

Evaluación 2 - IA en Robótica.

Las siguientes instrucciones son las necesarias para poder entrenar un agente de RL usando el DuckieGym.

Lo primero que debemos hacer es clonar el repositorio de la url: https://github.com/duckietown/challenge-aido_LF-baseline-RL-sim-pytorch (https://github.com/duckietown/challenge-aido_LF-baseline-RL-sim-pytorch) dentro de la carpeta de Gym-Duckietown.

Una vez clonado y activado el ambiente de conda tenemos que instalar en el ambiente el proyecto pueden utilizar 'pip3 install -e .'

El proyecto consiste en entrenar un ente autónomo utilizando Reinforcement Learning o RL.

Para el proyecto requieran de una GPU para hacerlo de manera mucho mas rápida. Se recomienda una GTX 1060 o superior. Si no poseen una GPU para entrenar pueden hablar con Rodrigo Delgado para ver opciones.

Para poder entrenar deben editar el archivo dentro de la carpeta duckietown-rl/scripts/train_cnn.py

Una vez editado el archivo deben ejecutarlo parándose en la carpeta duckietown-rl. Luego deben ejecutar el comando

`python -m scripts.train_cnn.py --seed 123`. Este comando ejecutará en la terminal el entrenador sin una visualización. Pueden modificar el seed para modificar el entrenamiento.

Para probar lo que entrenaron:

- copiar el archivo `duckietown_rl/ddpg.py` a la carpeta root (deberia ser un `cd ../` si estan parados en duckietown_rl) con el nombre `model.py`.

- copiar los archivo `scripts/pytorch_models/DDPG_XXX_actor.pth` y `scripts/pytorch_models/DDPG_XXX_critic.pth` a la carpeta `models` en el root con los nombres: `model_actor.pth` y `model_critic.pth` respectivamente. Los valores de `XXX` corresponde al seed utilizado para entrenar.

- Asegurarse que todos los wrappers utilizados en `duckietown_rl/wrappers.py` esten en el archivo `wrappers.py` de la carpeta root.

Cómo mejoramos la política (de entrenamiento)?

- Modificar las recompensas del DTRewardWrapper en el archivo `duckietown_rl/wrappers.py`
- Modificar el tamaño de las imágenes, a menor tamaño mas rápido se entrena pero se deben modificar las capas en el archivo `ddpg.py`
- Utilizar escala de grises en vez de rgb.
- Cortar el horizonte (cielo), si lo hacen deben ajustar los parámetros de la red.
- Tunear los hiperparámetros del archivo `duckietown_rl/args.py`.
- Agregar un Wrapper de contraste que permita separar mejor el color de las lineas del negro de la pista.

Hiperparámetros ya utilizados

Algunos de los hiperparámetros ya utilizados para entrenar este tipo de agente que han funcionado con cierto grado de autonomía son los siguientes:

- `start_timesteps`: 10000. Cantidad de pasos de tiempo que se ejecuta la política de manera aleatoria.
- `eval_freq`: 5000. Frecuencia con la que se evalua la política.
- `expl_noise`: 0.2. Ruido de exploración gaussiano.
- `batch_size`: 64. Tamaño del lote para el actor como para el critic.
- `discount`: 0.99. Factor de descuento utilizado en el calculo del valor de `target_Q`.
- `tau`: 0.001. Tasa de actualización de la Red objetivo.
- `policy_noise`: 0.2. Ruido agregado a la politica de destino durante la actualización crítica.
- `noise_clip`: 0.5. Rango para recortar el ruido de la política objetivo.
- `policy_freq`: 2. Frecuencia de las actualizaciones de políticas.
- `env_timesteps`: 500. Frecuencia de las actualizaciones de políticas demoradas.
- `replay_buffer_max_size`: 10000. número máximo de pasos para mantener en el buffer de reproducción.
- `max_timesteps`: 400000. Pasos de tiempo máximo para ejecutar el entorno de entrenamiento.

Recomendación: aumentar el `max_timesteps` a cerca de 1M.

Entregas y evaluación.

Deben entregar todos los códigos relevantes para realizar las pruebas del agente autónomo, si modificaron algun archivo necesario para poder ejecutar el agente deben entregarlo.

Deben crear un video que muestre la ejecución del código en vivo de no mas de 5 minutos con una pequeña explicación de las modificaciones realizadas.

Ademas deben entregar un informe de no mas de 5 páginas donde expliquen que modificaciones le hicieron, que ajustaron y que resultados obtuvieron, incluyendo una pequeña sección con posibles mejoras que hubriesen realizado con mas tiempo.