Lecture 20 - Reinforcement Learning from Human Feedback

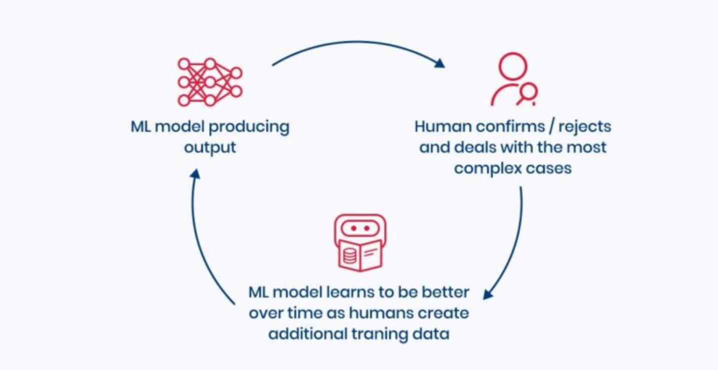
Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

DDA4230: Reinforcement Learning
Course Page: [Click]

Human in-the-loop Machine Learning

Human-in-the-loop machine learning (HITL) is a branch of artificial intelligence that leverages both human and machine intelligence to create machine learning models.



Human in-the-loop Machine Learning

Human-in-the-loop machine learning (HITL) is a branch of artificial intelligence that leverages both human and machine intelligence to create machine learning models.

- It involves a continuous feedback loop where humans train, tune, and test an algorithm, and intervene when the machine is not able to solve a problem.
- HITL improves machine learning over random sampling by selecting the most critical data needed to refine the model.

Human in the Loop Reinforcement Learning

Human-in-the-loop reinforcement learning (HITL-RL) is a form of reinforcement learning where a human interacts directly with the learning process by:

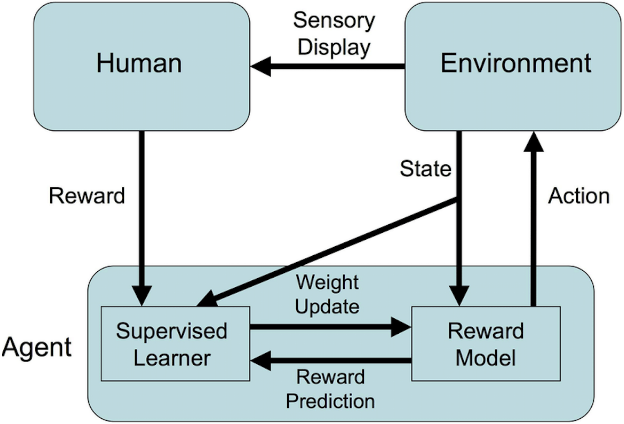
- **Providing feedback**: The human can provide feedback on the agent's actions. This feedback can supplement or replace the rewards in the environment.
- **Setting goals**: The human can define the goals or tasks that the agent should strive to achieve. This can be particularly useful in complex environments where defining a suitable reward function is challenging.
- **Demonstrating /Correcting actions**: The human can show the agent how to perform certain actions or behaviors. If the agent makes a mistake or an incorrect decision, the human can step in and correct it.

Human in the Loop Reinforcement Learning

In this lecture, we are particularly interested in learning reward functions from human feedback, since:

- Reward functions can efficiently represent the signals (including feedback, goals, and preferred actions) listed above.
- In the task where the reward signals are unavailable, the learned reward functions can enable training RL models for efficient control.
- Reward functions summarize human preference, which facilitates the interpretation of reinforcement learning.

Human in the Loop Reinforcement Learning



Knox, W. Bradley, and Peter Stone. "Tamer: Training an agent manually via evaluative reinforcement." 2008 7th IEEE international conference on development and learning. IEEE, 2008.

Human in the Loop Reinforcement Learning

However, asking humans to directly design the rewards functions manually is problematic,

- **Sub-optimal Control Performance.** Manual reward engineering might lead to sub-optimal control performance in finishing the task.
- **Safety Issues.** Using a simple proxy for a complex goal, or getting the complex goal a bit wrong, can lead to undesirable and even dangerous behavior.

Reinforcement Learning from Human Preferences

Instead of directly providing the reward, humans provide the ranking of actions. E.g., in a state s, $a_i > a_j$, meaning that the a_i is better than a_j.

Reinforcement Learning from Human Preferences

The Plackett-Luce ranking model defines the probability of a state-action pair (s, a_i) being the largest among a given set $(s, a_i)_{i=0}^{K-1}$ as:

$$p(a_i > a_j, \forall j \neq i | s) = \frac{exp(r_\theta(s, a_i))}{\sum_{j=0}^{K-1} exp(r_\theta(s, a_j))}$$

Let $\sigma : [K] \to [K]$ denote the output of the human labeler, which is a permutation function representing the ranking of the actions. The distribution of σ follows:

$$p(\sigma | s, a_0, a_1, \ldots, a_{K-1}) = \prod_{k=0}^{K-1} \frac{exp(r_\theta(s, a_{\sigma(k)}))}{\sum_{j=k}^{K-1} exp(r_\theta(s, a_{\sigma(j)}))}$$

Reinforcement Learning from Human Preferences

When $K = 2$, the above representation reduces to the pairwise comparison of the Bradley-Terry-Luce (BTL) model:

$$p(y = l | s, a_0, a_1) = \frac{exp(r_\theta(s, a_l))}{exp(r_\theta(s, a_0)) + exp(r_\theta(s, a_1))}$$

The MLE objective of RLHF can be defined as:

$$\hat{\theta}_{MLE} \in \arg\max_{\theta} \ell_{\mathcal{D}}(\theta),$$

$$\ell_{\mathcal{D}}(\theta) = -\sum_{i=1}^{n} \log\Big(\sum_{y^i = \{0,1\}} \frac{y^i \cdot \exp r_\theta(s^i, a^i_{y^i})}{\exp r_\theta(s^i, a^i_0) + \exp r_\theta(s^i, a^i_1)} \Big)$$

Reinforcement Learning from Human Preferences

Lemma

(Strong convexity of ℓ.) We first show that $\ell_{\mathcal{D}}$ is strongly convex at θ with respect to the semi-norm $\|\cdot\|_{\Sigma_{\mathcal{D}}}$, meaning that there is some constant $\gamma > 0$ such that:

$$\ell_{\mathcal{D}}(\theta + \Delta) - \ell_{\mathcal{D}}(\theta) - \langle \nabla \ell_{\mathcal{D}}(\theta^*), \Delta \rangle \leq \gamma \|\Delta\|_{\Sigma_{\mathcal{D}}}^2$$

for all perturbations $\Delta \in \mathrm{R}^d$ such that $\theta + \Delta \in \Theta$.

Reinforcement Learning from Human Preferences

Lemma

For any $\lambda > 0$*, with probability at least* $1-\delta$*,*

$$\|\hat{\theta}_{MLE} - \theta^*\|_{\Sigma_{\mathcal{D}}+\lambda I} \leq C \cdot \sqrt{\frac{d + \log(1/\delta)}{\gamma^2 n} + \lambda B^2}.$$

Here $\Sigma_{\mathcal{D}} = \frac{1}{n}\sum_{i=1}^n (\phi(s^i, a_1^i) - \phi(s^i, a_0^i))(\phi(s^i, a_1^i) - \phi(s^i, a_0^i))^T$*,*
$\gamma = 1/(2 + \exp(-LB) + \exp(LB))$.

Reinforcement Learning from Human Preferences

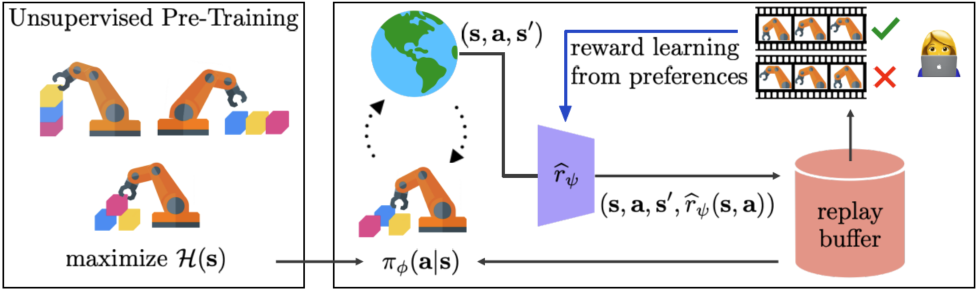
Instead of labeling state-action pairs, one can label the entire trajectory. The MLE objective of RLHF can be updated to:

$$\hat{\theta}_{MLE} \in \arg\max_{\theta} \ell_{\mathcal{D}}(\theta),$$

$$\ell_{\mathcal{D}}(\theta) = -\sum_{i=1}^{n} \log\Big(\sum_{y^i=\{0,1\}} \frac{y^i \cdot \exp \sum_t r_\theta(s_t^i, a_{t,y^i}^i)}{\exp \sum_t r_\theta(s^i, a_{0,t}^i) + \exp \sum_t r_\theta(s_t^i, a_{1,t}^i)} \Big)$$

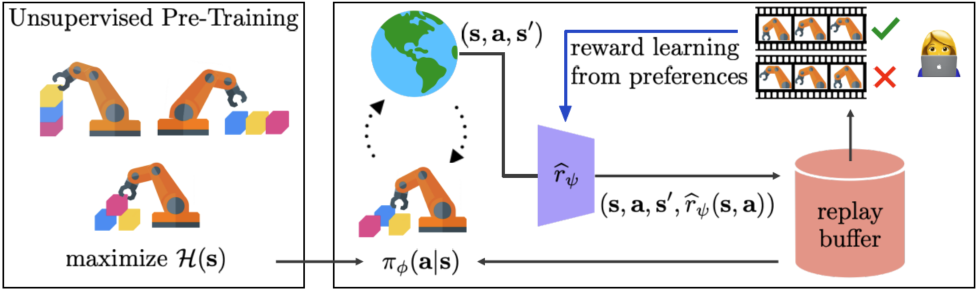
Reinforcement Learning from Human Preferences

Preference-Based Reward Learning. Step 1) The agent engages in unsupervised pre-training during which it is encouraged to visit a diverse set of states to collect diverse experiences (left).



Reinforcement Learning from Human Preferences

Preference-Based Reward Learning. Step 2) A teacher provides preferences between two clips of behavior. A reward model is learned based on them. The agent is updated to maximize the expected return under the model (right).



Train ChatGPT with RLHF

What is GPT?

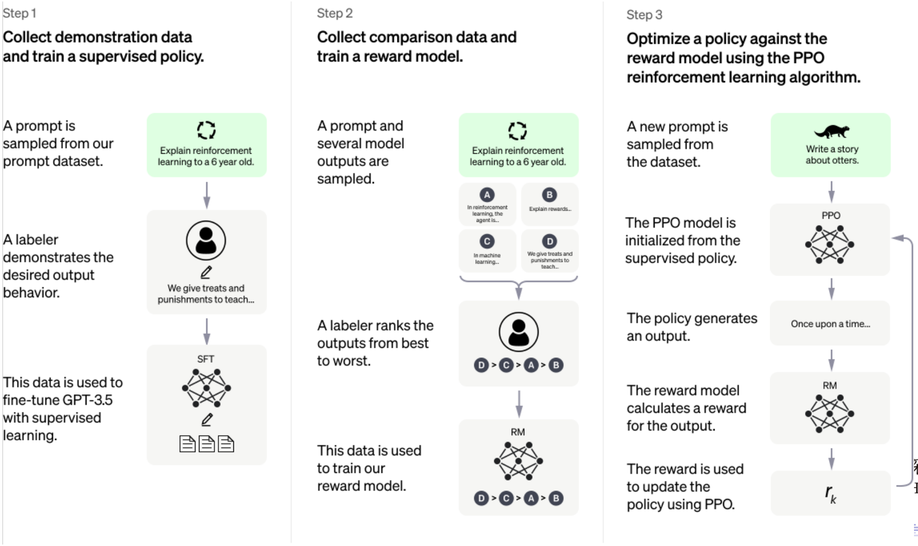
Generative Pre-trained Transformers (GPT), commonly known as GPT, are a family of neural network models that use the transformer architecture and are a key advancement in artificial intelligence (AI) powering generative AI applications such as ChatGPT.

Train ChatGPT with RLHF

Trained a GPT model using Reinforcement Learning from Human Feedback (RLHF):

- Train an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides-the user and an AI assistant.
- Learn a reward model. Collect comparison data, which consisted of two or more model responses ranked by quality by 1) randomly selecting a model-written message, 2) sampling several alternative completions, and 3) having AI trainers rank them. Learn the model based on the collected dataset.
- Fine-tune the GPT model using Proximal Policy Optimization (PPO).

Train ChatGPT with RLHF



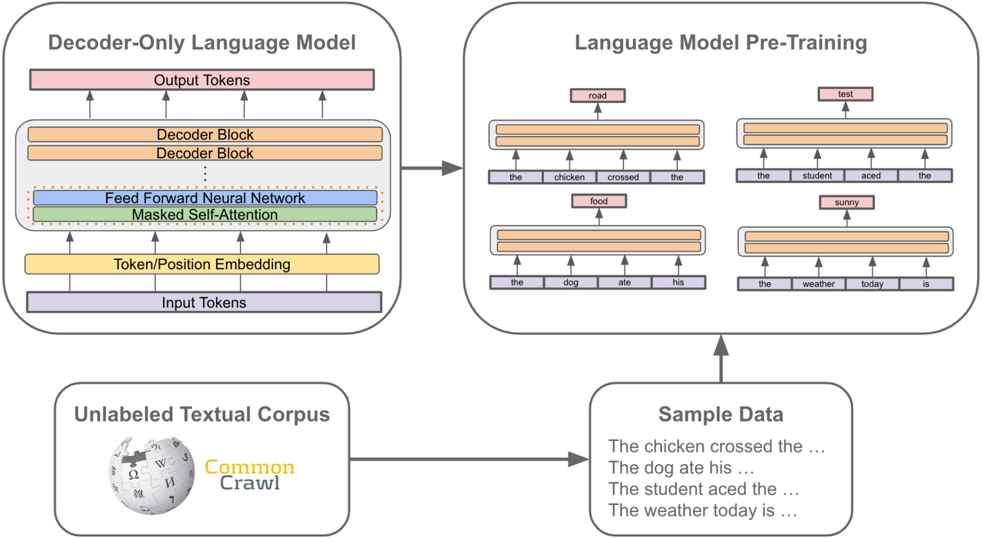
Train a GPT Model using Supervised Fine-Tuning

GPT models are pre-trained over a corpus/dataset of unlabeled textual data using a language modeling objective by

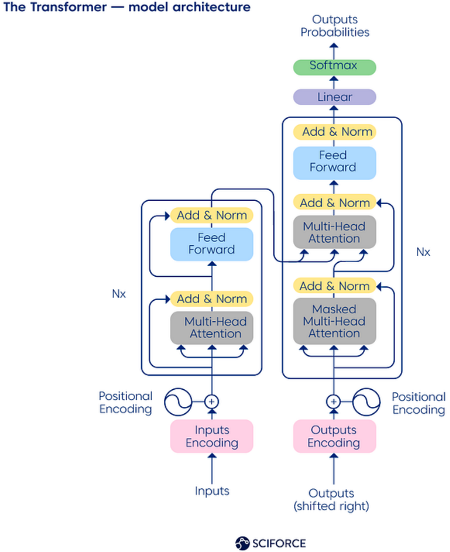
- sampling some text from the dataset.
- training the model to predict the next word.

This pre-training procedure is a form of self-supervised learning, as the correct "next" word can be determined by simply looking at the next word in the dataset.

Train a GPT Model using Supervised Fine-Tuning



Train a GPT Model using Supervised Fine-Tuning



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Train a GPT Model using Supervised Fine-Tuning

Let us denote this set of tokens (of size N) that comprise our pre-training dataset:

$$\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_N\}$$

Given a deep learning model with parameters θ, a language modeling objective tries to maximize the likelihood shown below.

$$\ell_{(}\theta) = \sum_{i=1}^{N} \log(P(\mu_i | \mu_{i-k}, \ldots, \mu_{i-1}; \theta))$$

This loss characterizes the model's probability of predicting the correct next token given k preceding tokens as context

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

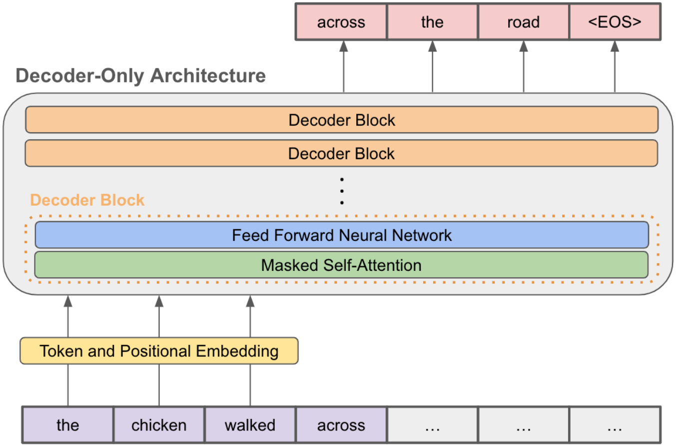
Train a GPT Model using Supervised Fine-Tuning

GPT uses a **decoder-only** transformer architecture. A decoder-only architecture removes the following components from the transformer:

- The entire encoder module;
- All encoder-decoder self-attention modules in the decoder.

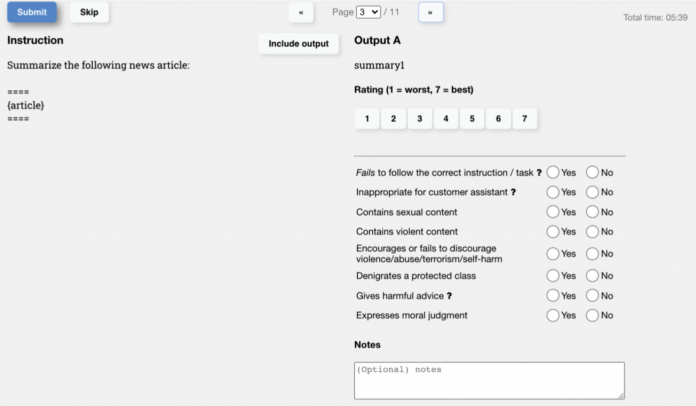
After these components have been removed, each layer of the decoder simply consists of a masked self-attention layer followed by a feed-forward neural network. Stacking several of such layers on top of each other forms a deep, decoder-only transformer architecture.

Train a GPT Model using Supervised Fine-Tuning



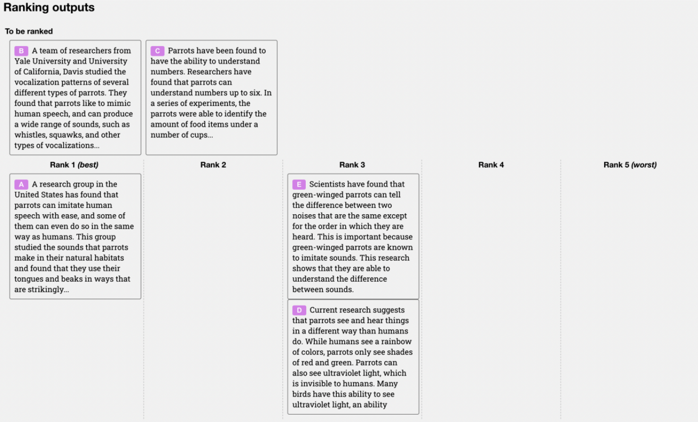
Learn a Reward Model

Collect comparison data. Below is a screenshot of the UI that OpenAI's labelers used to create training data for InstructGPT's RM. Labelers rank the responses in the order of preference, but only the ranking is used to train the RM.



Learn a Reward Model

Collect comparison data. The inter-labeler agreement is around 73%, which means if they ask 10 people to rank 2 responses, 7 of them will have the same ranking.



Learn a Reward Model

Learn the reward model. Given a set of tokens (of size N):

$$\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_N]$$

Let $\boldsymbol{\mu}_q$ denotes the question and $\boldsymbol{\mu}_r$ denotes the response. Let's denote $s_t = [\boldsymbol{\mu}_q, \mu_{r,1}, \mu_{r,2}, ..., \mu_{r,t-1}]$ $(t \leq N)$ and action $a_t = \mu_{r,t}$. Learning the reward model $r_\theta(\cdot)$ (shares the similar structure as the actor) by utilizing the MLE loss:

$$\hat{\theta}_{MLE} \in \arg\max_{\theta} \ell_{\mathcal{D}}(\theta),$$

$$\ell_{\mathcal{D}}(\theta) = -\sum_{i=1}^{n} \log\Big(\sum_{y^i=\{0,1\}} \frac{y^i \cdot \exp \sum_t r_\theta(s_t^i, a_{t,y^i}^i)}{\exp \sum_t r_\theta(s^i, a_{0,t}^i) + \exp \sum_t r_\theta(s_t^i, a_{1,t}^i)} \Big)$$

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Fine-tune the GPT model

Fine-tune the GPT model with PPO:

Algorithm 1 PPO-Clip

1: Input: initial policy parameters θ_0, initial value function parameters ϕ_0

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

3: Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

4: Compute rewards-to-go $\hat{R}_t$.

5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.

6: Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.

7: Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.

8: **end for**

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Limitation of ChatGPT

ChatGPT sometimes writes plausible-sounding but incorrect or nonsensical answers.

Fixing this issue is challenging, as:

- During RL training, there's currently no source of truth;
- Training the model to be more cautious causes it to decline questions that it can answer correctly;
- Supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows.

Reference

- RLHF: Reinforcement Learning from Human Feedback

  https://huyenchip.com/2023/05/02/rlhf.html.

- Introducing ChatGPT. https://openai.com/blog/chatgpt

- Language Models: GPT and GPT-2.

  https://cameronrwolfe.substack.com/p/language-models-gpt-and-gpt-2.

- Zhu, Banghua, Jiantao Jiao, and Michael I. Jordan. "Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons." arXiv preprint arXiv:2301.11270 (2023).

- Lee, Kimin, Laura M. Smith, and Pieter Abbeel. "PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training." International Conference on Machine Learning. PMLR, 2021.

Question and Answering (Q&A)