

Universidad Politécnica Salesiana

jarevalop1@est.ups.edu.ec

Jorge Arévalo

Profesor: Ing. Diego Quisi

Materia: Inteligencia Artificial

Universe

Concepto

La plataforma Universe “está compuesta por unos mil 'entornos' formados por avanzados videojuegos, como Minecraft o Grand Theft Auto V, que se utilizan para simular la vida real" (El País, 2016). En esta línea, esta plataforma permite que un agente de inteligencia artificial use una computadora de la misma manera en que lo hace un ser humano: observando los píxeles en pantalla y operando sobre un teclado o ratón virtual

Que se ha hecho con esta herramienta

Universe incluye los juegos Atari 2600 del Arcade Learning Environment. Estos entornos ahora se ejecutan de forma asíncrona dentro de la imagen de Docker quay.io/openai/universe.gym-core y permiten que el agente se conecte a través de la red, lo que significa que el agente debe manejar el retraso y las bajas velocidades de cuadros. Corriendo sobre una red local en la nube, generalmente vemos 60 cuadros por segundo, retrasos de observación de 20 ms y retrasos de acción de 10 ms; a través de Internet público, esto se reduce a 20 fotogramas por segundo, retrasos de observación de 80 ms y retrasos de acción de 30 ms.

Juegos

Juegos de Atari. - Universe incluye los juegos Atari 2600 del Arcade Learning Environment. Estos entornos ahora se ejecutan de forma asíncrona dentro de la imagen de Docker quay.io/openai/universe.gym-core y permiten que el agente se conecte a través de la red, lo que significa que el agente debe manejar el retraso y las bajas velocidades de cuadros.

Juegos flash. - Son penetrantes en Internet, generalmente presentan gráficos más ricos que Atari, pero aún son individualmente simples. Hemos examinado más de 30,000 hasta ahora, y estimamos que hay un orden de magnitud más.

```
import gym
import universe # register Universe environments into Gym

env = gym.make('flashgames.DuskDrive-v0') # any Universe [environment
ID](https://github.com/openai/universe/blob/master/universe/__init__.py#L297)
here
# If using docker-machine, replace "localhost" with your Docker IP
env.configure(remotes="vnc://localhost:5900+15900")
observation_n = env.reset()

while True:
    # agent which presses the Up arrow 60 times per second
    action_n = [(['KeyEvent', 'ArrowUp', True]) for _ in observation_n]
    observation_n, reward_n, done_n, info = env.step(action_n)
    env.render()
```

Puede mantener abierta su propia conexión VNC y ver jugar al agente, o incluso usar el teclado y el mouse junto al agente en modo cooperativo humano / agente.

Opinión

Debido a que los entornos se ejecutan como procesos de servidor, pueden ejecutarse en máquinas remotas, posiblemente dentro de un clúster o incluso a través de Internet público. Debe existir soporte para Windows.

Conclusiones

Proporciona tiempos de ejecución en un clúster Kubernetes bajo demanda, y que podemos usar para conectar un proceso de agente único a cientos de entornos simultáneos. Los agentes del universo deben lidiar con la suciedad del mundo real de la que los agentes tradicionales de RL están protegidos: los agentes deben ejecutarse en tiempo real y dar cuenta de la acción fluctuante y el retraso de observación.

GYM

Concepto

Gym es un juego de herramientas para desarrollar y comparar algoritmos de aprendizaje de refuerzo. Es compatible con los agentes docentes, todo caminando para jugar juegos como Apestar o Pinball.

Interfaz de código abierto para tareas de aprendizaje de refuerzo. La biblioteca del gym ofrece un conjunto de tareas de aprendizaje de refuerzo fácil de usar.

Que se ha hecho con esta herramienta

El aprendizaje por refuerzo (RL) es el subcampo del aprendizaje automático relacionado con la toma de decisiones y el control motor. Estudia cómo un agente puede aprender a alcanzar objetivos en un entorno complejo e incierto. Es emocionante por dos razones:

RL es muy general, abarca todos los problemas que implican tomar una secuencia de decisiones: por ejemplo, controlar los motores de un robot para que pueda correr y saltar, tomar decisiones comerciales como la gestión de precios e inventario, o jugar videojuegos y juegos de mesa.

Los algoritmos RL han comenzado a lograr buenos resultados en muchos entornos difíciles.

Los resultados de DeepMind en Atari, BRETT del grupo de Pieter Abbeel y AlphaGo utilizaron algoritmos de RL profundos que no hicieron demasiadas suposiciones sobre su entorno y, por lo tanto, se pueden aplicar en otros entornos.

Juegos

Atari: juega los videojuegos clásicos de Atari.

Juegos de mesa: juega Go en tableros de 9×9 y 19×19.

Ejemplo de un juego. Debemos instalar la librería y probar en Python.

```
import gym
env = gym.make("CartPole-v1")
```

```
observation = env.reset()
for _ in range(1000):
    env.render()
    action = env.action_space.sample() # your agent here (this takes random
actions)
    observation, reward, done, info = env.step(action)

    if done:
        observation = env.reset()
env.close()
```

Opinión

Es una librería que todavía esta vigente para desarrollar y comparar algoritmos de aprendizaje de refuerzo que apoya a los agentes de enseñanza, desde caminar hasta jugar juegos como Pongo o Go.

Conclusiones

Este kit de herramientas para crear algoritmos de aprendizaje es útil y fácil de aprender. Mejor que escribir desde cero a través de scikit learn y bueno para probar el rendimiento que la validación cruzada pura.