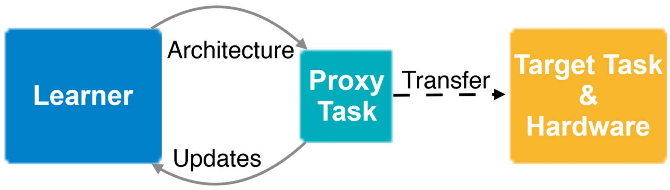
Proxy-less Architecture Search via Binarized Path Learning

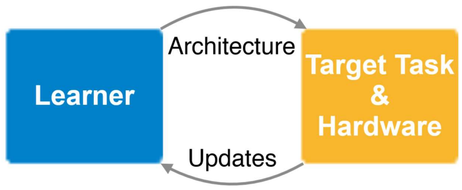
Han Cai, **Ligeng Zhu**, Song Han
Massachusetts Institute of Technology

Project Overview

(1) Previous proxy-based approach



(2) Our proxy-less approach



NAS needs to utilize **proxy** tasks due to its high cost:

- CIFAR-10 -> ImageNet
- Small arch space -> large arch space
- Fewer epochs training -> full training

Limitations

- **Suboptimal** for the target task
- Blocks are forced to **share the same structure**

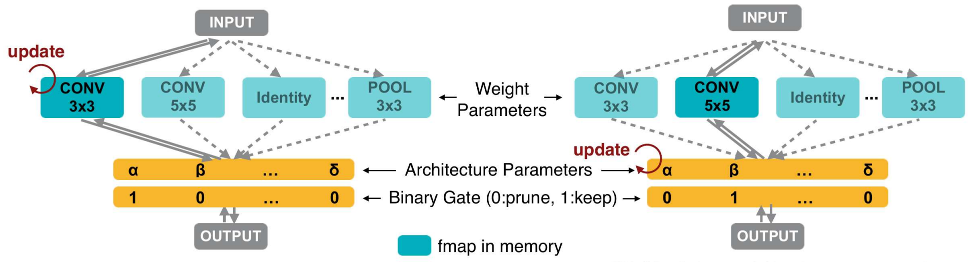
Goal: Directly learn architectures on the large-scale target task while allowing all blocks to have different structures

We achieve this by reducing the cost of NAS (GPU hours and GPU memory) to the **same level of normal training**.

Method

GPU hour-wise: Simplify NAS to be **a single training process** of a **cumbersome network**

1. Build the cumbersome network **with all candidate paths**
2. Use architecture parameters to identify and prune redundant paths (**path-level pruning**)



(1) Update weight parameters (2) Update architecture parameters

GPU memory-wise: Binarize architecture parameters and allow only one path of activation to be active in memory at run-time. Learn binarized architecture parameters via

1. Modified gradient decent based on BinaryConnect
2. REINFORCE-based algorithm for non-differentiable objectives (e.g. latency, energy and memory)

Proxyless NAS results on CIFAR-10

Model	Params	Test error
DenseNet-BC (Huang et al., 2017)	25.6M	3.46
PyramidNet (Han et al., 2017)	26.0M	3.31
Shake-Shake + c/o (DeVries & Taylor, 2017)	26.2M	2.56
PyramidNet + SD (Yamada et al., 2018)	26.0M	2.31
ENAS + c/o (Pham et al., 2018)	4.6M	2.89
DARTS + c/o (Liu et al., 2018c)	3.4M	2.83
NASNet-A + c/o (Zoph et al., 2018)	27.6M	2.40
PathLevel EAS + c/o (Cai et al., 2018b)	14.3M	2.30
AmoebaNet-B + c/o (Real et al., 2018)	34.9M	2.13
Proxyless-R + c/o (ours)	5.8M	2.30
Proxyless-G + c/o (ours)	5.7M	**2.08**

- **Directly** explore a huge space: 54 distinct blocks and $7^{54\times12} \approx 10^{547}$ possible architectures
- State-of-the-art test error with 6X fewer params

Proxyless NAS results on ImageNet

Model	Top-1	Top-5	GPU latency
MobileNetV2 (Sandler et al., 2018)	72.0	91.0	6.1ms
ShuffleNetV2 (1.5) (Ma et al., 2018)	72.6	-	7.3ms
ResNet-34 (He et al., 2016)	73.3	91.4	8.0ms
NASNet-A Zoph et al. (2018)	74.0	91.3	38.3ms
MnasNet (Tan et al., 2018)	74.0	91.8	6.1ms
Proxyless (ours)	**74.5**	**92.1**	**5.1ms**

Method	GPU hours	GPU memory
NASNet	10^4	$\mathbf{10^1}$
DARTS	$\mathbf{10^2}$	10^2
Mnas	10^4	$\mathbf{10^1}$
Ours	$\mathbf{10^2}$	$\mathbf{10^1}$

(a) Efficient CPU model found by Proxyless NAS (deep and thin, only 3x3 and 5x5, late pooling).

(b) Efficient GPU model found by Proxyless NAS (shallow and wide, many 7x7, early pooling).

Specialize Network Architectures for Different Platforms

Model	Top-1 (%)	GPU	CPU
Proxyless on GPU	74.5	5.1ms	204.0ms
Proxyless on CPU	74.6	7.4ms	134.8ms

Hardware prefers specialized models. Proxyless NAS provides an efficient, automated way to design specialized models for different hardware.

#AI Research Week

hosted by MIT-IBM Watson AI Lab

Han Cai, hancai@mit.edu
Ligeng Zhu, ligeng@mit.edu
Song Han, songhan@mit.edu