Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC3675 - Aprendizaje Reforzado
21 de agosto, 2024

Tarea 4: Métodos Aproximados

Javier Campos A. & Pedro Palma V.

El presente informe responde - en orden - las preguntas del enunciado de la tarea 4 del curso, en base a los datos obtenidos al experimentar con los diferentes algoritmos. Los resultados obtenidos y sus gráficos pueden ser reproducidos desde el código disponible en nuestro Github.

a)

En la figura 1 se muestra el desempeño de los algoritmos Q-Learning, Sarsa y Sarsa(λ) en el dominio *MountainCar-v0*. Los hiperparámetros utilizados fueron $\epsilon = 0.0$, $\gamma = 1.0$, $\alpha = 0.5/8$ y $\lambda = 0.5$.

El gráfico esta construido usando el largo promedio de 1000 episodios, sobre 30 experimentos (*runs*), reportado cada 10 episodios.

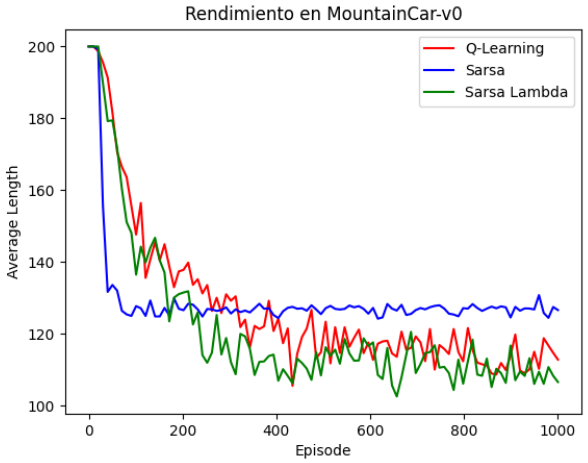

Figura 1: Largo promedio de 1000 episodios en *MountainCar-v0* para los algortimos Q-Learning, Sarsa y Sarsa(λ)

Observamos que el algoritmo Sarsa(λ) (verde) tiene, claramente, el mejor rendimiento, seguido de cerca por Q-Learning. (rojo).

b)

Comparamos el desempeño de un algoritmo Actor-Critic con la implementación de *stable_baselines3* de Soft Actor-Critic, sobre el dominio *MountainCarConinuous-v0*. El gráfico de la figura 2 presenta el largo promedio de 1000 episodios, sobre 30 experimentos (*runs*), muestreado cada 10 episodios.

Hyper-Parameter	Value
learning_rate	0.0003
buffer_size	1 000 000
learning_starts	100
batch_size	256
τ	0.005

Tabla 1: Hiperparámetros default de interés en SAC

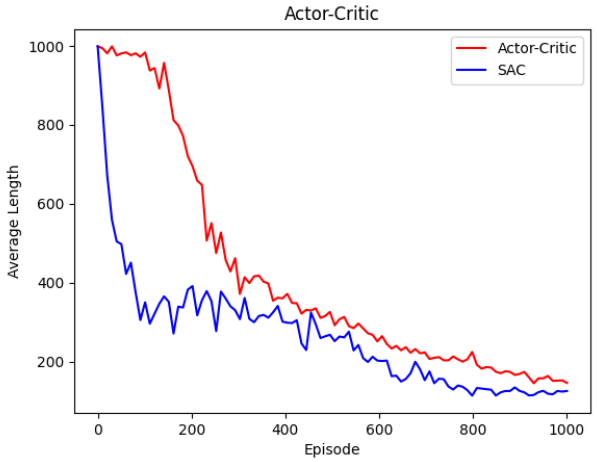

Figura 2: Largo promedio de 1000 episodios en *MountainCarContinuous-v0* para los algortimos Actor-Critic y SAC.

Los hiperparámetros utilizados en Actor-Critic fueron $\gamma = 1.0$, $\alpha_v = 0.001$ y $\alpha_\pi = 0.0001$. Para SAC se definió $\gamma = 1.0$, train_freq $= 32$, use_sde $=$ True (activa la exploración usando generalized state dependent exploration, en lugar de action noise exploration). El resto de hiperparámetros de SAC se dejaron como default, algunos de ellos se muestran en la tabla 1.

Observamos que SAC (azul) es claramente superior, teniendo un descenso rápido en el largo promedio de los episodios al inicio, para luego converger a un largo menor que Actor-Critic (rojo) en el final.

c)

Proponemos 10 nuevas combinaciones de hiperparámetros, descritas a continuación:

- En la figura 3 exploramos la variación del learning rate
- en la figura 4 probamos aumentando learning starts
- en la figura 5 variamos el batch size.
- en la figura 6 variamos tau.

Los resultados de los experimentos fueron filtrados usando un promedio móvil, con una ventana de largo 20 episodios, para poder realmente visualizar las tendencias de las curvas, ya que al ser una

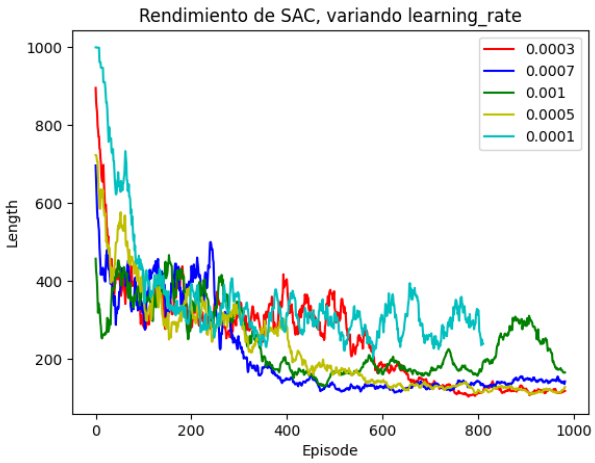

Figura 3: Rendimiento de SAC al variar learning rate.

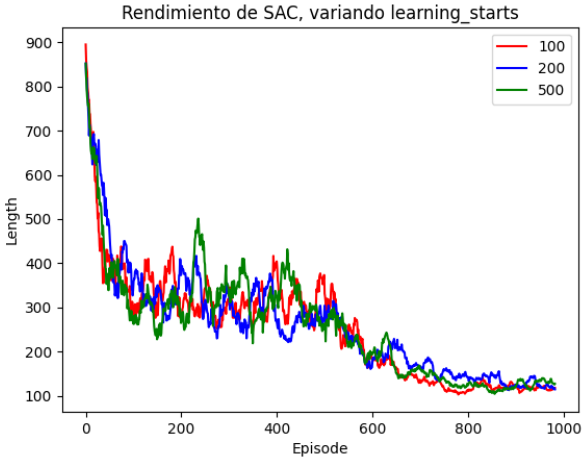

Figura 4: Rendimiento de SAC al variar learning starts.

sola run contienen bastante ruido (el promedio sobre 30 runs lo suavizaba en la sección anterior).

En base a los resultados de estos 10 experimentos, concluimos que subir el learning rate a 0.0005 y subir τ a 0.008 debería mejorar el rendimiento de SAC.
Al comparar los parámetros originales con esta nueva combinación (ver figura 7) podemos observar que, efectivamente, posee un rendimiento igual o superior en todos los tramos del gráfico (línea azul).

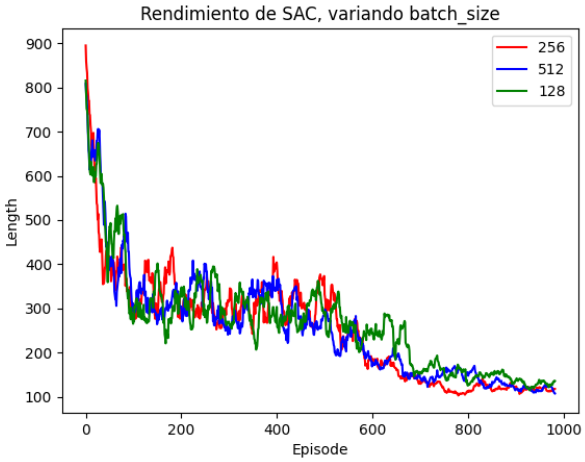

Figura 5: Rendimiento de SAC al variar batch size.

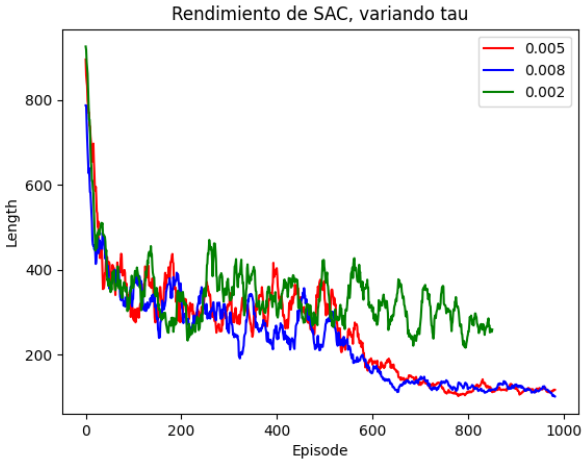

Figura 6: Rendimiento de SAC al variar τ.

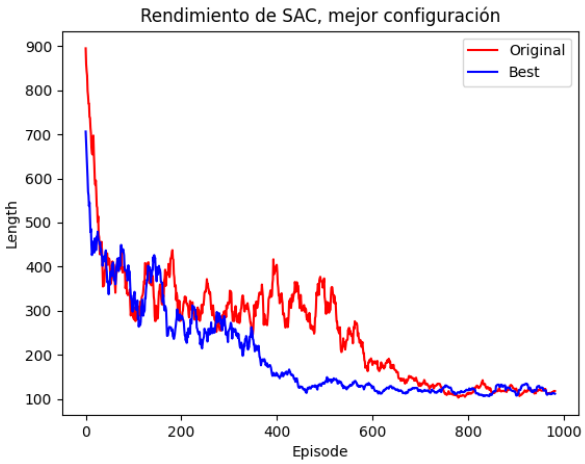

Figura 7: Comparación de los mejores parámetros encontrados vs los parámetros originales en SAC.