Lecture 5a: Trust Regions, Proximal Policies
CS885 Reinforcement Learning

2025-01-21

Complementary readings:
Schulman, Levine, Moritz, Jordan, Abbeel (2015) Trust Region Policy Optimization, ICML.
Schulman, Wolski, Dhariwal, Radford, Klimov (2017) Proximal Policy Optimization, arXiv.

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

Gradient Policy Optimization

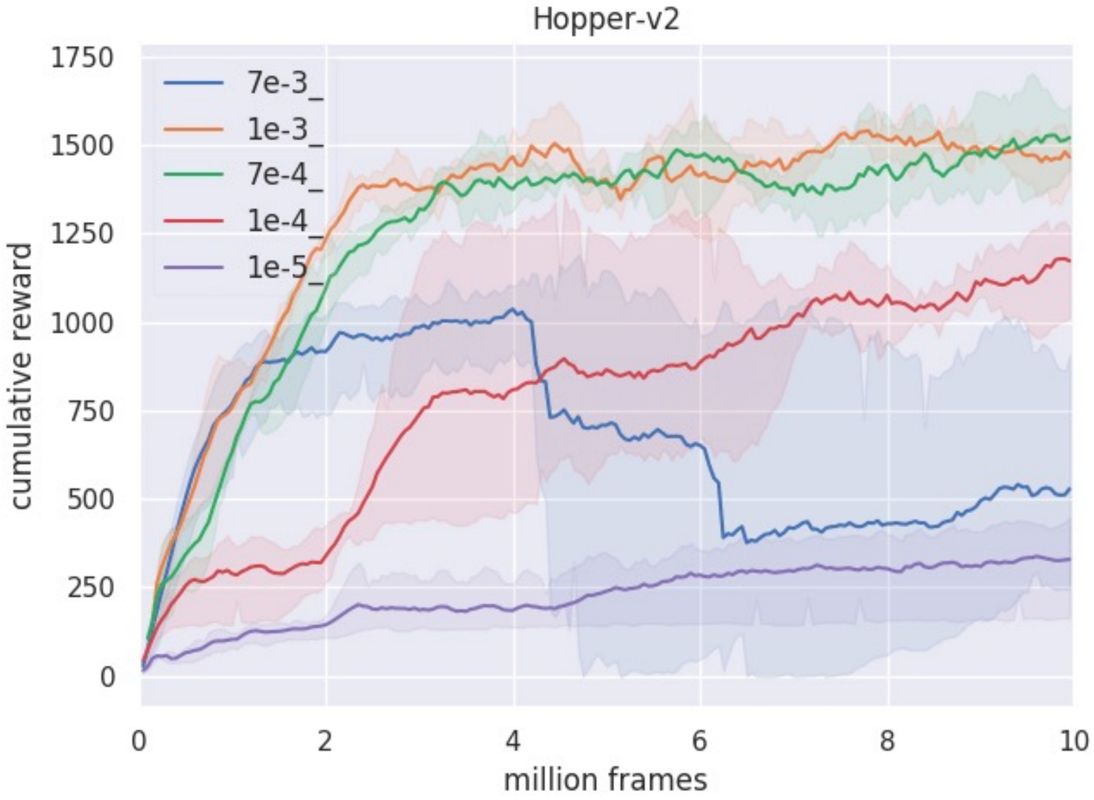
- REINFORCE algorithm
- Advantage Actor Critic (A2C)
- Deterministic Policy Gradient (DPG)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)

Recall Policy Gradient

Gradient update: $\theta \leftarrow \theta + \alpha\, \gamma^n A(s_n, a_n) \nabla \log \pi_\theta(a_n|s_n)$

α is difficult to set

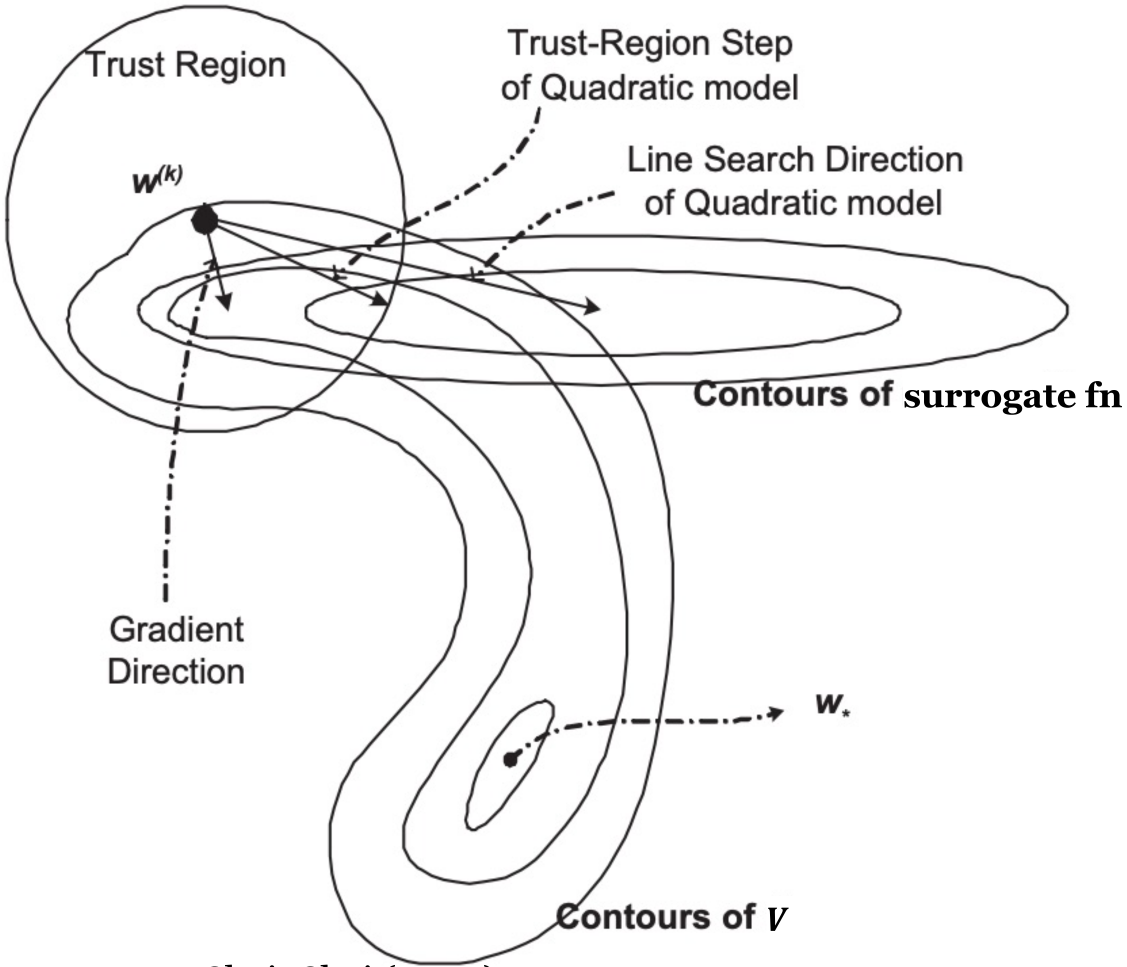
- Small α: slow
  but reliable convergence
- Big α: fast
  but unreliable



A2C on hopper-v2 with different α's
Wu, Sun et al. (2018)

Trust Region Method

- We often optimize a surrogate objective (approximation of V)
- Surrogate objective may be trustable (close to V) only in a small region
- Limit search to small trust region



Choi, Choi (2005)

UNIVERSITY OF WATERLOO

Trust Region for Policies

- Let θ be the parameters for policy $\pi_\theta(a|s)$
- We can define a region around θ: $\{\theta' | D(\theta, \theta') < \delta\}$
  or around π_θ: $\{\theta' | D(\pi_\theta, \pi_{\theta'}) < \delta\}$
  where D is a distance measure
- V often varies more smoothly with π_θ than θ
  small change in π_θ —usually→ small change in V
  small change in θ —more often→ large change in V
- Hence, define policy trust regions

Kullback-Leibler Divergence

KL-Divergence is a common distance measure for distributions:

$$D_{KL}(p,q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Intuition: expectation of the logarithm difference between p and q

KL-Divergence for policies at a state s:

$$D_{KL}\left(\pi_\theta(\cdot\,|s), \pi_{\tilde{\theta}}(\cdot\,|s)\right) = \sum_a \pi_\theta(a|s) \log \frac{\pi_\theta(a|s)}{\pi_{\tilde{\theta}}(a|s)}$$

Trust Region Policy Optimization

- Consider an initial state distribution $p(s_0)$

- Update step: $\theta \leftarrow \underset{\tilde{\theta}}{\operatorname{argmax}} E_{s_0 \sim p}[V^{\pi_{\tilde{\theta}}}(s_0) - V^{\pi_\theta}(s_0)]$

  $\text{subject to} \max_s D_{KL}\big(\pi_\theta(\cdot\,|s), \pi_{\tilde{\theta}}(\cdot\,|s)\big) \leq \delta$

Reformulation

- Since the objective is not directly computable, let's approximate it:

$$\underset{\tilde{\theta}}{\operatorname{argmax}}\, E_{s_0\sim p}[V^{\pi_{\tilde{\theta}}}(s_0) - V^{\pi_\theta}(s_0)] \approx \underset{\tilde{\theta}}{\operatorname{argmax}}\, E_{s\sim\mu_\theta,\, a\sim\pi_\theta}\left[\frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)} A_\theta(s,a)\right]$$

where $\mu_\theta(\mathrm{s})$ is the stationary state distribution for π

- Let's also relax the bound on the max KL-divergence to a bound on the expected KL-divergence

$$\max_{\mathrm{s}} D_{KL}\big(\pi_\theta(\cdot\,|s), \pi_{\tilde{\theta}}(\cdot\,|s)\big) \le \delta$$

is relaxed to $E_{s\sim\mu_\theta}\left[D_{KL}\left(\pi_\theta(\cdot\,|s), \pi_{\tilde{\theta}}(\cdot\,|s)\right)\right] \le \delta$

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

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

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

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

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

Trust Region Policy Optimization (TRPO)

Initialize π_θ to anything
Loop forever (for each episode)
  Sample s_0 and set $n \leftarrow 0$
  Repeat N times
    Sample $a_n \sim \pi_\theta(a|s_n)$
    Execute a_n, observe s_{n+1}, r_n
    $\delta \leftarrow r_n + \gamma \max_{a_{n+1}} Q_w(s_{n+1}, a_{n+1}) - Q_w(s_n, a_n)$
    $A(s_n, a_n) \leftarrow r_n + \gamma \max_{a_{n+1}} Q_w(s_{n+1}, a_{n+1}) - \sum_a \pi_\theta(a|s_n) Q_w(s_n, a)$
    Update Q: $w \leftarrow w + \alpha_w \, \delta \, \nabla_w Q_w(s_n, a_n)$
    $n \leftarrow n + 1$
  Update π: $\theta \leftarrow \underset{\tilde{\theta}}{\operatorname{argmax}} \frac{1}{N} \sum_{n=0}^{N-1} \frac{\pi_{\tilde{\theta}}(a_n|s_n)}{\pi_\theta(a_n|s_n)} A_\theta(s_n, a_n)$
    subject to $\frac{1}{N} \sum_{n=0}^{N-1} D_{KL}\left(\pi_\theta(\cdot\,|s_n), \pi_{\tilde{\theta}}(\cdot\,|s_n)\right) \leq \delta$

linear approximation

quadratic approximation

Constrained Optimization

- TRPO is conceptually and computationally challenging in large part because of the constraint in the optimization.

$$\max_s D_{KL}\left(\pi_\theta(\cdot|s), \pi_{\tilde{\theta}}(\cdot|s)\right) \leq \delta$$

- What is the effect of the constraint?
- Recall KL-Divergence:

$$D_{KL}\left(\pi_\theta(\cdot|s), \pi_{\tilde{\theta}}(\cdot|s)\right) = \sum_a \pi_\theta(a|s) \log \frac{\pi_\theta(a|s)}{\pi_{\tilde{\theta}}(a|s)}$$

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

Simpler Objective

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

$$\underset{\tilde{\theta}}{\text{argmax}}\, E_{s\sim\mu_\theta,\, a\sim\pi_\theta} \min \begin{Bmatrix} \frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)} A_\theta(s,a), \\ clip\left(\frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)}, 1-\epsilon, 1+\epsilon\right) A_\theta(s,a) \end{Bmatrix}$$

where $clip(x, 1-\epsilon, 1+\epsilon) = \begin{cases} 1-\epsilon & if\ x < 1-\epsilon \\ x & if\ 1-\epsilon \leq x \leq 1+\epsilon \\ 1+\epsilon & if\ x > 1+\epsilon \end{cases}$

Proximal Policy Optimization (PPO)

PPO version based on TRPO

Initialize π_θ to anything
Loop forever (for each episode)
  Sample s_0 and set $n \leftarrow 0$
  Repeat N times
    Sample $a_n \sim \pi_\theta(a|s_n)$
    Execute a_n, observe s_{n+1}, r_n
    $\delta \leftarrow r_n + \gamma \max_{a_{n+1}} Q_w(s_{n+1}, a_{n+1}) - Q_w(s_n, a_n)$
    $A(s_n, a_n) \leftarrow r_n + \gamma \max_{a_{n+1}} Q_w(s_{n+1}, a_{n+1}) - \sum_a \pi_\theta(a|s_n) Q_w(s_n, a)$
    Update Q: $w \leftarrow w + \alpha_w \, \delta \, \nabla_w Q_w(s_n, a_n)$
    $n \leftarrow n + 1$
  Update π:

optimize by stochastic gradient descent

$$\theta \leftarrow \operatorname{argmax}_{\tilde{\theta}} \frac{1}{N} \sum_{n=0}^{N-1} \min \left\{ \begin{array}{c} \frac{\pi_{\tilde{\theta}}(a_n|s_n)}{\pi_\theta(a_n|s_n)} A(s_n, a_n), \\ clip\left(\frac{\pi_{\tilde{\theta}}(a_n|s_n)}{\pi_\theta(a_n|s_n)}, 1-\epsilon, 1+\epsilon\right) A(s_n, a_n) \end{array} \right\}$$

UNIVERSITY OF WATERLOO

Proximal Policy Optimization (PPO)

PPO version based on Reinforce with a Baseline

Initialize π_θ and V_w to anything
Loop forever (for each episode)
    Generate episode $s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{N-1}, a_{N-1}, r_{N-1}$ with π_θ
    Loop for each step of the episode $n = 0, 1, \ldots, N-1$
        $G_n \leftarrow \sum_{t=0}^{N-1-n} \gamma^t \, r_{n+t}$
        $\delta \leftarrow G_n - V_w(s_n)$
        Update value function: $w \leftarrow w + \alpha_w \, \delta \nabla_w V_w(s_n)$
        $A(s_n, a_n) \leftarrow \delta$
    Update π: optimize by stochastic gradient descent

$$\theta \leftarrow \underset{\tilde{\theta}}{\operatorname{argmax}} \frac{1}{N} \sum_{n=0}^{N-1} \min \left\{ \begin{array}{c} \frac{\pi_{\tilde{\theta}}(a_n|s_n)}{\pi_\theta(a_n|s_n)} A(s_n, a_n), \\ clip\left(\frac{\pi_{\tilde{\theta}}(a_n|s_n)}{\pi_\theta(a_n|s_n)}, 1-\epsilon, 1+\epsilon\right) A(s_n, a_n) \end{array} \right\}$$

Application: Large Language Models (LLMs)

MISTRAL AI_

OpenAI

Claude BY ANTHROPIC

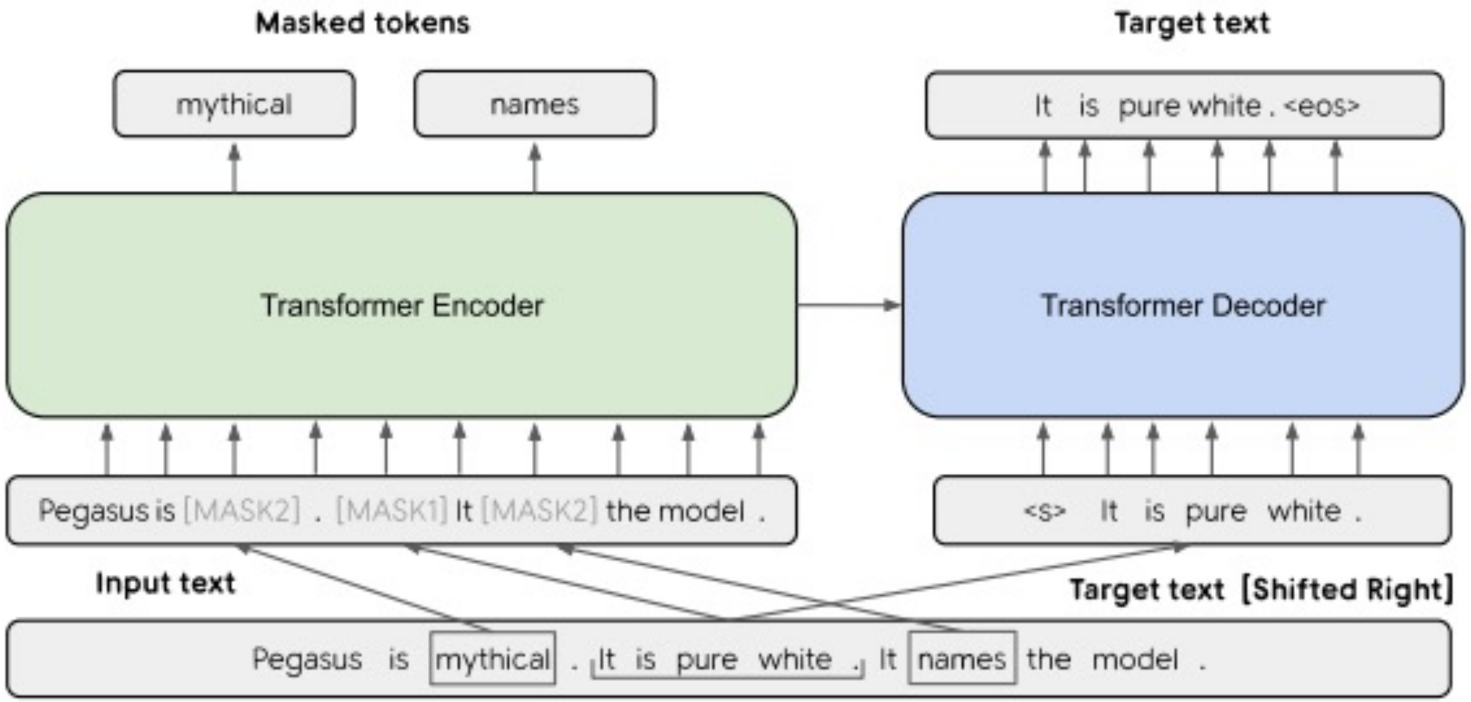
LLaMA by Meta

cohere

Gemini

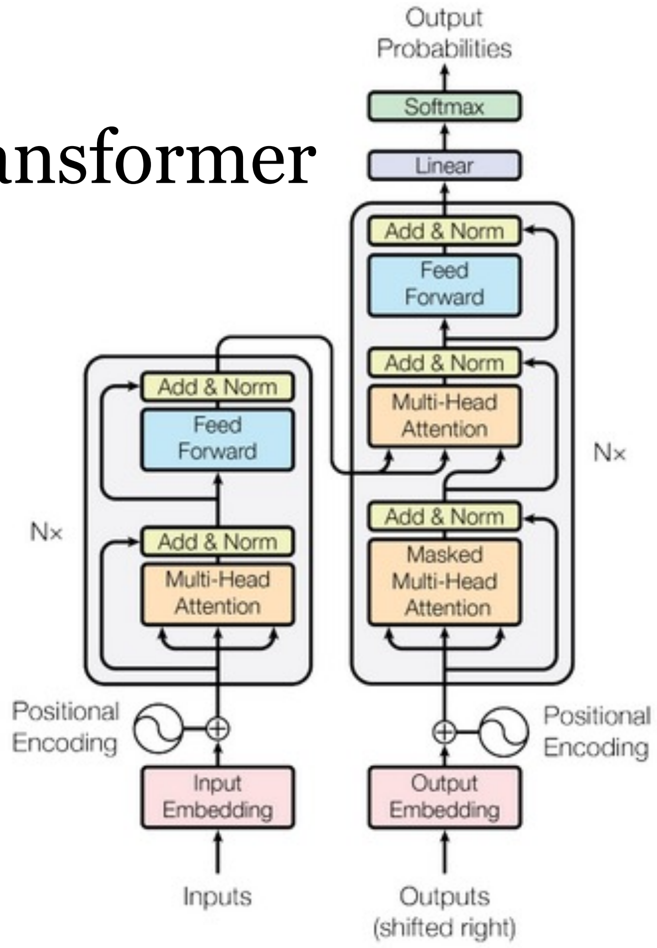
Self-Supervised Learning in LLMs

Encoder and Decoder



Credit: Zhang et al., 2020

Transformer



Credit: Vaswani et al., 2017

UNIVERSITY OF WATERLOO

GPT Models

InstructGPT (GPT-3)
January 27, 2022

Reinforcement Learning from human feedback

Ouyang et al. (2022) Training language models to follow instructions with human feedback

ChatGPT (GPT-3.5)
November 30, 2022

Multi-turn conversations

No techreport

GPT-4
March 14, 2023

Multi-modal (text and images)

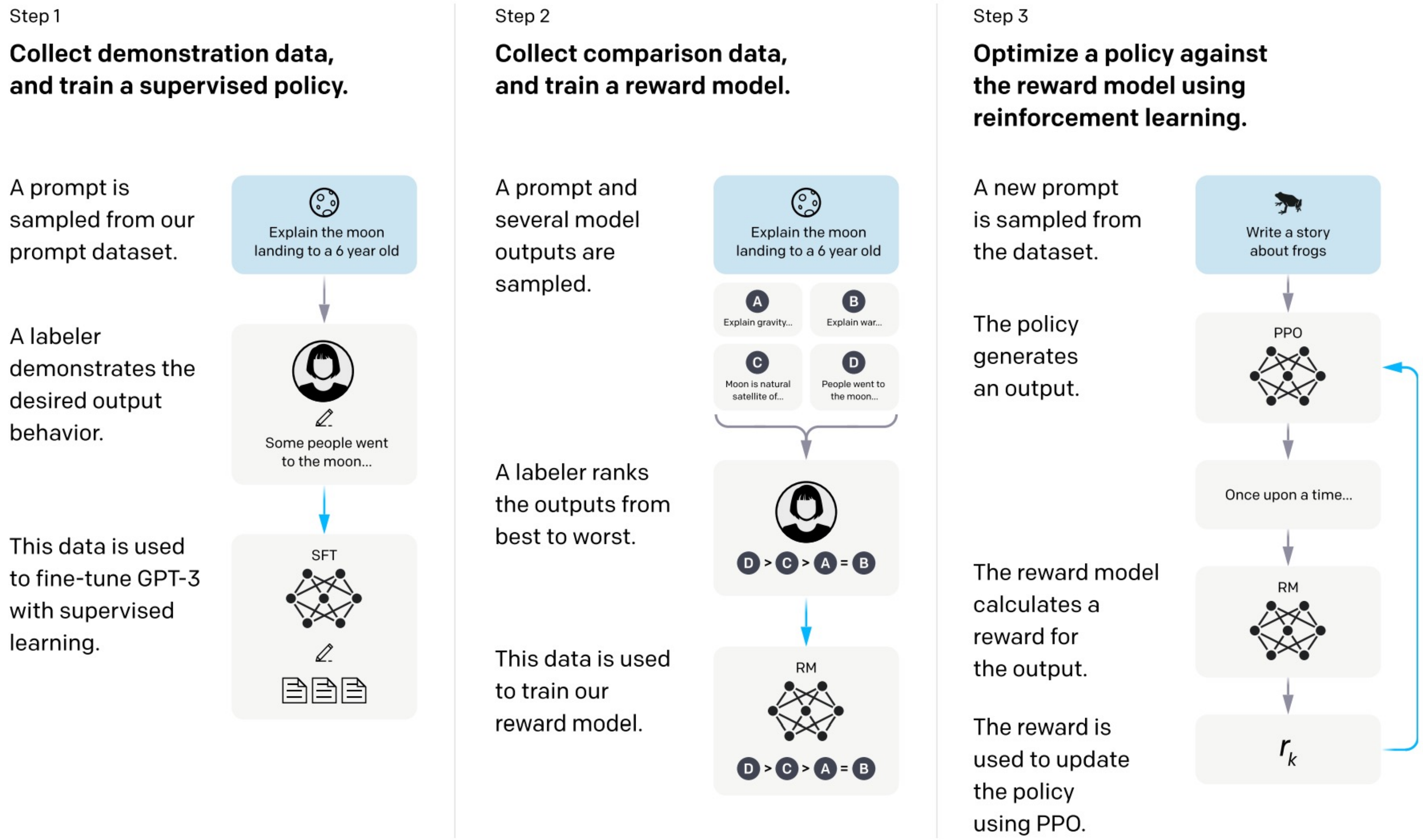
GPT-4 Technical Report (2023)

GPT Omni GPT-4o
May 13, 2024

End-to-end Multi-modal (text, audio and images)

https://openai.com/index/hello-gpt-4o/

Learning from Human Feedback



UNIVERSITY OF WATERLOO

Response Ranking

PP Explain reinforcement learning in three sentences.

‹ 1 / 2 › Reinforcement learning is a type of machine learning where an agent learns to make decisions by receiving feedback in the form of rewards or punishments. The agent's goal is to maximize its cumulative reward over time by exploring different actions and learning which actions lead to the highest rewards. Reinforcement learning has been successfully applied to a variety of tasks, including game playing, robotics, and recommendation systems.

‹ 2 / 2 › Reinforcement learning is a type of machine learning that focuses on training an agent to make decisions based on trial and error. The agent receives feedback in the form of rewards or penalties for each decision it makes. By learning from this feedback, the agent can improve its decision-making abilities over time.

Was this response better or worse? Better Worse Same

Reward Model

Stiennon, Ouyang, Wu, Ziegler, Lowe Voss, Radford, Amodei, Christiano (2020) **Learning to summarize from human feedback**, *NeurIPS*.

- $\boldsymbol{s}$: user prompt
- $\boldsymbol{a}$: system response
- Reward function: $\boldsymbol{r_\theta(s,a)} = real\ number$
- Consider several possible responses $a_1 \succcurlyeq a_2 \succcurlyeq \cdots \succcurlyeq a_k$ ranked by annotator
- Training reward function to be consistent with the ranking:

$$Loss(\theta) = -\frac{1}{\binom{k}{2}} E_{(s,a_i,a_j)\in Dataset} \log \sigma\left(r_\theta(s,a_i) - r_\theta(s,a_j)\right)$$

UNIVERSITY OF WATERLOO

Reinforcement Learning

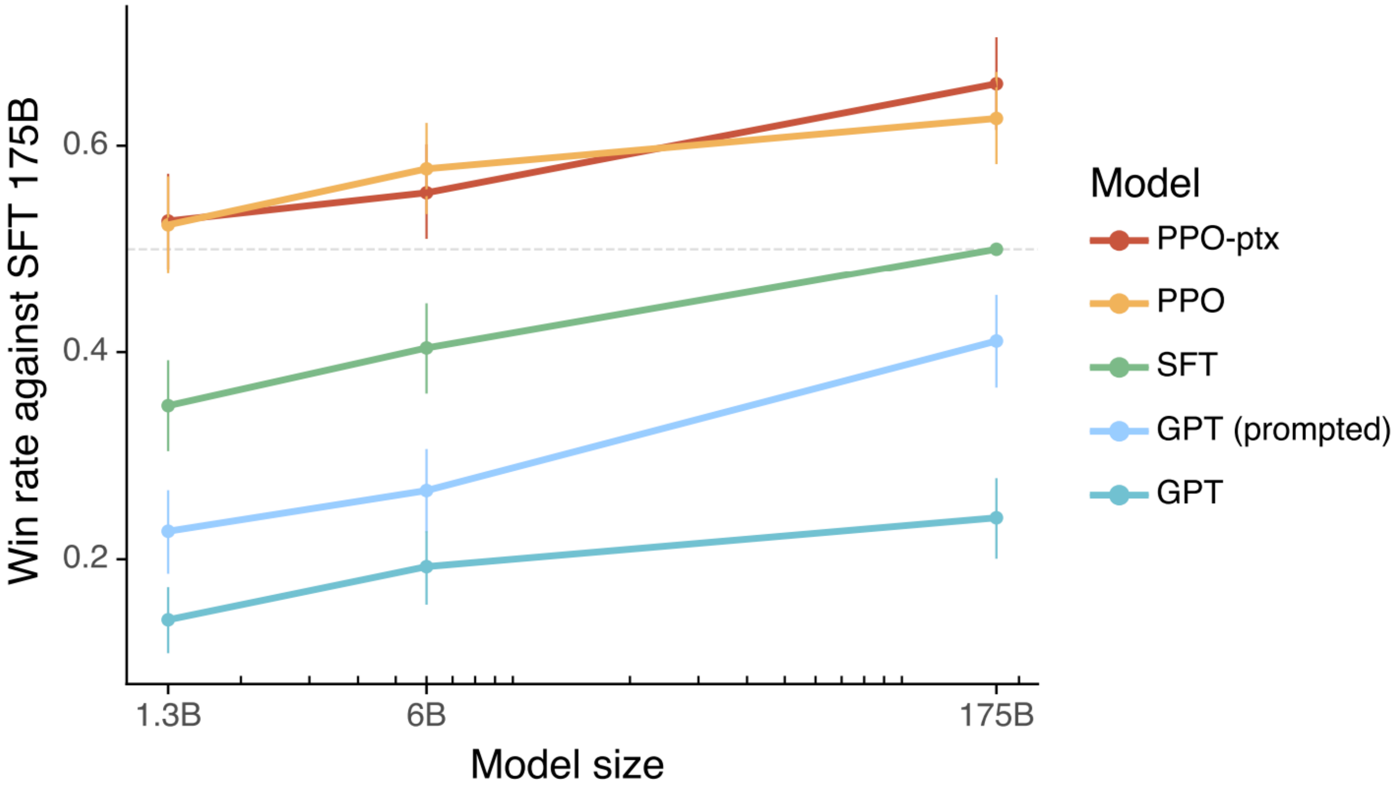
Ouyang, Wu, Jiang, Wainwright, et al. (2022) **Training language models to follow instructions with human feedback**, *NeurIPS*.

- Pretrain language model (GPT-3)
- Fine-Tune GPT-3 by RL to obtain InstructGPT
  - Policy (language model): $\pi_\phi(s) = a$
  - Optimize $\pi_\phi(s)$ by policy gradient (PPO)

$$\max_\phi E_{s \in Dataset}\left[E_{a \sim \pi_\phi(a|s)}[r_\theta(s,a)] - \beta\, KL\big(\pi_\phi(\cdot\,|s)\big|\pi_{ref}(\cdot\,|s)\big)\right]$$

InstructGPT Results

Ouyang, Wu, Jiang, Wainwright, et al. (2022) **Training language models to follow instructions with human feedback**, *NeurIPS*.



UNIVERSITY OF WATERLOO