CuriosityViz : Final Project

CS 8395-03 Visual Analytics & Machine Learning

Saroj Kumar Sahoo
Vanderbilt University
saroj.k.sahoo@vanderbilt.edu

1 Introduction

With the ever-increasing amount of data being generated the number of problems that can be solved using Artificial Intelligence nowadays is enormous. Deep Learning received a lot of attention from researchers and can be seen in use in our day to day lives. Whereas, Deep reinforcement learning is still in its research phase and is believed to have an equal impact as Deep Learning. Due to the black box nature of Deep Reinforcement Learning visualizing and understanding the decision-making process can help improve the behavior of the models. However, this topic is still to be explored. In this project, different visualization techniques were used to understand and compare the policy development of RL agent under different reward settings (extrinsic and intrinsic).

2 Background

Typically in Reinforcement Learning, we have an agent, the environment, continuous or discrete state, and action space. The agent when in certain state takes an action and goes to another state receiving a reward in the process. The magnitude of the reward determines how good or bad the action taken was. In, general setting the reward is extrinsic to the agent and needs to carefully hand-crafted humans and often is problem specific. But, most of the real world problems constructing a good reward function can prove to really difficult resulting in really sparse or missing rewards altogether. Thus, having a reward function that is intrinsic to the agent can help the agent learn faster hopefully. [1] proposed a formulation for intrinsic reward function. Visualizing the policy development of this method under different reward dense and sparse settings can help us understand better. No prior work has been done on this particular topic. Some work has been done on visualizing deep reinforcement learning. [2] used saliency maps to show the attention of the agent throughout the training process. [3] used different types of plots like line charts, stacked area charts, pie charts, bar charts views etc to visualize various stages of the learning process.

3 Data

For the purpose of experimentation, the breakout game (part of the atari 2600 games suite) was used. The various type of data that was collected are :

- Average Reward over all the episodes.
- Actions taken in an episode
- Extrinsic Reward collected by the agent in an episode.
- CNN features extracted from the ActorCriticNetwork of every state in an episode.

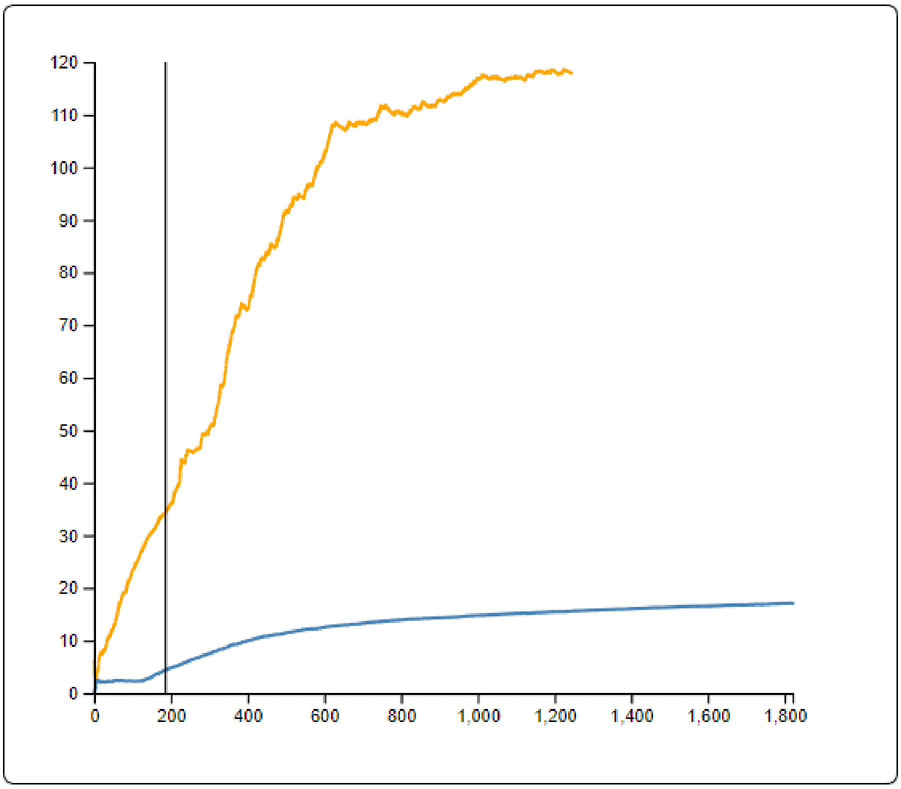
4 Goals

1. How to monitor the training progress? Did it train well? How fast or slow did the models train compared to each other? How much rewards did the models accumulate over episodes?
2. How well are the overall actions and rewards taken are distributed over all the episodes?

3. Compare each episode? What all states did the agents explore in an episode?
4. In a given episode how well are the actions distributed? Which action lead to how much reward? Were there any bias in the actions chosen by the agent?

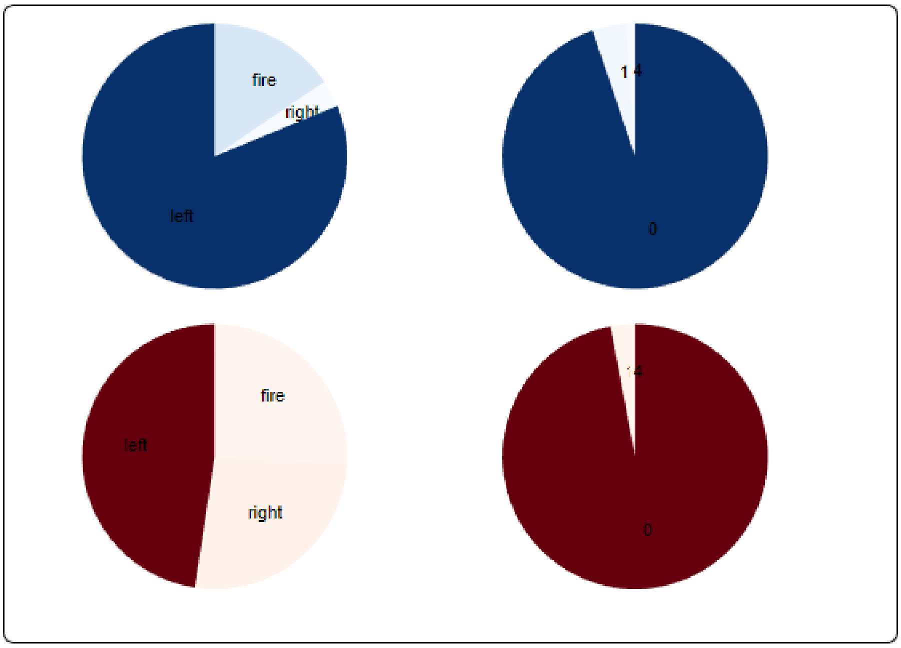
5 Components of CuriosityViz Dashboard

1. To achieve Goal 1, a simple line chart was used drawing average reward per episode for both the models. The x-channel represents episode number and y-channel represents average reward accumulated up to that episode. "Red" color represents model with only Extrinsic Reward whereas "Blue" color represents model with Intrinsic + Extrinsic Reward (though extrinsic reward has low weight).



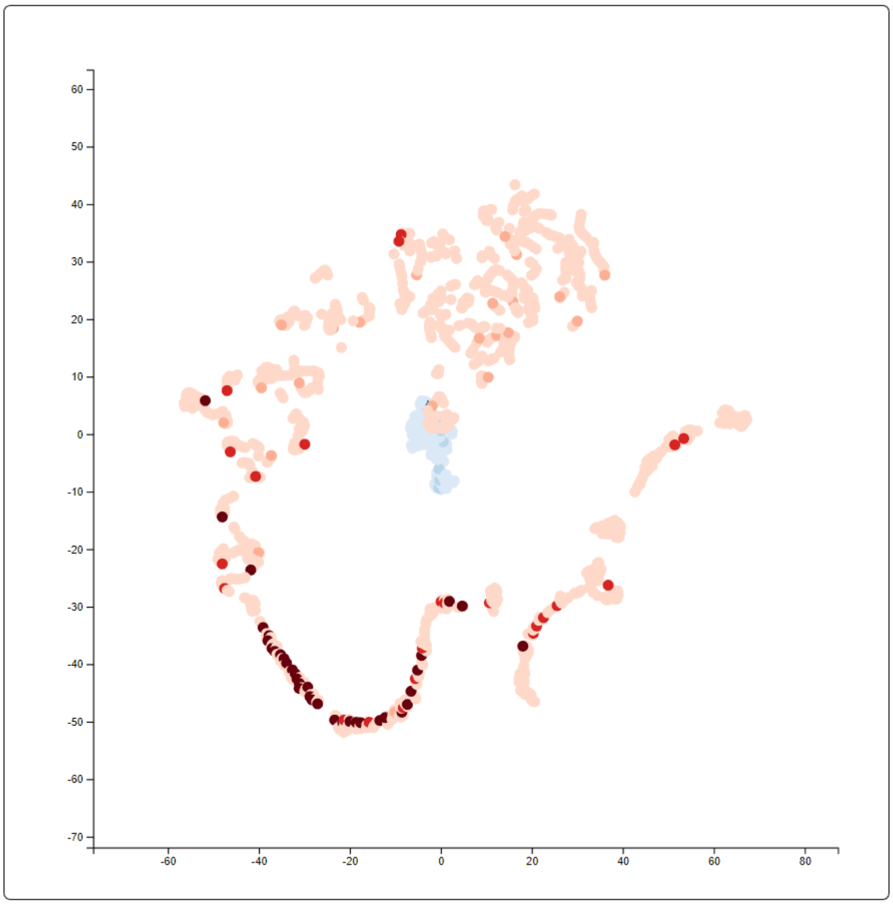
From the graph we can see that the agent trained on extrinsic rewards only learned quickly and has a nice graph showing an increase in average reward as the episode increased! On the other hand clearly the agent which relied mainly on intrinsic rewards and occasionally on extrinsic rewards learned relatively slowly, even though the learning process was slow it sure did learn. Point to be kept in mind, both the agents were using the same common hyperparameters including the learning rate. Thus, we can conclude from this visualization that the agent using intrinsic rewards turns out to be slow learner and comparatively needs a lot more episodes of training.

2. To achieve the 2nd Goal, the following visualization was used! In the pie chart we have action and reward distribution. Sequential color channel was used to color code the pie chart. Blue color shades represent Agent using Intrinsic Rewards and Red color represent agent using Extrinsic Rewards. The distributions can be clearly seen, for rewards majority of them are 0, showing that majority of the actions taken while training yielded no reward at all .

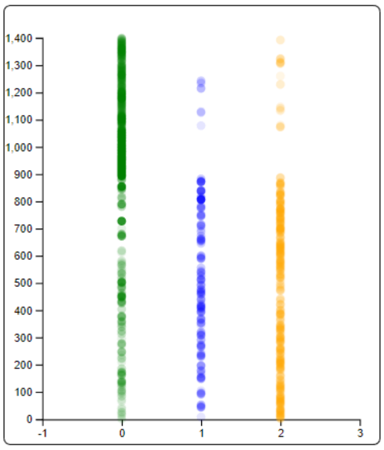


3. To achieve the 3rd Goal, a scatter plot was used. The states visited by the agent in an episode can only be visualized in a lower dimensional space all at once. Thus, the image features of the states were extracted from the ActorCriticNetwork which is used to predict the value and action distribution while training. Since the features are extracted from the network that was used to train the agent we can say that when a lower dimensional representation is obtained two points close to each other are indeed two states that are close to each other in the environment. To obtain the lower dimensional representation TSNE was used. Each point in the scatter plot is then color coded with respect to the amount of reward accumulated in that particular states. Thus points that are darker in shade shows that a high amount of reward was gained in that particular state.

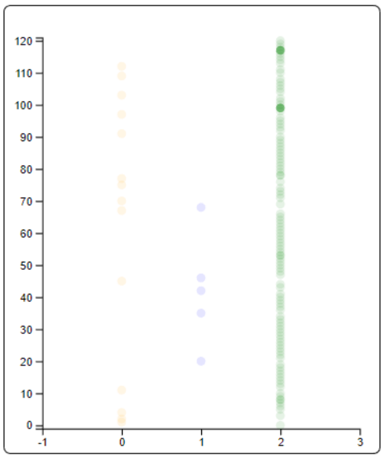
   Takeaways from the visualization, we can clearly see the difference in the states space explored by both the agents. While using Extrinsic Rewards the agent has explored a lot more state compared to the agent using Intrinsic Rewards. The later agent barely managed to accumulate any reward where as the former agent learned the tunneling effect and was able to accumulate a lot of rewards. A clear path taken by the agent can be seen as it accumulated a bunch of +7 rewards bouncing between top layer and wall of the breakout game. Sorting the reward based on the reward makes it easier to notice the states that lead to high reward but the temporal information is lost. Here we can see how the agent moved!

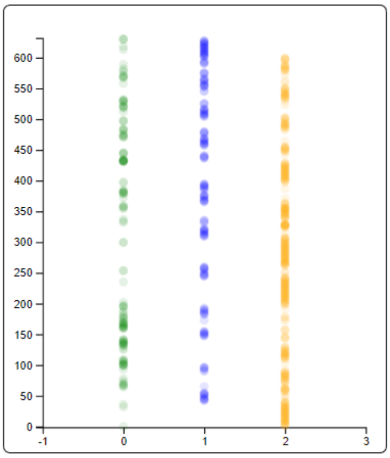


4. To achieve the final goal, a dot plot was used. The x-channel represents different actions, color coded by "Green" as "Left" action, "Blue" as "Fire" action and "Orange" as "Right" action. The y-channel represents the step number i.e. the state in which the action was taken in. The opacity of the plot is calculated using the reward the action resulted in. Higher opacity means the action resulted in high reward!



From this graph an interesting observation was done, the agent using only intrinsic reward was biased toward only one of the actions and was taking that action all the time where the action distribution of the other agent was more uniform. This might be the reason behind the better performance of one model over other.





References

[1] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

[2] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*, 2017.

[3] Junpeng Wang, Liang Gou, Han-Wei Shen, and Hao Yang. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE transactions on visualization and computer graphics*, 25(1):288–298, 2019.