CS5180 Reinforcement Learning and Decision Making
Project Presentation

Lingzhi Kong, Shaoshu Xu

Northeastern University

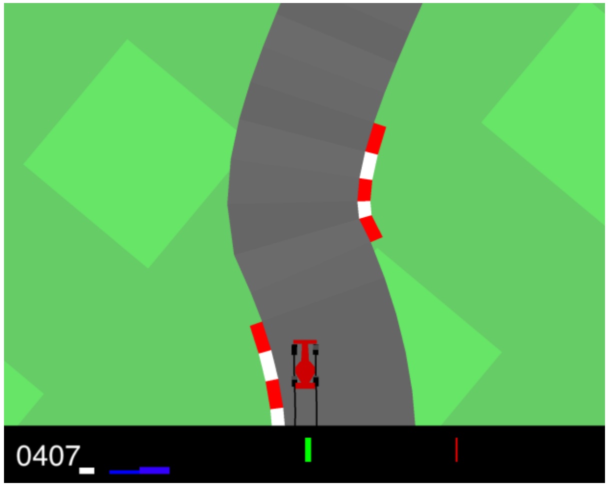
Dec. 3rd, 2019

Content

Content

Environment

- Environment
  - CarRacing-v0: continuous control task to learn from pixels; top-down racing environment;
  - State: adjacent 4 frames of 96*96 gray image in shape of (4, 96, 96)
  - Action: steering, gas, brake; every action will be repeated for 8 frames;
  - Reward: -0.1 every frame and +1000/N for every track tile visited, where N is the total number of tiles in track; if on the green area, -0.5 reward every frame;



Content

PPO

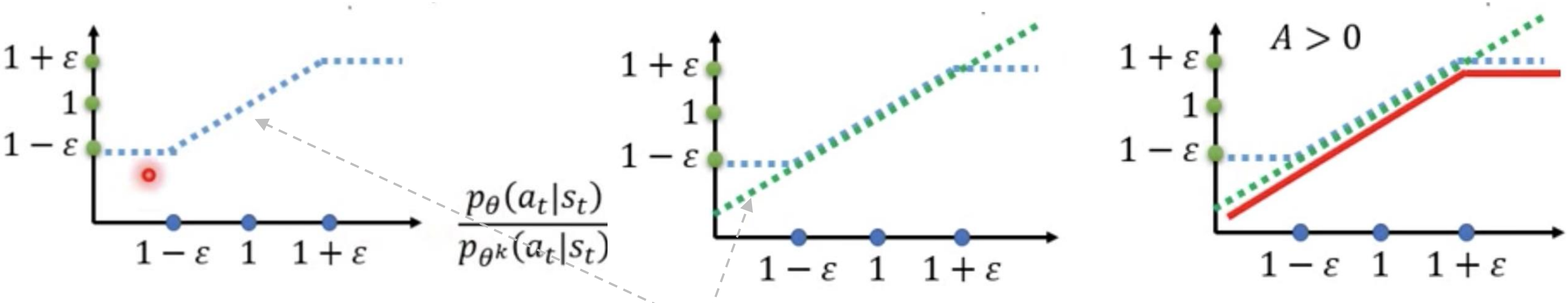
- Algorithm: PPO with Clipped Objective

Two policies: current policy $\pi_\theta(a|s)$, old policy $\pi_{\theta_{\text{old}}}(a|s)$

From the idea from importance sampling: $r(\theta) = \dfrac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$

Clip the estimated advantage function if the new policy is far away from the old one:

$$\text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)$$

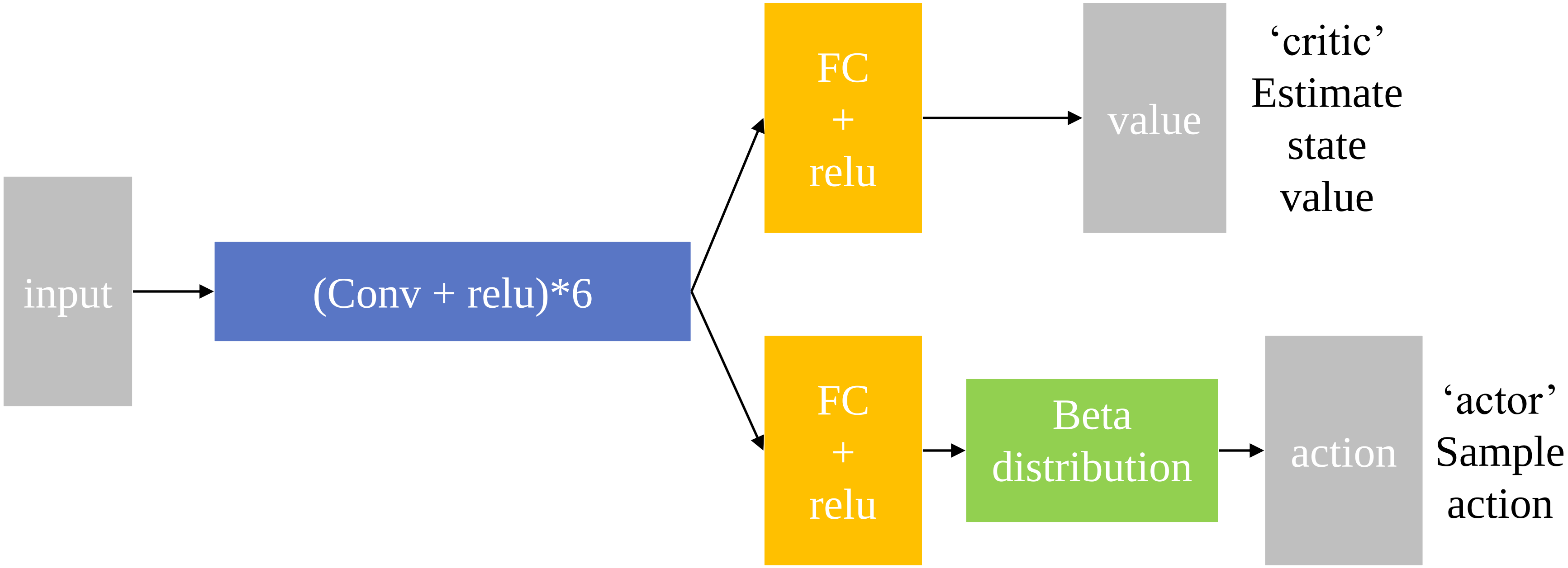


New objective function:

$$J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(\underline{r(\theta)\hat{A}_{\theta_{\text{old}}}(s,a)}, \underline{\text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}}(s,a)})]$$

PPO

- Training
  - Use a two heads network to represent the actor and critic respectively



PPO

- Algorithm: PPO with Clipped Objective

Algorithm 5 PPO with Clipped Objective

Input: initial policy parameters θ_0, clipping threshold ϵ

for $k = 0, 1, 2, ...$ **do**

Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau\sim\pi_k}\left[\sum_{t=0}^{T}\left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_k})\right]\right]$$

end for

Joshua Achiam, UC Berkeley, OpenAI, 2017

PPO

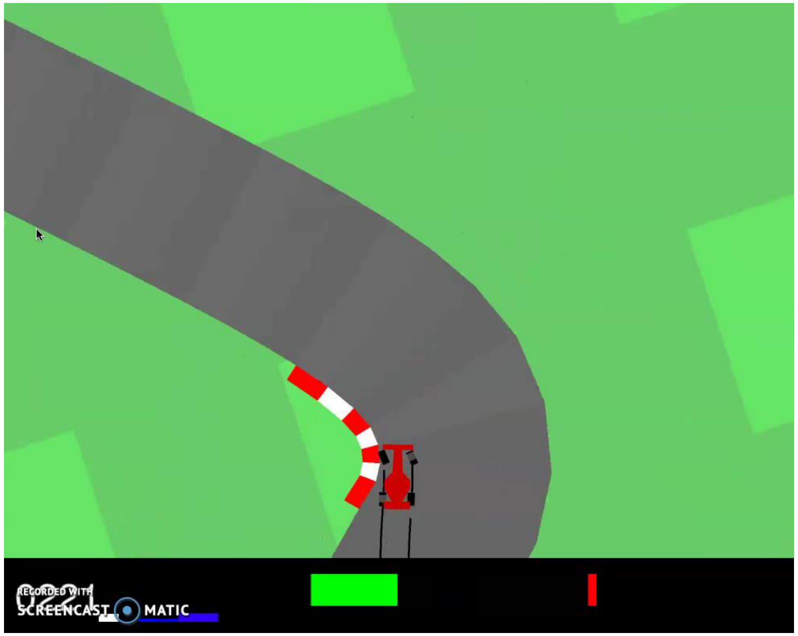
- Demo

Epoch = 90

Epoch = 2000

PPO

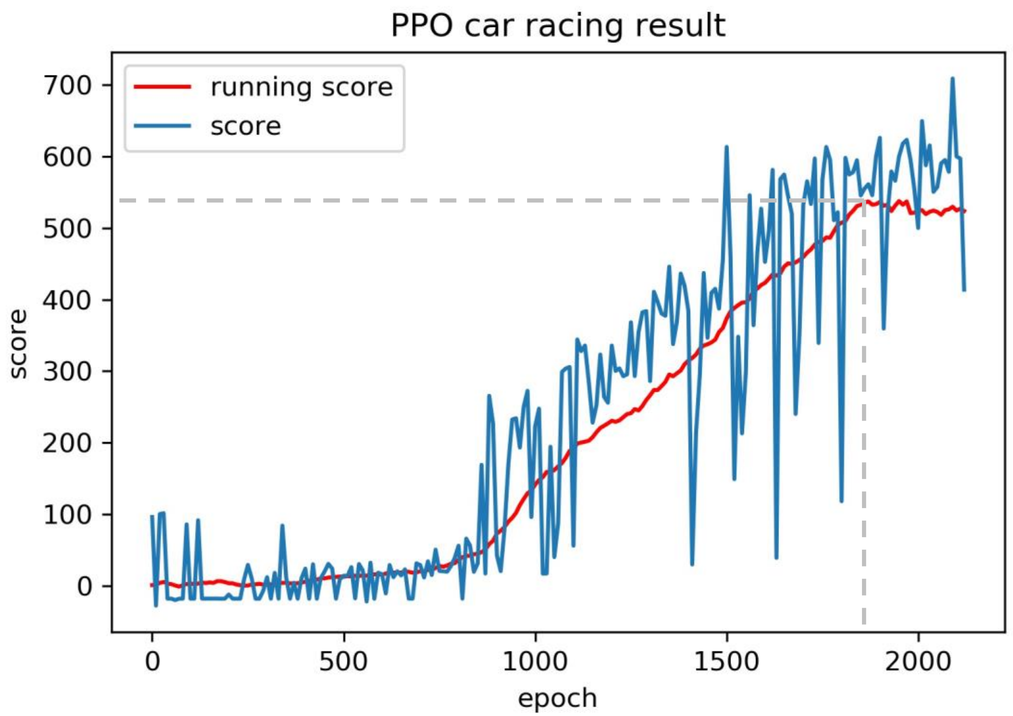
- Needs improvement



Turning too fast, skid, reckless driving

PPO

- Results



Total score of this epoch: score = score + reward

Weighted accumulative score: running_score = running_score * 0.99 + score * 0.01

PPO

- What will be done next?

Tuning hyper parameters of the network; Choosing a better clip-parameter e;

Try different environment;

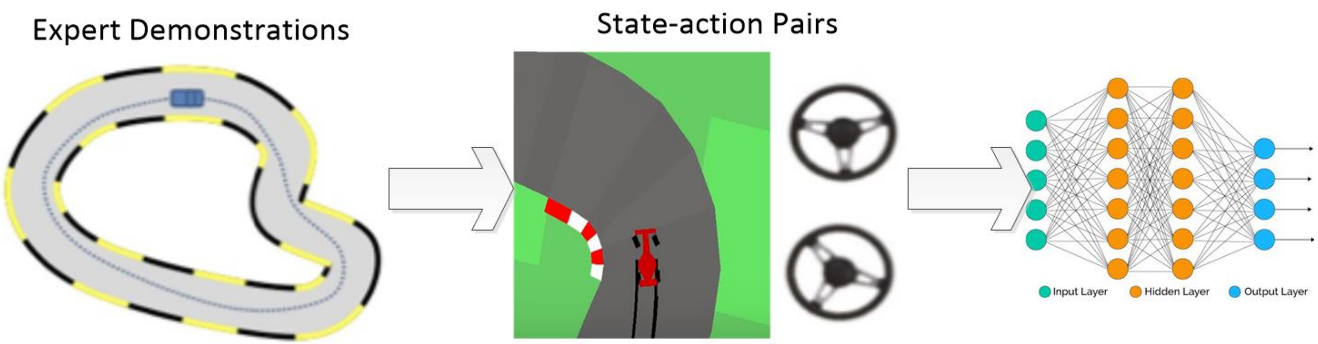
Augment the objective by adding an entropy bonus to ensure sufficient;

$$J^{\mathrm{CLIP'}}(\theta) = \mathbb{E}[J^{\mathrm{CLIP}}(\theta) - c_1(V_\theta(s) - V_{\mathrm{target}})^2 + c_2 H(s, \pi_\theta(.))]$$

Content

Dagger

- Behavior Cloning



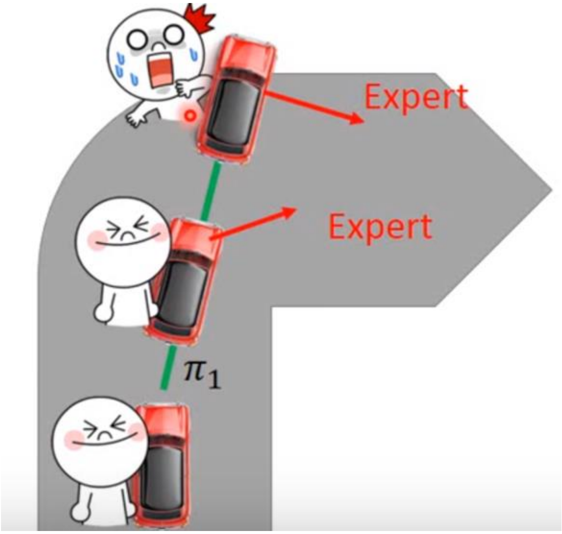
Ross, et al. 2011. Behavior Cloning

- Problems:
  - Expert only samples limited observations (states)
  - A small error e may lead to T*T*e mistakes after T steps.

Dagger

- Dataset Aggregation
  - Get expert by behavior cloning
  - Use Policy 1 to interact with the environment
    - Ask the expert to label the observations of Policy 1
    - Record states, and expert suggested actions.

  - Use new data to train Policy 2
  - Train Policy m on D1 ∪ … ∪ Dm



Dagger

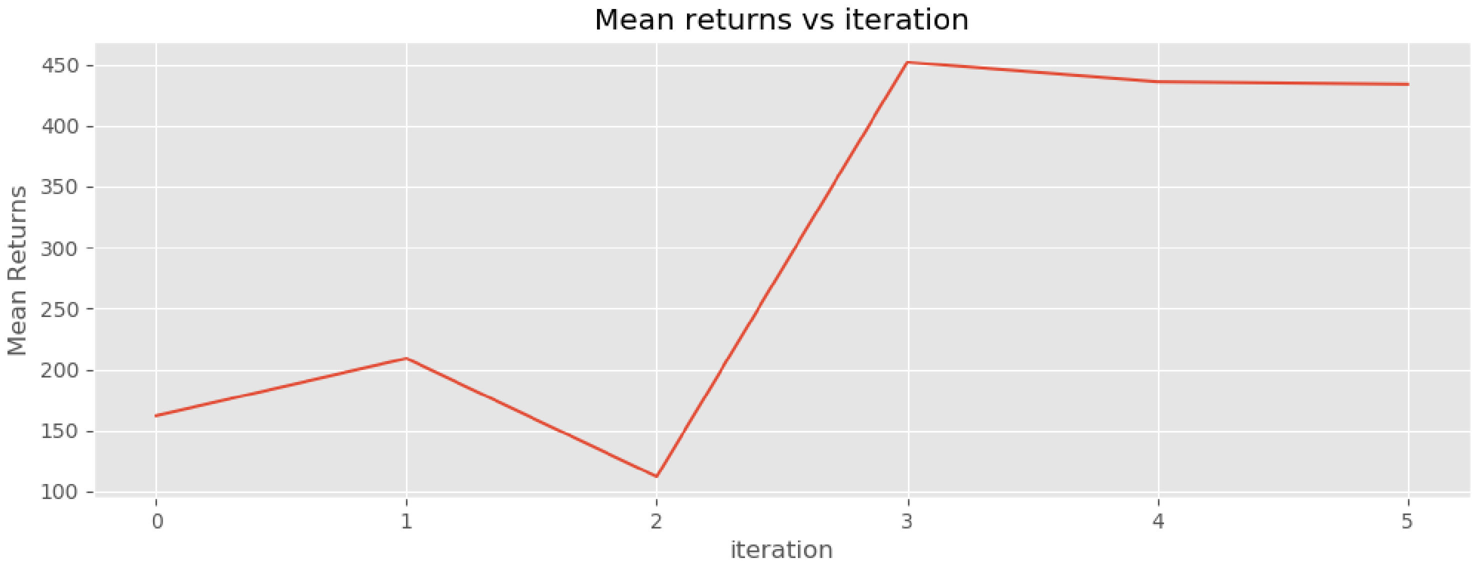
- Dagger Algorithm

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Ross et al. 2011

Dagger

- Experiments
  - iteration vs average return

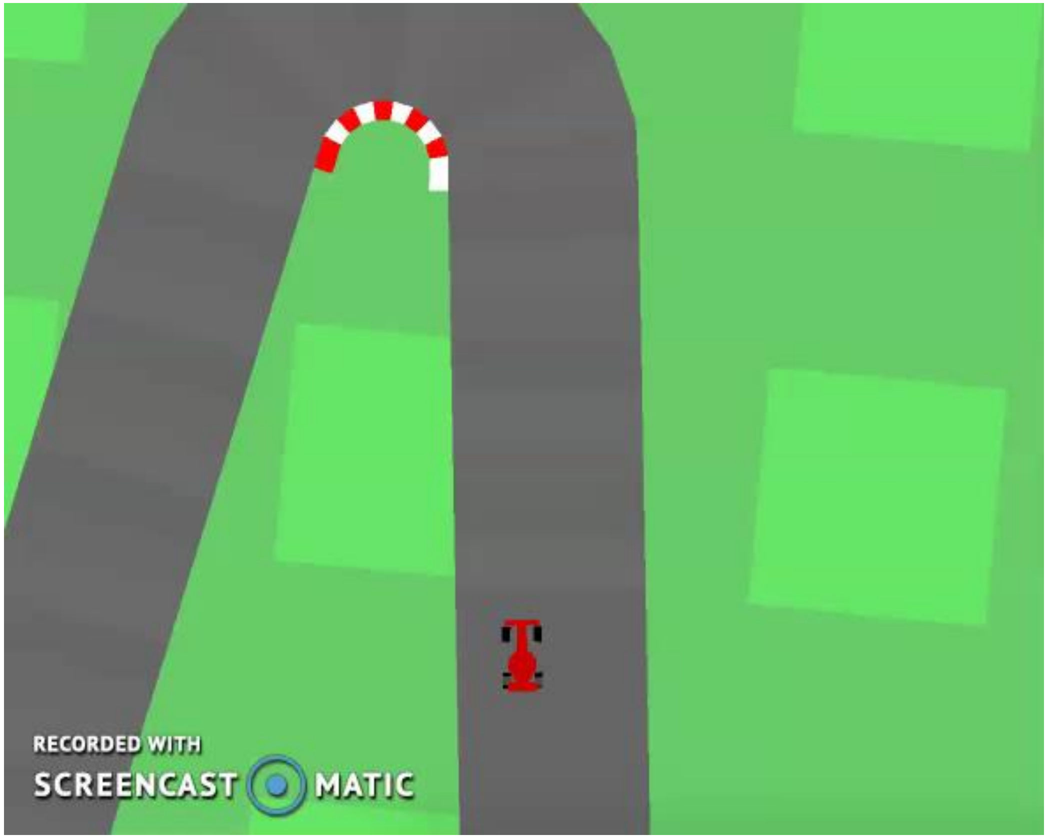


Dagger

- Video earlier iteration

Dagger

- Video final iteration



Dagger

- What will be done next?
  - Extend to other environment;
  - Imitate different policy;

Thank You!