Reinforcement Learning

Jordan Boyd-Graber

University of Maryland

Alignment

Is the LLM just “predicting the next word”?

- Not anymore!
- An LM needs to
  - “plan” what it’s going to say
  - Follow instructions
  - Obey rules
- How does that happen?

Is the LLM just “predicting the next word”?

- Not anymore!
- An LM needs to
  - “plan” what it’s going to say
  - Follow instructions
  - Obey rules
- How does that happen? **Reinforcement Learning!**

Is the LLM just “predicting the next word”?

- Not anymore!
- An LM needs to
  - “plan” what it’s going to say
  - Follow instructions
  - Obey rules
- How does that happen? **Reinforcement Learning!**
- Goal for today: RLHF: Reinforcement Learning with Human Feedback

Steve Gorton and Tim Ridley, Alexander Hafemann/Getty Images

From iScoop

Control Learning

Consider learning to choose actions, e.g.,

- Roomba learning to dock on battery charger
- Learning to choose actions to optimize factory output
- Learning to play Backgammon

Control Learning

Consider learning to choose actions, e.g.,

- Roomba learning to dock on battery charger
- Learning to choose actions to optimize factory output
- Learning to play Backgammon
- Generating a sentence

Problem characteristics:

- Delayed reward
- Opportunity for active exploration
- Possibility that state only partially observable
- Possible need to learn multiple tasks with same input/output

Control Learning

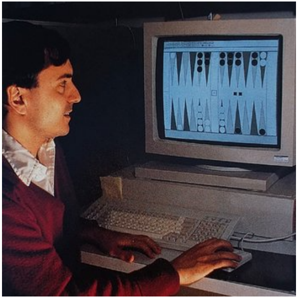
Consider learning to choose actions, e.g.,

- Roomba learning to dock on battery charger
- Learning to choose actions to optimize factory output
- Learning to play Backgammon
- Generating a sentence

Problem characteristics:

- Delayed reward (Only know if a sentence is good once you've generated it)
- Opportunity for active exploration (Hallucination / generalization)
- Possibility that state only partially observable (Who know's what's going on in LLM)
- Possible need to learn multiple tasks with same input/output (Answer questions, write poetry, summarize article)

Early Example: TD-Gammon



Learn to play Backgammon [Tesauro, 1995]

- +100 if win
- -100 if lose
- 0 for all other states

Trained by playing 1.5 million games against itself
Approximately equal to best human player

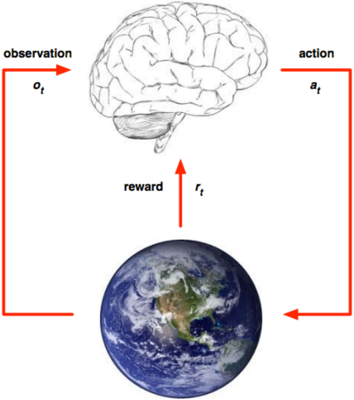
Where RL is Now

- Language Model Alignment
- Machine Translation
- Question answering
- Starcraft
- Go
- Atari
- Robotics

The Problem of Delayed Reward

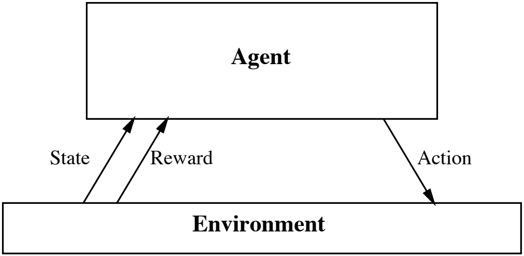
- Mistakes now could have a big cost in the future
- You need to set up an opportunity now for a payoff in the future
- Hard to know which is which

Reinforcement Learning Problem



- At each step t the agent:
  - Executes action a_t
  - Receives observation o_t
  - Receives scalar reward r_t
- The environment:
  - Receives action a_t
  - Emits observation o_{t+1}
  - Emits scalar reward r_{t+1}

Reinforcement Learning Problem



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \ldots$$

Goal: Learn to choose actions that maximize

$$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \text{ , where } 0 \leq \gamma < 1$$

What makes an RL agent?

- Policy: agent's behaviour function
- Value function: how good is each state and/or action
- Model: agent's representation of the environment

Policy

- A policy is the agent's behavior
- State: Agent's internal state, world history (e.g., current masked decoder heads)
  - It is a map from state to action:
  - Deterministic policy: $a = \pi(s)$
  - Stochastic policy: $\pi(a \mid s) = p(a \mid s)$

Likelihood Ratio Policy Gradient

Let τ be state-action $s_0, u_0, \ldots, s_H, u_H$. Utility of policy π parametrized by θ is

$$U(\theta) = \mathbb{E}_{\pi_\theta, U}\left[\sum_t^H R(s_t, u_t); \pi_\theta\right] = \sum_\tau P(\tau; \theta) R(\tau). \tag{1}$$

Our goal is to find θ:

$$\max_\theta U(\theta) = \max_\theta \sum_\tau p(\tau; \theta) R(\tau) \tag{2}$$

Approaches to RL

Value-based RL

- Estimate the optimal value function $Q^\star(s,a)$
- This is the maximum value achievable under any policy

Policy-based RL

- Search directly for the optimal policy $\pi^\star$
- This is the policy achieving maximum future reward

Model-based RL

- Build a model of the environment
- Plan (e.g. by lookahead) using model

Approaches to RL

Value-based RL

- Estimate the optimal value function $Q^\star(s, a)$
- This is the maximum value achievable under any policy

Policy-based RL

- Search directly for the optimal policy $\pi^\star$
- This is the policy achieving maximum future reward

Model-based RL

- Build a model of the environment
- Plan (e.g. by lookahead) using model