SGAS: Sequential Greedy Architecture Search

Guohao Li*, Guocheng Qian*, Itzel C. Delgadillo*, Matthias Muller, Ali Thabet, Bernard Ghanem
Background

- NAS
- Existing Problems

SGAS: Sequential Greedy Architecture Search

- Greedy Decision
- Selection Criteria

Search architectures for CNN and GCN

- CNN (CIFAR-10 & ImagNet)
- GCN (ModelNet & PPI)
NAS - Motivation

Designing Arch. is Painful!

A Smarter Way?

NAS with RL

AlexNet

VGG19

GoogleNet

ResNet
NAS - Problem Definition

Given: A Search Space

Goal: Find The Best Arch.

Approach: Design A Good Search Strategy

Design a search strategy instead of an instance of architecture
NAS - RL Based

Search Space:
- Filter Height [1, 3, 5, 7]
- Filter Width [1, 3, 5, 7]
- # of Filters [24, 36, 48, 64]

Search Strategy:
- A RNN Controller
- Learn to Generate Model Descriptions
- Trained by REINFORCE

Zoph B, Le Q. Neural Architecture Search with Reinforcement Learning (ICLR’2017)
**NAS - RL Based**

**Search Space:**
- Filter Height [1, 3, 5, 7]
- Filter Width [1, 3, 5, 7]
- # of Filters [24, 36, 48, 64]

**Search Strategy:**
- A RNN Controller
- Learn to Generate Model Descriptions
- Trained by REINFORCE

Zoph B, Le Q. Neural Architecture Search with Reinforcement Learning (ICLR’2017)

800 GPUs, 21-28 Days
NAS - RL Based

Sample architecture A with probability p

The controller (RNN)

Trains a child network with architecture A to get accuracy R

Compute gradient of p and scale it by R to update the controller

**NASNet:** Cell Based, 450 GPUs, 3-4 Days

**ENAS:** Weight Sharing, 1 GPU, 0.45 (4*) Days

**MnasNet:** Platform-Aware, 64 TPUs, 4.5 Days

**MobileNetV3, EfficientNet, NAS-FPN, ...**

Zoph B, Le Q. Neural Architecture Search with Reinforcement Learning (ICLR’2017)
NAS - Evolution Based

AmoebaNet

Search Space:
Cell Based
Pairwise Combinations of ops
Candidate ops:
none (identity);
3x3, 5x5 and 7x7 sep. conv.;
3x3 average pool; 3x3 max pool;
3x3 dilated sep. conv.;
1x7 then 7x1 conv.

Search Strategy:
Aging Evolution:
Population, Mutation, Aging

2000 GPU Days

SMASH, Hier. Evo., SPOS, FairNAS, AutoML-Zero...

Real, E., Aggarwal, A., Huang, Y. and Le, Q.V., Regularized Evolution for Image Classifier Architecture Search (AAAI'2019)
NAS - Gradient Based

DARTS

Search Space:
Cell Based
DAG
Candidate ops:
- identity, and zero;
- $3 \times 3$, $5 \times 5$ sep. conv.;
- $3 \times 3$, $5 \times 5$ dilated sep. conv.;
- $3 \times 3$ max pool, $3 \times 3$ average pool.

Search Strategy:
Super-net
Softmax Over All Possible ops (like attention)
Weight Sharing
Trained by Gradient Descent

NAS - Gradient Based

DARTS

Softmax Over All Possible ops:

\[ \tilde{\sigma}^{(i,j)}(x) = \sum_{\sigma' \in \mathcal{O}} \frac{\exp(\alpha_{\sigma'}^{(i,j)})}{\sum_{\sigma'' \in \mathcal{O}} \exp(\alpha_{\sigma''}^{(i,j)})} \sigma(x) \]

Select The Most Likely op:

\[ o^{(i,j)} = \arg\max_{\sigma \in \mathcal{O}} \alpha_{\sigma}^{(i,j)} \]

Only 1 GPU Day!

SNAS, FBNet, ProxylessNAS, P-DARTS, GDAS, MdeNAS, PC-DARTS, FairDARTS, ...

Existing Problems

DARTS

Devils in Softmax and Weight Sharing:

1. Softmax is too soft
2. Soft model cannot reflect the true accuracy
3. Discrete model w/o weight sharing never gets evaluated during the search

An Extreme Case:

skip-connect (0.34), 3x3Conv (0.33), 5x5 Conv (0.33) & skip-connect (0.33), 3x3Conv (0.34), 5x5 Conv (0.33)

Architectures with a higher validation accuracy during the search phase may perform worse in the evaluation (see Figure 1).

Figure 1. Comparison of search-evaluation Kendall coefficients.
SGAS: Sequential Greedy Architecture Search (CVPR’2020, Guohao Li et.al)

https://www.deepgcns.org/auto/sgas
Aiming to alleviate this common issue, we introduce **sequential greedy architecture search** (SGAS), an efficient method for neural architecture search.

By dividing the search procedure into **sub-problems**, SGAS chooses and prunes candidate operations in a greedy fashion.

---

**Figure 2.** Illustration of Sequential Greedy Architecture Search.
Figure 2. Illustration of Sequential Greedy Architecture Search.
Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^*, j^*)\) based on the greedy Selection Criterion.
Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^\dagger, j^\dagger)\) based on the greedy Selection Criterion.

2. Determine the operation by replacing \(\overline{o}(i^\dagger, j^\dagger)\) with \(o(i^\dagger, j^\dagger) = \arg\max_{o \in \mathcal{O}} \alpha_o(i^\dagger, j^\dagger)\)
**Figure 2. Illustration of Sequential Greedy Architecture Search.**

1. If a decision epoch, select an edge \((i^+, j^+)\) based on the greedy Selection Criterion.
2. Determine the operation by replacing \(o(i^+, j^+)\) with \(\alpha(i^+, j^+) = \max_{o \in \mathcal{O}} \alpha_o(i^+, j^+)\).
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^+, j^+)\) from \(\mathcal{A}\).
Repeat…

Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^\dagger, j^\dagger)\) based on the greedy Selection Criterion
2. Determine the operation by replacing \(\tilde{o}(i^\dagger, j^\dagger)\) with \(o(i^\dagger, j^\dagger) = \arg\max_{o \in \mathcal{O}} \alpha_o(i^\dagger, j^\dagger)\)
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^\dagger, j^\dagger)\) from \(\mathcal{A}\)
Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^\uparrow, j^\uparrow)\) based on the greedy Selection Criterion.
2. Determine the operation by replacing \(\tilde{o}(i^\uparrow, j^\uparrow)\) with \(o(i^\uparrow, j^\uparrow) = \arg\max_{o \in O} \alpha_o(i^\uparrow, j^\uparrow)\).
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^\uparrow, j^\uparrow)\) from \(\mathcal{A}\).
Repeat…

Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^t, j^t)\) based on the greedy Selection Criterion
2. Determine the operation by replacing \(\overline{o}(i^t, j^t)\) with \(o(i^t, j^t) = \arg\max_{o \in O} \alpha_o(i^t, j^t)\)
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^t, j^t)\) from \(\mathcal{A}\)
Figure 2. Illustration of Sequential Greedy Architecture Search.

1. If a decision epoch, select an edge \((i^+, j^+)\) based on the greedy Selection Criterion
2. Determine the operation by replacing \(\tilde{o}(i^+, j^+)\) with \(o(i^+, j^+) = \arg\max_{o \in \mathcal{O}} \alpha_o(i^+, j^+)\)
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^+, j^+)\) from \(\mathcal{A}\)
To maintain the optimality, the design of the selection criterion is crucial.

**Edge Importance:**

\[
S_{EI}^{(i,j)} = \sum_{o \in \mathcal{O}, o \neq \text{zero}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}
\]

**Selection Certainty:**

\[
p_o^{(i,j)} = \frac{\exp(\alpha_{o}^{(i,j)})}{S_{EI}^{(i,j)} \sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}, o \in \mathcal{O}, o \neq \text{zero}
\]

\[
S_{SC}^{(i,j)} = 1 - \frac{-\sum_{o \in \mathcal{O}, o \neq \text{zero}} p_o^{(i,j)} \log(p_o^{(i,j)})}{\log(|\mathcal{O}| - 1)}
\]

**Selection Stability:**

\[
S_{SS}^{(i,j)} = \frac{1}{K} \sum_{t=T-K_{o_t}, \in \mathcal{O}_{o_t}, \neq \text{zero}} \sum_{T-1}^{T-1} \min(p_{o_t}^{(i,j)}, p_{o_T}^{(i,j)})
\]
SGAS - Selection Criteria

**Criterion 1:**

\[ S_{1}^{(i,j)} = \text{normalize}(S_{EI}^{(i,j)}) \times \text{normalize}(S_{SC}^{(i,j)}) \]

**Criterion 2:**

\[ S_{2}^{(i,j)} = S_{1}^{(i,j)} \times \text{normalize}(S_{SS}^{(i,j)}) \]

normalize(·): a standard Min-Max scaling normalization

**Edge Importance:**

\[ S_{EI}^{(i,j)} = \frac{\exp(\alpha_{o}^{(i,j)})}{\sum_{o' \in \mathcal{O}, o' \neq \text{zero}} \exp(\alpha_{o'}^{(i,j)})} \]

**Selection Certainty:**

\[ p_{o}^{(i,j)} = \frac{\exp(\alpha_{o}^{(i,j)})}{S_{EI}^{(i,j)} \sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}, o \in \mathcal{O}, o \neq \text{zero} \]

\[ S_{SC}^{(i,j)} = 1 - \frac{-\sum_{o \in \mathcal{O}, o \neq \text{zero}} p_{o}^{(i,j)} \log(p_{o}^{(i,j)})}{\log(|\mathcal{O}| - 1)} \]

**Selection Stability:**

\[ S_{SS}^{(i,j)} = \frac{1}{K} \sum_{t=T-K}^{T-1} \sum_{o_{t} \in \mathcal{O}_{o_{t}} \neq \text{zero}} \min(p_{o_{t}}^{(i,j)}, p_{o_{t}}^{(i,j)}) \]
<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Err. (%)</th>
<th>Params (M)</th>
<th>Search Cost (GPU-days)</th>
<th>Search Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseNet-BC [18]</td>
<td>3.46</td>
<td>25.6</td>
<td>-</td>
<td>manual</td>
</tr>
<tr>
<td>NASNet-A [55]</td>
<td>2.65</td>
<td>3.3</td>
<td>1800</td>
<td>RL</td>
</tr>
<tr>
<td>AmoebaNet-A [36]</td>
<td>3.34±0.06</td>
<td>3.2</td>
<td>1350</td>
<td>evolution</td>
</tr>
<tr>
<td>AmoebaNet-B [36]</td>
<td>2.55±0.05</td>
<td>2.8</td>
<td>3150</td>
<td>evolution</td>
</tr>
<tr>
<td>Hier-Evolution [28]</td>
<td>3.75±0.12</td>
<td>15.7</td>
<td>300</td>
<td>evolution</td>
</tr>
<tr>
<td>PNAS [27]</td>
<td>3.41±0.09</td>
<td>3.2</td>
<td>225</td>
<td>SMBO</td>
</tr>
<tr>
<td>ENAS [34]</td>
<td>2.89</td>
<td>4.6</td>
<td>0.5</td>
<td>RL</td>
</tr>
<tr>
<td>NAONet-WS [31]</td>
<td>3.53</td>
<td>3.1</td>
<td>0.4</td>
<td>NAO</td>
</tr>
<tr>
<td>DARTS (1st order) [29]</td>
<td>3.00±0.14</td>
<td>3.3</td>
<td>0.4</td>
<td>gradient</td>
</tr>
<tr>
<td>DARTS (2nd order) [29]</td>
<td>2.76±0.09</td>
<td>3.3</td>
<td>1</td>
<td>gradient</td>
</tr>
<tr>
<td>SNAS (mild) [49]</td>
<td>2.98</td>
<td>2.9</td>
<td>1.5</td>
<td>gradient</td>
</tr>
<tr>
<td>ProxynasNAS [7]</td>
<td>2.08</td>
<td>-</td>
<td>4</td>
<td>gradient</td>
</tr>
<tr>
<td>P-DARTS [8]</td>
<td>2.5</td>
<td>3.4</td>
<td>0.3</td>
<td>gradient</td>
</tr>
<tr>
<td>BayesNAS [52]</td>
<td>2.81±0.04</td>
<td>3.4</td>
<td>0.2</td>
<td>gradient</td>
</tr>
<tr>
<td>PC-DARTS [50]</td>
<td>2.57±0.07</td>
<td>3.6</td>
<td>0.1</td>
<td>gradient</td>
</tr>
<tr>
<td>SGAS (Cri.1 avg.)</td>
<td>2.66±0.24*</td>
<td>3.7</td>
<td>0.25</td>
<td>gradient</td>
</tr>
<tr>
<td>SGAS (Cri.1 best)</td>
<td>2.39</td>
<td>3.8</td>
<td>0.25</td>
<td>gradient</td>
</tr>
<tr>
<td>SGAS (Cri.2 avg.)</td>
<td>2.67±0.21*</td>
<td>3.9</td>
<td>0.25</td>
<td>gradient</td>
</tr>
<tr>
<td>SGAS (Cri.2 best)</td>
<td>2.44</td>
<td>4.1</td>
<td>0.25</td>
<td>gradient</td>
</tr>
</tbody>
</table>

Table 3. Performance comparison with state-of-the-art image classifiers on CIFAR-10.
SGAS for CNN on CIFAR-10

(a) Normal cell of the best model with SGAS (Cri. 1) on CIFAR-10

(b) Reduction cell of the best model with SGAS (Cri. 1) on CIFAR-10

(c) Normal cell of the best model with SGAS (Cri. 2) on CIFAR-10

(d) Reduction cell of the best model with SGAS (Cri. 2) on CIFAR-10
Table 4. Performance comparison with state-of-the-art image classifiers on ImageNet.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Err. (%)</th>
<th>Params (M)</th>
<th>Search Cost (GPU-days)</th>
<th>Search Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>top-1</td>
<td>top-5</td>
<td>×+</td>
<td></td>
</tr>
<tr>
<td>Inception-v1 [41]</td>
<td>30.2</td>
<td>10.1</td>
<td>6.6</td>
<td>1448</td>
</tr>
<tr>
<td>MobileNet [16]</td>
<td>29.4</td>
<td>10.5</td>
<td>4.2</td>
<td>569</td>
</tr>
<tr>
<td>ShuffleNet 2x (v1) [51]</td>
<td>26.4</td>
<td>10.2</td>
<td>~5</td>
<td>524</td>
</tr>
<tr>
<td>ShuffleNet 2x (v2) [32]</td>
<td>25.1</td>
<td>-</td>
<td>~5</td>
<td>591</td>
</tr>
<tr>
<td>NASNet-A [55]</td>
<td>26</td>
<td>8.4</td>
<td>5.3</td>
<td>564</td>
</tr>
<tr>
<td>NASNet-B [55]</td>
<td>27.2</td>
<td>8.7</td>
<td>5.3</td>
<td>488</td>
</tr>
<tr>
<td>NASNet-C [55]</td>
<td>27.5</td>
<td>9</td>
<td>4.9</td>
<td>558</td>
</tr>
<tr>
<td>AmoebaNet-A [36]</td>
<td>25.5</td>
<td>8</td>
<td>5.1</td>
<td>555</td>
</tr>
<tr>
<td>AmoebaNet-B [36]</td>
<td>26</td>
<td>8.5</td>
<td>5.3</td>
<td>555</td>
</tr>
<tr>
<td>AmoebaNet-C [36]</td>
<td>24.3</td>
<td>7.6</td>
<td>6.4</td>
<td>570</td>
</tr>
<tr>
<td>PNAS [27]</td>
<td>25.8</td>
<td>8.1</td>
<td>5.1</td>
<td>588</td>
</tr>
<tr>
<td>MnasNet-92 [42]</td>
<td>25.2</td>
<td>8</td>
<td>4.4</td>
<td>388</td>
</tr>
<tr>
<td>DARTS (2nd order) [29]</td>
<td>26.7</td>
<td>8.7</td>
<td>4.7</td>
<td>574</td>
</tr>
<tr>
<td>SNAS (mild) [49]</td>
<td>27.3</td>
<td>9.2</td>
<td>4.3</td>
<td>522</td>
</tr>
<tr>
<td>ProxylessNAS [7]</td>
<td>24.9</td>
<td>7.5</td>
<td>7.1</td>
<td>465</td>
</tr>
<tr>
<td>P-DARTS [8]</td>
<td>24.4</td>
<td>7.4</td>
<td>4.9</td>
<td>557</td>
</tr>
<tr>
<td>BayesNAS [52]</td>
<td>26.5</td>
<td>8.9</td>
<td>3.9</td>
<td>-</td>
</tr>
<tr>
<td>PC-DARTS [50]</td>
<td>25.1</td>
<td>7.8</td>
<td>5.3</td>
<td>586</td>
</tr>
<tr>
<td>SGAS (Cri.1 avg.)</td>
<td>24.4±0.2</td>
<td>7.3±0.1</td>
<td>5.3</td>
<td>579</td>
</tr>
<tr>
<td>SGAS (Cri.1 best)</td>
<td>24.2</td>
<td>7.2</td>
<td>5.3</td>
<td>585</td>
</tr>
<tr>
<td>SGAS (Cri.2 avg.)</td>
<td>24.4±0.2</td>
<td>7.4±0.1</td>
<td>5.4</td>
<td>597</td>
</tr>
<tr>
<td>SGAS (Cri.2 best)</td>
<td>24.1</td>
<td>7.3</td>
<td>5.4</td>
<td>598</td>
</tr>
</tbody>
</table>
SGAS for CNN on ImageNet

(a) Normal cell of the best model with SGAS (Cri. 1) on ImageNet

(b) Reduction cell of the best model with SGAS (Cri. 1) on ImageNet

(c) Normal cell of the best model with SGAS (Cri. 2) on ImageNet

(d) Reduction cell of the best model with SGAS (Cri. 2) on ImageNet
SGAS for GCN

Kipf, T.N. and Welling, M., 2016. Semi-Supervised Classification with Graph Convolutional Networks.


SemiGCN, GraphSage, GAT, EdgeConv, DeepGCNs, … are manually designed

Search Space:
skip-connect, zero operation, conv-1×1, MRConv, EdgeConv, GAT, SemiGCN, GIN, SAGE, RelSAGE,
Table 1. Comparison with state-of-the-art architectures for 3D object classification on ModelNet40.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>OA (%)</th>
<th>Params (M)</th>
<th>Search Cost (GPU-days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DmFV-Net [3]</td>
<td>91.6</td>
<td>45.77</td>
<td>manual</td>
</tr>
<tr>
<td>SpecGCN [46]</td>
<td>91.5</td>
<td>2.05</td>
<td>manual</td>
</tr>
<tr>
<td>PointNet++ [37]</td>
<td>90.7</td>
<td>1.48</td>
<td>manual</td>
</tr>
<tr>
<td>PCNN [2]</td>
<td>92.3</td>
<td>8.2</td>
<td>manual</td>
</tr>
<tr>
<td>PointCNN [25]</td>
<td>92.2</td>
<td>0.6</td>
<td>manual</td>
</tr>
<tr>
<td>DGCNN [47]</td>
<td>92.2</td>
<td>1.84</td>
<td>manual</td>
</tr>
<tr>
<td>KPConv [44]</td>
<td>92.9</td>
<td>14.3</td>
<td>manual</td>
</tr>
<tr>
<td>Random Search</td>
<td>92.65±0.33</td>
<td>8.77</td>
<td>random</td>
</tr>
<tr>
<td>SGAS (Cri.1 avg.)</td>
<td>92.69±0.20</td>
<td>8.78</td>
<td>0.19</td>
</tr>
<tr>
<td>SGAS (Cri.1 best)</td>
<td>92.87</td>
<td>8.63</td>
<td>0.19</td>
</tr>
<tr>
<td>SGAS (Cri.2 avg.)</td>
<td>92.92±0.19</td>
<td>8.87</td>
<td>0.19</td>
</tr>
<tr>
<td>SGAS (Cri.2 best)</td>
<td><strong>93.23</strong></td>
<td>8.49</td>
<td>0.19</td>
</tr>
<tr>
<td>SGAS (Cri.2 small best)</td>
<td>93.07</td>
<td>3.86</td>
<td>0.19</td>
</tr>
</tbody>
</table>

(a) Normal cell of the best model with SGAS (Cri. 1) on ModelNet

(b) Normal cell of the best model with SGAS (Cri. 2) on ModelNet
**Table 2. Comparison with state-of-the-art architectures for node classification on PPI.**
Conclusion

**Greedy Fashion:**
- Alleviate the degenerate search-evaluation correlation problem
- Very Fast and less GPU memory usage
- Subproblems are easier to optimize, which leads to a better solution

**Selection Criteria:** Maintain optimality in the search space

**Application:**
- Design CNN architectures (CIFAR-10 & ImageNet)
- Design GCN architectures (ModelNet & PPI)
Team

Guohao Li  Guocheng Qian  Itzel C. Delgadillo

Matthias Müller  Ali Thabet  Bernard Ghanem (PI)
Project Links

- **Arxiv:** https://arxiv.org/abs/1912.00195
- **Project webpage:** https://www.deepgcns.org/auto/sgas
- **Code:** https://github.com/lightaime/sgas
SGAS: Sequential Greedy Architecture Search

More information see my personal website https://ghli.org