Midway Presentation

CSED 490Y: Optimization for Machine Learning

Jiwoong Shin

2022.05.18

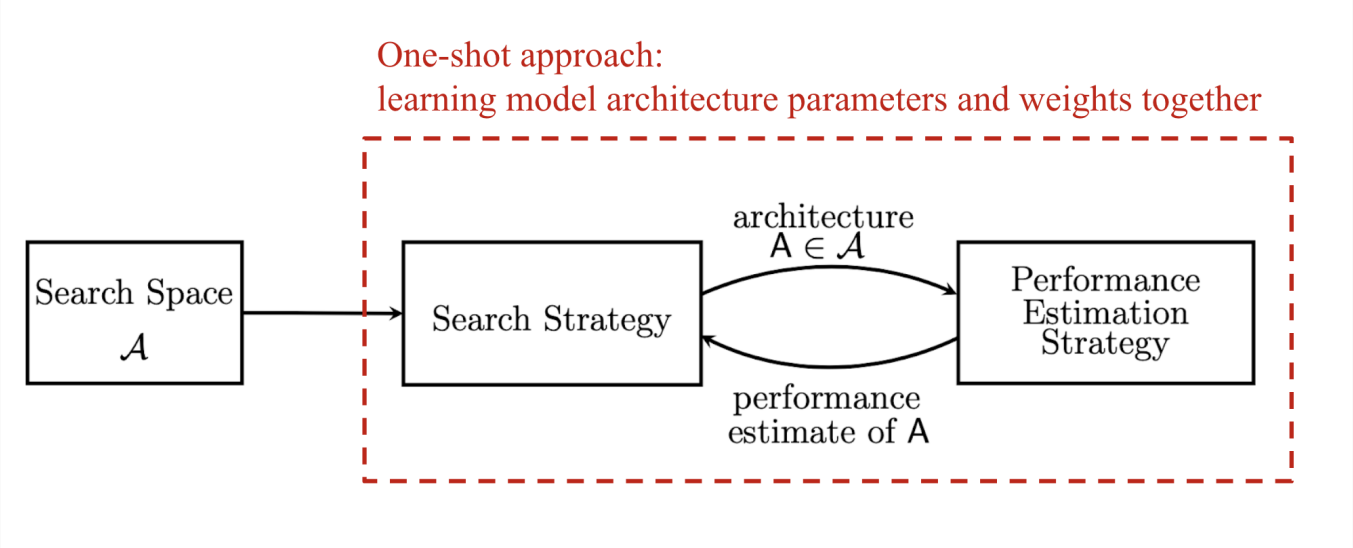
Group 9

AutoML

AutoML is a concept to automatically design machine learning method by machine learning.

1. Automated Feature Extraction
    - Find meaningful features, but not useful for deep learning
2. **Architecture Search**
    - Find optimal network structure
3. Hyperparameter Optimization
    - Predict hyperparameters such as batch size, learning rate, etc.

Neural Architecture Search (NAS)



Neural Architecture Search is a concept to automatically search neural architecture in specific search space.

- **Search Space**
  - All candidate architectures
- **Search Strategy**
  - How to search?
- **Performance Estimation Strategy**
  - How to evaluate performance of an architecture?

Early Stage NAS

In early stage NAS, because of **discrete objective of NAS**, many frameworks use reinforcement learning[7, 8, 4] or evolutionary algorithm[5].

However, these methods require **a lots of resources and time**.

- NASRL [7]
  - 800 GPUs for 1 month
- NASNet [8]
  - 450 GPUs for 3 days
- AmoebaNet [5]
  - 3150 GPU days

DARTS: Differentiable ARchiTecture Search

Continuous Relaxation

Neural Architecture Search is originally discrete optimization problem.

- We should select architecture components for a network!

How about relax this constraint?

- Discrete optimization $\rightarrow$ **Continuous optimization**
- Use continuous parameters to represent architecture selection.
- Convert continuous parameters to an discrete neural architecture in final step.

This relaxation significantly reduce the search cost!

Neural Architecture Search Formulation

With relaxed constraints, we can formulate neural architecture search as follows.

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

- w: weight parameters
- α: architecture parameters
- $w^*(\alpha)$: optimal weight parameters for architecture α

Supernet

However, structure of neural networks are various.

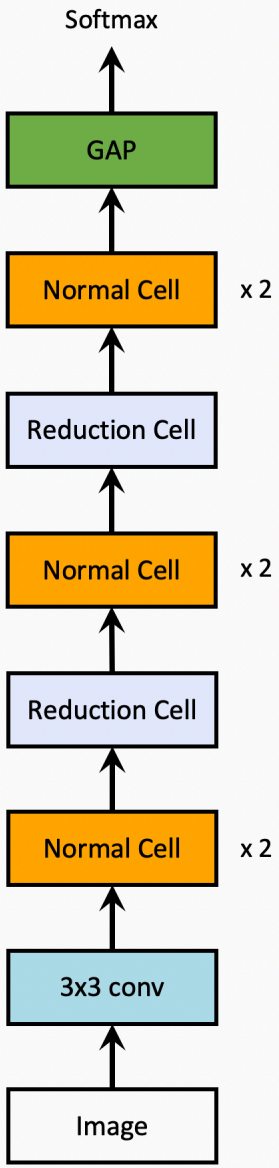
- How can we determine the structure with α?

We restrict our search space to **supernet**.

- Train single network which is superposition of various networks
- Can make an architecture by selecting partial components from supernet.

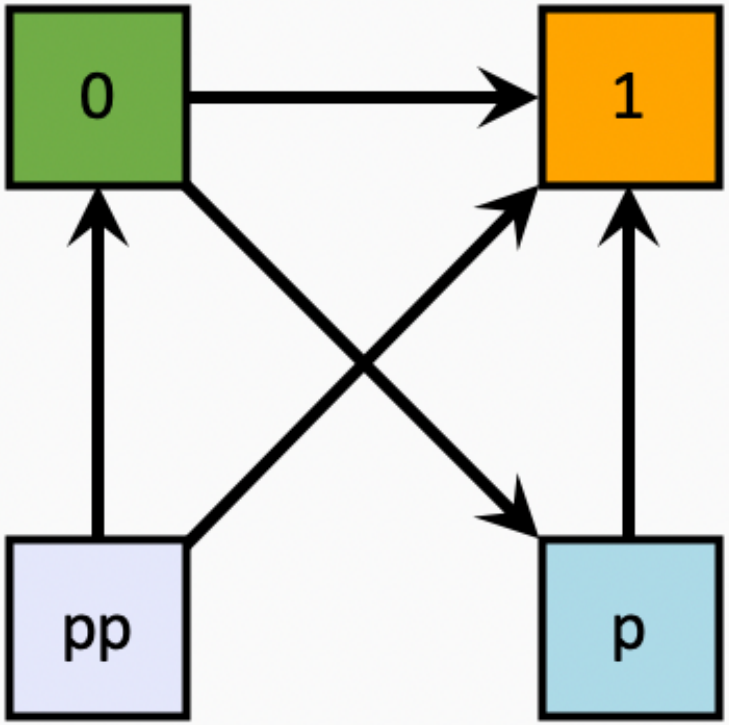
Supernet (Network-level)

- Normal Cell
  - Maintain resolution
  - Maintain the number of channels
- Reduction Cell
  - Reduce resolution by 1/4
  - Double the number of channels



CIFAR10
Architecture

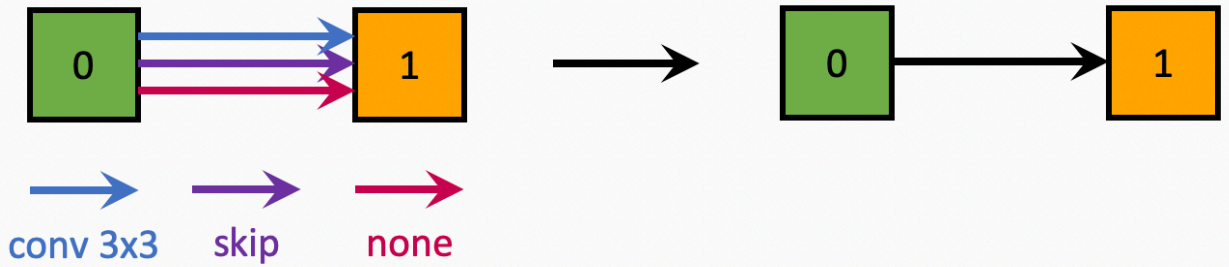
Supernet (Cell-level)



Each cell is **directed acyclic graph (DAG)**.

- Node: intermediate feature map
- Edge: mixed operation - weighted sum of candidate operations

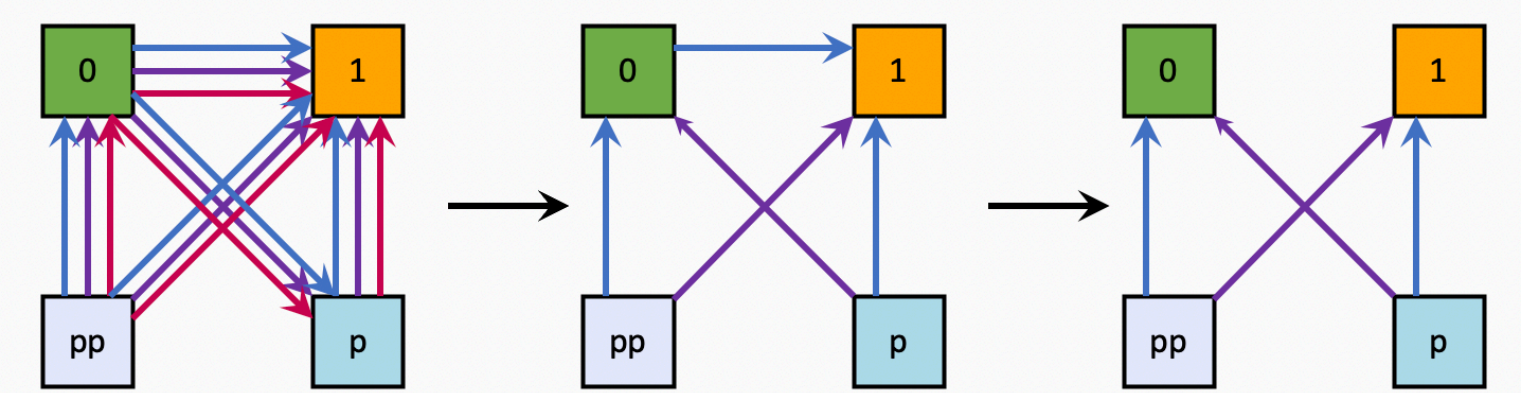
Mixed Operation



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

- $\mathcal{O}$: operation set
- o: operation
  - e.g. sep conv 3x3, skip connection, zeroize
- α: architecture parameters
- $\alpha_o^{(i,j)}$: architecture parameter for operation o of (i, j) edge

Architecture Derivation



Cell derivation

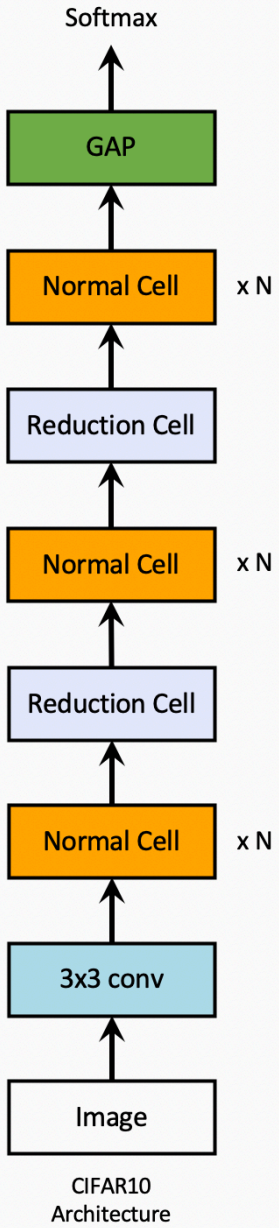
After train the supernet...

1. For each edge, choose operation o which has the largest architecture parameter α_o
2. For each node, choose operation which has top-2 architecture parameter.

Architecture Derivation

Architecture Construction

- By stacking searched cells, we can get an architecture.
  - e.g. $N = 6$ in the paper



Issue in Gradient Descent Based Optimization

$$\min_{\alpha} \ \mathcal{L}_{val}(w^*(\alpha), \alpha)$$

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

Now we have proper model and objective.

Can we optimize original objective with gradient descent?

- **No!**
- Formulation for NAS can not be directly used!
  - The number of candidates architecture is too many.
  - Training of each architecture is not easy.

Gradient Approximation

To directly use architecture gradient $\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha)$ is infeasible.

So, let approximate the gradient $\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha)$!

- Assumption 1: $w^* = w$
  - Current weight parameter is optimal weight parameter.
  - Easy and low cost optimization process

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

- Assumption 2: $w^* = w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha)$
  - By single training step, we can get optimal weight parameters.
  - Complex and high cost optimization process

$$\begin{aligned}\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) &\approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha) \\ &= \nabla_\alpha \mathcal{L}_{val}(w', \alpha) - \xi \nabla^2_{\alpha, w} \mathcal{L}_{train}(w, \alpha) \nabla_{w'} \mathcal{L}(w', \alpha)\end{aligned}$$

  - $w' = w - \xi \nabla \mathcal{L}_{train}(w, \alpha)$

Experiment Results

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10 (lower error rate is better). Note the search cost for DARTS does not include the selection cost (1 GPU day) or the final evaluation cost by training the selected architecture from scratch (1.5 GPU days).

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

Optimizer Combination Analysis for Differentiable Neural Architecture Search

Fixed Optimizer Setting

DARTS[3] and its variants[1, 2] use **fixed optimizer setting** to search architecture in supernet.

- weight parameters: SGD
- architecture parameters: Adam

However, is it the optimal choice? Is there any room for improvement?

- No specific experiments on this
- No theoretical guarantee

Plan for Project

Some study [6] show that the Adam optimizer makes some regularization techniques meaningless, which reduces the generalization performance of the architecture being explored.

To extend this, I will **investigate effect of optimizer setting in architecture search** with original DARTS[3], Smooth DARTS[1] and DARTS-[2].

- Momentum
  - e.g SGD with momentum
- Adaptive gradient methods
  - e.g. RMSProp, Adam

References i

X. Chen and C. Hsieh.
Stabilizing differentiable architecture search via perturbation-based regularization.
In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1554–1565. PMLR, 2020.

X. Chu, X. Wang, B. Zhang, S. Lu, X. Wei, and J. Yan.
DARTS-: robustly stepping out of performance collapse without indicators.
In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

References ii

H. Liu, K. Simonyan, and Y. Yang.
DARTS: differentiable architecture search.
In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean.
Efficient neural architecture search via parameter sharing.
In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4092–4101. PMLR, 2018.

References iii

E. Real, A. Aggarwal, Y. Huang, and Q. V. Le.
Regularized evolution for image classifier architecture search.
In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4780–4789. AAAI Press, 2019.

P. Ye, B. Li, Y. Li, T. Chen, J. Fan, and W. Ouyang.
β-darts: Beta-decay regularization for differentiable architecture search.
CoRR, abs/2203.01665, 2022.

References iv

B. Zoph and Q. V. Le.
Neural architecture search with reinforcement learning.
In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le.
Learning transferable architectures for scalable image recognition.
In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8697–8710. Computer Vision Foundation / IEEE Computer Society, 2018.