Reinforcement Learning from Human Feedback

Jason Brown

17/03/2025

Reinforcement Learning

- General paradigm for solving sequential decision making problems
- Leverages a reward function, $R : S \times A \times S \to \mathbb{R}$
- Theoretically applicable to a vast number of domains

Reward Specification

3 Big Problems

1. "True" reward function might be too sparse for learning
2. A shaped reward function might be undesirably exploitable
3. Desired behaviour might be complex, thus difficult to specify

(a) Win The Race

(b) Maximise Score

(c) Be Helpful and Harmless

Can we do better?

1. Try really hard to specify good reward functions
   - Takes lots of time
   - Difficult
   - Unpredictable defects

2. Utilise some other source of data
   - Demonstrations, preference feedback, etc.
   - Learn policy directly or learn reward model?

	Demonstrations	Preferences	...
Policy	Behavioral Cloning	Preference-based RL	...
Reward Model	Inverse RL	RLHF	...

Reward Modelling

- Parameterise reward function and apply supervised learning
- Why not learn policy directly?
  - More data efficient
  - Robust to changes in dynamics, agent, etc.
  - Factorising the problem - we are free to apply our favourite RL algorithm

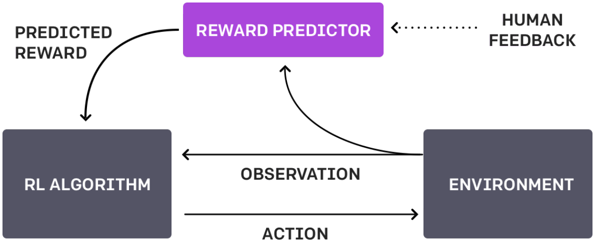

Diagram: Amodei et al. [2017]

Human Preferences

Motivation

- Human can recognise good behaviour
- Preferences encode utility functions

Method

- Human given two examples of agent behaviour
- Human picks favourite
- Reward model should output higher reward for favoured one

Modelling Assumptions

- Humans choose approximately rationally
- Bradley-Terry Preference Model

The Maths

Notation

Learnt reward function	$\hat{R}_\theta$
Parameters	θ
Trajectory	τ
Human dataset	$(\tau_a, \tau_b) \in \mathcal{D}$, where $\tau_a \succ \tau_b$

Equations

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, P(\mathcal{D}|\theta)$$

$$P(\mathcal{D}|\theta) = \prod_{(\tau_a, \tau_b) \in \mathcal{D}} \frac{e^{\hat{R}_\theta(\tau_a)}}{e^{\hat{R}_\theta(\tau_a)} + e^{\hat{R}_\theta(\tau_b)}}$$

$$\mathcal{L} = -\log P(\mathcal{D}|\theta)$$

RLHF Algorithm

The Basics

1. Initialise agent and reward model
2. Optimise agent, producing trajectories
3. Sample pairs of trajectories, get human comparisons
4. Optimise reward model parameters
5. Go to step 2

Additional Tricks

- Sample fragments of trajectories, not whole thing
- Pre-train reward model on prefs over random policy rollouts
- Frontload preferences
- Ensemble of reward models

RLHF with LLMs

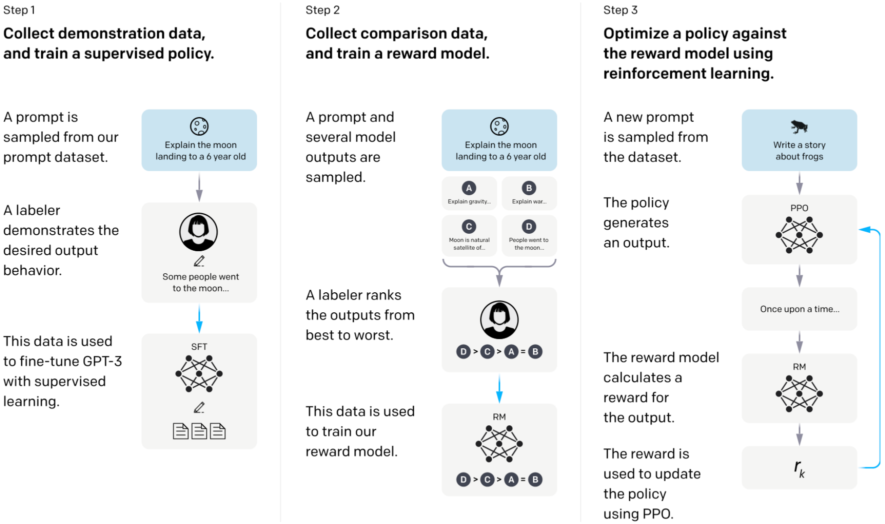

Diagram: Ouyang et al. [2022]

RLHF with LLMs

Additional Details

- Initialise reward model from supervised-finetuned (SFT) model
- Typically use PPO
- KL-Divergence penalty between PPO model and SFT model
- Few RLHF iterations, or even just one
- Quality over quantity
- Many other methods derived from this basic setup...

Issues

- Aligned to who?
- Reward hacking
- Doesn't solve (inner or outer) alignment

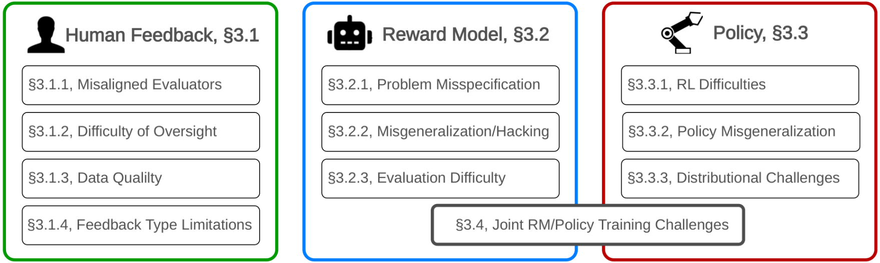

Diagram: Casper et al. [2023]

Further Reading

RL Specification Gaming

DeepMind [2020], OpenAI [2016]

RLHF Basics

Christiano et al. [2017], Thakur [2023]

LLM Finetuning

Ouyang et al. [2022], Stiennon et al. [2020], Ziegler et al. [2019]

Broader Reward Modelling & Imitation

Jeon et al. [2020], Wang et al. [2020]

Issues

Casper et al. [2023], Yudkowsky [2022]

Your Task

Getting Started

1. Download the code: https://github.com/jr-brown/rlhf-workshop
2. Setup environment:
   2.1 Create and activate a new conda/virtual environment
   2.2 Run 'pip install -r requirements.txt'
3. Implement the loss function in 'main.py'
4. Train an agent to balance a pole (exciting!)

Your Task

Have some fun (pick whatever sounds most interesting)

- Explore hyperparameters to minimise required preferences
  - What happens if you increase/decrease train epochs, batch size, or fragment length?
  - Try different network sizes
- Try a harder environment (https://gymnasium.farama.org/)
  - Half Cheetah?
- Swap out the oracle and query the user
- Try different choice models
  - Scale rewards before softmax?
  - Hinge preferences?
- Try and improve algorithm
  - Stop training based on loss covergence instead of fixed number of steps?
  - Select preferences based on uncertainty instead of randomly?

References

Dario Amodei, Paul Christiano, and Alex Ray. Learning from human preferences. `https://openai.com/research/learning-from-human-preferences`, 2017.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

DeepMind. Specification gaming; the flip side of ai ingenuity. `https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity`, 2020.

Hong Jun Jeon, Smitha Milli, and Anca Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. *Advances in Neural Information Processing Systems*, 33:4415–4426, 2020.

OpenAI. Faulty reward functions in the wild. `https://openai.com/blog/faulty-reward-functions/`, 2016.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Ayush Thakur. Understanding reinforcement learning from human feedback. `https://wandb.ai/ayush-thakur/RLHF/reports/Understanding-Reinforcement-Learning-from-Human-Feedback-RLHF-Part-1--VmlldzoyODk5MTIx`, 2023.

Steven Wang, Sam Toyer, Adam Gleave, and Scott Emmons. The `imitation` library for imitation learning and inverse reinforcement learning. `https://github.com/HumanCompatibleAI/imitation`, 2020.

Eliezer Yudkowsky. Agi ruin: A list of lethalities. `https://www.alignmentforum.org/posts/uMQ3cqWDPHhjtiesc/agi-ruin-a-list-of-lethalities`, 2022.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.