Universitat Oberta de Catalunya

uoc.edu

# M2.883 Aprendizaje por refuerzo

## Práctica:

## *Implementación de un agente para la guía autónoma*

# Contenidos

# 1. Presentación

A lo largo de las tres partes de la asignatura hemos entrado en contacto con diferentes clases de algoritmos de aprendizaje por refuerzo que permiten solucionar problemas de control en una gran variedad de entornos.

Esta práctica, que se va a extender a lo largo de un mes aproximadamente, da la posibilidad de enfrentarse al diseño de un agente para solucionar un caso específico de guía autónoma.

Atacaremos el problema a partir de la exploración del entorno y sus observaciones. Luego, pasaremos a la selección del algoritmo más oportuno para solucionar el entorno en cuestión con las observaciones seleccionadas. Finalmente, pasaremos por el entrenamiento y la prueba del agente, hasta llegar al análisis de su rendimiento.

Para ello, se presentará antes el entorno de referencia, poniendo énfasis en las diferentes observaciones disponibles. Luego, se pasará a la implementación de un agente Deep Q-Network (DQN) que lo solucione. Después de estas dos primeras fases de toma de contacto con el problema, se buscará otra observación del entorno que pueda mejorar el rendimiento del agente DQN anteriormente implementado.

# 2. Competencias

En esta actividad se trabajan las siguientes competencias:

- Capacidad para analizar un problema desde el punto de vista del aprendizaje por refuerzo.
- Capacidad para analizar un problema en el nivel de abstracción adecuado en cada situación y aplicar las habilidades y conocimientos adquiridos para resolverlos.

# 3. Objetivos

Los objetivos concretos de esta actividad son:

- Conocer y profundizar en el desarrollo de un entorno real que se pueda resolver mediante técnicas de aprendizaje por refuerzo.

- Aprender a aplicar y comparar diferentes métodos de aprendizaje por refuerzo para poder seleccionar el más adecuado a un entorno y problemática concretos.
- Saber implementar los diferentes métodos, basados en soluciones tabulares y soluciones aproximadas, para resolver un problema concreto.
- Extraer conclusiones a partir de los resultados obtenidos.

# 4. Entorno

Estamos trabajando sobre el problema de guía autónoma y en particular queremos solucionar el caso de conducción por carretera.

Para ello, se elige **highway-env** como entorno simplificado. El entorno se puede encontrar en el siguiente enlace:
https://github.com/eleurent/highway-env

En particular, nos centramos en el entorno **highway**, donde se representa una carretera con múltiples carriles y con un cierto número de coches transitando.

El objetivo de nuestro agente es de controlar un vehículo de modo que pueda:
- evitar colisiones con los otros vehículos en carretera;
- tener la máxima velocidad posible;
- mantenerse en el carril derecho lo más posible.

La función de recompensa que devuelve el entorno se puede ver representada en la siguiente fórmula:

$$R(s,a) = a\frac{v - v_{\min}}{v_{\max} - v_{\min}} - b\,\text{collision}$$

Donde $a$ y $b$ son dos coeficientes, $v$ es la velocidad del vehículo, y $v_{min}$ y $v_{max}$ su velocidad mínima y máxima, respectivamente.

El entorno se considera superado cuando en el tiempo de observación máximo establecido, se hayan verificado las tres condiciones anteriormente introducidas.

El entorno también proporciona diferentes tipos de observaciones que se pueden ver en la documentación adjunta:
https://highway-env.readthedocs.io/en/latest/observations/index.html#id1

**Ejercicio 1.1 (0.5 puntos)**
Se pide explorar el entorno y representar una ejecución aleatoria.

**Ejercicio 1.2 (0.5 puntos)**
Explicar los posibles espacios de observaciones y de acciones (informe escrito).

Nota: Si se usa Colab, para cargar el entorno se pueden seguir las instrucciones en el siguiente enlace: https://colab.research.google.com/github/eleurent/highway-env/blob/master/scripts/highway_planning.ipynb

Para la correcta ejecución, recordar añadir en la primera línea este comando:

```
!apt-get update
```

# 5. Agente de referencia

En la parte III de la asignatura hemos introducido el agente DQN con *replay buffer* y *target network*, que resulta ser un buen candidato para la solución del problema de conducción en carretera, visto que permite controlar entornos con un número elevado de estados y acciones de forma eficiente.

Se pide resolver los 3 ejercicios siguientes.

**Ejercicio 2.1 (1.5 puntos)**
Seleccionar la observación **kinematics** e implementar un agente DQN para el entorno **highway**. Se valorará implementación propia.

**Ejercicio 2.2 (1 punto)**
Entrenar el agente DQN y buscar los valores de los hiperparámetros que obtengan un alto rendimiento del agente. Para ello, es necesario listar los hiperparámetros bajo estudio y presentar las gráficas de las métricas que describen el aprendizaje.

**Ejercicio 2.3 (0.5 puntos)**
Probar el agente entrenado en el entorno de prueba. Visualizar su comportamiento (a través de gráficas de las métricas más oportunas).

# 6. Propuesta de mejora

En esta parte se pide proponer una solución alternativa al problema de conducción por carretera que pueda ser más eficiente con respecto a lo implementado anteriormente. Para alcanzar este objetivo, se debe usar un tipo de observación diferente de **kinematics**. Se deja la posibilidad de adaptar el agente DQN, implementado anteriormente, al nuevo espacio de observaciones o implementar un nuevo agente, basado en los algoritmos que hemos visto a lo largo de la asignatura.

En particular, se pide solucionar los 3 puntos siguientes.

**Ejercicio 3.1 (2 puntos)**
Implementar el agente identificado para el tipo de observación seleccionada en el entorno **highway**.

Justificar las razones que han llevado a probar este tipo de observación entre las disponibles y porque se ha elegido este tipo de agente. Detallar qué tipos de problemas se espera se puedan solucionar con respecto a la implementación anterior. Se valorará implementación propia.

**Ejercicio 3.2 (2 puntos)**
Entrenar el agente identificado y buscar los valores de los hiperpárametros que obtengan el rendimiento “óptimo” del agente.

**Ejercicio 3.3 (2 puntos)**
Analizar el comportamiento del agente identificado entrenado en el entorno de prueba y compararlo con el agente implementado en el punto 2 (a través de gráficas de las métricas más oportunas).

# 7. Entrega

El entregable será un fichero comprimido en formato ZIP con los siguientes dos documentos:

1. **Informe en formato PDF** de entre 10 y 15 páginas de longitud, aproximadamente;
2. **Código** utilizado, ya sea en ficheros Jupyter notebook (.ipynb) o Python (.py)

Para el **informe** se puede usar la siguiente guía:

- Tamaño de fuente 11 o 12.
- Fuente: Arial o similar.
- Interlineado sencillo.
- Tres apartados definidos según el guión.
    - Especificar el ejercicio correspondiente como subapartado.
    - Por ejemplo, Apartado 2 Agente de referencia, apartado 2.1 Implementación agente de referencia, apartado 2.2 Entrenamiento agente de referencia, apartado 2.3 Prueba agente de referencia.
- Las capturas de pantalla (por ejemplo, las gráficas de rendimiento) o los fragmentos de código (si se consideran relevantes) deben estar pensados para ilustrar y no para ser protagonistas.

El **código fuente** empleado para todas las etapas de la práctica debe estar correctamente comentado para facilitar su comprensión. Pueden emplearse ficheros Python nativos (.py) o basados en Jupyter Notebook (en este caso se debe entregar la versión .ipynb y la exportación en formato .html)