RLgraph: Modular Computation Graphs for Deep Reinforcement Learning
M. Schaarschmidt, S. Mika, K. Fricke, E. Yoneki at SysML, 2019

R244 Large-scale data processing and optimisation
Presentation by Martin Graf on 19/10/2022
Reinforcement Learning is hard
Reinforcement Learning is hard

• Algorithmic instability
Reinforcement Learning is hard

- Algorithmic instability
- Diversity of models and optimization strategies
Reinforcement Learning is hard

• Algorithmic instability
• Diversity of models and optimization strategies
• Highly varied resource requirements
Reinforcement Learning is hard

• Algorithmic instability
• Diversity of models and optimization strategies
• Highly varied resource requirements
• Heterogeneous distributed communication patterns
Existing Reinforcement Learning Tooling
Existing Reinforcement Learning Tooling

• Reference implementations on benchmark tasks
  • OpenAI baselines (Sidor & Schulman, 2017)
  • Keras-rl (Plappert, 2016)
  • …
Existing Reinforcement Learning Tooling

• Reference implementations on benchmark tasks
  • OpenAI baselines (Sidor & Schulman, 2017)
  • Keras-rl (Plappert, 2016)
  • …
• Ray RLlib (Liang et al., 2018)
Existing Reinforcement Learning Tooling

• Reference implementations on benchmark tasks
  • OpenAI baselines (Sidor & Schulman, 2017)
  • Keras-rl (Plappert, 2016)
  • …
• Ray RLlib (Liang et al., 2018)
• TensorForce (Schaarschmidt et al., 2018)
Existing Reinforcement Learning Tooling

• Reference implementations on benchmark tasks
  • OpenAI baselines (Sidor & Schulman, 2017)
  • Keras-rl (Plappert, 2016)
  • …
  • Ray RLlib (Liang et al., 2018)
  • TensorForce (Schaarschmidt et al., 2018)

Problem: No separation of concerns
Trends in Machine Learning Tooling
Trends in Machine Learning Tooling

• Towards higher level APIs and standardization
  • Keras (Chollet et al., 2015)
  • ONNX (Facebook Inc., 2017)
Trends in Machine Learning Tooling

• Towards higher level APIs and standardization
  • Keras (Chollet et al., 2015)
  • ONNX (Facebook Inc., 2017)

• Towards better performance
  • Hardware improvements
  • Software improvements
    • Weld (Palkar et al., 2017)
    • FlexFlow (Jia et al., 2018)
  • …
Value Hypothesis

Trend: Standardization, higher level APIs

Current RL tooling: No separation of concerns

Trend: Continuous Performance Improvements

?
Value Hypothesis

Current RL tooling:
No separation of concerns

Trend: Standardization,
higher level APIs

Trend: Continuous
Performance
Improvements

RLgraph
Key Ideas
Key Ideas

• Separate execution details and user code
Key Ideas

• Separate execution details and user code
• No-code distributed computation
Key Ideas

• Separate execution details and user code
• No-code distributed computation
• Backend agnostic, high level API
Key Ideas

• Separate execution details and user code
• No-code distributed computation
• Backend agnostic, high level API
• Testable
Introducing: RLgraph

high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms
Introducing: RLgraph

high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms

Agent
Policy  Loss  Optimizer  …
Memory  …
Introducing: RLgraph

high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms

Multi-framework
• TensorFlow
• PyTorch

Multi-paradigm
• distributed TensorFlow (Abadi et al., 2016)
• Ray (Moritz et al., 2017)
Introducing: RLgraph

A high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms.
Introducing: RLgraph

high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms.
Introducing: RLgraph

**high-level** backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms.

Introducing: RLgraph

high-level backend-agnostic scalable graph-based testable library with a component-based modular build-system for designing and executing fast, robust, incrementally testable, and easy to extend or re-use reinforcement learning algorithms
Performance
Performance

(a) Build overheads.

Performance

(a) Build overheads.  (b) Worker act performance.

Performance

RLgraph: Modular Computation Graphs for Deep Reinforcement Learning

Reinforcement learning (RL) tasks are challenging to implement, execute, and test due to algorithmic instability, hyper-parameter sensitivity, and heterogeneous distributed communication patterns. We argue for the separation of logical component composition, backend graph definition, and distributed execution. To this end, we introduce RLgraph, a library for designing and executing reinforcement learning tasks in both static graph and define-by-run paradigms. The resulting implementations are robust, incrementally testable …

Source: https://scholar.google.com, accessed on 18/10/2022

Source: https://github.com/rlgraph/rlgraph, accessed on 18/10/2022
RLgraph: Modular Computation Graphs for Deep Reinforcement Learning

Reinforcement learning (RL) tasks are challenging to implement, execute and test due to algorithmic instability, hyper-parameter sensitivity, and heterogeneous distributed communication patterns. We argue for the separation of logical component composition, backend graph definition, and distributed execution. To this end, we introduce RLgraph, a library for designing and executing reinforcement learning tasks in both static graph and define-by-run paradigms. The resulting implementations are robust, incrementally testable …

Cited by 13

rlgraph / rlgraph

RLgraph: Modular computation graphs for deep reinforcement learning

Apache-2.0 license

297 stars 40 forks

michaelschaarschmidt

on 5 Nov 2019
RLgraph still relevant? 2022…

- Ray RLlib incorporates concepts of RLgraph

RLgraph still relevant? 2022…

• Ray RLlib incorporates concepts of RLgraph


• Autograph far more capable
Critique

• Is being backend agnostic really beneficial?
  • Constant updates with new backend versions necessary
  • Increased maintenance effort
Critique

• Is being backend agnostic really beneficial?
  • Constant updates with new backend versions necessary
  • Increased maintenance effort
• Problems in one specific backend should be addressed in that backend
Critique

- Is being backend agnostic really beneficial?
  - Constant updates with new backend versions necessary
  - Increased maintenance effort
- Problems in one specific backend should be addressed in that backend
- Is mixing Python control flow with machine learning framework code really bad?
  - Autograph
References


• Chollet, F. et al. Keras. https://keras.io, 2014.

References


References


References


RLgraph: Modular Computation Graphs for Deep Reinforcement Learning
M. Schaarschmidt, S. Mika, K. Fricke, E. Yoneki at SysML, 2019

R244 Large-scale data processing and optimisation
Presentation by Martin Graf on 19/10/2022