# Adaptive Stochastic Natural Gradient Method for One-Shot Neural Architecture Search

○Youhei Akimoto (University of Tsukuba / RIKEN AIP)
Shinichi Shirakawa (Yokohama National University)
Nozomu Yoshinari (Yokohama National University)
Kento Uchida (Yokohama National University)
Shota Saito (Yokohama National University)
Kouhei Nishida (Shinshu University)

# Neural Architecture

## Neural Network Architectures

VGGNet

ResNet

Inception

...

often pre-trained
on some datasets


Sometimes...

- a known architecture works well on our tasks. Happy!

Other times...

- Find a good one
- Design a brand-new architecture and train it

Trial and Error!

# One-Shot Neural Architecture Search

## Joint Optimization of Architecture $c$ and Weights $w$

NAS as hyper-parameter search

$$\max_{c} f(w^*(c), c)$$

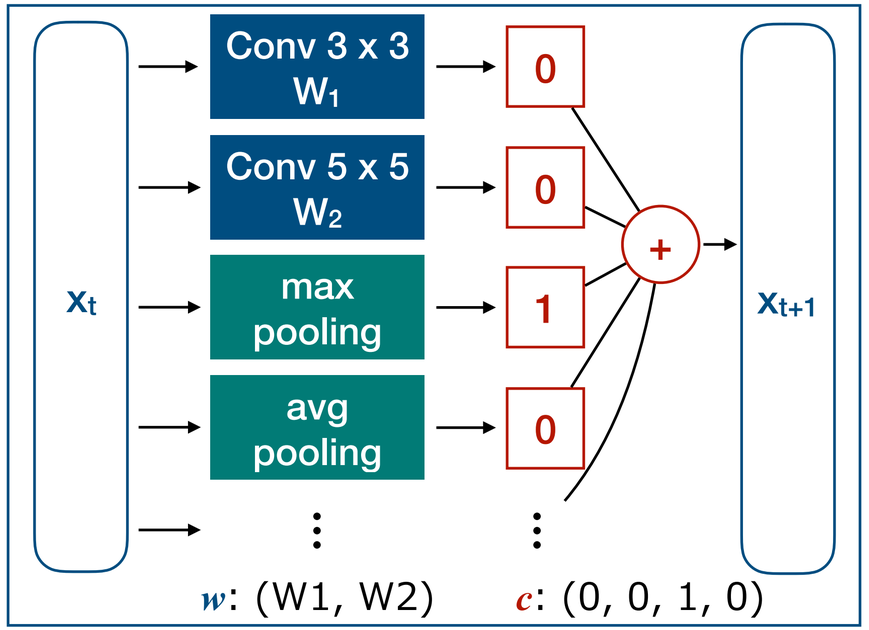
$$\text{subject to } w^*(c) = \underset{w}{\text{argmax}}\, f(w, c)$$

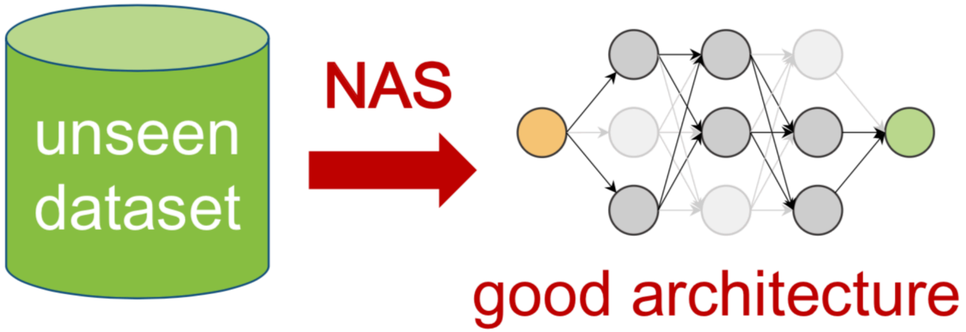
1 c evaluation
= 1 training

One-shot NAS

$$\max_{w,c} f(w, c)$$

optimization of x and
c within 1 training

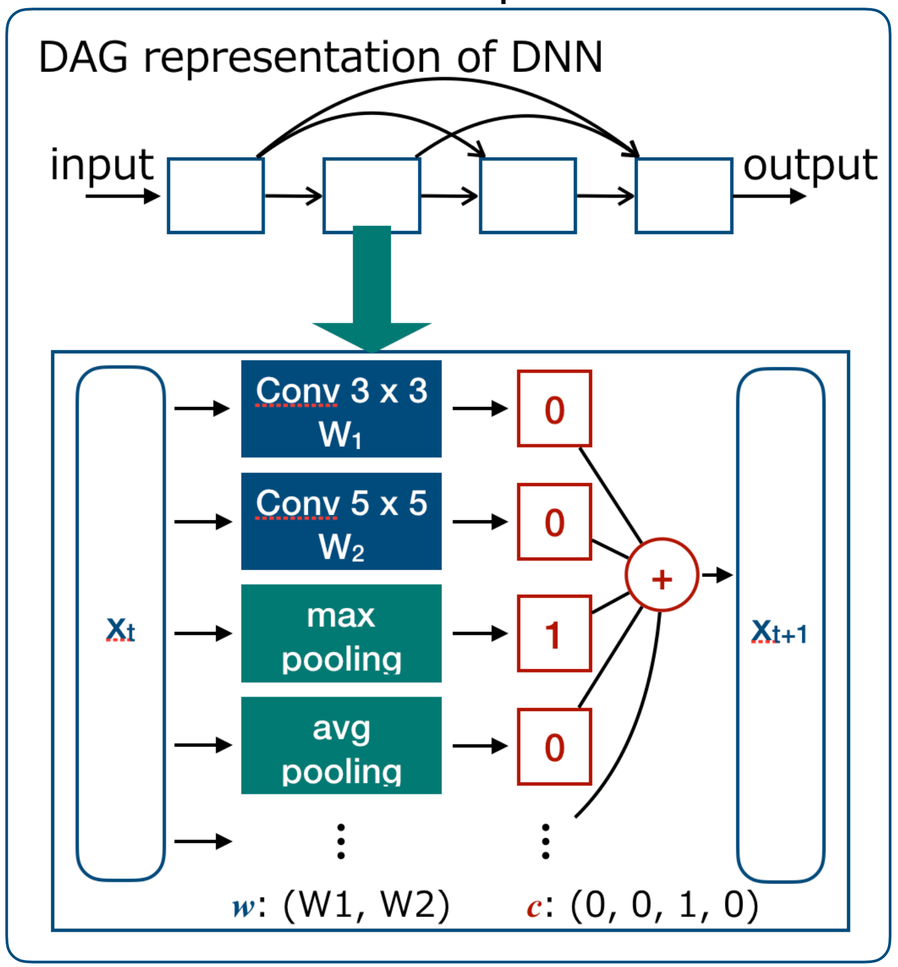



# Difficulties for Practitioners

## How to choose / tune the search strategy?

Search Space




Search Strategy

Gradient-Based Method

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \epsilon_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} f(\boldsymbol{w}, \boldsymbol{c}(\boldsymbol{\theta}))$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon_{\boldsymbol{\theta}} \nabla_{\theta} f(\boldsymbol{w}, \boldsymbol{c}(\boldsymbol{\theta}))$$

hyper-parameter: step-size

Other Choices

- Evolutionary Computation Based
- Reinforcement Learning Based

- how to treat integer variables such as #filters?
- how to tune the hyper-parameters in such situations?

# Contributions

## Novel Search Strategy for One-shot NAS

1. arbitrary search space (categorical + ordinal)
2. robust against its inputs (hyper-param. and search space)

**Our approach**

1. Stochastic Relaxation

$$\max_{\boldsymbol{w},\boldsymbol{c}} f(\boldsymbol{w},\boldsymbol{c}) \Rightarrow \max_{\boldsymbol{w},\boldsymbol{\theta}} J(\boldsymbol{w},\boldsymbol{\theta}) := \int f(\boldsymbol{w},\boldsymbol{c}) p(\boldsymbol{c} \mid \boldsymbol{\theta}) d\boldsymbol{c}$$

exponential family

differentiable w.r.t. w and θ

2. Stochastic Natural Gradient + Adaptive Step-Size

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \epsilon_{\boldsymbol{w}}^t \nabla_{\boldsymbol{w}} \widehat{J(\boldsymbol{w}^t, \boldsymbol{\theta}^t)}$$

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \epsilon_{\boldsymbol{\theta}}^t \mathbf{F}(\boldsymbol{\theta}^t)^{-1} \nabla_{\boldsymbol{\theta}} \widehat{J(\boldsymbol{w}^{t+1}, \boldsymbol{\theta}^t)}$$

Natural Gradient

➡ Under appropriate step-size

$$J(\boldsymbol{w}^t, \boldsymbol{\theta}^t) < J(\boldsymbol{w}^{t+1}, \boldsymbol{\theta}^t) < J(\boldsymbol{w}^{t+1}, \boldsymbol{\theta}^{t+1})$$

Monotone Improvement

# Results and Details

- Faster & Competitive Accuracy to other one-shot NAS

Table 1: Comparison of different architecture search methods on CIFAR-10. The search cost indicates GPU days for architecture search excluding the retraining cost.

| Method | Search Cost (GPU days) | Params (M) | Test Error (%) |
|---|---|---|---|
| NASNet-A (Zoph et al., 2018) | 1800 | 3.3 | 2.65 |
| NAONet (Luo et al., 2018) | 200 | 128 | 2.11 |
| ProxylessNAS-G (Cai et al., 2019) | 4 | 5.7 | 2.08 |
| SMASHv2 (Brock et al., 2018) | 1.5 | 16.0 | 4.03 |
| DARTS second order (Liu et al., 2019) | 4 | 3.3 | 2.76 ($\pm$0.09) |
| DARTS first order (Liu et al., 2019) | 1.5 | 3.3 | 3.00 ($\pm$0.14) |
| SNAS (Xie et al., 2019) | 1.5 | 2.8 | 2.85 ($\pm$0.02) |
| ENAS (Pham et al., 2018) | 0.45 | 4.6 | 2.89 |
| ASNG-NAS | 0.11 | 3.9 | 2.83 ($\pm$0.14) |

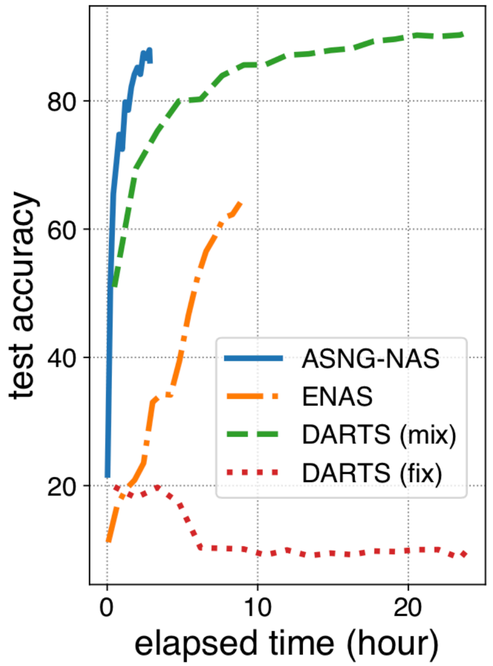



Figure 2: Transitions of test error against elapsed time in the architecture search phase.

The detail will be explained at Poster #53