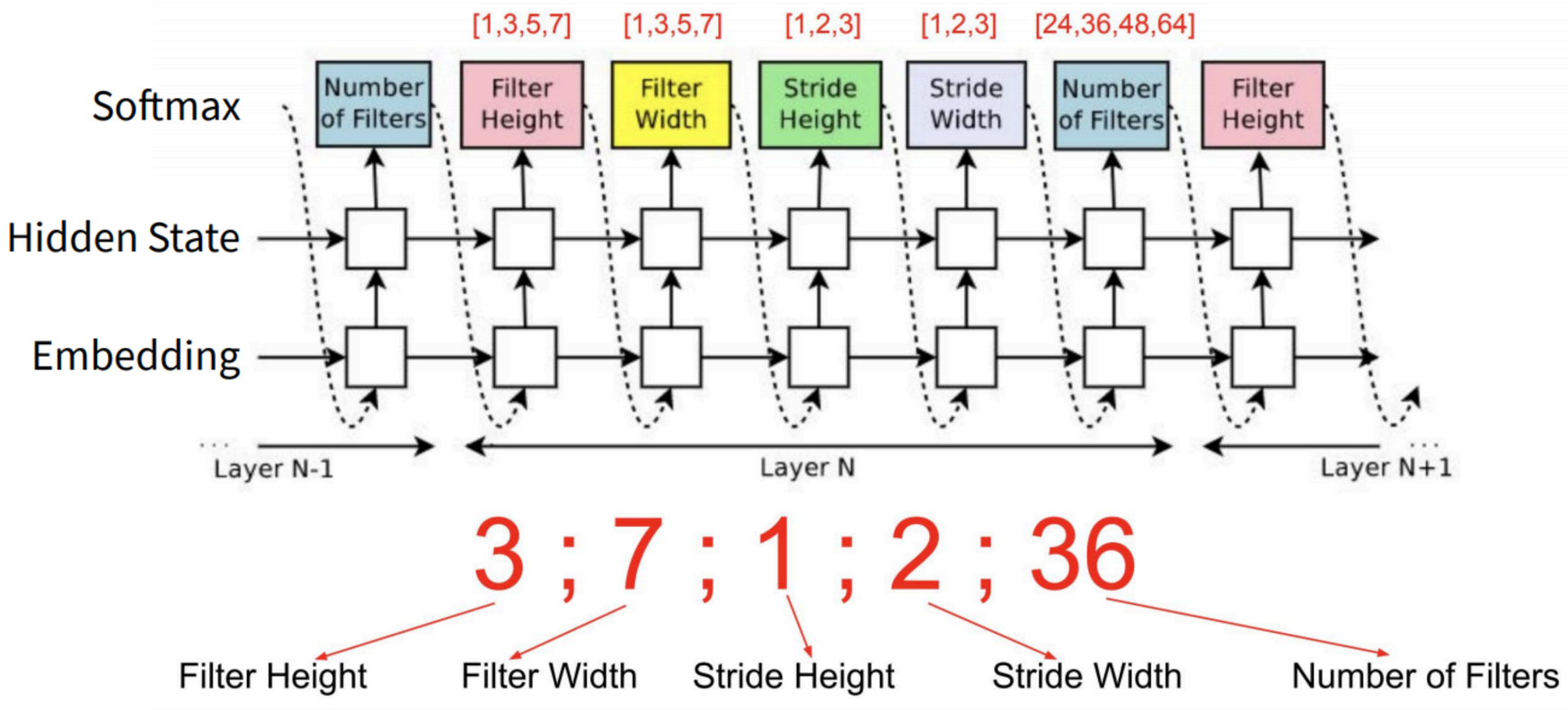
[Zoph and Quoc] Neural Architecture Search with Reinforcement Learning. ICLR, 2017.
[Baker, Gupta, Naik, Raskar] Designing Neural Network Architectures using Reinforcement Learning. ICLR, 2017.

- Successful results, but need hundreds of GPU days

Architecture	Test Error (%)	Search Cost (GPU days)	Search Method
ResNet (He et al., 2016)	4.62	-	manual
DenseNet-BC (Huang et al., 2017)	3.46	-	manual
NAS-RL (Zoph & Le, 2017)	3.65	22,400	RL

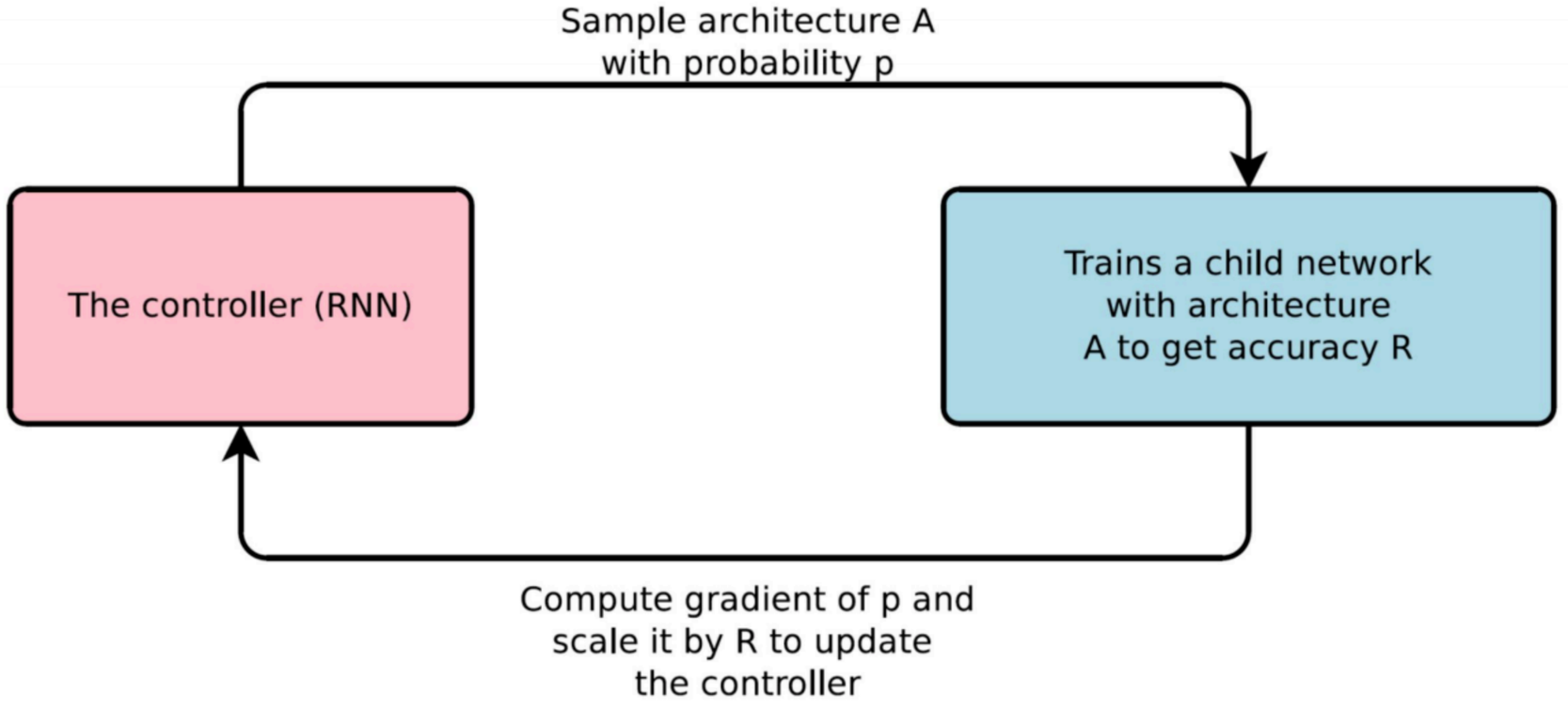
Neural Architecture Search

NAS with Reinforcement Learning



Neural Architecture Search

Training RNN controller by RL

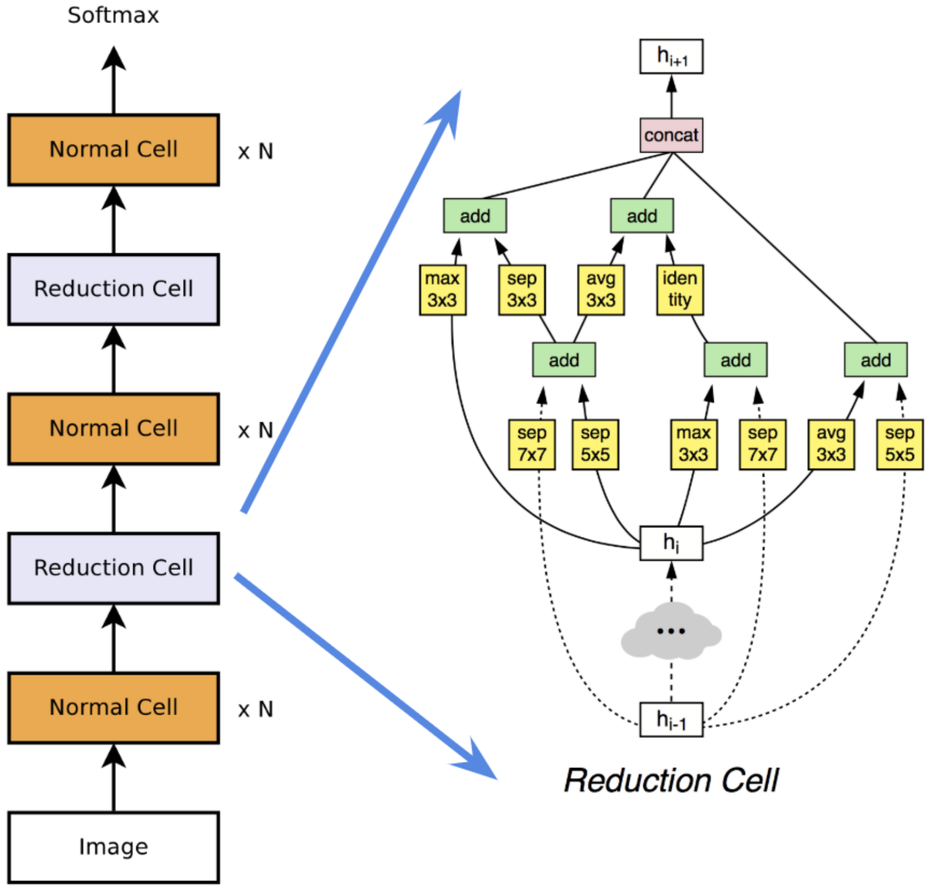


Extremely slow (>20,000 GPU days)

Neural Architecture Search

Cell-based Search Space (NASNet)

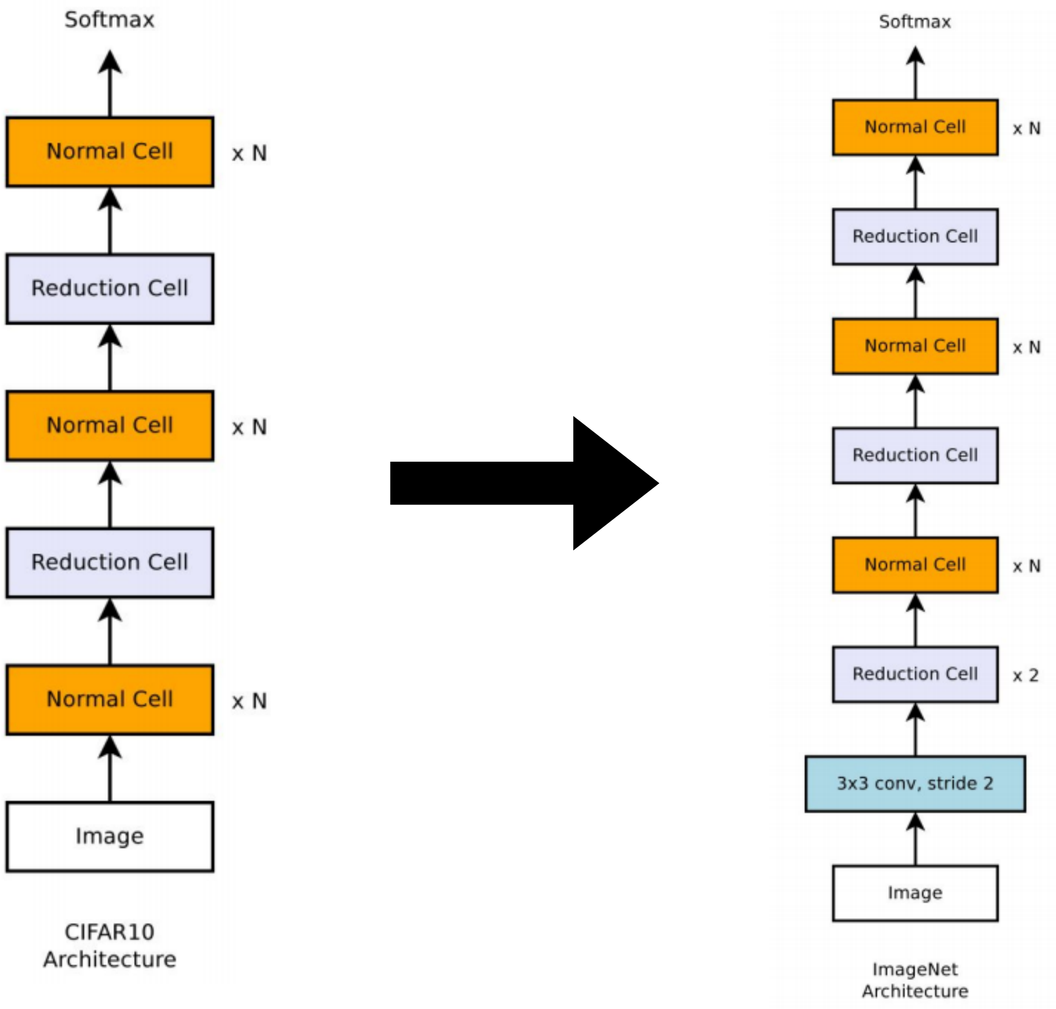
- Direct search on the global space:
  - Expensive; can't transfer to other datasets
- Cell-based search space:
  - Repeated cells (like ResNet)
  - Can use less blocks in searching
  - Can generalize to more complex datasets by stacking more blocks
- Compared with (Zoph & Le, 2017):
  - Error: 3.65 -> 2.65
  - Search cost: 22,400 -> 2000 GPU days



[Zoph, Vasudevan, Shlens, Le] Learning Transferable Architectures for Scalable Image Recognition. In CVPR, 2018.

Neural Architecture Search

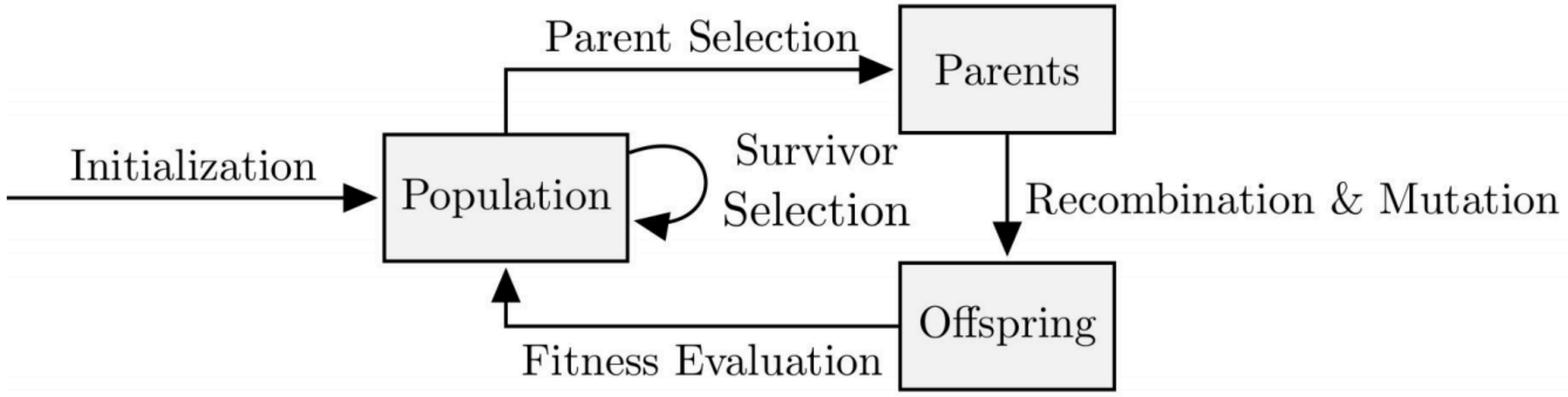
Generalize from CIFAR-10 to ImageNet



Neural Architecture Search

Evolutionary Algorithm

- Evolutionary algorithm also becomes possible with this search space



[Real, Aggarwak, Huang, Le] Regularized Evolution for Image Classifier Architecture Search. AAAI, 2019.

Neural Architecture Search

Other RL or evolutionary algorithms proposed

Reference	Error (%)	Params (Millions)	GPU Days	
Baker et al. (2017)	6.92	11.18	100	Reinforcement Learning
Zoph and Le (2017)	3.65	37.4	22,400	
Cai et al. (2018a)	4.23	23.4	10	
Zoph et al. (2018)	3.41	3.3	2,000	
Zoph et al. (2018) + Cutout	2.65	3.3	2,000	
Zhong et al. (2018)	3.54	39.8	96	
Cai et al. (2018b)	2.99	5.7	200	
Cai et al. (2018b) + Cutout	2.49	5.7	200	
Real et al. (2017)	5.40	5.4	2,600	Evolution
Xie and Yuille (2017)	5.39	N/A	17	
Suganuma et al. (2017)	5.98	1.7	14.9	
Liu et al. (2018b)	3.75	15.7	300	
Real et al. (2019)	3.34	3.2	3,150	

Designing competitive networks can take hundreds of GPU-days!
How to make neural architecture search more efficient?

Search typically takes **hundreds of GPU days**! Impractical for typical users.

Neural Architecture Search

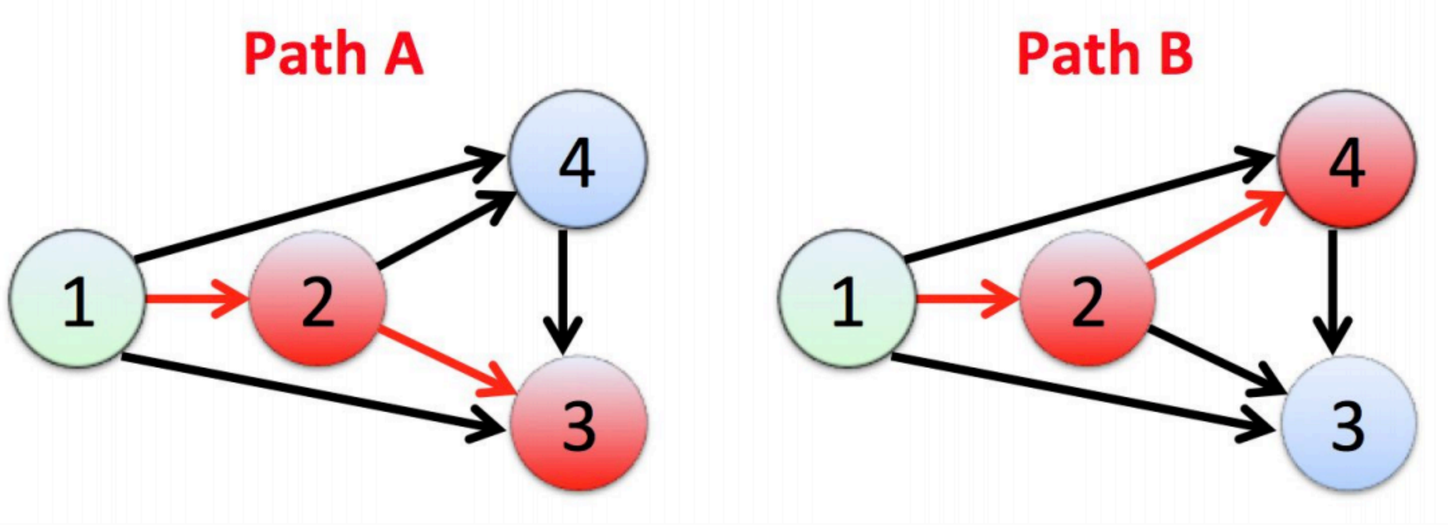
Significantly reduced search time since 2018

Architecture	Test Error (%)	Search Cost (GPU days)	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	-	manual
NAS-RL (Zoph & Le, 2017)	3.65	22,400	RL
NASNet-A (Zoph et al., 2018)	2.65	2000	RL
BlockQNN (Zhong et al., 2018)	3.54	96	RL
AmoebaNet (Real et al., 2019)	3.34 ± 0.06	3150	evolution
Hierarchical GA (Liu et al., 2018)	3.75	300	evolution
GCP (Suganuma et al., 2017)	5.98	15	evolution
DARTS (1st) (Liu et al., 2019)	3.00 ± 0.14	0.4	differentiable
DARTS (2nd) (Liu et al., 2019)	2.76 ± 0.09	1.0	differentiable
SNAS (moderate) (Xie et al., 2019)	2.85 ± 0.02	1.5	differentiable
GDAS (Dong & Yang, 2019)	2.93	0.3	differentiable
ProxylessNAS (Cai et al., 2019)†	2.08	4.0	differentiable
PC-DARTS (Xu et al., 2020)	2.57 ± 0.07	0.1	differentiable
NASP (Yao et al., 2019)	2.83 ± 0.09	0.1	differentiable
SDARTS-ADV (Chen & Hsieh, 2020)	2.61 ± 0.02	1.3	differentiable
DrNAS (Chen et al., 2019)	2.46 ± 0.03	0.6‡	differentiable
DARTS+PT (Wang et al., 2020)	2.61 ± 0.08	0.8	differentiable

Can run on a single GPU machine!

Neural Architecture Search

Concept of Weight Sharing

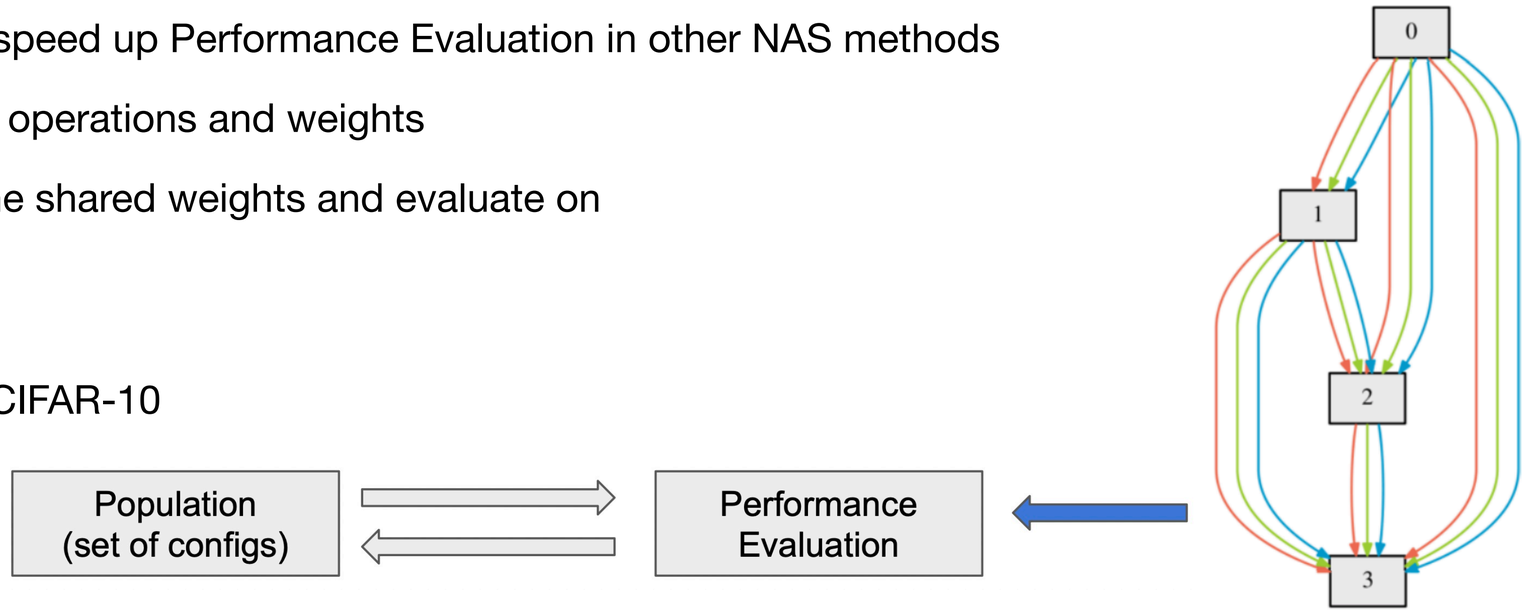


- Models defined by Path A and Path B should be trained separately
- Can we assume Path A and Path B share the same weight at 1->2?
  - Weight Sharing!
  - Avoid retraining for each new architecture

Neural Architecture Search

Concept of Weight Sharing

- Supernet: ensemble of many architectures
- All the architectures share the same w (weight sharing)
- Weight sharing can be directly used to speed up Performance Evaluation in other NAS methods
  - Train a “supernet” containing all the operations and weights
  - For any architecture, directly take the shared weights and evaluate on validation set
  - ENAS: weight sharing + RL
    - 0.5 GPU days with 2.9 error on CIFAR-10

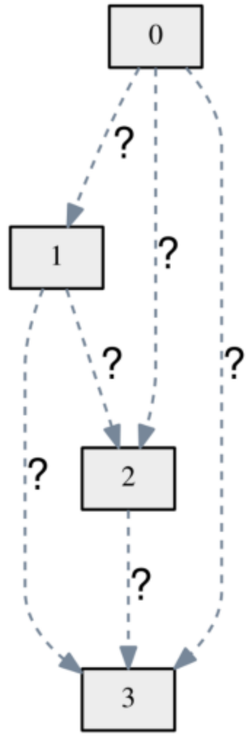


[Pham, Guan, Zoph, Le, Dean] Efficient Neural Architecture Search via Parameter Sharing. ICML, 2018.

Differentiable NAS

Can we directly obtain the final architecture through supernet training?

- Each edge is chosen from a pool of operations:
- Conv3x3, Conv5x5, Conv7x7, skip_connect, max_pool, avg_pool, zero, noise, ...
- One operation per edge => a discrete problem



(a)

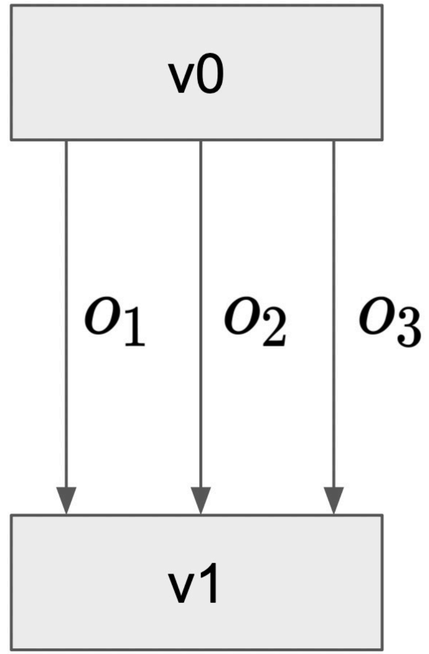
Differentiable NAS

Continuous Relaxation

- For simplicity, assume 3 operations
  o_1 : Conv3×3,o_2 : skip connect, o_3 : $Zero$
- Assume each edge is a mixed of three operations:

$$v_1 = \alpha_1 o_1(v_0) + \alpha_2 o_2(v_0) + \alpha_3 o_3(v_0)$$

Weight of each operation



[Liu, Simonyan, Yang] DARTS: Differentiable Architecture Search. In ICLR 2019.

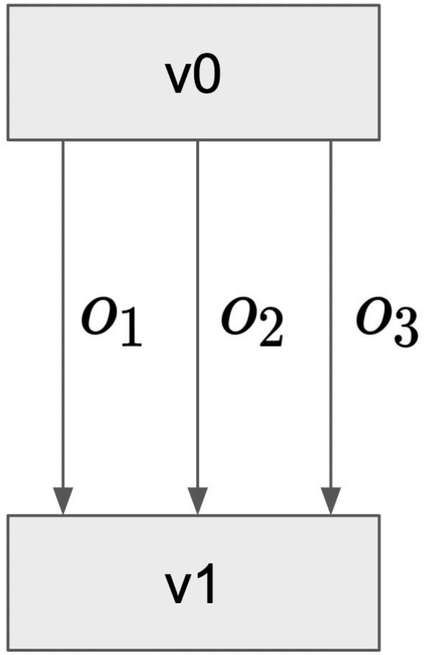
Differentiable NAS

Continuous Relaxation

- For simplicity, assume 3 operations
  o_1 : Conv3×3,o_2 : skip connect, o_3 : $Zero$
- Assume each edge is a mixed of three operations:

$$v_1 = \alpha_1 o_1(v_0) + \alpha_2 o_2(v_0) + \alpha_3 o_3(v_0)$$

Weight of each operation



- Can use softmax to ensure the weights form a prob. distribution

$$v_{\text{out}} = \sum_o \frac{\exp \alpha_o}{\sum_{o'} \exp \alpha_{o'}} o(v_{\text{in}})$$

[Liu, Simonyan, Yang] DARTS: Differentiable Architecture Search. In ICLR 2019.

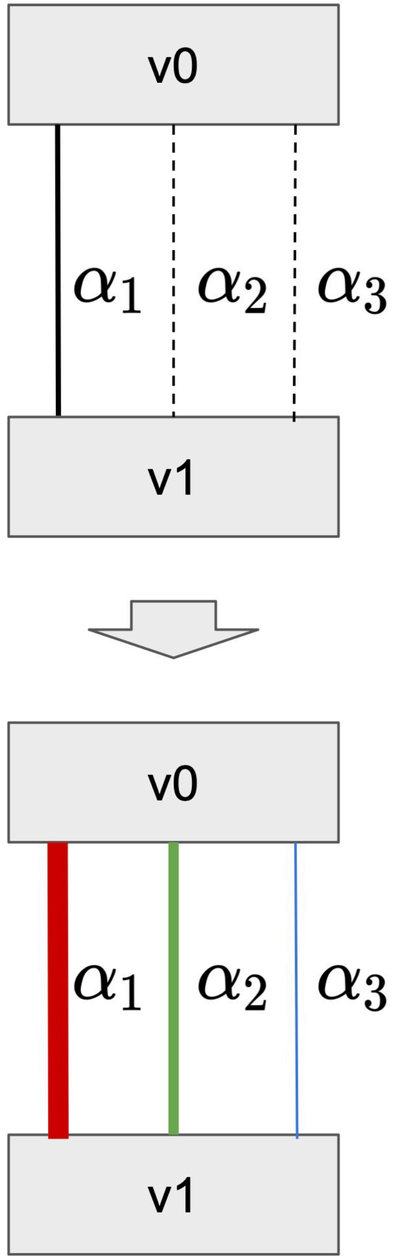
Differentiable NAS

Continuous Relaxation

- Final architecture: $[\alpha_1, \alpha_2, \alpha_3]$ is a one-hot vector
- Relax to continuous values in the search phase=> Bi-level optimization for finding α

$$\min_{\alpha} L_{\text{val}}(w^*(\alpha), \alpha)$$

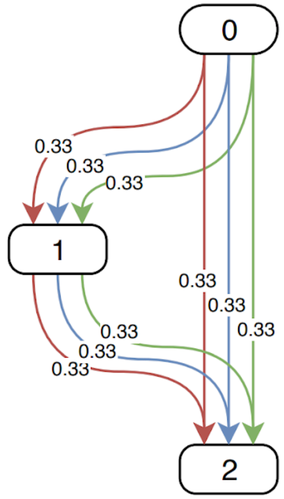
$$\text{s.t. } w^*(\alpha) = \arg\min_{w} L_{\text{train}}(w, \alpha)$$



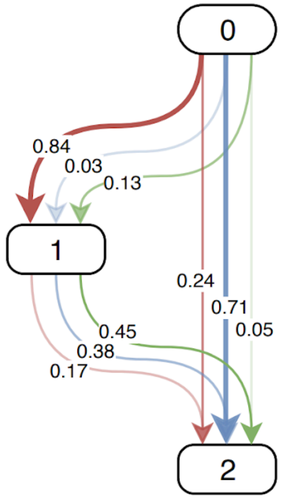
Differentiable NAS

Differentiable Neural Architecture Search (DARTS)

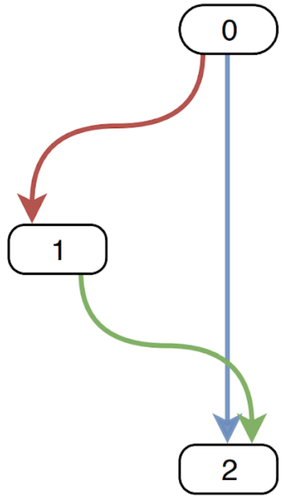
- Solve the bi-level optimization problem to obtain (α^*, w^*) (supernet)
- Use magnitude of α^* to choose the final architecture



(d) Search start



(e) Search end



(f) Final cell

Differentiable NAS

How to solve bi-level optimization?

$$\min_{\alpha} L_{\text{val}}(w^*(\alpha), \alpha)$$

$$\text{s.t. } w^*(\alpha) = \arg\min_{w} L_{\text{train}}(w, \alpha)$$

- Iteratively update w and α
- Update w:
  - Time consuming to compute w* exactly => approximate by one SGD step
    - $w' \leftarrow w - \eta \nabla_w L_{\text{train}}(w, \alpha)$
- Update α:
  - First order DARTS: assume w is constant w.r.t. α
    - $\alpha \leftarrow \alpha - c \nabla_\alpha L_{\text{val}}(w', \alpha)$

Differentiable NAS

Complexity of DARTS

- Time complexity: training the supernet only once
  - Supernet is a network with K operations with each edge
    => only K times slower than standard training
  - Usually good enough
- Memory complexity (GPU memory):
  - Backprop on all the operations on each edge
    => K times memory consumption
  - Prohibits for many problems

Differentiable NAS

Performance on CIFAR-10

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	#ops	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
Random search baseline‡ + cutout	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

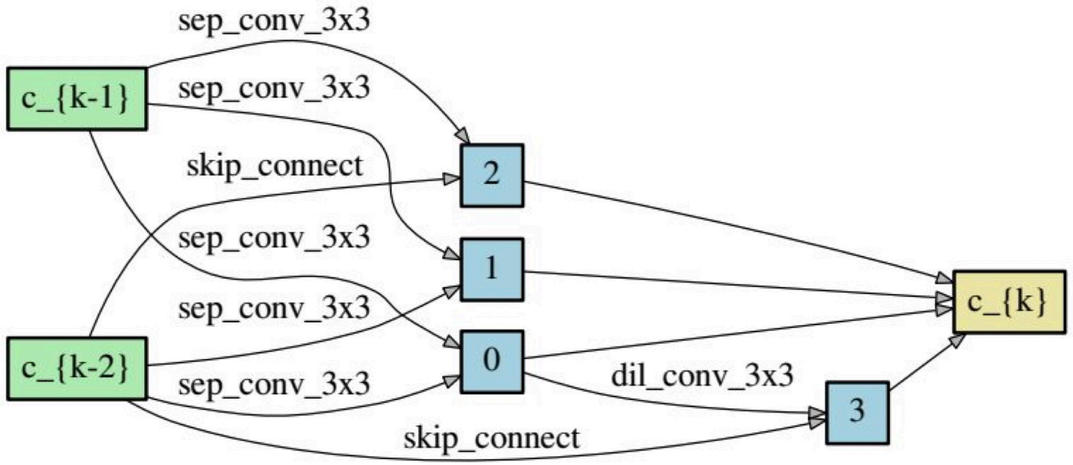
Differentiable NAS

Transfer to ImageNet (mobile setting)

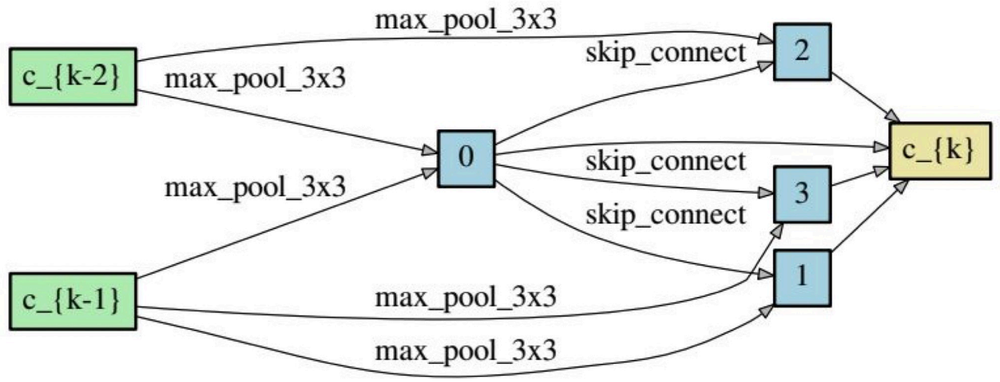
Architecture	Test Error (%)		Params	+×	Search Cost	Search
	top-1	top-5	(M)	(M)	(GPU days)	Method
Inception-v1 (Szegedy et al., 2015)	30.2	10.1	6.6	1448	–	manual
MobileNet (Howard et al., 2017)	29.4	10.5	4.2	569	–	manual
ShuffleNet 2× ($g = 3$) (Zhang et al., 2017)	26.3	–	~5	524	–	manual
NASNet-A (Zoph et al., 2018)	26.0	8.4	5.3	564	2000	RL
NASNet-B (Zoph et al., 2018)	27.2	8.7	5.3	488	2000	RL
NASNet-C (Zoph et al., 2018)	27.5	9.0	4.9	558	2000	RL
AmoebaNet-A (Real et al., 2018)	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B (Real et al., 2018)	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C (Real et al., 2018)	24.3	7.6	6.4	570	3150	evolution
PNAS (Liu et al., 2018a)	25.8	8.1	5.1	588	~225	SMBO
DARTS (searched on CIFAR-10)	26.7	8.7	4.7	574	4	gradient-based

Differentiable NAS

Architectures found by DARTS



Normal cells on CIFAR-10



Reduction cells on CIFAR-10