Shagun Uppal, Sarthak Bhagat, Deepak Srivatsav
(2016088, 2016189, 2016030)

Learning to Paint using Model-based Deep Reinforcement Learning

Instructions

Add detail to every slide that follows. Only add to the provided text box using the default font and size (Arial 18) and do not exceed the default text box of the slide.

Source Code

www.github.com/

Goal of the Project

Teach an agent to use a replicate an image by decomposing it to a set of strokes that can be painted on a canvas in some sequence. The agent will learn to paint the next stroke based on the current set of strokes on the canvas and the image being replicated. For this we follow the approach followed in paper Huang *et al.* alongside some of our own modifications. We aim to design a model-based TD3 approach and compare it to their model-based DDPG for the task of learning to paint.

Referenced Papers

Jia *et al.* takes into a similar yet different task of painting images for curriculum learning. The agent tries to learn to paint an image, given the patches from the reference image. It takes into consideration, the pen pressure, tilt as well as the color for the action space to be defined.

Xie *et al.* tackle a problem of automatic stroke generation in oriental ink painting. They model a brush as a reinforcement learning agent and learn brush strokes. They formulate stroke drawing as an MDP and use a Policy Gradient algorithm to learn an optimal drawing policy.

Song *et al.* explore an alternate method of learning to sketch, by means of encoders and decoders and Cycle Consistency. They also introduced a shortcut consistency that was enforced at the encoder bottleneck.

Novel Contribution by the Team

1. Trained the actor model using TD3 instead of DDPG. Within TD3, we implemented a policy update using both Proximal Policy gradient, and naive policy optimization.

   For TD3, we add an additional critic model in order to overcome the shortcomings of DDPG which is well known to overestimate the learnt value functions.
   PPO is an efficient version of TRPO, which uses a trust region to figure the direction and magnitude of gradient update

2. Compared the quantitative as well as qualitative performance of the TD3 (PPO and naive) model with baseline (with DDPG).

Environment, Agents, Rewards, and etc.

- The problem can be modeled as a Markov Decision Process.
- A state is defined by the state of the canvas, the target image and the step number.
- An action is defined by a set of parameters - position, shape, color and transparency of a stroke.
- Rewards are calculated based on the Wasserstein-l distance (which is commonly used in image restoration), using a discriminator.

RL Algorithm

1. **Deep Deterministic Policy Gradients**: DDPG is a suitable method for training the agent as the action space is continuous. A DDPG has an actor that predicts the action taken, and a critic to estimate the reward post taking an action, through means of a Bellman Equation and off policy data. This will act as our baseline to which we would compare our own approach.
2. **TD3**: DDPG is prone to overestimating Q values. TD3 introduces a second set of Q values. The minimum of the two Q values is used to form targets in the Bellman updates.
   TD3 updates policies a lot less frequently then DDPG.
   TD3 also adds noise to the target action, to help prevent the policy from exploiting Q-value errors.

Instructions for Results (4 - 5 slides max, use plots as far as possible)

<Show comparison if any with referenced papers>

<Demonstrate impact of any novelty you introduced>

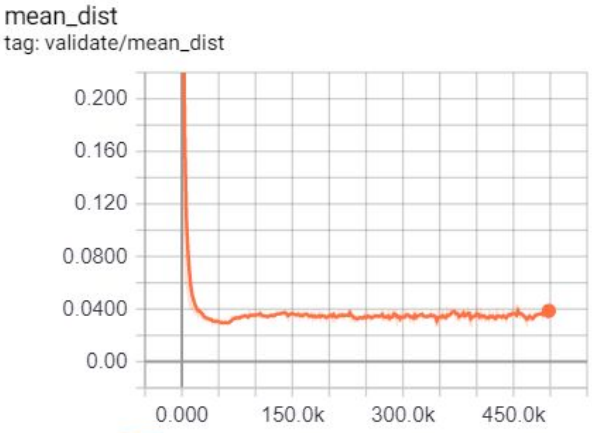
<Average cumulative reward or episodic reward averaged over many episodes>

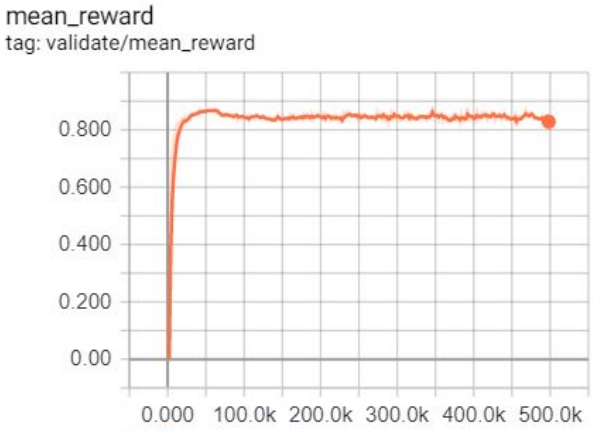
<Standard deviation or variance of the average cumulative reward>

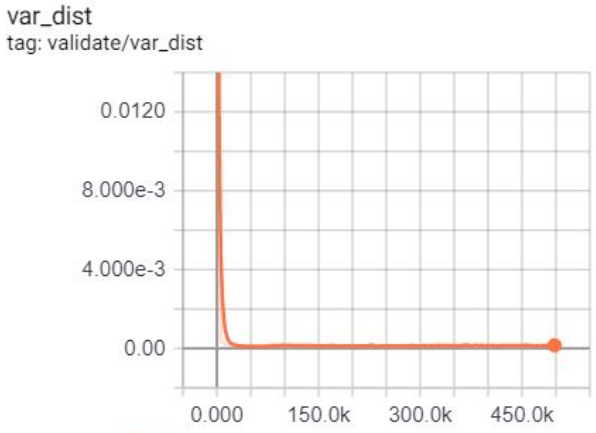
<Training Curves - Showing the progress of the training over time.>

<Any other plots or performance indicators>

Validation Set Reward, Mean and Variance of Wasserstein Distance







Results - 2

Results - 3

Results - 4

Results - 5