SGAS: Sequential Greedy Architecture Search

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

DeepGCNs.org
Overview

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?
Design a search strategy instead of an instance of architecture

- **Given:** A Search Space
- **Goal:** Find The Best Arch.
- **Approach:** Design A Good Search Strategy
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...
NAS - Gradient Based

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

Liu H, Simonyan K, Yang Y. DARTS: Differentiable Architecture Search (ICLR’2019)
DARTS

Softmax Over All Possible ops:

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

Select The Most Likely op:

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

Only 1 GPU Day!

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

Liu H, Simonyan K, Yang Y. DARTS: Differentiable Architecture Search (ICLR’2019)
Existing Problems

**DARTS**

![Diagram of DARTS architecture]

**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)

Liu H, Simonyan K, Yang Y. DARTS: Differentiable Architecture Search (ICLR’2019)
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.

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 \((i^+, j^+)\) based on the greedy Selection Criterion

2. Determine the operation by replacing \(\tilde{o}(i^+, j^+)\) with \(o(i^+, j^+) = \arg\max_{o \in \mathcal{O}} \alpha_o(i^+, j^+)\)
SGAS - Greedy Decision

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 \(o(i^t, j^t) = \arg\max_{o \in \mathcal{O}} \alpha_o(i^t, j^t)\).

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 \((i^\dagger, j^\dagger)\) based on the greedy Selection Criterion
2. Determine the operation by replacing \(\overline{o}(i^\dagger, j^\dagger)\) with \(o(i^\dagger, j^\dagger) = \arg\max_{o \in O} \alpha_o(i^\dagger, j^\dagger)\)
3. Prune unchosen weights from \(W\), Remove \(\alpha(i^\dagger, j^\dagger)\) from \(A\)
SGAS: Sequential Greedy Architecture Search (CVPR’2020)

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 \(\bar{o}(i^\dagger, j^\dagger)\) with \(o(i^\dagger, j^\dagger) = \arg\max_{o \in \mathcal{O}} a_o(i^\dagger, j^\dagger)\)
3. Prune unchosen weights from \(\mathcal{W}\), Remove \(a(i^\dagger, j^\dagger)\) 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 \(\bar{o}(i^+, j^+)\) 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 \(\mathcal{A}\)
1. If a decision epoch, select an edge \((i^\dagger, j^\dagger)\) based on the greedy Selection Criterion

2. Determine the operation by replacing \(o(i^\dagger, j^\dagger)\) with \(\alpha(i^\dagger, j^\dagger)\) = \(\arg\max_{o \in O} \alpha_o(i^\dagger, j^\dagger)\)

3. Prune unchosen weights from \(\mathcal{W}\), Remove \(\alpha(i^\dagger, j^\dagger)\) from \(\mathcal{A}\)

Figure 2. Illustration of Sequential Greedy Architecture Search.
SGAS - Selection Criteria

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:**

$$S_{SS}^{(i,j)} = \frac{1}{K} \sum_{t=T-K}^{T-1} \sum_{o_t \in \mathcal{O}_t, \neq \text{zero}} \min(p_{o_t}^{(i,j)}, p_o^{(i,j)})$$
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:**

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

**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:**

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

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
SGAS for CNN on ImageNet

| Architecture        | Test Err. (%) |
|---------------------|---------------|
|                     | top-1 | top-5 | Params (M) | ×+ (M) | Search Cost (GPU-days) | Search Method |
| Inception-v1 [41]   | 30.2  | 10.1  | 6.6  | 1448 | - | manual |
| MobileNet [16]      | 29.4  | 10.5  | 4.2  | 569  | - | manual |
| ShuffleNet 2x (v1) [51] | 26.4  | 10.2  | ~5   | 524  | - | manual |
| ShuffleNet 2x (v2) [32] | 25.1  | -     | ~5   | 591  | - | manual |
| NASNet-A [55]       | 26    | 8.4   | 5.3  | 564  | 1800 | RL |
| NASNet-B [55]       | 27.2  | 8.7   | 5.3  | 488  | 1800 | RL |
| NASNet-C [55]       | 27.5  | 9     | 4.9  | 558  | 1800 | RL |
| AmoebaNet-A [36]    | 25.5  | 8     | 5.1  | 555  | 3150 | evolution |
| AmoebaNet-B [36]    | 26    | 8.5   | 5.3  | 555  | 3150 | evolution |
| AmoebaNet-C [36]    | 24.3  | 7.6   | 6.4  | 570  | 3150 | evolution |
| PNAS [27]           | 25.8  | 8.1   | 5.1  | 588  | 225  | SMBO |
| MnasNet-92 [42]     | 25.2  | 8     | 4.4  | 388  | - | RL |
| DARTS (2nd order) [29] | 26.7  | 8.7   | 4.7  | 574  | 4.0 | gradient |
| SNAS (mild) [49]    | 27.3  | 9.2   | 4.3  | 522  | 1.5 | gradient |
| ProxylessNAS [7]    | 24.9  | 7.5   | 7.1  | 465  | 8.3 | gradient |
| P-DARTS [8]         | 24.4  | 7.4   | 4.9  | 557  | 0.3 | gradient |
| BayesNAS [52]       | 26.5  | 8.9   | 3.9  | -    | 0.2 | gradient |
| PC-DARTS [50]       | 25.1  | 7.8   | 5.3  | 586  | 0.1 | gradient |
| SGAS (Cri.1 avg.)   | 24.4 ±0.2 | 7.3 ±0.1 | 5.3  | 579  | 0.25 | gradient |
| SGAS (Cri.1 best)   | 24.2  | 7.2   | 5.3  | 585  | 0.25 | gradient |
| SGAS (Cri.2 avg.)   | 24.4 ±0.2 | 7.4 ±0.1 | 5.4  | 597  | 0.25 | gradient |
| SGAS (Cri.2 best)   | **24.1** | 7.3   | 5.4  | 598  | 0.25 | gradient |

Table 4. Performance comparison with state-of-the-art image classifiers on ImageNet.
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.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. and Bengio, Y., 2018. Graph Attention Networks.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M., 2018. Dynamic Graph CNN for Learning on Point Clouds.

Hamilton, W.L., Ying, R. and Leskovec, J., 2017. Inductive Representation Learning on Large Graphs.

LI, G., Müller, M., Thabet, A. and Ghanem, B., 2019. DeepGCNs: Can GCNs Go as Deep as CNNs?

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 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)

DeepGCNs.org
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