On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning

Han Cai, Tianzhe Wang, Zhanghao Wu, Kuan Wang, Ji Lin, Song Han

Massachusetts Institute of Technology
From Manual Design to Automatic Design

Use Human Expertise

Manual Architecture Design

- VGGNets
- Inception Models
- ResNets
- DenseNets
- ...

Use Machine Learning (NAS)

Automatic Architecture Search

- Reinforcement Learning
- Neuro-evolution
- Bayesian Optimization
- Monte Carlo Tree Search
- ...

Computational Resources

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
From General Design to Specialized CNN

Previous Paradigm:
One CNN for all platforms.

Sub-optimal, different in:
- Degree of parallelism
- Cache size
- Memory BW
- ...

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
From General Design to Specialized CNN

**Previous Paradigm:**
One CNN for all platforms.

**Proxyless NAS:**
Customize CNN for each platform.

ResNet → Proxyless NAS → Inception → Proxyless NAS → DenseNet → Proxyless NAS → MobileNet → Proxyless NAS → ShuffleNet → Proxyless NAS

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
Conventional NAS: Computation Expensive, thus Proxy-Based

Current neural architecture search (NAS) is VERY EXPENSIVE.

- NASNet: 48,000 GPU hours ≈ 5 years on single GPU
- DARTS: 100Gb GPU memory* ≈ 9 times of modern GPU
  *if directly search on ImageNet, like us

Therefore, previous work have to utilize proxy tasks:

- CIFAR-10 -> ImageNet
- Search a block -> stack to build a full net
- Fewer epochs training -> full training

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
Conventional NAS: Computation Expensive, thus Proxy-Based

Limitations of Proxy

• **Suboptimal** for the target task
• Blocks are forced to **share the same structure**.
• Cannot optimize for **specific hardware**.

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
Proxyless, Save GPU Hours by 200x

Goal: Directly learn architectures on the target task and hardware. We achieved by

1. Reducing the cost of NAS (GPU hours and memory) to the same level of regular training.
2. Cooperating hardware feedback (e.g. latency) into the search process.
Simplify NAS to be a single training process of an over-parameterized network. Build the cumbersome network with all candidate paths. No meta controller.

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
Save GPU Memory

Binarize the architecture parameters and allow only one path of activation in memory.

We propose gradient-based and RL methods to update the architecture parameters.

Reduce the memory footprint from $O(N)$ to $O(1)$.

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
Direct Search on Target Hardware: Making Latency Differentiable

- Mobile farm infrastructure is expensive and slow.
- Use the latency estimation model as an economical alternative
- Optimize during search stage use Gradient.

\[
\mathbb{E}[\text{Latency}] = \alpha \times F(\text{conv.3x3}) + \\
\beta \times F(\text{conv.5x5}) + \\
\sigma \times F(\text{identity}) + \\
\cdots \cdots \\
\zeta \times F(\text{pool.3x3})
\]

\[
\mathbb{E}[\text{latency}] = \sum_i \mathbb{E}[\text{latency}_i]
\]

\[
Loss = Loss_{CE} + \lambda_1 \|w\|^2 + \lambda_2 \mathbb{E}[\text{latency}]
\]
Efficiently search a model

- **Normal Train**
  - >200 GPU hours
  - >48,000 GPU hours!
  - $400 Cloud Compute Cost
  - 47 pound CO₂

- **ProxylessNAS**
  - ~$100,000 Cloud Compute Cost
  - ~11,000 pound CO₂

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19

On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019

Search an efficient model

- **Conventional NAS**
  - Much Higher Latency
  - Top1 74.5
  - Top1 74.0

- **ProxylessNAS**
  - 2% Higher Top1 Acc with Similar Latency
  - Top1 74.7
  - Top1 74.6

- **MobileNetV2 (1.4)**
  - Top1 74.6
The History of Architectures

(1) The history of finding efficient Mobile model

(2) The history of finding efficient CPU model

(3) The history of finding efficient GPU model

ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
On-Device Image Classification with Proxyless Neural Architecture Search and Quantization-Aware Fine-tuning, ICCV Workshop’2019
## Results for LPIRC

<table>
<thead>
<tr>
<th>Model</th>
<th>Setting</th>
<th>Accuracy</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoblieNetV2</td>
<td>224-0.5</td>
<td>63.7%(65.4%)</td>
<td>28ms</td>
</tr>
<tr>
<td>MobileNetV2</td>
<td>192-0.75</td>
<td>67.4%(68.7%)</td>
<td>36ms</td>
</tr>
<tr>
<td>MobileNetV2</td>
<td>160-1.0</td>
<td>67.4%(68.8%)</td>
<td>31ms</td>
</tr>
<tr>
<td>ProxylessNAS</td>
<td>224-0.5</td>
<td>65.7%(67.0%)</td>
<td>31ms</td>
</tr>
<tr>
<td>ProxylessNAS</td>
<td>160-1.0</td>
<td><strong>69.2%(70.3%)</strong></td>
<td>35ms</td>
</tr>
</tbody>
</table>

Table 1. Results of 8-bit model using different preprocessing, the number in the bracket denotes the full-precision model’s top-1 accuracy on ImageNet The latency is directly measured on Google Pixel 2. It takes only 200 GPU hours to find the specialized model with ProxylessNAS in the table.
Open-source

• Both search code and models are released on Github:

```python
# https://github.com/MIT-HAN-LAB/ProxylessNAS
from proxyless_nas import *
net = proxyless_cpu(pretrained=True)
net = proxyless_gpu(pretrained=True)
net = proxyless_mobile(pretrained=True)
```
Open-source

• ProxylessNAS is available on PyTorch Hub:

```python
# https://pytorch.org/hub/pytorch_vision_proxylessnas
import torch
target_platform = 'proxyless_mobile'
net = torch.hub.load('mit-han-lab/ProxylessNAS',
    target_platform, pretrained=True)
```
Thank you!

Hardware, AI and Neural-nets

songhan@mit.edu