What the hell do the numbers we play with represent?

Proximal Policy Optimization (PPO)

16 de octubre de 2025

1. Conceptos Fundamentales del Aprendizaje por Refuerzo

En el aprendizaje por refuerzo (RL), definimos un problema a través de varios componentes clave:

- **Agente:** La entidad que aprende y toma decisiones.
- **Entorno:** El mundo con el que interactúa el agente.
- **Estado (s):** Una descripción de la configuración actual del entorno. El conjunto de todos los estados posibles es $\mathcal{S}$.
- **Acción (a):** Una decisión tomada por el agente. El conjunto de todas las acciones posibles es $\mathcal{A}$.
- **Recompensa (r):** Una señal numérica que indica qué tan buena fue una acción en un estado.
- **Política (π):** La estrategia o çerebro"del agente. Es una función que mapea estados a una distribución de probabilidad sobre acciones, denotada como $\pi(a|s)$. El objetivo es encontrar la política óptima, π^*.
- **Función de Valor ($V(s)$):** Predice el retorno esperado (suma de recompensas futuras con descuento) desde un estado s, siguiendo la política π. Se define como:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right]$$

donde $\gamma \in [0, 1]$ es el factor de descuento.

2. El Ciclo Principal de PPO: Recolectar y Aprender

PPO opera en un ciclo iterativo que alterna entre dos fases principales:

1. **Fase de Recolección de Datos (Rollout):** El agente interactúa con el entorno usando su política actual para recolectar un lote de experiencias.
2. **Fase de Optimización (Aprendizaje):** El agente utiliza los datos recolectados para actualizar y mejorar su política.

2.1. 'Steps' y el Buffer de Experiencia

Un **'step'** es la unidad fundamental de interacción. En cada paso t, el agente realiza la tupla de experiencia: $(s_t, a_t, r_{t+1}, s_{t+1})$. El hiperparámetro `n_steps` define la **cantidad total de 'steps' que se recolectan antes de iniciar la fase de optimización**.

2.2. 'Epochs' y 'Batches': La Fase de Aprendizaje

Una vez que el buffer está lleno, se itera sobre estos datos varias veces.

- **'Epoch' (Época):** Una pasada completa sobre todo el conjunto de datos recolectados en el buffer.
- **'Batch' (Lote):** Un subconjunto de los datos del buffer, utilizado para cada actualización de gradiente.

3. El Corazón de PPO: La Optimización Segura

3.1. Cálculo de la Ventaja ($\hat{A}_t$)

Antes de optimizar, calculamos la **función de ventaja**, $\hat{A}_t$, para cada 'step' t. La ventaja nos dice si una acción fue mejor ($\hat{A}_t > 0$) o peor ($\hat{A}_t < 0$) que el promedio esperado desde ese estado.

3.2. La Función Objetivo Çlipped"

PPO busca maximizar el rendimiento de la política con una restricción para evitar cambios drásticos. Se define la relación de probabilidad entre la nueva política (π_θ) y la vieja ($\pi_{\theta_{old}}$):

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$

El objetivo de PPO, conocido como la **función objetivo subrogada recortada (clipped surrogate objective)**, es:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\text{mín} \left(r_t(\theta)\hat{A}_t, \quad \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

Donde ϵ es un hiperparámetro (usualmente 0.2) que define el tamaño de la "zona de confianza". Esta función es la clave de la estabilidad y el rendimiento de PPO.

4. Ejemplo Concreto: El Problema del Viajante

Para ilustrar el funcionamiento de PPO, consideremos un problema donde un agente debe encontrar el camino más corto que conecte un conjunto de nodos.

- **Nodos:** $\{1, 4, 7, 9\}$
- **Objetivo:** Encontrar la secuencia de visita que minimice la distancia total recorrida.
- **Recompensa:** La recompensa en cada paso es el negativo de la distancia recorrida.

4.1. Fase 1: Recolección de Experiencia (Rollout)

El agente "juegaün episodio, comenzando en el **nodo 1**.

4.1.1. Step 1

- **Estado (s_1):** En el **nodo 1**. Opciones: $\{4, 7, 9\}$.
- **Política ($\pi_{\theta_{old}}$):** Asigna probabilidades casi aleatorias: $P(4) = 0{,}35, P(7) = 0{,}33, P(9) = 0{,}32$.
- **Acción (a_1):** Elige ir al **nodo 4**.
- **Recompensa (r_2):** Si $dist(1, 4) = 3$, entonces $r_2 = -3$.
- **Nuevo Estado (s_2):** En el **nodo 4**. Opciones: $\{7, 9\}$.

Se guarda la tupla $(s_1, a_{\text{nodo 4}}, r_2, s_2)$ en el buffer.

4.1.2. Desarrollo del Episodio

El agente continúa hasta visitar todos los nodos. Supongamos que la trayectoria final es $1 \to 4 \to 9 \to 7$. La recompensa total del episodio (retorno) es la suma de las recompensas negativas por cada salto, por ejemplo: $R = -3(\text{de 1 a 4}) - 5(\text{de 4 a 9}) - 2(\text{de 9 a 7}) = -10$.

4.2. Fase 2: Aprendizaje y Optimización

Con el buffer lleno de experiencias de múltiples episodios, el agente aprende.

1. **Cálculo de la Ventaja:** El algoritmo evalúa cada decisión. La acción de ir del **nodo 4 al 9**, que contribuyó a un resultado de -10, se compara con otras trayectorias. Si ir del **nodo 4 al 7** hubiese llevado a un mejor resultado final (ej. -8), la acción $a_{\text{nodo 9}}$ recibiría una **ventaja negativa**.
2. **Actualización de la Política:** Usando L^{CLIP}, la política se actualiza para favorecer acciones con ventaja positiva. La probabilidad de ir del nodo 4 al 7 aumentaría, mientras que la de ir al 9 disminuiría. Por ejemplo:

   $$\pi_{\text{vieja}}(a|s_2) = \{P(7) = 0{,}51, P(9) = 0{,}49\} \quad \to \quad \pi_{\text{nueva}}(a|s_2) = \{P(7) = 0{,}65, P(9) = 0{,}35\}$$

Este ciclo de **recolección-optimización** se repite, llevando a la política a converger hacia la solución óptima (el camino más corto).