

# EL7021: Tarea 4

Profesor: Javier Ruiz del Solar  
Auxiliar: Francisco Leiva  
Ayudante: Rodrigo Salas

Mayo, 2023

## Requisitos

- Python  $\geq 3.x$
- NumPy
- Matplotlib
- PyTorch
- OpenAI Gym (`pip install gym==0.23.1`)

## Fechas de entrega

Parte I (Avance) : 22 de mayo, hasta las 23:59  
Parte II (Final) : 29 de mayo, hasta las 23:59

## Parte I

### Descripción

Implementará parte de un algoritmo de aprendizaje reforzado basado en modelo: el muestreo de experiencias y el entrenamiento del modelo.

### Instrucciones

#### 1. Parametrización del modelo

1.1 Complete la clase `Model` en el archivo `mbrl.py`.

#### 2. Muestreo de experiencias y entrenamiento

2.1 Complete la clase `Buffer` en el archivo `buffer.py`.

2.2 Complete la función `update_model` en `mbrl.py`.

2.3 Complete la función `select_action` en `mbrl.py` (solo para acciones aleatorias).

2.4 Compruebe que sea posible entrenar al modelo (muestre la evolución de la función de pérdida para `CartPole` o `Pendulum` [solo un entrenamiento independiente]).

## Parte II

### Descripción

Implementará un planificador simple que use el modelo de la parte I, y pondrá a prueba al algoritmo abordando problemas clásicos de control.

### Instrucciones

1. Planificador y selección de acciones
  - 1.1 Complete la clase asociada al planificador en el archivo `mbrl.py`.
  - 1.2 Complete la función `select_action` en `mbrl.py`.
2. Funciones de recompensa
  - 2.1 Programe las funciones de recompensa en el archivo `train_agent.py`.
3. Experimentos
  - 3.1 Entrene en el ambiente `CartPole` usando la configuración del código base, y las siguientes combinaciones de parámetros:

ID Experimento	planning_horizon	nb_trajectories
exp-11	1	200
exp-21	5	200
exp-31	15	200

Reporte sus resultados en un único gráfico de recompensas por episodio promedio (cada gráfico debe tener tres curvas, cada curva debe ser generada considerando tres entrenamientos independientes). Adjunte los archivos `.csv` con los resultados obtenidos, y también el código empleado para cargar, procesar y graficar estos datos.

- 3.2 Entrene en el ambiente `Pendulum` usando la configuración del código base, y las siguientes combinaciones de parámetros:

ID Experimento	planning_horizon	nb_trajectories
exp-11	30	100
exp-21	30	500
exp-31	30	1000

Reporte sus resultados siguiendo la misma metodología que para el ambiente `CartPole`. Nuevamente, adjunte los archivos `.csv` con los resultados obtenidos, y también el código empleado para cargar, procesar y graficar estos datos.

- 3.3 Analice los resultados obtenidos para el ambiente `CartPole` y `Pendulum`. En particular, comente las variaciones en los resultados obtenidos al modificar los parámetros que afectan el proceso de planificación.

## Reglas de formato

Las entregas deben cumplir con los siguientes requerimientos:

- Reporte en formato PDF.
- Incluya un archivo README.txt junto al código, donde indique las versiones de las dependencias que utilizó, y las instrucciones de ejecución de su código.
- Entregas parciales y finales en formato zip (reporte y código en un único archivo).
- Figuras legibles, de preferencia vectorizadas.
- Enumerar las respuestas de la misma forma en que se enumeran las preguntas.
- Respuestas concisas. No es necesario describir gráficos, es suficiente con mostrarlos en el reporte, y responder las preguntas que se realizan textualmente.
- No es necesario crear una portada, no obstante, la primera página debe contar con:
  - Título con formato “Avance Tarea X o Entrega Tarea X”, según corresponda.
  - Código del curso.
  - Nombre del estudiante.