GNAS: A Greedy Neural Architecture Search Method for Multi-Attribute Learning

Siyu Huang, Xi Li, Zhi-Qi Cheng,
Zhongfei Zhang, Alexander Hauptmann

Carnegie Mellon University

ACM MM 2018

Background: Automated Machine Learning (AutoML)

Goal: Towards the automation of machine learning pipelines.

- to make ML available for non-ML experts
- to accelerate research on ML

Tasks

- data preparation
- model selection
- hyperparameter optimization

……

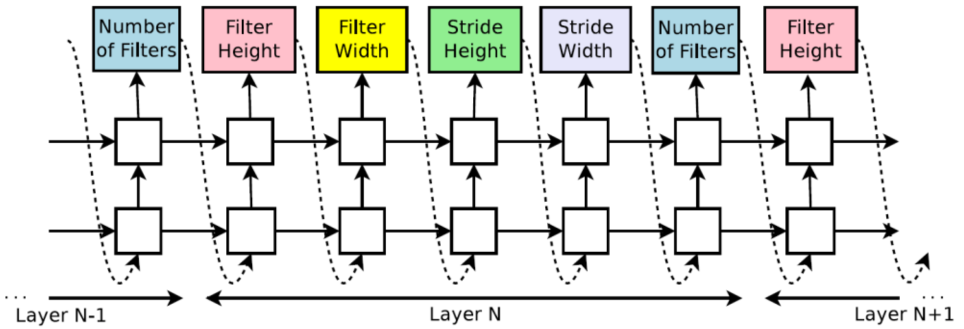
- **deep neural network architecture search**

Background: Neural Architecture Search (NAS)

Goal: To automate the architecture design of neural networks.

Typical approaches

- random search
- Bayesian optimization
- evolutionary algorithm
- reinforcement learning



RL Controller [*Zoph and Le, ICLR'17*]

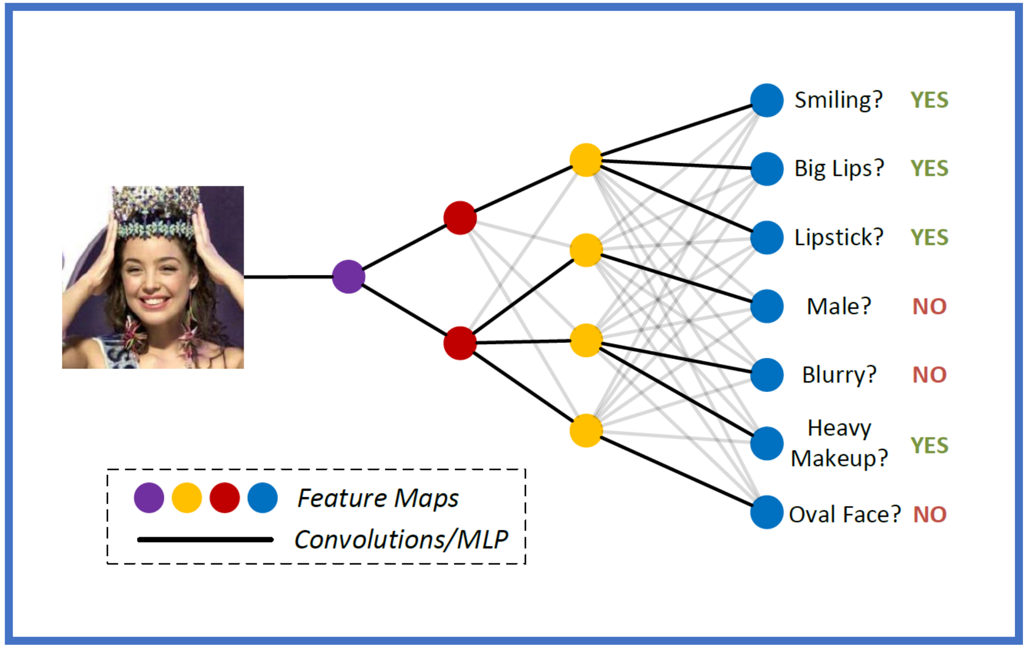
Method	GPUs	Times (days)	Params (million)	Error (%)
Budgeted Super Nets (Veniat & Denoyer, 2017)	–	–	–	9.21
ConvFabrics (Saxena & Verbeek, 2016)	–	–	21.2	7.43
Macro NAS + Q-Learning (Baker et al., 2017a)	10	8-10	11.2	6.92
Net Transformation (Cai et al., 2018)	5	2	19.7	5.70
FractalNet (Larsson et al., 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!

[*Pham et al., ICML'18*]

Our Work: Developing NAS to multi-attribute learning

Goal: To find the optimal tree-like neural network topology.



Formulation

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

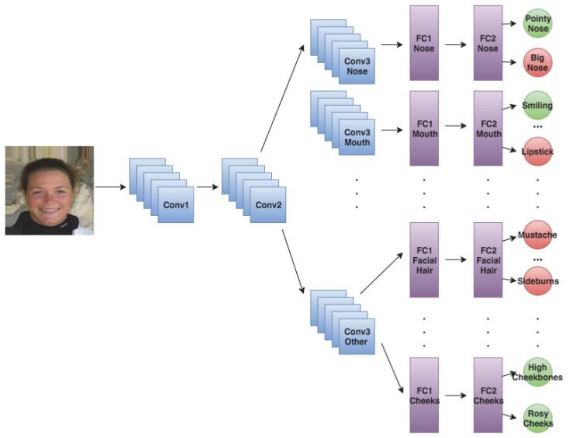
A difficult black-box optimization problem

- A huge number of candidate architectures
- Huge evaluation costs of candidate architectures
  - Training every candidate until convergence

Existing architecture design methods for attribute prediction

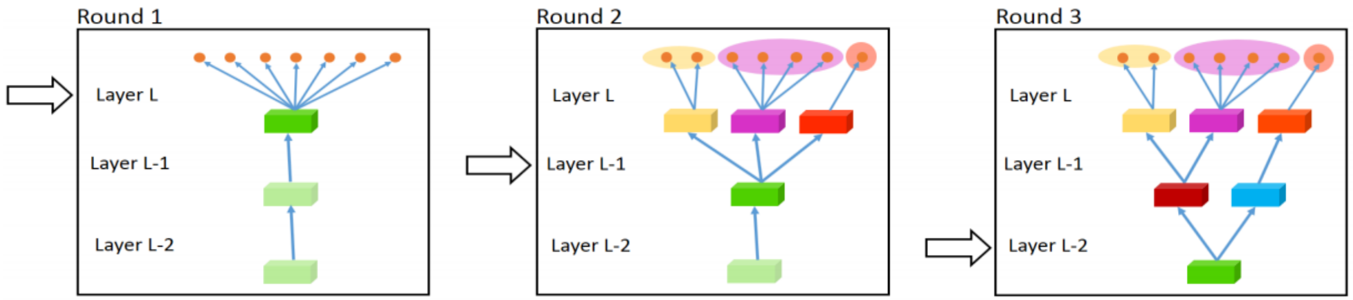
Hand-crafted

Group	Attributes
Gender	Male
Nose	Big Nose, Pointy Nose
Mouth	Big Lips, Lipstick, Mouth Slightly Open, Smiling
Eyes	Arched Eyebrows, Bags Under Eyes, Bushy Eyebrows, Eyeglasses, Narrow Eyes
Face	Attractive, Blurry, Heavy Makeup, Oval Face, Pale Skin, Young
AroundHead	Balding, Bangs, Black Hair, Blond Hair, Brown Hair, Earrings, Gray Hair, Hat, Necklace, Necktie, Receding Hairline, Straight Hair, Wavy Hair
FacialHair	5 o'clock Shadow, Goatee, Mustache, No Beard, Sideburns
Cheeks	High Cheekbones, Rosy Cheeks
Fat	Chubby, Double Chin



[*Hand and Chellappa, AAAI'17*]

Learning-based

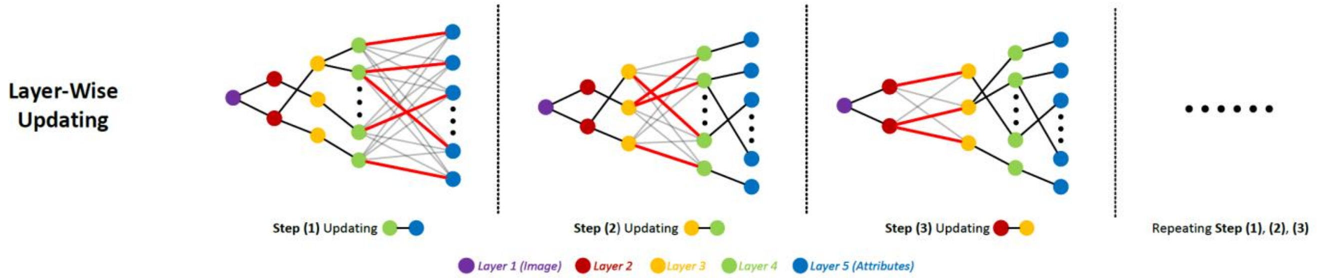


[*Lu et al., CVPR'17*]

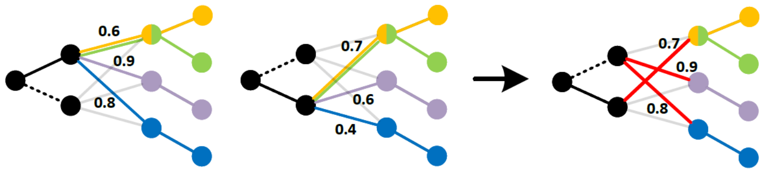
Our Key Idea

Exploiting the nature of tree structure ⟶ Greedy NAS strategies

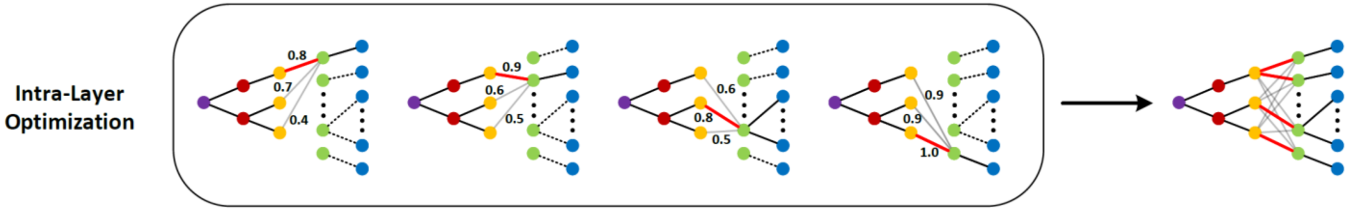
GNAS Strategy #1: Global —> Layers



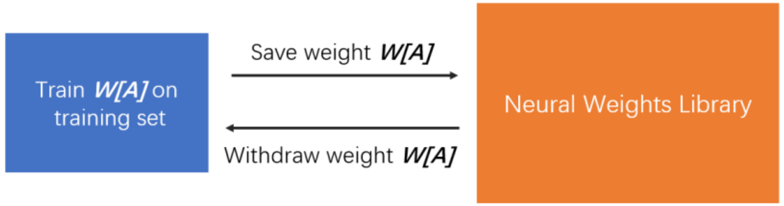
GNAS Strategy #3:
Evaluate connections in together



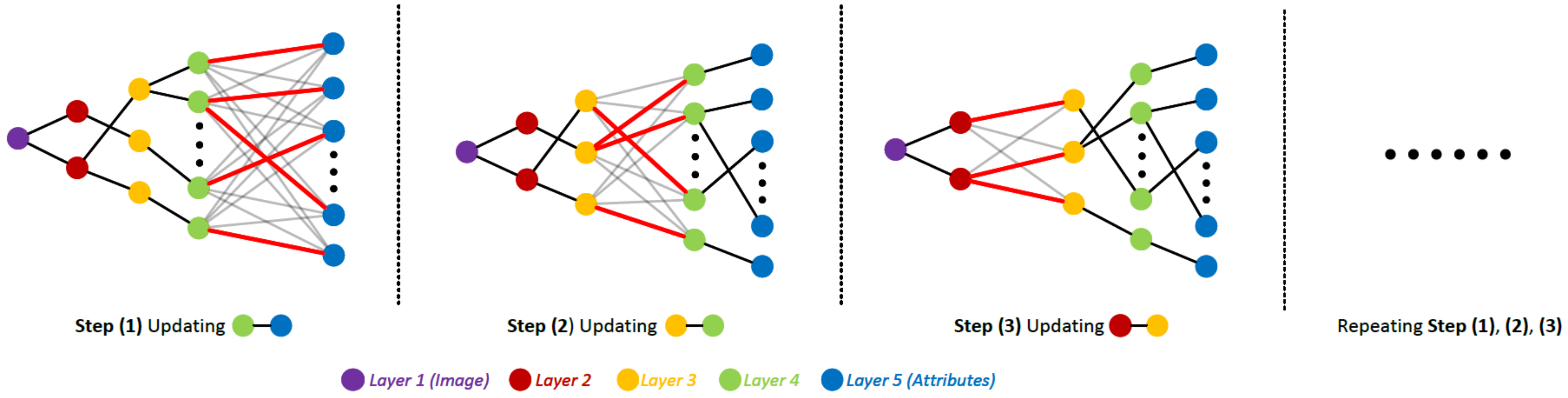
GNAS Strategy #2: Layer —> Connections



GNAS Strategy #4: Neural weight sharing



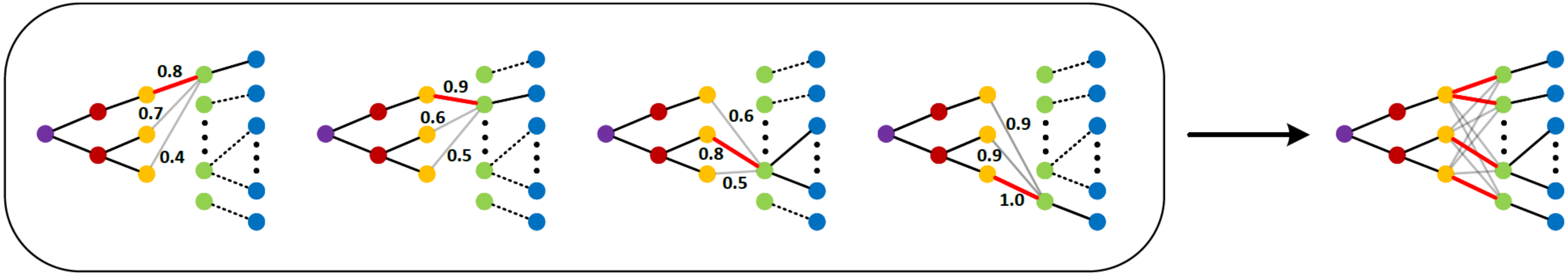
GNAS Strategy #1: Global —> Layers



$$\hat{A} = \left\{ \underset{A^{(l)}}{\arg\max}\, R\left(A^{(l)} \,\middle|\, \boxed{A^{(L)}, L \neq l}\right), \text{s.t.} \sum_{i,j} A_{i,j}^{(l)} = N \right\} \text{ for } l = 1, \ldots, M-1$$

Architectures of the other layers are fixed.

GNAS Strategy #2: Layer —> Connections



To find the best-***N*** connections within the *l*-th layer.

$$\arg\max_{A^{(l)}} R\left(A^{(l)} \mid A^{(L)}, L \neq l\right), \qquad \boxed{\text{s.t.} \sum_{i,j} A_{i,j}^{(l)} = N} \qquad (4)$$

$$= \arg\max_{A^{(l)}} \frac{1}{N} \sum_{n=1}^{N} r_n \left(A^{(l)} \mid A^{(L)}, L \neq l\right), \quad \text{s.t.} \sum_{i,j} A_{i,j}^{(l)} = N$$

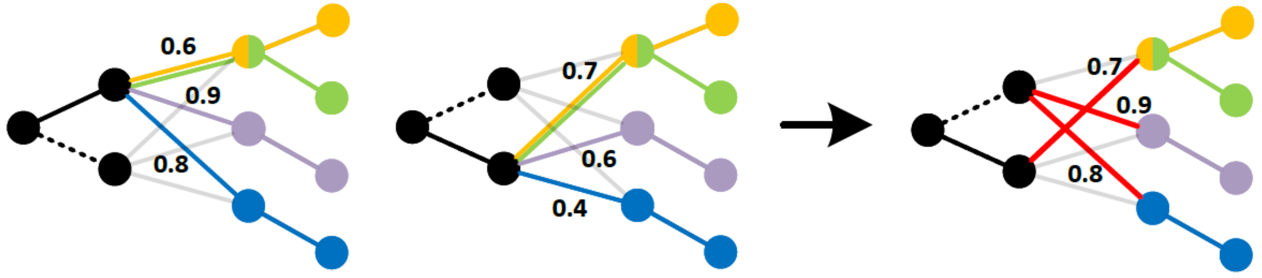
$$\simeq \left\{ \arg\max_{A^{(l)}} r_n \left(A^{(l)} \mid A^{(L)}, L \neq l\right), \quad \text{s.t.} \sum_{i,j} A_{i,j}^{(l)} = 1 \right\} \text{ for } n = 1, \ldots, N$$

To find the best-**1** connection w.r.t each attribute.

Number of candidate architectures within one layer:

$$B_l^{B_{l+1}} \rightarrow B_l \cdot B_{l+1}$$

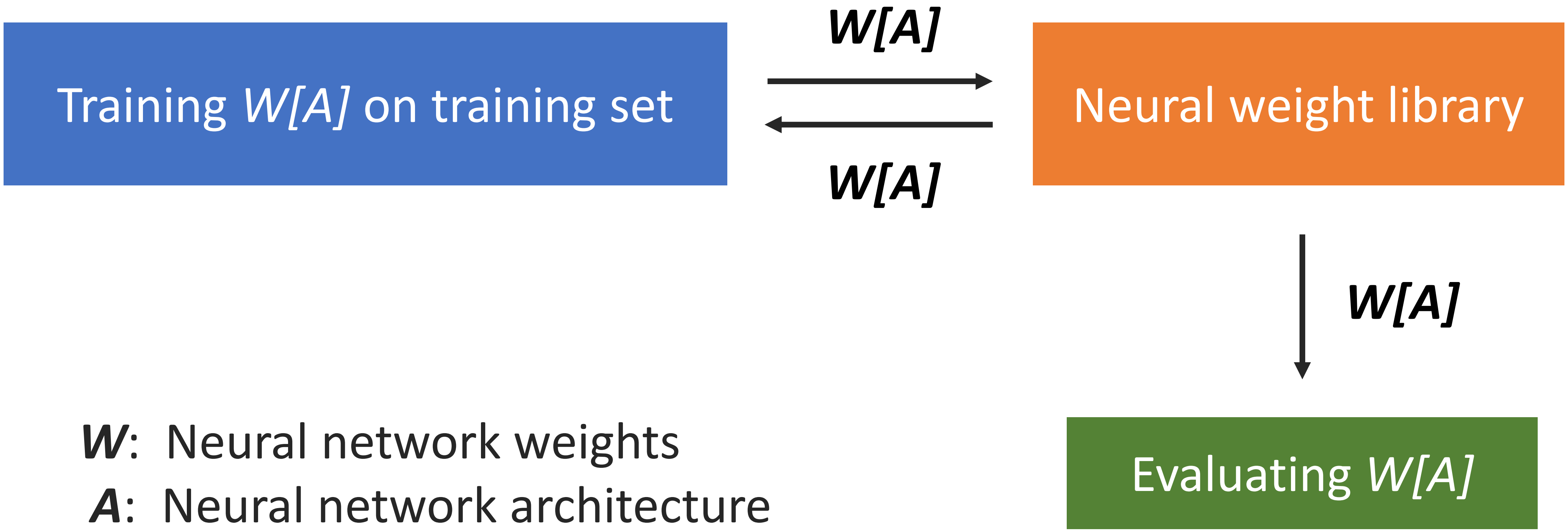
GNAS Strategy #3: Evaluate connections in together



Number of candidate architectures within one layer:

$$B_l^{B_{l+1}} \rightarrow B_l \cdot B_{l+1} \rightarrow B_l$$

GNAS Strategy #4: Neural weight sharing [*ENAS, ICML'18*]



W: Neural network weights
A: Neural network architecture

Efficiency of GNAS

Algorithm 1: Greedy neural architecture search (GNAS)

Input: Training set D_{train}, validation set D_{valid}, layer number M, block number B
Output: Neural network architecture A
1. Initialization
- Randomly initialize architecture A subject to Eq. 2;
- Randomly initialize neural network weights W;

2. Updating
- **while** *not converged* **do**
  - **for** l=M-1 *downto* 1 **do**
    - **for** b=1 *to* B_l **do**
      - $A_{i,j}^{(l)} \leftarrow \begin{cases} 1, & i = b \\ 0, & i \neq b \end{cases}$;
      - Train $W[A]$ on batches of D_{train};
      - $r(A) \leftarrow$ Evaluate $W[A]$ on batches of D_{valid};
    - Update layer architecture $A^{(l)}$ based on r by Eq. 6;

Training cost:

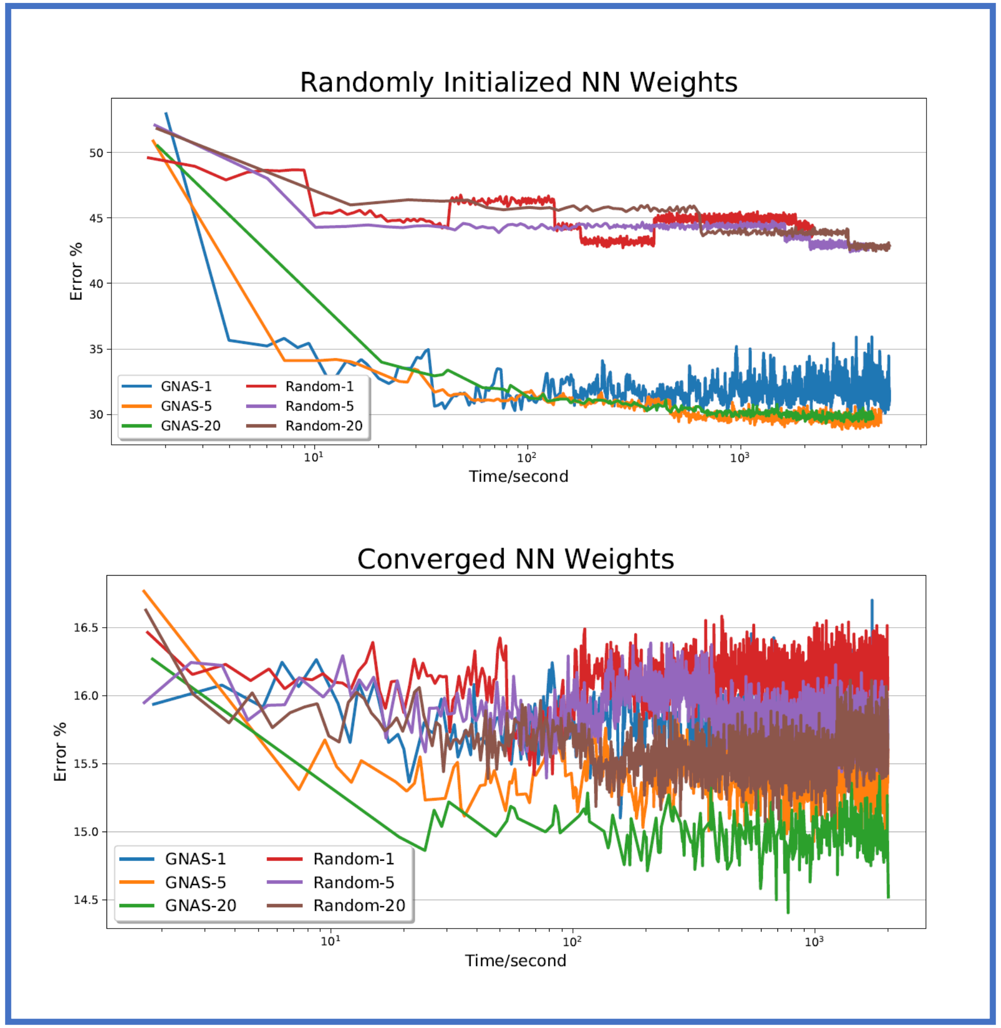
1 GPU * 1 day on LFWA (6k images),
Market-1501 (17k images)

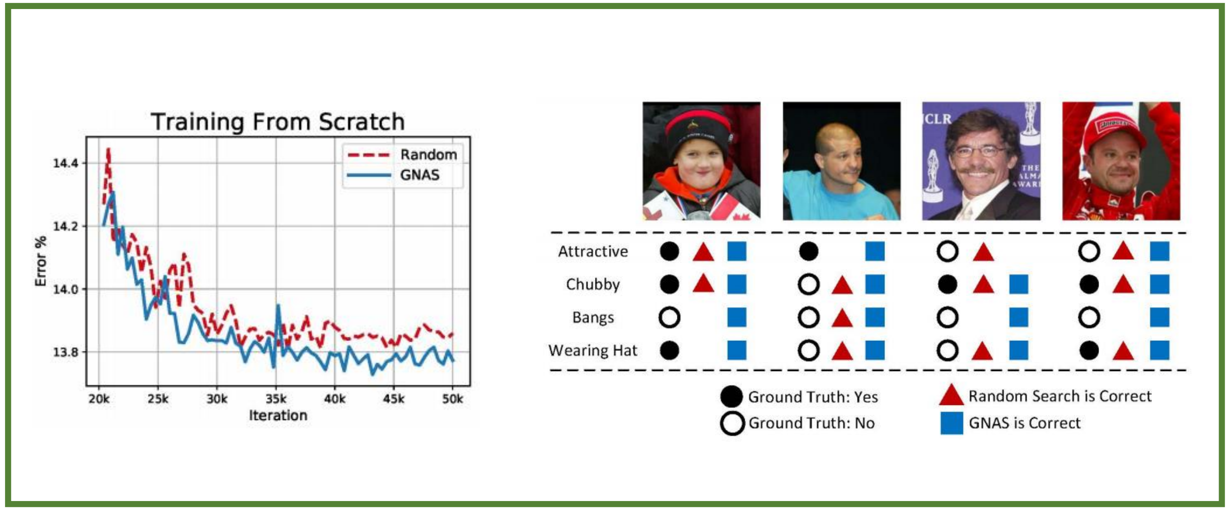
1 GPU * 2 days on CelebA (180k images)

Advantages:

1) Reducing numerous candidate architectures
2) Accelerating training by weight sharing
3) Large search space
4) Non-parametric

Experiment #1: GNAS vs. Random search





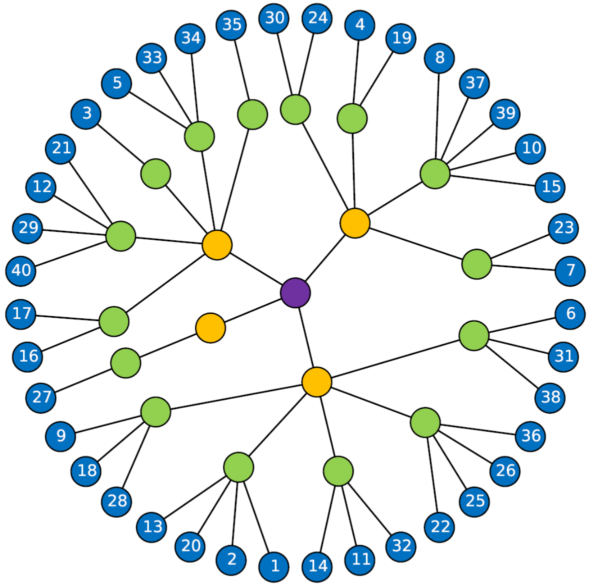
Results

1) GNAS has **better performance** and **faster convergence speed**.
2) **Larger validation batch size is better** for both GNAS and random search.

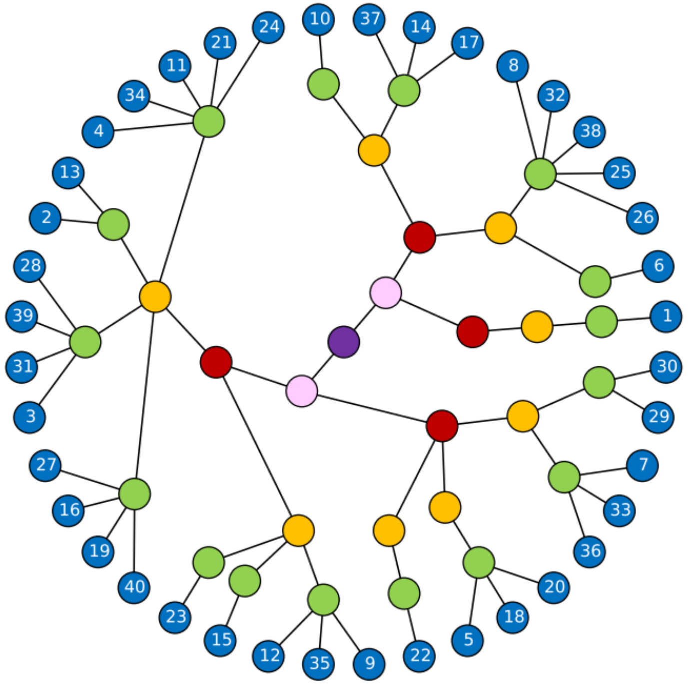
Experiment #2: GNAS vs. Hand-crafted architectures

Search space configuration

Layer	Kernel	Shallow			Deep		
		Block	Channel		Block	Channel	
			Thin	Wide		Thin	Wide
Conv-1	7×7	1	16	64	1	16	64
Conv-2	3×3	1	32	128	**2**	16	64
Conv-3	3×3	1	64	256	**4**	16	64
Conv-4	3×3	**4**	32	128	**8**	16	64
Conv-5	3×3	**16**	16	32	**16**	16	32
FC-1	-	***N***	64	128	***N***	64	128
FC-2	-	*N*	64	128	*N*	64	128
FC-3	-	*N*	2	2	*N*	2	2



Shallow



Deep

Experiment #2: GNAS vs. Hand-crafted architectures

Results

Facial attribute prediction

Method	Mean Error (%)		Params (million)	Test Speed (ms)	Adaptive?
	CelebA	LFWA			
LNets+ANet [14]	13	16	-	-	No
Separate Task [22]	9.78	-	-	-	No
MOON [22]	9.06	-	119.73	12.53	No
Independent Group [6]	8.94	13.72	-	-	No
MCNN [6]	8.74	13.73	-	-	No
MCNN-AUX [6]	8.71	13.69	-	-	No
VGG-16 Baseline [15]	8.56	-	134.41	12.60	No
Low-rank Baseline [15]	9.12	-	4.52	6.07	No
SOMP-thin-32 [15]	10.04	-	0.22	1.94	Yes
SOMP-branch-64 [15]	8.74	-	4.99	5.77	Yes
SOMP-joint-64 [15]	8.98	-	10.53	6.18	Yes
PaW-subnet [5]	9.11	-	0.27	-	Yes
PaW [5]	8.77	-	11	-	Yes
GNAS-Shallow-Thin	8.70	13.84	1.57	0.33	Yes
GNAS-Shallow-Wide	**8.37**	**13.63**	7.73	0.64	Yes
GNAS-Deep-Thin	9.10	14.12	1.47	0.87	Yes
GNAS-Deep-Wide	8.64	13.94	6.41	0.89	Yes

Person attribute prediction

Method	Market-1501 (%)
Ped-Attribute-Net [14]	13.81
Separate Models [8]	13.32
APR [14]	11.84
Equal-Weight [29]	13.16
Adapt-Weight [8]	11.51
Random-Thin	11.94
Random-Wide	11.42
GNAS-Thin	11.37
GNAS-Wide	**11.17**

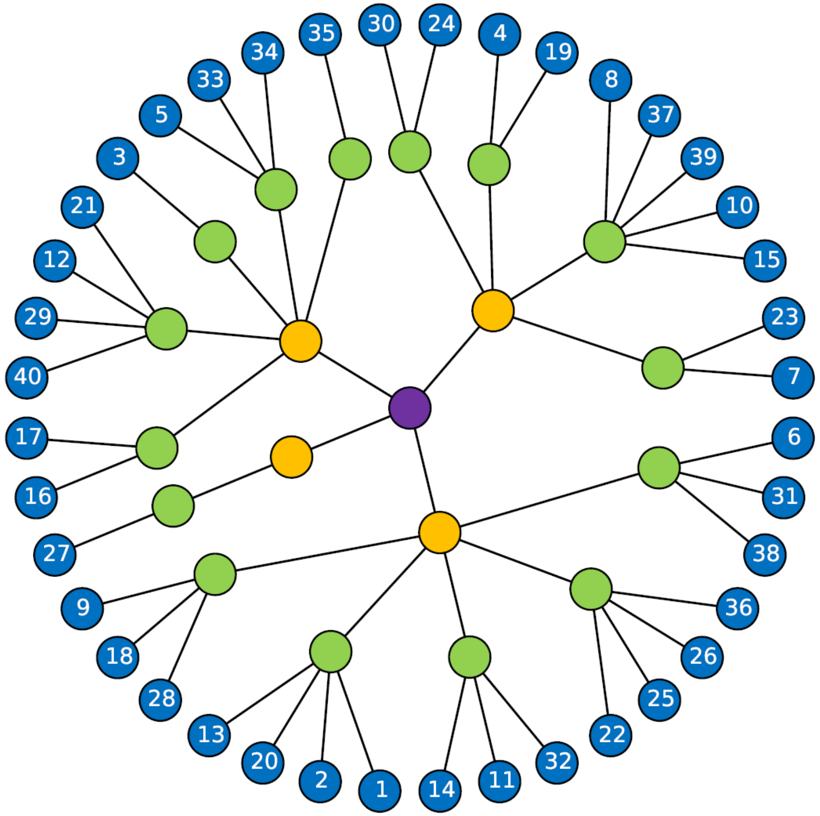
Per-attribute performance

Attribute \ Method	LANet	Inde.	MCNN	M-AUX	PaW	GNAS
5'o Clock Shadow	9.00	6.06	5.59	5.49	5.36	**5.24**
Arched Eyebrows	21.00	16.84	16.45	16.58	16.99	**15.75**
Attractive	19.00	17.78	17.06	**16.94**	17.14	**16.94**
Bags Under Eyes	21.00	15.17	15.11	15.08	15.42	**14.13**
Bald	2.00	1.15	1.13	1.10	1.07	**1.04**
Bangs	5.00	4.01	3.96	3.95	4.07	**3.80**
Big Lips	32.00	29.20	28.80	28.53	28.54	**28.21**
Big Nose	22.00	15.53	15.50	15.47	16.37	**14.90**
Black Hair	12.00	10.59	10.13	10.22	10.16	**9.76**
Blond Hair	5.00	4.12	4.03	3.99	4.15	**3.89**
Blurry	16.00	3.93	3.92	3.83	3.89	**3.58**
Brown Hair	20.00	11.25	11.01	10.85	11.50	**10.25**
Bushy Eyebrows	10.00	7.13	7.20	7.16	7.38	**7.01**
Chubby	9.00	4.45	4.34	4.33	4.54	**4.07**
Double Chin	8.00	3.57	3.59	3.68	3.74	**3.52**
Eyeglasses	1.00	0.33	0.37	0.37	0.41	**0.31**
Goatee	5.00	2.87	2.70	2.76	2.62	**2.41**
Gray Hair	3.00	1.93	1.80	1.80	1.79	**1.63**
Heavy Makeup	10.00	9.05	8.63	8.45	8.47	**8.18**
High Cheekbones	12.00	12.66	12.45	12.42	12.56	**11.95**
Male	2.00	1.98	1.84	1.83	1.61	**1.50**
Mouth Slightly Open	8.00	6.01	6.26	6.26	5.95	**5.84**
Mustache	5.00	3.33	3.07	3.12	3.10	**2.97**
Narrow Eyes	19.00	12.78	12.84	12.77	12.44	**12.34**
No Beard	5.00	4.07	3.89	3.95	3.78	**3.70**
Oval Face	34.00	25.30	24.19	**24.16**	24.97	24.43
Pale Skin	9.00	2.93	2.99	2.95	2.92	**2.76**
Pointy Nose	28.00	22.53	22.53	22.53	22.65	**21.76**
Receding Hairline	11.00	6.59	6.19	6.19	6.56	**6.06**
Rosy Cheeks	10.00	4.98	4.87	**4.84**	4.93	4.99
Sideburns	4.00	2.23	2.18	2.15	2.36	**2.04**
Smiling	8.00	7.35	7.34	7.27	7.27	**6.76**
Straight Hair	27.00	17.38	16.61	16.42	16.48	**15.23**
Wavy Hair	20.00	16.76	16.08	16.09	15.93	**15.48**
Wearing Earrings	18.00	9.65	9.68	9.57	10.07	**9.02**
Wearing Hat	1.00	1.03	0.96	0.95	0.98	**0.88**
Wearing Lipstick	7.00	6.20	6.05	5.89	5.76	**5.59**
Wearing Necklace	29.00	13.59	13.18	13.37	12.30	**12.39**
Wearing Necktie	7.00	3.29	3.47	3.49	**3.15**	3.24
Young	13.00	12.02	11.70	11.52	11.41	**11.11**
Ave.	12.67	8.94	8.74	8.71	8.77	**8.37**

GNAS architecture

1) Better performance
2) Fewer model parameters
3) Faster forward speed

Experiment #3: Architecture discovered by GNAS



Related attributes are grouped together

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*

Take-Home Messages

- Searching for tree-like NN topology
- Improving search efficiency by multiple greedy strategies

References

[1] B Zoph and QV Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

[2] H Pham et al. Efficient Neural Architecture Search via Parameter Sharing. In *ICML*, 2018.

[3] Hand et al. Attributes for Improved Attributes: A Multi-Task Network Utilizing Implicit and Explicit Relationships for Facial Attribute Classification. In *AAAI*, 2017.

[4] Lu et al. Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. In *CVPR*, 2017.

Contact

Siyu Huang
siyuhuang@zju.edu.cn
siyuhuang.github.io