

Tarea 2 - Aprendizaje por Refuerzo

Fecha de entrega: Jueves 14 de diciembre

Para esta tarea, a diferencia de la anterior, utilizaremos Google Colab. En este caso, la tarea consiste en implementar el método Deep Q network (DQN) junto con algunas extensiones y utilizarlo en un ambiente tipo Frogger (<https://es.wikipedia.org/wiki/Frogger>). Esta tarea está adaptada del curso de DeepRL de mi colega Ioannis Karamouzas.

1.- (15 pts) Como se mencionó arriba, su tarea es implementar DQN y entrenarlo para el ambiente indicado "frogger-v5", el cual se caracteriza por:

- Estados: el estado del ambiente consiste de la posición, velocidad y radio del agente, como también de las posiciones, velocidades y tamaños de los vehículos en la calle.
- Observación: el agente no puede acceder directamente al estado completo del ambiente, pero recibe una observación parcial que consiste en mediciones de un radar tipo lidar de 2D, ubicado en el centro del agente. Por defecto, la observación incluye las lecturas de los últimos 2 frames con un campo de visión de 360 grados, entregando en total 60 valores de distancia. La distancia del agente al objetivo también se agrega al vector de observación. Recuerde que el input de la Q network es una observación.
- Acciones: El agente tiene tres acciones de tipo discreto: no hacer nada, moverse hacia arriba y moverse hacia abajo.
- Reward: La recompensa motiva al agente a moverse hacia arriba y cruzar la calle, evadiendo colisiones. El agente recibe un +1 si llega a su objetivo, un -0.5 si choca con un vehículo, y una pequeña retroalimentación que es positiva si se mueve hacia el objetivo y negativa si se aleja de él.
- Condiciones de finalización: Un episodio termina si el agente llega a su objetivo o si choca con un vehículo, o si sobrepasa un límite de tiempo.

Notas:

El trabajo con DeepRL involucra bastantes ajustes a los hiperparámetros. En este caso (DQN), se debe ajustar el tamaño del buffer de replay memory, el tamaño del minibatch, el learning rate, la frecuencia de actualización de la target network, la frecuencia de actualización de la q-network, el método de optimización (para gradient descent), la arquitectura de la red neuronal y la estrategia de reducción del valor de épsilon (exploración).

Que debe subir a Canvas:

- Su código (o link a su Colab)
- El modelo final del entrenamiento (revise las funciones en el colab que ayudan a exportar un archivo checkpoint). Recuerde que Google Colab borra los archivos de su cuenta después de cierto tiempo, por lo que debe descargarlos o exportarlos a su Drive a tiempo.
- Un breve informe donde explique la arquitectura de su red neuronal, junto con los resultados. ¿Cómo obtener los resultados? durante el entrenamiento, cada cierto número de pasos, ejecute por algunas iteraciones (por ejemplo, 10) la política greedy para que el agente explote su conocimiento y pueda recolectar el retorno total (promediado por las 10 iteraciones).

El código en Google Colab está disponible aquí:

https://colab.research.google.com/drive/14bycwCA7tLcYKsyBff8nacTsrV_2DVqn?usp=sharing

Más información sobre el ambiente frogger-v5 se puede encontrar en:

<https://gymnasium.farama.org/environments/atari/frogger/>

No olvide crear su copia local y trabajar sobre ella.