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

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

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

Huge computing cost!

---

RL Controller [Zoph and Le, ICLR’17]

---

<table>
<thead>
<tr>
<th>Method</th>
<th>GPUs</th>
<th>Times (days)</th>
<th>Params (million)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budgeted Super Nets (Veniat &amp; Denoyer, 2017)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9.21</td>
</tr>
<tr>
<td>ConvFabrics (Saxena &amp; Verbeek, 2016)</td>
<td>-</td>
<td>-</td>
<td>21.2</td>
<td>7.43</td>
</tr>
<tr>
<td>Macro NAS + Q-Learning (Baker et al., 2017a)</td>
<td>10</td>
<td>8-10</td>
<td>11.2</td>
<td>6.92</td>
</tr>
<tr>
<td>Net Transformation (Cai et al., 2018)</td>
<td>5</td>
<td>2</td>
<td>19.7</td>
<td>5.70</td>
</tr>
<tr>
<td>FractalNet (Larsson et al., 2017)</td>
<td>-</td>
<td>-</td>
<td>38.6</td>
<td>4.60</td>
</tr>
<tr>
<td>SMASH (Rado et al., 2018)</td>
<td>16.0</td>
<td>4.03</td>
<td>38.6</td>
<td>4.60</td>
</tr>
<tr>
<td>NAS (Zoph &amp; Le, 2017)</td>
<td>800</td>
<td>21-28</td>
<td>7.1</td>
<td>4.47</td>
</tr>
<tr>
<td>NAS + more filters (Zoph &amp; Le, 2017)</td>
<td>800</td>
<td>21-28</td>
<td>37.4</td>
<td>3.65</td>
</tr>
</tbody>
</table>

[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

[Hand and Chellappa, AAAI’17]

Learning-based

[Lu et al., CVPR’17]
Our Key Idea

Exploiting the nature of tree structure $\rightarrow$ Greedy NAS strategies

GNAS Strategy #1: Global $\rightarrow$ Layers

GNAS Strategy #2: Layer $\rightarrow$ Connections

GNAS Strategy #3: Evaluate connections in together

GNAS Strategy #4: Neural weight sharing

Our Key Idea

- GNAS Strategy #1: Global $\rightarrow$ Layers
  - Layer-Wise Updating

- GNAS Strategy #2: Layer $\rightarrow$ Connections
  - Intra-Layer Optimization

- GNAS Strategy #3: Evaluate connections in together

- GNAS Strategy #4: Neural weight sharing
  - Train $W[A]$ on training set
  - Save weight $W[A]$ to Neural Weights Library
  - Withdraw weight $W[A]$
GNAS Strategy #1: Global $\rightarrow$ Layers

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel</th>
<th>Shallow Block</th>
<th>Shallow Channel</th>
<th>Deep Block</th>
<th>Deep Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Thin</td>
<td>Wide</td>
<td>Thin</td>
<td>Wide</td>
</tr>
<tr>
<td>Conv-1</td>
<td>7×7</td>
<td>1</td>
<td>16</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>Conv-2</td>
<td>3×3</td>
<td>1</td>
<td>32</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>Conv-3</td>
<td>3×3</td>
<td>1</td>
<td>64</td>
<td>256</td>
<td>4</td>
</tr>
<tr>
<td>Conv-4</td>
<td>3×3</td>
<td>4</td>
<td>32</td>
<td>128</td>
<td>8</td>
</tr>
<tr>
<td>Conv-5</td>
<td>3×3</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>FC-1</td>
<td>-</td>
<td>N</td>
<td>64</td>
<td>128</td>
<td>N</td>
</tr>
<tr>
<td>FC-2</td>
<td>-</td>
<td>N</td>
<td>64</td>
<td>128</td>
<td>N</td>
</tr>
<tr>
<td>FC-3</td>
<td>-</td>
<td>N</td>
<td>2</td>
<td>2</td>
<td>N</td>
</tr>
</tbody>
</table>

Shallow

Deep
**Experiment #2: GNAS vs. Hand-crafted architectures**

### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>facial attribute prediction</th>
<th>Person attribute prediction</th>
<th>Per-attribute performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Facial attribute prediction</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LNet-ANet [14]</td>
<td>13</td>
<td>13.81</td>
<td></td>
</tr>
<tr>
<td>Separate Task [22]</td>
<td>9.78</td>
<td>13.32</td>
<td></td>
</tr>
<tr>
<td>MOON [22]</td>
<td>9.06</td>
<td>11.84</td>
<td></td>
</tr>
<tr>
<td>MCNN [6]</td>
<td>8.74</td>
<td>13.75</td>
<td></td>
</tr>
<tr>
<td>MCNN AUX [6]</td>
<td>8.71</td>
<td>13.69</td>
<td></td>
</tr>
<tr>
<td>VGG-16 Baseline [15]</td>
<td>8.56</td>
<td>12.41</td>
<td></td>
</tr>
<tr>
<td>Low-Rank Baseline [15]</td>
<td>9.12</td>
<td>12.60</td>
<td></td>
</tr>
<tr>
<td>SOPM-thin-32 [15]</td>
<td>10.64</td>
<td>11.48</td>
<td></td>
</tr>
<tr>
<td>SOPM-branch-64 [15]</td>
<td>8.74</td>
<td>11.48</td>
<td></td>
</tr>
<tr>
<td>SOPM-joint-64 [15]</td>
<td>8.98</td>
<td>11.48</td>
<td></td>
</tr>
<tr>
<td>PaW [5]</td>
<td>8.77</td>
<td>11.48</td>
<td></td>
</tr>
<tr>
<td><strong>GNAS-Shallow-Thin</strong></td>
<td>8.97</td>
<td>11.63</td>
<td></td>
</tr>
<tr>
<td><strong>GNAS-Deep-Thin</strong></td>
<td>8.64</td>
<td>11.63</td>
<td></td>
</tr>
<tr>
<td><strong>GNAS-Deep-Wide</strong></td>
<td>8.64</td>
<td>11.63</td>
<td></td>
</tr>
</tbody>
</table>

| **Person attribute prediction** |                           |                             |                           |
| Separate Models [8]        | 13.32                     | 13.32                       |                           |
| APR [14]                  | 11.84                     | 11.84                       |                           |
| Equal-Weight [29]         | 13.16                     | 13.16                       |                           |
| Adapt-Weight [8]          | 11.51                     | 11.51                       |                           |
| **Random-Thin**           | 11.94                     | 11.94                       |                           |
| **Random-Wide**           | 11.42                     | 11.42                       |                           |
| **GNAS-Thin**             | 11.37                     | 11.37                       |                           |
| **GNAS-Wide**             | 11.17                     | 11.17                       |                           |

**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


Contact

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