Verifiable Reinforcement Learning via Policy Extraction

Osbert Bastani, Yewen Pu, Armando Solar-Lezama
Deep Reinforcement Learning
Deep Reinforcement Learning

control problem

Deep Reinforcement Learning (RL) is a control problem where a neural network controller $\pi_{NN}$ is trained to control actions in a system. The system dynamics $(x_{\text{ball}}', y_{\text{ball}}', ...)$ are updated based on the state vector $(x, y, ...)$.

- Easy to train
- Hard to verify
Our Approach

- Control problem
- Neural network controller $\pi_{NN}$
- RL
- Supervised learning
  - Easy to verify
  - Hard to train
- RL
- Decision tree controller $\pi_{DT}$
- Verification
  - Certificate of Correctness
Background
Imitation Learning

Abbeel & Ng 2004
Imitation Learning

Demonstrations from Neural Network

Decision Tree Controller

Abbeel & Ng 2004
Imitation Learning

Step 1: Use NN to generate states

Step 2: Use NN to obtain actions

Step 3: Use supervised learning to train a decision tree

Ross & Bagnell 2011
Imitation Learning

Ross & Bagnell 2011
Dataset Aggregation (DAgger)

Ross & Bagnell 2011
Dataset Aggregation (DAgger)

Ross & Bagnell 2011
Viper Algorithm
Insight: Critical States

actions are similar (non-critical state)

must move right! (critical state)
Our Approach: Leverage the $Q$-Function

$$Q(s, a) = \text{“how good is action } a \text{ in state } s?\text{”} \in \mathbb{R}$$
Our Approach: Leverage the $Q$-Function

\[ Q(s, \pi_{NN}(s)) \approx \min_{a \in A} Q(s, a) \]

non-critical state (low priority)

\[ Q(s, \pi_{NN}(s)) \gg \min_{a \in A} Q(s, a) \]

critical state (high priority)

worst-case $Q$ value

optimal $Q$ value

worst-case $Q$ value

optimal $Q$ value
Viper Algorithm

• DAgger treats all state-action pairs equally:

$$\pi_{DT} = \arg \min_{\pi} \sum_{s \in D} \mathbb{I}[\pi(s) = \pi_{NN}(s)]$$

• Viper weights state-action pairs by the $Q$-function:

$$\pi_{DT} = \arg \min_{\pi} \sum_{s \in D} \left( Q(s, \pi_{NN}(s)) - \min_{a' \in A} Q(s, a') \right) \mathbb{I}[\pi(s) = \pi_{NN}(s)]$$

optimal $Q$ value

worst-case $Q$ value
Theoretical Guarantees

**Theorem.** For any $\delta > 0$, there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$ such that

$$J(\hat{\pi}) \leq J(\pi^*) + T\epsilon_N + \tilde{O}(1)$$

with probability at least $1 - \delta$, as long as $N = \tilde{\Theta}(\ell_{\max}^2 T^2 \log(1/\delta))$. 
Evaluation
vs. Decision Trees via RL (on Cart-Pole)

![Graph showing comparison between Fitted Q and Viper](image)

- Fitted Q
- Viper

**Legend:**
- Training neural network controller $\pi_{NN}$
vs. to DAgger (on Atari Pong)

# Nodes

Reward

- DAgger
- Viper
Verifying Correctness of a Toy Pong Controller

• **Toy Pong**
  • states = $\mathbb{R}^5$
  • actions = \{left, right, stay\}

• **Neural network:**
  • trained using policy gradients
  • 600 neurons

• **Decision tree:**
  • extracted using Viper
  • 31 nodes
Verifying Correctness of a Toy Pong Controller

• Inductive invariant:
  \[ s(0) \in \text{blue} \Rightarrow s(t) \in \text{blue} \]

• Verification algorithm
  • dynamics are piecewise linear
  • SMT formula over linear arithmetic
  • solved by Z3 in < 5 seconds

• Results:
  • error when ball starts on the right
  • fixed when paddle is slightly longer!
Conclusion

Verifiability is critical to enabling application of deep reinforcement learning to safe-critical systems.