Decision Transformers for Reinforcement Learning

Lidia Gutierrez | lg_087@usc.edu
SHINE Lab
Granada Hills Charter High School, Class of 2025
USC Viterbi | Computer Science and Robotics, SHINE 2023

Introduction

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Utilizing the simplicity and scalability of the transformer architecture and associated advances in language modeling such as GPT-X and BERT, we study a framework that abstracts Reinforcement Learning (RL) as a sequence modeling problem. Transformers can model high-dimensional distributions of semantic concepts at scale. We will train this on collected experience using a sequence modeling objective.

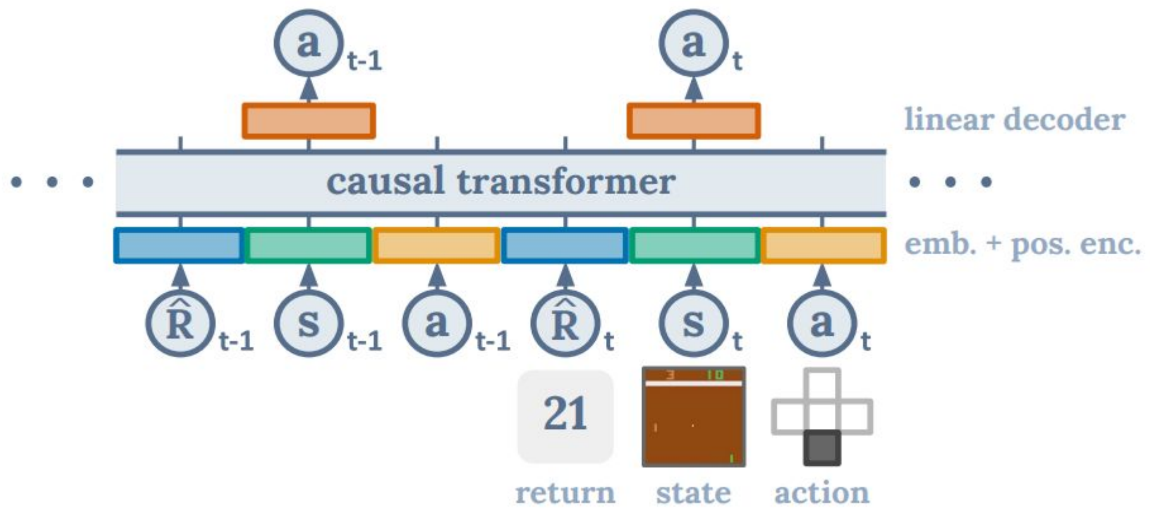

fig 1. decision transformer architecture

Objective & Impact of Professor’s Research

Professor Rahul Jain’s research focuses on Reinforcement Learning, Stochastic Control, Game Theory and Networks. Recently, he has succeeded in teaching a quadrupedal robot to walk. His recent studies are on ‘Safe and Intelligent Autonomy’: the development of reinforcement learning algorithms for robots and vehicles. Because Artificial Intelligence/Machine Learning is becoming increasingly prevalent, Professor Jain’s research is becoming increasingly more crucial to the scientific community.

Acknowledgements

I’d like to thank professor Jain for accepting me into his laboratory for research, and my mentor Rishabh for teaching me about the ins and outs of machine learning, as well as how to use different IDEs. I’d also like to thank my family for supporting me in my pursuit of the computer science field. Additionally I’d like to thank the conductor of the 7:14 Van Nuys to Union Station train for getting me to SHINE every day.

Research & Learning Process

We first started our research with a simple introduction to machine learning. Machine learning is a field of research that makes use of data to gain insights about it and use it for making predictions. Machine learning can be broken up into three categories, reinforcement learning, supervised and unsupervised. These three categories consist of many different algorithms.

We then studied neural networks, an architecture used in the field of machine learning. In a neural network, there are three types of layers; input layer, output layer and several hidden layers. Layers consist of nodes, which connect to each other in a way similar to the neurons in a brain, hence the name neural network. Observations are fed into the input layer, which are then processed by the hidden layers, the outcome of which will serve as an input for one of the nodes in the output layer. Furthermore, we discussed the activation functions and loss function associated with neural network, and how the neural networks learn.

We then turned our attention to methodologies within reinforcement learning paradigm, starting with Q-learning. In a Q-Learning algorithm, the agent maintains a q-table to maximize the cumulative reward achieved by the actions taken in an episode. Every step, the q-table is updated to take into account the reward returned by the environment. The main limitation of Q-learning is that it necessitates a q-table, making it impractical for continuous spaces.

Utilizing the learned concept of Neural Networks and Q-Learning, we then studied about Deep Q-Networks [2], which are a combination of both Q-learning algorithms and neural networks. We learned about 2 important concepts used in this algorithm, having an experience replay approach to mitigate the issue of correlated data, and maintaining a fixed target network in addition to the main Q-network to deal with the issue of constantly changing target values.

After understanding the basic concepts of neural networks and some RL algorithms, we then moved to the final part of the project where we applied transformers in an RL setup.

Methods & Results

We used a Decision Transformer [1], which models trajectories autoregressively with minimal modification to the transformer architecture.

Trajectory Representation - Instead of feeding the rewards directly, we feed the model with the returns-to-go. This is done since we would like the model to generate actions based on future desired returns, rather than past rewards.

$$\tau = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T\right).$$

Architecture - We feed the last K timesteps into Decision Transformer, for a total of 3K tokens (one
for each modality: return-to-go, state, or action). To obtain token embeddings, we learn a linear layer for each modality, which projects raw inputs to the embedding dimension, followed by layer normalization. The tokens are then processed by a GPT model, which predicts future action tokens via autoregressive modeling.

Training - We sampled mini batches of sequence length K from the dataset of offline trajectories. The prediction head corresponding to the input token (state) is trained to predict (action) – either with cross-entropy loss for discrete actions or mean-squared error for continuous actions – and the losses for each timestep are averaged.

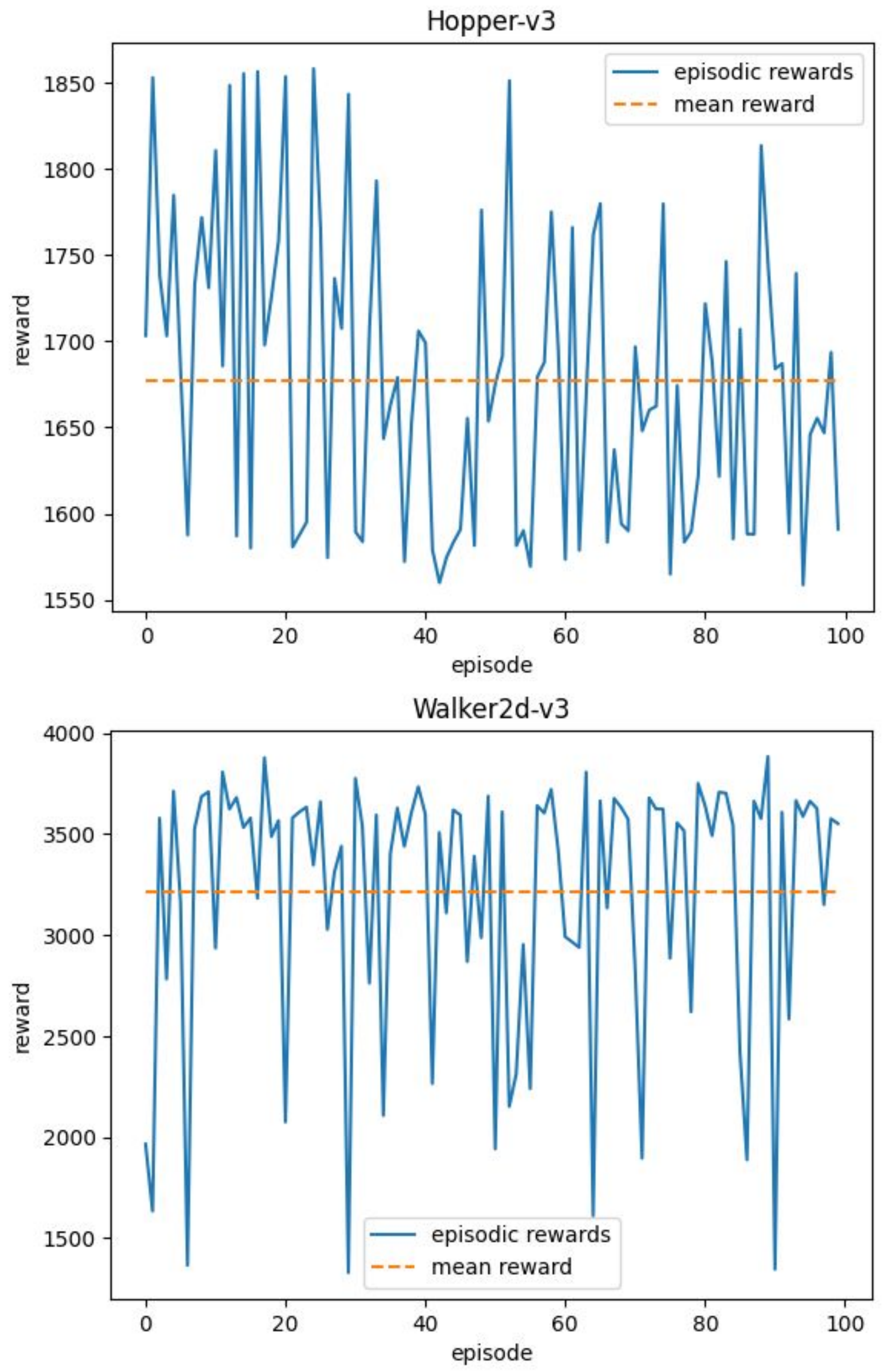

fig 2.- Performance of trained decision transformer model on Hopper-v3 and Walker2d-v3 mujoco environments.

Results Analysis

We achieved an average reward of **1672.53** on hopper-v3 mujoco environment, and an average reward of **3211.27** on walker-2d mujoco environment. Unlike prior approaches to RL that fit value functions or compute policy gradients, Decision Transformer simply outputs the optimal actions by leveraging a causally masked Transformer. On standard offline RL benchmarks, we observed Decision Transformer can match or outperform strong algorithms designed explicitly for offline RL with minimal modifications from standard language modeling architectures. By conditioning an autoregressive model on the desired return (reward), past states, and actions, the Decision Transformer model can generate future actions that achieve the desired return.

Next Steps for You & Advice to Future SHINE participants

I hope to continue to learn about machine learning. I plan to study computer science in the future with that studying I intend to contribute my knowledge to the development of the field.

To future SHINE students, my advice is to work with your mentors and to to talk to other students. Attend the events and make sure to check Minga often.

Citations

[1] Chen, Lili, et al. "Decision transformer: Reinforcement learning via sequence modeling." Advances in neural information processing systems 34 (2021): 15084-15097.

[2] Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).