Rogues Gallery – OpenAI Gymnasium Team (Fall 2023)

Sri Ranganathan Palaniappan, Tejeshwar Natarajan, Rahul Narayanan, Chacko Abraham, Jeongyeop Han, Mugdha Daftardar, Dorsa Ajami

Introduction & Goals

Introduction

The Neuromorphic Team specializes in neuromorphic, also known as brain-inspired, applications.

Semester Goals

- Create a basic implementation of a reinforcement learning algorithm involving Q-learning
- Perform hyperparameter optimization to optimize the learning process of the algorithm
- Explore how Deep Q-Learning can improve on the Q-Learning implementation of a problem

Methodology

Q-Learning

- Method of reinforcement learning that learns to choose the best action in different states by observing the consequences of its actions and estimating the future reward for each potential action.
- Observation space is divided evenly into discrete chunks for the Q-Table, with each chunk representing a cell, and each cell holding different q values for each possible action.
- Values in the Q-Table are randomly initialized, and for each action taken, values begin to update based on the Q-Learning equation.
- Q-Learning equation determines the action that results in the highest reward for the agent at a given position, and updates the entry for it in the Q-Table with a higher value.

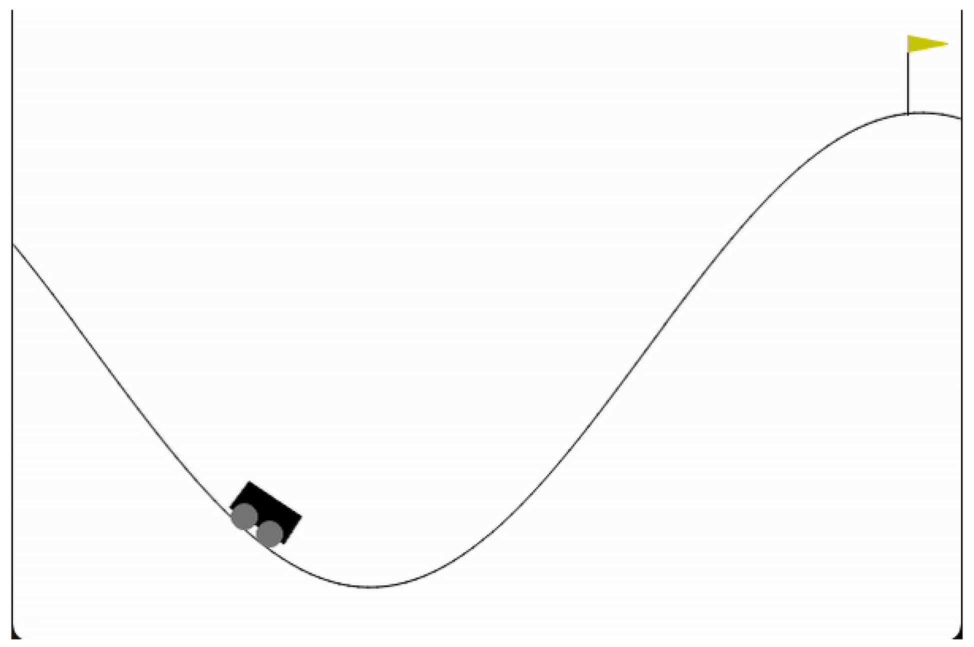
$$Q(s_t,a_t) \leftarrow (1-\alpha)\cdot \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_{a} Q(s_{t+1},a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Q-learning equation

Key Experiments & Results

Mountain Car Problem

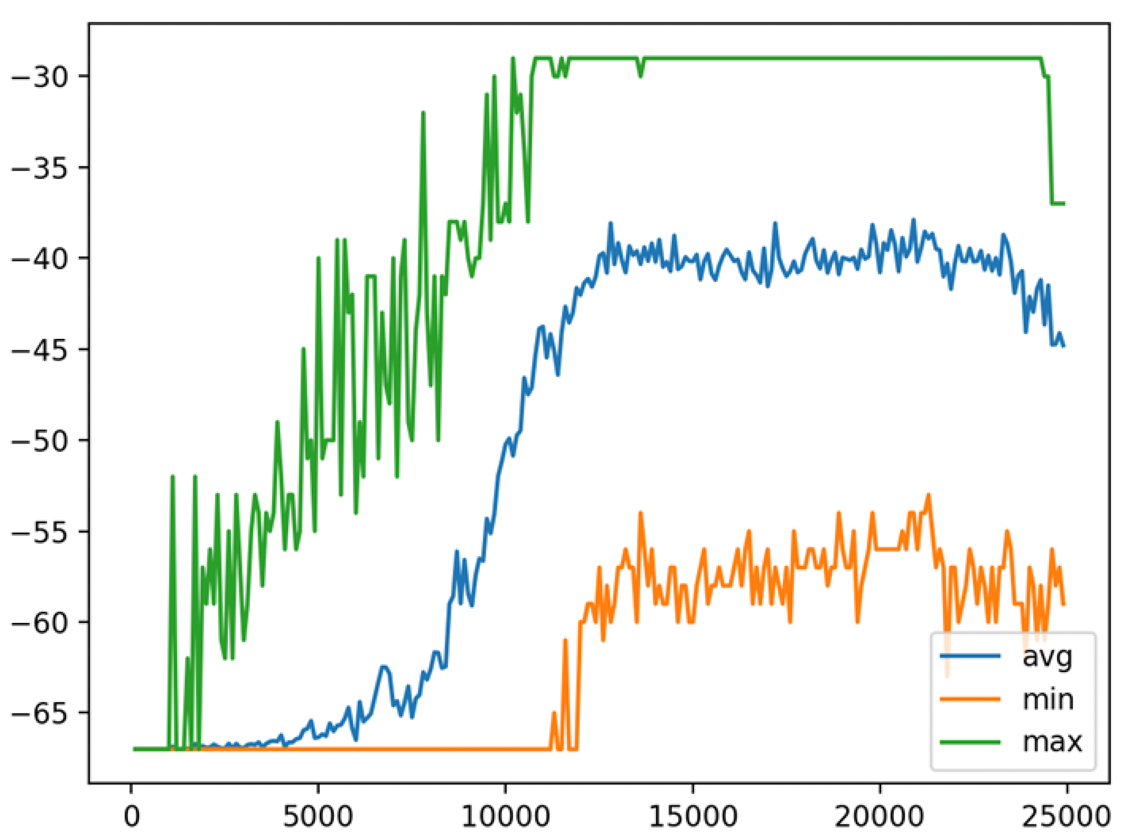
- The mountain car environment requires a car (agent) with insufficient power to ascend the steep hill to reach the flag.
- The car must learn optimal actions to build enough momentum and reach the goal position.



Hyperparameter Optimization

- Employed grid search over a valid range for learning rate and discount factor to get all combination of values and evaluate the agent's performance over a series of episodes.
- Iteratively refined hyperparameters based on empirical performance over multiple training runs.
- Evaluated average time to reach the goal over a series of episodes as the "score" (primary performance metric).
- Fine-tuned parameters resulted in improved stability and faster convergence of the Q-learning algorithm.

Change in Avg, Min, and Max Rewards Over Episodes



Next Steps & Challenges

Challenges

- Encountered difficulties in rendering the environment due to reliance on outdated documentation for the OpenAI Gym framework instead of the updated Gymnasium version.
- Additionally, experienced challenges in determining the optimal epsilon value during the hyperparameter tuning phase.

Next steps: Deep Q-Learning

- Conducted an in-depth investigation into the distinctions between Q-learning and Deep Q-learning, focusing on understanding the nuances involved in learning optimal policies.
- Next steps should include developing a DQL algorithm implementation for the Mountain Car problem, addressing key challenges such as handling continuous state spaces, employing a neural network as a Q-function approximator, implementing experience replay, and incorporating a target network.

Rogues Gallery – NVIDIA Jetson Team (Fall 2023)

Srijan Ponnala, Divij Karnani, Suzan Manasreh, James White

Introduction & Goals

Introduction

The Neuromorphic Team specializes in neuromorphic, also known as brain-inspired, computing applications. This semester, most of our efforts have been centered on implementing different neuromorphic algorithms, mainly focused on object detection, on the NVIDIA Jetson platform.

Semester Goals

- Implement the NVIDIA Jetson platform and delve into its robust capabilities for processing and analyzing visual data
- Focus on mastering running the various sophisticated image detection algorithms
- Explore DetectNet for training precise and specialized object detection models

Key Concepts & Results

Overview

- The Nvidia Jetson platform allows for use of NVIDIA TensorRT, which is specifically designed to take full advantage of the Jetson chip's architecture
- We are able to make use of TensorFlow-TensorRT, a wrapper for TensorRT that uses the more familiar syntax and functionality of Tensorflow.
- The NVIDIA Deep Learning GPU Training System (DIGITS) can perform common deep learning tasks including training models in parallel and choosing the best model from the results after training.

Image Classification

Object Detection

Pose Estimation

Semantic Segmentation

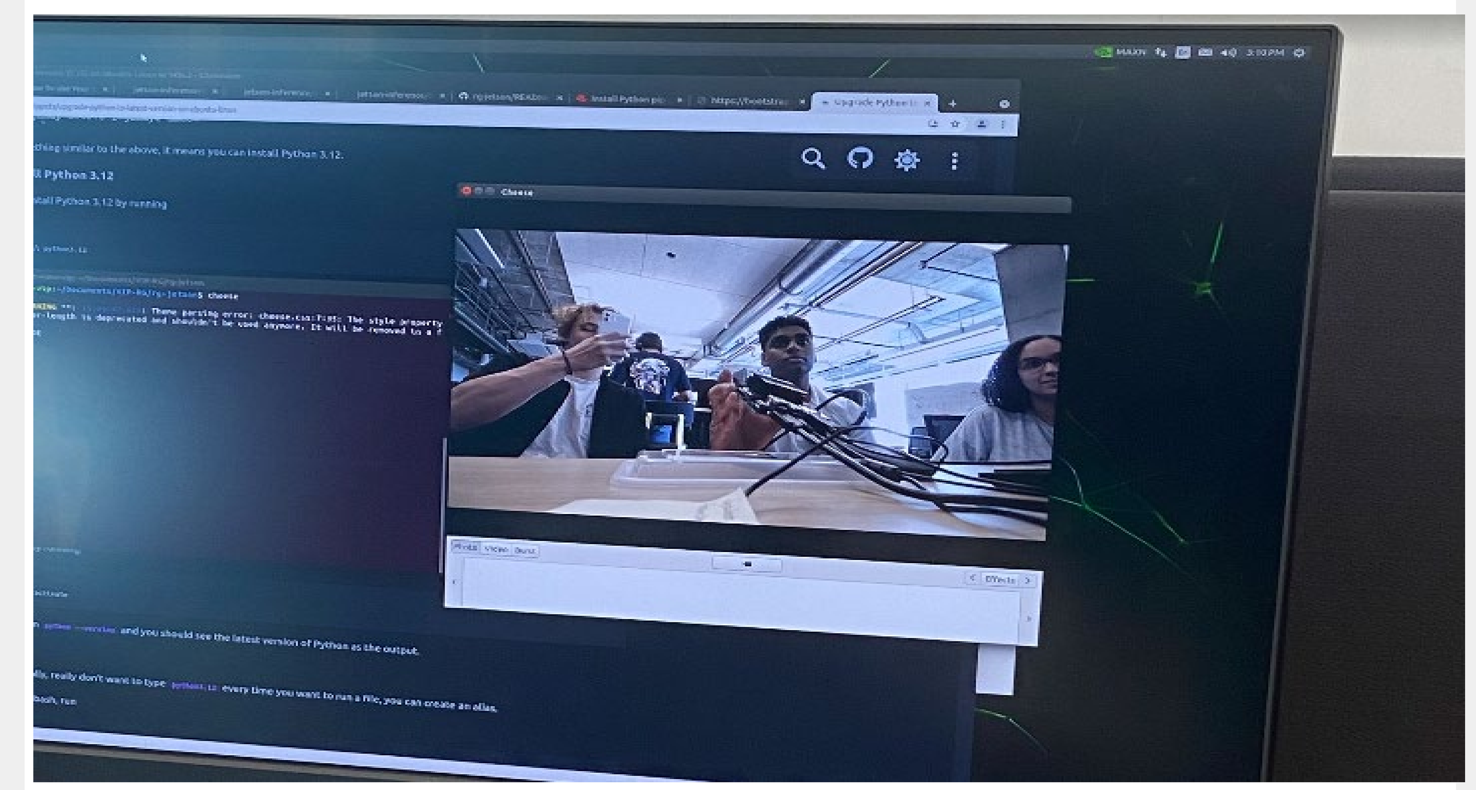
Hello AI World, rg-jetson

- The guide for inference and realtime vision DNN library for the NVIDIA Jetson platform. It uses TensorRT (SDK for high-performance deep learning inference) to run optimized networks on GPUs from Python, and PyTorch for training models.
- rg-jetson is the repository that contains the code we are working on to implement our object detection algorithms

DetectNet, YOLO

- DetectNet is a deep neural network designed for object detection and localization in images, optimized for performance and accuracy on NVIDIA platforms like Jetson.
- For object detection, algorithms like YOLO can also be used.
- These algorithms excel in identifying and localizing multiple objects within an image, offering a balance between accuracy and real-time processing

Challenges & Next Steps



Challenges

- We are still unable to find a good way to set up the jetson developer kit within a Python virtual environment, meaning only the chip we physically have is in use right now

Next Steps

- Run more complex image recognition algorithms on the Jetson
- Run comparisons between algorithms ran on the chip and algorithms ran on typical non-neuromorphic hardware

Credits
(Advisor) Jeff Young: jyoung9@gatech.edu
(Advisor) Aaron Jezghani: ajezghani3@gatech.edu
Srijan Ponnala: sponnala6@gatech.edu
Suzan Manasreh: smanasreh6@gatech.edu
James White: jwhite398@gatech.edu