Learning Transferable Architectures for Scalable Image Recognition

Recent Trends in Automated Machine Learning
Technische Universität München

Ahmed Bahnasy
June 2nd, 2021
Motivation

- CNN models require significant architectural engineering!
- Can we design an algorithm to design the architecture for us?!

Source: https://arxiv.org/abs/1409.4842
NAS (2017) Revisited

- NAS Components
  - Search Space
  - Search Algorithms
  - Child Model Evolution

Neural Architecture Search with Reinforcement Learning [Barret & Quoc 2017]
NAS (2017) vs. NASNet (2018)

- **NAS (2017) Limitations**
  - Computationally expensive for small dataset
  - No transferability between datasets

- **NASNet (2018)**
  - Reduced Computation Cost
  - Transferable from small dataset to large dataset
  - Generic learned features for multiple Computer Vision tasks
Inspiration from the Literature

- Repeated Convolution Blocks
- Going Deep by skip connections
NASNet Search Space

- Design a Cell, Repeat, Stack = Architecture
- Search Space outputs Cells not Architectures
- Architecture Complexity is decoupled from depth of the network and size of the input images
NASNet Search Space

- Predictions of each cell are grouped into $B$ blocks
- Each block has 5 prediction steps
- The combination can be addition or concatenation

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-separable conv
NASNet Search Space
NASNet Architecture

- Composed of **repetitive** convolution cells
  - Normal Cell
  - Reduction Cell
- Free Parameters
  - \# Block repetitions (N)
  - \# initial convolutions

![Diagram of NASNet Architecture]

---

Technische Universität München

Learning Transferable Architectures for Scalable Image Recognition
Seminar on Recent trends in Automated Machine Learning
Ahmed Bahnasy | 2021
Experiments: CIFAR-10

NASNet-A Architecture, best one!
Experiments: Computing Resources

- NAS (2017)
  - 800 GPUs
  - 28 Days
  - 22,400 GPU hours
  - Nvidia K40 GPUs

- NASNet (2018)
  - 500 GPUs
  - 4 Days
  - 2,000 GPU hours
  - NVidia P100s

*Discounting the fact of different Hardware, it is estimated to be roughly 7x more efficient*
## Experiments: ImageNet

<table>
<thead>
<tr>
<th>Model</th>
<th>Image size</th>
<th># parameters</th>
<th>Mult-Add</th>
<th>Top 1 Acc. (%)</th>
<th>Top 5 Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V2 [29]</td>
<td>224 × 224</td>
<td>11.2 M</td>
<td>1.94 B</td>
<td>74.8</td>
<td>92.2</td>
</tr>
<tr>
<td>NASNet-A (5 @ 1538)</td>
<td>299 × 299</td>
<td>10.9 M</td>
<td>2.35 B</td>
<td>78.6</td>
<td>94.2</td>
</tr>
<tr>
<td>Inception V3 [60]</td>
<td>299 × 299</td>
<td>23.8 M</td>
<td>5.72 B</td>
<td>78.8</td>
<td>94.4</td>
</tr>
<tr>
<td>Xception [9]</td>
<td>299 × 299</td>
<td>22.8 M</td>
<td>8.38 B</td>
<td>79.0</td>
<td>94.5</td>
</tr>
<tr>
<td>Inception ResNet V2 [58]</td>
<td>299 × 299</td>
<td>55.8 M</td>
<td>13.2 B</td>
<td>80.1</td>
<td>95.1</td>
</tr>
<tr>
<td>NASNet-A (7 @ 1920)</td>
<td>299 × 299</td>
<td>22.6 M</td>
<td>4.93 B</td>
<td>80.8</td>
<td>95.3</td>
</tr>
<tr>
<td>ResNeXt-101 (64 x 4d) [68]</td>
<td>320 × 320</td>
<td>83.6 M</td>
<td>31.5 B</td>
<td>80.9</td>
<td>95.6</td>
</tr>
<tr>
<td>Polynet [69]</td>
<td>331 × 331</td>
<td>92 M</td>
<td>34.7 B</td>
<td>81.3</td>
<td>95.8</td>
</tr>
<tr>
<td>DPN-131 [8]</td>
<td>320 × 320</td>
<td>79.5 M</td>
<td>32.0 B</td>
<td>81.5</td>
<td>95.8</td>
</tr>
<tr>
<td>SENet [25]</td>
<td>320 × 320</td>
<td>145.8 M</td>
<td>42.3 B</td>
<td>82.7</td>
<td>96.2</td>
</tr>
<tr>
<td>NASNet-A (6 @ 4032)</td>
<td>331 × 331</td>
<td>88.9 M</td>
<td>23.8 B</td>
<td>82.7</td>
<td>96.2</td>
</tr>
</tbody>
</table>

### Mobile Scale Networks

- **NASNet-A (4 @ 1056)**: Top 1 Acc. 74.0%, Top 5 Acc. 91.6%
- **NASNet-B (4 @ 1536)**: Top 1 Acc. 72.8%, Top 5 Acc. 91.3%
- **NASNet-C (3 @ 960)**: Top 1 Acc. 72.5%, Top 5 Acc. 91.0%
Experiments: ImageNet Computational Demand
Experiments: COCO Dataset

Object Detection Task

NASNet-A + RCNN Framework

<table>
<thead>
<tr>
<th>Model</th>
<th>resolution</th>
<th>mAP (mini-val)</th>
<th>mAP (test-dev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNet-224 [24]</td>
<td>600 x 600</td>
<td>19.8%</td>
<td>-</td>
</tr>
<tr>
<td>ShuffleNet (2x) [70]</td>
<td>600 x 600</td>
<td>24.5%[^1]</td>
<td>-</td>
</tr>
<tr>
<td><strong>NASNet-A (4 @ 1056)</strong></td>
<td>600 x 600</td>
<td><strong>29.6%</strong></td>
<td>-</td>
</tr>
<tr>
<td>ResNet-101-FPN [36]</td>
<td>800 (short side)</td>
<td>-</td>
<td>36.2%</td>
</tr>
<tr>
<td>Inception-ResNet-v2 (G-RMI) [28]</td>
<td>600 x 600</td>
<td>35.7%</td>
<td>35.6%</td>
</tr>
<tr>
<td>Inception-ResNet-v2 (TDM) [52]</td>
<td>600 x 1000</td>
<td>37.3%</td>
<td>36.8%</td>
</tr>
<tr>
<td><strong>NASNet-A (6 @ 4032)</strong></td>
<td>800 x 800</td>
<td><strong>41.3%</strong></td>
<td>40.7%</td>
</tr>
<tr>
<td><strong>NASNet-A (6 @ 4032)</strong></td>
<td>1200 x 1200</td>
<td><strong>43.2%</strong></td>
<td><strong>43.1%</strong></td>
</tr>
<tr>
<td>ResNet-101-FPN (RetinaNet) [37]</td>
<td>800 (short side)</td>
<td>-</td>
<td>39.1%</td>
</tr>
</tbody>
</table>
Experiments: Architecture search methods

- **Random Search**
  - Strong Baseline!
- **Reinforcement Learning**
  - Entire range superior quality models
  - Best model differs ~ 1% accuracy
  - Better mean performance on the top-5 and top-25
Experiments: Scheduled DropPath

- **DropPath**
  - each path in the cell is stochastically dropped with some fixed probability during training

- **Scheduled DropPath**
  - each path in the cell is dropped out with a probability that is linearly increased over the course of training

- Observed through experiments

No Further Explanation!
The paper through the eyes of 2021

- EfficientNet-L2 90.2%
- ViT - 88.55%
- NASNet - 82.7%

https://paperswithcode.com/sota/image-classification-on-imagenet
Thank you :)

Questions ??
References

- Learning Transferable Architectures for Scalable Image Recognition
- Neural architecture search with reinforcement learning
- FractalNet: Ultra-Deep Neural Networks without Residuals
- EfficientNet: Rethinking model scaling for convolutional neural networks
- EfficientNetV2: Smaller Models and Faster Training