ROB 537 Homework 4: Learning-Based Control

Nathan Butler | [butlnath@oregonstate.edu](mailto:butlnath@oregonstate.edu)

# Q-Learning Agent

1) Create a Q-learning agent that learns to solve the "Cart Pole" environment. The agent should balance the pole for 100 time steps.

* How will you handle the continuous state space?

A graph and a graph

Description automatically generated with medium confidence

Figure 1. DQN performance and epsilon decay during training

# Evolved Neural Network

2) Evolve a neural network to solve the previous task.

* What will you use for your evaluation function?
* What mapping should the network learn?

A graph with blue lines

Description automatically generated

Figure 2. Average neural network performance and best overall performance during training

# Discussion

Your report should include:

* Answers to previous questions
* Performance over time curves for each experiment.
* A description of your algorithms.
* Similarities and differences in performance for the two algorithms + an explanation.

A graph of a graph

Description automatically generatedA graph of a graph

Description automatically generated

A graph with blue lines

Description automatically generated

A graph with blue lines

Description automatically generated

# References

[1] https://blog.gofynd.com/building-a-deep-q-network-in-pytorch-fa1086aa5435 - cartpole-specific (beefed-up)

https://gibberblot.github.io/rl-notes/single-agent/function-approximation.html - example using street crossing thing

Training architecture adapted from: <https://stackoverflow.com/questions/56964657/cartpole-v0-loss-increasing-using-dqn>

Mutation strategies from: <https://stackoverflow.com/questions/31708478/how-to-evolve-weights-of-a-neural-network-in-neuroevolution>

# Appendix

Link to github code repository: <https://github.com/natbut/rob537_hw>