

Proyecto Final

Pac-Man

Alvaro Gómez Hermosillo Reyes
Fac. de Matemáticas - UADY
the_golden_mistery@hotmail.com

Andrea Eugenia Ortiz Pavón
Fac. de Matemáticas - UADY
andrea_piperhalliwell@hotmail.com

Rudy Carlos Sánchez Paz
Fac. de Matemáticas - UADY
charlyboy_2904@hotmail.com

December 15, 2015

Abstract

En este documento se busca representar los conceptos importantes visto a lo largo del curso de Inteligencia Artificial, abordando diferentes temas como agentes, su entorno y percepciones, el tipo de aprendizaje que puede tener nuestro agente, entre otros, aplicándolos a un juego de nuestra elección, el cual es Pac-man.

Pac-man es uno de los juegos más populares del mundo. De origen japonés, se lanzó públicamente el 22 de mayo de 1980 y fue desarrollado por Toru Iwatani para la empresa de juegos Namco. El protagonista del videojuego Pac-Man es un círculo amarillo al que le falta un sector, por lo que parece tener boca. Aparece en laberintos donde debe comer puntos pequeños (llamados "Pac-dots"), puntos mayores y otros premios con forma de frutas y otros objetos.

El objetivo del personaje es comer todos los puntos de la pantalla, momento en el que se pasa al siguiente nivel o pantalla (el juego tiene 256 niveles). Sin embargo, cuatro fantasmas o monstruos, Shadow (Blinky), Speedy (Pinky), Bashful (Inky) y Pokey (Clyde), recorren el laberinto para intentar capturar a Pac-Man; estos fantasmas son, respectivamente, de colores rojo, rosa, cian y naranja.

Para darle emoción al juego, se le equipó con un poco de inteligencia artificial, los fantasmas se agruparían y atacarían al jugador y también se dispersarían. Cada fantasma tiene su propia inteligencia artificial.

Hay cuatro puntos más grandes de lo normal situados cerca de las esquinas del laberinto llamados "Power Pellets", y que proporcionan a Pac-Man la habilidad temporal de comerse a los monstruos (todos ellos se vuelven azules mientras Pac-Man tiene esa habilidad). Después de haber sido tragados, los fantasmas se regeneran en "casa" (una caja situada en el centro del laberinto).

Este juego está disponible en más de 35 plataformas diferentes, y al ser tan conocido, se nos hizo interesante poder crear un agente en un mundo pequeño el cual fuera inteligente y aprendiera a medida que jugara más cantidades de partidas, y que con base a la experiencia poder tener más oportunidades de ganar en el juego.

0.1 Introducción

1.1. Objetivo del Proyecto

Con el presente proyecto buscamos lograr diseñar un agente capaz de jugar el juego de Pac-man en un ambiente de dificultad media con el propósito de poner en práctica los conceptos aprendidos en el curso. El agente tendrá un aprendizaje por refuerzo. La meta del agente es completar el nivel de la mejor manera posible teniendo por lo menos un fantasma que estará a la caza del Pac-man, con esto se tiene ambiente más complicado.

Buscamos que el agente se pueda desenvolver en su entorno logrando que mejore desempeño el cual será apoyado de los aprendizajes posibles de la Inteligencia Artificial, con el fin de que logre realizar acciones por sí mismo con el pasar de diferentes rondas del juego.

1.2. Antecedentes

A lo largo de la vida, la tecnología ha ido evolucionando y mejorando haciendo más practica las actividades de las personas, la Inteligencia Artificial ha servido como una base para lograr algunas de las cosas que tenemos hoy.

Los esfuerzos por reproducir algunas habilidades mentales humanas en máquinas y androides se remontan muy atrás en la historia. El mito del coloso de Rodas entre los griegos, las estatuas "parlantes" del medioevo, el androide de Von Kempelen que jugó al ajedrez con Napoleón, y el "motor analítico" de Charles Babbage que calculaba logaritmos, son sólo algunos de los ejemplos de este antiguo interés. Igualmente, la concepción de la inteligencia humana como un mecanismo no es reciente ni ha estado disociada de la psicología: Descartes, Hobbes, Leibniz, y el mismo Hume se refirieron a la mente humana como una forma de mecanismo.

La Inteligencia Artificial fue introducida a la comunidad científica en 1950 por el inglés Alan Turing en su artículo "Maquinaria Computacional e Inteligencia." A pesar de que la investigación sobre el diseño y las capacidades de las computadoras comenzaron algún tiempo antes, fue hasta que apareció el artículo de Turing que la idea de una máquina inteligente cautivó la atención de los científicos.

Dos de las contribuciones más importantes de Turing a la Inteligencia Artificial, fueron el diseño de la primera computadora capaz de jugar ajedrez y, más importante que esto, el establecimiento de la naturaleza simbólica de la computación.

0.2 Marco Teórico

2.1. Tema

Desarrollo de un agente que se desenvuelva en un entorno apoyado de la Inteligencia Artificial y todos los conceptos que rodean a esta.

2.2. Definiciones de interés

- Inteligencia Artificial: “IA... construcción de agentes que se comportan racionalmente (dados los recursos disponibles)” (Russell & (Russell & Norvig, 1995)
- Aprendizaje por refuerzo:
Suposición:
 - Los objetivos se pueden expresar a través de una función (reward) que asigna un valor numérico para cada acción que el agente ejecuta en cada estado.
 - Valores numéricos positivos informan al agente que la acción que acaba de ejecutar ha sido buena, mientras que valores negativos informan al agente que la acción que acaba de ejecutar ha sido mala.

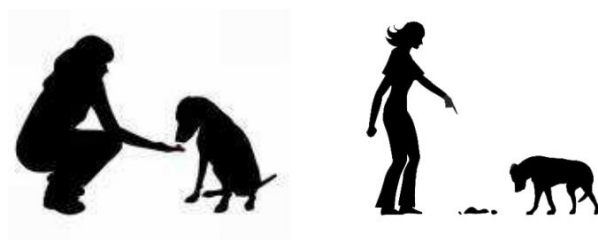


Figure 1: Recompensas o Castigos

2.3. Problema de Investigación

¿Cómo puede un agente autónomo, que percibe su entorno y actúa en ello, aprender a seleccionar las acciones óptimas que hay que ejecutar en cada estado para alcanzar sus objetivos? Ese es nuestro objetivo principal,

buscamos que del aprendizaje que va realizando nuestro agente por medio de castigos en puntuación al momento de fallar vaya analizando sus errores y los pueda ir mejorando poco a poco a través de todos los estados posibles por los que va pasando. Queremos añadirle a un agente la capacidad de poder decidir en sus acciones para lograr sus objetivos definidos, que se puede desempeñar libremente en su entorno y de la mejor manera posible, con esto lograr implementarle a nuestro agente una Inteligencia Artificial.

2.4. Justificación de la Investigación

Es muy común en nuestros días poder apreciar la Inteligencia Artificial en las cosas que nos rodea, aunque es claro que nos falta mucho camino por recorrer ya que las tecnologías no dejan de innovar y de actualizarse, es por eso que con este trabajo queremos aprender y enseñar un poco más de lo que es capaz de lograr la Inteligencia Artificial y de las herramientas de las que se apoya para poder llevar el trabajo que realiza en ciertos entornos, ejemplificaremos todo esto por medio del Juego de Pac-Man que sabemos que es un juego bastante conocido pero que como cualquier otro juego de vídeo se basa en la IA para su desarrollo. El objetivo es mostrar que tan rápido puede aprender en base a sus errores y los castigos que se le dan por fallar en su meta y por las victorias que se obtenga.

2.5. Hipótesis

La hipótesis de este proyecto consiste en demostrar la utilidad de la Inteligencia Artificial en los desarrollos de programas que nos rodea, su funcionamiento y su utilidad, esto apoyado del juego de Pac-Man que se encuentra en python y en la muestra de avance del aprendizaje que tiene este al momento de jugar.

2.6. Marco Teórico General

2.6.1. Desarrollo de la IA

Para mucha gente las máquinas inteligentes existen solamente en las novelas y en las películas de ciencia ficción. Pero su creación no parece estar tan lejos y para algunos expertos podría ser un error, incluso el peor error de la historia humana, según un reportaje elaborado por el diario 'The Independent' citando a los físicos Stephen Hawking, Stuart Russell, Max Tegmark y Frank Wilczek.

Los científicos están avanzando rápidamente en las investigaciones en lo referente al área de la Inteligencia Artificial. Los coches sin conductor, un ordenador capaz de ganar el juego intelectual 'Jeopardy!', y los asistentes

personales digitales Siri y Cortana son los principales hitos de una carrera armamentista establecida en torno al desarrollo de la Inteligencia Artificial y acelerada por unas inversiones sin precedentes.

Los beneficios potenciales son enormes, ya que todo lo que la civilización ofrece es un producto de la inteligencia humana. El éxito de la creación de Inteligencia Artificial sería el evento más grande de la historia de la civilización.

2.6.2. Importancia de la Inteligencia Artificial

Las computadoras son fundamentales hoy día en nuestras vidas afectando todos los aspectos de esta. La Inteligencia Artificial se crea con la implementación en las computadoras para realizar mecanismo de computación que utiliza programas fijos y contiene ciertas series de reglas que lo hacen funcionar. Esto permite a las computadoras a ser creadas en máquinas artificiales que desempeñan tareas monótonas, repetitivas y simples más eficiente y efectivas que un ser humano. La Inteligencia Artificial que busca el mejorar, aprender, entender y el razonamiento del comportamiento de las computadoras en situaciones complejas. El campo de la ciencia de Inteligencia Artificial está todavía en etapas de crecimiento comparadas con otras ramas de computadoras pero poco a poco el estudio del comportamiento humano dará paso para aplicar estos conocimientos a las computadoras y estas lograr de manera primitiva razonar sobre diferentes situaciones.

2.6.3. Amplitud de la Inteligencia Artificial

La Inteligencia Artificial se está dividiendo y encontrando otros campos relacionados: lógica, redes neuronales, programación orientada a objetos, lenguajes formales, robótica, etc. Esto explica por qué el estudio de IA no está confinado a la matemática, ciencias de la computación, ingeniería (particularmente la electrónica y la mecánica), o a la ciencia cognoscitiva, sino que cada una de estas disciplinas es un potencial contribuyente. La robótica es considerada como un campo interdisciplinario que combina conceptos y técnicas de IA, con ingeniería óptica, electrónica y mecánica.

0.3 Diseño del Experimento

El presente trabajo hará uso del código base desarrollado por John DeNero y Dan Klein de la UC Berkeley, <http://ai.berkeley.edu>. Ese proyecto fue realizado en el lenguaje de programación python que simula el mundo de Pac-man, diseñado con varios escenarios en los que poder jugar.



Figure 2: Simulación del juego

Para correr esta versión de Pac-man se necesita contar con la versión python 2.7.x en su computadora.

3.1. Acciones

Pac-man puede realizar las siguientes acciones:

- Detenerse
- Desplazarse una posición al norte
- Desplazarse una posición al sur
- Desplazarse una posición al este
- Desplazarse una posición al oeste

3.2. Percepciones

Es muy importante mencionar la clase GameState definida en pacman.py debido a que lo que el agente Pac-man pueda percibir está definido ahí. El agente puede obtener las siguientes percepciones en cualquier momento:

- Su posición
- La posición de todos los fantasmas
- La locación de las paredes
- La posición de las capsulas
- La posición de bolita de comida

- ### 3.3. Ambiente

El ambiente es esencialmente una retícula de cuadrados. Para cualquier momento dado Pac-man ocupa un solo cuadrado y haciendo cara a alguna de las cuatro direcciones: norte, sur, este u oeste. Se debe tener en cuenta que cada cuadrado puede tener solo alguna de las siguientes cosas en su interior:

- Para este proyecto nuestro agente se desempeñará en el mundo (layout) “smallClasic” la cual tiene la siguiente retícula:



Como se puede observar en la imagen el agente solo se enfrentará a un fantasma, en futuras etapas se podría aumentar la cantidad de fantasmas para aumentar la dificultad del juego.

3.4. Agentes desarrollados para el proyecto:

Se hizo el diseño de tres agentes: DumbAgent, ReflexAgent y myAgent todos en el archivo Agents.py.

Los primeros dos agentes fueron realizados en la primera fase del proyecto, por razones de entrega de avances y para familiarizarnos con las acciones y percepciones que puede tener un agente. DumbAgent solo se dirige al oeste mientras puede. ReflexAgent selecciona una acción al azar, si esta acción lo lleva a ingerir alguna bolita de comida entonces la toma, si ninguna de las acciones disponibles lo llevan a ingerir entonces se ejecuta alguna acción al azar.

Y por último, el más importante, el agente que tendrá un aprendizaje por refuerzo. myAgent: myAgent toma solo 3 percepciones para aprender; posición actual del Pac-man (en x y en y), posición actual del fantasma (en x y en y), y ciertas cantidades de comida aún disponible. Se decidió no tomar más percepciones en cuenta para no obtener una base de datos demasiado grande como para alentar la ejecución de cada iteración del juego.

Recompensas: es muy importante elegir muy bien la recompensa otorgada al agente para cualquier estado que se visite, según la ponderación de estas, podría provocar un favorecimiento al aprendizaje. Nosotros elegimos otorgar las siguientes cantidades de recompensas:

- Si el agente gana recompensa =1000
- Si queda de 0 a 15 cantidades de comida
 - Y si el agente está a menos de 3 de distancia del fantasma recompensa=-500
 - Y si no, recompensa =250
- Si queda de 16 a 30 cantidades de comida
 - Y si el agente está a menos de 3 de distancia del fantasma reward=-500
 - Y si no, recompensa =50

- Si el agente está a menos de 3 de distancia del fantasma recompensa = -500
- Si el agente toma una decisión no disponible recompensa = -10000

Para la toma de decisiones el agente tomará la acción con la recompensa más alta el 60% de las veces, el 40% tomará otras acciones disponibles, esto con la finalidad de que se vayan explorando todos los estados con todas sus acciones.

Para mayor información de la clase myAgent puede revisar el archivo Agents.py este se encuentra totalmente documentado.

3.5. Archivos de información

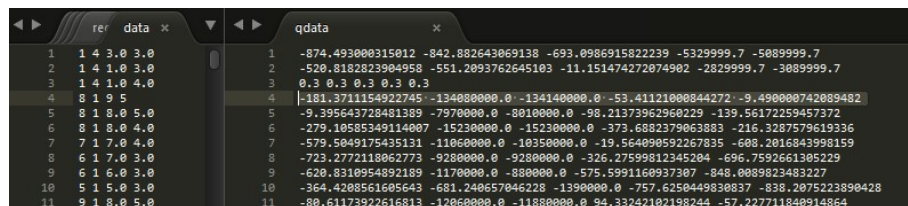
La información es guarda en 3 archivos diferentes.

“**data**”: archivo donde se guarda la siguiente información en el orden dicho: posición x del Pac-man, posición y del Pac-man, posición x del fantasma, posición y del fantasma. Esta información se puede considerar como los estados visitados por el agente.

“**qdata**”: archivo donde se guarda la puntuación para cada uno de los diferentes estados en “data” la información está escrita en el siguiente orden: detenerse, norte, sur, este y oeste.

“**registro**”: archivo donde se almacena el progreso del aprendizaje del agente. La información está escrita en el siguiente orden: número de serie jugado, puntuación promedio obtenida y cantidad de victorias en esa serie.

A continuación se explica un ejemplo para entender mejor cómo el agente interpreta estos archivos.



data	qdata
1 1 4 3.0 3.0	1 -874.493000315012 -842.802643069138 -693.0986915822239 -5329999.7 -5089999.7
2 1 4 1.0 3.0	2 -520.8182823904958 -551.2093762645103 -11.151474272074902 -2829999.7 -3089999.7
3 1 4 1.0 4.0	3 0.3 0.3 0.3 0.3 0.3
4 8 1 9 5	4 -181.3711154922745 -134080000.0 -134140000.0 -53.41121000844272 -9.490000742089482
5 8 1 8.0 5.0	5 -9.395643728481389 -7970000.0 -8010000.0 -98.21373962960229 -139.56172259457372
6 8 1 8.0 4.0	6 -279.10585349114007 -15230000.0 -15230000.0 -373.6882379063883 -216.3287579619336
7 7 1 7.0 4.0	7 -579.5049175435131 -11060000.0 -10350000.0 -19.564090592267835 -608.2016843998159
8 6 1 7.0 3.0	8 -723.2772118062773 -9280000.0 -9280000.0 -326.27599812345204 -696.7592661305229
9 6 1 6.0 3.0	9 -620.8310954892189 -1170000.0 -880000.0 -575.5991160937307 -848.0089823483227
10 5 1 5.0 3.0	10 -364.4208561605643 -681.240657046228 -1390000.0 -757.6250449830837 -838.2075223890428
11 9 1 8.0 5.0	11 -80.61173922616813 -12060000.0 -11880000.0 94.33242102198244 -57.227711840914864

Figure 4: archivos data y qdata

Para el estado en el que el Pac-man está en la posición (8,1) y el fantasma está en la posición (9,5), el agente interpreta que para la acción detenerse se tiene una puntuación de -181.3711154922745, desplazarse al norte tiene una puntuación de -134080000.0 y así sucesivamente.

3.6.Ejecución del juego

Para ejecutar el juego se escribe en la terminal el siguiente comando:
`python pacman.py`

Para ejecutar el juego por el agente `myAgent` es necesario agregar al primer comando lo siguiente: `-p myAgent`

Para ejecutar el juego en algún mundo (layout) por ejemplo “`smallClasic`” es necesario agregar al primer comando lo siguiente:
`-l smallClasic`

Por lo que si se quiere probar al agente en el mundo para el cual entrenó se debe ejecutar el siguiente comando:
`python pacman.py -p myAgent -l smallClasic`

En el archivo “`commands.txt`” están escritos más ejemplos de comandos que acepta el juego.

A continuación se mencionan los resultados obtenidos por el agente `myAgent`

0.4 Resultados

El agente fue entrenado en 45 series de 2500 juegos cada uno, es decir, se entrenó 112500 veces. Se espera que entre más entrene mejor resultados arrojará.

En el archivo llamado “registro” se guardó la información del avance del agente a lo largo de sus 45 series jugadas. A continuación se muestra la información en forma de gráficos para una fácil comprensión.

La gráfica siguiente representa la puntuación promedio obtenida en cada serie. Por ejemplo, en la quinta serie se obtuvo un promedio de -360 de 2500 juegos jugados.

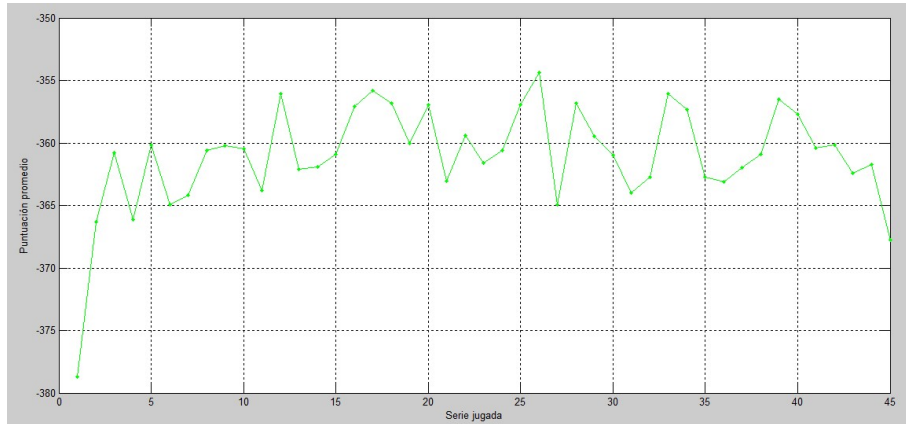


Figure 5: series jugada contra puntuación promedio obtenida

Se observa una mejoría a lo largo de las series no es la esperada pero es un hecho que el agente está aprendiendo.

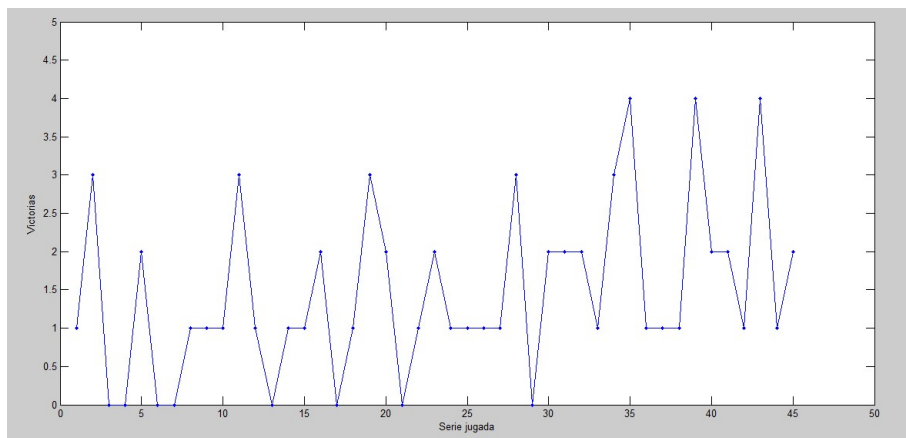


Figure 6: series jugada contra cantidad de victorias obtenidas en la serie

En la gráfica anterior se representa la cantidad de victorias obtenidas en cada una de las 45 series jugadas. Hay ocasiones en las que el agente no obtenía ni una sola victoria en su serie y otras en las que a lo más conseguía 4 victorias, no son tan exorbitantes la cantidad de victorias pero con esto se puede comprobar que el agente va teniendo un avance, cada vez va tomando mejores decisiones que lo lleven a la anhelada victoria.

Como ya se mencionó antes, para el aprendizaje solo se tomó en cuenta la posición actual del Pac-man, del fantasma y cantidades de bolitas de comida disponible. A mayores percepciones tomadas en cuenta mejor es el desempeño del aprendizaje. Es de mencionar que solo con las percepciones tomadas en cuenta se obtuvo 12476 estados diferentes cada uno con sus 5 acciones disponibles esto a lo largo de las 112500 partidas jugadas. Lo que haría más lento y complicado la escritura y actualización de la base de datos del agente.

0.5 Conclusiones

Para finalizar con este proyecto, quisiéramos decir lo interesante que fue realizarlo, ya que no sabíamos que sería posible que nosotros tomáramos un juego como es Pac-man y poder programar un agente el cual junto con la Inteligencia Artificial y un método de aprendizaje, fuera capaz de mejorar su desempeño en su entorno.

Con esto, vemos cual importante es la Inteligencia Artificial en nuestras vidas, como se puede relacionar con algo tan simple y poder mejorarlo hasta cierto punto. Estamos satisfechos con lo que hemos logrado a través de este proyecto ya que aplicamos conocimientos adquiridos durante el curso de la materia de Inteligencia Artificial. A futuro es posible implementar la Inteligencia Artificial de nuevas maneras y a diferentes aspectos, lo cual será beneficioso.

Esto nos impulsa a seguir trabajando duro para poder mejorar el futuro con nuevas formas, y sentimos que definitivamente la Inteligencia Artificial va a influenciar en esto.

Bibliography

- [1] J. R. QUINLAN, INDUCTION OF DECISION TREES, MACH. LEARN, 1986, Volumen 1, pages 81-106.
- [2] ITAM, BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL, editorial desconocida, 1987.
- [3] R. P. DÍEZ, A.G. GÓMEZ y N.A. MARTÍNEZ, INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL: SISTEMAS EXPERTOS, REDES NEURONALES ARTIFICIALES Y COMPUTACIÓN EVOLUTIVA, MACH. LEARN, 1986, Volumen 1, pages 81-106.
- [4] RAMÓN HERMOSO y MATTEO VASIRANI, APRENDIZAJE POR REFUERZO, Ing. Sup. en Informática, Universidad Rey Juan Carlos, 2011-2012.
- [5] HAWKING, EL DESARROLLO DE LA INTELIGENCIA ARTIFICIAL PUEDE SER EL ÚLTIMO LOGRO HUMANO, 2 mayo 2014, artículo.
- [6] LORENZO CINTRON RODRIGUEZO, IMPORTANCIA DE LA INTELIGENCIA ARTIFICIAL, Management Information Systems, 10 edition, 2006.
- [7] AUTOR DESCONOCIDO, es.wikipedia.org/wiki/Pac-Man#Modo_de_juego.
- [8] JIMMY, PACMAN: HISTORIA Y CURIOSIDADES, Artículo PixFans, may 8, 2007.
- [9] UC BERKELEY, SITIO DEL CÓDIGO BASE, <http://ai.berkeley.edu/home.html>.