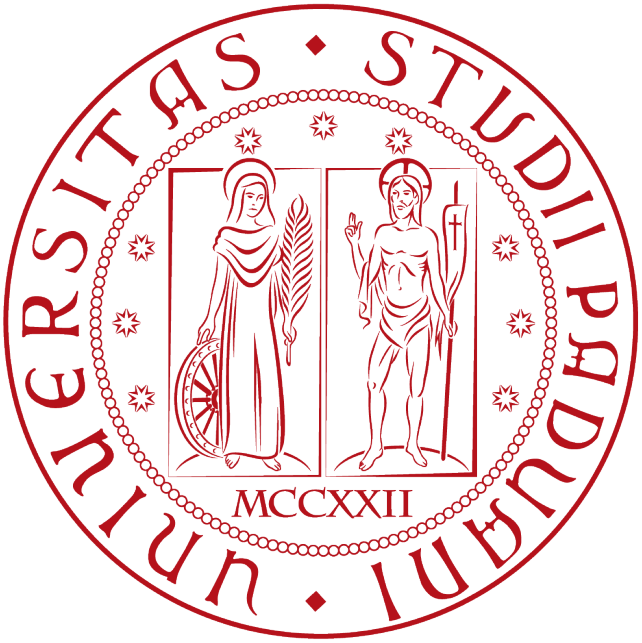
University of Padua

Deep Learning and Neural Networks
Reinforcement Learning

Third Assignment Report

Author:
Ufuk Baran Karakaya (1215960)
ufukbaran.karakaya@studenti.unipd.it

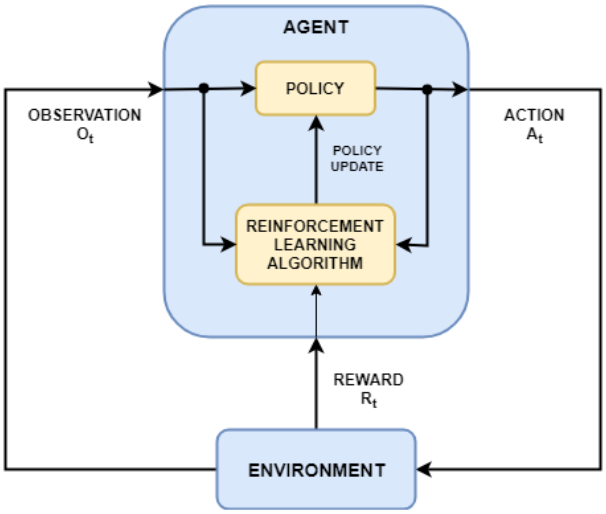
June 28, 2021



Introduction

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

In Reinforcement Learning, the algorithm is all about making decisions sequentially. Summary, the output depends on the state of the current input and the next input depends on the output of the previous input. On the other hand, in Supervised Learning, the decision is made on the initial input or the input given at the start.



Approach

The reinforcement learning follows these steps which are given the below:

- Input: The input should be an initial state from which the model will start
- Output: There are many possible output as there are variety of solution to a particular problem
- Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

Advantages of Reinforcement Learning

Reinforcement learning is applicable to a wide range of complex problems that cannot be handled with other machine learning algorithms. RL is closer to artificial general intelligence (AGI), as it possesses the ability to seek a long-term goal by exploring various possibilities on its own. Some of the benefits of RL include:

Conventional machine learning algorithms are designed to excel in specific secondary tasks, without a consideration of all perspective. RL, on the other hand, does not divide the problem into subproblems; it works directly to maximize long-term rewards. It has an obvious purpose, understands the goal, and is able to exchange short-term rewards for long-term benefits.

It does not need a separate step for data collection. In RL, training data is obtained through direct agent interaction with the environment. The training data is the experience of the learning agent, not a separate collection of data that must be provided to the algorithm. This significantly reduces the complexity on the supervisor in charge of the training process.

Work in dynamic and uncertain environments. RL algorithms are inherently adaptive and built to respond to changes in the environment. In RL, time matters, and the experience the agent collects is not independently and identically distributed (i.i.d.), unlike conventional machine learning algorithms. Since the dimension of time is deeply buried in the mechanics of RL, learning is inherently adaptive.

Algorithm

The project is based on implementation of three parts. First part covers extension of the softmax and epsilon greedy. The second part implements rgb pixel based controlling and the last part implements rl in another environment.

When the agent looks at the current state of the environment and chooses an action, the environment changes to a new state and also returns a reward indicating the consequences of the action. In this task, the rewards are +1 for each incremental time step and the environment ends if the pole falls too far or if the cart moves more than bounded units from the center. This means that the best-performing scenarios will run for a longer duration, accumulating a higher return.

The CartPole activity is designed so that the inputs to the agent are 4 real values representing the state of the environment (position, speed, etc.). However, neural networks can solve the task simply by observing the scene, so it will be used a cart-centered screen patch as input.

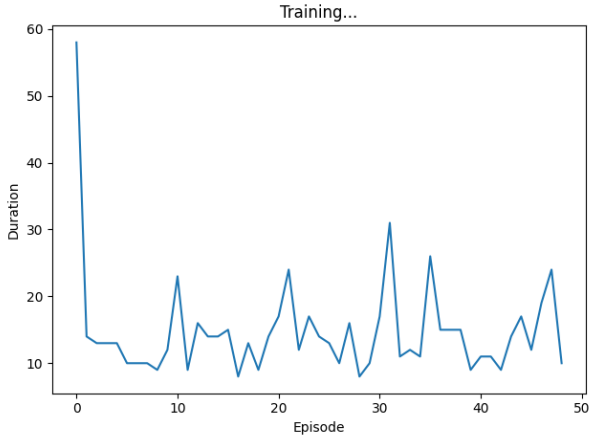
It will be presented the status as the difference between the current and previous screen patch. This will allow the agent to take pole speed from an image into account.

Essentially, in Q-Network, the model will be a convolutional neural network that takes in the difference between the current and previous screen patches. It has two outputs, representing Q(s,left) and Q(s, right) (where s is the input to the network). In effect, the network is trying to predict the expected return of taking each action given the current input.

For single based approach, it provides us learning the model with more flexibility however the accuracy and robustness of the model is quite better in state-based approach. Nevertheless, different pixel-based approaches such as model-free algorithms and model-based algorithms.

Results and Analysis

In the second part, rgb model is implemented to learn the CartPole. The duration of CartPole for every episode is illustrated on the graph.



References

- Hafner et al. Learning Latent Dynamics for Planning from Pixels. ICML 2019.
- Chen et al. A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020.
- Kostrikov et al. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. arXiv 2020.