Temas Final RL

Grupos: Hasta 3 estudiantes

Elija uno de los siguientes temas:

A. Sin usar librerías de RL

Las siguiente consigas son las más guiadas, no pueden usar ninguna librería con algoritmos ya programados de RL como los disponibles en Stable Baselines3:

1. Implemente PPO-C, SAC o un algoritmo similar y resuelva uno de estos ambientes:
    - https://gymnasium.farama.org/environments/box2d/
    - https://pypi.org/project/gym-sailing/
    - https://github.com/markub3327/flappy-bird-gymnasium
    - https://ale.farama.org/environments/
    - Otro ambiente aprobado por la cátedra (No se recomienda usar ambientes complejos donde la observación sean píxeles crudos)
2. Implemente Rainbow: Combining Improvements in Deep Reinforcement Learning y recree los experimentos usando un ambiente MinAtar

B. Puede usar librerías de RL ya implementadas (permitido usar SB3, etc.)

3. Generalización en Procgen: elija un experimento y repita el experimento de generalización siguiendo el paper original.
4. Dyna-Q tabular vs Q-learning: implemente Dyna-Q tabular y compárelo con Q-learning.
5. Lea Learning to Drive a Bicycle using Reinforcement Learning y recree la imagen que está en la tapa del Sutton y Barto usando este ambiente de aprendizaje
6. Lea Policy Invariance Under Reward Transformations explíquelo y corra experimentos donde se destaquen los puntos principales de este trabajo.
7. Lea Smooth Exploration for Robotic Reinforcement Learning explíquelo y corra experimentos donde se destaquen los puntos principales de este trabajo.
8. Lea Go-Explore: a New Approach for Hard-Exploration Problems Implemente el algoritmo y pruébelo en un gridworld.
9. Lea Time Limits in Reinforcement Learning y reproduzca experimentos que muestren los efectos de truncation vs termination y el manejo correcto del tiempo límite.
10. Lea Hindsight Experience Replay explique el método y demuéstrelo en Bit-Flipping y en un entorno tipo FetchReach con recompensa escasa

De aquí para abajo puede requerirse desarrollar herramientas ad hoc, modificar o crear ambientes.

11. Lea Recurrent World Models Facilitate Policy Evolution y programe un agente para CarRacing-v3 usando un world model recurrente.
12. Lea Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents y muestre por qué las Sticky Actions son importantes creando un agente que explote el determinismo del ambiente y comparándolo con un agente de referencia como DQN.
13. Lea Deep reinforcement learning from human preferences y entrene el ambiente pendulum con el método.
14. Lea A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots y cree un agente similar usando el dataset de conducción autónoma de UdeSA.
15. Creen un agente de RL que mitigue el efecto de ondas de transito en una ring-road. Por ejemplo en https://highway-env.farama.org/
16. Cree un simulador para: https://github.com/Gabo-Tor/pendulum entrene un agente en el usando domain randomization y luego pruébelo en el hardware real.
17. Modele: https://github.com/Gabo-Tor/pendulum y fitee los parámetros del modelo usando datos reales mediante gradient descent.
18. Entrene en hardware: https://github.com/Gabo-Tor/pendulum y haga un estudio de ablaciones y del efecto de distintos hiperparámetros.