RLgraph: Modular Computation Graphs for Deep Reinforcement Learning

Michael Schaarschmidt, Sven Mika, Kai Fricke, Eiko Yoneki
Reinforcement Learning (RL)

Image taken from https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html
Supervised Learning vs RL

• Supervised Learning
  • Training data beforehand

• Reinforcement Learning
  • Learn and collect data at the same time
  • No labeled dataset
  • Sensitive to hyper parameters
Related Works

• Existing RL Libraries
  • OpenAI baselines, TensorForce, Ray RLlib

• Pros
  • Present good results on existing environments in library
  • Code is concise

• Cons
  • Hard to adapt other environments since components tightly coupled
  • Restricted to a single backend
  • Unable to test a subcomponent individually
## RLgraph

<table>
<thead>
<tr>
<th>API, Component configuration</th>
<th>Prebuilt models, inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLgraph component graph</td>
<td>Model design, dataflow composition</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>Local backends variables/operations</td>
</tr>
<tr>
<td>PyTorch</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Distributed TF</td>
<td>Distributed execution engine</td>
</tr>
<tr>
<td>Horovod</td>
<td></td>
</tr>
<tr>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Hardware: CPU, GPU, TPU, FPGAs...</td>
<td>Execution, orchestration</td>
</tr>
</tbody>
</table>
The First Layer

Image taken from SysML 19: https://www.youtube.com/watch?v=96cludHRSYM&t=1073s
The Second Layer

Image taken from SysML 19: https://www.youtube.com/watch?v=96cludHRSYM&t=1073s
The Third Layer

When component ready, **explicitly:**
- Assign device
- Assign scopes
- Create (and share) variables
- Call computations

Image taken from SysML 19: https://www.youtube.com/watch?v=96cludHSYM&t=1073s
The Fourth Layer

General purpose API: `get_action`, `update`, `export`, ...

Agent API

- Graph executor/devices/profiling
- Graph Builder
- OP registry
- Local backends

Distributed coordination layer

- Ray executor
- Distributed TF/PS

Vectorized sample collection

- Ray Worker_1
- Ray Worker_n
- TF Worker_1
- TF Worker_n

Local RLgraph agent

Graph executor syncs variables to PS, manages plugins (Horovod)
Evaluation
Build Overhead

![Graph showing build overhead comparison between TF data flow, TF build, PT data flow, and PT build](image_taken_from_the_paper)
Runtime Overhead

![Graph showing runtime overhead comparison between TF RLgraph, PT RLgraph, and PT hand-tuned versions.](Image taken from the paper)
RLgraph vs RLlib
RLgraph with multi-GPUs
Summary

• Introduce modularity to RL Tools
• Focus on dataflow design instead of backend tools
• Future work
  • Integrate AutoGraph / JIT Tracing into build process
• Reference
  • https://www.youtube.com/watch?v=96cludHRSYM&t=1073s