Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Trabajo Final de Graduación
Proyecto: Método basado en aprendizaje reforzado para el control automático de una planta no lineal.
Estudiante: Oscar Andrés Rojas Fonseca
I Semestre 2024

Firma del asesor

Bitácora de trabajo

Fecha	Actividad	Anotaciones	Horas dedicadas
01/04/2024	**1**. Continuación de revisión del código original para *DQN* [1].	*a*) Identificación de la estructura del código ya facilitada (*Imports*, *Memory*, *DQN Algorithm* y *Training*) y sub etapas (funciones importantes).	4 horas
02/04/2024	**2**. Continuación de revisión del código original para *DQN* [1].	*a*) Estudio de las funciones de optimización (*optimize_model*()) y selección de acciones (*select_action*()).	2 horas
02/04/2024	**3**. Prueba de implementación CUDA en env anaconda Ubuntu.	*a*) Eliminación y replanteo de env conda con énfasis en herramientas *pytorch* y CUDA. *b*) Resultado de la prueba: fallida, apunta a problemas de versiones.	6 horas
03/04/2024	**4**. Reunión de seguimiento con el profesor asesor del proyecto.	*a*) Revisión de avance en el código y errores de forma. *b*) Dado el factor tiempo y el poco avance realizado, se acordó enfocarse directamente en el env *Pendulum* de Gymnasium [2].	2 horas

04/04/2024	**5**. Pruebas con entorno *CartPole* y *Pendulum* para implementación de CUDA.	*a*) Por recomendación del profesor asesor, la implementación se mudó a Ubuntu/Linux para meyor control y referencia. *b*) Instalación y creación de nuevos *environments* mediante la herramienta *micro mamba*.	6 horas
05/04/2024	**6**. Estudio de las implicaciones de variación en el algoritmo *DQN* [1] de dos salidas discretas (*CartPole*) a una salida continua (*Pendulum*).	*a*) Se identificaron los puntos clave a remplazar o modificar para el manejo de variables continuas. *b*) Las primeras pruebas de modificación apuntan a una necesidad de discretización del rango de torque ($[-2.0, 2.0]$) para adaptar el modelo al env *Pendulum*.	4 horas
		Total de horas de trabajo:	24 horas

Discusión

Código referencia a método *DQN* [1]

El código de implementación de control al env *CartPole* de Gymnasium mediante *DQN* presente mucho potencial para su adaptación al env *Pendulum*, en primera instancia y luego para experimentación previa al PAMH. Se encuentra debidamente comentado y su estructura esta clara y sencilla.

Luego de la reunión de seguimiento con el profesor asesor, se acordó saltar directamente a las pruebas con *Pendulum* dados los problemas de tiempo del proyecto. Además, se detallaron puntos importantes al respecto como la diferenciación entre los tipos de *actionspace* que presentan cada uno de los envs en cuestión, donde *CartPole* implementa

$$Discrete(2)$$

donde las dos posibles opciones son el 0 para mover el carro base a la izquierda y 1 para mover el carro base a la derecha.

Por otro lado, *Pendulum* utiliza la estructura

$$Box(-2.0,\ 2.0,\ (1,),\ float32)$$

referente a un solo valor de acción posible de tipo decimal y en el rango permitido de $[-2.0,\ 2.0]$.

Lo anterior apunta a la necesidad de una manipulación superficial de la red neuronal base (capa de salida) y la función $select_action()$, función directamente asociada a la estructura de la red neuronal y puente para el $step()$ del env Gymnasium.

Errores de implementación *CUDA* y *Pytorch*

El código de referencia [1] cuenta con la opción para utilizar CUDA con GPU y optimzar el tiempo de entrenamiento, con valores definidos claramente de 600 episodios con GPU y 50 episodios con CPU, donde ya el segundo demuestra la necesidad del uso de CUDA.

Sin embargo, las pruebas realizadas para su utilización arrojan varios errores referentes, en principio, a las versiones de todos los paquetes implicados; CUDA (11.8, 12.1, 12.4), *Pytorch* (2.1.2, 2.2.2) y el controlador de la GPU en Ubuntu (versión 535).

Todas las pruebas de env realizadas se evidencian con el comando

$$print(torch.cuda.is\_available())$$

donde una respuesta de $False$ demuestra el problema de incongruencia entre versiones y en ocasiones tampoco se determina bien las razones del error. En proceso de resolución.

Referencias

[1] A. Paszke and M. Towers, "Reinforcement learning (dqn) tutorial," *PyTorch*.

[2] T. F. Foundation, "Gymnasium documentation," https://gymnasium.farama.org/environments/classic_control/, 2024.