A Tutorial on Neural Architecture Search

IEEE ICDM 2019 Tutorials on Automated Deep Learning:
Theory, Algorithms, Platforms, and Applications

Nov. 09, 2019

Speaker: Dr. Siyu Huang

Big Data Lab, Baidu Research, Beijing

Siyu Huang (黄思羽)

Research Scientist at Big Data Lab, Baidu Research

Research Interests: Neural Architecture Search, Generative Model, Multi-Modal Analysis

Ph.D. at College of Information Science and Electronic Engineering, Zhejiang University, China
Sept 2014 - July 2019
Supervised by Prof. Zhongfei (Mark) Zhang and Prof. Xi Li

Visiting Scholar at Language Technologies Institute, School of Computer Science, Carnegie Mellon University, USA
Jan 2018 - Jan 2019
Supervised by Prof. Alexander G. Hauptmann.

Bachelor at College of Information Science and Electronic Engineering, Zhejiang University, China
Sept 2010 - June 2014
Chu Kochen Honors College

Goal of This Tutorial: A Brief Overview of NAS

- **AutoDL and NAS**
- **NAS with RL**
  - NAS with RL
  - NAS-Net
  - Weight sharing mechanism
  - *Our work: Adaptive cross-domain NAS*
  - *Our work: AutoGAN*
- **Differentiable NAS**
  - Architecture embedding method
  - Continuous relaxation of discrete architecture
  - *Our work: NAS-based state-of-the-art image classification model*
  - *Our work: Resource-customized NAS*
- **Greedy NAS**
  - *Our work: NAS for tree-topology network*

Automated Deep Learning (AutoDL)

- **AutoDL: Automation of DL Pipelines**
  - Accelerate DL research and industrial applications
  - Benefit communities other than DL professionals Efficiently

- **AutoDL includes**
  - Automated data preprocessing and cleansing
  - Automated model selection
  - Automated hyperparameter optimization
  - **Neural architecture search**
  - ……

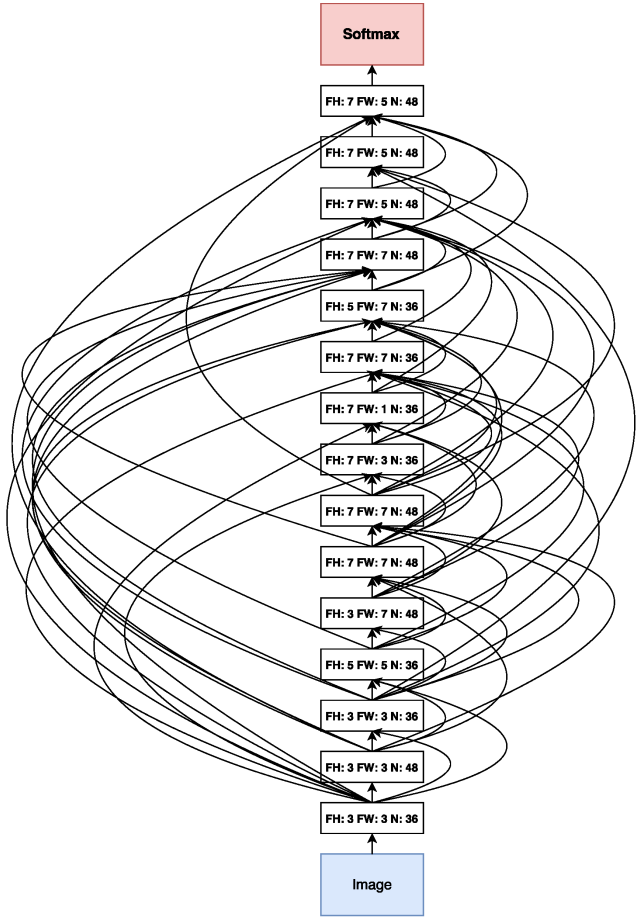
Baidu 大脑 | EasyDL

Google Cloud
CLOUD AUTOML

AUTO KERAS

Neural Architecture Search (NAS)

- **NAS: The automated design of neural network architectures**
  - Expensive black-box optimization problem
- **Current mainstream NAS methods:**
  - Random/Grid Search
  - Bayesian Optimization (BO)
  - Evolutionary Algorithm (EA)
  - **Reinforcement Learning (RL)**
  - **Differentiable NAS**

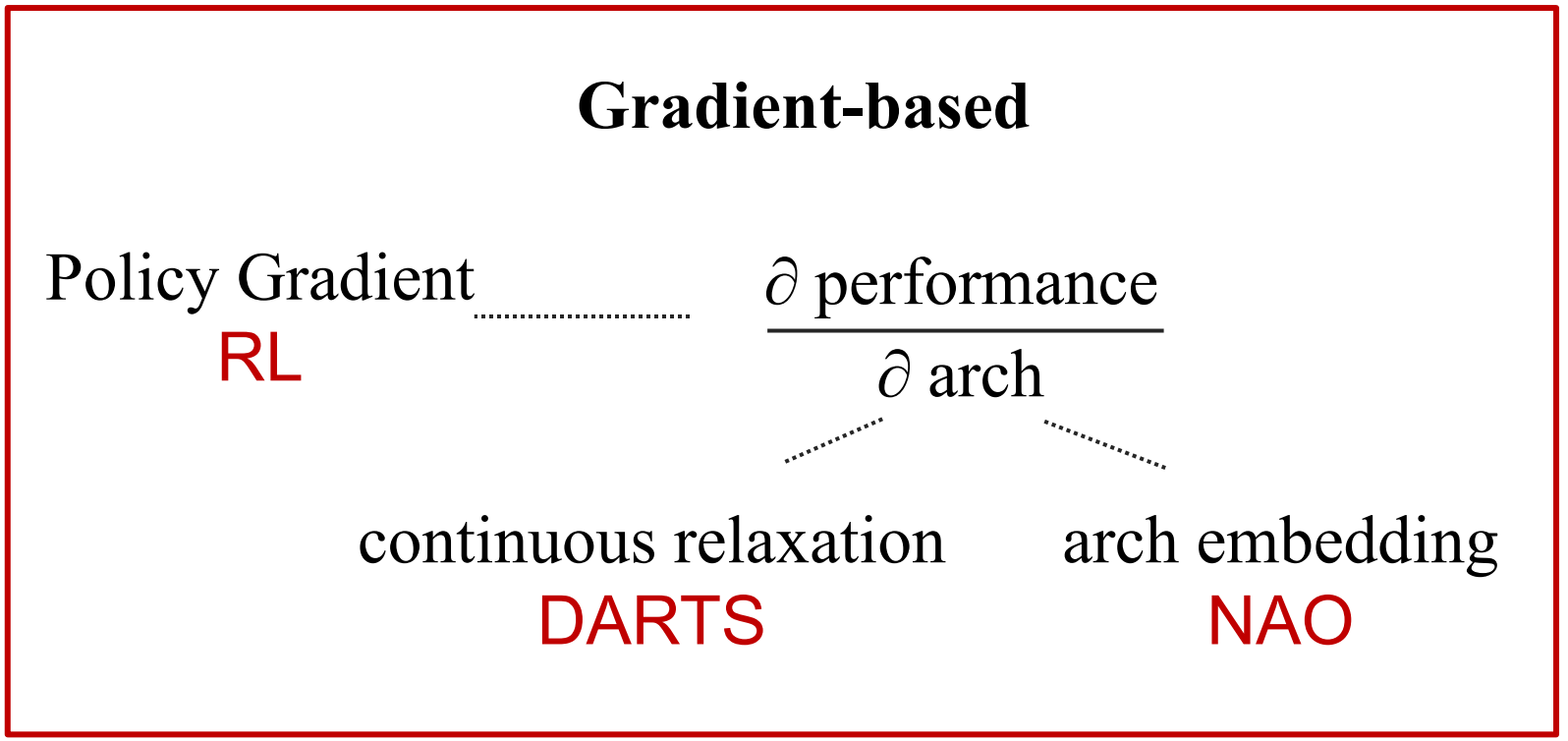


Zoph and Le, ICLR'17

Goal of This Tutorial: A Brief Overview of NAS

- **AutoDL and NAS**
- **NAS with RL**
  - NAS with RL
  - NAS-Net
  - Weight sharing mechanism
  - *Our work: Adaptive cross-domain NAS*
  - *Our work: AutoGAN*
- **Differentiable NAS**
  - Architecture embedding method
  - Continuous relaxation of discrete architecture
  - *Our work: NAS-based state-of-the-art image classification model*
  - *Our work: Resource-customized NAS*
- **Greedy NAS**
  - *Our work: NAS for tree-topology network*

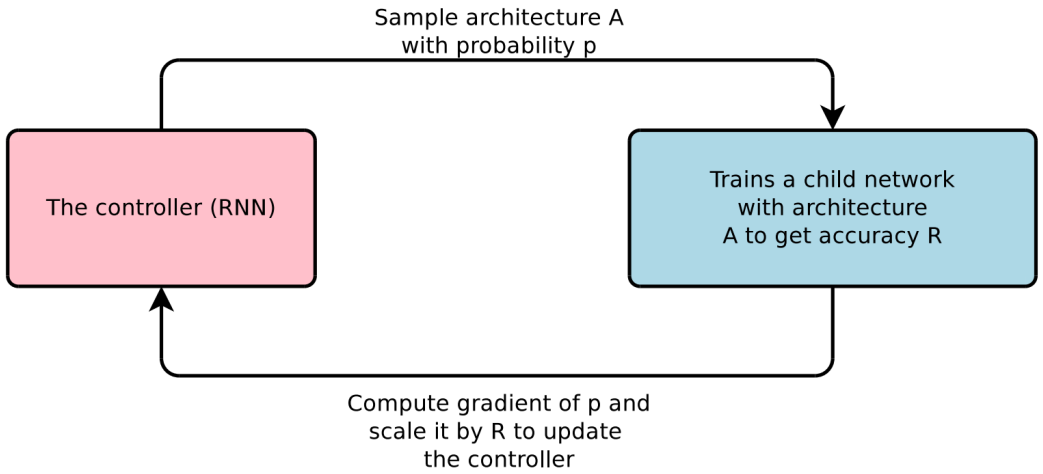
NAS with RL & Differentiable NAS



NAS with RL

- Basic Procedure

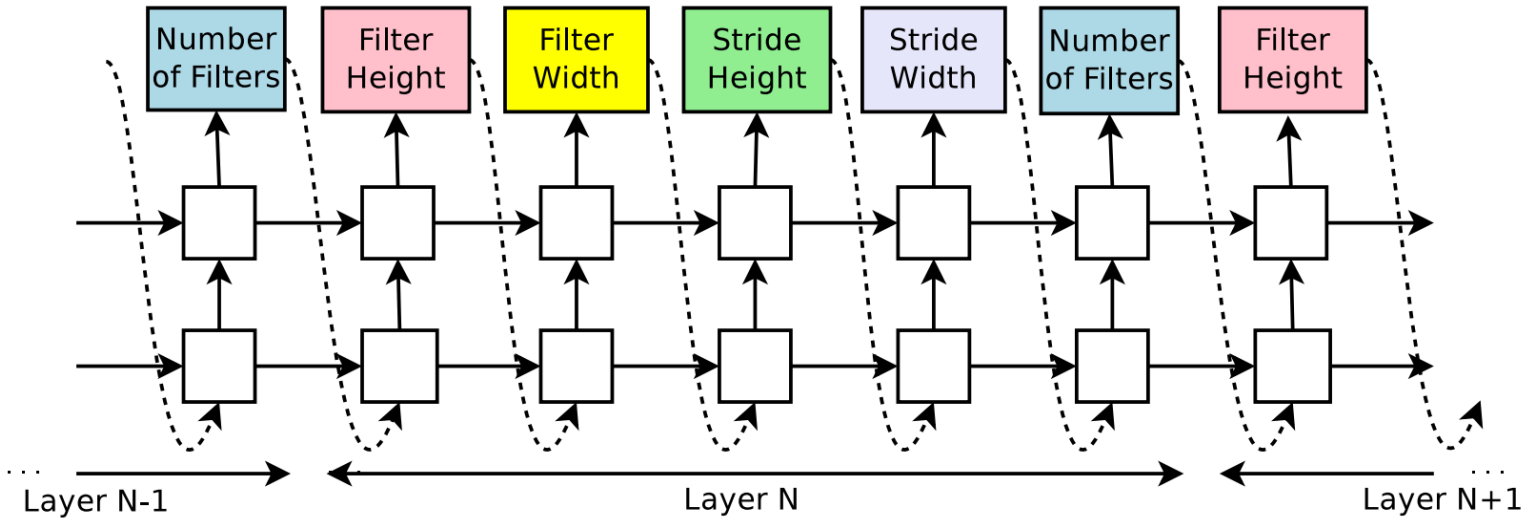
1. Sample candidate architectures with a Controller.
2. Train candidate architectures on a specific task and evaluate the performances on validation set as rewards for the candidates.
3. Update Controller based on rewards.



Zoph and Le, ICLR'17

NAS with RL

- Controller



Controller (RNN)

Zoph and Le, ICLR'17

NAS with RL

- The Optimization of Controller

Objective： Maximize architecture's validation performance *R*.

$$J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R]$$

NAS with RL

- The Optimization of Controller

Objective： Maximize architecture's validation performance *R*.

$$J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R]$$

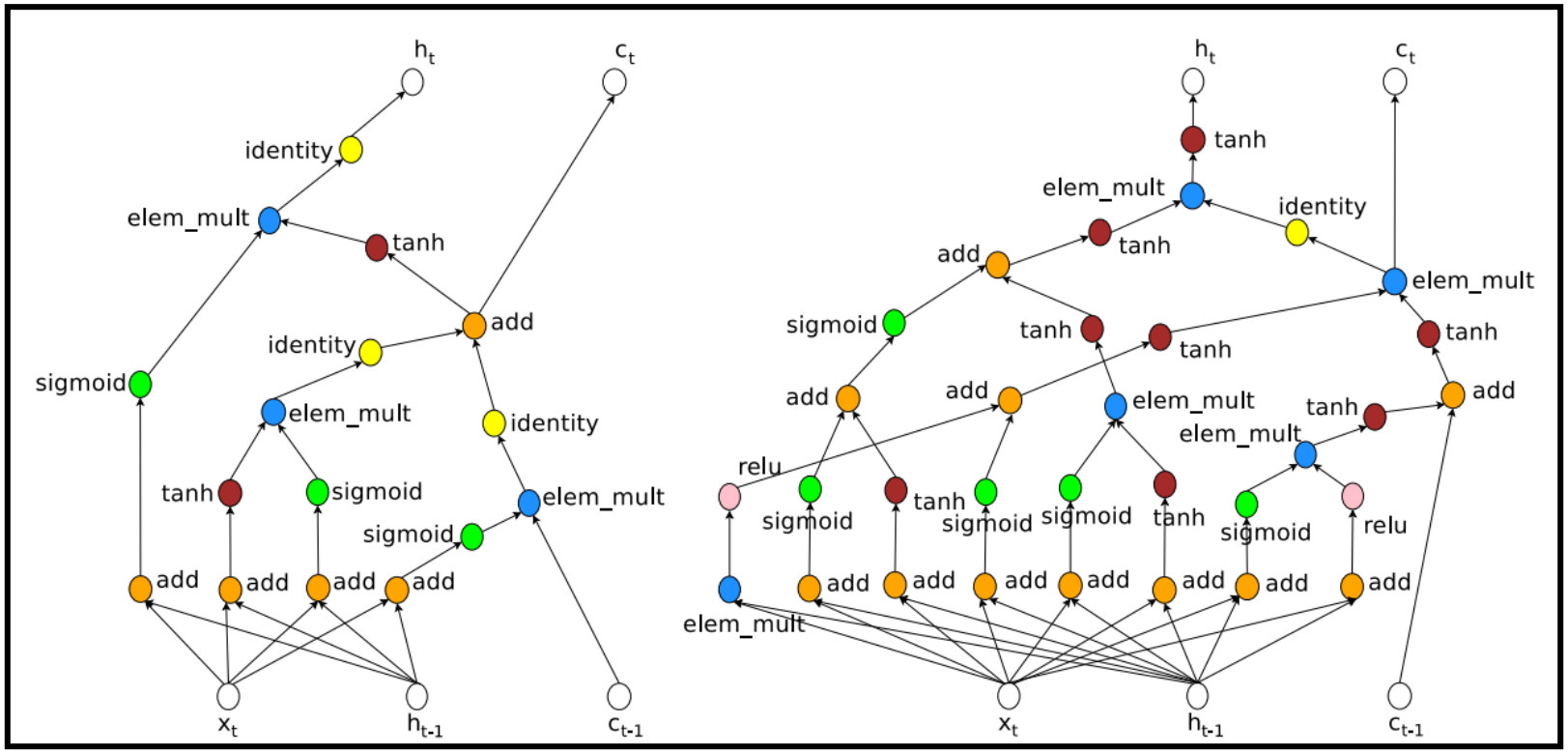
***R* is not differentiable w.r.t. archs. We introduce Policy Gradient (REINFORCE) to update Controller.**

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} E_{P(a_{1:T};\theta_c)}\left[\nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R\right]$$

$$\approx \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} \nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R_k$$

NAS with RL

- NAS outperforms handcrafted archs for the first time
- Language model (Penn Treebank dataset)



LSTM Cell NAS Cell

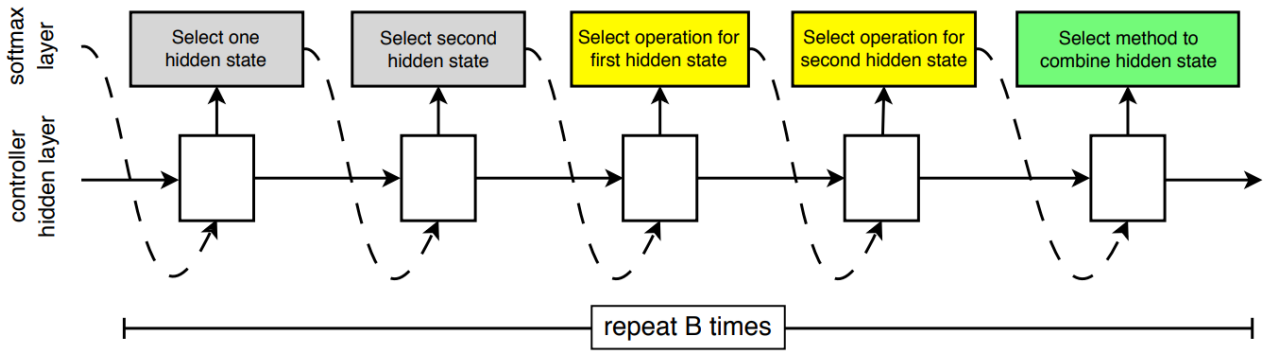
RNN Cells

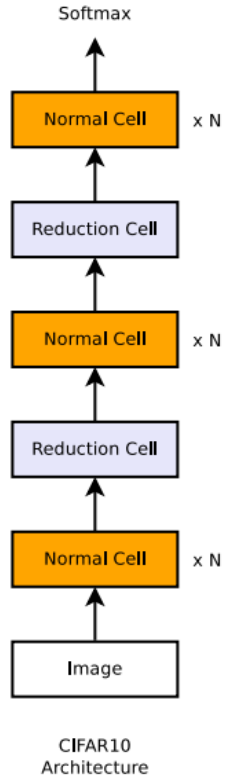
Zoph and Le, ICLR'17

NAS with RL

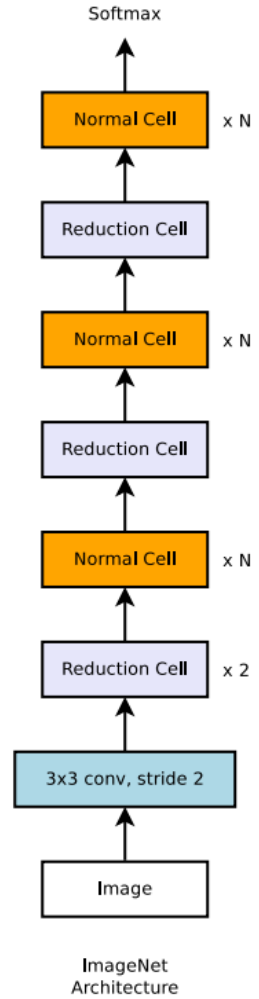
- NAS-Net (Zoph *et al.* CVPR'18)

- **Discovering Archs in a Convolutional Cell**
  - Introduce Normal Cell and Reduction Cell
  - **Normal Cell:** Same feature map size between input and output
  - **Reduction Cell:** Output has smaller feature map size but more filters





CIFAR10 Architecture



ImageNet Architecture

NAS with RL

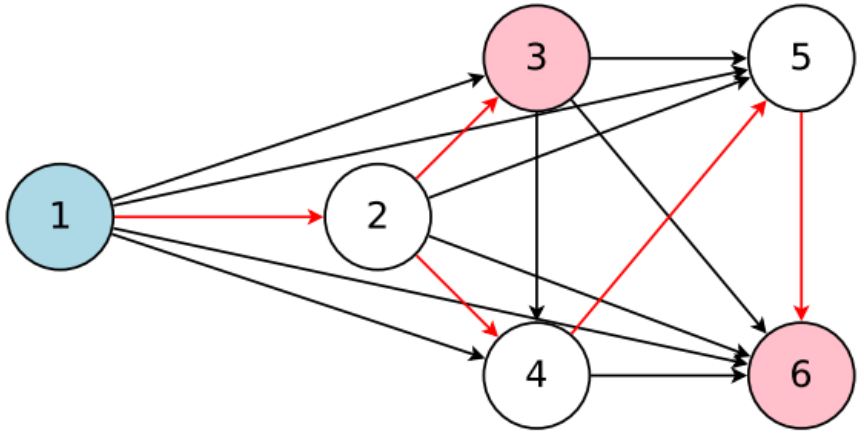
- Weight Sharing Mechanism (Pham *et al.* ICML'18)

Method	GPUs	Times (days)	Params (million)	Error (%)
Budgeted Super Nets (Veniat & Denoyer, 2017)	–	–	–	9.21
ConvFabrics (Saxena & Verbeek, 2016)	–	–	21.2	7.43
Macro NAS + Q-Learning (Baker et al., 2017a)	10	8-10	11.2	6.92
Net Transformation (Cai et al., 2018)	5	2	19.7	5.70
FractalNet (Larsson et al., 2017)	–	–	38.6	4.60
SMASH (Brock et al., 2018)	1	1.5	16.0	4.03
NAS (Zoph & Le, 2017)	800	21-28	7.1	4.47
NAS + more filters (Zoph & Le, 2017)	800	21-28	37.4	**3.65**

Huge computing cost

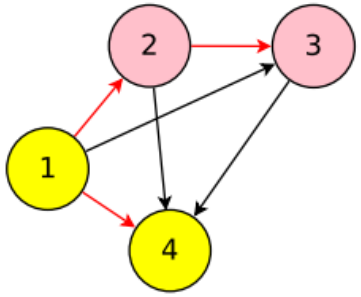
Weight Sharing Mechanism

- All candidates belong to one hyper-graph (DAG)
- Build candidates by activating paths of hyper-graph
- Neural weights shared between candidates

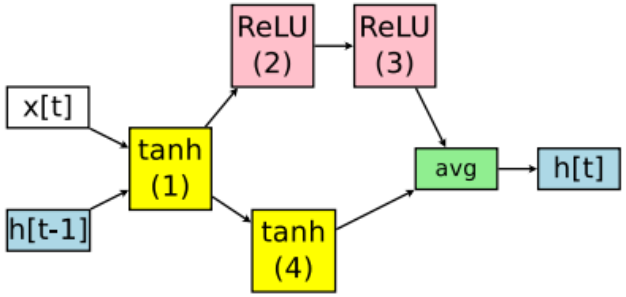


NAS with RL

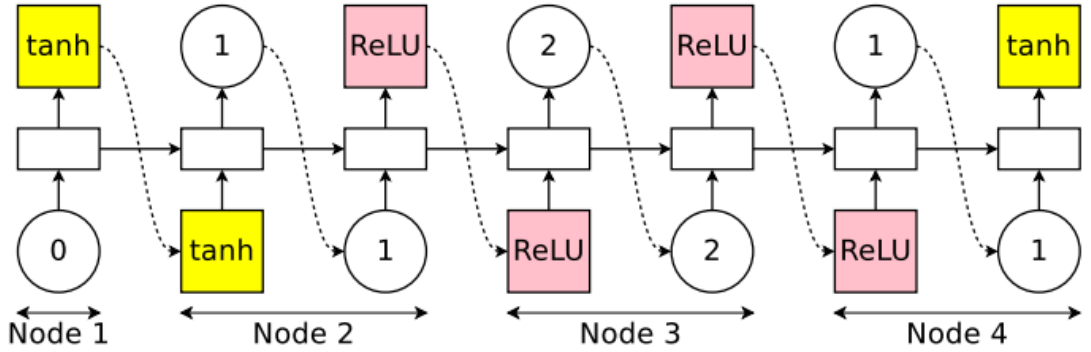
- Weight Sharing Mechanism (Pham *et al.* ICML'18)



hyper-graph (DAG)



RNN cell



sampling by Controller

low computing cost in practice

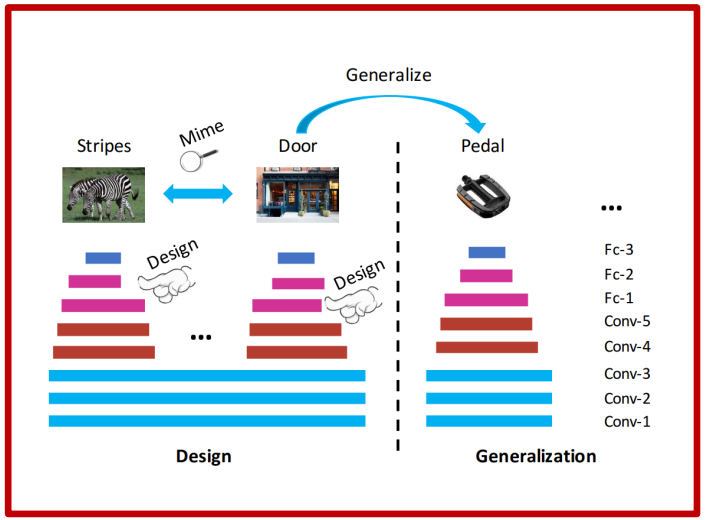
Method	GPUs	Times (days)	Params (million)	Error (%)
Hierarchical NAS (Liu et al., 2018)	200	1.5	61.3	3.63
Micro NAS + Q-Learning (Zhong et al., 2018)	32	3	–	3.60
Progressive NAS (Liu et al., 2017)	100	1.5	3.2	3.63
NASNet-A (Zoph et al., 2018)	450	3-4	3.3	3.41
NASNet-A + CutOut (Zoph et al., 2018)	450	3-4	3.3	**2.65**
ENAS + micro search space	1	0.45	4.6	3.54
ENAS + micro search space + CutOut	1	0.45	4.6	**2.89**

NAS with RL

- Our work: Adaptive Cross-Domain NAS (Cheng *et al.* ACM MM'18)

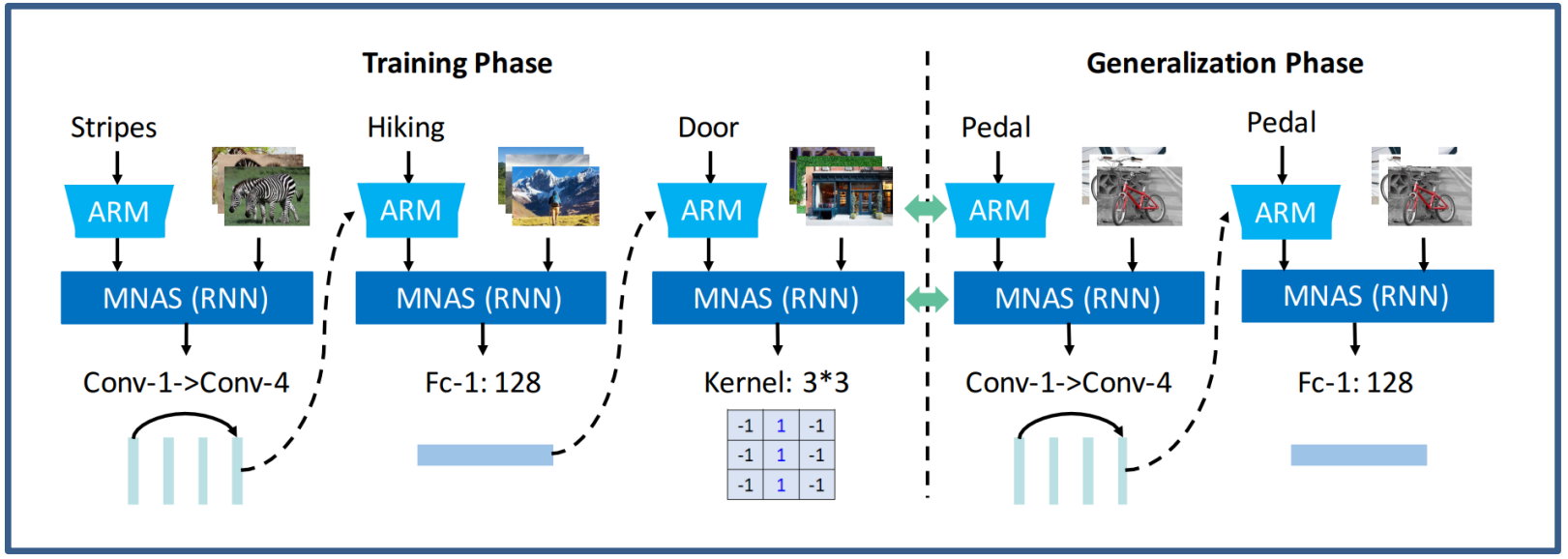
☐ New task: Generalize NAS strategies between domains

➢ Generalizable NAS



➢ Attribute guided NAS strategy

➢ Generalize strategy to unseen domains

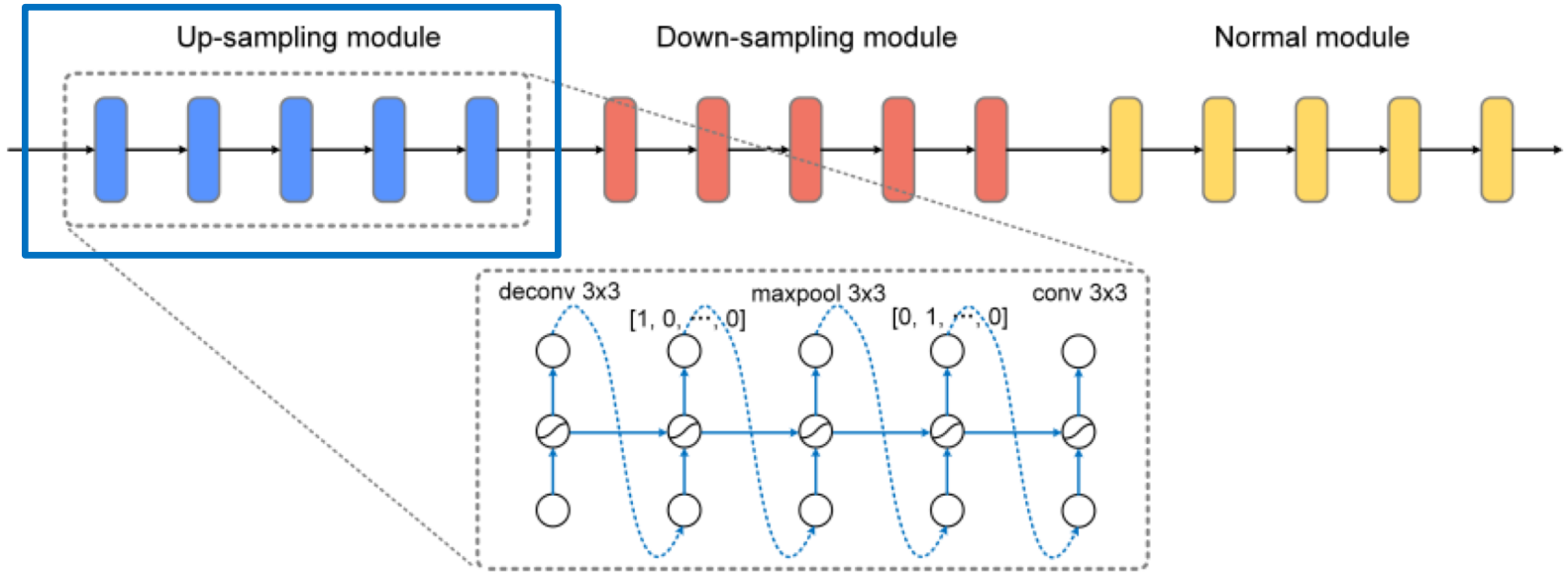


NAS with RL

- Our work: A-GAN (Tech Report)

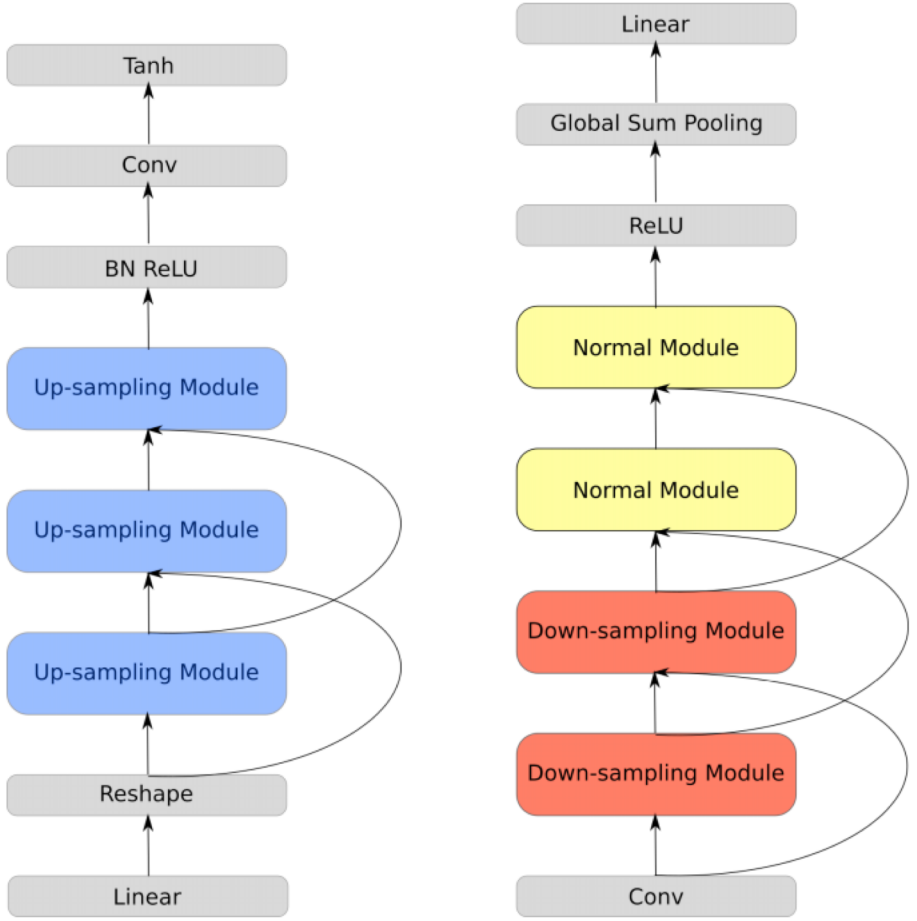
☐ Searching for Architecture of Generative Model

➢ Up-sampling module is rarely studied in NAS



NAS with RL

- Our work: A-GAN (Tech Report)



➢ Search Space

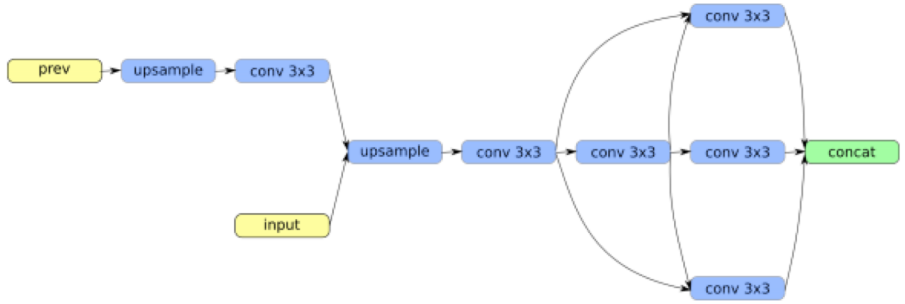
- identity
- 1×1 conv
- 3×3 conv
- 3×3 dilated conv
- 3×3 separable conv
- 5×5 separable conv
- 7×7 separable conv
- 1×3 then 3×1 conv
- 1×5 then 5×1 conv
- 1×7 then 7×1 conv
- 3×3 max pool
- 5×5 max pool
- 7×7 max pool
- 3×3 average pool
- 5×5 average pool
- 7×7 average pool

- 3×3, 5×5 or 7×7 transposed convolution.
- Nearest-neighbor interpolation followed by any convolution operation in the list of normal module.

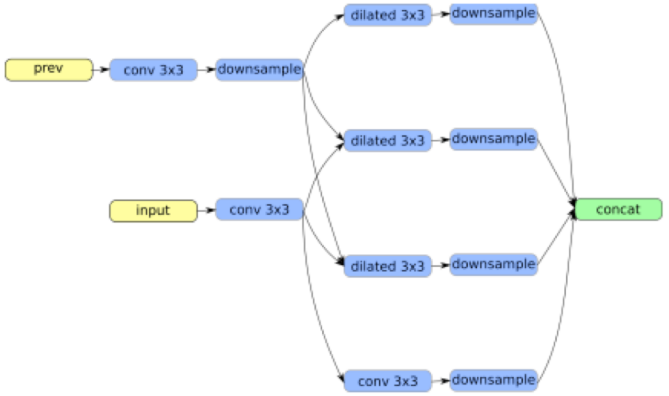
- A convolutional layer followed by an average pooling layer with a stride of 2.
- An average pooling layer with a stride of 2 followed by a convolutional layer.

NAS with RL

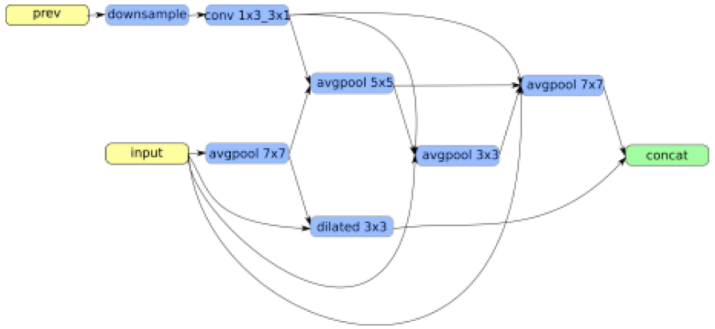
- Our work: **A-GAN** (Tech Report)



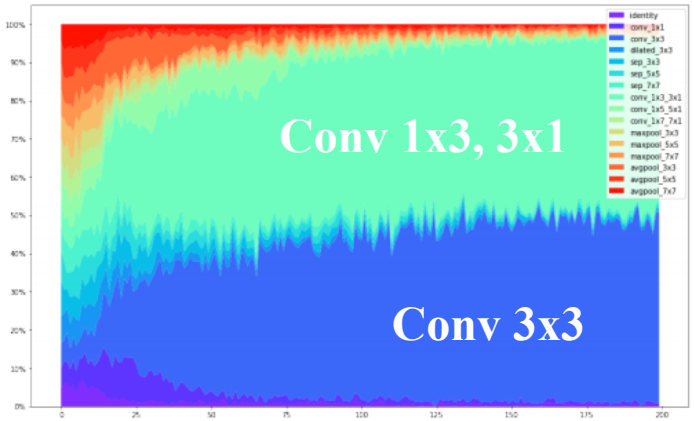
(a) Up-sampling module



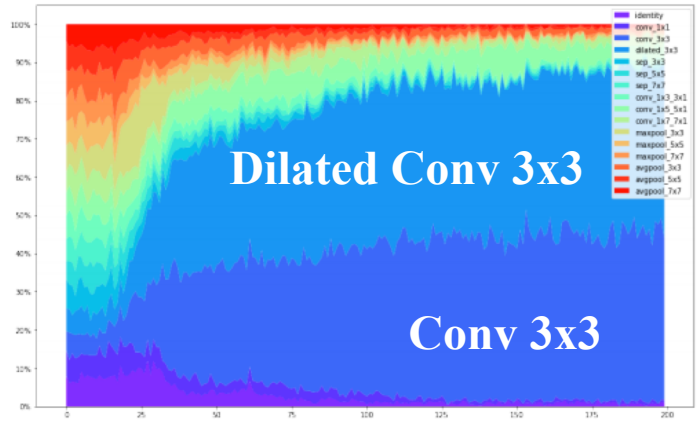
(b) Down-sampling module



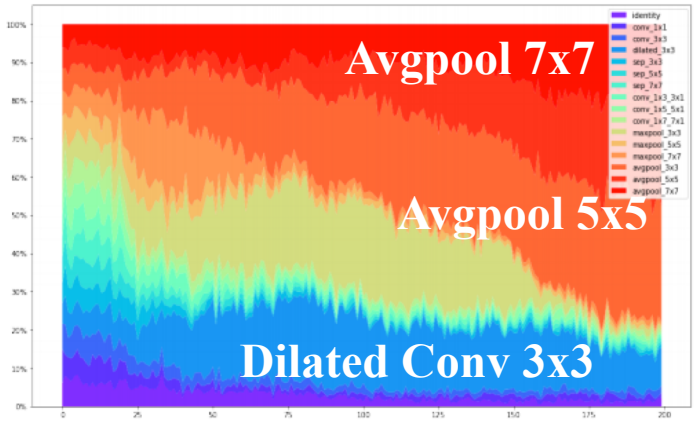
(c) Normal module



(a) Up-sampling



(b) Down-sampling

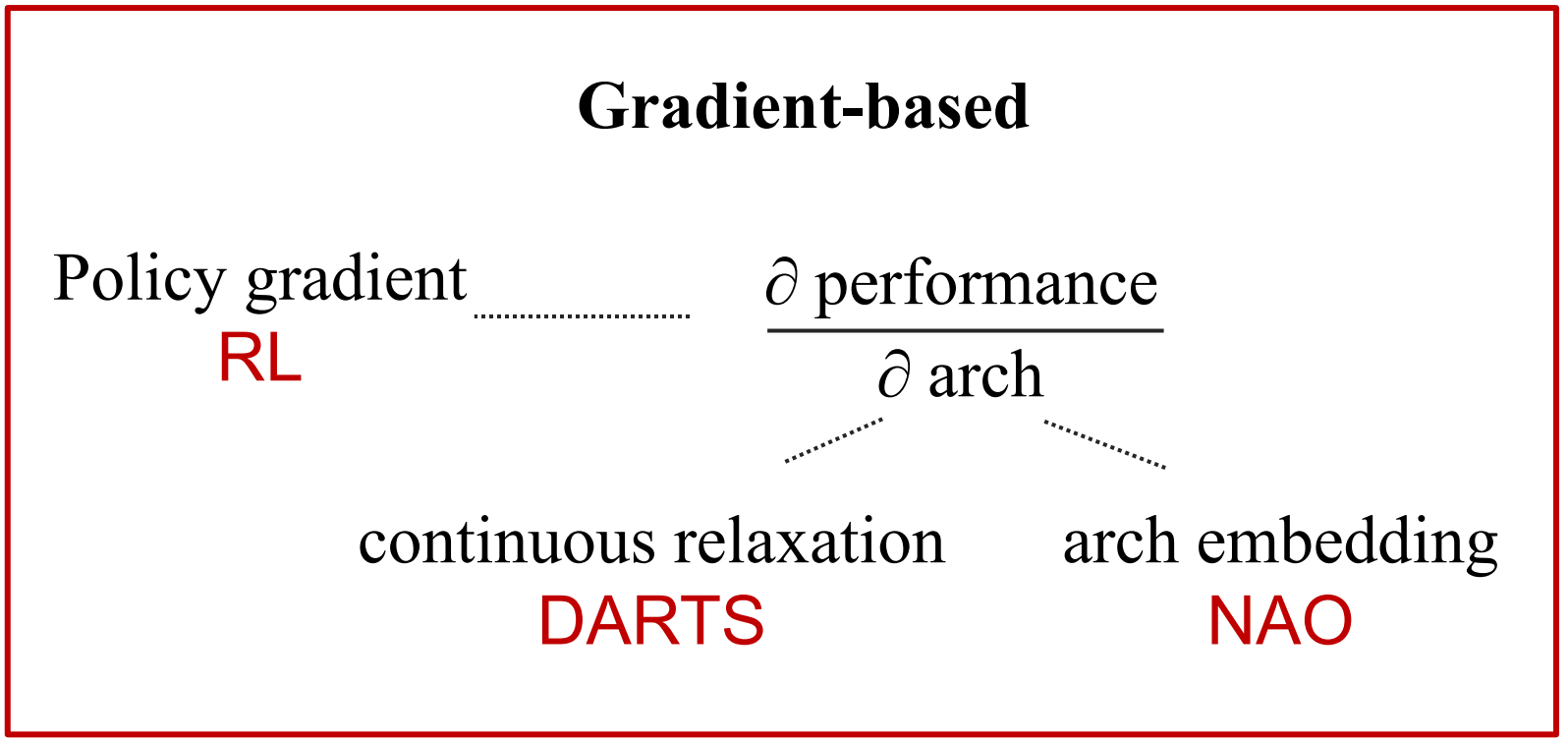


(c) Normal

Goal: A Brief Overview of NAS

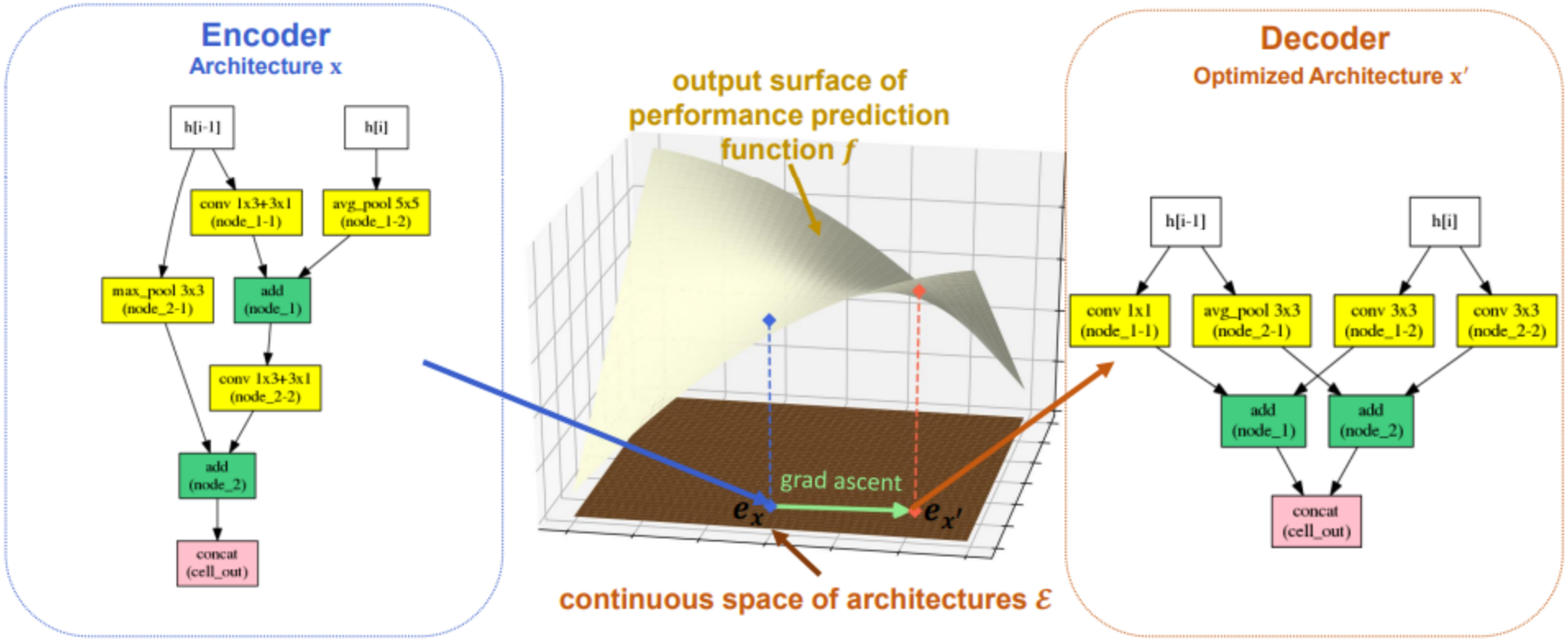
- **AutoDL and NAS**
- **NAS with RL**
  - NAS with RL
  - NAS-Net
  - Weight sharing mechanism
  - *Our work: Adaptive cross-domain NAS*
  - *Our work: AutoGAN*
- **Differentiable NAS**
  - Architecture embedding method
  - Continuous relaxation of discrete architecture
  - *Our work: NAS-based state-of-the-art image classification model*
  - *Our work: Resource-customized NAS*
- **Greedy NAS**
  - *Our work: NAS for tree-topology network*

Differentiable NAS



Differentiable NAS

- **Architecture Embedding** (Luo *et al.* NIPS'18)



$$h'_t = h_t + \eta \frac{\partial f}{\partial h_t}, \quad e_{x'} = \{h'_1, \cdots, h'_T\}$$

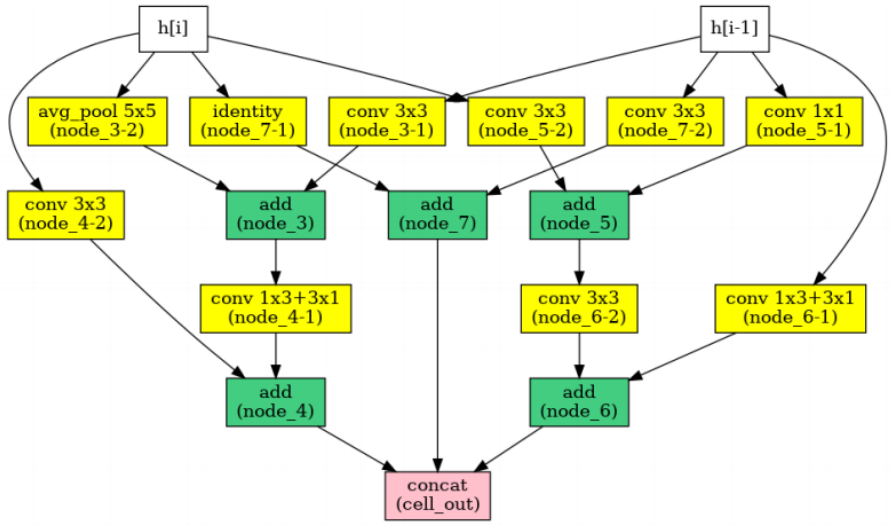
Differentiable NAS

- Architecture Embedding (Luo *et al.* NIPS'18)

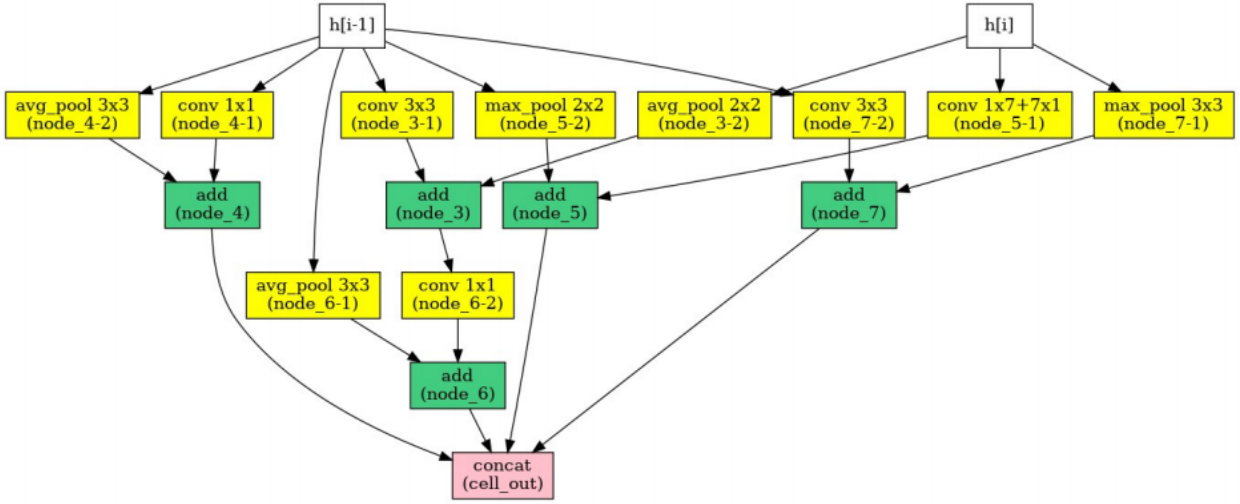
Performance prediction error of arch x

$$L = \lambda L_{pp} + (1-\lambda) L_{rec} = \lambda \sum_{x \in X} (s_x - f(E(x))^2 - (1-\lambda) \sum_{x \in X} \log P_D(x|E(x))$$

Reconstruction error of arch x



Normal Cell

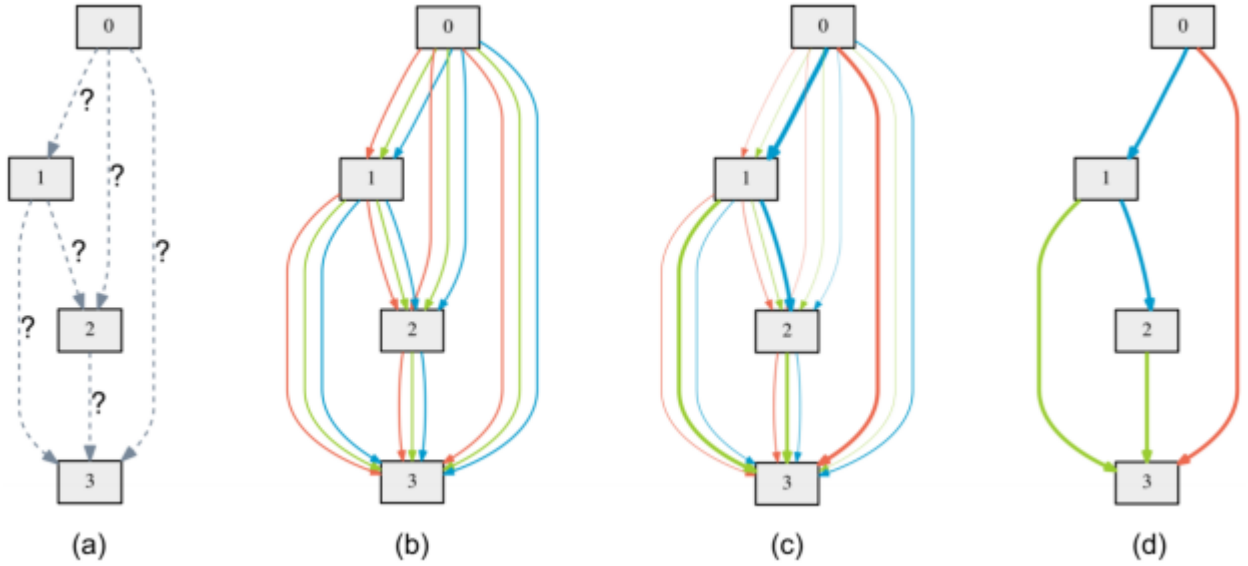


Reduction Cell

Differentiable NAS

- DARTS: Contionuous Relaxation of Architecture

(Liu *et al.* ICLR'19)



Continuous relaxation of discrete operations

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

Differentiable NAS

- DARTS (Liu *et al.* ICLR'19)

- DARTS Algorithm— Update arch and weights alternately
  - Update arch on validation set $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$
  - Update weights on training set $\nabla_w \mathcal{L}_{train}(w, \alpha)$

Low computing cost

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

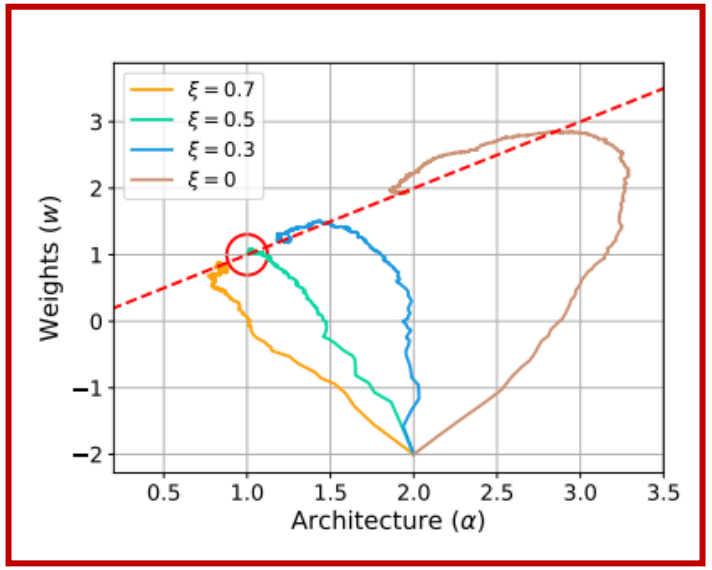
Liu et al. ICLR'19

Differentiable NAS

- Optimization in DARTS (Liu *et al.* ICLR'19)

1. Objective

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\text{s.t.} \quad w^*(\alpha) = \text{argmin}_w \ \mathcal{L}_{train}(w, \alpha)$$



2. One-step optimum of *w**

$$\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$$

3. Chain-Rule

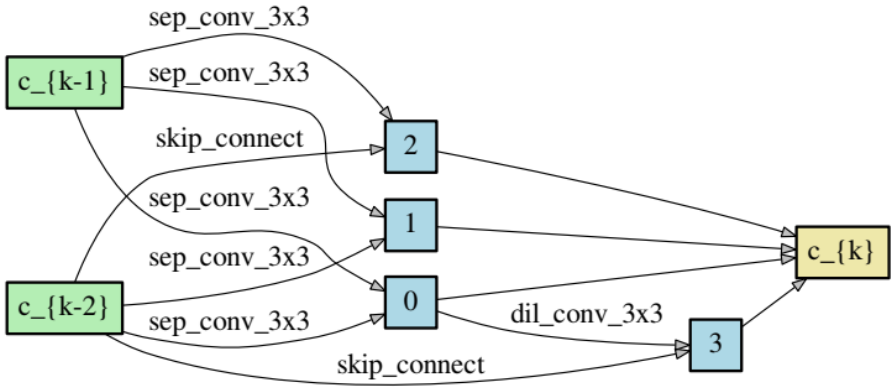
$$\nabla_\alpha \mathcal{L}_{val}(w', \alpha) - \xi \nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$$
$$w' = w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha)$$

4. Finite-Difference Approximation

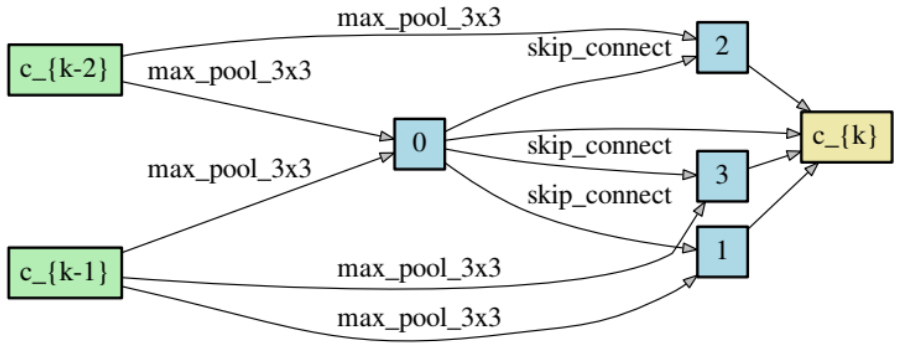
$$\nabla^2_{\alpha,w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}_{val}(w', \alpha) \approx \frac{\nabla_\alpha \mathcal{L}_{train}(w^+, \alpha) - \nabla_\alpha \mathcal{L}_{train}(w^-, \alpha)}{2\epsilon}$$
$$w^\pm = w \pm \epsilon \nabla_{w'} \mathcal{L}_{val}(w', \alpha)$$

Differentiable NAS

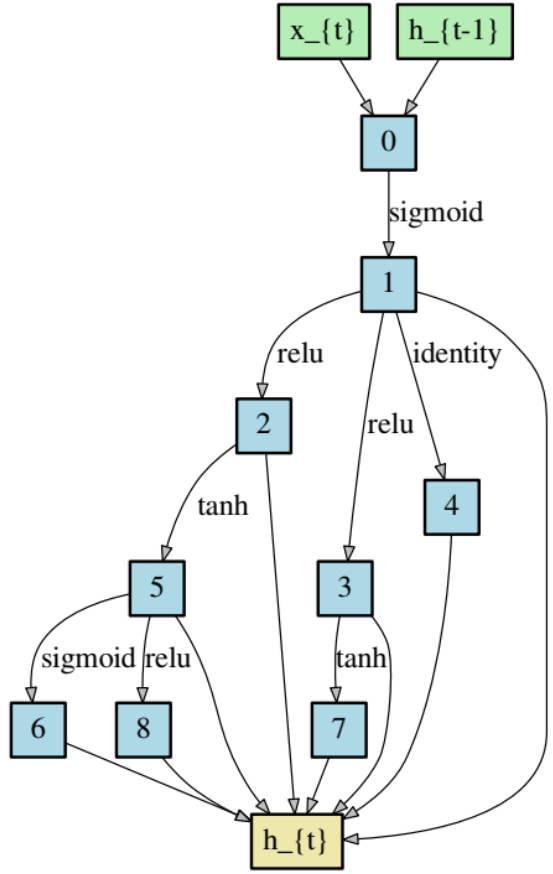
- Architectures Discovered by DARTS (Liu *et al.* ICLR'19)



Normal Cell



Reduction Cell



Recurrent Cell

Differentiable NAS

- Our Work: State-of-the-Art Cifar-10 Model (Tech Report)

- Based on Differentiable NAS
- 98.01% accuracy on CIFAR-10 (with normal data augmentations)

Model	Acc.	Model	Acc.
vgg_15_BN_64	93.0	densenet_BC_100_12	95.5
vgg_16	93.6	resnext_29_8x64d	96.2
resnet_32	92.5	DARC	96.6
resnet_44	93.0	**shake_shake_64d_cutout**	**97.1**
resnet_56	93.3	NAS	95.4
resnet_110	93.5	**AdaNet**	**97.6**
wide_resnet	95.0	**AutoDL**	**98.0**

Differentiable NAS

- Our Work: Resource-Customized Differentiable NAS (Tech Report)

Optimization Under Resource Constraint

$$\min_{\alpha} \mathcal{L}_{\alpha}(w,\alpha) + \min_{w} \mathcal{L}_{w}(w,\alpha)$$

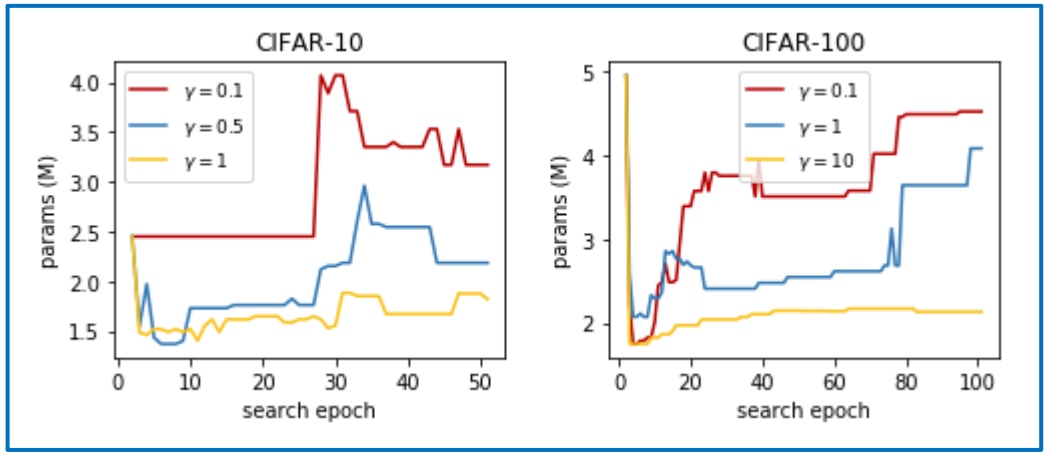
where

$$\mathcal{L}_{\alpha} = \mathcal{L}_{val} + \gamma \mathcal{L}_{resource}$$
$$\mathcal{L}_{w} = \mathcal{L}_{train}$$

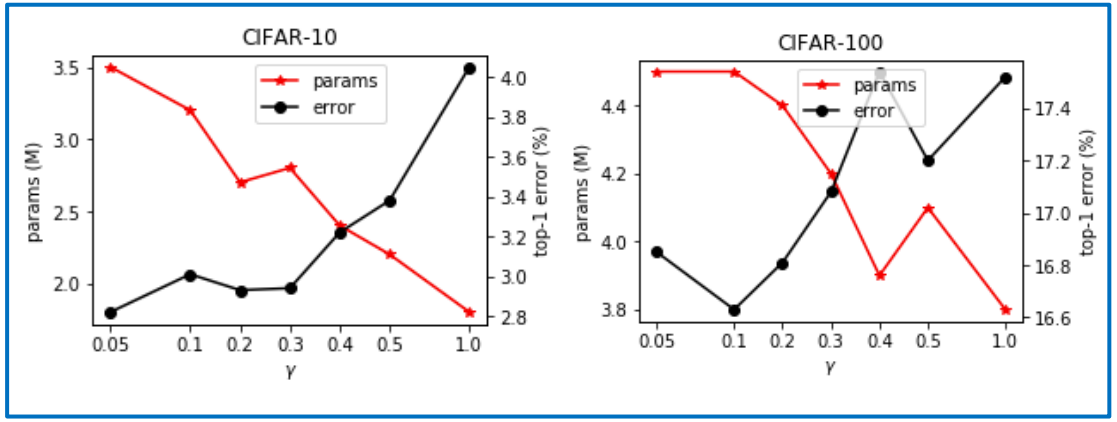
Resource Constraint

$$\mathcal{L}_{resource} = \sum_{j}\sum_{i<j}\sum_{o\in\mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o'\in\mathcal{O}}\exp(\alpha_{o'}^{(i,j)})} r(o^{(i,j)})$$

Arch Params vs. Search Epochs



Params/Test Errors vs. γ



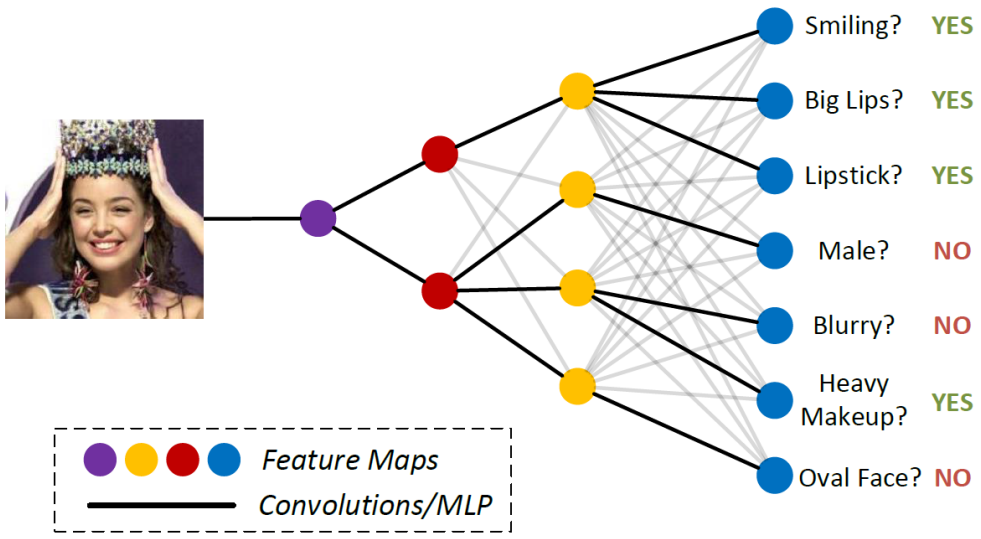
Goal: A Brief Overview of NAS

- **AutoDL and NAS**
- **NAS with RL**
  - NAS with RL
  - NAS-Net
  - Weight sharing mechanism
  - *Our work: Adaptive cross-domain NAS*
  - *Our work: AutoGAN*
- **Differentiable NAS**
  - Architecture embedding method
  - Continuous relaxation of discrete architecture
  - *Our work: NAS-based state-of-the-art image classification model*
  - *Our work: Resource-customized NAS*
- **Greedy NAS**
  - *Our work: NAS for tree-topology network*

Greedy NAS

- Our Work: NAS for Tree-Topology Network (Huang *et al.* ACM MM'18)

Multi-Attribute (Multi-Task) Learning



$$\hat{G} = \arg\max_{G} R(G)$$

$$= \arg\max_{G} \frac{1}{N} \sum_{n=1}^{N} r_n(G)$$

Attributes share representations via tree-topology

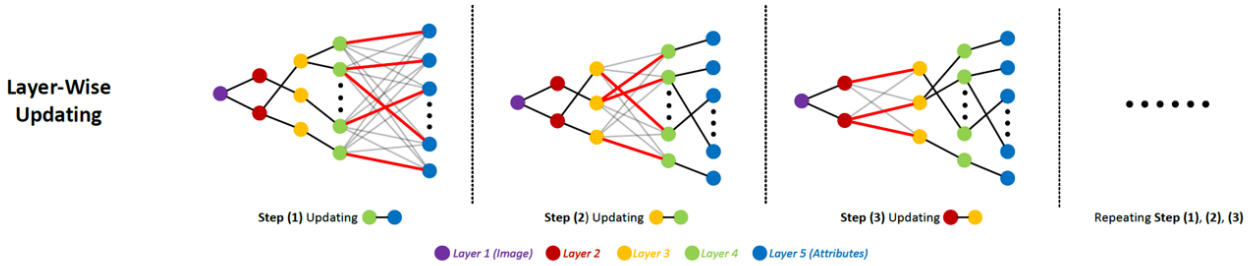
Greedy NAS

- Our Work: NAS for Tree-Topology Network (Huang *et al.* ACM MM'18)

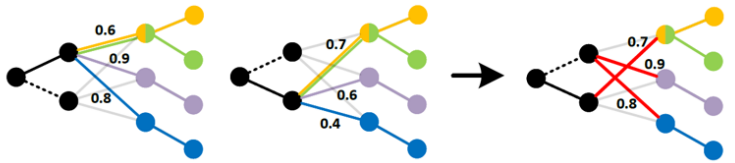
➢ **Idea: Divide-and-Conquer the Search Space**

➢ **Integrating Different Greedy Strategies**

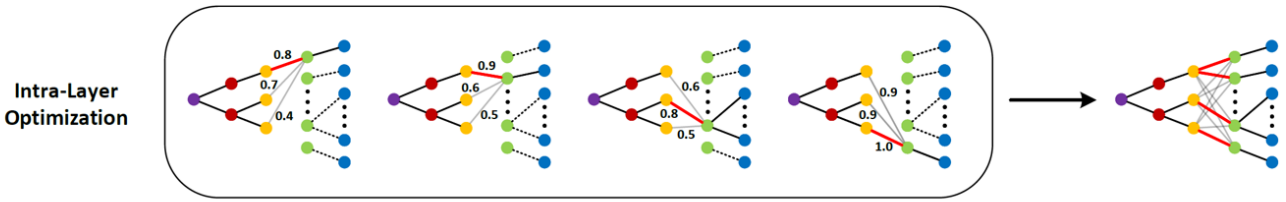
GNAS Strategy #1: Global —> Layers



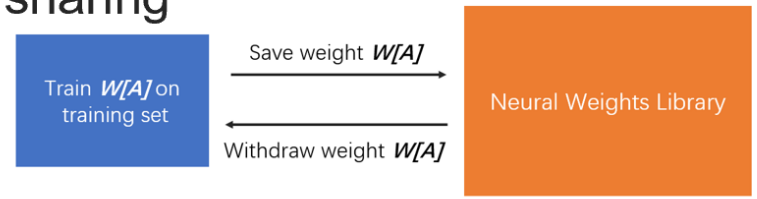
GNAS Strategy #3: Evaluate connections in together



GNAS Strategy #2: Layer —> Connections



GNAS Strategy #4: Neural weight sharing

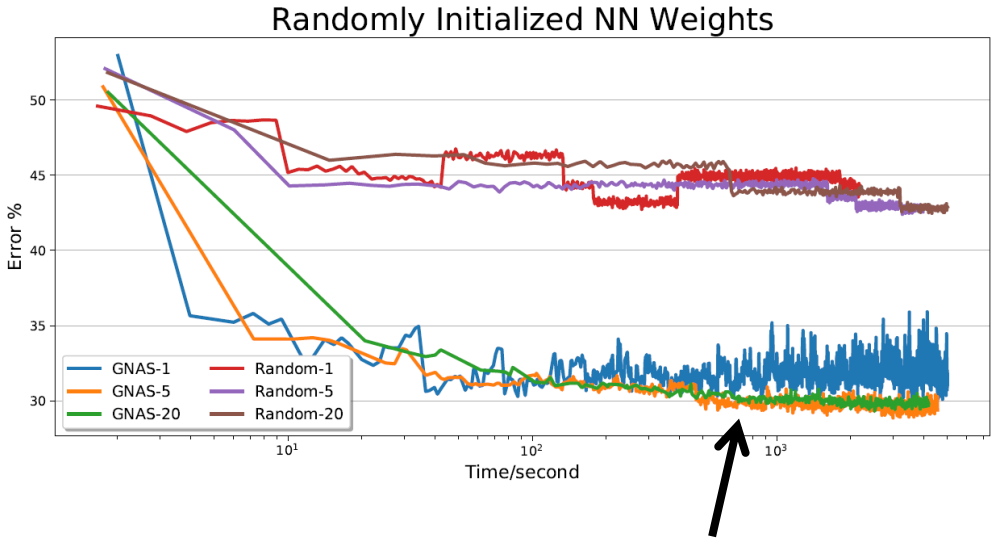


Greedy NAS

- Our Work: NAS for Tree-Topology Network (Huang *et al.* ACM MM'18)

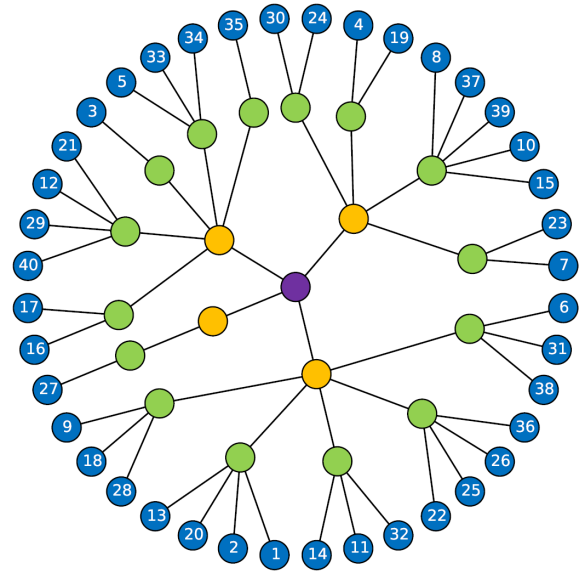
Search Cost

1 GPU * 1 day on LFWA (6,000 images)
on Market-1501 (17,000 images)
1 GPU * 2 days on CelebA (180,000 images)



Significantly better than random search

Tree-Topology—Reasonable Hierarchical Grouping of Attributes



1 *5'o Clock Shadow* 2 *Arched Eyebrows* 3 *Attractive* 4 *Bags Under Eyes* 5 *Bald* 6 *Bangs* 7 *Big Lips* 8 *Big Nose* 9 *Black Hair* 10 *Blond Hair* 11 *Blurry* 12 *Brown Hair* 13 *Bushy Eyebrows* 14 *Chubby* 15 *Double Chin* 16 *Eyeglasses* 17 *Goatee* 18 *Gray Hair* 19 *Heavy Makeup* 20 *High Cheekbones* 21 *Male* 22 *Mouth Slightly Open* 23 *Mustache* 24 *Narrow Eyes* 25 *No Beard* 26 *Oval Face* 27 *Pale Skin* 28 *Pointy Nose* 29 *Receding Hairline* 30 *Rosy Cheeks* 31 *Sideburns* 32 *Smiling* 33 *Straight Hair* 34 *Wavy Hair* 35 *Wearing Earrings* 36 *Wearing Hat* 37 *Wearing Lipstick* 38 *Wearing Necklace* 39 *Wearing Necktie* 40 *Young*

References

1. B Zoph and QV Le. Neural Architecture Search with Reinforcement Learning. In ICLR, 2017.
2. B Zoph et al. Learning Transferable Architectures for Scalable Image Recognition. In CVPR, 2018.
3. H Pham et al. Efficient Neural Architecture Search via Parameter Sharing. In ICML, 2018.
4. Z Cheng et al. Learning to Transfer: Generalizable Attribute Learning with Multitask Neural Model Search. In ACM MM, 2018.
5. Wang H, Huan J. AGAN: Towards Automated Design of Generative Adversarial Networks. arXiv:1906.11080, 2019.
6. R Luo et al. Neural Architecture Optimization. In NIPS, 2018.
7. H Liu et al. Darts: Differentiable Architecture Search. In ICLR, 2019
8. S Huang et al. GNAS: A Greedy Neural Architecture Search Method for Multi-Attribute Learning. In ACM MM, 2018.

Dr. Siyu Huang

Baidu Research

huangsiyu@baidu.com

Homepage: https://siyuhuang.github.io/