Federated Reinforcement Learning Framework for Autonomous Driving

Drive Link:
https://drive.google.com/drive/u/1/folders/1zokNQSImZNWMtvPXvKqVZmCcPic_N9pB

Code Documentation

1. **Simulation Environment**
    A. Simulation Environment: The Open Racing Car Simulator [TORCS]
    B. Official Website: http://torcs.sourceforge.net/index.php
    C. Download the simulator from the website: http://torcs.sourceforge.net/index.php?name=Sections&op=viewarticle&artid=3
    D. Download the TORCS setup for FRL framework: https://drive.google.com/drive/folders/1pLEyMRb20DuX0hlLT1BRq_4cHZLLJ4_p?usp=sharing
    Changes: Added the src_server code in the Drivers folder. 10 server XML files are added to enable multiple client connections from a single computer.
    E. Important links
        1) T.O.R.C.S. Manual installation and Robot tutorial: https://docplayer.net/28813194-T-o-r-c-s-manual-installation-and-robot-tutorial.html

2. **Client Code[Reinforcement Learning]**
    A. Reinforcement Learning Algorithm: Deep Deterministic Policy Gradient [DDPG]
    B. Code Link: https://drive.google.com/drive/folders/1vyKvNpQICM8lWI4wGAGe5TAhbaRbLLPb?usp=sharing
    C. Code Directory Structure

- Best Critic Weights #store globally aggregated critic weights received from the server
    aggreagted_critic.py
- Best Actor Weights #store globally aggregated actor weights received from the server
    aggreagted_actor.py
- Brain
    Algorithms.py #ddpg algorithm training steps
    NeuralArch.py #neural architecture classes
- Connection
    Ddpg_clients.py #connect agent to global server
    Models.py #initialize models
    Serverconnection.py #communication with global server
    Utils.py #for visualization
- Logs #ddpg algorithm training steps
    - Log
        Log.txt #log summary of each episode

Reward.csv #log average reward per episode

  - ReplayBuffers #log replay for each episode
  - Weights_Critic #log critic weights for each episode
  - Weights_Policy #log actor weights for each episode

- Utilities

  Init_system.py #initial steps before starting the agent
  Params.py #initialize parameters
  Utils.py #OU- Exploration Process Code

Gym_torcs.py #ddpg reward functions and environment initialization code
Snakeoil3_gym.py #connect Torcs environment and code
Train.py #main code. Run train.py to start the agent and enter server connection port number

D. Steps to connect client

1) Start the simulator by running wtorcs.exe
2) Click on Race → Quick Race → Configure Race → Select Track → Accept → Select driver src_server 'n' from the list → Accept →Accept → New Race.
   N can be between 1 to 10
3) Run train.py. Enter the port number to connect to torcs which will be shown on the TORCS terminal. If N is 1 then enter 3001.

E. Important Links

1) Research papers
   a) End-to-End Autonomous Driving Decision Based on Deep Reinforcement Learning
      https://ieeexplore.ieee.org/document/8813431
   b) Deep Reinforcement Learning for Autonomous Driving
      https://arxiv.org/abs/1811.11329

      Drive link:
      https://drive.google.com/drive/u/1/folders/1_9zm8hcKcHFkMFjJAOXoJY7MbEQMOrha
2) Youtube Links
   a) Reinforcement Learning - "DDPG" explained
      https://www.youtube.com/watch?v=oydExwuuUCw
   b) Reinforcement Learning in Continuous Action Spaces | DDPG Tutorial (Pytorch)
      https://www.youtube.com/watch?v=6Yd5WnYIs_Y&t=2082s
3) Blogs
   a) Deep Reinforcement Learning. Deep Deterministic Policy Gradient (DDPG) algorithm.
      https://markus-x-buchholz.medium.com/deep-reinforcement-learning-deep-deterministic-policy-gradient-ddpg-algoritm-5a823da91b43

        b) Using Keras and Deep Deterministic Policy Gradient to play TORCS https://yanpanlau.github.io/2016/10/11/Torcs-Keras.html

3. Global Server Code [Federated Learning]

A. Federated Learning algorithm: FedAvg

B. Code link:
https://drive.google.com/drive/folders/15S31Iz9V1tu60sW7dnFRe1K8J6RfM1km?usp=sharing

C. Code Directory Structure

- Log #server logs
- Models #stores global model
- Src
    - Connection.py #open server machine port for accepting client connections
    - Client.py #client connection
    - Models.py #initialize model architecture
    - Server.py #server operations
    - Utils.py #utility functions
- StaticModelAggregation #static model aggregation code
  Config.yaml #configuration parameters
  Connectclientscopy.py #client communication through the server
  Main.py #start server machine
  Requirements.txt #requirements file

D. Steps to start the server

1) Transfer the code into the server machine using WinScp
2) Log in to the server machine using putty terminal
3) Run main.py to start the server.

E. Important Links

1) Research Papers
    a) Towards Federated Learning At Scale: System Design
    b) RefinedFed: A Refining Algorithm for Federated Learning
    c) Federated Reinforcement Learning For Fast Personalization
    d) Federated Transfer Reinforcement Learning
    e) Federated Reinforcement Learning: Techniques, Applications, and Open Challenges
    f) Federated Learning: Challenges, Methods, and Future Directions
    g) Federated Learning with Non-IID Data
    h) A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison

Drive link:
https://drive.google.com/drive/u/1/folders/1_y3s3-WodKRmZUJ6-VN6kU4pf2PB6Y5v

4. Server Hosting Platform

A. Hosting Platform: Linode https://www.linode.com/

B. Server Configurations
Ubuntu 22.04 LTS Disk Profile – GRUB 2 1 GB RAM
Plan - Nanode 1 GB 5$/month plan

C. Softwares

1. puTTY: https://www.putty.org/
   Use: PuTTY is a free and open-source terminal emulator used for accessing the server terminal hosted on Linode.
2. WinSCP: https://winscp.net/eng/index.php
   Use: WinSCP is a free and open-source SSH File Transfer Protocol used for transferring code from local machine to server machine hosted on Linode.

D. Important Links

1. How to Host a Python Socket Server Online (For Free)
   https://www.youtube.com/watch?v=KQasIwElg3w&t=9s
2. Linode Beginner's Guide
   https://www.linode.com/docs/guides/linode-beginners-guide/