Weight Agnostic Neural Networks

Adam Gaier\textsuperscript{1,2}, David Ha\textsuperscript{1}

1 Google Brain, 2 Inria / CNRS / Université de Lorraine / Bonn-Rhein-Sieg University of Applied Sciences
Architecture is a Powerful Prior

- Deep Image Prior

- ConvNet with randomly initialized weights can still perform many image processing tasks
- Without learning, the network structure alone is a strong enough prior


Innate abilities in animals
Innate abilities in machines

To what extent can neural net architectures *alone* encode solutions to tasks?
Neural Architecture Search

Searching for trainable networks

- Architectures, once trained, outperform hand designed networks
- Expensive -- training of network required to judge performance
- Solution is still encoded in weights of network, not in architecture
Searching for Architectures

Searching for Architectures

Search Space \( \mathcal{A} \) → Search Strategy → Performance Estimation Strategy

architecture \( A \in \mathcal{A} \)

performance estimate of \( A \)

How can we search for architectures...*not* weights?
Search without Training

Assume weights are drawn from a particular distribution
- Search for architecture to perform given weights from this distribution

Replace inner loop training with sampling
- Draw new weights from distribution at each rollout
- Judge network on zero-shot performance
Weight Sharing

Single shared weight value used for all connections

- Weight value selected from distribution at each rollout
- Reduces number of parameters of network to 1
  - Reliable expected reward of topology

Architecture search

- Explore space of network topologies
- Judge network architecture based on performance over a series of rollouts
Topology Search
1.) Initialize
Create population of minimal networks.

2.) Evaluate
Test with range of shared weight values.

3.) Rank
Rank by performance and complexity

4.) Vary
Create new population by varying best networks.
WANNs find solutions in variety of RL tasks
WANNs perform with and without training

<table>
<thead>
<tr>
<th>Swing Up</th>
<th>Random Weights</th>
<th>Random Shared Weight</th>
<th>Tuned Shared Weight</th>
<th>Tuned Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>WANN</td>
<td>57 ± 121</td>
<td>515 ± 58</td>
<td>723 ± 16</td>
<td>932 ± 6</td>
</tr>
<tr>
<td>Fixed Topology</td>
<td>21 ± 43</td>
<td>7 ± 2</td>
<td>8 ± 1</td>
<td>918 ± 7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Biped</th>
<th>Random Weights</th>
<th>Random Shared Weight</th>
<th>Tuned Shared Weight</th>
<th>Tuned Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>WANN</td>
<td>-46 ± 54</td>
<td>51 ± 108</td>
<td>261 ± 58</td>
<td>332 ± 1</td>
</tr>
<tr>
<td>Fixed Topology</td>
<td>-129 ± 28</td>
<td>-107 ± 12</td>
<td>-35 ± 23</td>
<td>347 ± 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CarRacing</th>
<th>Random Weights</th>
<th>Random Shared Weight</th>
<th>Tuned Shared Weight</th>
<th>Tuned Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>WANN</td>
<td>-69 ± 31</td>
<td>375 ± 177</td>
<td>608 ± 161</td>
<td>893 ± 74</td>
</tr>
<tr>
<td>Fixed Topology</td>
<td>-82 ± 13</td>
<td>-85 ± 27</td>
<td>-37 ± 36</td>
<td>906 ± 21</td>
</tr>
</tbody>
</table>
ANN Bipedal Walker
(2760 connections, weights)

WANN Bipedal Walker
(44 connections, 1 weight)
Can we find WANNs outside of reinforcement learning domains?
<table>
<thead>
<tr>
<th>WANN</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Weight</td>
<td>82.0% ± 18.7%</td>
</tr>
<tr>
<td>Ensemble Weights</td>
<td>91.6%</td>
</tr>
<tr>
<td>Tuned Weight</td>
<td>91.9%</td>
</tr>
<tr>
<td>Trained Weights</td>
<td>94.2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANN</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression</td>
<td>91.6% [50]</td>
</tr>
<tr>
<td>Two-Layer CNN</td>
<td>99.3% [12]</td>
</tr>
</tbody>
</table>
Searching for Building Blocks

First steps toward a different kind of architecture search

- Network architectures with innate biases can perform a variety of tasks
- ...and these biases can be found through search

Weight tolerance as a heuristic for new building blocks

- ConvNets and LSTMs can work even untrained
- Finding novel building blocks at least as important as new arrangements of those which already exist
interactive article @: weightagnostic.github.io

poster @: wednesday 10:45