Universitat
de les Illes Balears

TRABAJO DE FIN DE MÁSTER

Evaluación de Políticas de Aprendizaje por refuerzo Aplicado a un Agente Bípedo

Aleix Seguí Cordero

Máster Universitario en Sistemas Inteligentes (MUSI)

Especialidad: Inteligencia artificial y robótica móvil

Centro de Estudios de Posgrado

Año Académico 2020-21

Evaluación de Políticas de Aprendizaje por refuerzo Aplicado a un Agente Bípedo

Aleix Seguí Cordero

Trabajo de Fin de Máster
Centro de Estudios de Posgrado
Universidad de las Illes Balears

Año Académico 2020-21

Palabras clave del trabajo:
Inteligencia artificial, aprendizaje por refuerzo, algoritmo genético, Pygame ...

Gabriel Moyà Alcover
Jose Maria Buades Rubio

Evaluación de Políticas de Aprendizaje por Refuerzo Aplicado a un Agente Bípedo

Aleix Seguí Cordero
Tutor: Gabriel Moyà Alcover y Jose Maria Buades Rubio
Trabajo de fin de Máster Universitario en Sistemas Inteligentes (MUSI)
Universitat de les Illes Balears
07122 Palma, Illes Balears, Espanya
aleix_segui@hotmail.com

***Resumen*—Desde hace muchos años los videojuegos han adoptado las tecnologías que aporta la inteligencia artificial para mejorar sus prestaciones, pudiendo así crear objetos con una inteligencia independiente al jugador. Dentro de las tecnologías de la inteligencia artificial más utilizadas en el campo de los videojuegos, se encuentra el reinforcement learning o los algoritmos genéticos. Tenemos un claro ejemplo, uno de los que motivo la realización de este trabajo, con el DeepMind de Google, en el cual cuerpos con distintas anatomías aprenden a desplazarse por terrenos irregulares. En el presente trabajo se ha construido un cuerpo bípedo mediante librerías de videojuegos, como son Pymunk y Pygame, el cual intentará aprender a caminar mediante el uso de algoritmos de aprendizaje por refuerzo y algoritmos genéticos. Los agentes están restringidos a usar una red neuronal básica que contiene 12 neuronas. El algoritmo modifica los parámetros de la red neuronal básica, mediante la aplicación de ruido y selección de los mejores agentes. La selección de los agentes se hará mediante políticas de aprendizaje, estas políticas valoran características tales como la velocidad, la distancia recorrida o si está caminando de forma erguida. Se comparan las distintas políticas mediante un análisis estadístico de los resultados obtenidos. También, se analiza la convergencia de los algoritmos para asegurar que el número de generaciones no impiden llegar al punto máximo del aprendizaje. En los resultados obtenidos ningún agente emula la biomecánica del andar humano. Por otra parte, se concluye que las políticas donde se combinan diferentes recompensas obtienen mejores resultados.**

.

ABSTRACT

For many years, in the field of videogames have adopted technologies provided by artificial inteligence for improve their qualities. From this technologies, the videogames started to create objects with an independent inteligence of the player. The reinforcement learning and genetics algorithms are one of the most used IA technologies in videogames. We have a clear example on Google's Deep Mind, that was an inspiration for doing this proyect. The main goal of Google's Deep mind is to achieve that some bipedal structures learns to walk and overcome a diferent type of obstacles with an algorithm that combines diferents technologies of reinforcement learning and genetics algorithms. On this proyect we will build a bipedal structure using videogames libraries, such as Pygame and Pymunk, wich will attempt to learn to walk correctly. We will use reinforcement learning and genetics algorithms with a simple neural network, that only has 12 neurons. Our algorithm modifies the neuron's parameters and will choose the best individuals of every generation. The algorithm will choose the individuals of the next generation evaluating their performance during a simulation. The evaluation is determined by a learning policies that reward the distance, the velocity snd the medium heigh during during the simulation. At the end of this proyect we will make an analiticla study for determine the best learning political. Also, the convergence of the algorithm is analyzed to ensure that the numbers of generations reach the maximum point of learning. In the results obtained, the agents don't emulates the human's biomechanics. On the other hand, we will conclued that policies where different rewards are combined obtain better results

***Index Terms*—inteligencia artificial, aprendizaje por refuerzo, algoritmo genético, pygame.**

I. INTRODUCCIÓN

En el presente trabajo final de máster se va a exponer el proceso de construcción de un agente bípedo virtual, con la ayuda de librerías utilizadas para el desarrollo de videojuegos para evaluar si es posible que este aprenda a caminar desde cero mediante el uso de algoritmos de inteligencia artificial. Para el desarrollo de este trabajo se han utilizado algoritmos de aprendizaje automático. Se ha combinado una red neuronal, con un algoritmo genético y aprendizaje por refuerzo que ayudará al agente bípedo a tomar las mejores decisiones premiando sus acciones de forma positiva o negativa.

A continuación, se definirán los conceptos mencionados anteriormente y se expondrán como se han utilizado para llevar a cabo este trabajo.

I-A. Machine learning

El aprendizaje automático es una rama de la inteligencia artificial que brinda a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia sin estar programados explícitamente [5] para la realización de una tarea. El proceso de aprendizaje comienza con observaciones o datos, como ejemplos, experiencia directa o instrucciones, para buscar patrones y tomar mejores decisiones en el futuro.

Dentro del aprendizaje automático existen distintos tipos de algoritmos que se diferencian entre sí:

- **Aprendizaje supervisado**: A partir de unos datos etiquetados, es decir datos ya clasificados previamente, el algoritmo creará un modelo de predicción e intentará clasificar datos nunca vistos por el. Algunos de los algoritmos más utilizados por la familia de aprendizaje supervisado son la regresión logística,los árboles de decisión, las maquinas de vectores de soporte(SVM) o el método Naive Bayes [10].
- **Aprendizaje no supervisado**: En este caso los datos no tienen clase asignada, por lo que los algoritmos no conocerán a que grupo pertenecen cada uno de los datos. Por tanto, la familia de algoritmos de aprendizaje no supervisado intentará obtener patrones y crear un modelo de clasificación. Los algoritmos de clustering son unos de los más utilizados en esta rama [8].
- **Aprendizaje por refuerzo**:Estos algoritmos se caracterizan por una discretización del espacio donde se realiza el aprendizaje y se le recompensará según el estado en el que el algoritmo decida ir. El objetivo es que el individuo aprenda a actuar de manera que maximice la recompensa obtenida [6].

En este caso, el aprendizaje por refuerzo nos permitirá establecer diferentes políticas de aprendizaje para favorecer que el bípedo aprenda diferentes comportamientos. No se usará el aprendizaje por refuerzo como algoritmo principal ya que solo se usa en entornos discretos, y en este caso el entorno es totalmente continuo. Se usará como complemento al algoritmo genético, en los casos que se decida que el individuo deba de actuar de manera distinta según en el estado.

I-B. Algoritmos evolutivos

Los algoritmos evolutivos son métodos y técnicas de optimización de soluciones basados en los postulados de la evolución biológica que han demostrado ser muy efectivos en la optimización de procesos no lineales [15, 17]. Este conjunto de métodos consiste en considerar un conjunto de individuos como una generación, que a lo largo del tiempo, es decir con el paso de las generaciones, los individuos serán más aptos para resolver el problema planteado. Las características de cada uno de los individuos harán que tengan una distinta adaptación al medio, lo que hará que tengan distintas posibilidades de reproducirse. Cada uno de los individuos están compuestos por genes, los cuales determinarán el comportamiento de cada uno de ellos en el medio. La adaptación al medio determinará las posibilidades del individuo para pasar a la siguiente generación, tal como pasa en la evolución biológica.

En el proceso evolutivo los individuos sufren mutaciones, es decir alteraciones genéticas que modifican su comportamiento. Las mutaciones sirven para que los individuos vayan cambiando entre las generaciones y así poder explorar otras vías de búsqueda de rendimiento máximo.

El cruce es una operación donde se intercambian los genes entre individuos. El fin del cruce entre individuos es el mismo que las mutaciones, encontrar vías alternativas que nos puedan dar una mejor solución.

II. Estado del arte

El uso del aprendizaje por refuerzo y los algoritmos evolutivos se ha aplicado en distintos campos como diagnósticos médicos [16], coches autónomos [2] o incluso en clasificación de secuencias de ADN [9]. Pero sin duda el campo de los videojuegos es donde más abunda.

Se expondrán los ejemplos que están relacionados con el caso práctico que se quiere resolver en el presente trabajo en los que se usan algoritmos genéticos y reinforcement learning.

Un ejemplo donde usa la misma estructura que en el presente trabajo es el aprendizaje de una red a jugar al Flappy Bird [4]. El objetivo de este proyecto es aprender la mejor política para que el individuo pueda pasar todos los obstáculos del juego. Como se puede ver hay un grupo de individuos que intentarán pasar los obstáculos y a medida que van pasando las generaciones los individuos son más óptimos y pasan todos los obstáculos del juego tal y como podemos observar en la Figura 1.

Figura 1. IA donde los individuos intentan superar los obstáculos del juego Flappy Bird

Otro ejemplo, que va estrechamente ligado con el anterior es el proyecto *Deep Learning Cars* [1]. Es una estructura muy similar a la anterior, donde el objetivo es que los coches consigan acabar un circuito sin chocar con los laterales. En la figura 2 podemos ver los coches y las marcas de los sensores que tienen cada uno de ellos.

En el 2017 *Google* creó una inteligencia artificial, donde un bípedo aprendía a caminar por distintos entornos. Los individuos que aprenden a caminar tienen unos sensores virtuales que transmitían información del entorno al agente bípedo [7]. A cada uno de los individuos se le dio un punto de partida y una meta, y a partir de esos inputs, aprendieron la mejor política de decisiones para superar los obstáculos y llegar al punto de meta. Como se puede observar en la figura 3, el bípedo creado por Google es muy realista y elaborado, en cuanto movimientos y estructura, lo que ha permitido que los resultados se han obtenido sean cercanos a la forma de moverse de un bípedo.

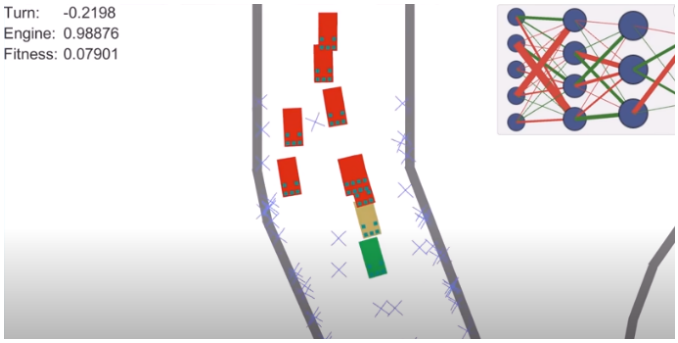

Figura 2. IA donde los coches intentarán superar todo el circuito sin chocarse

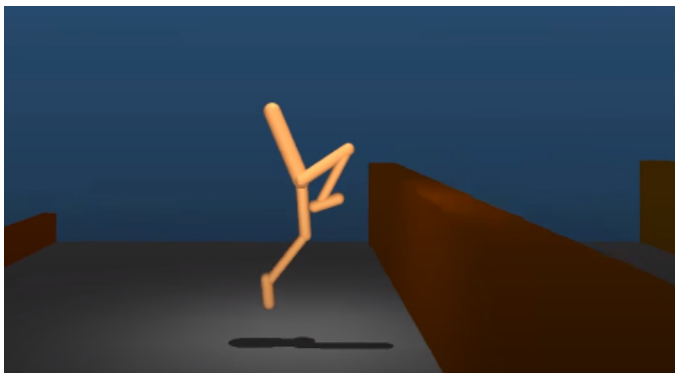

Figura 3. IA de Google donde enseña a un bípedo a caminar y a superar los obstáculos

III. Entorno de prueba

Como se ha explicado en la introducción, los algoritmos genéticos se basan en la evolución de unos individuos a través de las generaciones hasta que estos sean aptos para resolver el problema inicialmente planteado. En el presente caso los individuos que irán evolucionando serán redes neuronales que modificaran sus parámetros en cada generación.

Para poder ver el comportamiento de cada uno de los individuos, y así poder evaluarlo, se ha creado un agente bípedo 2d que podrá interactuar con la red neuronal, permitiendo que el individuo se pueda mover en base a los parámetros de la red.

Para la construcción del agente bípedo, y el entorno donde se moverá, se ha utilizado la librería Pymunk, que permite crear un cuerpo físico en 2D y poder controlar sus funciones móviles. También se ha utilizado la librería Pygame para poder visualizar el entorno con el agente bípedo.

Se ha generado un entorno donde el agente podrá moverse y así poder analizar su comportamiento durante el proceso evolutivo. El entorno se compone de un suelo rectilíneo, para intentar facilitar el aprendizaje del agente. El suelo tendrá un coeficiente de fricción de 1, es decir, tendrá la máxima fricción posible, para evitar posibles deslizamientos de los cuerpos sobre el suelo a la hora de moverse, cosa que dificultaría el aprendizaje. El propio entorno está configurado para que tenga una gravedad como la terrestre y así simular el comportamiento de un cuerpo real.

El agente que simulará el comportamiento de cada individuo esta compuesto por un chasis central, que simula la masa del tronco superior de un bípedo y dos piernas, unidas por una articulación central, la cual podríamos equiparar a la cadera. Dichas piernas estarán formadas por dos objetos Pymunk, los que representarán la parte de la pierna sostenida por el fémur y la parte de la pierna sostenida por la tibia y peroné, unidos por una articulación que representará la rodilla.

Las articulaciones del agente están compuestas con un motor interno, el cual les permitirá que las articulaciones giren hacia un sentido u otro.

Cada uno de los cuerpos que forman el agente tendrán un filtro que hará que los cuerpos no puedan colisionar entre sí, permitiendo movimientos típicos de seres bípedos, como cruzar las piernas o agacharse. Puede ser que la red neuronal aprenda a caminar de muchas maneras distintas a la que tiene el ser humano. Los cuerpos que forman las piernas se les asignará un coeficiente de fricción de 1, para evitar deslizamientos con el suelo.

Las dimensiones de las piernas del agente bípedo han intentado emular a la estructura humana de la pierna en cuanto dimensiones y masa, de tal manera que el 45 % del tamaño de la pierna estará representado en el segmento superior y el 55 % restante en el segmento inferior. En cuanto a la masa se ha querido emular la estructura de la pierna humana dándole más peso a la parte superior de la pierna.

Se ha asignado un color a cada parte del cuerpo para poder distinguirlas cuando se esté visualizando la simulación.

En la figura 4 del presente artículo se puede ver como quedará el agente implementado, sobre cual se realizará el aprendizaje.

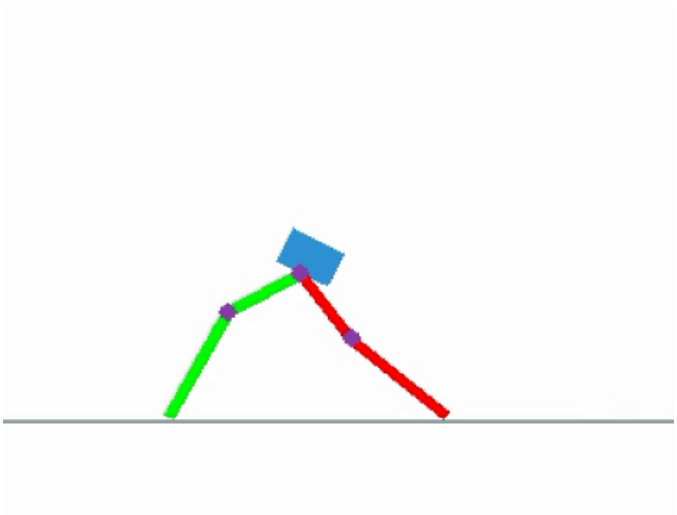

Figura 4. Agente bípedo en el cual se simulará el comportamiento de la red

IV. Metodología

El procedimiento experimental ha consistido en hacer uso de algoritmos genéticos y aprendizaje por refuerzo, que mediante redes neuronales básicas irán seleccionando a los agentes que mejor se vayan adaptando al medio durante el proceso de aprendizaje. Se ha utilizado un algoritmo probabilístico

para determinar que individuos se reproducirán a la siguiente generación, dando más probabilidad de reproducirse a la siguiente generación a los individuos que hayan obtenido mayor recompensa.

La recompensa de los individuos estará determinada por la política de aprendizaje, la cual evalúa el comportamiento del individuo en el medio. La política elegida será importante en el comportamiento de los agentes finales, ya que esta irá guiando a los individuos durante el aprendizaje.

La primera generación se inicializará con individuos con genes aleatorios, por lo que su comportamiento no estará condicionado bajo ningún criterio. A partir de la primera generación los agentes serán evaluados por la política de aprendizaje escogida.

La reproducción probabilística definirá la generación posterior en función de los individuos de la presente generación. Se generará una función que determinará la probabilidad de reproducción de cada uno de los agentes en función de como haya evaluado su comportamiento la política de aprendizaje.

Una vez el individuo consiga reproducirse, sufrirá una mutación en sus genes en forma de ruido, es decir, sus parámetros se modificarán mediante una distribución entre -0.1 y 0.1. Como se ha dicho anteriormente las mutaciones harán que el comportamiento del individuo cambie y así poder explorar otras vías de conseguir el objetivo. En el presente trabajo las mutaciones marcarán el cambio de comportamiento de los individuos a medida que vayan pasando las generaciones, por lo que si el ruido no existiese el comportamiento de los individuos no sería cambiante y por lo tanto el aprendizaje no sería posible.

Se ha decidido añadir a cada nueva generación un pequeño número de individuos que se inicializan de manera aleatoria. Con esto se intentará que la aleatoriedad pueda llevar a encontrar otras vías de resolver el problema de lo que esta haciendo el grueso de la población.

En el algoritmo 1 del presente trabajo se puede ver esquematizado el flujo que sigue el código.

Algorithm 1: Diagrama de flujo

```
Result: Individuos de la última generación del
        algoritmo
Inicialización de la primera generación;
while Generaciones ≤ 100 do
    if Primera generación then
        Nueva generación = Método Rolutte Wheel();
    else
        Nueva generación = Método Rolutte Wheel();
        Ruido =>Nueva generación;
        Agentes aleatorios =>Nueva generación;
    end
    Recompensas obtenidas = Simulación(Nueva
     generación);
    Población = Nueva Generación;
end
```

IV-A. Modelo de aprendizaje

Para el desarrollo del presente trabajo, se ha puesto como requerimiento que la red neuronal sea lo más básica posible. Por lo que se ha optado que sea una red con 4 inputs, 4 outpus y una sola capa intermedia de 4 neuronas. Los inputs son el estado presente de las articulaciones y los outputs es la acción que harán cada una de ellas. En la figura 5 se puede observar como se estructura la red neuronal que se utilizará en el algoritmo.

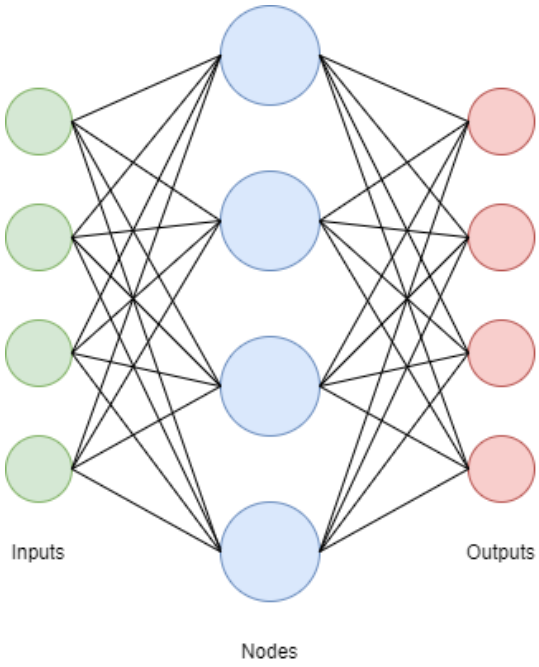

Figura 5. Diagrama de la estructura neuronal

Las neuronas de la red son de tipo sigmoide, que son el tipo de neuronas más presentes para este tipo de aprendizaje.

La neurona sigmoide se comporta de manera similar al perceptrón, pero su función de activación hace que la salida no sea binaria, sino que su rango de salida sea continuo. Las figuras 6 y 7 del presente trabajo muestran como se diferencian las salidas de la neurona perceptrón y la neurona sigmoide.

La función de activación del perceptrón es la suma ponderada de las salidas anteriores con los pesos que se le asignan a estas más el sesgo. En cambio la neurona sigmoide tiene una función de activación exponencial, gracias a esto podremos evitar que un pequeño cambio en los parámetros de la red haga que una neurona pase de tener una salida 0 a 1 o viceversa y así optimizar el aprendizaje [11].

IV-B. Simulación en cada individuo

En este apartado se describirá como se estructura la simulación de cada uno de los individuos. De esta simulación se obtendrá la recompensa obtenida por cada uno de ellos. En el algoritmo 2 del presente trabajo se plasma el flujo de la simulación.

Para simular el comportamiento del individuo, deberemos conectar la red neuronal al agente bípedo, que podrá moverse

gracias al feedback mutuo entre ellos. Gracias a esta simulación podremos evaluar el rendimiento de cada individuo, mediante la política de aprendizaje, y así determinar la probabilidad de reproducción para la siguiente generación.

Una vez conectada la red al agente, se inicializará la parte física del entorno y entrará en simulación. La simulación la podemos definir como un bucle el cual solo podrá finalizar bajo estas condiciones:

- Cuando el chasis del agente bípedo toque el suelo.
- Cuando la simulación supera un máximo de iteraciones, para evitar que el agente se quede parado.

Algorithm 2: Diagrama de flujo

```
Result: Valor fitness de la simulación
Piernas =>Inicialización física del agente 2D;
Valor fitness = 0;
while No cae al suelo OR No se acaba el deadline do
    Acción articulaciones = FeedForward(estado
     articulaciones);
    if Si cae o suelo OR se acaba deadline temporal
     then
        Acaba simulación;
        Valor fitness = Fitness function(altura
         media,distancia,velocidad);
    else
        Sigue simulación;
    end
    Actualizar ( distancia, velocidad y altura media);
    Actualizar (tiempo de simulación);
end
```

V. Experimentación

En este apartado del artículo se describirán los experimentos realizados para intentar validar la hipótesis de que los individuos irán mejorando su rendimiento hasta conseguir el fin planteado, que es aprender a caminar o desplazarse de la manera más erguida posible.

Antes de describir cada uno de los experimentos realizados cabe recordar como se ha dimensionado el algoritmo genético y las redes neuronales. Se ha optado por una población de 150 individuos y un proceso evolutivo de 100 generaciones, donde los individuos serán redes neuronales de 4 nodos y una sola capa intermedia. Las generaciones y el número de individuos que hay en cada una de ellas, se han visto limitadas por la capacidad de computación que se disponía para realizar el trabajo.

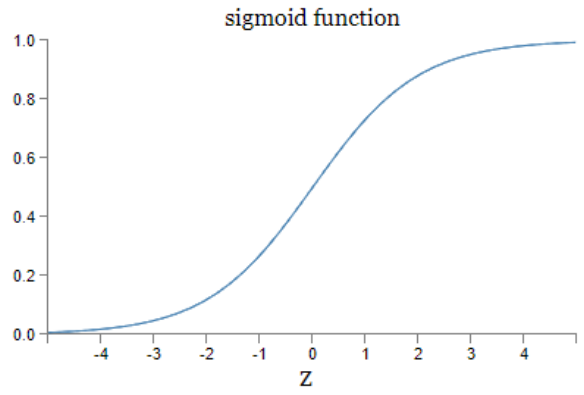

Figura 6. Salida de la neurona sigmoide

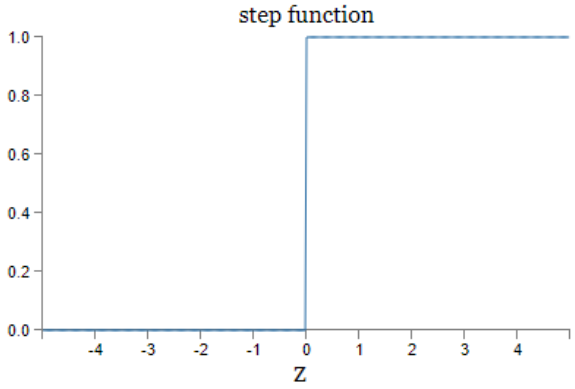

Figura 7. Salida binaria de la neurona perceptrón

Se han comparado 10 políticas de aprendizaje distintas, las cuales puntúan según la velocidad de desplazamiento, la distancia recorrida o si se realiza de forma erguida. Se ha realizado una recopilación de datos para cada una de las políticas, con el objetivo de realizar un análisis estadístico, y concluir, que políticas han obtenido mejores resultados, y si el aprendizaje converge.

La primera recopilación de datos consistirá en guardar el mejor individuo de la última generación de cada proceso evolutivo para cada una de las políticas. Este proceso se repetirá 30 veces para poder tener una muestra significativa. se ha elegido N = 30, ya que la distribución de algunos estimadores, cuando N es igual a 30, hay poca diferencia respecto a una distribución asintótica cuando N tiende a infinito [**?**]. Este experimento nos permitirá averiguar que políticas han obtenido mejores resultados.

Se hará otro experimento para evaluar la convergencia del aprendizaje en cada política. Se guardará el mejor individuo de cada generación durante todo el proceso evolutivo, para evaluar la distancia recorrida durante el aprendizaje. Este proceso de recopilación se hará 30 veces para tener una muestra significativa. El objetivo de este experimento consiste en cerciorarnos que el número de generaciones es suficiente, y no se finaliza el aprendizaje antes de tiempo.

V-A. Políticas de aprendizaje utilizadas

En este apartado se expondrán las 10 políticas de aprendizaje que evaluaremos. Todas y cada una de las pruebas realizadas se han hecho en el escenario descrito en la tercera sección del presente trabajo.

Las políticas que puntuarán el comportamiento de los agentes mediante la distancia que camina, la velocidad en la que lo realiza y si lo hace de manera erguida. No solo se ha tenido en

cuenta la distancia como modo de recompensar a un agente, ya que podría darse el caso de que los individuos aprendan a desplazarse con el torso muy bajo o que se desplacen muy lentamente, tal como se puede observar en la figura 8 del presente trabajo.

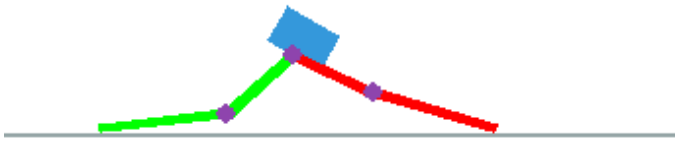

Figura 8. Agente aprende a desplazarse con el torso muy bajo

Las 10 políticas de aprendizaje que recompensarán a los individuos son las siguientes:

- **Distancia**: La recompensa será la distancia que ha avanzado el agente bípedo. Se contabilizará la distancia que ha recorrido el chasis del agente bípedo.
- **Distancia al cuadrado**: Esto hará que la función premie con más severidad a los que hayan recorrido más distancia.
- **Velocidad**: Premiará a los individuos más veloces. Se ha conseguido dividiendo la distancia por las iteraciones que ha necesitado.
- **Suma de distancias de todos los cuerpos**: Es otra manera de premiar la distancia recorrida, pero teniendo en cuenta los avances de las dos piernas.
- **Altura media 1 y distancia al cuadrado 10**: Aquí tenemos la primera política donde hay más de una recompensa. El cambio de recompensa lo determinará en la generación en la que se encuentra el algoritmo. Cada 10 iteraciones se premiará a los individuos que hayan tenido más altura media durante el recorrido, así intentar que los individuos tengan un comportamiento más erguido.
- **Altura media 1 y distancia al cuadrado 5**: En esta política se combinarán las mismas formas de recompensar al individuo, pero en este caso habrá el doble de iteraciones donde se premie la altura media del recorrido, con el fin de acentuar la forma erguida de caminar.
- **Velocidad 1 y distancia al cuadrado 10**: En este caso se combina premiar la distancia y la velocidad, donde cada 10 iteraciones se premiará a los individuos más veloces. Con esto se intentará que los individuos mejoren la velocidad.
- **Velocidad 1 y distancia al cuadrado 5**: Se combinará las mismas formas de recompensar que en la política anterior, en este caso se premiará el doble de veces la velocidad, intentando que los individuos sean más rápidos.
- **Altura media 1 y punto más lejano 10**: Se vuelven a combinar recompensas de distancia y altura media, pero se cambiará la recompensa de distancia por la suma de distancia de todos los cuerpos.
- **Altura media 1 y punto más lejano 5**: La misma forma de evaluar que en la anterior política, pero incrementando al doble las ocasiones que evalúa por altura media.

VI. Resultados

En esta sección se mostrarán los resultados obtenidos de los experimentos expuestos en el apartado anterior. Con los resultados obtenidos se intentará probar que el algoritmo no necesitaba más generaciones para llegar a un máximo y demostrar cual, o cuales, políticas de recompensas han sido las más beneficiosas para el algoritmo.

VI-A. Experimento A

El experimento consiste en la recopilación de 30 procesos de aprendizaje, para cada una de las políticas de aprendizaje. Una vez recopilados los procesos de aprendizajes se hará una gráfica con la media de las 30 muestras del mejor individuo de cada generación. El objetivo de esta gráfica será visualizar el aprendizaje del individuo a medida que vayan pasando las generaciones.

Recordamos que el algoritmo trabaja con un proceso de aprendizaje de 100 generaciones, donde cada una de ellas esta compuesta de 150 agentes.

Se ha hecho la prueba con cada una de las políticas de aprendizaje expuestas anteriormente, los resultados han sido muy parecidos en cada una de las políticas. Se mostrarán el resultado de tres políticas distintas para verificar, que el comportamiento del aprendizaje tiene el mismo patrón en cada una de ellas.

En este caso se mostrará tres políticas de aprendizajes distintas, y compararemos como se han comportado los agentes mediante las recompensas asignadas.

- Distancia al cuadrado, ver figura 9.

Figura 9. Evolución del aprendizaje de la política que premia la distancia recorrida

- Combinación entre distancia y altura media, ver figura 10.
- Suma de cuerpos del objeto, ver figura 11.

El resultado de este experimento muestra como hay un aprendizaje muy marcado en las primeras generaciones, pero una estabilización muy temprana. Podemos apreciar este comportamiento en las tres gráficas adjuntadas al presente trabajo,

Figura 10. Evolución del aprendizaje de la política que premia la distancia y la altura media

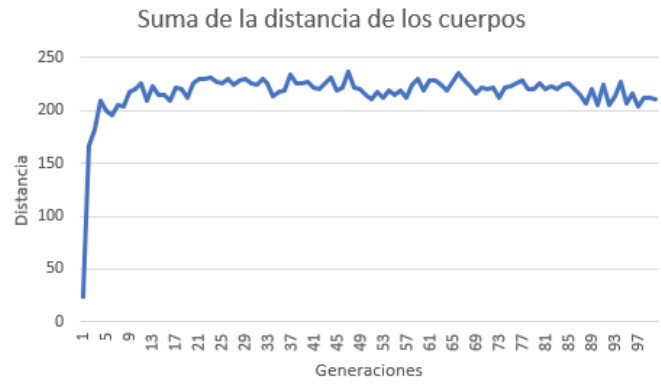

Figura 11. Evolución del aprendizaje de la política que premia la distancia recorrida de todos los cuerpos del agente

donde la primera generación, al tener unos genes aleatorios, recorre muy poca distancia, pero su rendimiento crece de manera pronunciada. Este crecimiento para de manera repentina aproximadamente al llegar a la décima generación. Este comportamiento es un patrón en las tres políticas expuestas en el experimento.

Podemos concluir de este experimento que el aprendizaje converge antes de que este se acabe, asegurando así que hemos obtenido el máximo rendimiento posible al finalizar la ultima generación. Cabe añadir que vemos como en todos los casos el aprendizaje máximo se alcanza muy rápidamente, aproximadamente al 10 % del aprendizaje. Esto nos puede indicar que las redes neuronales han tenido una capacidad de aprendizaje limitado.

VI-B. Experimento B

Este experimento pretende concluir que política es la más beneficiosa para que el agente bípedo se desplace de manera más eficiente.

La primera parte del experimento consiste en la recopilación de datos del proceso evolutivo. El dato para determinar la eficiencia del agente es la distancia recorrida por el mejor individuo de la última generación, después de 30 procesos evolutivos. Este dato se recogen 30 muestras para cada política. Una vez recogidos los datos para cada una de las políticas se realiza un contraste de hipótesis para determinar, en caso de que así fuera, cual es la mejor política.

El primer test que se plantea es el test Anova, también conocida como test de análisis factorial. Fue desarrollado por Fisher en 1930 [12], constituye la herramienta básica para el estudio del efecto de uno o más factores sobre la media de una variable continua. Esta técnica puede generalizarse también para estudiar los posibles efectos de los factores sobre la varianza de una variable.

Para que el test se pueda aplicar para todas las variables aleatorias deben tener distribución normal y debe cumplirse la homogeneidad de la varianza. Para verificar que todas las variables, en este caso las políticas, cumplen esta normalidad se realizará el test de Shapiro-Wilk [14], el cual es uno de los más utilizados para comprobar la normalidad de una variable. La variable cumplirá la normalidad siempre que el estadístico W arroje un p-valor mayor que $0{,}05$.

Como podemos ver en el cuadro I ninguna de las políticas cumplen la condición de normalidad, por lo que debemos descartar el test Anova y buscar otras alternativas para contrastar la hipótesis de que todas las políticas son iguales.

Cuadro I
RESULTADO DEL TEST SHAPIRO-WILK.

Política	p**-valor**
distancia	$8{,}16\text{e}-06$
distancia al cuadrado	$0{,}008602$
Velocidad	$1{,}932\text{e}-07$
Suma de la distancia de los cuerpos	$0{,}002216$
altura 1 distancia al cuadrado 10	$0{,}01197$
altura 1 distancia al cuadrado 5	$0{,}0224$
velocidad 1 distancia al cuadrado 10	$0{,}008261$
velocidad 1 distancia al cuadrado 5	$0{,}01262$
altura 1 Suma de la distancia de los cuerpos 10	$0{,}0007361$
altura 1 Suma de la distancia de los cuerpos 5	$0{,}01338$

Se plantea el test de Kruskal-Wallis [13], el cual es una alternativa no paramétrica al test Anova. A diferencia del test Anova en el que se comparan medias, el test de Kruskal-Wallis contrasta si las diferentes muestras están equidistribuidas y que por lo tanto pertenecen a una misma distribución. Bajo ciertas simplificaciones puede considerarse que el test de Kruskal-Wallis compara las medianas.

En tal caso se aplica el test de Kruskal-Wallis. Es estadístico usado tiene una distribución χ^2_9 y el valor obtenido es $80{,}465$ por lo que su p-valor es $1{,}306 \cdot 10^{-13}$ por lo que se descarta la hipótesis nula que provienen de la misma población y asumimos que provienen de poblaciones distintas.

Para determinar diferencias significativas entre las diferentes políticas se aplica el test por parejas de rangos con signo, el cual calcula las comparaciones por pares entre distintos grupos, valorando sus medianas. Estos grupos que se formarán nos ayudarán a clasificar las variables, con ayuda de un diagrama de cajas creado donde muestra la mediana de cada uno de los grupos.

El resultado es una matriz de semejanza que compara parejas de variables y calcula el p-valor a partir del estadístico W^+ de cada pareja, el cual si es superior a $0{,}15$, asumiremos

que pertenecen al mismo grupo. Esta matriz de semejanza esta plasmada en la tabla III del presente trabajo.

A partir de esta matriz se obtendrán los grupos donde clasificará a las políticas en función de las distancias obtenidas, plasmados en el cuadro II.

Cuadro II
GRUPOS FORMADOS POR EL TEST DE WILCOXON.

Política	**Grupos**
distancia	d
distancia al cuadrado	abcd
Velocidad	e
Suma de la distancia de los cuerpos	bcd
altura 1 distancia al cuadrado 10	a
altura 1 distancia al cuadrado 5	a
velocidad 1 distancia al cuadrado 10	a
velocidad 1 distancia al cuadrado 5	ab
altura 1 Suma de la distancia de los cuerpos 10	cd
altura 1 Suma de la distancia de los cuerpos 5	bcd

Una vez obtenido los resultados de los grupos formados. Se ha realizado un diagrama de cajas y bigotes de la mediana de la distancia recorrida para cada una de las políticas que se han probado sobre el agente bípedo. Este diagrama de cajas y bigotes se muestra en la figura 12

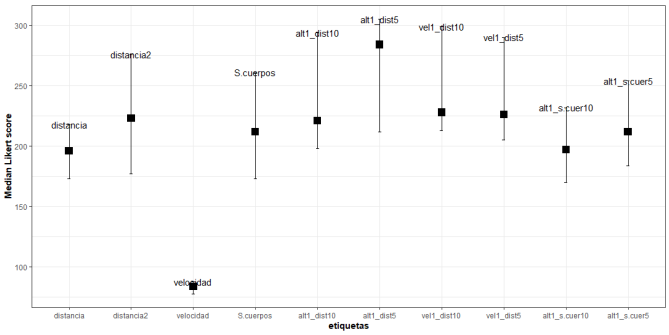

Figura 12. Diagrama de cajas y bigotes de las políticas estudiadas.

Gracias a los grupos formados por el test de Wilcoxon, que agrupa a las distintas políticas por la distribución de su distancia recorrida, y al diagrama de cajas y bigotes de la mediana de las distancias, se podrá concluir que políticas o conjunto de políticas han sido las más beneficiosas para el aprendizaje.

Si analizamos los grupos formados por el test vemos que las políticas que forman el grupo A son las que han conseguido unos resultados más favorables, en cambio el grupo E, donde solo esta la política de la velocidad, se sitúa como el grupo que ha obtenido el peor resultado.

Si intentamos obtener las políticas que han sido más beneficiosas, deberemos mirar el grupo A y ver si las políticas que forman el grupo están incluidas otros grupos, que han obtenido peores resultados. Vemos que las únicas dos políticas que permanecen solamente en el grupo A son las políticas que combinan la distancia con la altura media del recorrido, las cuales consideraremos que han sido las más beneficiosas para el aprendizaje.

VII. CONCLUSIONES

En el presente trabajo los agentes aprenden a desplazarse de forma erguida.

En base a los datos obtenidos, el aprendizaje por refuerzo junto con un algoritmo genético logra aprender a caminar. Las políticas donde se alterna el refuerzo obtienen mejores resultados. En concreto, la política que obtiene mejores resultados es la que combina premiar 5 generaciones la distancia recorrida y premiar 1 generación la altura del cuerpo.

A continuación se presenta una discusión y posibles trabajos futuros.

- Las políticas de aprendizaje se han basado en premiar a los individuos en función de la distancia recorrida, la velocidad durante el trayecto y la altura media. En cambio, no se ha tenido en cuenta, en ninguna de las políticas, la postura de los individuos durante el aprendizaje, lo cual ayudaría al intentar mantener una postura más óptima para aprender a caminar.
- Una de las limitaciones para el trabajo, era que se partiera de unas redes neuronales básicas, cosa que ha dificultado el presente trabajo. La dimensionalidad de esta red ha sido muy limitada, lo que ha hecho que perdiera capacidad de aprendizaje. En cambio esto ha permitido que el coste del aprendizaje fuera menor.
- La red neuronal no guarda información sobre las anteriores iteraciones del movimiento. Eso hace que su memoria sobre los estados anteriores sea inexistente. Se podrían mejorar usando redes neuronales recurrentes, las cuales se retro-alimentan, permitiendo que la información persista durante el movimiento, por ejemplo, que pierna está avanzando y cual retrocediendo. Esto permitiría tener memoria de los estados anteriores en los que ha pasado el individuo. Por contrapartida, el uso de las redes neuronales recurrentes haría que el aprendizaje fuera más costoso.
- Aunque la construcción del agente bípedo virtual haya sido costosa, se aleja de una estructura física plenamente humana. El agente bípedo creado no tiene restricciones de movilidades en cada una de sus articulaciones, lo cual permite que el individuo pueda hacer movimientos poco apropiados para desplazarse . Por otra parte, los agentes bípedos han carecido de pies, que le pudieran dar un soporte extra para mantener el equilibrio.

REFERENCIAS

[1] S. Artz. (22 diciembre 2020) deep learning cars. https://arztsamuel.github.io/en/projects/unity/deepCars/deepCars.html, 2016.
[2] S. Bhutani. Deep learning en vehículos autónomos. https://www.sciencedirect.com/science/article/pii/S0167739X19303772?casa_token=MXE_hKk5aVsAAAAA:oHnZA2LEQMpz5ROAKsxbE4-Vgbv8M6pITqYnr30U2fcwgT7p_4CKlM0Q0YSOc-Vr23_azlSaF3M, 2016.
[3] M. J. G. Cebrian. Distribuciones muestrales. http://recursostic.educacion.es/descartes/web/materiales_didacticos/inferencia_estadistica/distrib_muestrales.htm, 2016.
[4] K. Chen. Deep reinforcement learning for flappy bird. http://cs229.stanford.edu/proj2015/362_report.pdf, 2015.
[5] E.S.Team. What is machine learning? a definition. https://expertsystem.com/machine-learning-definition/, 2020.
[6] G.Hayes. Reinforcement learning. https://sitiobigdata.com/2019/12/31/reinforcement-learning-con-mario-bros-con-mario-bros-parte-1/, 2020.

Cuadro III
MATRIZ DE SEMEJANZA RESULTANTE AL TEST DE WILCOXON

Políticas	dist	$dist^2$	velocidad	S.cuerpos	alt^1dist^{10}	alt^1dist^5	vel^1dist^{10}	vel^1dist^5	$alt^1S.cuerpos^{10}$
$dist^2$	0.1462	-	-	-	-	-	-	-	-
velocidad	$1{,}1e^{-11}$	$6{,}3e^{-13}$	-	-	-	-	-	-	-
S.cuerpos	0.4147	0.4946	$1{,}3e^{-11}$	-	-	-	-	-	-
alt^1dist^{10}	0.0253	0.4761	$9{,}1e^{-13}$	0.1547	-	-	-	-	-
alt^1dist^5	0.0039	0.1681	$7{,}4e^{-13}$	0.0243	0.5718	-	-	-	-
vel^1dist^{10}	0.0088	0.3581	$1{,}7e^{-12}$	0.0850	0.7412	0.7191	-	-	-
vel^1dist^5	0.0253	0.6757	$1{,}7e^{-12}$	0.2418	0.8087	0.2861	0.5422	-	-
$alt^1S.cuerpos^{10}$	0.8660	0.2729	$1{,}5e^{-11}$	0.6757	0.0591	0.0169	0.0345	0.1058	-
$alt^1S.cuerpos^5$	0.4317	0.4946	$1{,}5e^{-11}$	0.8776	0.1547	0.0296	0.0850	0.2478	0.7412

[7] Google. (12 febero 2021) producing flexible behaviours in simulated environments. https://arztsamuel.github.io/en/projects/unity/deepCars/deepCars.html, 2016.

[8] L.González. Aprendizaje no supervisado. https://aprendeia.com/aprendizaje-no-supervisado-machine-learning/, 2020.

[9] I. P. M. J. S. M. R. L. Luis A. santamaría, Sarahí Zuñiga. Reconocimiento de genes en secuencias de adn por medio de imágenes. https://sci2s.ugr.es/caepia18/proceedings/docs/CAEPIA2018_paper_78.pdf, 2018.

[10] Mathworks. Técnica de machine learning para crear modelos predictivos a partir de datos de entrada y respuesta correctos. https://es.mathworks.com/discovery/supervised-learning.html, 2020.

[11] M. Nielsen. Neural network and deep learning. http://neuralnetworksanddeeplearning.com/chap1.html, 2019.

[12] Rstudio. (24 marzo 2021) anova con r. https://rpubs.com/Joaquin_AR/219148, 2016.

[13] Rstudio. (24 marzo 2021) kruskal-wallis con r. https://www.scientific-european-federation-osteopaths.org/wp-content/uploads/2019/01/ALGUNAS-PRUEBAS-NO-PARAM%C3%89TRICAS.pdf, 2016.

[14] Rstudio. (24 marzo 2021) saphiro-wilk con r. https://www.rpubs.com/F3rnando/507482, 2019.

[15] Wikipedia. (20 enero 2021) algoritmos genéticos. https://es.wikipedia.org/wiki/Algoritmo_genÃl'tico, 2020.

[16] H. Y. T. W. Zhuo Liu, Chenhui Yao. Aprendizaje por refuerzo profundo con su aplicación para la detección del cáncer. https://www.sciencedirect.com/science/article/pii/S0167739X19303772?casa_token=MXE_hKk5aVsAAAAA:oHnZA2LEQMpz5ROAKsxbE4-Vgbv8M6pITqYnr30U2fcwgT7p_4CKlM0Q0YSOc-Vr23_azlSaF3M, 2019.

[17] A. Álvarez Diaz, Marcos; Álvarez. predicción no-lineal de tipos de cambio. aplicación de un algoritmo genético. https://www.redalyc.org/pdf/969/96918123003.pdf, 2004.