Towards TempoRL: Learning When to Act

André Biedenkapp, Raghu Rajan, Frank Hutter & Marius Lindauer
In a Nutshell

1. We propose a proactive way of doing RL
2. We introduce skip-connections into MDPs
   ○ through action repetition
   ○ allows for faster propagation of rewards
3. We propose a novel algorithm using skip-connections
   ○ learn *what* action to take & *when* to make new decisions
   ○ condition *when* on *what*
4. We evaluate our approach with tabular Q-learning on small grid worlds
Motivation

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$r = 0$
Motivation

$r = 0$

$G$
Optimal Policies
Optimal Policies: When do we need to act?

# Steps: 16
# Decisions: 16
Optimal Policies: When do we need to act?

# Steps: 16
# Decisions: 5
Optimal Policies: When do we need to act?

# Steps: 16
# Decisions: 4
Optimal Policies: When do we need to act?

# Steps: 16
# Decisions: 3
Proactive Decision Making

# Steps: 16
# Decisions: 16

~80% fewer Decision points

# Steps: 16
# Decisions: 3
Skip MDPs
1. Use standard Q-learning to determine the behaviour
   \[ Q^\pi(s_t, a) \rightarrow a \]

2. Condition skips on the chosen action.
   \[ Q^{\pi,j}(s_t, j|a) \rightarrow j \]

3. Play action \(a\) for the next \(j\) steps

The action Q-function

The skip Q-function can be learned using n-step updates
Experimental Evaluation

12.41 \times \text{speedup}

13.57 \times \text{speedup}

Reward

1

0

-1

10^0 \quad 10^1 \quad 10^2 \quad 10^3 \quad 10^4

\#Episodes

Q

t-Q
Wrap-Up

Code & Data available:

https://github.com/automl/TabularTempoRL

Future work:

- Use deep function approximation
- Different exploration mechanisms for skip and behaviour policies