Neural Logic Reinforcement Learning

Zhengyao Jiang, Shan Luo

Department of Computer Science,
University of Liverpool,
Liverpool, United Kingdom

z.jiang22@student.liverpool.ac.uk
shan.luo@liverpool.ac.uk
Differentiable Inductive Logic Programming

- It is to learn logic rules to express a concept, for example, grandfather(X,Y)
- It needs background knowledge:
  - father(david, bert), father(bert, tom), father(richard, lily), father(sam, richard)
- The positive and negative examples are provided:
  - Positive: grandfather(david, tom), grandfather(sam, lily)
  - Negative: any other atoms involving grandfather
- We can induce: grandfather(X,Y):- father(X,Z), father(Z,Y)
- The differentiable ILP algorithms parameterise possible rules and use gradient descent to induce the correct rule
  - The algorithm we will use for NLRL is \( \partial \text{ILP} \) [1]

Learn Logic Rules with Policy Gradient - NLRL

- The state and background knowledge is encoded as a knowledge base
- In each decision step, after deduction of DILP, the truth score of the action atoms is used to derive the action probability
- The DILP architecture is trained with REINFORCE [2]

## NLRL vs Symbolic Planning

<table>
<thead>
<tr>
<th></th>
<th>NLRL</th>
<th>Symbolic Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm Based On</td>
<td>Policy Gradient</td>
<td>Searching</td>
</tr>
<tr>
<td>Known Dynamics</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>Data (Experience)</td>
<td>Required</td>
<td>Not Required</td>
</tr>
<tr>
<td>Reward</td>
<td>Any Step, Any Value</td>
<td>0/1 at the last step</td>
</tr>
<tr>
<td>Allow Stochascity</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
# NLRL vs Deep Reinforcement Learning

<table>
<thead>
<tr>
<th></th>
<th>NLRL</th>
<th>DRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential Architecture</td>
<td>Differentiable ILP</td>
<td>Neural Network</td>
</tr>
<tr>
<td>Interpretability</td>
<td>Good</td>
<td>Poor</td>
</tr>
<tr>
<td>Generalizability</td>
<td>Strong</td>
<td>Weak</td>
</tr>
<tr>
<td>Allow Continuous Action</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow Sensory Input</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Experiment Results

(a) UNSTACK
(b) STACK
(c) ON
(d) Cliff-walking
(e) Windy Cliff-walking
Explain the Induced Policies

- Take the UNSTACK as an example:
  
  \[
  0.972 : \text{move}(X, Y) \leftarrow \text{isFloor}(Y), \text{pred}(X) \\
  0.987 : \text{pred}(X) \leftarrow \text{pred2}(X), \text{top}(X) \\
  0.997 : \text{pred2}(X) \leftarrow \text{on}(X, Y), \text{on}(Y, Z)
  \]

- \text{pred2} labels blocks whose height is at least 2.
  
  - In the figure below, \{b, c\}

- \text{pred} labels blocks to be moved.
  
  - On the top and whose height is at least 2. In the figure below, \{c\}
Neural Logic Reinforcement Learning

Welcome to our poster: Pacific Ballroom #104

Code available: https://github.com/ZhengyaoJiang/NLRL