Differentiable Architecture Search

Alexander Becker

Seminar: Recent Trends in Automated Machine Learning
Technical University of Munich
July 4, 2019
Overview

- Increasing interest in automatic architecture discovery
- Most approaches are computationally expensive, e.g. on ImageNet/CIFAR-10:
  - 2000 GPU days of reinforcement learning by Zoph et al. (2017)
  - 3150 GPU days of evolution by Real et al. (2018)
- Problem: Optimization over discrete domain, requiring many evaluations
- Liu et al. (2018) propose continuous relaxation of the search space
Search Space

- Search for building blocks instead of entire network architecture
- Building blocks ("cells") can then be stacked/connected recurrently
- Cell is represented as a directed acyclic graph with latent representations as nodes:
Continuous Relaxation

- Let $O$ be a set of possible operations
- Relax each edge to a softmax weighted mixture of operations from $O$

$$
\bar{o}^{(i,j)}(x) = \sum_{o \in O} \frac{\exp(\alpha^{(i,j)}_o)}{\sum_{o' \in O} \exp(\alpha^{(i,j)}_{o'})} o(x)
$$

- Parametrization by $\alpha = \{\alpha^{(i,j)}\}$
- After the search, $\bar{o}^{(i,j)}$ can be discretized by $o^{(i,j)} = \text{argmax}_{o \in O} \alpha^{(i,j)}_o$
Continuous Relaxation

(a) 
(b) 
(c) 
(d)
Optimization

- Joint optimization of model weights and architecture parameters
- Optimize validation loss using gradient descent
- Bi-level optimization problem:

\[
\min_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \\
\text{s.t.} \quad w^*(\alpha) = \arg\min_w \mathcal{L}_{train}(w, \alpha)
\]

- Gradient can be expressed as

\[
\nabla_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \\
\approx \nabla_{\alpha} \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)
\]
Optimization

\textbf{Algorithm 1: DARTS – Differentiable Architecture Search}

Create a mixed operation $\bar{o}^{(i,j)}$ parametrized by $\alpha^{(i,j)}$ for each edge $(i, j)$

\textbf{while not converged do}

\begin{enumerate}
\item Update architecture $\alpha$ by descending $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$
\quad ($\xi = 0$ if using first-order approximation)
\item Update weights $w$ by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$
\end{enumerate}

Derive the final architecture based on the learned $\alpha$. 
Optimization

Recent Trends in Automated Machine Learning

Epoch 0
Experiments & Results

Recent Trends in Automated Machine Learning

Alexander Becker
Experiments & Results

Recent Trends in Automated Machine Learning

Alexander Becker
Experiments & Results

Recent Trends in Automated Machine Learning

Alexander Becker
<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Error (%)</th>
<th>Params (M)</th>
<th>Search Cost (GPU days)</th>
<th>#ops</th>
<th>Search Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseNet-BC (Huang et al., 2017)</td>
<td>3.46</td>
<td>25.6</td>
<td>–</td>
<td>–</td>
<td>manual</td>
</tr>
<tr>
<td>NASNet-A + cutout (Zoph et al., 2018)</td>
<td>2.65</td>
<td>3.3</td>
<td>2000</td>
<td>13</td>
<td>RL</td>
</tr>
<tr>
<td>NASNet-A + cutout (Zoph et al., 2018)†</td>
<td>2.83</td>
<td>3.1</td>
<td>2000</td>
<td>13</td>
<td>RL</td>
</tr>
<tr>
<td>BlockQNN (Zhong et al., 2018)</td>
<td>3.54</td>
<td>39.8</td>
<td>96</td>
<td>8</td>
<td>RL</td>
</tr>
<tr>
<td>AmoebaNet-A (Real et al., 2018)</td>
<td>3.34 ± 0.06</td>
<td>3.2</td>
<td>3150</td>
<td>19</td>
<td>evolution</td>
</tr>
<tr>
<td>AmoebaNet-A + cutout (Real et al., 2018)†</td>
<td>3.12</td>
<td>3.1</td>
<td>3150</td>
<td>19</td>
<td>evolution</td>
</tr>
<tr>
<td>AmoebaNet-B + cutout (Real et al., 2018)</td>
<td>2.55 ± 0.05</td>
<td>2.8</td>
<td>3150</td>
<td>19</td>
<td>evolution</td>
</tr>
<tr>
<td>Hierarchical evolution (Liu et al., 2018b)</td>
<td>3.75 ± 0.12</td>
<td>15.7</td>
<td>300</td>
<td>6</td>
<td>evolution</td>
</tr>
<tr>
<td>PNAS (Liu et al., 2018a)</td>
<td>3.41 ± 0.09</td>
<td>3.2</td>
<td>225</td>
<td>8</td>
<td>SMBO</td>
</tr>
<tr>
<td>ENAS + cutout (Pham et al., 2018b)</td>
<td>2.89</td>
<td>4.6</td>
<td>0.5</td>
<td>6</td>
<td>RL</td>
</tr>
<tr>
<td>ENAS + cutout (Pham et al., 2018b)†</td>
<td>2.91</td>
<td>4.2</td>
<td>4</td>
<td>6</td>
<td>RL</td>
</tr>
<tr>
<td>Random search baseline† + cutout</td>
<td>3.29 ± 0.15</td>
<td>3.2</td>
<td>4</td>
<td>7</td>
<td>random</td>
</tr>
<tr>
<td>DARTS (first order) + cutout</td>
<td>3.00 ± 0.14</td>
<td>3.3</td>
<td>1.5</td>
<td>7</td>
<td>gradient-based</td>
</tr>
<tr>
<td>DARTS (second order) + cutout</td>
<td>2.76 ± 0.09</td>
<td>3.3</td>
<td>4</td>
<td>7</td>
<td>gradient-based</td>
</tr>
</tbody>
</table>
# Experiments & Results

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Perplexity</th>
<th>Params</th>
<th>Search Cost</th>
<th>#ops</th>
<th>Search Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>valid</td>
<td>test</td>
<td>(M)</td>
<td>(GPU days)</td>
<td></td>
</tr>
<tr>
<td>Variational RHN (Zilly et al., 2016)</td>
<td>67.9</td>
<td>65.4</td>
<td>23</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>LSTM (Merity et al., 2018)</td>
<td>60.7</td>
<td>58.8</td>
<td>24</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>LSTM + skip connections (Melis et al., 2018)</td>
<td>60.9</td>
<td>58.3</td>
<td>24</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>LSTM + 15 softmax experts (Yang et al., 2018)</td>
<td>58.1</td>
<td>56.0</td>
<td>22</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NAS (Zoph &amp; Le, 2017)</td>
<td>–</td>
<td>64.0</td>
<td>25</td>
<td>1e4 CPU days</td>
<td>4</td>
</tr>
<tr>
<td>ENAS (Pham et al., 2018b)*</td>
<td>68.3</td>
<td>63.1</td>
<td>24</td>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>ENAS (Pham et al., 2018b)*</td>
<td>60.8</td>
<td>58.6</td>
<td>24</td>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>Random search baseline‡</td>
<td>61.8</td>
<td>59.4</td>
<td>23</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DARTS (first order)</td>
<td>60.2</td>
<td>57.6</td>
<td>23</td>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>DARTS (second order)</td>
<td>58.1</td>
<td>55.7</td>
<td>23</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Conclusion

- DARTS proves feasibility of architecture search using gradient descent
- More efficient than non-differentiable approaches and reaches similar performance
- Simple and powerful
- Part of the one-shot family of algorithm search
- Possibly large gap between continuous solution and derived discrete architecture
- Does not find novel architectures in a broad sense
- Bi-level solving algorithm is not mathematically derived but rather a heuristic
Questions?