Universidad Zaragoza
1542

Trabajo Fin de Máster

Aprendizaje adaptativo basado en el mantenimiento de invarianzas organizacionales neuronales

Adaptative learning based on the maintenance of neural organizational invariances

Autor/es

RUBÉN CRUELLAS LABELLA

Director/es

Manuel González Bedia

Miguel Aguilera Lizarraga

Titulación del autor

Ingeniería industrial

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2020

RESUMEN

Durante las últimas décadas se están utilizando herramientas que vienen de la física para estudiar sistemas biológicos. En estos estudios se ha observado que los sistemas biológicos no se posicionan en una fase u otra, sino que suelen posicionarse en la transición de ambas para mejorar su capacidad de adaptación.

En un artículo de la revista Scientific Reports [1] se propone un mecanismo de aprendizaje que genera comportamientos adaptativos, en entornos sencillos, buscando los puntos críticos, que son las transiciones de fase comentadas anteriormente.

Este trabajo consiste en replicar el algoritmo de aprendizaje y extender sus resultados en entornos más complejos. Los entornos, que se explicarán en las siguientes secciones, son: un robot móvil en un entorno uniforme y en otro irregular, un péndulo con la barra fija y un robot de dos piernas que aprende a andar y a levantarse.

En todos los entornos se observa que, simplemente buscando estos puntos críticos, los agentes generan comportamientos interesantes sin haber sido programados para ello. Además, la complejidad de su comportamiento aumenta tal como se aumenta el tamaño de las redes neuronales.

Índice

Índice de imágenes

01. INTRODUCCIÓN

01.1. Motivación del proyecto

En los últimos años, se han realizado una gran cantidad de estudios para comprender cómo se adaptan los sistemas cognitivos y biológicos a sus entornos. La realidad es que el conocimiento que tenemos sobre este campo es muy limitado y hay una gran variedad de preguntas sin resolver.

Durante las últimas décadas, una alternativa para abordar algunas de las limitaciones de los métodos clásicos en biología para tratar con sistemas de alta dimensionalidad ha sido recurrir a métodos utilizados en termodinámica y física estadística. Uno de los descubrimientos es que un gran número de estos sistemas se posicionan en un punto crítico. Es decir, no se desarrollan por completo en una fase o en otra, si no que los sistemas permanecen en un equilibrio entre varias fases manteniendo simultáneamente propiedades de ambas. Esto les proporciona la capacidad de procesar información de forma más eficiente [5] y de aumentar la sensibilidad produciendo grandes cambios con pequeñas variaciones.

En la actualidad, no se tiene la certeza de cómo surge el posicionamiento en el punto crítico. Para intentar dar respuesta a esta pregunta, en el artículo tomado como base [1] se diseña una regla de aprendizaje para tratar de posicionar varios sistemas en una transición de fase. El resultado que se espera obtener es que, tras lograr posicionar al agente en un punto crítico, este sea capaz de adaptarse a su entorno, generando comportamientos espontáneos y complejos, sin habérselos definido de forma explícita.

01.2. Fases del proyecto

Este proyecto se basa en un estudio realizado y publicado en la revista Scientific Reports [1]. En dicho estudio, se genera un algoritmo adaptando las correlaciones entre neuronas para tratar de posicionar al agente en un punto crítico. Tras el entrenamiento de la red, se analiza el funcionamiento del agente y se observa si la red ha sido capaz de hacer que el agente esté en un punto crítico, y si este ha sido capaz de adaptarse a su entorno generando comportamientos espontáneos.

En el proyecto tomado como modelo, se han realizado pruebas de la red neuronal controlando varios agentes en dos entornos: el Acrobot y el Mountain Car. Durante el proyecto se realizarán pruebas en otros entornos. Todos los entornos analizados tanto en el artículo de referencia, como en este proyecto se han obtenido de la web de código abierto OpenAI [2].
La web OpenAI se trata de un entorno de benchmarking de IA en el que han participado, entre otros, Microsoft y Elon Musk. Su objetivo es promover y desarrollar la inteligencia artificial. Para ello han creado diferentes entornos sobre los que se pueden probar algoritmos de inteligencia artificial de forma gratuita.

El primer entorno sobre el que se realizarán pruebas será una modificación del Mountain Car. Las modificaciones sobre el entorno, y el análisis de los resultados obtenidos, se realizan para comprender en profundidad las características de la red neuronal.

A continuación, se probará la red neuronal en otro entorno, de una dificultad similar al Mountain Car. En este caso se trata de un péndulo. El objetivo de estas pruebas, además de observar el funcionamiento de la red neuronal, es familiarizarse con el modelo para adquirir los conocimientos necesarios para probarlo en entornos más complejos.

Cuando se consiguen los resultados esperados sobre los entornos comentados, se procede a realizar las pruebas sobre un entorno más complejo. Este entorno se denomina "Walker". Se realizarán modificaciones para que la red sea capaz de hacer que el agente consiga su objetivo y tras familiarizarse con el funcionamiento del modelo se intentará realizar lo siguiente:

- Realizar las modificaciones necesarias, tanto en el robot, como en la red neuronal, para que este sea capaz de avanzar de forma estable y con una velocidad razonable
- Modificar el robot para que se encuentre "tumbado", y posteriormente analizar si la red es capaz de hacer que este se levante manteniendo una cierta estabilidad.

Tras realizar todo el proceso descrito anteriormente, se realizarán análisis para evaluar si la red neuronal consigue que los agentes se adapten a sus entornos, ya sean simples o más complejos. También se sabrá si esta red es capaz de hacer que los agentes cumplan sus objetivos. En este punto se analizará si la red es capaz de llevar a los agentes a un punto crítico.

Para ello, en el artículo tomado como base para el proyecto, se calcula la capacidad calorífica del sistema. Una divergencia de la capacidad calorífica del sistema, tal como se incremente el tamaño de la red, es condición suficiente que indica que el sistema se encuentra en un punto crítico. La capacidad calorífica se calcula a través de la entropía, lo cual tiene un coste computacional muy elevado.

En el proyecto se propondrá un método alternativo para el cálculo de la capacidad calorífica. En lugar de calcularla a partir de la entropía, se hará a partir de la energía del sistema. Se esperan obtener los mismos resultados que con el cálculo a partir de la entropía, pero con la ventaja de que tiene un coste computacional mejor.

02. CONSTRUCCIÓN DEL MODELO

El modelo se genera a partir del modelo de Ising bidimensional. La principal característica de este modelo es que, con unas condiciones determinadas, presenta una transición de fase que tiene solución analítica. Las particularidades del modelo de Ising se explican en el Anexo I.

02.1. Generar correlaciones

En el artículo de referencia [1], el primer paso consiste en obtener las correlaciones de un modelo de Ising que se encuentra en un punto crítico y estas correlaciones se utilizarán posteriormente para el aprendizaje de la red. El objetivo es reproducir las correlaciones de un sistema que esté en un punto crítico para hacer que otro sistema diferente se posicione en un punto crítico similar. Esto se conoce como la propiedad de la universalidad, ya que en criticalidad muchos sistemas pertenecen a la misma familia de puntos críticos y tienen propiedades similares.

Se selecciona un modelo de Ising bidimensional 20x20. Sobre dicho modelo se asigna a la matriz J el valor $\log(1+\sqrt{2})/2\beta$ (Ver Anexo I). Debido a que las conexiones entre neuronas son simétricas, la matriz J será simétrica. Por lo tanto, únicamente se considerarán los elementos que estén por encima de la diagonal.

El siguiente paso consiste en simular el funcionamiento del modelo actualizando el estado de cada elemento con la dinámica de Glauber. Esta es una técnica para generar una dinámica del sistema de acuerdo con la distribución de energía del sistema:

$$P\big(s_i(t+1)\big) = \left[1 + e^{-\beta 2 H_i s_i(t+1)}\right]^{-1}$$

Dónde:

- $H_i = h_i + \sum_j J_{ij} s_i(t)$

El estado del modelo se actualizará aplicando la dinámica de Glauber en cada paso de la simulación, a todos los elementos, en un orden aleatorio. La energía que necesita una neurona para cambiar su signo es $2H_i s_i(t+1)$. Tras realizar la simulación, el modelo de Ising con estas características alcanzará el equilibrio en un estado de máxima entropía:

$$P(s) = \frac{1}{Z} exp[\beta(\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j)]$$

La distribución sigue una exponencial $P(s) = \frac{1}{Z} e^{-\beta E(s)}$, dónde Z es el valor de normalización. En los casos de redes bidimensionales, es conocido que las distribuciones de correlaciones siguen la función asintótica $c(r) \propto 1/r^n$, dónde $\eta = 1/4^2$ y r es la distancia entre unidades.

Debido a que el modelo está lejos del límite termodinámico, en lugar de utilizar la función descrita anteriormente, se calculan las correlaciones sobre el modelo descrito, ya que se encuentra en un punto crítico.

Durante la simulación de la red se han calculado tanto las correlaciones, c_{ij}, como la media, m_i. Debido a que los campos "h" de todas las unidades son igual a cero, la media de todos los elementos también será igual a 0. La distribución de las correlaciones obtenida es la siguiente:

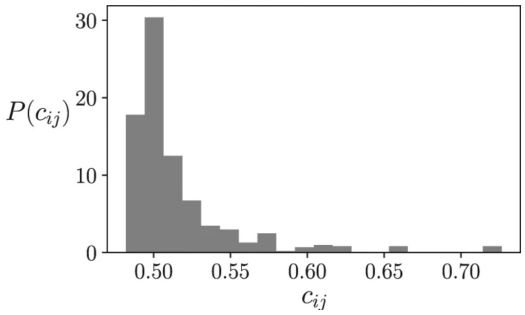

Imagen 1. Distribución de correlaciones obtenidas en el modelo teórico. La imagen se ha obtenido del artículo de referencia [1].

02.2. Entrenar los agentes

Una vez se obtiene la distribución de correlaciones en un modelo teórico posicionado en un punto crítico, se procede a entrenar el agente. A cada neurona se le asigna un valor de c_{ij} aleatoriamente, siguiendo la distribución obtenida anteriormente.

En cada paso de la simulación, se calculará la correlación real de cada neurona con sus neuronas vecinas. Además, se generará un valor de referencia, ordenando las correlaciones de referencia, para que encajen con el orden de las correlaciones reales. Esto se realiza para que el modelo sea más flexible. Esta acción no afecta negativamente al modelo, ya que el orden de las correlaciones no es importante, sino que lo importante es su distribución. Los valores de la media de referencia serán iguales a cero.

A continuación, se tratará de obtener h_i y J_{ij} a partir de las correlaciones obtenidas. Este proceso es conocido como el modelo de Ising inverso. Dicho modelo se resuelve con una regla de descenso del gradiente. Esta, junto con la distribución de correlaciones, es la regla de aprendizaje del algoritmo:

$$h_i \rightarrow h_i + \mu\,(m_i^* - m_i^m)$$

$$J_{ij} \rightarrow J_{ij} + \mu\,(c_{ij}^* - c_{ij}^m)$$

Dónde:

- μ: Tasa de aprendizaje constante. El valor se toma como 0.01.
- m_i^*: Valor de la media de referencia.
- m_i^m: Valor de la media real.
- c_{ij}^*: Correlación de referencia.
- c_{ij}^m: Correlación real.

Para una simulación de un tiempo "T" el cálculo de la media se realiza con la expresión $m_i = \frac{\sum_t^T s_i}{T}$ y el cálculo de las correlaciones se realiza con la expresión: $c_{ij} = \frac{\sum_t^T (s_i * s_j)}{T}$.

Dado que cada paso de la simulación es muy costoso computacionalmente debido a que hay que calcular todos los estados de "s", se toma un método alternativo. Se calculan las correlaciones aproximadas con la ecuación de la dinámica de Glauber.

Tras realizar esta simulación, se obtienen los valores asociados a cada neurona de la red neuronal del modelo. Los siguientes apartados del proyecto se centrarán en validar si esta red es capaz de llevar a los agentes a puntos críticos, y de observar si la red es capaz de hacer que los agentes se adapten a los entornos, consiguiendo los distintos objetivos.

02.3. Calculo de la capacidad calorífica

Una condición suficiente que indique que el sistema se encuentra cerca de un punto crítico es una divergencia en la capacidad calorífica, C. Además, esta magnitud es un indicador de la complejidad de la red neuronal.

La capacidad calorífica del sistema se define como:

$$C(\beta) = -\beta \frac{\partial H}{\partial \beta} = \beta^2\left(E^2(s)\right) - (E(s))^2)$$

Dónde:

- $E(s) = -\sum_i h_i s_i - \sum_{i<j} J_{ij} s_i s_j$

Debido a la definición de la capacidad calorífica, si se detecta que la entropía presenta un pico continuo, en el que su derivada (capacidad calorífica) diverja tal como aumente el tamaño del sistema, se garantizará que el sistema está en un punto crítico.

En el modelo de Ising, la entropía del sistema es: $H = -\sum_s P(s) log(P(s))$. Por lo tanto, se calculará en diferentes puntos, con diferentes tamaños, para concluir si el sistema se encuentra en un punto crítico.

La Entropía se calcula para diferentes valores de β, y para diferentes tamaños un número determinado de iteraciones. Este método de cálculo requiere un elevado coste computacional, por lo que, en siguientes apartados, se propondrán alternativas, con menor coste computacional, para garantizar que el sistema se encuentra en un punto crítico.

02.4. Conclusiones esperadas

Los resultados que se esperan de esta red neuronal, es que esta consiga llevar los agentes a un punto crítico, y de esta forma, conseguir maximizar las fluctuaciones en la energía del sistema. A la red neuronal no se le definirán objetivos, pero se espera que pueda generar nuevas soluciones y comportamientos espontáneos en los agentes, consiguiendo que se adapten a los entornos.

A pesar de ello, no se espera que la red consiga hacer que cualquier agente consiga sus objetivos en cualquier entorno. Por ejemplo, un entorno en el que el objetivo sea mantener el equilibrio (mantener una persona de pie, mantener una barra en vertical moviendo horizontalmente su base…) probablemente no sea adecuado para esta red.

Se han seleccionado los entornos para que el objetivo sea una situación con variaciones de energía. Por ejemplo, en el caso del "Mountain Car (Ver apartado 03.1)", si el coche sube la montaña de forma repetida, generará fluctuaciones de energía.

Durante las simulaciones se visualizarán, por un lado, el comportamiento del agente en el entorno, y por otro su capacidad calorífica.

Visualizando el comportamiento del agente, se podrá observar si este consigue generar comportamientos que puedan parecer complejos, pero estas apreciaciones no serán suficientes para

garantizar que el agente se encuentra en un punto crítico. Es decir, el coche puede subir la montaña, sin estar en un punto crítico.

Para solucionar esto, se calcula la capacidad calorífica del sistema en cada iteración tal como se ha explicado en el apartado anterior.

Con esta información, se podrá asegurar si el agente se encuentra en un punto crítico, y si este punto crítico permite que la red neuronal sea capaz de conseguir que los agentes generen comportamientos complejos, sin definírselos explícitamente.

01.1. Conclusiones del artículo

El primer entorno de estudio ya había sido analizado en el proyecto tomado como base del trabajo fin de máster. Dicho entorno consiste en el denominado "Mountain Car".

El entorno "Mountain Car" (Ver Imagen 2) está formado por dos montañas y un valle. Hay un coche que inicialmente se encuentra en el valle. El problema es que el motor no tiene la fuerza suficiente para vencer la gravedad y subir la montaña. Por lo tanto, el agente debe aprender a aplicar la acción del motor (tanto hacia delante como hacia atrás) en los momentos oportunos para conseguirlo, aprovechando la inercia del movimiento.

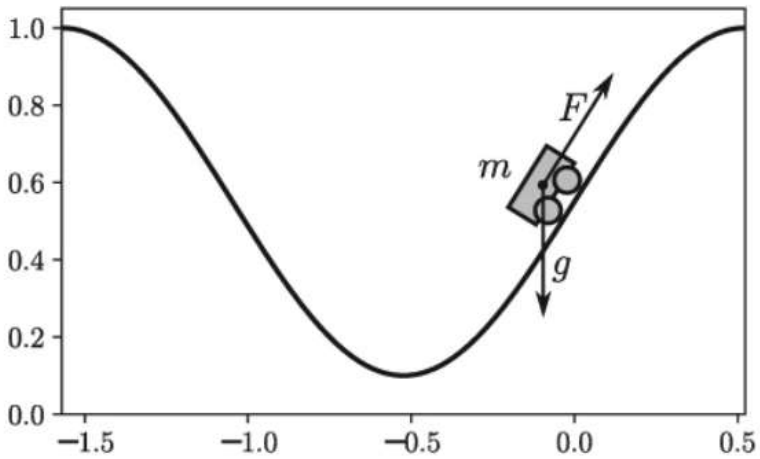

Imagen 2. Descripción gráfica del modelo "Mountain Car". La imagen se ha obtenido del artículo de referencia [1].

La red neuronal que controla el agente está formada por 4 neuronas sesitivas, 2 motoras y un número variable de neuronas ocultas (N_h, hidden neurons). El tamaño de la red es igual a:

$$N = N_m + N_s + N_h = 2 + 4 + N_h = 6 + N_h$$

Los sensores recibirán un input externo, comparable al campo externo ($h_i = I_i$). El valor de dicho input se definirá en función del estado del sistema. Las neuronas motoras definen la acción que realizarán los agentes. En el caso del coche, definen si el motor ejercerá fuerza hacia delante, hacia atrás, o si no ejercerá fuerza.

El modelo de red neuronal que se escogió es el siguiente: Los sensores y las neuronas motoras únicamente estarán conectadas a las neuronas ocultas, y las neuronas ocultas estarán conectadas al resto de neuronas del sistema (Ver Imagen 3).

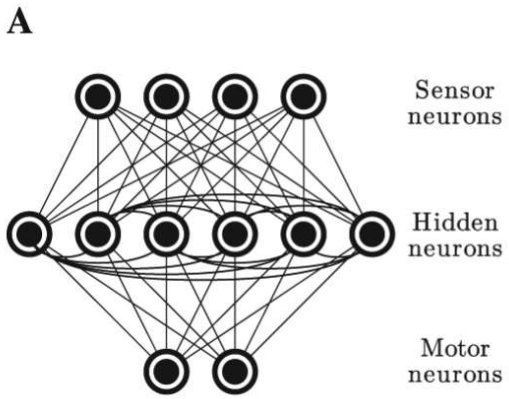

Imagen 3. Distribución de la red neuronal. La imagen se ha obtenido del artículo de referencia [1].

A pesar de que la distribución de la red no es crítica para el modelo, se escogió esta debido a que es una que está ampliamente extendida y permite conexiones recurrentes entre neuronas ocultas.

En este entorno se entrenaron 10 agentes, realizando 1000 iteraciones con T=5000 para redes de diferentes tamaños: 7, 8, 10, 14, 22, 38 y 70. El número de sensores y neuronas motoras se mantuvo para cualquier tamaño de la red.

Las redes se entrenaron siguiendo el método explicado anteriormente. El primer punto del análisis consistió en observar la distribución de las correlaciones obtenidas. En la Imagen 4 se observa que, tras el entrenamiento, se ha conseguido que sigan la distribución de correlaciones esperada.

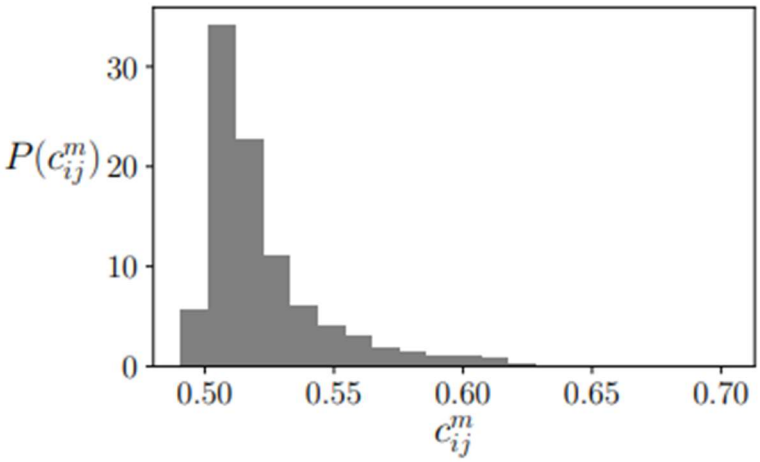

Imagen 4. Distribución de correlaciones de la red neuronal del Mountain Car con 64 neuronas ocultas y 10^6 simulaciones. La imagen se ha obtenido del artículo de referencia [1].

A continuación, se realizó una simulación del comportamiento del agente en el entorno. Se observa que el agente es capaz de subir la montaña y adquirir velocidad aumentando la energía del sistema (Ver imágenes 5 y 6).

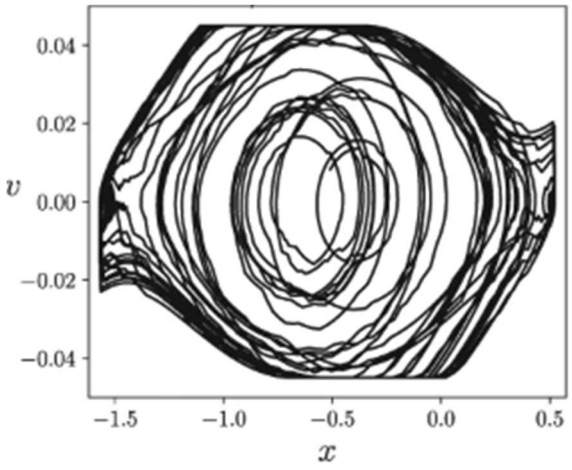

Imagen 5. Velocidad frente a la posición del Mountain Car con 64 neuronas ocultas y β=1. La imagen se ha obtenido del artículo de referencia [1].

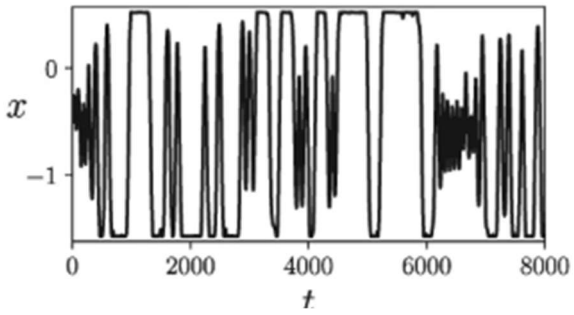

Imagen 6. Posición frente al tiempo del Mountain Car con 64 neuronas ocultas y β=1. La imagen se ha obtenido del artículo de referencia [1].

Esto es un indicador de que el sistema se está adaptando al entorno y probablemente se encuentre en un punto crítico. A pesar de esto, no es una condición suficiente. Para saber si se ha llevado el sistema a un punto crítico, se calculó la entropía del sistema, y su variación para obtener la capacidad calorífica (Ver imagen 7).

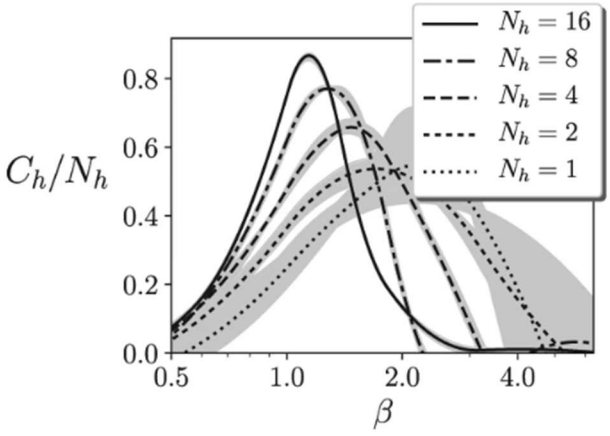

Imagen 7. Cociente de la capacidad calorífica y el número de neuronas ocultas para diferentes tamaños de red. La imagen se ha obtenido del artículo de referencia [1].

03. ANÁLISIS DE RESULTADOS EN DISTINTOS ENTORNOS

03.1. Mountain Car continuo

Todos los resultados mostrados en el apartado anterior se obtuvieron en el proyecto tomado como base para el proyecto. La primera modificación en el testeo de la red neuronal, propia de este trabajo, se realizó sobre el entorno "Mountain Car". De esta forma, aunque simplemente era una ligera modificación, se conseguiría coger la soltura necesaria con la manipulación del modelo para poder adaptar la red sobre otros entornos. Con esta modificación también se podía analizar la adaptación del agente a entornos más complejos.

En el "Mountain Car" explicado en el apartado anterior, cuando el coche conseguía llegar a la cima de cualquiera de las dos montañas, se impedía que el coche superara esa posición, y la velocidad se igualaba a 0. De esta forma, el coche siempre estaría operando entre las dos cimas.

La modificación introducida consistía en hacer un entorno con "montañas infinitas". Para poder visualizarlo con claridad, se creó un entorno con cinco valles y cinco montañas, donde el coche puede avanzar por ellas sin restricciones. Se muestran 5 montañas para visualizar mejor el comportamiento (Ver Imagen 8), aunque a efectos prácticos, el entorno es infinito, y el número de montañas visualizadas no tiene impacto en los resultados. En caso de que el vehículo llegara a un extremo del mapa, superando la posición máxima, automáticamente se llevaría al coche al otro extremo manteniendo su velocidad.

En resumen, se trata de un sistema continuo, ya que, si el vehículo llega a la posición máxima, o mínima, este es trasladado al otro extremo del mapa, manteniendo la velocidad.

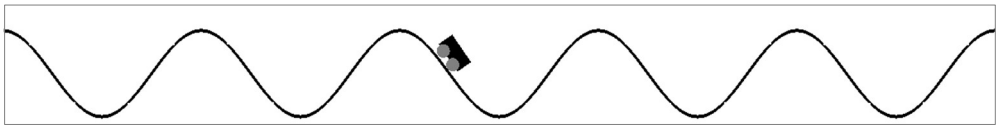

Imagen 8. Imagen del entorno "Mountain Car" continuo.

Definir el entorno de esta forma, no sólo permite observar que el coche es capaz de utilizar la inercia del movimiento (explicado en el apartado anterior), sino que se puede aumentar la velocidad máxima del sistema, y analizar el comportamiento del sistema cuando no se le define dicho límite.

En las simulaciones anteriores, la velocidad máxima del vehículo se fijaba a ±0.45 m/s para aumentar la dificultad de subir la montaña. En este caso se ha aumentado la velocidad máxima hasta ±4 m/s para observar el comportamiento del vehículo. Tras analizar varias veces el comportamiento del mismo, se ha observado que a pesar de que la velocidad máxima definida es de ±4 m/s, el vehículo no alcanza velocidades mayores de ±0.4 m/s. En la imagen 9 se observa la variación de la velocidad del coche en el tiempo.

En las imágenes siguientes (9 y 10) se observa que hay grandes variaciones de velocidad, tanto positivas como negativas, evidenciando unas fluctuaciones de energía propias de una transición de fase. En la Imagen 10 se puede observar el histograma de la distancia recorrida (L), entre cambios de dirección, en escala logarítmica. En dicha gráfica se observa que el comportamiento del vehículo es complejo por su variación. El comportamiento del agente se puede ver en un vídeo adjunto al

material suplementario (Ver MS2 → TFM - RCL → 00. Vídeos → MountainCar_Continuo.mkv; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/MountainCar_Continuo.mkv)

En la Imagen 10 se muestra la varianza de la variable "L", que es una medida de la complejidad del comportamiento (variación de las distancias recorridas). Este valor se utilizará como comparación en el siguiente apartado.

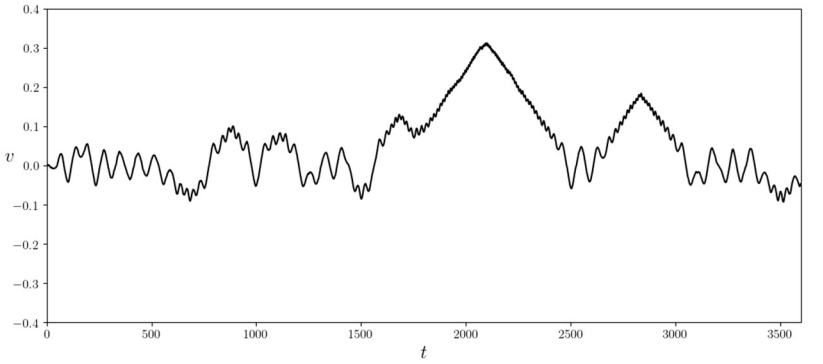

Imagen 9. Velocidad a lo largo del tiempo, para una red de 70 neuronas.

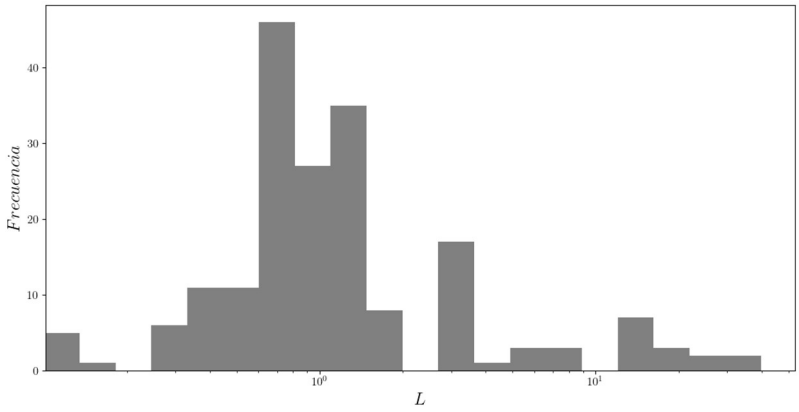

Imagen 10. Histograma de la distancia recorrida en escala logarítmica para una red neuronal de 44 neuronas ocultas y 1000 iteraciones durante T=18000. En este caso, var(L) = 16.67865.

03.2. Mountain Car continuo e irregular

Sobre el entorno descrito en el apartado anterior, se ha realizado una modificación adicional. Con esta modificación se espera crear un entorno en el que el agente pueda desarrollar comportamientos más complejos, para explorar hasta qué punto puede llegar la red neuronal. Se han introducido irregularidades en el entorno. En los casos anteriores, la altura del terreno correspondía a una función senoidal simple. En este caso, la altura, "y", se define de la siguiente forma: $y = \sin x + \cos 3x + 3 \cos 2x$.

Esto provoca que el terreno tenga "montañas" más altas que otras (Ver Imagen 11), y, por lo tanto, la dificultad del coche para adaptarse al entorno aumenta.

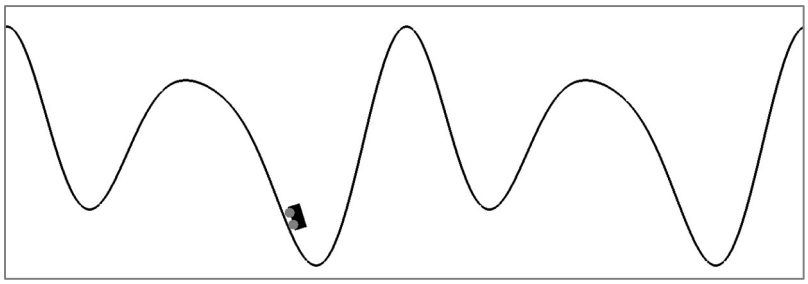

Imagen 11. Entorno irregular para el Mountain Car.

A pesar de que el entorno ha cambiado, los resultados que se obtienen de la adaptación del agente son similares a los obtenidos en el caso anterior. En esta situación, el coche también consigue subir las montañas, independientemente la irregularidad del entorno. Cuando el límite que se introduce en la velocidad es elevado, el coche tiende a alcanzar velocidades elevadas, con fluctuaciones de energía que son indicios de que el sistema está posicionado en una transición de fase. Los límites de estas fluctuaciones, a pesar del límite definido, son del mismo orden que en el caso anterior, aproximadamente 0,3-0,4 m/s (Ver Imagen 12). Se ha subido un vídeo del entorno al material suplementario (Ver MS2 → TFM - RCL → 00. Vídeos → MountainCarIrregular.mkv; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/MountainCarIrregular.mkv)

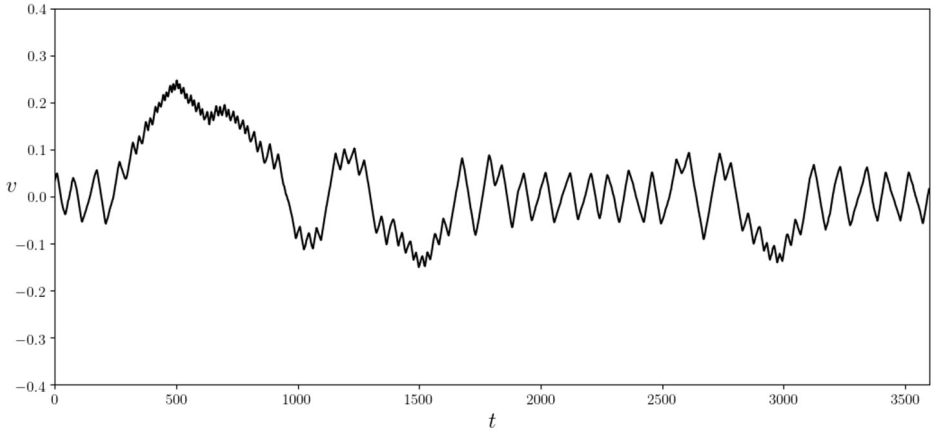

Imagen 12. Velocidad a lo largo del tiempo, para una red neuronal de 70 neuronas-

Sobre este entorno se ha decidido analizar si el sistema presenta comportamientos más complejos tal como se aumenta el tamaño de la red neuronal. La complejidad del comportamiento se medirá analizando las variaciones del movimiento del coche. Un ejemplo de comportamiento simple es: el coche se encuentra en el mismo valle durante toda la simulación, realizando balanceos similares durante toda la simulación. Por el contrario, si el coche cambia de valle, variando la longitud de sus movimientos (L) entre cambios de dirección, significará que sigue un comportamiento complejo.

Los resultados obtenidos con 44 neuronas ocultas se podrán comparar con el caso anterior, para analizar si el cambio del entorno ha supuesto que el agente presente comportamientos más complejos.

Para expresar lo anterior de forma gráfica, se representan los histogramas, para diferentes tamaños de redes, de la distancia recorrida hasta que el coche cambia de dirección. También se contabilizará

el número de cambios de dirección. Se espera que en los comportamientos más simples el número de cambios de dirección sea más elevado que en las simulaciones con redes neuronales de mayor tamaño.

También se calcula la variancia de las distancias recorridas en los distintos tamaños de red. Tal como se ha explicado anteriormente, esto es una medición de la complejidad del comportamiento ya que muestra la variación existente entre distancias recorridas.

A continuación, se muestran las gráficas para redes neuronales con una neurona oculta (Ver Imagen 13), con 24 neuronas ocultas (Ver Imagen 14) y con 44 neuronas ocultas (Ver Imagen 15). Se puede observar que, para el primer caso, todas las distancias recorridas entre cambios de dirección son muy reducidas, menores que 0,25 m. En el segundo caso el espectro es más amplio, recorriéndose distancias de hasta cerca de un metro. En el último caso la variación es destacable. Aunque el grueso de movimientos tiene una distancia de 1-2 m, hay recorridos de hasta 46 m.

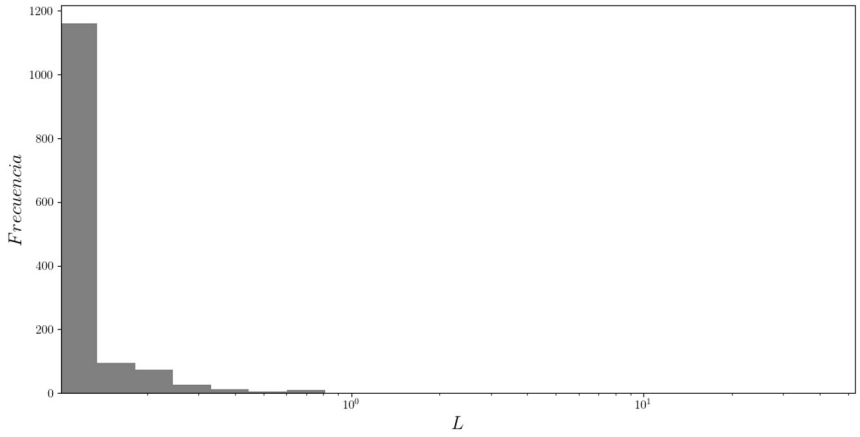

Imagen 13. Histograma que muestra la distancia recorrida en cada cambio de dirección (1 neurona oculta) en escala logarítmica. En este caso, var(L) = 0.00451.

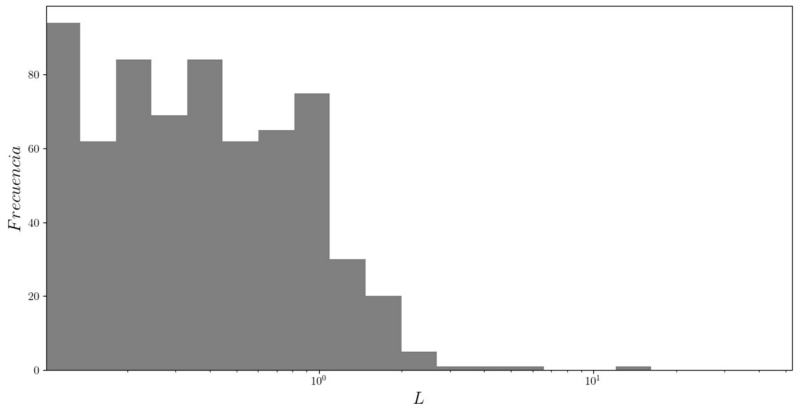

Imagen 14. Histograma que muestra la distancia recorrida en cada cambio de dirección (24 neuronas ocultas) en escala logarítmica. En este caso, var(L) = 0.69650.

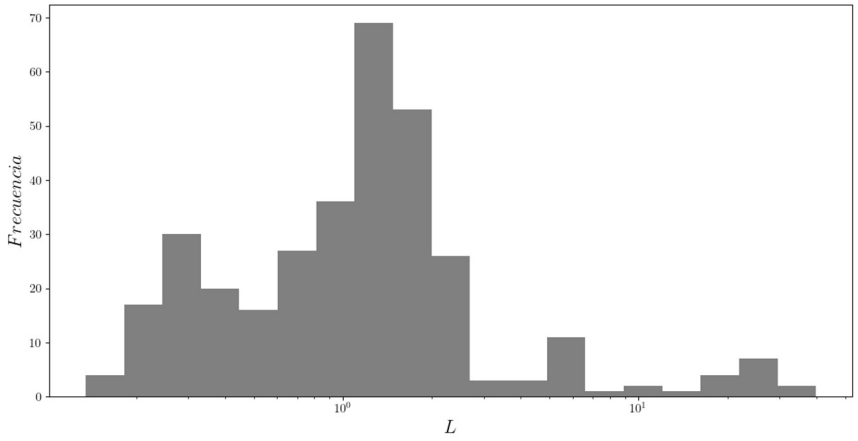

Imagen 15. Histograma que muestra la distancia recorrida en cada cambio de dirección (44 neuronas ocultas) en escala logarítmica. En este caso, var(L) = 19.43373.

Estos resultados son los esperados. Se evidencia que cuando se aumenta el tamaño de red, la complejidad del comportamiento del agente aumenta, incrementado la variabilidad del comportamiento.

Para todos los casos se ha calculado el número de cambios de dirección. Los resultados son: 1186 para el primer caso, 632 para el segundo y 196 para el tercero. Esto también son resultados esperados ya que, cuando el comportamiento es simple y el coche está en el mimo valle durante toda la simulación, realizará los cambios de dirección con más frecuencia que en las simulaciones con mayor tamaño de red.

Por otro lado, se puede comparar la Imagen 15 con la Imagen 10. En ambas simulaciones se han introducido los mismos parámetros (comentados anteriormente). La única diferencia entre los dos es el entorno en el que se mueven. En el caso de la Imagen 15, el entorno es más complejo ya que hay montañas de diferentes tamaños. Cuando se analizan las dos imágenes, se observa que el comportamiento del segundo caso, aunque ligeramente, presenta más variaciones y por lo tanto se deduce que es más complejo. Además, la variancia en el entorno irregular es ligeramente mayor que en el de la imagen 10.

03.3. Péndulo

Tras conseguir realizar que el agente funcione en un entorno diferente, aunque similar al original, se procede a probarlo en un entorno totalmente distinto. Dicho entorno también se ha obtenido la web OpenAI [2].

El sistema seleccionado está formado por una barra fija en uno de sus extremos. La fijación sólo restringe la posición, pero permite el giro sin fricción (Ver Imagen 16).

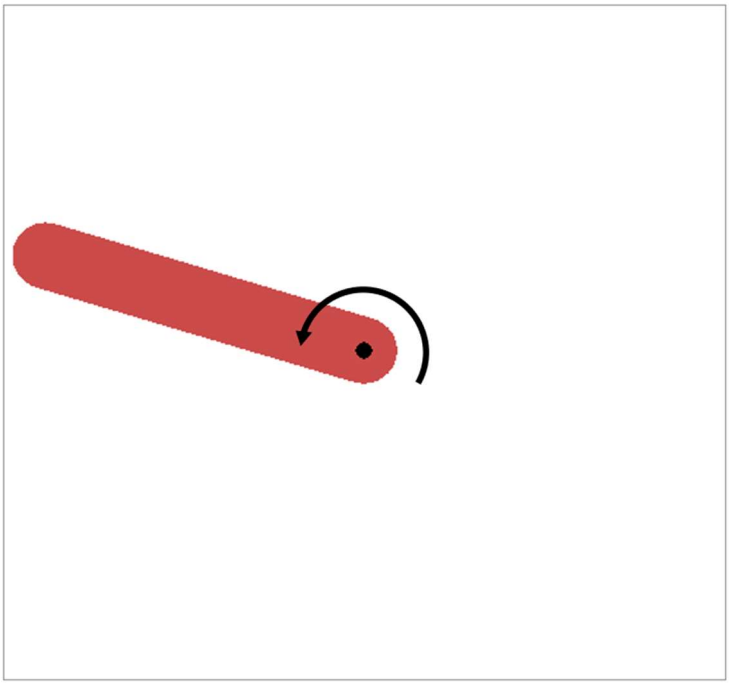

Imagen 16. Entorno del péndulo, que es una barra fija por un extremo (punto negro).

La posición inicial de la barra siempre es el punto más bajo, el de mínima energía, cuando está en vertical. El sistema tiene un motor que puede transmitir aceleración angular en ambos sentidos. La acción que realice el motor está controlada por la red neuronal.

Los parámetros del sistema se han calculado de forma que la acción del motor en un sentido no es suficiente para conseguir que la barra dé una vuelta completa. Para dificultar más el objetivo, se ha disminuido la acción del motor y se ha aumentado la acción de la gravedad.

Los cálculos se han realizado sobre un sistema estático, debido a que simplemente se quiere analizar que el sistema necesita utilizar la inercia del movimiento para conseguir dar la vuelta completa. En el momento que la barra está en horizontal, si estuviera en equilibrio, las ecuaciones de equilibrio, obtenidas de la segunda ley de Newton para la rotación, son:

$$\frac{dL}{dt} = M \rightarrow I\dot{\omega} = \frac{mgl}{2} \rightarrow \frac{ml^2}{3}\,\dot{\omega} = \frac{mgl}{2}$$

Dónde:

- L es el momento angular
- M es el momento neto de las fuerzas
- I es el momento de inercia, que para esta barra: $I = \frac{ml^2}{3}$
- $\dot{\omega}$ es la aceleración angular introducida sobre el sistema ($\dot{\omega}_{max} = 1s^{-2}$)
- $g = 10\ m/s^2$ es la gravedad
- m es la masa de la barra
- $l = 1$ es la longitud de la barra

Sustituyendo por los términos del sistema se obtiene que el término dependiente de la inercia y la aceleración angular ($I\dot{\omega} = 1/3$) es menor que el momento producido por el peso ($M = 4.9$). Por lo tanto, de este análisis se deduce que la acción de la gravedad se impone a la acción de giro, siempre que la barra no lleve una velocidad previa.

Para dificultar más el objetivo, se introducen los siguientes valores en el sistema:

- $\dot{\omega}_{max} = 0.8$
- $g = 10 * 1.2 = 12m/s^2$

A pesar de las dificultades a las que se ha sometido al agente, se ha observado que este es capaz de conseguir dar la vuelta completa. A continuación, se observan los resultados obtenidos para una red neuronal de 70 neuronas, con 4 neuronas sensitivas y 2 neuronas motoras. En las neuronas sensitivas se introduce la velocidad angular de la barra (transformada a código binario de 4 bits), y las motoras controlan la aceleración angular de la misma, que puede tomar los valores de 0.8, 0 ó -0,8m/s^2. Se ha observado que la barra es capaz de dar varias vueltas completas, elevando su velocidad, en ambos sentidos. Para evidenciar lo anterior, se muestran las figuras del ángulo y la posición del extremo libre de la barra, en las imágenes 17 y 18 respectivamente. También se adjunta un vídeo del comportamiento del péndulo en el material suplementario (Ver MS2 → TFM - RCL → 00. Vídeos → Péndulo_giro.mkv; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/Pendulo_giro.mkv)

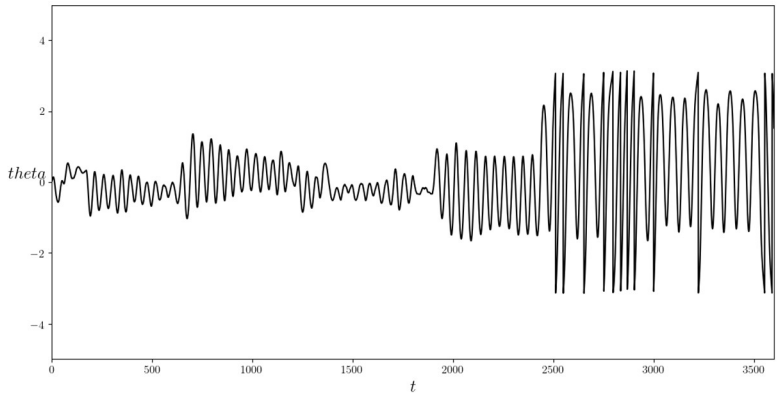

Imagen 17. Ángulo frente al tiempo, tomando theta = 0 cuando la barra está en vertical, en el punto más bajo.

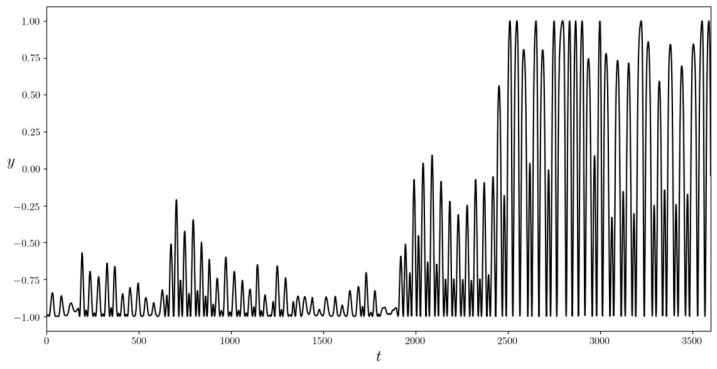

Imagen 18. Altura del extremo libre de la barra frente al tiempo.

Por otro lado, se ha obtenido la gráfica de la velocidad angular frente al tiempo (Ver Imagen 19). En dicha gráfica se observa que hay momentos en los que la velocidad se mantiene en una misma dirección (positiva o negativa) por un periodo de tiempo. Esto significa que la barra gira en el mismo sentido, dando varias vueltas completas.

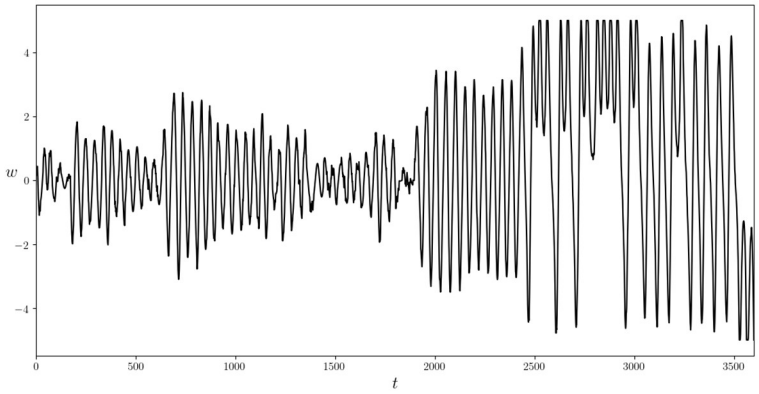

Imagen 19. Velocidad angular frente a la posición.

Todos los resultados anteriores se han obtenido estableciendo una velocidad angular máxima para el sistema relativamente de 5 rad/s. De esta forma se dificulta más la acción por limitar el efecto de la inercia.

Sobre el mismo entorno se han realizado modificaciones sobre este parámetro aumentándolo de forma considerable hasta los 30 rad/s. En este punto se ha observado que el sistema aumenta su energía tendiendo a velocidades elevadas en ambas direcciones. Esto es un resultado esperado ya que de esta red neuronal se espera que haga que los sistemas presenten elevadas fluctuaciones de energía (Ver Imagen 20). En el material suplementario se adjunta un vídeo en el que se observa la elevada velocidad de giro del péndulo (Ver MS2 → TFM - RCL → 00. Vídeos → Péndulo_maxv.mkv; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/Pendulo_maxv.mkv)

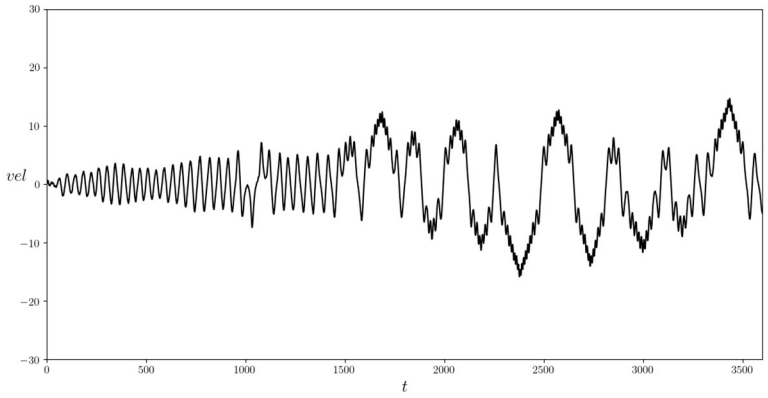

Imagen 20. Velocidad angular frente a la posición.

Las imágenes 21, 22 y 23 son histogramas del ángulo recorrido entre cambios de dirección. Lo que se observa en ellas es que el agente aumenta la complejidad de su comportamiento al incrementar el tamaño de red. Como se ha comentado anteriormente, la variancia de este ángulo recorrido es un indicador de la complejidad y se observa que aumenta al aumentar el tamaño de la red.

Este resultado es el esperado, y sigue la misma tendencia que el observado en el caso del Mountain Car.

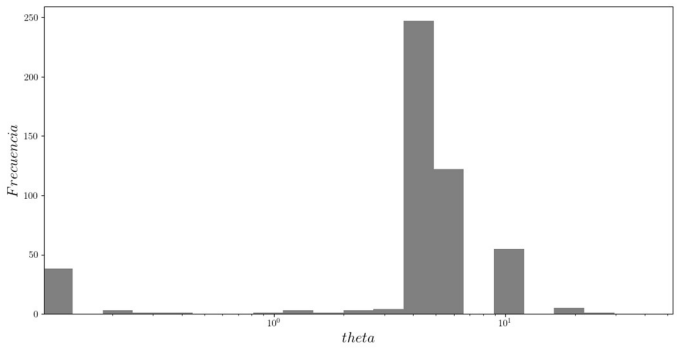

Imagen 21 Histograma del ángulo recorrido (theta) entre cambios de dirección en escala logarítmica para un tamaño de red de 1 neurona oculta. Para este caso, var(theta) = 8.95803.

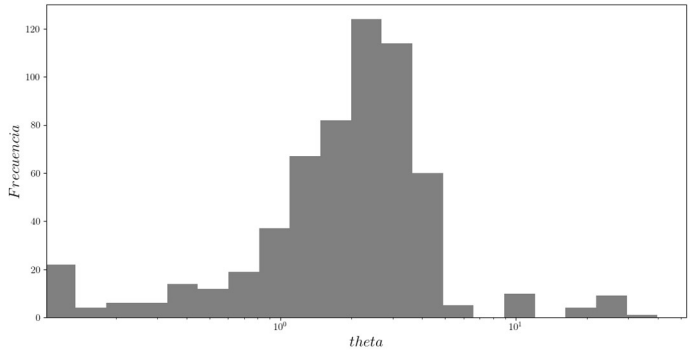

Imagen 22. Histograma del ángulo recorrido (theta) entre cambios de dirección en escala logarítmica para un tamaño de red de 44 neuronas ocultas. Para este caso, var(theta) = 70.13712.

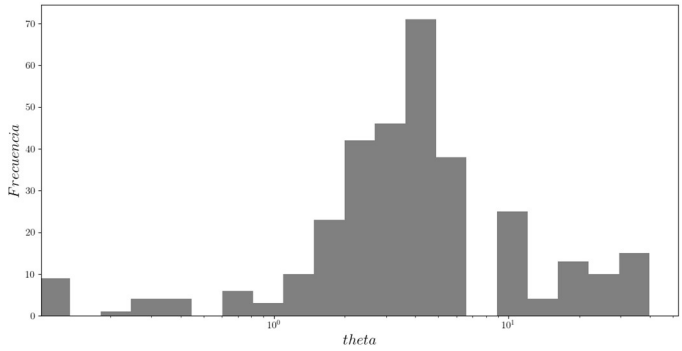

Imagen 23. Histograma del ángulo recorrido (theta) entre cambios de dirección en escala logarítmica para un tamaño de red de 64 neuronas ocultas. Para este caso, var(theta) = 100.61867.

03.4. Walker

04.05.1. Introducción

En el momento que se ha conseguido que la red neuronal controle los dos entornos descritos anteriormente se procede a aumentar el nivel de complejidad de los entornos. Este entorno se ha obtenido de la misma web que el anterior [2], aunque en este caso el sistema estaba dentro del módulo Box2D [3]. El agente es un robot formado por dos piernas y una carcasa (en adelante, "Walker"). La carcasa está colocada sobre las piernas (Ver Imagen 24). Las piernas están formadas por dos segmentos, de igual longitud, con una articulación en la unión. En la unión de la pierna con la carcasa también hay una articulación. Todos los movimientos de las articulaciones son independientes.

En el entorno seleccionado, el Walker camina por un suelo llano. A pesar de ser llano, este presenta alguna irregularidad para dificultar la tarea del robot.

Imagen 24. Walker original del entorno obtenido de Box2D [3]

Según se puede observar en la imagen, una de las tareas más complejas de este agente, es mantener el equilibrio. Tal como se ha comentado en esta memoria, el mantenimiento del equilibrio no es un objetivo que pueda conseguir esta red neuronal. Para confirmar lo anterior, se han realizado pruebas de la red neuronal sobre el Walker, sin obtener resultados positivos.

Por lo tanto, se procede a introducir una serie de cambios sobre el sistema para conseguir que el entorno consiga ser adecuado para la red neuronal en estudio. Esto es necesario hacerlo porque la red neuronal puede generar comportamientos complejos en el agente, pero no necesariamente tienen que ser comportamientos en los que el agente ande. Con estas modificaciones, se facilita que los comportamientos sean que el Walker ande.

04.05.2. Modificación del Walker

Para solucionar el problema de la estabilidad, se toma la solución de separarle las piernas al agente. Dicha separación se realiza de forma iterativa, hasta que se considera que tiene un equilibrio suficiente, pero que las piernas no están demasiado separadas ya que se busca que el Walker mantenga un alto grado de agilidad. Finalmente, la distancia entre piernas se define como se muestra en la imagen 25.

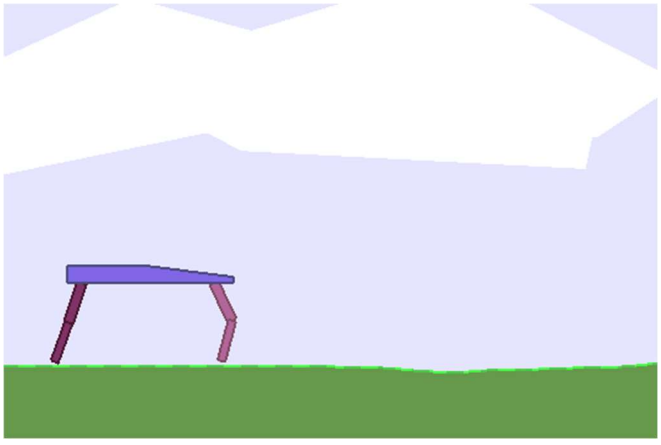

Imagen 25. Walker con las modificaciones para mejorar su estabilidad

En este punto, el Walker no tiene los problemas de estabilidad descritos anteriormente, por lo tanto, se realizarán varias pruebas de la red neuronal sobre el mismo. Debido a que tiene 4 articulaciones independientes, se definen cuatro neuronas motoras. Inicialmente se definen 16 neuronas sensitivas. Se asignan cuatro a la medición de la velocidad de cada articulación. La

medición se transforma a código binario, de máximo 4 bits, y esto es lo que se introduce como input en la red.

El número de neuronas ocultas se variará durante las pruebas realizadas, pero el tamaño con el que se realizaron las primeras pruebas fue de 30.

Con esta configuración, se procede a probar el comportamiento del Walker cuando lo controla la red neuronal descrita. Tras pocos intentos se observa que, a pesar de que el robot es mucho más estable que en un inicio, este no es capaz de avanzar en el entorno, incluso llegando a volcar.

Esto se debe a que, con la configuración inicial del Walker, las piernas tienen un rango de movimiento muy amplio, y la red no es capaz de definir los movimientos que deben realizar las articulaciones para que el robot avance.

Para tratar de solucionar los problemas descritos, se van a probar dos configuraciones más restrictivas. En primer lugar, se bloquearán completamente los movimientos de la pierna delantera del Walker, y sobre la pierna trasera, se aplicarán restricciones de forma que el ángulo de giro de cada articulación esté restringido.

El siguiente cambio que se introducirá sobre el Walker será desbloquear los movimientos de la pierna delantera. En esta situación, ambas piernas tendrán restringidos los ángulos de actuación, pero ambas tendrán cierta movilidad y se espera que en esta situación pueda ser más ágil que en el caso anterior.

En los próximos subapartados se detallarán los resultados obtenidos para los robots con las distintas configuraciones mencionadas.

04.05.3. Pierna delantera bloqueada

Al bloquearle la pierna delantera tal como se muestra en la Imagen 26, se consigue que la estabilidad de este aumente. Por contra, al aumentar su estabilidad, también se espera que pierda agilidad impidiendo que se pueda desplazar a velocidades elevadas.

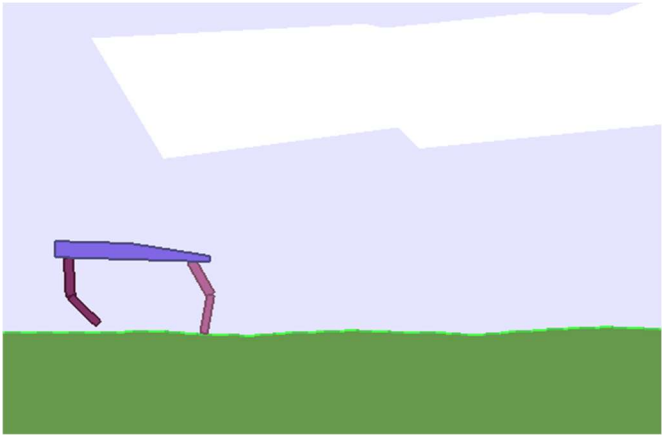

Imagen 26. Robot con la pierna delantera fija.

Los ángulos de actuación de las articulaciones de la pierna trasera se han restringido tal como se ha comentado en el apartado anterior.

En este punto, tras entrenar la red neuronal en este entorno, se ha probado como reaccionaba el robot ante esta casuística especial. Los resultados han sido los esperados, el Walker es capaz de

avanzar, aunque de forma lenta (Ver imagen 27). Se ha hecho un vídeo del comportamiento del Walker con la pierna delantera fija (Ver MS2 → TFM - RCL → 00. Vídeos → Walker_PiernasSeparadas_P1_fija.mkv)

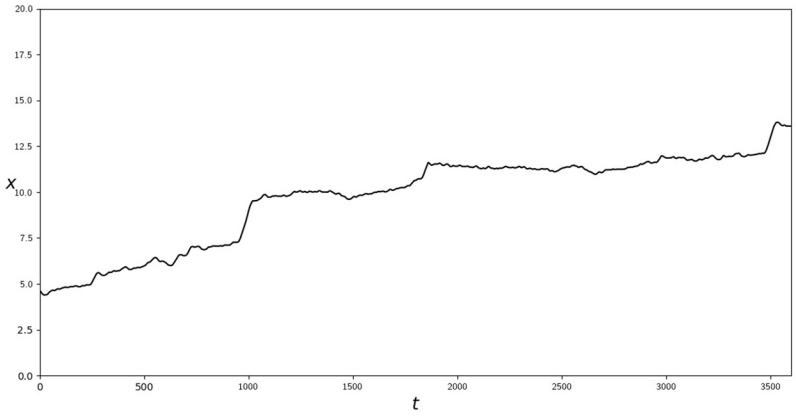

Imagen 27. Posición frente al tiempo, para el robot con la pierna delantera bloqueada (1000 iteraciones, t=3600 y size = 70).

Con estos resultados, se decide que es viable probar el funcionamiento de la red neuronal sobre el robot, aplicándole menos restricciones. Los resultados son satisfactorios porque, a pesar de que no se alcanzan velocidades altas, se comprueba que la red es capaz que el robot avance en unas determinadas condiciones.

04.05.4. Pierna delantera móvil

Tras los resultados satisfactorios obtenidos para la pierna delantera fija, se decide pasar a probar la red neuronal en un Walker más ágil y menos estable. Lo que se esperaría observar en este caso es que fuera capaz de avanzar más rápidamente. Para hacerlo más ágil, se "desbloquea" la pierna delantera. Esta tiene una movilidad similar a la de la pierna trasera, con los ángulos restringidos. A pesar de que el rango de movimiento de las articulaciones de ambas piernas es similar, la posición de las piernas no es igual.

Durante la construcción de las articulaciones, se observó la posición de las piernas en animales cuadrúpedos, y dicha estructura fue la que se intentó imitar en el robot. En la imagen 26 se observa el rango de movimientos que tiene el Walker.

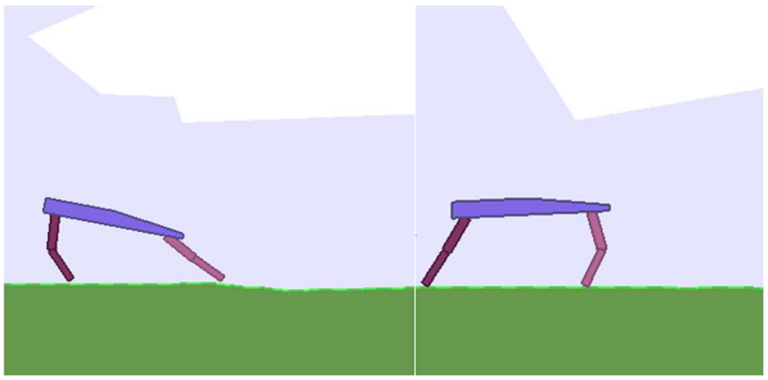

Imagen 28. Posiciones posibles de las piernas del robot con las restricciones de ángulos.

Tras generar el Walker como se ha indicado anteriormente, se ha procedido a analizar su comportamiento cuando está controlado por la red neuronal. Se ha observado que el robot presenta movimientos más dinámicos, pero muy inestables. En ocasiones se desplaza hacia atrás o permanece en el mismo lugar por un elevado periodo de tiempo.

Para tratar de solucionar esto, se introduce un cambio en la red neuronal. Se añaden 8 neuronas sensitivas (24 en total) para introducir dos inputs más en la red. Estos inputs son la posición velocidad en el eje x. Una vez se ha definido claramente la estructura de la red neuronal, se realiza todo el proceso de entrenamiento de la misma durante 1000 iteraciones, durante T=3600 para una red neuronal de un tamaño total de 90 neuronas (24 sensitivas y 4 motoras).

Cuando se ha simulado el comportamiento del Walker en esta situación se observa que este es capaz de avanzar de forma estable y más rápido que en el caso que tenía la pierna delantera bloqueada. A pesar de que ha mejorado la estabilidad, y aumentado la velocidad, se sigue desplazando, en ocasiones, en ambas direcciones (Ver Imágenes 29 y 30).

Este resultado se considera satisfactorio, ya que se ha conseguido que el Walker ande sobre el entorno, sin habérselo definido como objetivo. En otras palabras, ha aprendido a caminar por sí mismo. Debido a esto se considera que la red es capaz de hacer que se adapte a su entorno y de que avance con una velocidad elevada sobre el mismo. Una muestra del comportamiento del Walker se puede ver en el Material Suplementario (Ver MS2 → TFM - RCL → 00. Vídeos → Walker_PiernasSeparadas.mkv; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/Walker_PiernasSeparadas.mkv)

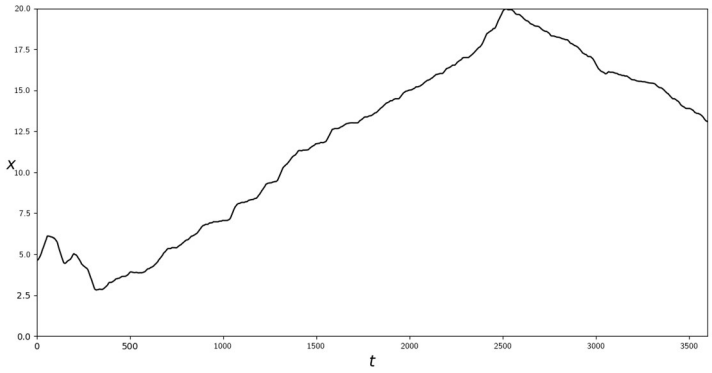

Imagen 29. Posición frente al tiempo (1000 iteraciones, t=3600 y size=90).

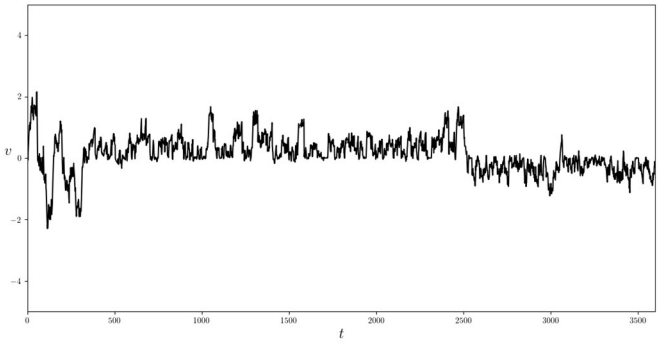

Imagen 30. Velocidad frente al tiempo (1000 iteraciones, t=3600 y size=90).

04.05.5. Levantarse

Tras los resultados satisfactorios obtenidos en los apartados anteriores sobre el Walker, se ha decidido cambiar su entorno para ver que comportamientos puede controlar la red neuronal.

En este caso, se quiere observar si la red es capaz de hacer que se puede levantar. La posición inicial será tener las piernas totalmente extendidas, en paralelo con el suelo (Ver imagen 31).

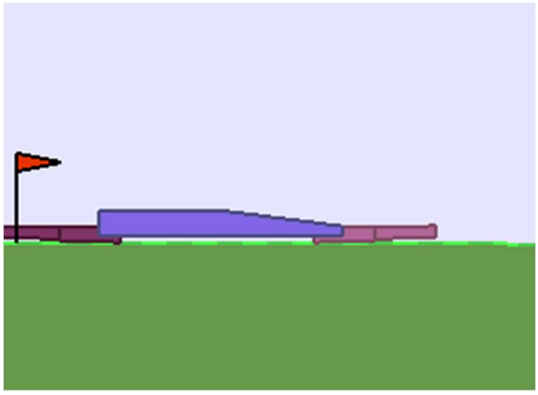

Imagen 31. Robot tumbado, con las piernas totalmente extendidas.

Se considerará que el robot se ha levantado cuando consiga una posición similar a las de los apartados anteriores (Ver imagen 32). En este punto, la carcasa alcanza posiciones de 0.8 m aproximadamente en el eje y.

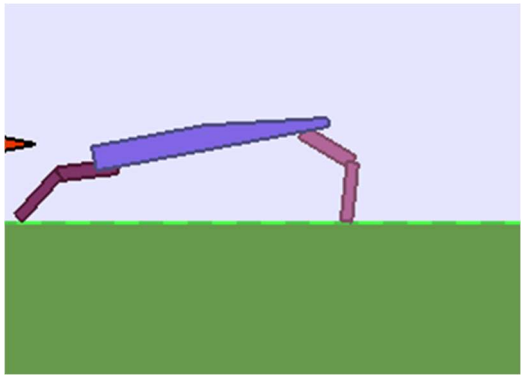

Imagen 32. Robot levantado.

Para realizar esta prueba, las restricciones de las articulaciones deben ser más amplias para que las piernas puedan llegar a la posición en la que el robot está tumbado.

La red que se ha utilizado para controlar el robot tenía un tamaño de 60 neuronas. 4 motoras y 20 sensitivas (16 para la velocidad de las articulaciones y 4 para la posición de la carcasa en el eje y). En este caso no se mide la velocidad del robot en el eje y, ya que se considera que es capaz de levantarse rápidamente sin introducir este parámetro.

Los resultados que se han obtenido tras las simulaciones se considera que son satisfactorios, ya que se ha observado que el robot es capaz de levantarse (ver Imagen 33),

A pesar de la parte positiva, se observa que, debido a la relajación en las restricciones de las articulaciones, el robot se levanta, pero presenta un comportamiento inestable, llegando incluso a

tumbarse varias veces en las simulaciones. Cuando se analiza el resultado, se considera lógico este comportamiento porque cuando el robot se levanta y se tumba, genera fluctuaciones en la energía del sistema, que es lo que se espera de la red neuronal. Este comportamiento se puede observar en un vídeo del material suplementario (Ver MS2 → TFM - RCL → 00. Vídeos → Walker_Levantándose.mp4; https://github.com/Rub22/TFM-RCL/blob/master/00.%20V%C3%ADdeos/Walker_Levant%C3%A1ndose.mp4)

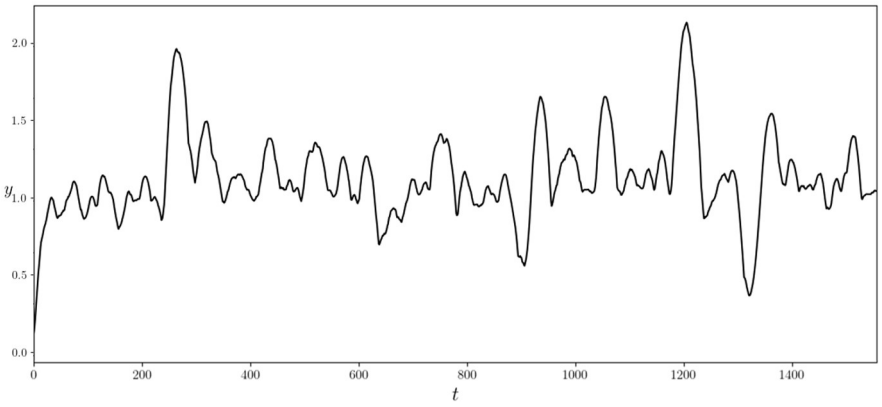

Imagen 33. Posición en el eje y frente al tiempo.

04. CAPACIDAD CALORÍFICA Y ENERGÍA

En el desarrollo de todos los apartados anteriores se ha observado que la red neuronal es capaz de hacer que los agentes se adapten a su entorno, consiguiendo los objetivos, sin establecérselos de forma explícita. A pesar de esto, el propósito del proyecto también es analizar si la red neuronal es capaz de posicionar los agentes en un punto crítico.

Como se ha comentado, el funcionamiento de los agentes y las evidencias de fluctuaciones de energía son un buen indicador del posicionamiento en una transición de fase, pero no es una condición suficiente. La condición suficiente es un incremento de la capacidad calorífica del sistema de forma continua, en este caso en beta igual a 1. Dicho incremento se deberá acentuar tal como se incremente el tamaño de la red.

En el artículo tomado como ejemplo [1], la capacidad calorífica se calcula a partir de la entropía. Esto tiene un coste computacional muy elevado, por lo que en este proyecto se decide calcularla a partir de la energía del sistema:

$$C(\beta) = \beta^2\left(E^2(s)\right) - (E(s))^2)$$

04.1. Cálculo de la energía

El primer paso para obtener la capacidad calorífica del sistema consiste en calcular la energía de la red neuronal para distintos valores de betas. El cálculo se realiza, para cada beta, 10^5 veces. Las redes se entrenaron para T=6000 durante 1000 iteraciones. Se espera que haya una transición de la función de la energía con beta igual a 1 debido a que esta es la temperatura del punto crítico en este modelo. La energía es alta con betas bajas y viceversa. Es decir, con temperaturas altas, el sistema se estabiliza en estados de elevada energía, y con temperaturas bajas el sistema se estabiliza en estados de baja energía.

Para comprobar si se cumple la teoría explicada, se graficará la energía frente a β, con β entre 0 y 3. Se eligen estos límites debido a que el punto de mayor interés es β=1. Dichas gráficas se obtienen para el entorno del Mountain Car y para el caso del Walker con ambas piernas móviles.

El resultado para el caso del Mountain Car, con varios tamaños de red, es el esperado (Ver Imagen 34). Se puede observar una transición de energía cuando β=1. Para betas elevadas, es decir, temperatura baja, se observan fluctuaciones. Esto se debe a que cuando la temperatura es elevada actúa como filtro de ruido, pero esto a bajas temperaturas no sucede.

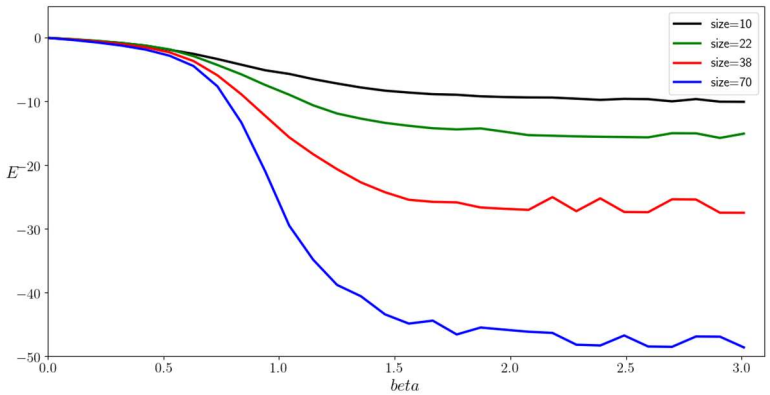

Imagen 34. Energía frente a beta para el caso del Mountain Car.

Para el Walker se obtiene la misma gráfica, y esta sigue la misma tendencia (Ver Imagen 35). A pesar de que se reinicia la posición del Walker cada 4 iteraciones para que no se estanque en un estado de funcionamiento, también aparecen fluctuaciones a betas altas.

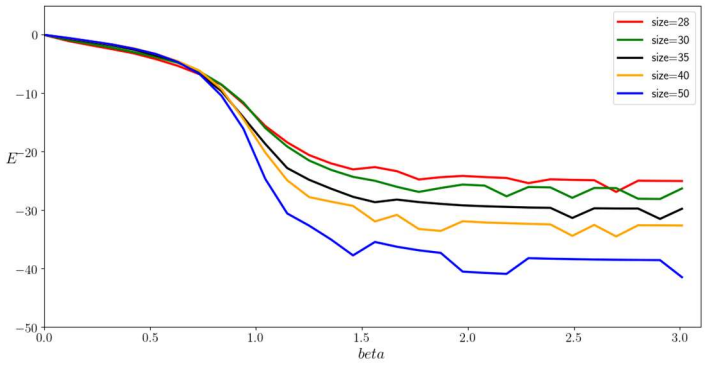

Imagen 35. Energía frente a beta para el caso del Walker, reiniciando la posición.

04.2. Cálculo de la capacidad calorífica

Tras el cálculo de la energía, se procede a calcular la capacidad calorífica en ambos entornos para diferentes tamaños de red. El resultado esperado, como ya se ha comentado, es un máximo en beta igual a uno. La capacidad calorífica es un indicador de la complejidad del sistema. Cuanto mayor sea el máximo en β=1, mayor será la complejidad de la red neuronal.

Para observar los comportamientos de las redes neuronales, se obtienen las gráficas de la capacidad calorífica entre el tamaño de red con betas entre 0 y 3. Esto será un indicador de si, independientemente de que aumente la complejidad de la red, aumenta la complejidad de cada neurona.

En la imagen 35 se puede observar que para el caso del Mountain Car el máximo de la capacidad aumenta ligeramente al aumentar el tamaño de red (Ver Imagen 36), pero para el caso del Walker, el máximo disminuye (Ver imagen 37). Este resultado no es el esperado. Se esperaban obtener resultados similares a los del artículo de referencia (Ver Imagen 7). En dicha imagen el máximo de la capacidad calorífica aumenta con el tamaño de red.

La diferencia de resultados se puede deber a que el número de simulaciones no sea suficiente. A pesar de que se han utilizado las mismas que en el artículo de referencia, estas pueden no ser suficientes por las variaciones en el método de cálculo y de los entornos. No se ha aumentado el número de simulaciones porque el tiempo de simulación sería excesivo (orden de semanas) para las redes definidas.

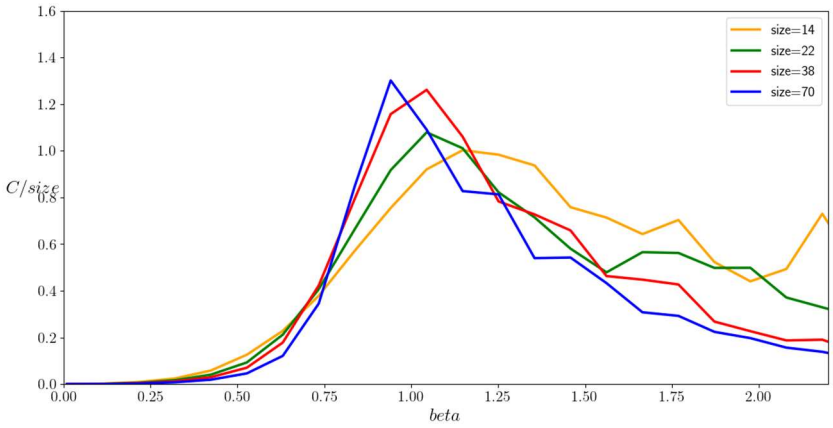

Imagen 36. Capacidad calorífica entre el tamaño de red para el Mountain Car

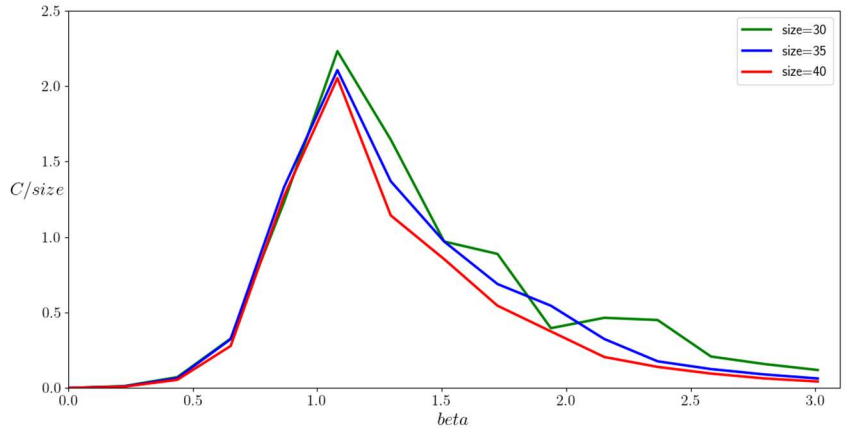

Imagen 37. Capacidad calorífica entre el tamaño de red para el Walker.

A pesar del resultado anterior, se va a analizar si las redes, para los dos entornos del caso anterior, aumentan su complejidad de forma global a aumentar el tamaño de red. Esta vez, los resultados han sido los esperados ya que en ambas gráficas (Ver imágenes 38 y 39) se observa un máximo de la capacidad calorífica en beta igual a uno, aumentado tal como se incrementa el tamaño de red.

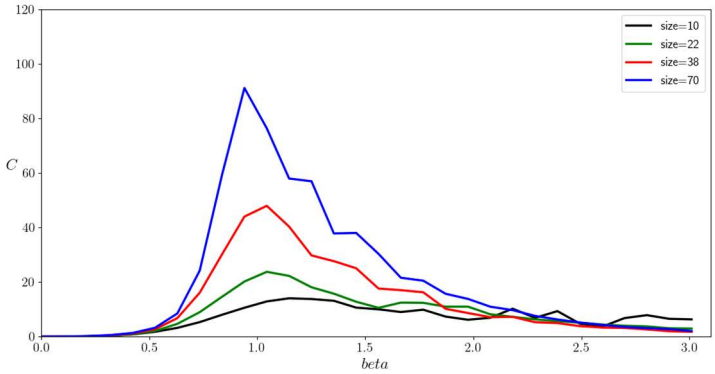

Imagen 38. Capacidad calorífica frente a beta para el Mountain Car.

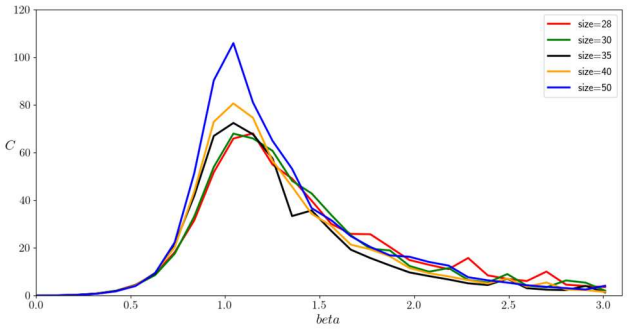

Imagen 39. Capacidad calorífica frente a beta para el Walker.

Por todo lo explicado en este anteriormente, se considera que la red neuronal es capaz de llevar los sistemas a un punto crítico.

Esto ya se había comprobado en el proyecto de referencia [1] para un entorno simple, pero en este proyecto se ha demostrado que esta afirmación también se cumple en un entorno más complejo como es el caso del Walker.

05. CONCLUSIONES

El punto de partida para realizar este proyecto se basa en la idea de que hay sistemas biológicos se comportan en puntos críticos parecidos a las transiciones de fase en termodinámica. En un artículo de la revista Scientific Reports [1] se propone un algoritmo de aprendizaje que intenta replicar estos comportamientos.

En dicho artículo se prueba la red neuronal en dos entornos simples y se comprueba que es capaz de hacer que varios agentes se adaptaran a su entorno llevándolos a un punto crítico. Los agentes presentan comportamientos espontáneos sin haberles definido objetivos.

En este proyecto se analiza si la red neuronal es capaz de hacer que otros sistemas más complejos se posicionen en un punto crítico y se adapten a su entorno generando comportamientos espontáneos. Primero se realizan las pruebas sobre un entorno muy similar al Mountain Car del artículo, pero con alguna modificación que aumenta su complejidad. Posteriormente se realizan pruebas sobre otro entorno de una complejidad similar, el Péndulo. Tras analizar los resultados en estos entornos se considera que son satisfactorios y que se pueden generalizar los resultados del artículo.

Tras realizar las pruebas en los entornos descritos, se afronta un reto más ambicioso sobre un entorno denominado "Walker". En este entorno se observa si la red es capaz de hacer que el Walker ande con cierta estabilidad y velocidad, y si es capaz de levantarse del suelo.

Dichos objetivos no se definen de forma explícita. Simplemente se le introducen los inputs de la velocidad de rotación de las articulaciones, y, en cada caso, la posición y velocidad en un eje concreto. A partir de aquí, con el entrenamiento de la red, el Walker cumple ambos objetivos, adaptándose al entorno generando comportamientos complejos y espontáneos. Este es un resultado muy destacable ya que, a pesar de que se ha limitado el comportamiento del Walker, la red neuronal ha sido capaz de hacer que un agente complejo sea capaz de caminar y levantarse sin habérselo definido de forma explícita.

Todo lo comentado anteriormente aporta indicios de que la red neuronal es capaz de llevar a los agentes a trabajar en un punto crítico, pero no aporta certeza. Para asegurar esto, en el proyecto modelo, se calcula la capacidad calorífica. Esto se realiza porque una condición suficiente de posicionamiento en un punto crítico es que la capacidad calorífica diverja tal como se incremente el tamaño de red. La capacidad calorífica también es un indicador de la complejidad del sistema.

Tras realizar todos los cálculos se observa que la red neuronal ha sido capaz de posicionarlos en un punto crítico.

Como conclusión, se considera que el resultado del proyecto es satisfactorio. Se ha observado que la red neuronal es capaz de posicionar los sistemas en una transición de fase, y de esta forma hacer que se adapten a su entorno consiguiendo los objetivos, pero sin definírselos de forma explícita.

05.1. Dificultades

A lo largo del desarrollo del proyecto se produjeron varios momentos de dificultad. Aunque se considera que todos ellos se han superado, en este apartado se indicarán los dos más destacables:

- Comprensión del modelo y del funcionamiento de la red neuronal: La primera parte del proyecto que consistía en la comprensión del artículo publicado en la revista Scientific Reports [1] y la revisión del código asociado fue bastante exigente.
  Su complejidad, y el hecho de que eran temas nuevos, hicieron que esto ocupara una parte importante del desarrollo del proyecto. Durante la comprensión del modelo también se tuvieron que estudiar en detalle modelos de redes neuronales inspirados en la física estadística, como el modelo de Ising.

- Prueba de la red neuronal en un entorno complejo: La búsqueda del entorno más complejo sobre el que probar la red también se realizó en en la web de OpenAI [2].
  Dentro de dicha web los entornos estaban agrupados por módulos, que requerían de instalaciones de complementos adicionales. El primer módulo que se intentó descargar fue uno denominado "MuJoCo". Este tenía varios entornos que cumplían los requisitos, pero tras la instalación del módulo en Windows, no se consiguió que la interfaz con Python funcionará correctamente.
  Tras analizar comentarios de otros usuarios, se descubrió que este módulo da problemas en las nuevas versiones de Windows.
  Como alternativa, se seleccionó otro módulo denominado "Box2d". Sobre dicho módulo se seleccionó un entorno que, tras varias modificaciones, cumplió con los requisitos esperados. Tras su correcta instalación, también hizo falta un gran esfuerzo para familiarizarse con las variables de dicho entorno y su funcionamiento.

06. TRABAJO FUTURO

Los algoritmos clásicos de inteligencia artificial suelen tener el problema de que no son capaces de adaptarse a entornos para los que no han sido diseñados. Por ejemplo, se pueden diseñar agentes que, para jugar al ajedrez, son mejores que los humanos, pero no pueden hacer nada cuando se cambian las reglas del juego para el que han sido programados.

Es decir, se diseñan para lograr objetivos muy concretos, pero después no son capaces de aplicar esa inteligencia a otros campos. Por lo tanto, escalar la inteligencia artificial a una amplia variedad de situaciones es muy costoso porque, para cada situación, el algoritmo debe sufrir cambios de diseño muy significativos.

Con este modelo de red neuronal, se abre una nueva posibilidad. Se ha comprobado que la red neuronal es capaz de controlar, de forma satisfactoria, diferentes agentes en situaciones muy distintas. En el futuro, se podrían explorar redes neuronales basadas en principios similares a los propuestos en este proyecto. Ya que es posible que, con estos modelos, sea mucho más fácil la escalabilidad de las redes neuronales.

También se podría analizar la viabilidad de que este algoritmo se integrara con otro que permita aprender un objetivo definido. Esto solucionaría algunos problemas encontrados en esta red ya que, por ejemplo, en el caso del Walker, no siempre el comportamiento generado de forma espontánea era compatible con el esperado por el diseñador.

Una continuación más específica podría consistir en lo siguiente. Se podrían realizar pruebas en entornos más complejos, con tamaños de red mayores (y por lo tanto mayor coste computacional), para analizar qué sistemas puede llegar a controlar esta red. En este proyecto se han realizado pruebas en entornos complejos, pero en todo momento han sido en dos dimensiones, y aplicando restricciones de movimiento. Una posible continuación podría consistir en aumentar las dimensiones y eliminar restricciones para observar el comportamiento.

Otra posible continuación se podría basar en lo siguiente: la red neuronal utilizada en este modelo es una red que simula conexiones simétricas, invariantes en el tiempo y sin acción de campo externo. La realidad es que esto es poco realista aplicado a los sistemas biológicos. En dichos sistemas hay acción de campos externos, con el tiempo varían tanto el campo externo como las correlaciones, y las relaciones entre neuronas pueden ser asimétricas (una neurona A puede influenciar una neurona B, y no viceversa).
Un modelo que contempla esta situación es el: “Kinetic Ising Model”. En este caso la red neuronal se asemejaría más a un sistema biológico real, y se podrían comparar los resultados obtenidos con ambas redes.

07. ANEXO I

07.1. Descripción del modelo de Ising

[6] El modelo de Ising es un sistema matemático utilizado principalmente para el estudio de las transiciones de fase. Está compuesto por elementos que pueden tomar el valor de -1 o +1. Para explicar el modelo se toma el ejemplo de un sistema magnético. El -1 indica que cierto elemento está apuntando abajo y el +1 indica que dicho elemento está apuntando arriba. En este modelo, también se contempla la existencia de un campo externo, descrito por la variable "h".

La energía del sistema se describe a través de Hamiltoniano, "H":

$$H = -\sum_{i<j} J s_i s_j + \sum_i h s_i$$

Dónde:

- J es el factor de escala entre interacción entre espines y energía.
- s_i es el valor del spin del elemento i-ésimo.

Atendiendo a lo anterior, y suponiendo que los elementos sólo afectan a los elementos "vecinos", se obtiene que, cuando los spines de dos elementos apuntan en la misma dirección, la energía es -J (sin tener en cuenta el campo externo). Por lo tanto, en el momento que los spines apuntan en la misma dirección, la energía del sistema disminuye. De acuerdo con lo anterior, podría esperarse que la tendencia fuera que todos los spines apuntaran en la misma dirección ya que esto minimizaría la energía del sistema.

A pesar de lo anterior, hay que tener en cuenta que, según la termodinámica, en estas condiciones, el equilibrio no vendrá dado por el estado de mínima energía, sino por el estado de mínima energía libre de Helmoltz. La energía libre de Helmoltz se define:

$$F = U - TS$$

Dónde:

- U es la energía interna
- T es la temperatura
- S es la entropía

Debido a la ecuación anterior, el mínimo de energía libre depende de una relación entre el producto TS y la energía interna.

La energía interna del sistema es menor en el momento que todos los spines apuntan en la misma dirección. En esta situación la entropía alcanza su mínimo, ya que es el estado más "ordenado" posible.

Si la temperatura es baja, el producto TS será bajo, y por lo tanto el término que tendrá mayor peso en el sistema será la energía interna, así que el sistema alcanzará el equilibrio cuando todos los spines apunten en la misma dirección.

En caso de que la temperatura aumente, el término que tendrá más peso en el sistema será el producto TS. Por lo tanto, el equilibrio se alcanzará cuando la entropía aumente, haciendo que la energía libre de Helmoltz se minimice.

El modelo de Ising tiene dos fases bien diferenciadas, una gobernada por la entropía, y la otra gobernada por la energía interna. La magnitud que define el régimen en el que se encuentra el sistema es la temperatura. En este punto aparece una transición de fase, o punto crítico, a la temperatura crítica. La temperatura crítica (T_c) es tal que si $T > T_c$, el sistema se encontrará en la fase dominada por la entropía, y si $T < T_c$, el sistema se encontrará en la fase dominada por la energía interna.

07.2. Selección del Modelo de Ising

En el artículo de referencia para este proyecto [1], se selecciona un modelo de Ising 2D sin acción de campos externos (i.e. h = 0) que se encuentra en el punto crítico. El punto crítico se alcanza, para estas condiciones, cuando el valor de J es igual a $\log(1+\sqrt{2})/2\beta$.

- $\beta = 1/(TK_b)$
- T, es la temperatura del sistema
- K_b = Constante de Boltzmann

Se toman estas condiciones para el modelo debido a que es una de las pocas situaciones en la que se puede obtener una solución analítica.

Para simplificar, el término β se toma como 1. Esto es posible porque el parámetro β se puede definir de forma arbitraria, ya que será el punto de referencia. Es decir, en este modelo, el valor de temperatura que haga que β=1 será la temperatura crítica.

En la Imagen 40 se muestra una red bidimensional (100x100) que representa el modelo de Ising. Los spines positivos y negativos están representados por los colores azul y amarillo. Se puede observar como a baja temperatura todos los spines apuntan en la misma dirección y a altas temperaturas el sistema se encuentra lo más "desordenado" posible.

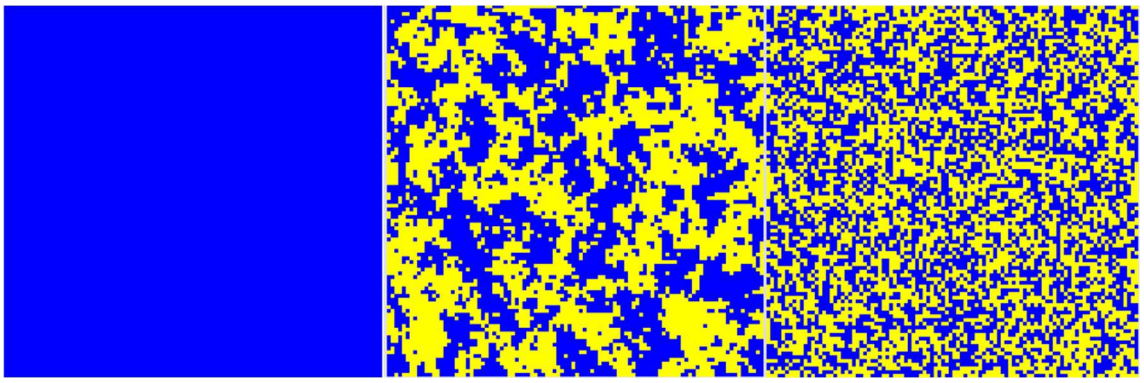

Imagen 40. Representación de los spines positivos y negativos (azul y amarillo) en el modelo de Ising a baja temperatura (izquierda), a la temperatura crítica (medio), y a alta temperatura (derecha) [7]

08. BIBLIOGRAFÍA

[1] Aguilera, M., Bedia, M.G. Adaptation to criticality through organizational invariance in embodied agents. Sci Rep 8, 7723 (2018). https://doi.org/10.1038/s41598-018-25925-4

[2] OpenAI, San Francisco, California. (2015) https://openai.com/

[3] Eric Catto, Box2d (2020) https://box2d.org/

[4] Departamento de física aplicada III, Universidad de Sevilla. Barra oscilando respecto a uno de sus extremos (GIC) (2014). https://laplace.us.es

[5] John M Beggs. The criticality hypothesis: how local cortical networks might optimize information processing. The royal Society (2007). https://doi.org/10.1098/rsta.2007.2092

[6] Statistical Mechanics II, Stanford University (2018). https://stanford.edu/~jeffjar/statmech2/

[7] Daniel V. Schroeder, Physics department, Weber State University. Consulta en junio de 2020 https://physics.weber.edu/schroeder/software/demos/IsingModel.html

09. MATERIAL SUPLEMENTARIO

- **MS1:** https://github.com/Rub22/TFM-RCL : En este proyecto se han subido diferentes partes del código utilizado para la realización del trabajo fin de máster. Se han subido por separado las partes en las que se simula el comportamiento del agente, y las partes en las que se calcula la capacidad calorífica. Se sube el código de:
  - El Mountain Car Original
  - El Mountain Car en un entorno continuo
  - El Mountain Car en un entorno continuo e irregular
  - El péndulo
  - El Walker avanzando
  - El Walker levantándose

- **MS2:** Adicionalmente, se suben vídeos de algunos de los comportamientos que se han simulado (https://github.com/Rub22/TFM-RCL/tree/master/00.%20V%C3%ADdeos). En concreto:
  - Comportamiento del Mountain Car en el entorno continuo y en el irregular
  - Comportamiento del péndulo (Con distintas vmax)
  - Walker avanzando con la pierna delantera bloqueada
  - Walker avanzando con las dos piernas móviles
  - Walker levantándos

- **MS3**: Tanto el código, como los vídeos obtenidos durante la realización del proyecto de referencia fueron subidos por sus autores al siguiente repositorio: https://github.com/MiguelAguilera/Adaptation-to-criticality-through-organizational-invariance