ECCV 2020 Tutorial on

From HPO to NAS: Automated Deep Learning

Hang Zhang[1], Linjie Yang[2], Peizhao Zhang[3], Matthias Seeger[1],
Mi Zhang[4], Haotian Tang[5], Zhijian Liu[5], Song Han[5]

Amazon[1], ByteDance[2], Facebook[3], MSU[4], MIT[5]

Instructors

Hang Zhang
Amazon

Linjie Yang
ByteDance

Peizhao Zhang
Facebook

Matthias Seeger
Amazon

Mi Zhang
MSU

Haotian Tang
MIT

Zhijian Liu
MIT

Song Han
MIT

CVPR Tutorial Review https://hangzhang.org/CVPR2020/

- AutoML Background Overview (by Cedric Archambeau)
- Introduction to an AutoML Toolkit (by Hang Zhang)
- Once-for-All Network (by Song Han)
- Hands-on Sections:
  - ProxylessNAS
  - OFA-Net
  - Fast AutoAugment

NAS: Neural Architecture Search

- What is NAS?

  We want the architecture that maximize the reward with the corresponding network weights that minimize training loss.

$$\max_{a} R(\hat{w}(a), a) \text{ st. } \hat{w}(a) = \text{argmin}_{w} \ell_{train}(w, a)$$

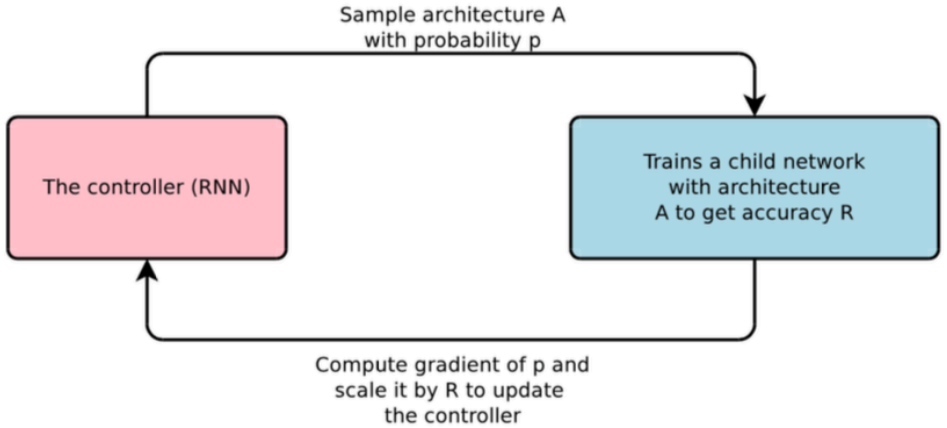
Different Way of Approaching NAS

- NAS with RL:
  Sampling a set of configurations and calculate reward and update θ

$$J(\theta) = E_{a \sim P(a;\theta)}[R(a)]$$

$$\frac{1}{m}\sum_{k=1}^{m} \log P(a_k;\theta)(R(a_k) - b)$$

- Policy improves over time



Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning."

Reinforce Algorithm http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture14.pdf

NASNet, MobileNetV3, MNASNet, AutoDeepLab

http://autogluon.s3.amazonaws.com/tutorials/course/object.html

Different Way of Approaching NAS

- Differentiable Architecture Search:

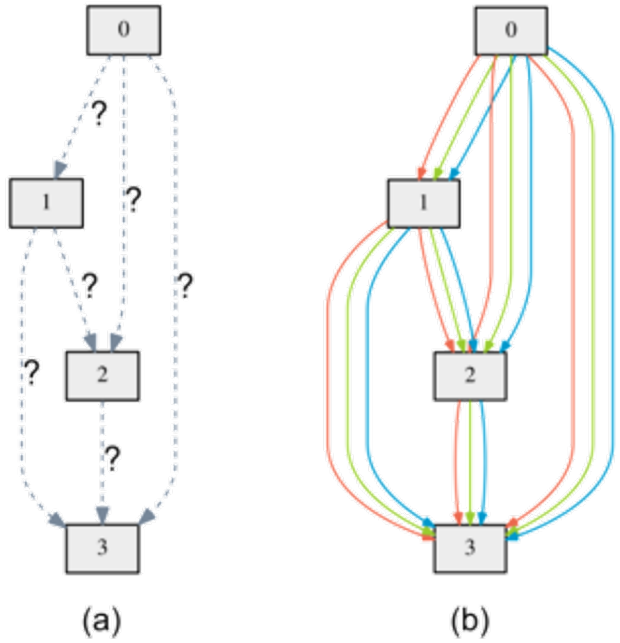
$$\mathrm{J}(\theta) = E_{a\sim P(a;\,\theta)}[R(a)] \approx R(E_{P(a;\theta)}[a])$$

- Soft combination of different ops:

$$O(x) = \sum_i softmax(\theta_i) \cdot o_i(x)$$

- $\mathrm{GumbleSoftMax}(\theta_i) = \dfrac{\exp(\theta_i + g_i)/\tau}{\sum_{\mathrm{j}} \exp(\theta_j + g_j)/\tau}$

- Difficulties: memory and computation



Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search.
Xie, Sirui, et al. "SNAS: stochastic neural architecture search."
Wu, Bichen, et al. "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search."

Scalable &Efficient Neural Architecture Search

- ENAS via parameter sharing (reward per-epoch)

$$\mathrm{J}(\theta) = E_{a\sim P(a;\,\theta)}[R(a, w^*)]$$

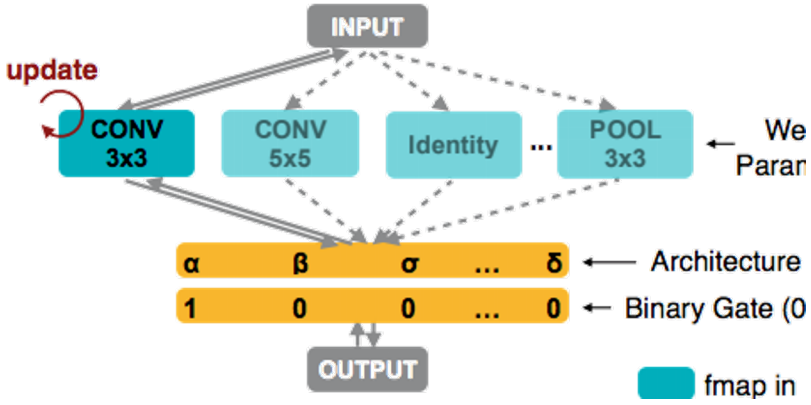
- ProxylessNAS: $g = [0,0..,1,\dots 0]$

$$O(x) = \sum_i g_i(\theta_i)\cdot o_i(x)$$

$$\frac{d\ell}{d_{\theta_j}} = \sum_i \frac{d\ell}{d_{g_i}}\cdot\frac{d_{g_i}}{d_{\theta_j}} \approx \sum_i \frac{d\ell}{d_{g_i}}\cdot\frac{d_{p_i}}{d_{\theta_j}}$$

$$\frac{d\ell}{d_{g_i}} = \frac{d\ell}{d_{O(x)}}\cdot\frac{d_{O(x)}}{d_{g_i}}$$

$$\frac{d_{O(x)}}{d_{g_i}} := o_i(x)$$



Pham, Hieu, et al. "Efficient neural architecture search via parameter sharing."

Cai, Han, Ligeng Zhu, and Song Han. "Proxylessnas: Direct neural architecture search on target task and hardware."

Tutorial Overview

- Introduction to Neural Architecture Search for Computer Vision (by Linjie Yang)
- Hardware-aware Deep Neural Architecture Search (by Peizhao Zhang)
- Model-based Asynchronous Hyperparameter and Neural Architecture Search (by Matthias Seeger)
- Does Unsupervised Architecture Representation Learning Help Neural Architecture Search (by Mi Zhang)
- AutoML for 3D Deep Learning (by Haotian Tang)