FACULTY OF NATURAL SCIENCES

Martin Kubovčík

RL-TOOLKIT: DESIGN AND IMPLEMENTATION OF A SET OF TOOLS FOR RIINFORCEMENT LEARNING IN ROBOTICS

supervisor: prof. RNDr. Jiří Pospíchal, DrSc., Faculty of natural sciences, University of SS. Cyril and Methodius in Trnava

MOTIVATION

RL-Toolkit is a novel software package designed for addressing problems involving the training of both simulated and real robots. This capability is made possible through the utilization of **DeepMind Reverb**, a database server, and employs a Server-Client architecture. The robots function as clients, while the server, equipped with numerous GPUs, processes the data gathered by the robots. This server is designated as the learner, responsible for storing the experiences acquired during interactions within the environment. To enhance the accuracy of Q-values, which denote the quality of actions, a smooth loss function is employed. The **logcosh** loss function is utilized due to its reduced sensitivity to outliers. Q-values are predicted as a distribution of quantiles, which offers even greater precision compared to solely predicting mean Q-values. To mitigate overestimation, the highest quantiles must be excluded from the predicted distributions. **The Soft Actor-Critic** technique is employed for training the robots. This off-policy method enables the utilization of historical data. Deep neural networks are used to predict actions based on observed states, ensuring a satisfactory level of precision. Similarly, deep neural networks are employed to predict Q-values.

OBJECTIVE

1.1 Entropy of predicted actions

Entropy is a measure of the level of unpredictability and uncertainty in predicting actions. This fundamental concept enables a robot to explore its environment effectively. The robot will not be able to successfully solve a task solely by taking deterministic actions.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_b p(x_i) = -\mathbb{E}\left[log\mathcal{N}(\mu,\, \sigma^2)\,\right]$$

1.2 Minimize actor loss

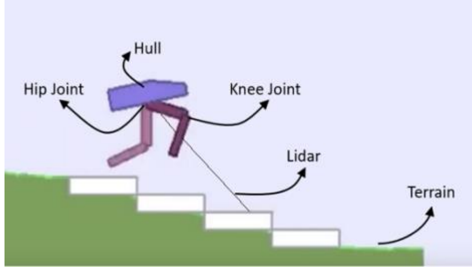
The objective is to optimize the actor's parameters by incorporating both entropy and maximized Q-value into the actor loss.

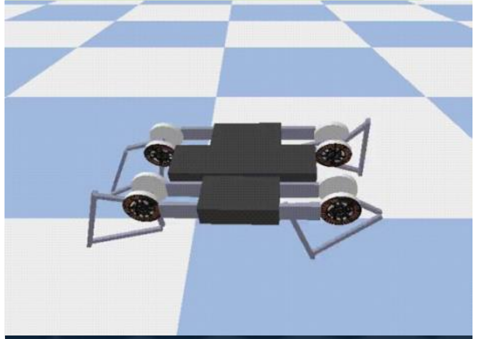
$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D,\epsilon_t\sim\mathcal{N}}\left[\alpha\log\pi_\phi(f_\phi(\epsilon_t, s_t)|s_t) - Q_\theta(s_t, f_\phi(\epsilon_t, s_t))\right]$$

1.3 Minimize critic loss

The critic loss is mean squared error. The target value of outputs is given as Bellman equation.

$$J_Q(\theta) = \mathbb{E}_{(s_t,\, a_t)\sim D}\left[\frac{1}{2}(Q_\theta(s_t,\, a_t) - (r(s_t,\, a_t) + \gamma(Q_\theta(s_{t+1},\, a_{t+1}) - \alpha\log\pi_\phi(a_{t+1}|s_{t+1}))))^2\right]$$

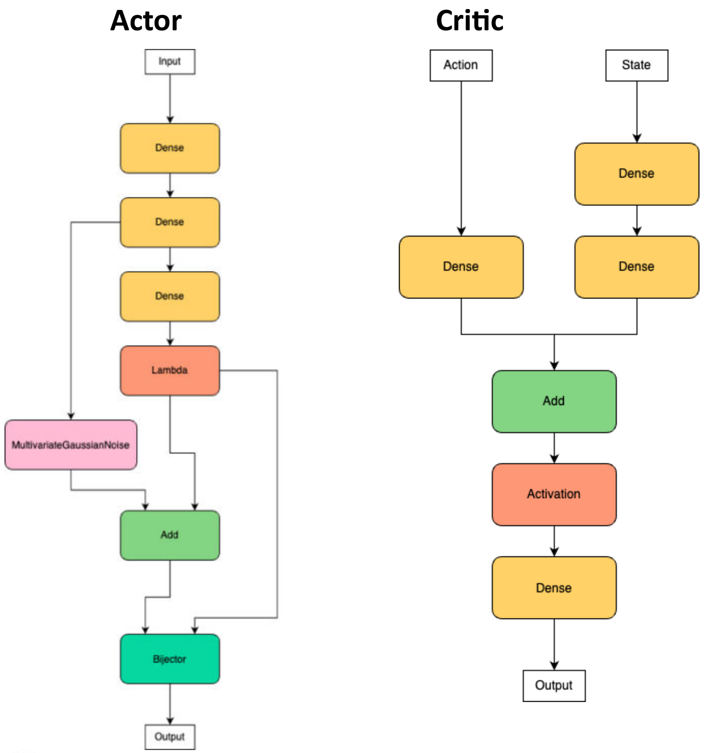


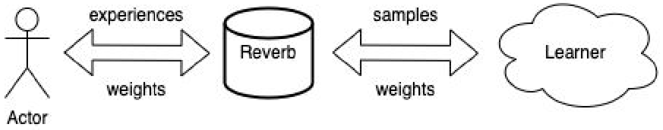


RESULTS

Utilizing a database server yielded significant benefits by distributing the problem across multiple instances. The employment of the database solution allowed for storage of a significantly larger number of interactions, thanks to the sample-to-insert ratio. Furthermore, by excluding the **last 3 quantiles** and employing a smooth error function instead of the original mean squared error, we achieved an average improvement of **9.38%** in solving the given problems across diverse environments.

Model scheme


Actor
Input
Dense
Dense
Dense
Lambda
MultivariateGaussianNoise
Add
Bijector
Output
Critic
Action
State
Dense
Dense
Dense
Add
Activation
Dense
Output


Actor
experiences
weights
Reverb
samples
weights
Learner