NetAdaptV2: Efficient Neural Architecture Search with Fast Super-Network Training and Architecture Optimization

Tien-Ju Yang, Yi-Lun Liao, Vivienne Sze
Massachusetts Institute of Technology
CVPR 2021
Introduction

• NetAdaptV2 is a neural architecture search (NAS) algorithm that can discover high-performance networks in a short time
  • Up to 5.8x search time reduction with better accuracy on ImageNet
• NetAdaptV2
  • balances and minimizes the time of each NAS step to improve speed
  • supports non-differentiable search metrics to improve network performance
Algorithm Overview

• 1) Train a super-network by jointly training networks in the search space
Algorithm Overview

1) Train a super-network by jointly training networks in the search space
2) Search for the optimal network using the proposed optimizer
   - It samples networks and evaluates them without further training
Algorithm Overview

1) Train a super-network by jointly training networks in the search space
2) Search for the optimal network using the proposed optimizer
   - It samples networks and evaluates them without further training
3) Fine-tune the discovered network until convergence
Proposed Techniques

1) Train a super-network by jointly training networks in the search space
   • **Ordered dropout (OD):** reduce the time for training a super-network
2) Search for the optimal network using the proposed optimizer
   • **Channel-level bypass connections (CBCs):** reduce the time for evaluating samples
   • **Multi-layer coordinate descent (MCD):** reduce the time for evaluating samples while supporting non-differentiable search metrics
3) Fine-tune the discovered network until convergence
Ordered Dropout

- Train multiple networks in a single pass to speed up super-network training
Ordered Dropout

- Train multiple networks in a single pass to speed up super-network training
- Architecture simulation: zero out different channels for different input images
  - Always zero out the last channels to avoid the training-evaluation mismatch
Channel-Level Bypass Connections

- NetAdaptV2 searches layer width, network depth, and kernel size

![Diagram showing channel-level bypass connections]
Channel-Level Bypass Connections

- NetAdaptV2 searches layer width, network depth, and kernel size

![Diagram of Channel-Level Bypass Connections]

W/o CBCs
Channel-Level Bypass Connections

- NetAdaptV2 searches layer width, network depth, and kernel size
- CBCs merge network depth and layer width into a single search dimension and allow searching only layer width
  - High-level idea: when a filter is removed, an input channel is bypassed
Multi-Layer Coordinate Descent

- MCD gradually and iteratively shrinks an initial network until the given constraints are satisfied.
Multi-Layer Coordinate Descent

- In each iteration, MCD
  - Generates $J$ coordinate directions by randomly shrinking $L$ layers
  - Finds the optimal network along them

---

**Sampled Network 1**
- (latency: 95ms)
- (accuracy: 90%)

**Sampled Network 2**
- (latency: 95ms)
- (accuracy: 80%)

---

**Output**
- (Fixed # Filters)

**CONV1**
- (8 Filters)

**CONV2**
- (28 Filters)

---

**CONV1**
- (10 Filters)

**CONV2**
- (26 Filters)
Multi-Layer Coordinate Descent

- This process continues until the given constraints are satisfied
Multi-Layer Coordinate Descent

- This process continues until the given constraints are satisfied

<table>
<thead>
<tr>
<th># of Filters in CONV Layer 1</th>
<th># of Filters in CONV Layer 2</th>
</tr>
</thead>
</table>

MCD does not require the search metrics to be differentiable
NetAdaptV2 achieves better accuracy-latency or accuracy-MAC trade-offs than related works with much lower search time.

- Dataset: ImageNet
- Latency measured on a Pixel 1 CPU
- Search time (GPU-Hours) measured on V100s (BigNAS on TPU V3s)

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 Accuracy</th>
<th>MAC (M)</th>
<th>Search Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGANetV2-m</td>
<td>78.3%</td>
<td>312</td>
<td>1674</td>
</tr>
<tr>
<td>EfficientNet-B0</td>
<td>77.3%</td>
<td>390</td>
<td>-</td>
</tr>
<tr>
<td>MixNet-M</td>
<td>77.0%</td>
<td>360</td>
<td>-</td>
</tr>
<tr>
<td>NetAdaptV2</td>
<td>78.5%</td>
<td>314</td>
<td>656</td>
</tr>
</tbody>
</table>

▲ Latency-Guided Search
▲ MAC-Guided Search
Thank You for Watching

Project website: http://netadapt.mit.edu