Reinforcement Learning: Zero to ChatGPT

Marty Mukherjee

March 3, 2023

Department of Applied Mathematics

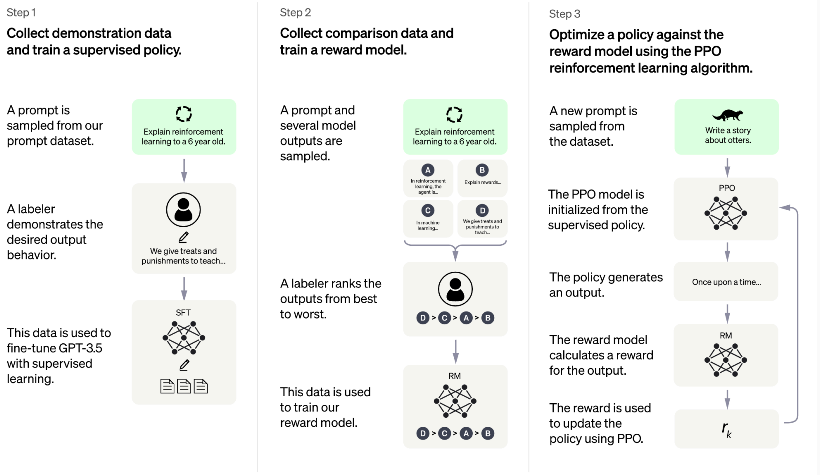

Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

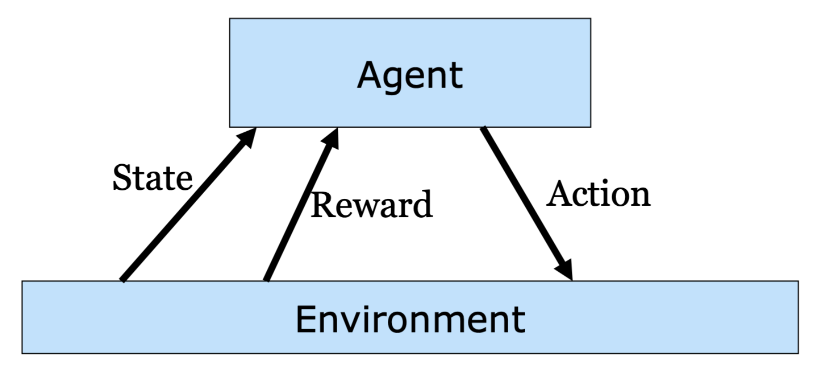
Table of contents

Intro

What is Reinforcement Learning

Wikipedia: reinforcement learning is an area of machine learning inspired by behavioral psychology, concerned with how software **agents** ought to take **actions** in an **environment** so as to maximize some notion of **cumulative reward**.

Reinforcement Learning Problem



(Poupart, 2022)

Goal: Learn to choose actions that maximize rewards

Motivation - Animal Psychology

Positive reinforcement:
- Food and petting

Negative reinforcement:
- Hunger and scolding

Goal: Maximize the quantity of positive reinforcement

Example: Inverted pendulum

State:

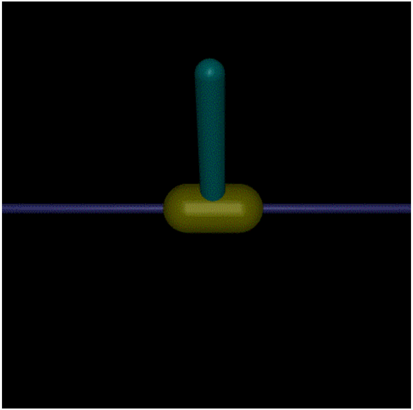
- Displacement of platform
- Velocity of platform
- Angular displacement of pole
- Angular velocity of pole

Action:

- Force applied to platform ($[-3, 3]$)

Reward $R(s, a)$:

- 1 for every time step the pole is upright
- 0 otherwise



(Goulão, 2022)

Important Components in RL

Dataset:

$$\{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$$

RL agents may or may not include the following components:

- Model $P(s'|s,a), P(r|s,a)$
  Environment dynamics and rewards
- Policy $\pi(a|s)$
  Agent action choices
- Value function $V(s)$
  Expected total rewards of the agent policy

Value function

Given policy π, estimate $V_\phi(s)$
Monte-Carlo estimation: $V_\phi(s) = E_\pi[\sum_t \gamma^t r_t]$

Bellman's Equation:
Optimal state value function $V^*(s)$:

$$V^*(s) = \max_a \left(E[r|s,a] + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right)$$

Update:

$$\phi \leftarrow \phi - \alpha \nabla_\phi \left(V_\phi(s_t) - (r_t + \gamma V_\phi(s_{t+1}))\right)^2$$

Extension: Q function

$Q(s,a)$ estimates total rewards of the agent policy at state s when applying action a

Update (Q-Learning):

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

Update (Deep Q-Learning):

$$w \leftarrow w - \alpha \nabla_w (r + \gamma \max_{a'} Q_w(s',a') - Q_w(s,a))^2$$

(Weights of $Q_w(s',a')$ are frozen)

Policy function

$\pi_\theta(a|s)$: Probability of choosing action a given state s

- Facilitates exploration

Goal of optimal policy: Choose a sequence of actions that maximize rewards.

$$\begin{aligned}
\nabla_\theta V_\theta(s_0) &= \textstyle\sum_{s\in S}\sum_{n=0}^{\infty}\gamma^n \Pr(s_0 \to s; n, \theta)\sum_a \nabla_\theta\, \pi_\theta(a|s)\, Q_\theta(s,a)\\
&= E_\theta[\textstyle\sum_{n=0}^{\infty}\gamma^n \sum_a Q_\theta(S_n,a)\nabla_\theta\, \pi_\theta(a|S_n)]\\
&= E_\theta\left[\textstyle\sum_{n=0}^{\infty}\gamma^n \sum_a \pi_\theta(a|S_n)\, Q_\theta(S_n,a)\,\frac{\nabla_\theta\, \pi_\theta(a|S_n)}{\pi_\theta(a|S_n)}\right]\\
&= E_\theta\left[\textstyle\sum_{n=0}^{\infty}\gamma^n\, Q_\theta(S_n,A_n)\,\frac{\nabla_\theta\, \pi_\theta(A_n|S_n)}{\pi_\theta(A_n|S_n)}\right]\\
&= E_\theta\left[\textstyle\sum_{n=0}^{\infty}\gamma^n\, G_n\,\frac{\nabla_\theta \pi_\theta(A_n|S_n)}{\pi_\theta(A_n|S_n)}\right]\\
&= E_\theta[\textstyle\sum_{n=0}^{\infty}\gamma^n\, G_n \nabla_\theta \log \pi_\theta(A_n|S_n)]
\end{aligned}$$

(Poupart, 2022)

Update (REINFORCE):

$$\theta \leftarrow \theta + \alpha\gamma^n G_n \nabla_\theta \log(\pi_\theta(a_n|s_n)), \text{ where } G_n = \sum_{t=0}^{T-n}\gamma^t r_{n+t}$$

Actor-Critic methods

Use a policy network $\pi_\theta(a|s)$ as an actor and a value network $V_\phi(s)$ as a critic.

Update (REINFORCE with Baseline):

$$\delta_n = G_n - V_\phi(s_n)$$

$$\phi \leftarrow \phi - \nabla_\phi \delta_n^2$$

$$\theta \leftarrow \theta + \alpha \gamma^n \delta_n \nabla_\theta \log(\pi_\theta(a_n|s_n))$$

See also: Advantage Actor-Critic, Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3)
- Uses Q Network as a critic rather than Value Network

Trust Region Policy Optimization (TRPO) & Proximal Policy Optimization (PPO)

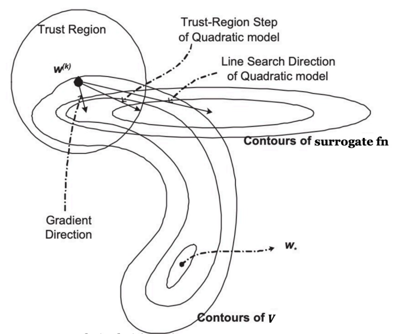
Recall Policy Gradient

Learning rate is difficult to set

- Small LR: Slow but reliable convergence
- Large LR: Fast but unreliable convergence

Trust region method

We want to optimize a surrogate objective function for the policy network using the value function. Surrogate objective may be trustable (close to V) only in a small region.
Limit search to small region.
Value network varies more smoothly with changes in policy network compared to policy network parameters.



Choi, Choi (2005)

Solution as proposed in TRPO

Define a policy trust region using the Kullback-Leibler divergence: $KL(\pi_\theta(\cdot|s)||\pi_{\theta_{old}}(\cdot|s))$

Optimization problem:

$$\theta \leftarrow \text{argmax}_\theta E\left[V^{\pi_\theta}(s) - V^{\pi_{\theta_{old}}}(s)\right] \text{ s.t. } E\left[KL(\pi_\theta(\cdot|s)||\pi_{\theta_{old}(\cdot|s)})\right] < \delta$$

Using the following approximation (Proof on next page):

$$V^{\pi_\theta}(s) - V^{\pi_{\theta_{old}}}(s) \approx \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A(s,a), \text{ where } A(s,a) = R(s,a) + \gamma V(s') - V(s)$$

Update step for TRPO:

$$\theta \leftarrow \text{argmax}_\theta E\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A(s,a)\right] \text{ s.t. } E\left[KL(\pi_\theta(\cdot|s)||\pi_{\theta_{old}(\cdot|s)})\right] < \delta$$

Derivation

$$\underset{\tilde\theta}{\operatorname{argmax}} E_{s\sim\mu_\theta,\, a\sim\pi_\theta}\left[\frac{\pi_{\tilde\theta}(a|s)}{\pi_\theta(a|s)} A_\theta(s,a)\right] = \underset{\tilde\theta}{\operatorname{argmax}} \sum_s \mu_\theta(s) \sum_a \pi_\theta(a|s)\left[\frac{\pi_{\tilde\theta}(a|s)}{\pi_\theta(a|s)} A_\theta(s,a)\right]$$

$$= \underset{\tilde\theta}{\operatorname{argmax}} \sum_s \mu_\theta(s) \sum_a \pi_{\tilde\theta}(a|s)\, A_\theta(s,a)$$

since $\mu_{\tilde\theta} \approx \mu_\theta$

$$\approx \underset{\tilde\theta}{\operatorname{argmax}} \sum_s \mu_{\tilde\theta}(s) \sum_a \pi_{\tilde\theta}(a|s) A_\theta(s,a)$$

since $\mu_{\tilde\theta}(s) \propto \sum_{n=0}^{\infty} \gamma^n P_{\tilde\theta}(s_n = s)$

$$= \underset{\tilde\theta}{\operatorname{argmax}} \sum_s \sum_{n=0}^{\infty} \gamma^n P_{\tilde\theta}(s_n = s) \sum_a \pi_{\tilde\theta}(a|s) A_\theta(s,a)$$

$$= \underset{\tilde\theta}{\operatorname{argmax}} E_{s_0, s_1, \ldots \sim P_{\tilde\theta},\, a_0, a_1, \ldots \sim \pi_{\tilde\theta}}\left[\sum_{n=0}^{\infty} \gamma^n A_\theta(s_n, a_n)\right]$$

Derivation (continued)

$$= \underset{\tilde{\theta}}{\text{argmax}}\, E_{s_0,s_1,\ldots\sim P_{\tilde{\theta}}\,,\, a_0,a_1,\ldots\sim\pi_{\tilde{\theta}}}\left[\sum_{n=0}^{\infty}\gamma^n A_\theta(s_n,a_n)\right]$$

since $A_\theta(s,a) = E_{s'\sim P(s'|s,a)}[r(s) + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)]$

$$= \underset{\tilde{\theta}}{\text{argmax}}\, E_{s_0,s_1,\ldots\sim P_{\tilde{\theta}}\,,\, a_0,a_1,\ldots\sim\pi_{\tilde{\theta}}}\left[\sum_{n=0}^{\infty}\gamma^n \left(r(s_n) + \gamma V^{\pi_\theta}(s_{n+1}) - V^{\pi_\theta}(s_n)\right)\right]$$

$$= \underset{\tilde{\theta}}{\text{argmax}}\, E_{s_0,s_1,\ldots\sim P_{\tilde{\theta}}\,,\, a_0,a_1,\ldots\sim\pi_{\tilde{\theta}}}\left[\sum_{n=0}^{\infty}\gamma^n r(s_n) - V^{\pi_\theta}(s_0)\right]$$

$$= \underset{\tilde{\theta}}{\text{argmax}}\, E_{s_0,s_1,\ldots\sim P_{\tilde{\theta}}\,,\, a_0,a_1,\ldots\sim\pi_{\tilde{\theta}}}\left[V^{\pi_{\tilde{\theta}}}(s_0) - V^{\pi_\theta}(s_0)\right]$$

$$= \underset{\tilde{\theta}}{\text{argmax}}\;\; E_{s_0\sim P}\left[V^{\pi_{\tilde{\theta}}}(s_0) - V^{\pi_\theta}(s_0)\right]$$

Constrained Optimization

$$\theta \leftarrow \text{argmax}_\theta E\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A(s,a)\right] \text{ s.t. } E\left[KL(\pi_\theta(\cdot|s)||\pi_{\theta_{old}(\cdot|s)})\right] < \delta$$

Problem with TRPO: Optimization problem is computationally expensive

Recall KL-Divergence:

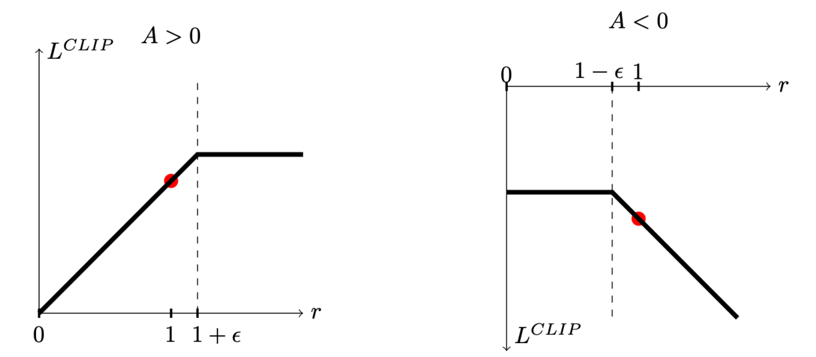
$$KL(\pi_\theta(\cdot|s)||\pi_{\theta_{old}}(\cdot|s)) = \sum_a \pi_\theta(a|s) \log(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)})$$

We are effectively constraining the ratio $\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$

Simpler Objective

Let's design a simpler objective that constrains $\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$

$$\text{argmax}_\theta E\left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A(s,a), clip(\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}, 1-\epsilon, 1+\epsilon)A(s,a)\right)\right]$$

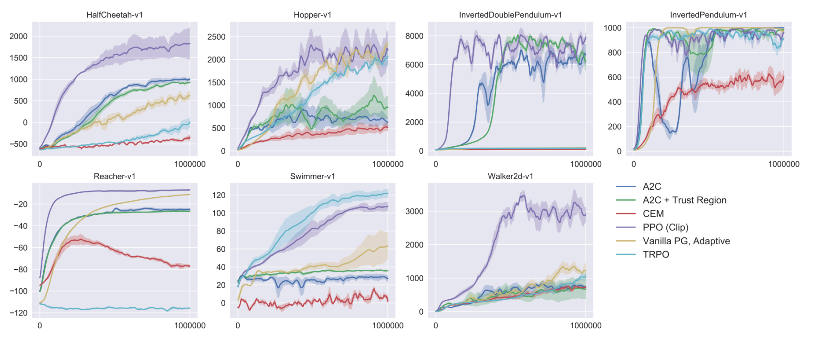


(Schulman et al., 2017)

PPO Algorithm

1. Initiate policy network parameter θ and value network parameter ϕ
2. Run the policy π_θ in the environment for T timesteps. Get dataset $\{s_t, a_t, R_t, s_{t+1}\}_{t=0}^{T-1}$
3. Compute the reward R_t and the advantage A_t
   $A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^{T-t-1}\delta_{T-1}$ where $\delta_t = R_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$, γ: Discount factor (≈ 0.99) and λ: Smoothing factor (≈ 0.95)
4. Compute $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$
5. Compute the objective function of the policy network:
   $L(\theta) = \frac{1}{T}\sum_{t=0}^{T-1} \min[r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t]$ where ϵ: clipping parameter (≈ 0.2)
6. Update $\theta \leftarrow \theta + \alpha_1 \nabla_\theta L(\theta)$
7. Compute the value network loss as:
   $J(\phi) = \frac{1}{T}\sum_{t=0}^{T-1} |V_\phi(s_t) - (R_t + \gamma V_\phi(s_{t+1}))|^2$
8. Update $\phi \leftarrow \phi - \alpha_2 \nabla_\phi J(\phi)$
9. Repeat steps 2–8 for multiple iterations

Results



(Schulman et al., 2017)

See also

- Soft Actor Critic (SAC)
  - Stochastic policy based on entropy maximization
- PPO-penalty
  - Constrained RL
- Conservative SAC
  - Offline RL
- C51
  - Distributional RL
- Decision Transformers
  - Partially Observable RL
- Hamilton Jacobi Bellman Proximal Policy Optimization (HJBPPO)
  - Continuous Time RL

PPO for ChatGPT

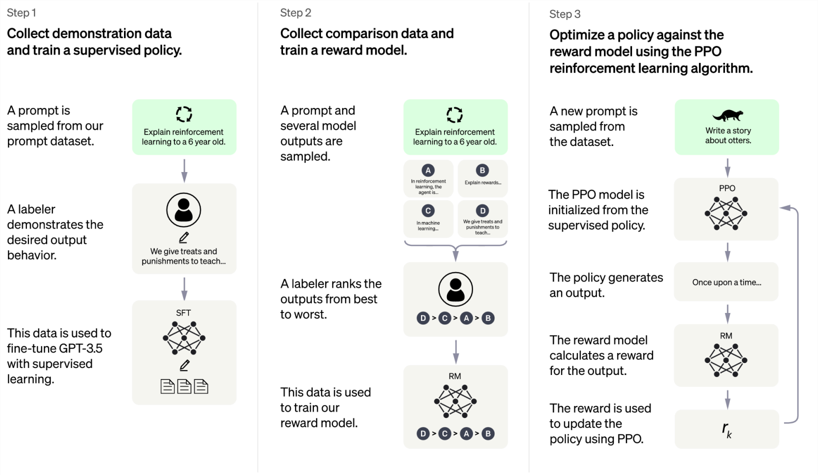
Reinforcement Learning from Human Feedback (RLHF)

Human AI trainers provided conversations where they played both sides - the user and a chatbot

Combine with InstructGPT dataset

Reward model

Comparison data
Model outputs multiple responses to a prompt
Human AI trainers ranked each response from best to worst
Rewards were computed from rank


Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

References

- OpenAI, *Introducing ChatGPT*, https://openai.com/blog/chatgpt, 2022
- Pascal Poupart, *CS885 Fall 2022 - Reinforcement Learning*, https://cs.uwaterloo.ca/ ppoupart/teaching/cs885-fall22/schedule.html, 2022
- Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction (2nd edition)*, 2018
- Schulman, Levine, Moritz, Jordan, Abbeel, *Trust Region Policy Optimization*, ICML, 2015
- Schulman, Wolski, Dhariwal, Radford, Klimov *Proximal policy optimization algorithms.* Preprint arXiv:1707.06347, 2017

Open Problem

How to use reinforcement learning to facilitate the training of transformers?