CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Ascendiendo a la montaña: aprendizaje por refuerzo colaborativo entre agentes con objetivos opuestos

Violeta Tejera-Munguía[1,*], Juan Jesús Roldán-Gómez[1], Jose Luis Jorro-Aragoneses[1]

[a]*Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, c/ Francisco Tomás y Valiente, nº 11, 28049, Madrid, España.*

To cite this article: Tejera-Munguía, V., Roldán-Gómez, J.J., Jorro-Aragoneses, J.L. 2026. Ascending the mountain: collaborative reinforcement learning between agents with opposing goals.
Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13783

Resumen

Este trabajo explora el concepto de Estupidez Artificial como complemento de la Inteligencia Artificial en el contexto del aprendizaje por refuerzo. Para ello, se realizan una serie de experimentos con el algoritmo Q-Learning en el entorno Mountain Car, comparando un agente inteligente con un agente estúpido que desarrollan comportamientos opuestos. Se proponen dos estrategias de entrenamiento: una reformativa, donde el agente inteligente se entrena a partir del estúpido, y otra colaborativa, donde ambos se entrenan simultáneamente compartiendo información. Los resultados muestran que la incorporación de información procedente del agente estúpido mejora el rendimiento del agente inteligente, acelerando su convergencia y mejorando su rendimiento.

Palabras clave: Inteligencia Artificial, Aprendizaje por refuerzo, Optimización

Ascending the mountain: collaborative reinforcement learning between agents with opposing goals

Abstract

This work explores the concept of Artificial Stupidity as a complement to Artificial Intelligence in the context of reinforcement learning. To this end, a series of experiments are conducted using the Q-Learning algorithm in the Mountain Car environment, comparing an intelligent agent with a stupid agent that exhibits opposite behaviors. Two training strategies are proposed: a reformative one, where the intelligent agent is trained from the stupid agent, and a collaborative one, where both are trained simultaneously by sharing information. The results show that incorporating information from the stupid agent improves the performance of the intelligent agent, accelerating its convergence and enhancing its overall performance.

Keywords: Artificial Intelligence, Reinforcement Learning, Optimization

1. Introducción

En los últimos años hemos sido testigos de un auge sin precedente de la Inteligencia Artificial (IA) como una herramienta clave en diversos ámbitos como la industria, la medicina o la economía. Por ello, resulta imperativo lograr modelos de aprendizaje automático óptimos para resolver los problemas de manera eficiente, evitando comprometer la seguridad de los sistemas en el proceso.

La IA se suele aplicar en forma de problemas de optimización, entrenándose uno o varios modelos para encontrar la solución óptima o una subóptima: por ejemplo, clasificando adecuadamente elementos en su categoría, prediciendo el valor de una variable a partir de unos atributos, realizando detección de objetos en una imagen...

Sin embargo, en algunos contextos han aparecido propuestas de "atontar" estas IAs para que cometan errores deliberadamente. En el año 1991, The Economist acuñó el término *Artificial Stupidity* (Estupidez Artificial, EA) para hacer referencia a este fenómeno (Economist, 1992), aludiendo al diseño

*Autor para correspondencia: violeta.tejera@uam.es

de un agente conversacional que se equivocaba de manera intencional al responder a los evaluadores, con el objetivo de superar el reto basado en el Test de Turing propuesto por el Loebner Prize de dicho año. La idea detrás de esta búsqueda pionera de una Estupidez Artificial era que "errar es humano" y, por tanto, una máquina debía cometer errores para pasar por humana.

En el desarrollo de videojuegos también se sigue esta filosofía a la hora de diseñar los comportamientos de los enemigos controlados por la máquina, buscando que no actúen de una manera óptima y que los jugadores humanos puedan aprovecharlo para vencerlos. En este caso, la EA es necesaria para dar realismo a los videojuegos y plantear a los jugadores un desafío con una dificultad controlada (Lidén, 2003; Climent et al., 2024).

Partiendo de un trabajo previo que exploraba el concepto de la EA y sus posibles aplicaciones en el ámbito de la robótica (Roldán-Gómez, 2022), este trabajo propone la aplicación combinada de la EA y la IA en el contexto del aprendizaje por refuerzo. El objetivo de esta investigación es comprobar si los agentes inteligentes y estúpidos se pueden complementar de manera que se acelere su convergencia hacia las soluciones y obtengan una mayor recompensa acumulada.

El resto del artículo se estructura de la siguiente manera: la Sección 2 formaliza la Estupidez Artificial para poder aplicarla en los problemas, la Sección 3 describe los algoritmos y entornos empleados en el trabajo, la Sección 4 presenta y analiza los resultados de los experimentos y, por último, la Sección 5 resume las conclusiones del trabajo.

2. Estupidez Artificial

El artículo de (Roldán-Gómez, 2022) propone la aplicación de la EA en el ámbito de la robótica, en el cual los modelos de IA suelen ser optimizados en entornos simulados para luego ser desplegados en sistemas reales. Las diferencias entre los entornos de entrenamiento y operación suelen dar lugar a una brecha de realidad (*reality gap*), que dificulta la generalización de los modelos y puede dar lugar a comportamientos indeseados e incluso peligrosos. En ese sentido, la aplicación de agentes estúpidos junto a los inteligentes puede servir para detectar potenciales desviaciones del plan y generar estrategias más seguras.

En dicho artículo se plantean dos formas de EA: una accidental, en la que los comportamientos indeseados son causados por errores, y otra bajo diseño, en la cual estos comportamientos son buscados. En esta última no se busca una solución óptima, el máximo global de la función objetivo del modelo, sino diversas soluciones subóptimas, un conjunto de mínimos locales de ésta.

En ese sentido, entrando en el terreno del aprendizaje por imitación en robótica, (Grollman and Billard, 2011) proponen el uso de demostraciones fallidas como información negativa para alejar al robot de esos comportamientos y, de paso, facilitar el aprendizaje de los correctos. Sin embargo, su enfoque está limitado por la necesidad de producir estas soluciones fallidas de manera manual, en lugar de contar con un modelo estúpido que las genere automáticamente.

Para que una solución estúpida sea considerada válida ha de satisfacer las siguientes condiciones:

- **La solución ha de ser suficientemente mala**: Tomando una definición flexible, cualquier solución que diverja de la óptima podría ser considerada como estúpida, independientemente de si sus resultados son negativos, neutros o incluso positivos. Sin embargo, dado que se pretende utilizar estas soluciones estúpidas para mejorar la seguridad de las inteligentes, conviene tomar una definición más rígida en la que estas llevan a resultados suficientemente negativos. En ese sentido, el desarollador de los experimentos deberá definir el umbral a partir del cual una solución se considera suficientemente pobre.
- **La solución ha de ser suficientemente realista**: Dado que el interés está en localizar desviaciones del plan y generar estrategias más seguras, la búsqueda de soluciones estúpidas se debe centrar en aquellas que son factibles, es decir, aquellas que pueden suceder en el entorno como resultado de un error. Por ejemplo, una colisión con un obstáculo próximo a la ruta óptima será más factible que otra que requiera desviarse una distancia más larga.
- **Todas las soluciones han de ser suficientemente diferentes entre sí**: Cada solución estúpida debe diferir de otras soluciones previamente alcanzadas y que lograrían un estado similar al aplicarse sobre el entorno, para así disponer de un conjunto variado de soluciones que ataquen a diferentes problemas del entorno. Por ejemplo, se prefiere hallar varias desviaciones que llevan a colisionar con diferentes obstáculos a encontrar varias formas de colisionar con el mismo.

3. Metodología

Como primer paso para abordar la pregunta de investigación, en este trabajo se ha utilizado el algoritmo Q-Learning (Watkins and Dayan, 1992; Sutton and Barto, 2018) sobre el entorno de simulación Mountain Car (Moore, 1990). Q-Learning es un algoritmo off-policy basado en valor, que resulta conveniente para este estudio por su carácter tabular, ya que permite explicar el comportamiento de los agentes. Por su parte, Mountain Car es un problema de control clásico en el que el agente controla un coche ubicado en el fondo de un valle sinusoidal y debe alcanzar una meta en lo alto de una colina (ver Figura 1). Este entorno fue seleccionado por su simplicidad y también por la existencia de múltiples trabajos validados en él (Teja Chavali et al., 2022; Bădică et al., 2022).

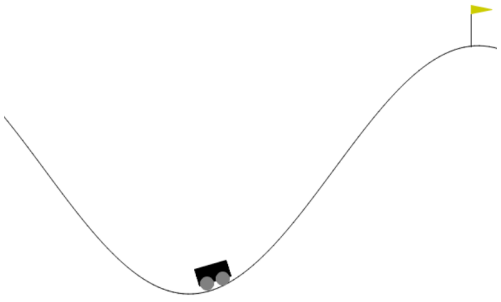


Figura 1: Vista del entorno Mountain Car.

El espacio de estados de Mountain Car es continuo y multivariable, contando con la posición y la velocidad del vehículo. Por lo tanto, para trabajar con un algoritmo tabular como Q-Learning es necesario discretizarlo. En este estudio se han discretizado las dos componentes del estado en 20 intervalos, de manera que el entorno cuenta con un total de 400 estados discretos (ver Figura 2 para el particionado de la posición).

Por su parte, el espacio de acciones de Mountain Car es discreto y cuenta con tres posibilidades: impulsar el vehículo hacia la izquierda, no impulsarlo hacia ningún extremo (de tal manera que el desplazamiento del vehículo se realiza acorde a su inercia) e impulsarlo hacia la derecha.

Considerando como único estado terminal la meta, se ha mantenido la función de recompensas predeterminada del entorno que se muestra en (1).

$$R(s,\ a) = \begin{cases} -1 & \text{si } s \text{ es un ESTADO NO TERMINAL} \\ 0 & \text{si } s \text{ es un ESTADO TERMINAL} \end{cases} \tag{1}$$

La regla de aprendizaje de Q-Learning original (Ecuación (2)) se utilizará para actualizar la matriz $Q_{s,\,a}$ en los agentes inteligentes, mientras que para los agentes estúpidos se ha modificado (Ecuación (3)) para elegir la peor acción en vez de la mejor, sustituyendo el máximo por un mínimo.

$$q_\pi(s,\ a) = q_\pi(s,\ a) + \alpha(R_t + \gamma max_a(q_\pi(S_{t+1},\ a)) - q_\pi(S_t,\ A_t)) \tag{2}$$

$$q_\pi(s,\ a) = q_\pi(s,\ a) + \alpha(R_t + \gamma min_a(q_\pi(S_{t+1},\ a)) - q_\pi(S_t,\ A_t)) \tag{3}$$

Debido a los refuerzos establecidos, un agente inteligente evitará dar pasos que lo desvíen del objetivo, intentando resolver el escenario con el menor número de pasos posibles; mientras que uno estúpido tenderá a dar pasos en falso, como subir la ladera correcta sin inercia o directamente escalar la ladera equivocada. Las consecuencias de estas conductas del agente estúpido pueden aportar información útil para el entrenamiento del agente inteligente. En este trabajo, se ha decidido dotar a cada agente de un máximo de 1.000 pasos por episodio para intentar completar el entorno durante tanto el entrenamiento como la evaluación.

Por último, con objeto de fomentar la exploración, en lugar de comenzar cada episodio de entrenamiento con una velocidad nula, se elegirá una aleatoria dentro del intervalo definido para esta variable.

3.1. Hiperparámetros

Para todos los tipos de agentes, tanto los inteligentes como el estúpido, se ha utilizado una tasa de aprendizaje $\alpha = 0{,}9$, un factor de descuento $\gamma = 0{,}9$ y una política de exploración $\epsilon-$greedy con decaimiento de ϵ de $\frac{2}{\text{número de episodios}}$ desde 1,0 hasta 0,01 en los agentes inteligentes o 0,005 en el estúpido.

Casi todos los agentes inteligentes han contado con 1500 episodios de entrenamiento, mientras que los estúpidos han tenido 500 para el primer caso (entrenamiento reformativo) y 1500 en el segundo (entrenamiento colaborativo), para entrenar un agente estúpido no hacen falta tantos episodios como para uno inteligente, sin embargo, como en el segundo caso se entrenan en simultaneidad, necesitamos que el número de episodios de entrenamiento sea el mismo para sendos agentes.

El único agente inteligente que no ha contado con 1500 episodios de entrenamiento ha sido la muestra de control del primer caso, que ha contado con 2000 en aras de realizar los experimentos de una manera más higiénica, evitando atribuir los resultados obtenidos por el agente inteligente AI_{reform} erróneamente a la influencia de la estupidez artificial si en realidad se hubieran obtenido gracias a realizar más episodios en total (sumando los del AE a los de ese propio modelo).

3.2. Diseño de los experimentos

En el proceso de experimentación se han planteado dos casos diferentes según la relación entre los agentes inteligente y estúpido. Estos casos se describen a continuación:

1. **Entrenamiento reformativo**: La idea detrás de esta estrategia es entrenar un comportamiento inteligente a partir de uno estúpido. Para ello, se entrenará un agente estúpido AE para el problema con el objetivo de utilizar su matriz de valor resultante $Q_{s,\,a}$ como inicial para entrenar un agente inteligente AI_{reform}. Esta prueba va encaminada a estudiar la posibilidad de corregir conductas estúpidas y tornarlas en inteligentes y, en caso de éxito, si hay mejoría en entrenar un agente inteligente a partir de uno estúpido frente a hacerlo desde cero.
2. **Entrenamiento colaborativo**: La idea de esta estrategia es entrenar comportamientos inteligentes y estúpidos de manera simultánea y compartiendo información sobre el entorno. Para ello, se entrenarán un agente estúpido AE y uno inteligente AI_{colab} de manera simultánea y compartiendo la misma matriz $Q_{s,\,a}$. Esta matriz será inicialmente nula y luego será modificada en cada paso por ambos modelos con objetivos opuestos: AI_{colab} maximizará sus acciones deseadas, mientras que AE minimizará las suyas. De esta manera, cada agente sigue su propia trayectoria, generalmente actualizando celdas distintas de la misma matriz $Q_{s,\,a}$. En este caso, el objetivo es analizar si este entrenamiento colaborativo aporta mejoras sobre los entrenamientos individuales, en términos de velocidad de convergencia y calidad de las soluciones.

Durante las pruebas se tomará como referencia un modelo inteligente AI entrenado de manera tradicional, de manera que se pueda comparar los modelos AI_{reform} e AI_{colab} con él. Por último, cada experimento se repetirá 10 veces para obtener sus resultados promedio, y sus modelos resultantes serán sometidos a 100 episodios de validación.

4. Resultados y discusión

En esta sección se van a analizar los resultados de los dos experimentos explicados anteriormente: el entrenamiento reformativo en la Sección 4.1 y el colaborativo en la Sección 4.2. Además, se llevará a cabo una discusión sobre la aplicabilidad de estas técnicas a otros escenarios en la Sección 4.3

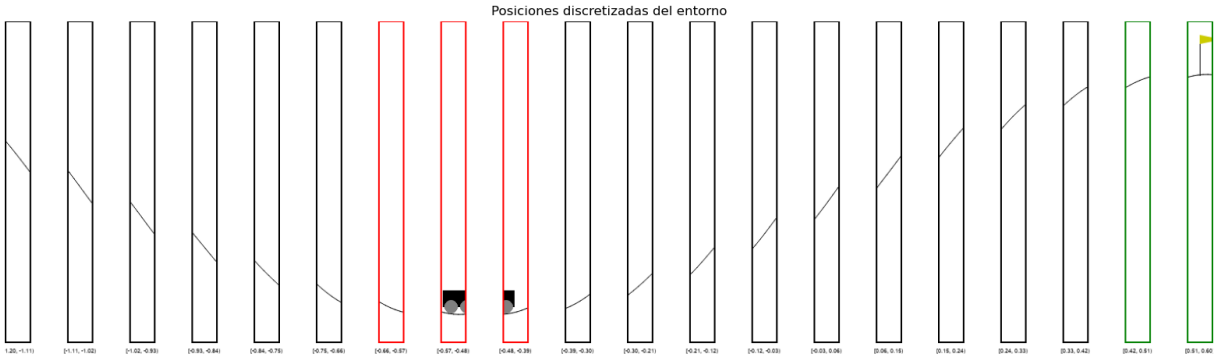


Figura 2: Laminado de la posición del estado en Mountain Car.

4.1. *Entrenamiento reformativo*

La muestra de control *AI* de este caso completó los episodios de prueba en un promedio de 170,3 ± 54,7 pasos, mientras que la AI_{reform} lo hizo en un promedio de 136 ± 29,8, lo que supone una mejoría en el retorno de ∼ 25,2 %. Por lo tanto, se puede afirmar que el modelo inteligente entrenado a partir del estúpido alcanza soluciones más eficientes que el entrenado desde cero.

Sin embargo, la hipótesis de que este entrenamiento reformativo puede converger más rápido que el entrenamiento desde cero no se pudo verificar en este experimento. Por el contrario, los tiempos de convergencia no presentaron diferencias significativas en ningún sentido.

4.2. *Entrenamiento colaborativo*

El entrenamiento colaborativo ha conseguido mejorar con creces los resultados del entrenamiento reformativo. En este caso, la AI_{colab} completó los episodios de prueba en un promedio de 115,7 ± 25,5 pasos. Esto supone una mejora del ∼ 40,6 % sobre su muestra de control *AI* (162,6 ± 28,2) y del ∼ 17,7 % sobre la AI_{reform} del experimento anterior (136,2 ± 25,9).

La Figura 3 contiene los resultados obtenidos por un modelo inteligente de cada tipo, es decir, un modelo de *AI*, uno de AI_{reform} y uno de AI_{colab}. En ellas se puede apreciar la mejora en la convergencia del modelo AI_{colab} con respecto a la muestra de control *AI*, y el otro modelo mejorado mediante técnicas de estupidez artificial, el AI_{reform}. Dado que las gráficas de entrenamiento presentan algo de ruido, se ha considerado como punto de referencia para la convergencia el primer episodio en el que se logra una recompensa superior al 90 % de la máxima obtenida. Así, el modelo de *AI* ha convergido, en promedio, cuando se han completado un 54,4 % ± 3,6 de los episodios de prueba, mientras que el AI_{reform} lo ha hecho cuando se han completado el 18,6 % ± 11,7. Por tanto, el entrenamiento colaborativo ha acelerado la convergencia de los modelos inteligentes en un 301,4 % ± 234,9.

A su vez, la distribución de los pasos por episodio de evaluación en la Figura 3 (a) cuenta con una ligera asimetría positiva según la cual la mitad de los episodios de test han sido completados en 80 pasos o menos; mientras que en la Figura 3 (c) esta distribución está mucho más sesgada hacia la derecha, y la mediana se ubica en algo más de 150 pasos por episodio de test, una cota considerablemente mayor.

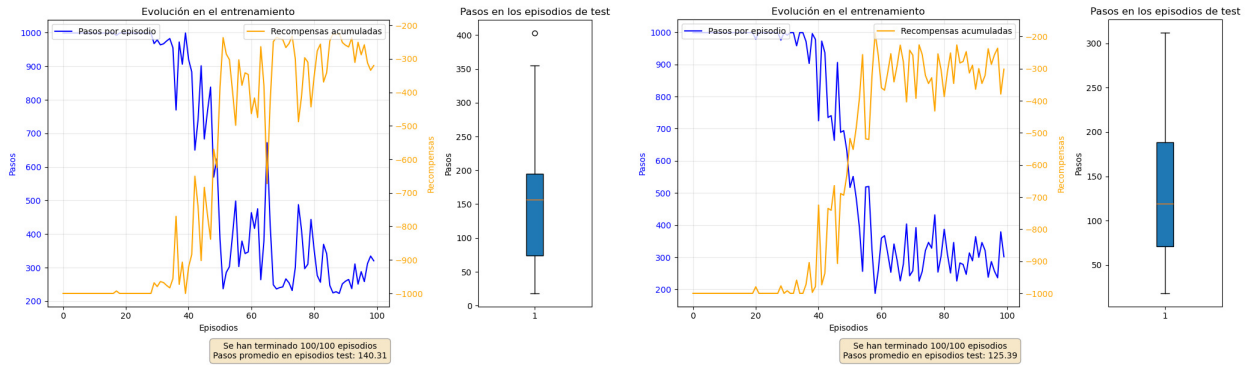
4.3. *Discusión de los resultados*

Si analizamos los resultados positivos de la interacción entre la Inteligencia Artificial (IA) y la Estupidez Artificial (EA) en los escenarios considerados, podemos atribuirlos en parte a que en Mountain Car existe una oposición fuerte entre las acciones inteligentes y estúpidas. De hecho, dado que el agente puede realizar tres acciones en cada estado (impulsarse a la izquierda, no impulsarse o impulsarse a la derecha), podemos intuir que en muchos casos la IA optará por impulsarse en la dirección que facilita llegar a la meta, mientras que la EA lo hará en la dirección contraria. En principio, la opción de no impulsarse sólo será relevante en aquellos estados en los que el vehículo tiene una inercia fuerte y puede cumplir con el objetivo sin necesidad de impulso. Este fenómeno también se muestra en que la IA y la EA pueden operar con el mismo sistema de refuerzos (refuerzo negativo en estados no terminales y nulo en el estado terminal), diferenciándose solamente en la búsqueda del máximo y el mínimo retorno respectivamente. Estas razones permiten que los descubrimientos de la EA añadidos a la matriz de valor sean útiles para la IA y viceversa.

La Figura 4 muestra un mapa de calor de Mountain Car basado en las matrices de valor para *AI* (fila superior) y *AE* (fila inferior). Los estados están agrupados según la laminación de la Figura 2, mostrándose a la derecha la colina donde se encuentra la meta, en el centro el valle donde comienza el agente y a la izquierda la colina contraria. En los ejes Y están los estados ordenados por posición y luego por velocidad, de manera que aquellos con la misma posición y distintas velocidades se muestran contiguos. Por su parte, en los ejes X se muestran las tres acciones viables en cada estado: impulso a izquierda, no impulso e impulso a derecha.

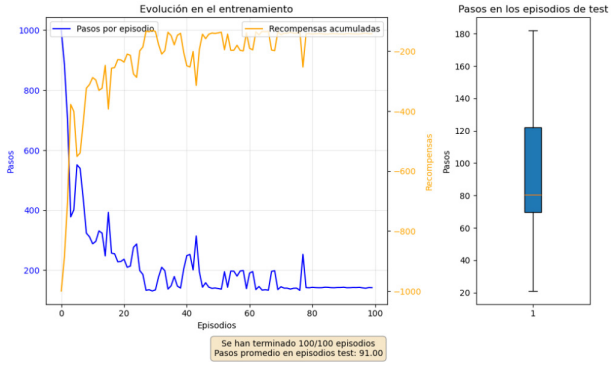
En la fila correspondiente a la inteligencia artificial, las celdas con colores rojizos muestran aquellos estados y acciones que dan menos valor al agente, mientras que las que tienen colores azulados revelan aquellos estados y acciones que le proporcionan un mayor valor. Para facilitar la comparación entre IA y EA, se ha decidido negar la matriz de valor de la de la estupidez artificial. Por lo tanto, en la fila correspondiente a ella, los colores rojizos representan aquellos estados y acciones que dan más valor y, por tanto, deben ser evitados por el agente estúpido, mientras que los azulados representan aquellos estados y acciones que dan menos valor y, en consecuencia, son preferidos por él.

En la Figura 4 se puede apreciar que hay una considera-



(a) Resultados de un AI

(b) Resultados de un AI_{reform}



(c) Resultados de un AI_{colab}

Figura 3: Resultados obtenidos por una instancia de cada uno de los modelos inteligentes desarrollados en este estudio

ble oposición entre inteligencia y estupidez artificiales, ya que los estados azulados en el AI suelen ser rojizos para el AE y viceversa. También se puede observar el hecho de que el AI persigue una solución óptima al problema, mientras que la AE busca un conjunto de soluciones diversas que se oponen a ella. Esto se aprecia en el predominio de las celdas rojizas para la inteligencia artificial y, por el contrario, de las azuladas en la estupidez artificial.

Por último, un hecho interesante de los mapas de calor es que en los estados a la derecha (posiciones mayores o iguales a 18), IA y EA no sólo no muestran oposición, sino que revelan una gran similitud. Esto se debe a que el agente se encuentra en posiciones en las que independientemente de sus acciones es muy probable que alcance la meta.

La generalización de estas técnicas de Mountain Car a otros escenarios puede no ser directa, dado que dichos escenarios pueden tener un mayor abanico de acciones, de manera que la IA y la EA no escojan necesariamente las opuestas. Por ejemplo, en entornos geográficos como GridWorld, Frozen Lake o Cliff Walking, los agentes se pueden desplazar en cuatro direcciones (izquierda, derecha, arriba y abajo) en cada estado (celda del mapa). En este contexto, un agente inteligente se optimizará para dirigirse por el camino más corto hacia la meta, mientras que uno estúpido podría hacerlo para colisionar con el obstáculo más cercano. Dado que hay cuatro direcciones, el camino de la IA hacia la meta no tendría por qué ser el opuesto de un potencial camino de la EA hacia el obstáculo más próximo. Por lo tanto, los descubrimientos de la EA podrían no tener un impacto tan relevante en el aprendizaje de la IA y viceversa. En trabajos futuros se probará la expansión de estas técnicas a dichos escenarios, realizando los cambios necesarios para conseguir que funcionen adecuadamente en ellos.

5. Conclusiones

El objetivo principal de este trabajo era el de investigar el uso potencial de la Estupidez Artificial, un concepto contrario a la Inteligencia Artificial, para mejorarla en el contexto del aprendizaje por refuerzo. En un primer análisis realizado con el algoritmo Q-Learning en el entorno de Mountain Car, se han obtenido grandes mejoras tanto en aceleración de la convergencia como de maximización del retorno del proceso de decisión de Markov. Los resultados han sido especialmente relevantes en el caso de un entrenamiento colaborativo de una IA y una EA que comparten la matriz de valor y la actualizan simultáneamente.

Como trabajo futuro se propone la aplicación de esta técnica en otros entornos en los que las conductas inteligentes y estúpidas presenten una menor oposición, como podrían ser entornos geográficos del estilo de GridWorld, Frozen Lake o Cliff Walking. Además, se probará el paso de algoritmos tabulares como Q-Learning a otros de aprendizaje profundo como Deep Q-Learning, con objeto de evaluar esta técnica en problemas con espacios de estados y acciones continuos.

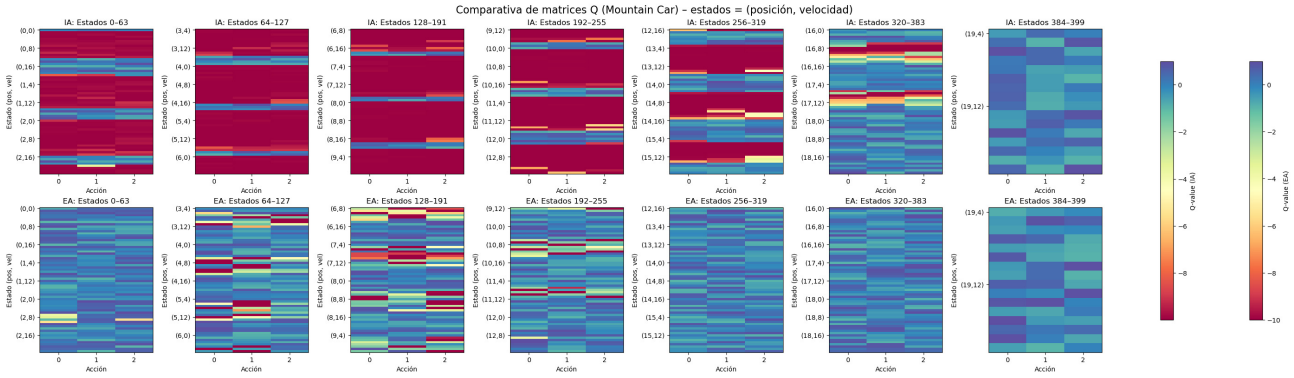


Figura 4: Mapa de calor de Mountain Car para los agentes *AI* y *AE* basado en el valor de sus estados y acciones.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo de los proyectos PID2022-139856NB-I00 financiado por MCIN/ AEI / 10.13039/501100011033 / FEDER, UE; SI4/PJI/2024-00032 financiado por la Comunidad de Madrid mediante el convenio de subvención dirigido a fomentar y promover la investigación y transferencia de tecnología en la Universidad Autónoma de Madrid; y "iRoboCity2030-CM, Robótica inteligente para ciudades sostenibles" (TEC-2024/TEC-62) financiado por el Programa de Actividades I+D en tecnologías de la Comunidad de Madrid.

Referencias

Bădică, A., Bădică, C., Ivanović, M., Logofătu, D., 2022. Experiments with solving mountain car problem using state discretization and q-learning. In: Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28–30, 2022, Proceedings, Part I. Springer-Verlag, Berlin, Heidelberg, p. 142–155.
URL: https://doi.org/10.1007/978-3-031-21743-2_12
DOI: 10.1007/978-3-031-21743-2¡2

Climent, L., Longhi, A., Arbelaez, A., Mancini, M., 2024. A framework for designing reinforcement learning agents with dynamic difficulty adjustment in single-player action video games. Entertainment Computing 50, 100686.
URL: https://www.sciencedirect.com/science/article/pii/S1875952124000545
DOI: https://doi.org/10.1016/j.entcom.2024.100686

Economist, T., 1992. Artificial stupidity. The Economist 324 (7770), 14.

Grollman, D. H., Billard, A., 2011. Donut as i do: Learning from failed demonstrations. In: 2011 IEEE international conference on robotics and automation. IEEE, pp. 3804–3809.

Lidén, L., 2003. Artificial stupidity: the art of intentional mistakes. Vol. 2. Charles River Media.

Moore, A. W., 1990. Efficient memory-based learning for robot control. Tech. rep., University of Cambridge.

Roldán-Gómez, J. J., 2022. Artificial Stupidity in Robotics: Something Unwanted or Somehow Useful? Lecture notes in networks and systems, 26–37.
URL: https://doi.org/10.1007/978-3-031-21062-4_3
DOI: 10.1007/978-3-031-21062-4{3

Sutton, R. S., Barto, A. G., 11 2018. Reinforcement Learning, second edition. MIT Press.

Teja Chavali, S., Tej Kandavalli, C., Sugash, T. M., Amudha, J., 2022. Modelling a reinforcement learning agent for mountain car problem using q – learning with tabular discretization. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). pp. 1–5.
DOI: 10.1109/MysuruCon55714.2022.9972352

Watkins, C. J. C. H., Dayan, P., 5 1992. Q-learning. Vol. 8.
URL: https://doi.org/10.1007/bf00992698
DOI: 10.1007/bf00992698