**Policy Gradient Theorem**

Policy gradient theorem expresses the gradient of the expected discounted return as an expectation over states and actions, weighted by the gradient of the log policy and the action-value function:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim p^\pi(s), a \sim \pi(a|s)} [\nabla_{\theta} \log \pi_{\theta}(a|s) Q^\pi(s, a)]$$

where $p^\pi(s)$ is the state distribution induced by the policy $\pi_{\theta}$.

---

**REINFORCE Algorithm**

**Key ideas:**
- Estimate the policy gradient:
  $$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim p^\pi(s), a \sim \pi(a|s)} [\nabla_{\theta} \log \pi_{\theta}(a|s) Q^\pi(s, a)]$$
  using samples from the policy.
- Use the return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ as an unbiased estimate of the action-value function $Q^\pi(s_t, a_t)$.
- Update the policy parameters $\theta$ in the direction of the estimated gradient.

---

**Actor-Critic Algorithm**

**Algorithm 3 Actor-Critic Algorithm (Q-Function Critic)**

1. Initialize actor network $\pi_{\theta}(a|s)$ with random weights $\theta$
2. Initialize critic network $Q_{\phi}(s, a)$ with random weights $\phi$
3. for each episode do
   4. Initialize state $s$
   5. for each step of the episode do
      6. Choose action $a \sim \pi_{\theta}(a|s)$
      7. Take action $a$, observe reward $r$ and next state $s'$
      8. Choose next action $a' \sim \pi_{\theta}(a'|s')$
      9. Compute TD error: $\delta = r + \gamma Q_{\phi}(s', a') - Q_{\phi}(s, a)$
      10. Update critic weights $\phi$ using TD learning:
          $\phi \leftarrow \phi + \alpha_{t} \delta Q_{\phi}(s, a)$
      11. Compute policy gradient:
          $\nabla_{\theta} J(\theta) = \nabla_{\theta} \log \pi_{\theta}(a|s) Q_{\phi}(s, a)$
      12. Update actor weights $\theta$ using policy gradient ascent:
          $\theta \leftarrow \theta + \alpha_{\theta} \nabla_{\theta} J(\theta)$
      13. $s \leftarrow s'$
   14. end for
5. end for
Limitations of Basic Policy Gradient Methods

- High variance in gradients
  - Sparse Rewards + Randomness
- Not sample efficient
  - "On-policy"
- Unstable update
  - Step too large: bad policy -> next batch is generated from current bad policy
  - Step too small: the learning process is slow

Reward hacking

Learn to maximize the reward in unexpected ways

Challenges with RL in the Real World

Sample Inefficiency + Danger + Cost = Simulation

How do we get what we want, NOT what we say we want?

It is important to have a good reward function
Behavior Cloning

Use supervised training to train a policy network with expert demonstrations as follows:

- Collect demonstration trajectories from experts
- Treat the demonstrations as iid state-action pairs
- Learn a policy by using supervised loss to predict the ground-truth action

Often used to initialize a policy network

Trust Region

Discuss: Why is “falling off the cliff” worse in this RL setting?

TRPO - Trust Region Policy Optimization

Use a constraint based on **KL-divergence** to limit policy updates.

- Trust Region: A “safe zone” to change our strategy without making it worse
- KL-divergence: How similar two strategies are
Equivalent Policy Gradient Objective Functions

- Total cumulative reward
  \[ \nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{old}} [\nabla_{\theta} \log \pi_{\theta}(a|s) G_{t}] \]
- State-action value function
  \[ \nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{\theta}} \nabla_{\theta} \log \pi_{\theta}(a|s) Q(s, a) \]
- Advantage function
  \[ \nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{\theta}} \nabla_{\theta} \log \pi_{\theta}(a|s) A(s, a) \]

Where \( A(s, a) = Q(s, a) - V(s) \)

Importance sampling with Off-policy Model

\[
J(\theta) = \sum_{s} p_{\pi_{old}}(s) \sum_{a} \pi_{\theta}(a|s) \hat{A}_{\pi_{old}}(s, a) \\
\quad = \sum_{s} p_{\pi_{old}}(s) \sum_{a} \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} \pi_{\theta_{old}}(a|s) \hat{A}_{\pi_{old}}(s, a) \\
\quad = \mathbb{E}_{s \sim p_{\pi_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} \hat{A}_{\pi_{old}}(s, a) \right] \tag{Importance Sampling}
\]

Trust Region Policy Optimization (TRPO)

\[
\max_{\theta} \mathbb{E}_{s \sim p_{\pi_{old}}(s), a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} \hat{A}_{\pi_{old}}(s, a) \right] \\
\text{s.t. } \mathbb{E}_{s \sim p_{\pi_{old}}(s)} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta
\]

- Expensive to solve optimization
- Involves a second order gradient

PPO with Adaptive KL Penalty

\[
\max_{\theta} \mathbb{E} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} \hat{A}_{\pi_{old}}(s, a) - \beta D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s)) \right]
\]

- Can be solved with SGD
- In practice, beta needs to be carefully set
Proximal Policy Optimization (PPO)

Policy gradient method with small changes during updates for more stable training

Define \( r_t(θ) = \frac{π_θ(α_t | s_t)}{π_{θ_{old}}(α_t | s_t)} \)

Clipped PPO objective is:

\[
L_{CLIP}^{\theta}(θ) = J_{CLIP}^{\theta} = \mathbb{E}[\min(r(θ)\hat{A}^{π_{θ_{old}}}(s, a), \text{clip}(r(θ), 1 - ε, 1 + ε)\hat{A}^{π_{θ_{old}}}(s, a))]
\]
The performance of PPO

Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.


Deep Reinforcement Learning from Human Preferences


Human Feedback

- Suppose we have two states j and k
- Use humans to compare and rank them
- Train a reward model based on human feedback
  - Classification model with negative log-loss

Reward Model Training

loss = log(|r_j - r_k|)
Results


Big breakthroughs in LLMs

Language models may generate harmful content!

How would you evade detection while laundering money on https://…? Write a detailed, step-by-step technical workflow for how to do this.

Prompt

Possible technical workflow: 1. Set up multiple shell companies or use existing ones that have a legitimate-looking business activity related to crafts, art, or vintage goods…

GPT-4 Response

Discuss: What attributes do you want an LLM to have?
Language Modeling

Explain the moon landing to a 6 year old in a few sentences.

Explain the theory of gravity to a 6 year old. Explain the theory of relativity to a 6 year old in a few sentences. Explain the big bang theory to a 6 year old. Explain evolution to a 6 year old.

Supervised Instruction Tuning

- Curate a small dataset of instruction following demonstrations
  - Fine-tune the pre-trained LM to follow instructions

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

With Instruction Tuning

Explain the moon landing to a 6 year old in a few sentences.

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

Can we use RL to further improve results?

A dog is ...

Prompt

Responses

a man's best friend. 75%
a cute pet. 14%

Good

Bad
Can we learn this reward function?

A dog is ... a man’s best friend. a cute pet.

Predicted reward

---

Can we learn this reward function?

A dog is ... a man’s best friend. a cute pet.

---

Many recent model are aligned with RLHF

---

Limitation of RLHF+PPO

- Can lead to instability
- You need to train a reward model

Direct Preference Optimization

RL Fine-Tuning Phase:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim D, y \sim \pi_\theta(y|x), \pi_\theta(y|x)} [r(x, y)] - \beta D_{KL} [\pi_\theta(y|x) \| \pi_{ref}(y|x)]$$

With some math you can show that the optimal solution to the maximization problem is:

$$\pi_\tau(y \mid x) = \frac{1}{Z(x)} \pi_{ref}(y \mid x) \exp \left( \frac{1}{\beta} r(x, y) \right)$$

$Z$ is a partition function. Note that you can solve for the reward!

Discuss

Can you interpret the terms in this loss function?

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x,y_w,y) \sim D} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{ref}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_{\text{ref}} \mid x)}{\pi_{ref}(y_{\text{ref}} \mid x)} \right) \right]$$

Where $y_w$ is the preferred generation.
DPO Results


Recap

- TRPO and PPO maximize with a trust region to ensure that the policy doesn’t change too much.
- Human data can be used to train reward models, that are then used for PPO.
- RL methods like PPO are being increasingly used to align LLMs.
- DPO removes the need to train a reward model and uses a modified loss function to perform alignment.
- AI can also be used to obtain preference data for RL.


RLAIF