Towards Future-Based Explanations for DRL Network Controllers

By: Sagar Patel, Sangeetha Abdu Jyothi, Nina Narodytska

UCI University of California, Irvine
vmware®
The Need for Explainability

• Deep Reinforcement Learning (DRL) offers high performance in increasingly more domains
• However, DRL lacks wide-scale deployment because operators cannot
  • Understand the blackbox neural network
  • Debug the controller when it misbehaves
  • Adjust to fix problems
Reinforcement Learning: ABR Perspective

- State
- Action
- Agent
- Reward
  - $+ \alpha$ Quality
  - $- \beta$ Quality Change
  - $- \gamma$ Stalling
- Environment
- bitrates
  - 480P
  - 720P
  - 1080P
Reinforcement Learning: ABR Perspective

- DRL network controllers are deployed in highly variable network environments
  - E.g., WAN
  - Cannot be faithfully modeled yet
  - During training
    - Replayed as external conditions by a dataset of traces
  - At deployment
    - Future is unknown
Prior Explainers for DRL Network Controllers

• Inputs
  • A black-box controller

• Procedure
  1. Sample states from the training environment
  2. Get actions for the states
  3. Train an explainable model using all the state-action pairs
     • Decision Tree [Metis (SIGCOMM ’20), Trustee (CCS ‘22)]
The Need for Future Based Explanations

Feature Based
+ Identify key features
+ Simplify the model
  (e.g. decision trees)

- Do not capture time-dependent nature of DRL
- Do not give insights into impact of actions

Future Based
+ Reveal future impact of actions
+ Allow contrasting actions and states
+ Enable network observability

- Cannot simplify the model
Explaining the future: What can we use?
Decomposed Future Rewards as Explanations

Agent

Environment

$r_{Quality, t+1}$

$r_{Quality Change, t+1}$

$r_{Stalling, t+1}$

$r_{Quality, t+2}$

$r_{Quality Change, t+2}$

$r_{Stalling, t+2}$

$r_{Quality, t+3}$

$r_{Quality Change, t+3}$

$r_{Stalling, t+3}$

$r_{Quality, t+n}$

$r_{Quality Change, t+n}$

$r_{Stalling, t+n}$
Decomposed Future Returns as Explanations

Agent

Environment

\[
\begin{align*}
    r_{\text{Quality}, t+1} & \quad r_{\text{Quality}, t+2} & \quad r_{\text{Quality}, t+3} & \quad r_{\text{Quality}, t+n} \\
    r_{\text{Quality Change}, t+1} & \quad r_{\text{Quality Change}, t+2} & \quad r_{\text{Quality Change}, t+3} & \quad \ldots & \quad r_{\text{Quality Change}, t+n} \\
    r_{\text{Stalling}, t+1} & \quad r_{\text{Stalling}, t+2} & \quad r_{\text{Stalling}, t+3} & \quad r_{\text{Stalling}, t+n} \\
\end{align*}
\]

Weighted Sum

\[
R_{\text{Quality}} \\
R_{\text{Quality Change}} \\
R_{\text{Stalling}}
\]
Decomposed Future Returns as Explanations

- We do not yet know how to model Network environments (e.g., WANs)
  - Future is unknown
  - Rewards cannot be calculated

\[
\begin{align*}
r_{\text{Quality}, t+1} & \quad r_{\text{Quality}, t+2} & \quad r_{\text{Quality}, t+3} & \quad r_{\text{Quality}, t+n} \\
r_{\text{Quality Change}, t+1} & \quad r_{\text{Quality Change}, t+2} & \quad r_{\text{Quality Change}, t+3} & \quad r_{\text{Quality Change}, t+n} \\
r_{\text{Stalling}, t+1} & \quad r_{\text{Stalling}, t+2} & \quad r_{\text{Stalling}, t+3} & \quad r_{\text{Stalling}, t+n}
\end{align*}
\]

Weighted Sum

\[
\begin{align*}
R_{\text{Quality}} \\
R_{\text{Quality Change}} \\
R_{\text{Stalling}}
\end{align*}
\]
CrystalBox

Data Generation

(s₁, a₁, r₁),
(s₂, a₂, r₂),
...

Supervised Learning

(s, a, R)

Explainer

Learned Predictor

Agent

Training Environment

Traces

Preprocess

Training Data

Predictor
Use Case 1: Network Observability

• Controllers experience a wide variety of network conditions
• Alerting for future performance drops before they happen is important
• Future-based explanations can help
Use Case 2: Guided Reward Design

- Fine tuning reward weights is tedious and resource inefficient
- Future-based explanations can help
- For example, keeping everything else constant, we change stall weight

![Graph showing dominant return component with weight on the stalling component and count values.](image)
Use Case 3: Cross-State Explainability

Agent

State 1: Medium Quality vs State 2: High Quality

State 1: Future-based Explanation

State 2: Future-based Explanation

Medium Quality

High Quality
Open Questions

• Using future states as explanations
• Online safety assurance
• Combining Feature + Future based explainers
Thank You