Práctica 02

Algoritmos evolutivos y redes neuronales

Objetivo

Esta práctica consiste en la implementación, entrenamiento y validación de la una serie de soluciones encontradas por medio de un A*lgoritmo Evolutivo (AE)* utilizando redes neuronales. El objetivo concreto de esta práctica consiste en realizar una serie de "subprácticas":

2.1.- Rehacer la práctica 01 utilizando un algoritmo evolutivo. Para este caso, es suficiente con que el cilindro rojo esté inmóvil. NO tiene que moverse.

2.2.- Cambiar al escenario 5, "Avoid the block" y conseguir que el Robobo aprenda a acercarse al cilindro rojo, en unas condiciones iniciales concretas.

2.3.- En el escenario "Avoid the block", aprender a llegar al cilindro rojo. Durante la fase de validación se puede utilizar 1) únicamente un AE, o 2) una combinación de AE y AR, con la política encontrada en la práctica 1.

Herramientas

Para realizar la práctica 02, se hará uso de librerías y herramientas ampliamente utilizadas en la comunidad, tales como el entorno de trabajo de Gymnasium (https://gymnasium.farama.org/), la librería NEAT-Python (https://neat-python.readthedocs.io/en/latest/). No obstante, estas son las principales, y cualquier otra librería puede ser utilizada/necesaria para llevar a cabo la práctica.

Como en la práctica anterior, como robot, se utilizará el robot Robobo, en su versión en simulación (está pendiente la implementación de confirmación la implementación en el robot real):

- Página principal del proyecto Robobo: https://therobobоproject.com
- Simulador RoboboSim: https://github.com/mintforpeople/robobo-programming/wiki/Unity
- Librería "Robobo.py" para programar el Robobo: https://github.com/mintforpeople/robobo-programming/wiki/python-doc https://mintforpeople.github.io/robobo.py/
- Librería "Robobosim.py" para utilizar funcionalidades exclusivas del simulador: https://github.com/mintforpeople/robobo-programming/wiki/robobosimpy

Cada sub-práctica tiene a su vez, su propio escenario y características.

Prácticas

Un punto común a todas las subprácticas es que, como en la práctica 1, se deberá usar el entorno de trabajo creado con Gymnasium (en el entorno de la práctica 01 se puede modificar todo lo que se quiera, siempre y cuando se mantenga la estructura básica de los entornos de Gymnasium).

A su vez, este entorno de Gymnasium será utilizado por la librería Python-NEAT para realizar el proceso de aprendizaje de acercarse al cilindro.

Otro aspecto a destacar es la recomendación/obligación de utilizar los sensores y actuadores de Robobo para el estado de observaciones y estado de acciones. En la función de fitness, se puede utilizar información propia del simulador.

La velocidad del simulador se fija a x4

Detalles Práctica 2.1.

Descripción

Objetivo: El Robobo debe aprender a acercarse al cilindro rojo inmóvil.

El escenario de trabajo que se utilizará con RoboboSim será el que se observa en la Figura 1.

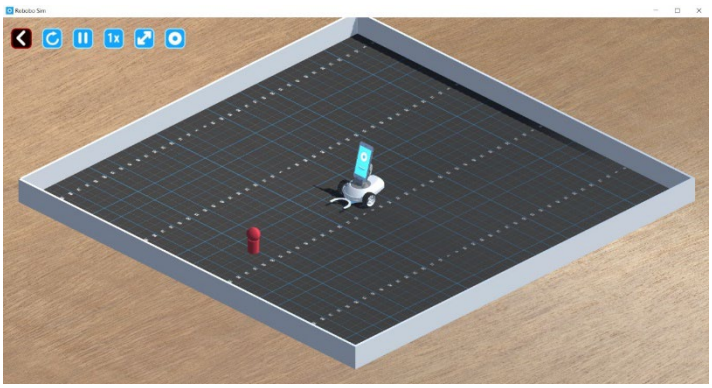

Figura 1. Escenario "cylinder" de RoboboSim

Entrenamiento

El entrenamiento, en este caso y en todos los que veremos en esta práctica, se desarrolla con el AE, por lo que se deberán de especificar los parámetros básicos del mismo, tal como el tamaño de la población, las generaciones totales, el número de pasos de tiempo que se evalúa cada individuo, la función de fitness, etc.
NOTA: Se sugiere emplear la física simplificada y aumentar la velocidad de simulación a x4 para acelerar el aprendizaje, y NO usar modelos de mundo aleatorios.

Representación de los resultados

Los resultados se presentarán por medio de gráficas, en donde se mostrará

Se presentarán las gráficas de:

- Aprendizaje.
- Configuración red neuronal.

Detalles Práctica 2.2

Descripción

Objetivo: El Robobo debe aprender a acercarse al cilindro rojo inmóvil.

El escenario de trabajo que se utilizará con RoboboSim será el que se observa en la Figura 2.

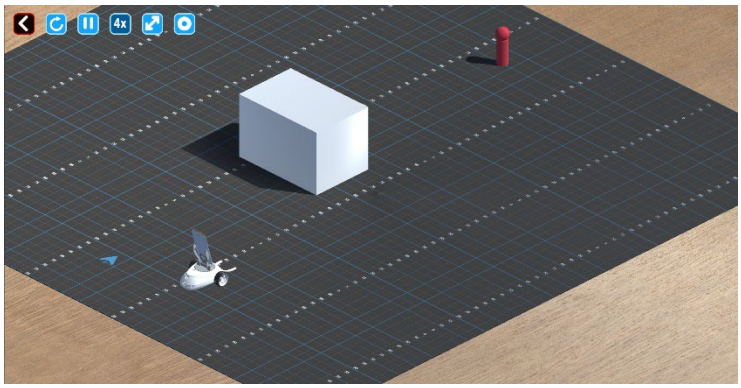

Figura 2. Escenario "Avoid the block" de RoboboSim

Como particularidad en esta práctica, el Robobo deberá de empezar en la posición:

{'x': -1000.0, 'y':39.0, 'z': -400.0}

Entrenamiento

El entrenamiento, se desarrolla con el AE, por lo que se deberán de especificar los parámetros básicos del mismo, tal como el tamaño de la población, las generaciones totales, el número de pasos de tiempo que se evalúa cada individuo, la función de fitness, etc.

NOTA: Se sugiere emplear la física simplificada y aumentar la velocidad de simulación a x4 para acelerar el aprendizaje, y NO usar modelos de mundo aleatorios.

Representación de los resultados

Los resultados se presentarán por medio de:

- *Script de validación o testeo*
  Nuevamente, La política aprendida, en este caso, individuo o genoma, deberá de poder cargarse en un nuevo programa o script, para verificar su correcto funcionamiento.
- *Gráficas*

Se presentarán las gráficas de:

- Aprendizaje.
- Configuración red neuronal.
- Especies obtenidas.
- Plano 2D solución.

Detalles Práctica 2.3

Objetivo: El Robobo debe aprender a acercarse al cilindro rojo inmóvil, con el sólido justo en medio.

El escenario de trabajo que se utilizará con RoboboSim será el que se observa en la Figura 3.

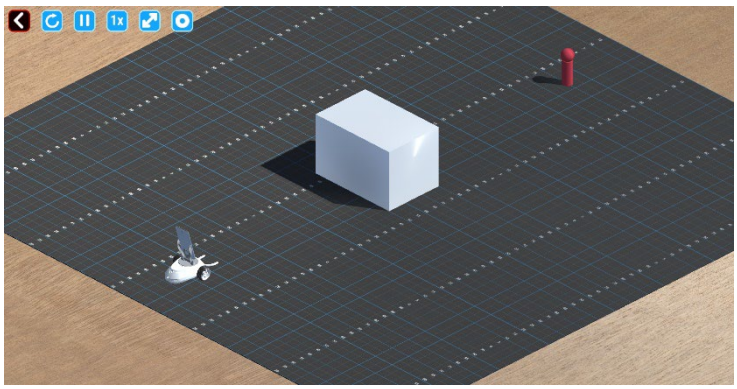

Figura 3. Escenario "Avoid the block" de RoboboSim

En este caso, no se modifica la posición inicial del Robobo.
Hay que tener en cuenta que la opción de AE + AR se reflejará en el caso de validación. El entrenamiento se ha de realizar por separado. Serán las políticas entrenadas lo que se valide.
En caso de optar por la opción de AE+AR, si se considera necesario, se puede hacer alguna modificación "a mano" para ajustar las condiciones del robot al pasar de una política a otra, pero el funcionamiento de las políticas aprendidas NO se puede modificar. Es decir, ambas políticas se tienen que ejecutar tal y como se aprendieron y sólo se permite una adaptación del robot en la transición de una a otra.

Entrenamiento

El entrenamiento, se desarrolla con el AE, por lo que se deberán de especificar los parámetros básicos del mismo, tal como el tamaño de la población, las generaciones totales, el número de pasos de tiempo que se evalúa cada individuo, la función de fitness, etc.
NOTA: Se sugiere emplear la física simplificada y aumentar la velocidad de simulación a x4 para acelerar el aprendizaje, y NO usar modelos de mundo aleatorios.

Representación de los resultados

Los resultados se presentarán por medio de:

- *Gráficas*
  Se presentarán las gráficas de:
  - Aprendizaje.
  - Configuración red neuronal.
  - Especies obtenidas.
  - Plano 2D solución.

Criterio de Evaluación:

- Hacer sub-práctica 2.1 (4 puntos)
- Hacer sub-práctica 2.2 (2 puntos)
- Hacer sub-práctica 2.3 (4 puntos)

Para cada práctica, se valorará proporcionalmente:

Definición del problema e implementación de los elementos básicos (70%):

  - Definición y complejidad de los espacios de estados, de acciones y función de fitness. Se valorará la función de fitness seleccionada y su justificación (10%).
  - Implementación del algoritmo de aprendizaje y calidad de la solución propuesta (60%):
    - **Cumplir con el objetivo/los objetivos establecidos en cada práctica (40%).**
    - Calidad y eficiencia de la solución obtenida. Rapidez del aprendizaje, robustez de la solución a la hora de aproximarse al objetivo, etc (20%)
- Presentación de la información (15%):
  - Presentación de las gráficas dadas por la librería NEAT:
    - Aprendizaje.
    - Forma red neuronal.
    - Especies.
  - Presentación en un plano 2D de las diferentes posiciones recorridas por el Robobo y situación de los objetos en el plano, en los casos de validación/testeo.
- Redacción y presentación de la memoria (15%)
  - Formato:
    - Nombres alumnos.
    - Texto justificado.

- Figuras con los pies de foto claros, explicativos y referenciados en el texto.
- Imágenes claras y legibles. Leyendas incluidas en las imágenes explicativas. El texto en las figuras se tiene que ver.

- Texto:
  - Explicación clara y precisa de las soluciones adoptadas, poniendo claramente las variables consideradas, y sus límites. Ejemplos:
    - Entradas de la red. Cuantas son, valores máximos y mínimos.
    - Función de fitness. Cálculo claro de la misma.
    - Evitar: "se aplican bonificaciones si se acerca al cilindro rojo y penalizaciones si se aleja" Falta indicar de qué tipo son y cuanto se aplica como bonificación/penalización.

Entrega:

Para que la práctica sea evaluable, se ha de cumplir:

1. En todas las prácticas, se deberá entregar la carpeta con el código fuente utilizado para entrenar y validar/testear la solución obtenida.
2. Tiene que haber un script, bien sea "test.py" bien "validación.py" para evaluar el/los comportamientos aprendidos.
3. Además, esta carpeta deberá de contener las gráficas e imágenes obtenidas, que deben de estar también incluidas en la memoria.
4. La ruta a los modelos obtenidos debe de ser una ruta absoluta, válida para cualquier equipo o asegurarse de que funciona en equipos diferentes.
5. No cumplir con los puntos anteriores puede suponer la invalidación de la práctica. Esto es, una calificación de 0 en la misma.
6.
7. La carpeta de entrega deberá de tener una ruta lo más corta posible, y los nombres no tendrán ni espacios en blanco ni acentos.

Viernes 7 de noviembre. A las 23:59. La práctica se deberá de entregar en un archivo ".zip" que contenga:

- Pequeña memoria de 5 páginas máximo con fuente a tamaño 12.
  - 2 páginas (aprox.) práctica 2.1
  - 1 página (aprox.) práctica 2.2
  - 2 páginas (aprox.) práctica 2.3
- Las figuras tendrán todas un pie de página y estar referenciadas en el texto.
- El código será entregado y deberá de ejecutarse sin problema.

Entregar esto se entregará a través de **Moodle** (no correo).