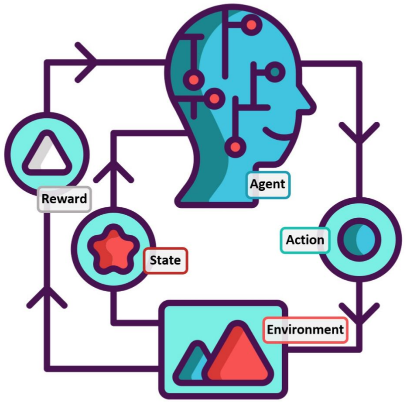
Transformers for Reinforcement Learning

Omar Bahri

Project Summary

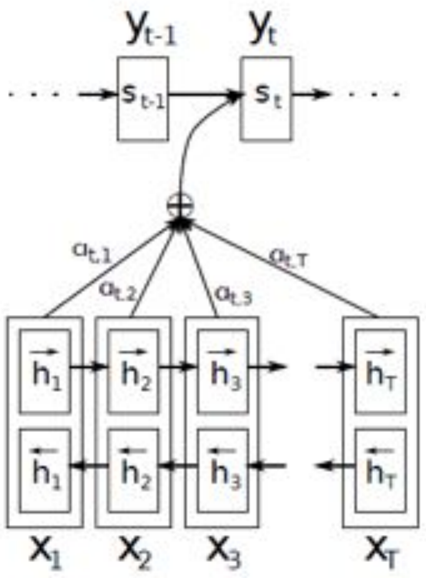
- Transformers are great for sequence modeling.
- By redefining RL as a sequence modeling problem, we can use transformers.
- Can the performance of Decision Transformer on **offline** benchmark datasets be achieved on a simple problem, with a custom dataset?

HOW?

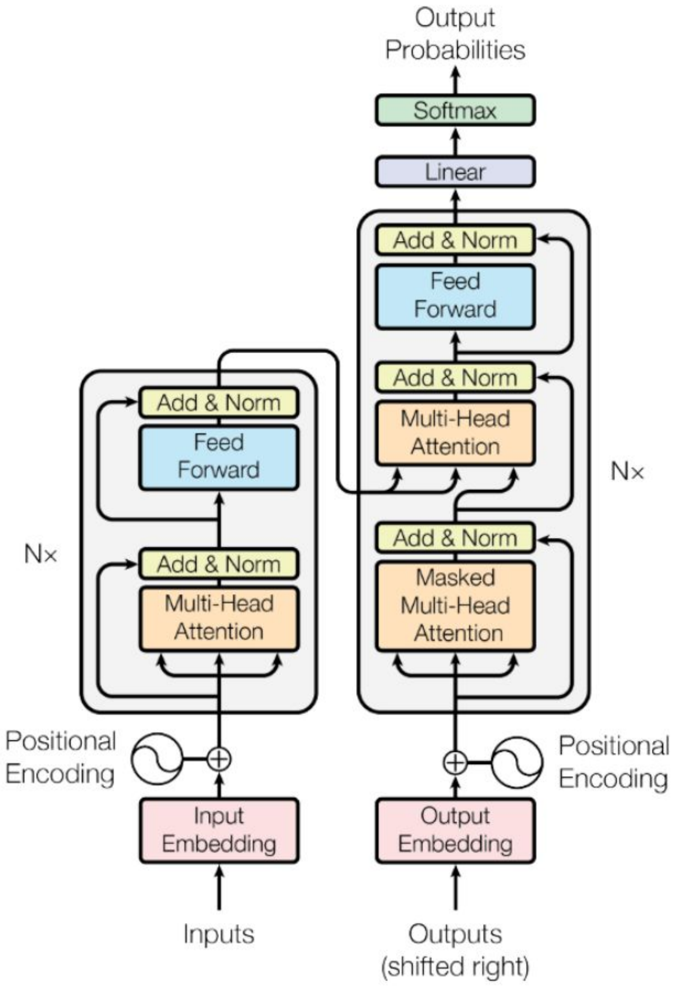


Attention

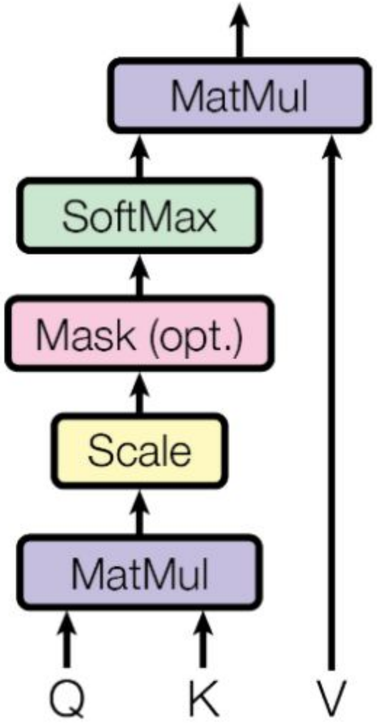
- Constructing a context vector:
  - **Normal LSTM**: take the embedding of the last layer (last hidden state).
  - **With attention**: consider all word embeddings simultaneously.



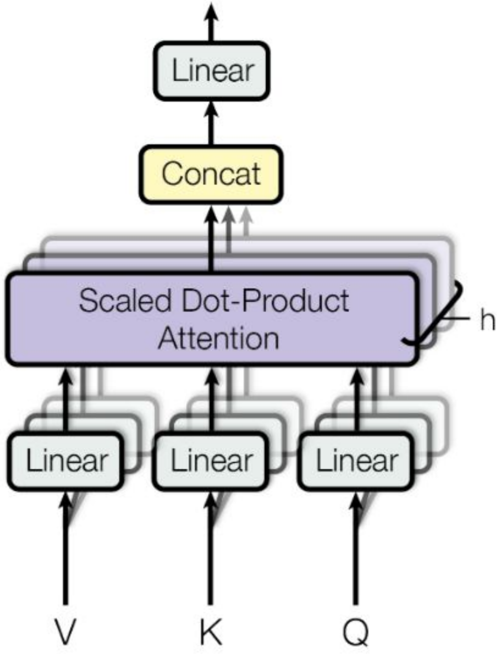
Transformers



Transformer



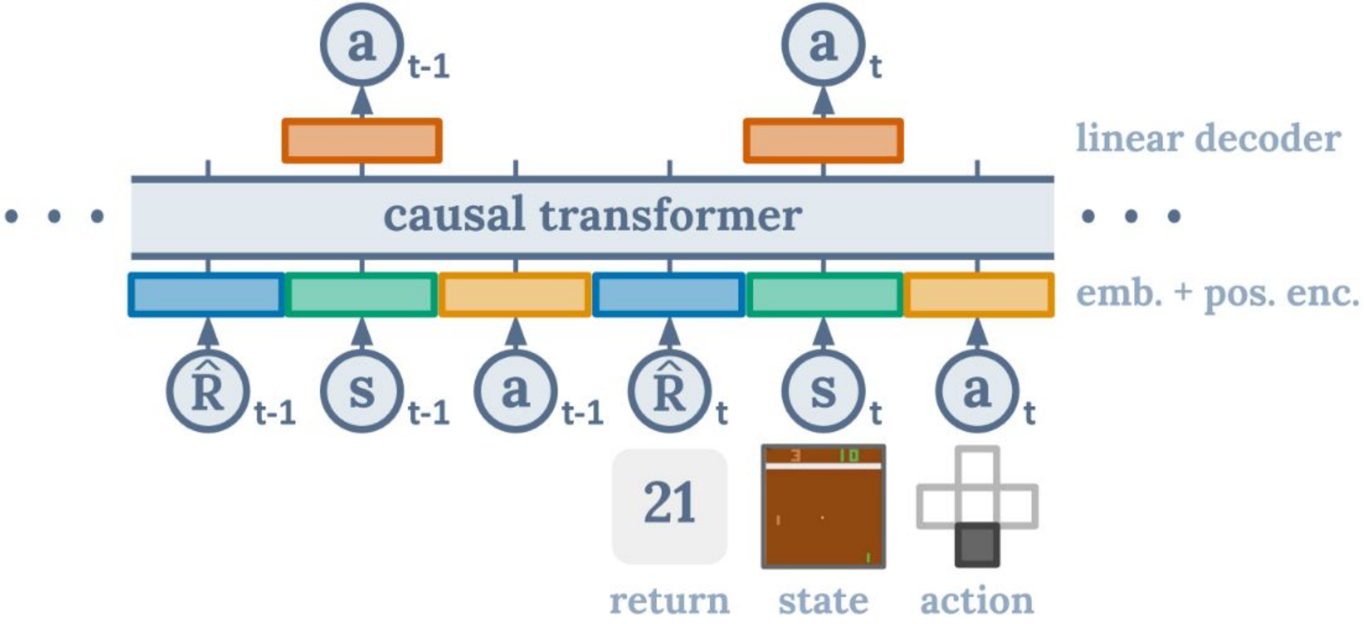
Scaled Dot Product Attention



Multi-Head Attention

Decision Transformer

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



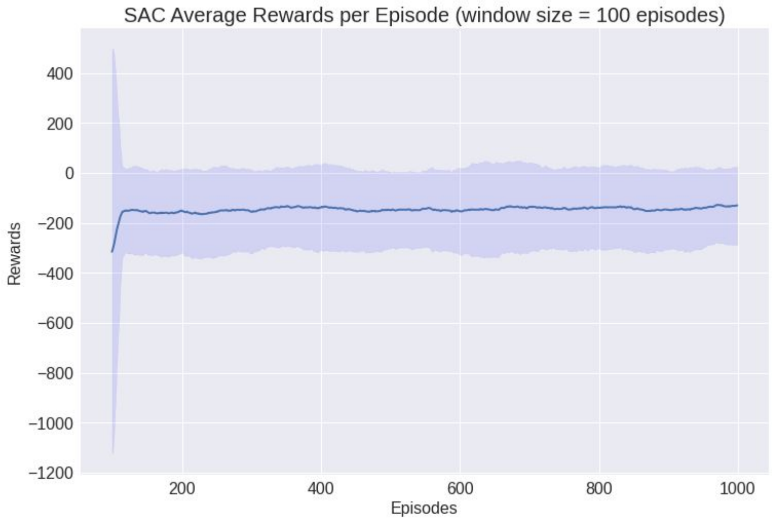
Methods

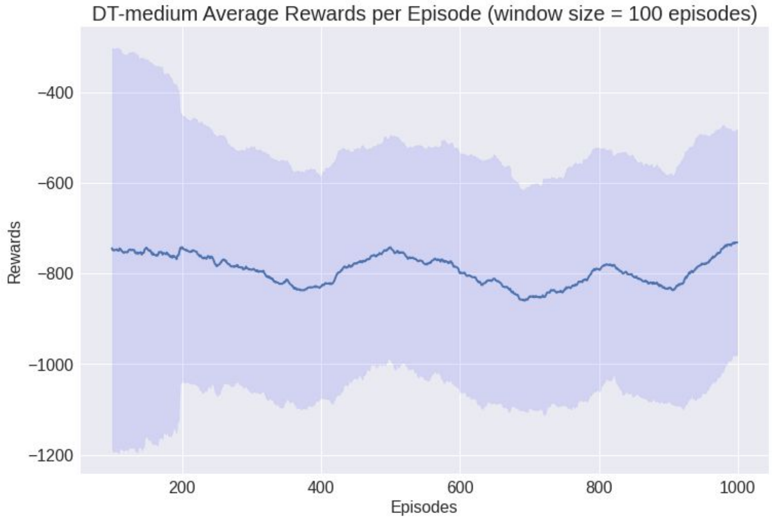
1. Generate three datasets using the OpenAI Gym Pendulum-v0 environment, in compliance with the D4RL benchmark format.
   - **pendulum-random**: random agent, 1,000,000 steps.
   - **pendulum-medium-replay**: the replay buffer of a Soft Actor-Critic agent, 1000 steps.
   - **pendulum-medium**: trained Soft Actor-Critic agent, 200,000 steps.

2. Train Decision Transformer on the datasets and compare to Soft Actor-Critic.

Results

Model	Average Rewards (200 episodes)
random agent	-1228.31
SAC agent	-138.44
pendulum-random	-1235.27
pendulum-medium -replay	-1025.76
pendulum-medium	-607.84





Conclusion

- Offline RL attempts to enable RL for real-world applications.
- The use of transformers might help this even further.
- However, good datasets are first needed.

- For this simple application:
  - Larger datasets?
  - More tuning?
  - Maybe it’s an overkill?