Towards modular and programmable architecture search

Renato Negrinho  Darshan Patil  Nghia Le  Daniel Ferreira  Matthew Gormley  Geoffrey Gordon

**Highlights**

**Motivation**
- Hyperparameter optimization not focused on architecture search use cases
- Current architecture search systems are monolithic
  - Ad-hoc encodings for search spaces
  - Intertwined search space and search algorithm
  - Task-specific, e.g., image classification.
- Programmable frameworks (e.g., deep learning) had transformative impact on machine learning

**Search space example**

**Description**
- Input convolutional layer
- Optional dropout w/ rate 0.25 or 0.5
- Two parallel convolutional chains:
  - One chain length 1, 2, or 4
  - Other chain double the first
- Chain outputs concatenated
- Each convolution has 64 or 128 filters (chosen separately)

25008 possible architectures

**Code**

```python
def search_space():
    k_h = [3, 5, 7]
    k_w = [3, 5, 7]
    c_inputs, c_outputs = conv2d(inputs, k_h, k_w)
    d_inputs, d_outputs = dropout(d_inputs, dropout_rate)
    c_outputs = concatenate([c_outputs, d_outputs])
    r1, r2 = repeat([r1, r2], times)
    c_outputs = concatenate([c_outputs, r1, r2])
    c_outputs = concatenate([c_outputs, r1, r2])
    return c_outputs
```

**Transitions**
- a) Search space encoded by code
  - a -> b) Value assigned to IH-1
    - Triggers value assignment to DH-1
    - Triggers substitution for Repeats (1 and 2)
  - b -> c) Value assigned to IH-2
    - Triggers substitution for Optional-1
  - c -> d) Assignments to IH-1 (64), IH-4 (128), IH-5 (128), IH-6 (64), and IH-7 (0.5)
  - d) can be mapped to implementation

**Language**

**Constructs**
- Independent hyperparameters
  - Value picked from set (e.g., IH-1)
- Dependent hyperparameters
  - Value computed as function of other hypers (e.g., DH-2)
- Basic modules
  - Deep learning operation (Conv2D-1)
- Substitution modules
  - Structural lazy transformations (e.g., Repeat-1) to computation graph through substitutions (replace and reroute)
- Auxiliary functions
  - Helps compose search spaces into larger search spaces
  - (e.g., rnn_cell)

**Mechanics**
- Search algorithms interface with search spaces by iteratively assigning values to independent hyperparameters.
- After all hyperparameters have values assigned, architecture mapped automatically to implementation

**Experiments**

Mix and match search spaces and search algorithms from the literature without implementing each combination from scratch

One search algorithm (random search), many search spaces

One search space (genetic), many search algorithms

**Table 1: Test results for search space experiments**

<table>
<thead>
<tr>
<th>Search Space</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic [23]</td>
<td>90.07</td>
</tr>
<tr>
<td>Flat [23]</td>
<td>90.59</td>
</tr>
<tr>
<td>Naive [24]</td>
<td>90.59</td>
</tr>
<tr>
<td>Nitro [24]</td>
<td>90.37</td>
</tr>
</tbody>
</table>

**Table 2: Test results for search algorithm experiments**

<table>
<thead>
<tr>
<th>Search Algorithm</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>90.61 ± 0.67</td>
</tr>
<tr>
<td>MCTS [25]</td>
<td>90.45 ± 0.31</td>
</tr>
<tr>
<td>SMB [10]</td>
<td>90.38 ± 0.42</td>
</tr>
<tr>
<td>Evolution [15]</td>
<td>90.22 ± 0.56</td>
</tr>
</tbody>
</table>