Universidad de Sevilla
Doble Grado en Física e Ingeniería de Materiales
Trabajo Fin de Grado

Aprendizaje de Refuerzo Cuántico

Marina Orozco González

Tutor:
Lucas Lamata Manuel

Departamento de Física Atómica, Molecular y Nuclear
Facultad de Física

Resumen

A través de la introducción a dos campos de estudio de actualidad, como lo son la computación cuántica y la inteligencia artificial, se pretende hacer una descripción de una de las disciplinas que pueden resultar de la combinación de las otras: el aprendizaje cuántico por refuerzo.

La introducción hacia el formalismo de la computación cuántica parte de los principios de la mecánica cuántica para buscar su aplicación en el campo de la computación. A través de los *qubits*, surge toda una serie de posibilidades de puertas lógicas, métodos de resolución de tareas y propiedades que además de novedosas frente a la computación clásica, presentan serias mejoras respecto a ella en lo que a optimización se refiere.

Cuando se introduce la inteligencia artificial, se presentan las distintas clasificaciones que se pueden llevar a cabo de ella así como las características que la definen y distinguen dentro de las ciencias de la computación. Se hace especial hincapié en el aprendizaje por refuerzo, del cual se describen sus principales propiedades detalladamente siguiendo un formalismo matemático.

Por último, se ahonda en el campo del aprendizaje cuántico por refuerzo y en las ventajas que éste tiene para ofrecer. Se desarrolla extensamente el conocido *algoritmo de Grover* como resultado ejemplar de los significativos avances que pueden obtenerse a través de un método de aprendizaje cuántico.

Abstract

Through an introduction to two timely fields of study, such as quantum computing and artificial intelligence, we aim to describe one of the disciplines that can result from the combination of the two: quantum reinforcement learning.

The introduction to the formalism of Quantum Computing is based on the principles of Quantum Mechanics in order to search for its application in the field of computing. Through *qubits*, a whole series of possibilities of logic gates, task-solving methods and properties arise that, in addition to being novel compared to classical computing, present serious improvements in terms of optimization.

When introducing Artificial Intelligence, the different classifications that can be carried out and the characteristics that define and distinguish it within the Computer Sciences are presented. Special emphasis is placed on Reinforcement Learning, of which its main properties are described in detail following a mathematical formalism.

Finally, we delve into the field of quantum reinforcement learning and the advantages it has to offer. The well-known *Grover's algorithm* is extensively developed as an exemplary result of the significant advances that can be achieved through a quantum learning method.

Índice general

Capítulo 1

Computación Cuántica

1.1. Introducción

La computación cuántica, así como la información cuántica, puede definirse como el estudio de las tareas que procesan información a través de sistemas mecánico-cuánticos [1]. Esta simple definición no hace justicia al extenso, complejo y bello campo de estudio que establece los pilares sobre los que se sostenta la computación cuántica. La mecánica cuántica remonta sus orígenes al cambio de siglo XIX al XX, presentándose como una revolución tanto para nuestra forma de comprender el universo, como para la física y tantas otras disciplinas.

En cuanto estuvieron asentadas las bases de esta, no hubo demora en buscar aplicaciones a tal novedoso campo de estudio, surgiendo entre ellas la computación cuántica.

Actualmente, existe un elevado grado de desarrollo en lo que a algoritmos cuánticos se refiere, presentando como principal ventaja frente a los clásicos una mejora drástica en la optimización y eficiencia, así como todo un nuevo abanico de oportunidades en criptografía, comunicación y almacenamiento de datos. No obstante, este gran desarrollo en algoritmos no va de la mano con los progresos en la construcción de sistemas de procesado de información cuántica que permitan su ejecución [1]. Si bien se está trabajando en ello con gran ímpetu, es una realidad que los mayores logros se encuentran aún en el desarrollo de algoritmos y puertas lógicas cuánticas, en el cual nos basaremos a lo largo de este trabajo.

La mecánica cuántica, tal como se presenta comúnmente en el campo de información

cuántica, se basa en algunos postulados simples [2]:

1. El estado puro de un sistema cuántico viene dado por un vector unitario $|\Psi\rangle$ en un espacio de Hilbert complejo.

2. La evolución del estado puro de un sistema cerrado se genera por un Hamiltoniano H, especificado por la ecuación de Schrödinger lineal: $i\frac{\partial}{\partial t}|\Psi\rangle = H|\Psi\rangle$

3. La estructura de los sistemas compuestos se da por el producto tensorial.

4. Las mediciones proyectivas, denominadas observables, se especifican por operadores hermitianos ideales no degenerados. El proceso de medición cambia la descripción del sistema observado del estado $|\Psi\rangle$ a un estado propio φ, con una probabilidad que viene dada por la regla de Born: $p(\varphi) = |\langle\Psi|\varphi\rangle|^2$

Estos principios, que conforman la linealidad de la teoría cuántica, aunque no completan la teoría cuántica en toda su extensión pues aún requiere algunos subsistemas más, dan lugar por sí solos a muchos de los fenómenos cuánticos más reseñables: como el de superposición, el de no clonación o el de entrelazamiento [2]. Algunos de estos serán explicados con algo más de detalle en los próximos párrafos.

Para hacernos una idea sobre las ventajas que presenta un ordenador cuántico frente a uno clásico introducimos la *teoría de la complejidad computacional*, la cual clasifica la dificultad en la resolución de distintos problemas computacionales (sean clásicos o cuánticos) en función de la idea básica de la *clase de complejidad*, en base a los recursos computacionales requeridos para solucionarlos. Dos de las clases de complejidad más relevantes son la clase **P** y la clase **NP**. A grandes rasgos se diferencian en que los de la clase **P** son problemas que pueden solucionarse rápidamente mediante un ordenador clásico, mientras que en los de la clase **NP** lo que se puede hacer rápidamente es comprobar que sus soluciones son correctas. Un ejemplo concreto sería el cálculo de los factores primos de un cierto número entero n. Actualmente no se conoce ninguna forma de calcular esto en un tiempo razonable. No obstante, sí es inmediato comprobar computacionalmente si un cierto número p es un factor primo de otro número n, por lo que podríamos considerar este problema dentro de la clase **NP**.

Que **P** sea una subclase dentro del conjunto de los problemas de clase **NP** es evidente, pero a día de hoy sigue siendo una cuestión teórica sin resolver si puede decirse que haya problemas que estén en **NP** que no estén en **P**. Esta cuestión resulta interesante en tanto

que si finalmente fuera cierto que $\boldsymbol{P \neq NP}$, no podría resolverse de forma eficiente ninguno de los problemas de clase **NP-completo** (un tipo específico de problemas pertenecientes a la clase **NP**) con un computador clásico. En contraste, está demostrado que los ordenadores cuánticos pueden resolver rápidamente algunos problemas contenidos en **NP**, como el de factorización. Si se consiguiese demostrar que con ordenadores cuánticos podemos resolver problemas **NP completos**, sabríamos resolver cualquier problema **NP** con ordenadores cuánticos [1].

1.2. Fundamentos de la computación cuántica

Antes de continuar hablando sobre las distintas peculiaridades que lleva ímplicita la computación cuántica, conviene introducir los elementos fundamentales con los que esta trabaja [1].

- El equivalente en computación cuántica al concepto fundamental de la computación clásica, esto es, el *bit*, asociado a un estado que podía ser 0 o 1; es el *qubit*. Como estado cuántico, es una combinación lineal de los estados clásicos: una superposición de ellos. Empleando la *notación de Dirac* se formulan de la siguiente forma:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1.1}$$

  donde α y β son números complejos, $|\Psi\rangle$ es el estado de nuestro *qubit*, y los estados $|0\rangle$ y $|1\rangle$ forman la base ortonormal de estados computacionales con la que trabajamos. En notación vectorial podemos representar $|\Psi\rangle$ como:

$$|\Psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{1.2}$$

  Este estado cuántico cumple las propiedades propias de los estados cuánticos de la Mecánica Cuántica que citamos anteriormente:

  - Cuando hacemos una medida del *qubit* este va a "colapsar" sobre uno de sus autoestados, $|0\rangle$ o $|1\rangle$.
  - La probabilidad de medir cada uno de esos autoestados vendrá dada por el cuadrado del módulo del coeficiente asociado a cada autoestado, respectivamente:

$|\alpha|^2$ y $|\beta|^2$.

- La suma de estas probabilidades debe ser la unidad (conservación de la probabilidad): $|\alpha|^2 + |\beta|^2 = 1$.

Además, podemos pensar en los *qubits* a través de su representación geométrica:

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\,, \tag{1.3}$$

donde θ y φ son números reales que pueden asociarse a las coordenadas angulares esféricas a través de la conocida representación de la *esfera de Bloch* :

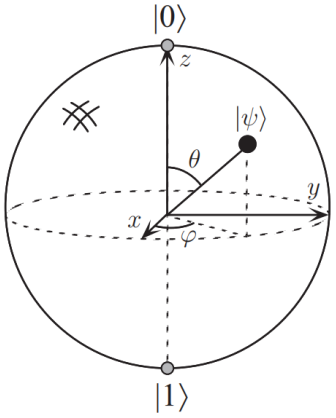

Figura 1.1: Representación de la esfera de Bloch asociada a un *qubit* [1].

- Otra base ortonormal que también es usada con frecuencia es la que está compuesta por los estados:

$$|+\rangle \equiv \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}, \tag{1.4}$$

$$|-\rangle \equiv \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}, \tag{1.5}$$

cuya equivalencia con la base descrita en (1.1) sería:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} + \beta\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} = \frac{\alpha+\beta}{\sqrt{2}}|+\rangle + \frac{\alpha-\beta}{\sqrt{2}}|-\rangle\,. \tag{1.6}$$

- Un sistema formado por dos *qubits* tendría la forma:

$$|\Psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,. \tag{1.7}$$

  En general, un sistema de n *qubits* tendría 2^n términos, y seguiría una expresión del tipo:

$$|\Psi\rangle = \sum_{i_1,i_2,\ldots,i_n=0,1} \alpha_{i_1,i_2,\ldots,i_n}\,|i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_n\rangle\,, \tag{1.8}$$

  donde $\otimes$ debe entenderse como un operador que actúa de la siguiente forma para estados de un solo *qubit*:

$$|\Psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}. \tag{1.9}$$

- Un sistema de dos *qubits* importante es el ***Bell State*** o ***par EPR***:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{1.10}$$

  el cual tiene la propiedad de tener la misma probabilidad al medir el primer *qubit* de obtener el resultado $|00\rangle$ que el de $|11\rangle$. Además, la medida del segundo *qubit* da siempre por resultado la misma medida que la tomada en el primer *qubit*. Esto quiere decir que las medidas del primer y segundo *qubit* están *correlacionadas* [1].

- Al igual que en la computación clásica, los circuitos de ordenadores cuánticos están compuestos por "cables" y por puertas lógicas. Los cables se usan para transportar información a lo largo del circuito, mientras que las puertas lógicas se encargan de transformar dicha información.

- La representación de los *circuitos cuánticos* se lleva a cabo a través de *cables*, que no tienen por qué corresponderse con cables del circuito físico, de tal forma que cada cable corresponde a un *qubit*. Deben leerse de izquierda a derecha, interpretándose tal lectura o bien como el movimiento que tendría un fotón moviéndose por el

espacio o bien como el paso del tiempo [1]. Véase el ejemplo de una serie de cables representando múltiples *qubits* pasando por una puerta lógica *Controlled-U* $C-U$ (esta será introducida en los próximos puntos):

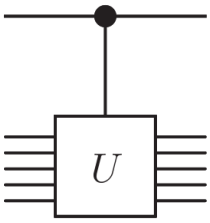

Figura 1.2: C-U GATE para múltiples *qubits* [1].

- El símbolo empleado para representar que se está produciendo una medición sobre el sistema colapsándolo se incluye en la siguiente figura.

  Por la alineación entre líneas, distinguimos el cable de línea doble que vemos en la siguiente figura de dos cables correspondientes a distintos *qubits*, indicándose así que el *qubit* ha sido colapsado a uno de sus autoestados $|0\rangle$ o $|1\rangle$.

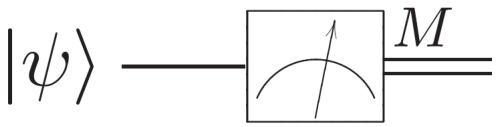

Figura 1.3: Símbolo empleado en circuitos cuánticos para indicar medida *qubits* [1].

- El análogo a las puertas lógicas de la computación clásica, es decir, las *puertas lógicas cuánticas*, corresponden a operaciones unitarias para así cumplir con la propiedad de conservación de la probabilidad. De esta forma, todas las matrices U que cumplan $U^\dagger U = UU^\dagger = I$, siendo I la matriz identidad, pueden relacionarse con una puerta lógica cuántica [1].

- Existen varias puertas lógicas de un único *qubit* que son no triviales, a diferencia del caso clásico en que tan solo la puerta lógica NOT es no trivial. En la siguiente tabla podemos encontrar algunas puertas lógicas cuánticas no triviales:

  - *NOT GATE:*

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{1.11}$$

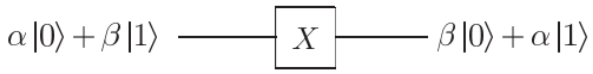

Figura 1.4: NOT GATE [1].

- *Z GATE:*

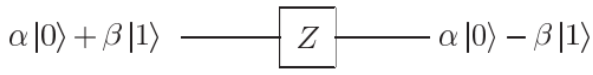

Figura 1.5: Z GATE [1].

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Z \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix}. \tag{1.12}$$

- *HADAMARD GATE:*

Figura 1.6: HADAMARD GATE [1].

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.13}$$

$$|0\rangle \to (|0\rangle + |1\rangle)/\sqrt{2}, \tag{1.14}$$

$$|1\rangle \to (|0\rangle - |1\rangle)/\sqrt{2}. \tag{1.15}$$

- *PHASE GATE:*

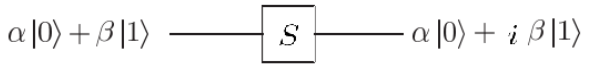

Figura 1.7: PHASE GATE [1].

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad S \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ i\,\beta \end{bmatrix}. \tag{1.16}$$

- $\frac{\pi}{8}$ *GATE:*

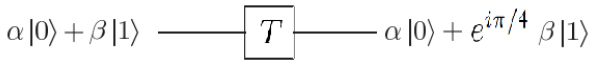

Figura 1.8: $\frac{\pi}{8}$ GATE. Edición a partir de [1].

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta\, e^{i\pi/4}. \end{bmatrix}. \tag{1.17}$$

- También existen puertas lógicas de múltiples *qubits*, siendo la puerta prototipo la puerta *controlled-NOT* o $CNOT$ la cual, va a recibir dos *qubits*: el *qubit de control* y el *qubit objetivo o target*. Su funcionamiento viene dado por:

  - *C-NOT GATE:*

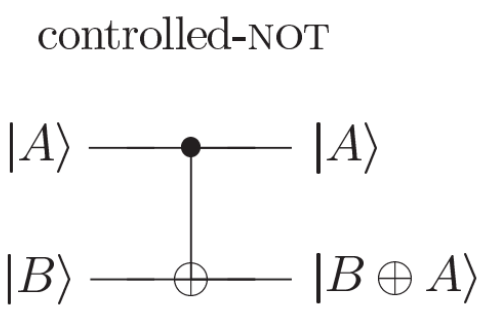

Figura 1.9: NOT GATE [1].

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{1.18}$$

$$|00\rangle \to |00\rangle\,; \quad |01\rangle \to |01\rangle\,; \quad |10\rangle \to |11\rangle\,; \quad |11\rangle \to |10\rangle\,. \tag{1.19}$$

Donde $\oplus$ es la operación de adición modular. Otra puerta lógica para múltiples *qubits* es:

- *TOFFOLI* GATE.

  Esta puerta lógica es reversible y universal en computación clásica, y tiene aplicaciones muy útiles dentro de la computación cuántica. Supongamos que tenemos $n+k$ *qubits*. La actuación de la *Toffoli gate* junto con un operador U unitario que actúa sobre k qubits es de la siguiente forma:

$$C^n(U)\left|x_1x_2...x_n\right\rangle\left|\psi\right\rangle = \left|x_1x_2...x_n\right\rangle U^{x_1x_2...x_n}\left|\psi\right\rangle, \tag{1.20}$$

  donde al estar elevado el operador U al producto de los *qubits* $x_1x_2...x_n$, este solo actuará sobre $|\psi\rangle$ si los n primeros *qubits* que lo componen están en el estado $|1\rangle$ ya que de lo contrario lo dejaría invariante [1].

  Como esta puerta lógica suele ser muy útil, se la representa con una notación propia concreta, que se indica a continuación para un ejemplo concreto en el que el número total de *qubits* es 7 con $n=4$ y $k=3$:

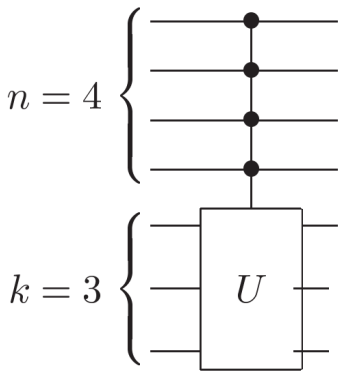

Figura 1.10: TOFFOLI GATE [1].

- En computación cuántica, se dice que una puerta lógica es *universal* cuando cualquier operación unitaria puede ser aproximada con precisión a partir de un circuito cuántico compuesto únicamente por tales puertas lógicas universales. Esta definición de universalidad se sostenta sobre tres principios:

  1. Un operador unitario arbitrario puede expresarse de forma exacta como un producto de operadores unitarios que actúen de forma no trivial únicamente sobre un subespacio generado por un par de estados de la base computacional.
  2. Un operador unitario arbitrario puede expresarse exactamente a partir de puertas lógicas de un solo *qubit* y de la puerta lógica para múltibles *qubits* $CNOT$

(1.18).

3. Las operaciones de un solo *qubit* pueden aproximarse con una precisión arbitraria utilizando las puertas lógicas *HADAMARD* (1.13), $\pi/8$ (1.17) y *de fase* (1.16). Esta última puede a su vez escribirse en función de dos puertas lógicas $\pi/8$, pero se incluye aquí por el rol que juega en ciertas construcciones tolerantes a fallos (conjunto de técnicas y metodologías utilizadas en computación cuántica para garantizar que los cálculos y operaciones sean robustos y precisos, incluso en presencia de errores en los componentes físicos de los sistemas cuánticos).

De esta forma, podemos escribir cualquier operación unitaria de forma aproximada con una precisión arbitraria a través de las puertas lógicas *HADAMARD*, $\pi/8$, *CNOT* y *de fase* [1].

- Los operadores lógicos clásicos solo tienen un equivalente cuántico si pueden representarse a través de matrices invertibles, consecuencia directa de su condición de unitariedad. La no reversibilidad conlleva asociada una pérdida de información en tanto que no podemos determinar cuál fue el estado *input* a la puerta lógica. Por tanto, quedan excluídas de la computación cuántica puertas lógicas clásicas como *NAND, OR* o *XOR general* y operaciones como *FANIN* (convierte dos *bits* en uno a través de la puerta *OR* y *FANOUT* (en la que se duplica un *bit*) [1]. Respecto a la clonación de *qubits*, resulta de especial interés el ***teorema de no clonación***, el cual tiene profundas implicaciones tanto en computación cuántica como en otras áreas de estudio relacionadas [3].

  El ***teorema de no clonación*** fue enunciado por Wootters, Zurek y Dieks en 1982, y en él se declara que es imposible crear una copia idéntica de un estado cuántico arbitrario desconocido [4]. Formalmente: No existe operador unitario U actuando sobre $H_A \otimes H_B$ de forma que para todos los estados normalizados $|\psi\rangle_A$ y $|e\rangle_B$ en H cumpla que:

$$U\,|\psi\rangle_A\,|e\rangle_B = |\psi\rangle_A\,|\psi\rangle_B\,. \tag{1.21}$$

  La demostración de este teorema es bastante simple y merece la pena dedicar brevemente nuestra antención a ello:

- Sean dos sistemas cuánticos A y B con un *espacio de Hilbert* común $H_A = H_B = H$. Si queremos copiar un estado normalizado $|\psi\rangle_A \in H_A$ en otro estado cuántico normalizado perteneciente al sistema B $|e\rangle_B \in H_B$ e independiente del estado a copiar $|\psi_A\rangle$, querremos buscar una transformación unitaria U tal que se cumpla (1.21).

  Esto debe cumplirse para cualquier estado de A arbitrario, de forma que (1.21) también será válido para otro estado arbitrario de A, $|\phi\rangle_A$:

$$U\,|\phi\rangle_A\,|e\rangle_B = |\phi\rangle_A\,|\phi\rangle_B\,. \tag{1.22}$$

  Las condiciones que deben cumplirse son:

  1. U debe ser una transformación unitaria.
  2. Debe poderse copiar cualquier estado arbitrario $|\psi\rangle$
  3. La copia o clonación debe realizarse sin colapsar el sistema a uno de sus autoestados, eso es, sin realizar observaciones sobre el sistema.

  Si tomamos el producto escalar de las expresiones (1.21) y (1.22):

$$\left(\langle\phi|_A\,\langle e|_B\,U^\dagger\right)\left(U\,|\psi\rangle_A\,|e\rangle_B\right). \tag{1.23}$$

  Esta expresión puede desarrollarse de dos formas:

  - O bien imponiendo la transformación de U de acuerdo con (1.21):

$$\left(\langle\phi|_A\,\langle e|_B\,U^\dagger\right)\left(U\,|\psi\rangle_A\,|e\rangle_B\right) = \left(\langle\phi|_A\,\langle\phi|_B\right)\left(|\psi\rangle_A\,|\psi\rangle_B\right) = \left(\langle\phi|\psi\rangle\right)^2, \tag{1.24}$$

  - o bien imponiendo la unitariedad de U (condición 1) a través de $U^\dagger U = I$:

$$\left(\langle\phi|_A\,\langle e|_B\,U^\dagger\right)\left(U\,|\psi\rangle_A\,|e\rangle_B\right) = \langle\phi|_A\,\langle e|_B\,|\psi\rangle_A\,|e\rangle_B = \langle\phi|\psi\rangle\,, \tag{1.25}$$

  de donde tendríamos que necesariamente:

$$\left(\langle\phi|\psi\rangle\right)^2 = \langle\phi|\psi\rangle\,. \tag{1.26}$$

  Lo cual solo es cierto si $|\psi\rangle = |\phi\rangle$ o si $|\psi\rangle$ y $|\phi\rangle$ son ortogonales y $\langle\psi|\phi\rangle = 0$, luego no puede ser un estado arbitrario. Esto implica que *no es posible emplear un U universal para clonar un estado cuántico arbitrario* [1].

- Un rasgo fundamental de los algoritmos cuánticos es la propiedad del *paralelismo cuántico*, y es que existe la posibilidad de evaluar una función $f(x)$ en diferentes valores x simultáneamente. La transformación correspondiente a la función $f(x)$ vendría dada por una puerta lógica que podemos denominar U_f a priori. En el caso de un sistema con dos *qubits*, convendría considerar un estado inicial $|x, y\rangle$. La transformación dejará el primer *qubit* invariante y actuará sobre el segundo *qubit* a través de una adición modular de la función aplicada sobre el primer *qubit*, con lo cual conviene establecer el segundo *qubit* en el estado $|0\rangle$, y convertir el primer *qubit* en una superposición de estados $|0\rangle$ y $|1\rangle$, por ejemplo a través de una puerta lógica *Hadamard* . Tendríamos pues como ejemplo:

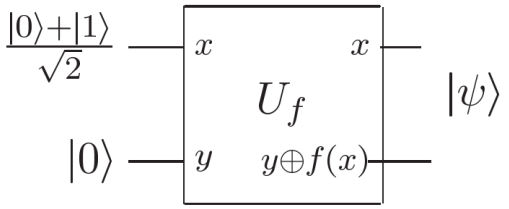

Figura 1.11: Ejemplo de circuito cuántico en que se evalúa $f(0)$ y $f(1)$ simultáneamente [1].

Este proceso puede generalizarse a funciones sobre un número arbitrario de *qubits*.

En un ordenador clásico, serían requeridos distintos circuitos para poder evaluar distintos valores de x en una función $f(x)$ de forma simultánea. En un ordenador cuántico podemos aprovechar la propiedad de la superposición de estados propia de la mecánica cuántica para evaluar distintos resultados con un solo circuito.

No obstante, cuando quieras obtener tal información, tendrías que colapsar el sistema. En nuestro ejemplo anterior, esto te daría únicamente la información sobre un valor concreto de x, es decir, colapsarías el sistema en $|0, f(0)\rangle$ o en $|1, f(1)\rangle$, lo cual no supone ningún avance respecto a un ordenador clásico. Para poder sacar provecho verdaderamente a esta propiedad cuántica, serán necesarios algoritmos como el de *Deutsch* o el de *Deutsch-Jozsa*, que nos permitan conocer información de distintos $f(x)$ con solo una medida [1].

1.3. Aceleración cuántica

Ya se ha citado anteriormente alguno de los problemas que resultan intratables para un ordenador clásico pero que sí son resolubles por ordenadores cuánticos, como los problemas

de factorización de números enteros grandes, pero existen otros muchos, como la solución de ecuaciones de Pell por ejemplo. También ofrece significativas mejoras en problemas de optimización y de simulación, tales como el cálculo de ciertas funciones de partición, algoritmos de temple simulado (SA) para problemas de optimización global, programación semidefinida (SDP), para mejorar la eficiencia en subrutinas y en base de datos, para problemas de búsqueda espacial, de evaluación a través de árboles booleanos, problemas de muestreo como las conocidas cadenas de Markov Monte Carlo y por supuesto, para simulación de sistemas cuánticos.

Esto último se debe a que, para estudiar un sistema de n partículas cuánticas (pongamos, átomos de dos niveles), será necesario emplear 2^n *bits* de información, uno para cada amplitud compleja. Un ordenador cuántico sin embargo, podrá representar estas amplitudes empleando n *qubits*, lo cual supone un aumento de velocidad exponencial respecto a los ordenadores clásicos, gracias a la capacidad que los ordenadores cuánticos tienen para aprovechar las interferencias cuánticas que tienen lugar entre las amplitudes. Esto se consigue a través de algoritmos como el de Shor, que hacen uso de una operación unitaria conocida como la *transformada cuántica de Fourier (QFT)* [5]. Desarrollar el algoritmo de Shor se sale de los objetivos de este trabajo, pero para hacer una idea de su relevancia en disciplinas como la criptografía, cabe decir que la puesta en práctica de un algoritmo como este a través de un ordenador cuántico llevaría a la obsolescencia de numerosas criptografías de clave pública, las cuales se basan en el cifrado a través de la descomposición en factores de números enteros de un cierto número N. Un algoritmo clásico no sería capaz de descifrarlo en un tiempo polinómico, mientras que el algoritmo cuántico de Shor sería capaz de hacerlo en un tiempo del orden de $log(N)^3$ [1].

Capítulo 2

Inteligencia Artificial

2.1. Introducción

Existe una idea confusa sobre a qué se refiere el concepto de *inteligencia artificial (IA)* y qué engloba. Aunque no existe una definición oficial por parte del sector científico, podemos definirla a través de dos características que les son comunes a todo este subconjunto de lo que se conoce como *ciencias de la computación*: la autonomía y la capacidad de adaptación [6]. Se entiende por autonomía la habilidad de llevar a cabo tareas en un medio complejo sin ser guiado continuamente por el usuario. En cuanto a la adaptación, se refiere a la habilidad de mejorar tus acciones a través del aprendizaje que te da la experiencia.

Si bien la inteligencia artifical es un subcampo dentro de las ciencias de la computación, también se dice que el *aprendizaje automático* o *Machine Learning* es a su vez un subcampo de la inteligencia artificial. Estos subcampos no están en absoluto bien acotados y a veces incluso puede incluirse el machine learning dentro de otras categorías como la estadística. Este es el que permite que las soluciones de la IA tengan la capacidad de ser adaptables. Concretamente puede definirse como aquel sistema capaz de mejorar el desempeño de su tarea a través de adquirir más experiencia y datos [6].

El aprendizaje automático puede clasificarse en dos amplios grupos según la forma que tengan de "aprender"(indagaremos en el uso de este término en este contexto más adelante): mediante big data o mediante interacciones [7].

Dentro del primer grupo encontramos dos subclases:

- El ***aprendizaje supervisado*** trabaja a partir de datos previamente clasificados con los que se entrena y genera un patrón de relación entre etiquetas y datos que le permita clasificar nuevos datos: dado un cierto número de puntos etiquetados $\{(x_i, y_i)\}_i$, donde x_i son los datos en forma de puntos e y_i sus etiquetas correspondientes, la tarea será encontrar una regla de etiquetado $x_i \to y_i$ que permita obtener una etiqueta a patir de unos datos distintos previamente desconocidos. Formalmente esta tarea consiste en buscar una distribución de probabilidad condicionada $P(Y = y|X = x)$. Su aplicación por excelencia es en problemas de reconocimiento de patrones.

- En el ***aprendizaje no supervisado*** el algoritmo recibe una serie de datos sin etiquetar y su tarea consiste en identificar las propiedades de la distribución $P(X = x)$ con la que se corresponden tales datos. Si bien las agrupaciones que se producen pueden ser interpretadas como etiquetas, la diferencia con el tipo de aprendizaje anterior está fundamentalmente en que este algoritmo no requiere un trabajo previo de etiquetado, una supervisión. Encuentra su uso en problemas de agrupamiento [2].

Un ejemplo ilustrativo de la diferencia entre ambos tipos de aprendizajes sería el siguiente:

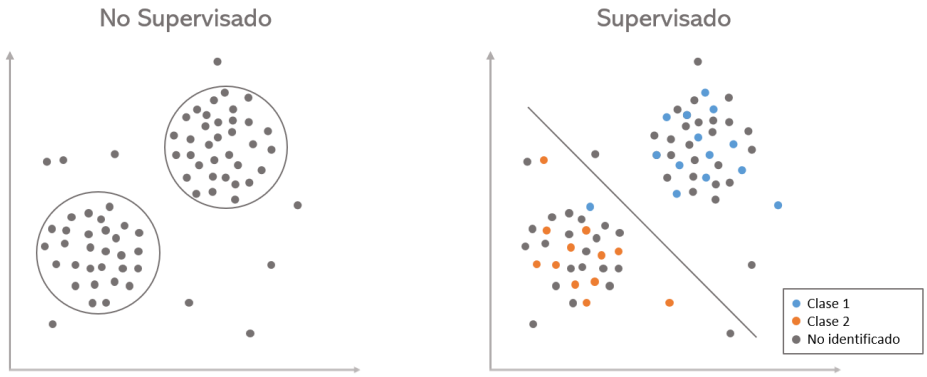

Figura 2.1: Representación gráfica del tipo de datos con el que trabajan los mecanismos de aprendizaje supervisado y no supervisado.

Dentro del segundo grupo, el de aprendizaje a través de interacciones, es donde se sitúa la clase del ***aprendizaje por refuerzo***. La propiedad que lo diferencia de las otras formas de aprendizaje es que se entrena a partir de la evaluación de las acciones que va tomando, en lugar de a partir del conocimiento de qué acciones son las correctas. Es por ello que esta forma de aprendizaje es la más próxima a la que se desarrolla en los

organismos biológicos inteligentes. Esto conlleva la necesidad de una exploración activa que, acompañada de una evaluación sobre las acciones tomadas, llegue a base de prueba y error al comportamiento óptimo. Cabe añadir que en ningún momento tienes información sobre si la acción tomada es la mejor que podrías haber llevado a cabo, sino solo cómo de adecuada es la que has tomado: la información depende de forma directa de las acciones que tomes, a diferencia de otros tipos de aprendizajes en los que la respuesta correcta es independiente a la que tú tomes. Está basada en una función de optimización que va evolucionando según recibe feedback de su entorno [8].

2.2. Aprendizaje por refuerzo

Si bien ya se han citado algunos de los elementos fundamentales del aprendizaje por refuerzo, en este apartado se pretende ahondar en ellos e ir desarrollando la base sobre la que se sustentan estos algoritmos.

A grandes rasgos, puede decirse que el aprendizaje por refuerzo funciona de la siguiente forma: partimos de dos sistemas básicos, un **agente** y un **medio**. El agente va a recibir de forma directa o a partir de un sistema auxiliar denominado sensor R que actúe como intermediario una *percepción* emitida por el medio. Con la información obtenida, el agente toma una decisión sobre qué **acción** A llevar a cabo. Como respuesta, el medio emite una **recompensa** o un **castigo** de vuelta al agente. La tarea del agente consiste en maximizar tales recompensas [7].

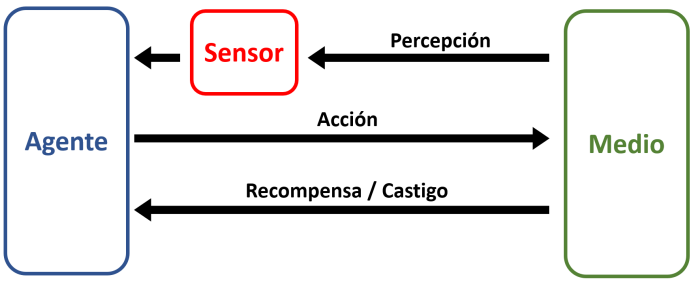

Figura 2.2: Interacción entre el agente y el medio en aprendizaje por refuerzo.

Por lo general se considera que todo aquello que no puede ser alterado de forma aleatoria por el agente es externo al agente y parte del medio [8]. Por ejemplo, la emisión de recompensas se considera externa al agente porque este no tiene capacidad de alterarlas. La frontera agente-medio puede estar situada en distintos sitios y su elección dependerá de

la tarea que queramos ejecutar. Normalmente se elige una vez hemos escogido los estados, acciones y recompensas que van a entrar en juego en nuestro problema, es decir, cuando hayamos descrito nuestra tarea.

Otra puntualización relevante es que no asumimos que todo el medio es por completo desconocido para el agente. El agente suele conocer que debe calcular las recompensas en función de las acciones que tome y de los estados de los que parta. Incluso puede darse que tenga una descripción completa de su entorno y el problema de qué acción tomar para ejecutar una cierta tarea de forma óptima siga sin ser trivial (por ejemplo, el resolver un cubo de Rubik) [8].

El agente y el medio interactúan entre sí en una secuencia de pasos discretos de tiempo $t = 0, 1, 2, ...$ Para cada paso de tiempo t el agente recibe una percepción representada por un ***estado*** del medio $S_t \in S$, siendo S el conjunto de posibles estados en los que se puede presentar el medio. En base a ello, selecciona una **acción** $A_t \in A(S_t)$, siendo $A(S_t)$ el conjunto de acciones que pueden tomarse cuando el medio presenta un estado S_t. Un paso temporal después $(t + 1)$, en función de la acción ejecutada por el agente este recibe una recompensa $R_{t+1} \in R$, siendo R el conjunto de valores reales que pueden recibirse como recompensa. Además, la acción tomada por el agente da lugar a su vez a un nuevo estado del medio S_{t+1} [8].

A cada paso temporal el agente va trazando un mapeado que relaciona los estados del medio con las probabilidades de seleccionar cada una de las posibles acciones que puede ejecutar ante tal estado. Este mapeado recibe el nombre de ***política del agente*** y es denotado como π_t, donde $\pi_t(a|s)$ es la probabilidad de que $A_t = a$ si $S_t = s$. A través del aprendizaje por refuerzo, el agente va cambiando su política como resultado de la experiencia con el fin último de maximizar la recompensa que recibe.

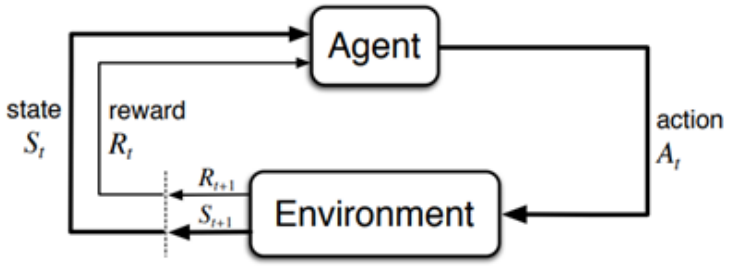

Figura 2.3: Interacción en cada paso temporal. [8]

2.2.1. Recompensas

Como ya hemos dicho, el objetivo del agente es maximizar la cantidad total de recompensas que recibe, las cuales forman parte del medio y son por lo tanto externas al agente. Esto quiere decir que lo que debe maximizar no son las recompensas inmediatas sino la suma total de todas las recompensas recibidas: el valor esperado de la suma de tales subrecompensas. Definimos pues como *señal escalar recibida* a esas recompensas inmediatas que se reciben en cada paso temporal. Estas serían las que recibiría el agente por cada acción que ejecute. No obstante, el objetivo que debe perseguir el agente no es ejecutar las acciones que le permitan obtener subrecompensas, sino conseguir cumplir un objetivo final cuantificado a través de la suma de tales subrecompensas. Si pensamos en una partida de ajedrez, nuestro objetivo último no debe ser colocarnos en las posiciones más ventajosas sino ganar la partida [8].

En general, podemos distinguir cuatro modelos diferentes de refuerzo ante el comportamiento del agente: *refuerzo positivo* cuando respondemos con una *recompensa* cuando la actuación sea correcta, *refuerzo negativo* cuando se le retira un *castigo* o *recompensa negativa* cuando la actuación sea correcta, *castigo positivo* cuando recibe un *castigo* ante una acción incorrecta y *castigo negativo* cuando se le retira una *recompensa* ante una acción incorrecta. En aprendizaje por refuerzo, es el de *refuerzo positivo* el que suele emplearse más comunmente [2].

Para formalizar el concepto de recompensa, introducimos el concepto de *respuesta esperada*, una función específica G_t (aquella que queremos maximizar) de las señales escalares que se espera recibir en las próximas secuencias temporales $R_{t+1}, R_{t+2}, ..., R_T$ (siendo T el último paso temporal). Se considera que una cierta tarea es **episódica** cuando podemos dividir la acción del agente en distintos episodios, siendo su expresión sencilla:

$$G_t = R_{t+1} + R_{t+2} + ... + R_T. \tag{2.1}$$

No obstante, cuando trabajas con **tareas continuas** que no se pueden dividir en pasos temporales como tal, nos encontramos con el problema de la divergencia de G_t como consecuencia de que T tiende al infinito a priori. Para solucionar esto se introducen los *descuentos* γ en la expresión de la respuesta esperada:

$$G_t = R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{2.2}$$

donde γ se denomina la *tasa de descuento* y toma un valor de entre $0 \leq \gamma \leq 1$. A través de este factor reducimos el valor de las recompensas que se obtendrán muchos pasos temporales por delante frente a los más inmediatos, y hacemos converger el valor de G_t cuando $T \to \infty$. Cuando $gamma = 1$ obtenemos la expresión correspondiente a la división de la acción en pasos temporales sucesivos (2.1), y cuando $\gamma = 0$ obtenemos el caso particular en que el objetivo es elegir la acción A_t que maximice la recompensa inmediata R_{t+1}.

Si bien el caso de tareas episódicas y continuas es distinto, podemos introducir un formalismo general que contemple ambos casos. Para no tener que recurrir a una única secuencia de pasos temporales muy larga, se introduce el concepto de **episodios**, cada uno de los cuales estará compuesto por una secuencia finita de pasos temporales. De esta forma, definiríamos todos los elementos propios del aprendizaje por refuerzo tales como el estado, la acción y demás como $S_{t,i}$ y $A_{t,i}$, donde t vendrá referido al paso temporal t del episodio i. No obstante, resulta que en el caso de tareas episódicas no es necesario especificar a qué episodio nos referimos por recibir todos el mismo tratamiento y por lo general se obvia el subíndice i. Para poder obtener un mismo formalismo para tareas episódicas y continuas, en que el número de sumandos pasa de ser finito a infinito, es necesario introducir en el caso continuo un ***estado absorbente*** que solo dé lugar a transiciones hacia sí mismo y a partir del cual todas las recompensas sean nulas [8].

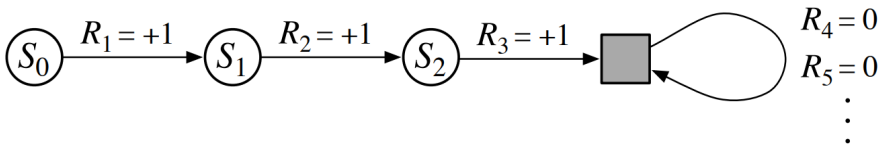

Figura 2.4: Estado absorbente. [8]

Podríamos definir de esta forma la respuesta esperada como:

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}, \tag{2.3}$$

contemplando así la posibilidad de que $T = \infty$ o la de que $\gamma = 1$, siempre que no se den ambas simultáneamente [8].

2.2.2. Medio

En esta sección ahondaremos en el *estado* del medio para analizar qué requerimos de dicho estado y qué tipo de información esperamos obtener de él.

Como ya introdujimos anteriormente, el agente va a tomar decisiones sobre qué acción ejecutar en función de las señales emitidas por el medio que detecte. Cabe recalcar que tal señal no incluye una descripción completa del medio sino solo aquella parte que el agente puede recibir.

La información que va a recibir el agente puede tratarse tanto de información recibida directamente por los sensores como de cálculos obtenidos indirectamente a partir del tratamiento de ciertas medidas. Un ejemplo de esto último sería la información que se obtiene sobre la velocidad de un móvil a partir de la medida de su posición en dos instantes de tiempo diferentes. Toda esta información va a recogerse en una memoria de percepciones, de tal forma que el estado del medio no va a estar restringido a percepciones inmediatas y directas [8].

Conseguir un estado que recoja toda la información obtenida con anterioridad y que vaya incluyendo la nueva que se va percibiendo implica que tal estado debe poseer la **propiedad markoviana**. La información que recogerá será mayor a la de las percepciones inmediatas pero nunca superior al conjunto de percepciones recibidas en conjunto. Un ejemplo de estado markoviano serían las posiciones de todas las piezas en una partida de damas, en la que tan solo con esta percepción podríamos obtener información sobre la secuencia de posiciones que llevaron a ella que nos es relevante. Y es que si bien perdemos cierta información, la que es relevante para el futuro de la partida queda retenida.

Para definir la propiedad de Markov formalmente, consideraremos que el medio va a responder en un paso temporal $t+1$ a las acciones tomadas en un tiempo t, y que hay un número finito de estados y de recompensas, siendo el valor de estas también finito. Esta última suposición no hace más que facilitar los cálculos, y es fácilmente extendible a casos en los que exista continuidad tanto de estados como de recompensas.

En el caso más general, la dinámica de los estados y recompensas del medio serán dependientes de todo lo que haya ocurrido con anterioridad y necesitaremos recurrir a la distribución de probabilidad completa, en la que hay que tener en cuenta el posible valor de todos los eventos pasados:

$$P_r\left\{R_{t+1}=r, S_{t+1}=s' | S_0, A_0, R_1, ..., S_{t-1}, A_{t-1}, R_t, S_t, A_t\right\}. \tag{2.4}$$

En el caso en que cumplan la propiedad de Markov, la respuesta del medio en el paso temporal $t+1$ solo dependerá del estado y de la acción en t, permitiéndonos definir la dinámica de una forma más sencilla:

$$p\left(s', r|s, a\right) = P_r\left\{R_{t+1} = r, S_{t+1} = s'|S_t, A_t\right\}. \tag{2.5}$$

Cuando se cumpla (2.5) para todos los valores de r, s', S_t y A_t, tendremos un estado de Markov. En estas circunstancias, será posible conocer la próxima respuesta del medio a partir del estado actual y la acción que se tome, ya que será lo único de lo que dependa. Extrapolando esta forma de actuar, podríamos llegar a predecir todos los estados futuros junto a sus recompensas correspondientes a partir del único conocimiento de la situación presente con la misma certeza que lo haríamos si conociéramos la historia completa del sistema hasta el momento actual [8].

Tener estados de Markov es la situación más favorable a la hora de elegir qué acciones tomar. No obstante, la realidad es que no todos los estados con los que se trata cumplen la propiedad de Markov estrictamente. Pese a ello, suele ser conveniente hacer una aproximación a un estado de Markov cuando sea posible, ya que gran parte de los algoritmos de aprendizaje por refuerzo se construyen partiendo de la premisa de que los estados cumplen con esta propiedad. También permitirá tratar problemas más complejos, partiendo de tal premisa y añadiendole ciertos matices.

2.2.3. Proceso de decisión

Cuando tratamos con una tarea en aprendizaje por refuerzo que satisface la propiedad de Markov se la denomina *proceso de decisión de Markov* (Markov decision process, MDP). Cuando además el espacio del estado y de las posibles acciones son finitos, hablamos de *procesos finitos de decisión de markov* (finite MDP). Este tipo de tareas son suficientes para comprender el 90 % del RL moderno [8], razón por la cual nos centraremos en estos casos.

La dinámica de un finite MDP, queda especificada por completo a partir de las cantidades que aparecen en la expresión (2.5) que vimos anteriormente [8]. En consecuencia, en función de ella podremos definir:

- Las recompensas esperadas para cada par estado-acción:

$$r(s,a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{r\in\mathbb{R}} r \sum_{s0\in\mathbb{S}} p\left(s', r|s, a\right). \tag{2.6}$$

- Las probabilidades de transición entre estados

$$p(s'|s,a) = P_r\left\{S_{t+1} = s'|S_t = s, A_t = a\right\} = \sum_{r\in\mathbb{R}} p(s', r|s, a). \tag{2.7}$$

- Las recompensas esperadas por cada triplete estado-acción-próximo estado:

$$r(s,a,s') = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r\in\mathbb{R}} rp(s', r|s, a)}{p(s'|s, a)}. \tag{2.8}$$

A través de los *gráficos de transición* podemos recoger la información que nos da la dinámica de un *finite MDP*. En él se representan *nodos de estado*, representados por un círculo blanco en la siguiente figura, y *nodos de acciones* para cada par estado-acción, representados por los círculos negros pequeños. Estos nodos están conectados de tal forma que los nodos de la acción que conecte dos estados estará en la línea que una los nodos de estado correspondientes. En tales líneas se indica tanto la probabilidad de transición entre estados $p(s'|s,a)$ como la recompensa esperada por tal transición $r(s,a,s')$. Siempre ha de cumplirse que la suma de las probabilidades de transición de cada triplete estado-acción estado debe ser uno para cada par estado-acción. Veamos un ejemplo a continuación:

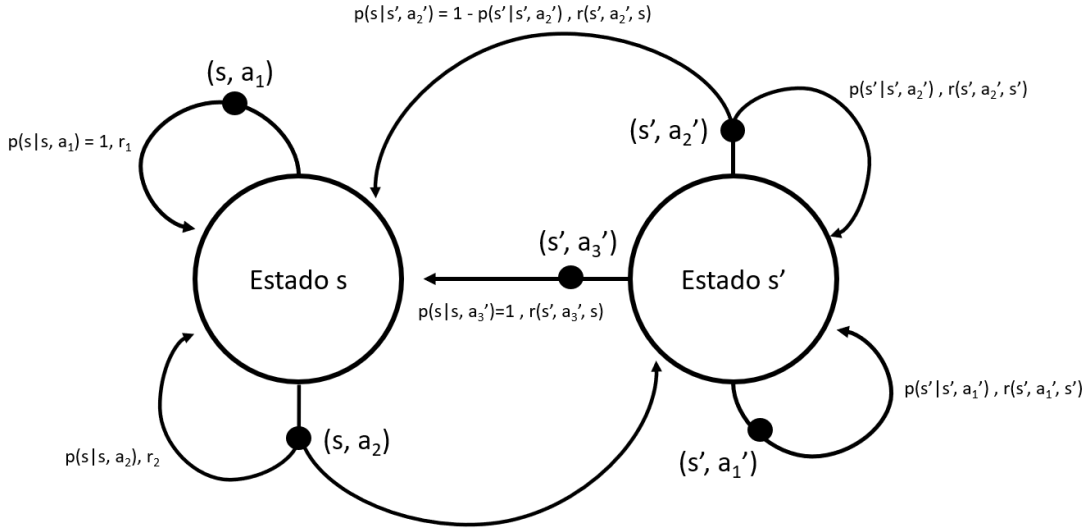

Figura 2.5: Gráfico de transición para un proceso de decisión de Markov.

2.2.4. Exploración y explotación

Un aspecto esencial cuando hablamos de aprendizaje por refuerzo es la relación existente entre exploración y explotación.

Ante un medio desconocido, está claro que es necesario ***explorar*** las distintas acciones que puedes ejecutar para tener un conocimiento sobre cuáles tienen mayor valor porque den lugar a mayores recompensas. Ahora bien, una vez tienes una cierta noción de algunas de las acciones que puedes tomar (pero no de todas, pues entonces el problema sería trivial), tienes que hacer una elección: puedes ceñirte a tomar la mejor de las opciones que ya conoces, en cuyo caso estarías ***explotando*** tu conocimiento del medio, o puedes ***explorar*** las acciones que aún no has tomado y que podrían dar lugar a mejores recompensas, aunque también a peores.

Puede demostrarse que en función del número de pasos que puedas ejecutar, será más conveniente decantarte por una u otra forma de actuar. Cuando tienes pocos intentos, la mejor opción será explotar, mientras que a lo largo de muchos intentos será más conveniente intercarlar la explotación con exploración [8].

En cada caso específico tomar la decisión entre explorar o explotar depende de una forma compleja del valor de las estimaciones, las incertidumbres y el número de pasos que aún puedas dar. Existen métodos muy sofisticados para hacer este balance para distintos problemas, aunque es cierto que muchos de estos métodos van de la mano de ciertas suposiciones que no son en absoluto triviales. A la hora de afrontar un problema de aprendizaje por refuerzo, será siempre necesario escoger un ratio adecuado para esta relación.

2.2.5. Valoración de las posibilidades.

Las ***funciones de valoración*** estiman cómo de óptimo (en términos de maximizar las recompensas futuras) es para el agente acceder a un cierto estado o ejecutar una acción determinada a partir de otro estado. Las recompensas obtenidas van a depender de las acciones que se tomen y, por lo tanto, las funciones de valoración estarán definidas respecto a políticas particulares de actuación $\pi_t(a|s)$, que como ya vimos anteriormente está determinado por las probabilidades de ejecutar una cierta acción a partiendo de un estado s.

Para cada política de actuación $\pi_t(a|s)$, definimos la valoración de los estados s como el valor esperado de dicho estado una vez de ha ejecutado la transición a él a través de

una acción a [8]. Formalmente, para procesos de Markov *MDP* se expresa como:

$$\nu_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s\right], \tag{2.9}$$

siendo $\mathbb{E}_\pi$ el valor esperado de una variable aleatoria suministrada por π_t y G_t la respuesta esperada, como ya vimos anteriormente. A la expresión anterior se la conoce como **función de estado-valor para la política** π.

También podemos hacer una definición semejante que además de estar condicionada al estado, lo esté también a la acción tomada desde él. Esta sería la **función de acción-valor para la política** π, y su expresión sería la siguiente:

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s, A_t = a\right]. \tag{2.10}$$

Estas expresiones cumplen una propiedad fundamental que se usa mucho en aprendizaje por refuerzo y que consiste en una relación recursiva particular: para cada política π y estado s se cumple la siguiente relación de consistencia entre el estado actual s y sus posibles estados sucesores s':

$$\begin{aligned}
\nu_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}\middle|S_t = s\right] \\
&= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)\left[r + \gamma\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+2}\middle|S_{t+1} = s'\right]\right] \\
&= \sum_a \pi(a|s)\sum_{s',r} p(s',r|s,a)\left[r + \gamma\nu_\pi(s')\right]
\end{aligned}$$

$$\nu_\pi(s) = \sum_a \pi(a|s)\sum_{s',r} p(s',r|s,a)\left[r + \gamma\nu_\pi(s')\right] \tag{2.11}$$

Esta ecuación es la ***ecuación de Bellman*** para ν_π, para la que ν_π es su única solución.

En ella se relaciona el valor esperado de un estado con el de los estados siguientes,

haciendo una suma sobre todos los posibles valores de las tres variables a, s' y r. para cada triplete de tales variables, se calcula la probabilidad de que se de ese triplete como la probabilidad de que tenga lugar la acción a partiendo del estado s a través de $\pi(a|s)$ por la probabilidad de que esa acción de lugar a un estado s' y a una recompensa s. Sumando para todas las posibilidades el producto de la probabilidad de ese triplete por la recompensa ponderada que se obtendría si tuviera lugar, obtenemos el valor esperado de las recompensas que obtendríamos si partiéramos de un estado s.

De esta forma, el valor esperado de un cierto estado s será el valor esperado descontado de sus estados siguientes sumado a la recompensa que se obtendría tras acceder a cada uno de ellos. Este planteamiento constituye una base desde la que operar con métodos de aprendizaje por refuerzo [8].

2.2.6. Función de valor óptimo

A la hora de escoger la política óptima a seguir, se considera que una política π es mejor o igual que otra π' si su valor esperado de retorno es mayor o igual a π' para todos sus estados.

$$\pi \geq \pi' \iff \nu_\pi(s) \geq \nu_{\pi'}(s), \quad s \in \mathbb{S}. \tag{2.12}$$

Siempre existe al menos una que sea mejor o igual a las demás. Al valor máximo de ν_π se la denota como $\nu_*(s)$ y viene dada por:

$$\nu_*(s) = max\, \nu_\pi(s), \quad \forall s \in \mathbb{S}, \tag{2.13}$$

pudiendo darse el caso de que más de una política compartan el mismo valor igualado al de la función óptima de valor de estado. Serán las políticas óptimas, que también deberán compartir el máximo valor en su función de valor acción:

$$q_*(s,a) = max\, \nu_\pi(s,a), \quad \forall s \in \mathbb{S}, \quad \forall a \in \mathbb{A}(s). \tag{2.14}$$

Cabe destacar que podemos escribir $q_*(s,a)$ en función de ν_* como:

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \nu_*(S_{t+1}) \middle| S_t = s, A_t = a \right]. \tag{2.15}$$

Como todas las funciones de valor de estado, nu_* sigue teniendo que cumplir la con-

dición de autoconsistencia dada por las *ecuación de Bellman*. No obstante, podremos escribir tal condición de una forma especial sin tener que referirnos a ninguna política π concretamente, ya que quedará implícito que se está tomando la mejor de las políticas. Esta forma de escribirse será la ***ecuación óptima de Bellman***. Es intuitivo considerar que el valor de un estado bajo la política óptima será el valor esperado recibido tras realizar la mejor acción posible a partir de un cierto estado [8]. Tendremos pues:

$$\nu_*(s) = \sum_{s',r} p(s', r|s, a) \left[r + \gamma \nu_*(s')\right], \tag{2.16}$$

$$q_*(s, a) = \sum_{s',r} p(s', r|s, a) \left[r + \gamma \, max \, q_*(s', a')\right]. \tag{2.17}$$

Ante un *MDP finito*, la ecuación óptima de Bellman tiene una solución única que será además independiente de la política π, de tal forma que para un problema en que existan N estados posibles, tendremos un sistema de ecuaciones con N ecuaciones y N incógnitas. Conociendo la dinámica p del medio, existirán múltiples métodos para solucionar sistemas no lineales de esta índole.

Capítulo 3

Aprendizaje por refuerzo cuántico

3.1. Introducción

En la sección anterior hemos estudiado los distintos elementos que componen el aprendizaje por refuerzo y hemos llegado a la *ecuación óptima de Bellman* como expresión fundamental. Resolver tal ecuación explícitamente nos permite obtener una política óptima y resolver en consecuencia la tarea a ejecutar a través de aprendizaje por refuerzo. No obstante, existe una serie de supuestos que no suelen cumplirse en la práctica y que hacen que hallar esta solución rara vez sea útil. El primero de ellos es que se parte del supuesto de que conocemos con exactitud la dinámica del entorno. Además, se asume que el proceso cumple la propiedad de Markov y que disponemos de los recursos computacionales suficientes para completar el cálculo, el cual realiza una búsqueda exhaustiva contemplando todas las posibilidades con todas sus probabilidades de ocurrir y todas sus recompensas esperadas. Concretamente, para un juego como el de *backgammon* tendríamos 10^{20} estados y un ordenador clásico tardaría miles de años en resolver la *ecuación de Bellman* para ν_* y q_* [8]. Otro factor limitante a tener en cuenta es la memoria disponible, ya que mucha memoria es necesaria para construir aproximaciones de las distintas funciones, políticas y modelos.

En consecuencia, un agente capaz de llegar a una política óptima es algo que en la práctica no va a darse habitualmente. Serían necesarios unos costes computacionales demasiado elevados.

Actualmente están en desarrollo distintas formas de combinar la computación cuánti-

ca con el Machine Learning. En primera instancia, esta unión parece razonable dadas las similitudes que tienen ambas disciplinas: por una parte, comparten su naturaleza estadística en lo que al tratamiento de la información respecta [2]. Ya citamos anteriormente en la sección de computación cuántica la imposibilidad de realizar una clonación de estados, cuestión que marcaba una diferencia intrínseca respecto a la computación clásica. No obstante, esta cuestión no resulta un impedimento en su aplicación al Machine Learning, puesto que cuando realizamos la clasificación o aprendizaje de variables aleatorias o de conceptos probabilísticos, en los que el objetivo es adjudicar a los datos la etiqueta que mejor se les ajuste, cuando entrenamos la red neuronal con una serie de ejemplos proporcionar dos ejemplos que son clones el uno del otro no te aporta ninguna información nueva [2]. Tanto en machine learning como en computación cuántica, se necesitarían infinitas muestras de una distribución aleatoria para poder tener el equivalente a una descripción clásica de tal sistema.

A la disciplina que combina Machine Learning y computación cuántica se la conoce como *Machine Learning Cuántico* (*QML* por sus siglas en inglés), y hoy en día podría identificarse tres tipos de problemas que pueden resolverse a través de ella [9] y que marcan tres formas de entender la disciplina:

- Tratamiento de sistemas cuánticamente a través de Machine Learning clásico. En esta clase de tratamientos se emplean datos y técnicas clásicas de aprendizaje para tratar un medio que es puramente cuántico.

- Tratando el *medio* clásicamente y al *agente* de forma cuántica, logrando aumentar la velocidad de toma de decisión del agente. Esto constituiría la adaptación de tareas propias del Machine Learning a ordenadores cuánticos.

- Tratando *medio* y *agente* cuánticamente.

Que el *medio* o el *agente* sean cuánticos implica que estos emitan estados cuánticos y superposiciones de ellos [9].

Centrándonos en el segundo caso, es una realidad que la mayoría de aplicaciones que encuentra la inteligencia artificial se desarrollan en un entorno físico clásico [10]. En estas circunstancias, es posible optimizar el proceso de selección de las acciones óptimas por parte del agente a través del empleo de algoritmos cuánticos como el ***algoritmo de Grover*** [11].

3.2. Algoritmo de Grover

El *algoritmo de Grover* es capaz de acelerar la velocidad de aprendizaje polinómicamente, mejorando así el rendimiento global. Será especialmente útil en aquellas tareas cuyas escalas de tiempo de cambio del estado del medio, que se presuponen siempre superiores al tiempo de deliberación del agente, sean próximas a estas [10].

Como dijimos en la sección de *aprendizaje por refuerzo*, cada vez que el agente tiene que escoger qué acción es más conveniente realizar, este ejecuta un algoritmo de búsqueda de camino aleatorio. Este algoritmo es el que podría realizarse a través de un algoritmo cuántico basado en la teoría de caminos cuánticos de tipo *Grover*.

Veamos en qué consiste tal algoritmo:

Partimos de $|A_s\rangle$, estado de acciones que pueden ser tomadas a partir de un cierto estado del medio s y que está formado por la superposición de todas las autoestados acciones que pueden tomarse $|a_n\rangle$, con $n \in N$, siendo N el número de autoestados de $|A\rangle$. Dicho estado tiene pues la siguiente forma:

$$|A_s\rangle = \sum_{n}^{N} C_n \, |a_n\rangle \, . \tag{3.1}$$

Este estado tendrá en cuenta la información sobre las recompensas esperadas para cada una de las acciones que pueden ejecutarse. Cuando se toma una medida de la acción que el agente toma, el estado colapsa a una de las autofunciones $|a_i\rangle$ con una probabilidad p_i que viene dada por el módulo al cuadrado de su coeficiente:

$$p_i = |\, \langle a_i|A\rangle \,|^2 = |\, \langle a_i| \sum_n C_n \, |a_n\rangle \,|^2 = |\, \sum_n C_n \, \langle a_i|a_n\rangle \,|^2 = |\, \sum_n C_n \delta(n-i)|^2 = |C_i|^2,$$

$$p_i = |C_i|^2. \tag{3.2}$$

Para preparar el estado $|A_s\rangle$ en computación cuántica podemos hacer uso de una *puerta Hadamard* [12] que actúe sobre m *qubits* preparados en el estado $|0\rangle$:

$$H^{\otimes m} \left| \overbrace{00\ldots0}^{m} \right\rangle = \frac{1}{\sqrt{2^m}} \left(\sum_{a_n=00\ldots0}^{\overbrace{11\ldots1}^{m}} |a_n\rangle \right). \tag{3.3}$$

Escritos de esta forma, observamos que $|A_s\rangle$ cuenta con $N = 2^m$ autoestados $|a_n\rangle$.

A continuación, procedemos a aplicarle el *algoritmo de Grover* a nuestro estado inicial ya en superposición. En primer lugar, resulta ilustrativo representar gráficamente la cuestión que se nos plantea. Nuestra misión es encontrar de entre todos los autoestados $|a_n\rangle$ aquel que de lugar a recompensas (estrictamente hablando sería aquel que diera lugar a la mayor recompensa esperada, pero supongamos por simplicidad que solo hubiera una acción premiable). A ese autoestado lo denominaremos $|win\rangle$, quedando la superposición de todos los demas $|a_n\rangle \neq |win\rangle$ como un estado ortogonal a él. A este último lo denotaremos como estado $|lose\rangle$. Nuestro estado inicial $|A\rangle$ puede expresarse en función de la base formada por los estados $|win\rangle$ y $|lose\rangle$ como:

$$|A_n\rangle = cos(\theta)\,|lose\rangle + sin(\theta)\,|win\rangle\,. \tag{3.4}$$

Pudiéndose además representar gráficamente de la siguiente forma:

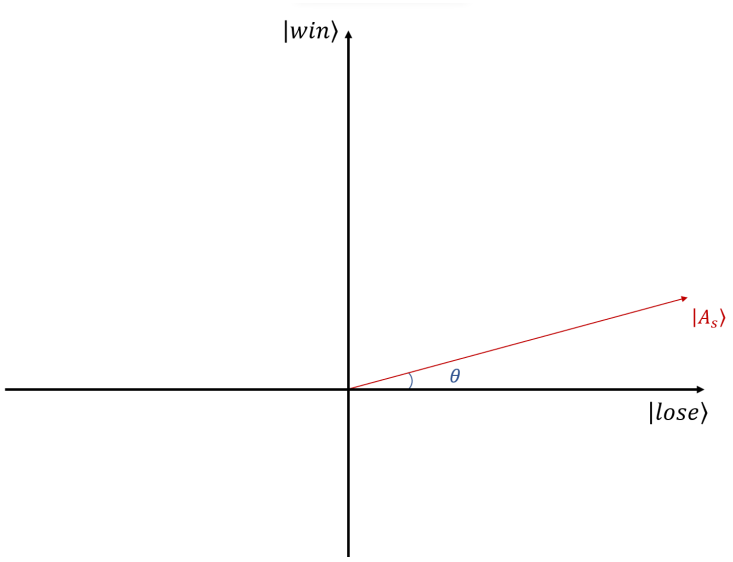

Figura 3.1: Estado inicial $|A_s\rangle$ en la base de estados $|win\rangle$ y $|lose\rangle$.

Donde θ es el ángulo que forma con el eje de abscisas. Al ser nuestro objetivo aumentar el coeficiente del autoestado $|win\rangle$, ejecutaremos el algoritmo de Grover para aproximar $|A_s\rangle$ al eje de ordenadas. Para ello, ejecutamos los siguientes pasos [9]:

1. Aplicamos un operador unitario U_E que cambie el *qubit* de recompensa del estado $|a_n\rangle$ cuando tal acción conlleve una recompensa. Reescribiendo:

$$|a_n\rangle = \left|a^{(n)}\right\rangle_A |i\rangle_R \ , \ i \in (0,1). \tag{3.5}$$

El operador actuaría como una puerta de control [11] tal que:

$$\begin{aligned} U_E \left|a^{(n)}\right\rangle_A |0\rangle_R &= \begin{cases} \left|a^{(n)}\right\rangle_A |1\rangle_R & si \quad r(a^{(n)}) > 0 \\ \left|a^{(n)}\right\rangle_A |0\rangle_R & si \quad r(a^{(n)}) = 0 \end{cases} , \\ U_E \left|a^{(n)}\right\rangle_A |1\rangle_R &= \begin{cases} \left|a^{(n)}\right\rangle_A |0\rangle_R & si \quad r(a^{(n)}) > 0 \\ \left|a^{(n)}\right\rangle_A |1\rangle_R & si \quad r(a^{(n)}) = 0 \end{cases} . \end{aligned} \tag{3.6}$$

Este operador provocará un cambio de signo sobre el estado $|win\rangle$ que buscamos. Esto puede demostrarse fácilmente a partir de la definición de (3.6) si expresamos el estado $|A_n\rangle$ que prepara el agente como:

$$|A_n\rangle = |\Psi\rangle_A |-\rangle_R = \Big[cos(\theta)\,|lose\rangle + sin(\theta)\,|win\rangle\Big]\Big[|0\rangle_R - |1\rangle_R\Big]/\sqrt{2}. \tag{3.7}$$

Observamos que, a diferencia de en la expresión (3.4), aquí no incluímos el estado de las recompensas dentro del de las acciones $|win\rangle$ y $|lose\rangle$. Aplicándole el operador U_E:

$$\begin{aligned} U_E\,|A_n\rangle &= U_E\left\{cos(\theta)\,|lose\rangle\left[|0\rangle_R - |1\rangle_R\right]/\sqrt{2} + sin(\theta)\,|win\rangle\left[|0\rangle_R - |1\rangle_R\right]/\sqrt{2}\right\} \\ &= cos(\theta)\,|lose\rangle\left[|0\rangle_R - |1\rangle_R\right]/\sqrt{2} + sin(\theta)\,|win\rangle\left[|1\rangle_R - |0\rangle_R\right]/\sqrt{2} \\ &= cos(\theta)\,|lose\rangle\left[|0\rangle_R - |1\rangle_R\right]/\sqrt{2} - sin(\theta)\,|win\rangle\left[|0\rangle_R - |1\rangle_R\right]/\sqrt{2} \\ &= \Big[cos(\theta)\,|lose\rangle - sin(\theta)\,|win\rangle\Big]\Big[|0\rangle_R - |1\rangle_R\Big]/\sqrt{2}. \end{aligned} \tag{3.8}$$

Continuando con nuestra representación gráfica, la actuación de este operador puede verse como la reflexión de $|A_n\rangle$ sobre el eje de abscisas:

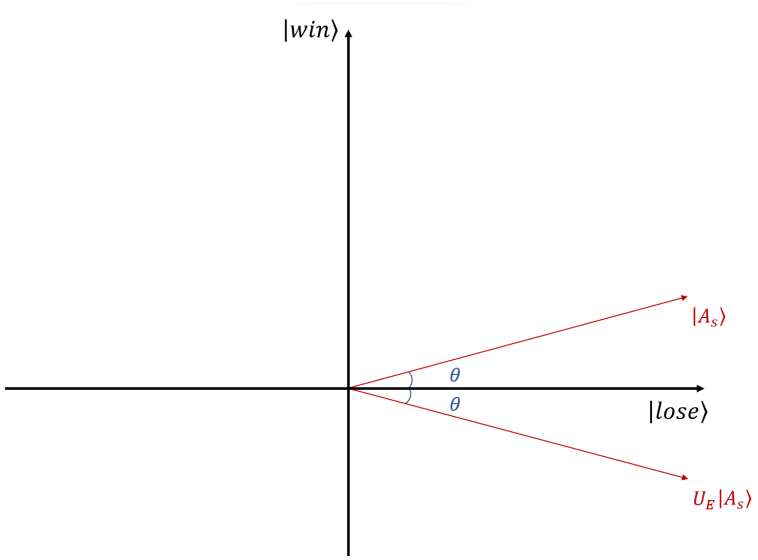

Figura 3.2: Actuación del operador U_E sobre el estado inicial $|A_n\rangle$.

2. El siguiente paso es hacer una reflexión del nuevo estado $U_E |A_n\rangle$ sobre el estado original $|A_n\rangle$ a través del operador U_R:

$$U_R = 2 |\Psi\rangle \langle A_n| - \mathbb{I}_A, \tag{3.9}$$

siendo $|\Psi\rangle$ el estado sobre el que actúa, en este caso $U_E |A_n\rangle$ y $|A_n\rangle$ el estado inicial [9]. Gráficamente, llegaríamos a:

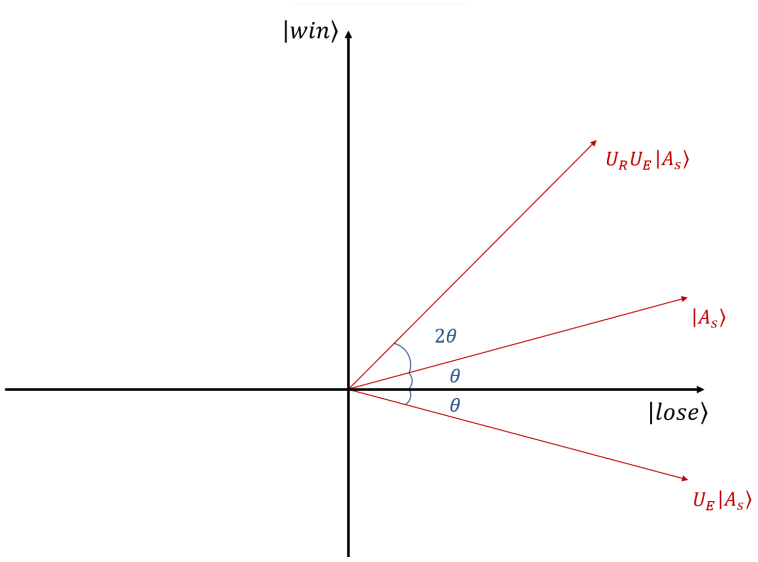

Figura 3.3: Actuación del operador U_R sobre el estado $U_E |A_n\rangle$.

Al realizar estos pasos hemos conseguido un aumento del coeficiente sobre el autoestado $|win\rangle$. Podemos realizar estos dos pasos, que son los que constituyen el *algoritmo de Grover*, sucesivamente obteniendo:

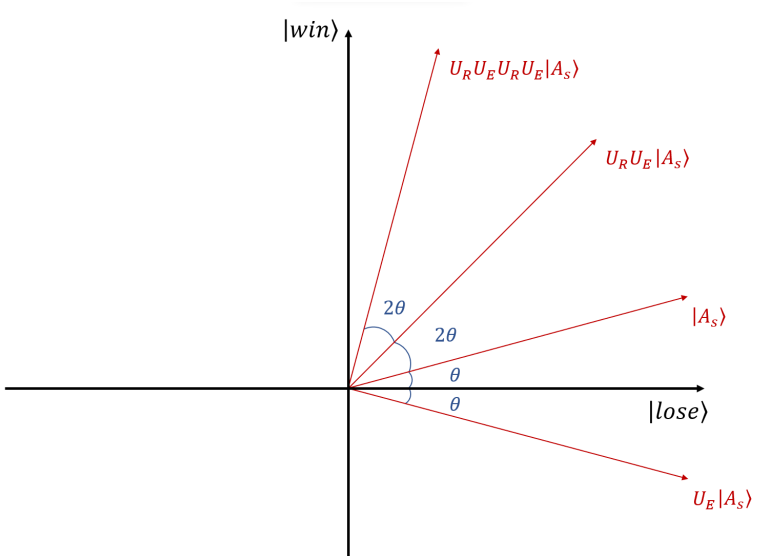

Figura 3.4: Actuación sucesiva del *algoritmo de Grover*.

Ahora bien, si no sabemos en qué momento dejar de aplicar este algoritmo, podríamos llegar a una situación como la siguiente en la que el coeficiente comience a disminuir:

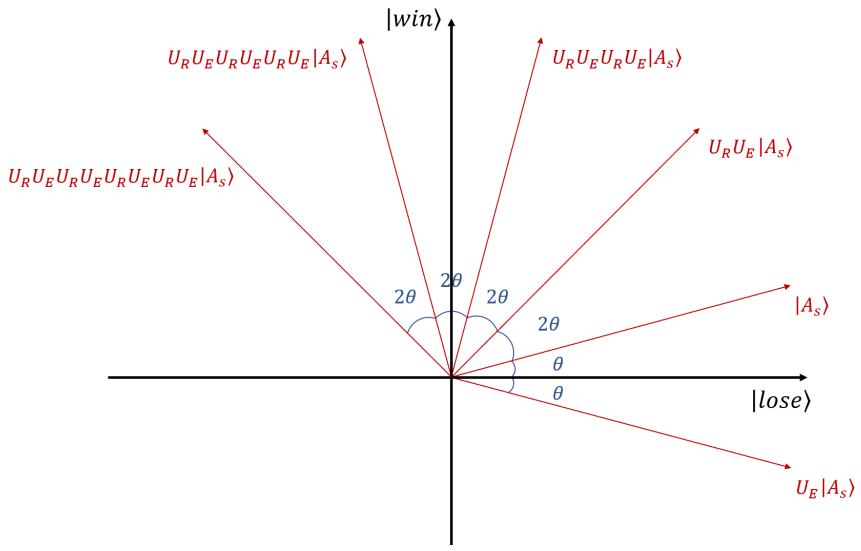

Figura 3.5: Actuación sucesiva del *algoritmo de Grover*.

Para saber cuántas iteraciones k debemos realizar antes de parar el algoritmo, imponemos lo siguiente [9]:

$$\theta + k\,2\theta \approx \frac{\pi}{2}, \tag{3.10}$$

donde, si tenemos un alto número de *qubits* N, inicialmente contaremos con una amplitud $\langle win|A_n\rangle = 1/\sqrt{N}$ muy pequeña, con lo que el ángulo θ será muy pequeño y podremos aproximar $\theta \approx \sin(\theta)$ y por lo tanto a $\theta \approx \sin(\theta) = 1/\sqrt{N}$. Volviendo sobre la expresión (3.10), llegaremos a:

$$k \approx \sqrt{N}\frac{\pi}{4}, \tag{3.11}$$

como el número de veces que hemos de iterar el algoritmo para acercarnos lo máximo posible al autoestado $|win\rangle$ y tener así la mayor posibilidad posible de que se ejecute la acción ganadora.

Es precisamente esta expresión la que nos permite confirmar la aceleración que produce este algoritmo cuántico, pues son necesarios un número de iteraciones del orden de $\sqrt{N}$ para hallar la solución, frente al caso clásico en que una media de $N/2$ intentos serían necesarios para encontrar el autoestado $|win\rangle$.

3.3. Agente híbrido

De acuerdo con lo visto anteriormente, para un mismo número de iteraciones, el algoritmo cuántico de Grover nos proporcionará mayores recompensas por reducir este polinómicamente la cantidad de iteraciones necesarias para obtenerlos. No obstante, el coeficiente máximo acompañante al autoestado $|win\rangle$ que puede conseguirse está limitado por la naturaleza sinusoidal que presenta [11]. En el caso clásico, el crecimiento del coeficiente con el número de iteraciones es más lento, pero es siempre creciente convergiendo hacia un valor máximo, tal y como se observa en la siguiente imagen.

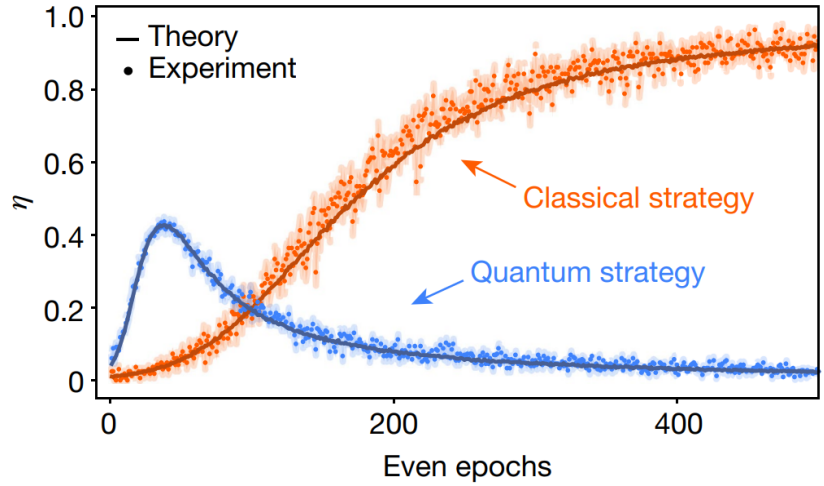

Figura 3.6: Representación de la recompensa esperada η en función del número de iteraciones calculado teóricamente con $n = 10000$ agentes y experimentalmente con $n = 165$ agentes mediante el empleo de agentes cuánticos y clásicos. [11]

Es interesante considerar el caso en que el agente sea *híbrido*, es decir, que combine algoritmos cuánticos y clásicos, con el fin de combinar las propiedades ventajosas de ambos. Una forma de hacer esto sería proceder a través del algoritmo cuántico hasta que este alcance la máxima recompensa que puede obtenerse mediante él, para después continuar con un algoritmo puramente clásico.

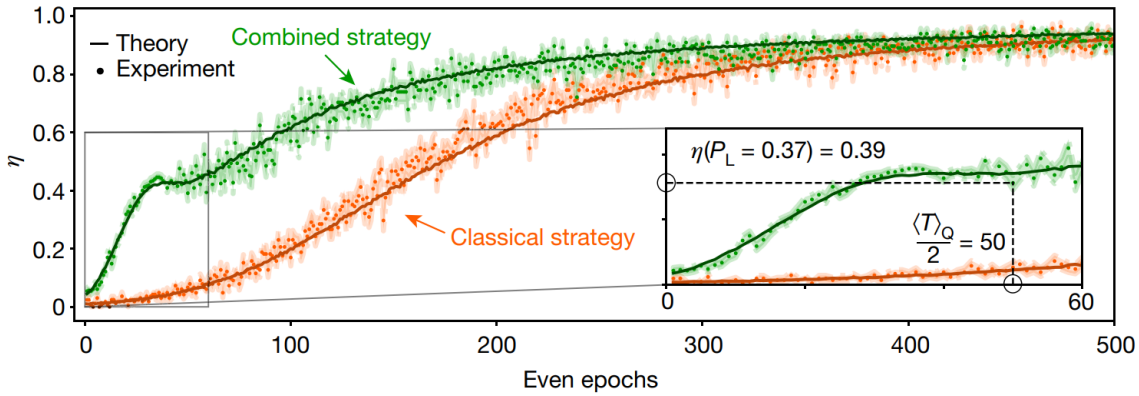

Figura 3.7: Representación de la recompensa esperada η en función del número de iteraciones calculado teóricamente con $n = 10000$ agentes y experimentalmente con $n = 165$ agentes mediante el empleo de agentes híbridos y clásicos. [11]

De esta forma, se han obtenido muy satisfactorios resultados en el plano teórico y experimental, tal y como podemos observar en el ejemplo de la figura superior.

Capítulo 4

Conclusiones

A lo largo de este trabajo se han introducido los distintos fundamentos de la computación cuántica y del aprendizaje por refuerzo, con el fin de buscar sus puntos en común y analizar las posibilidades a las que da lugar la combinación de ambas disciplinas. Existen distintas maneras de combinarlas, resultando la manera más intuitiva de distinguir unas de otras según de qué datos se hace un tratamiento cuántico.

El tratamiento cuántico de todos los elementos de una tarea a realizar: agente, medio, interacción, etc. pudiera parecer la forma óptima de aprovechar todas las ventajas que la computación cuántica parece ofrecer a priori. No obstante, plantear la resolución de una tarea de esta forma limita en exceso su aplicación, ya que la mayoría de las tareas que queremos resolver en nuestro entorno macroscópico no cuentan con ninguno de tales factores (agente, medio, ...) en condiciones tales que se manifiesten cuánticamente. En consecuencia, el uso de un tratamiento de este estilo para el aprendizaje cuántico por refuerzo quedará limitado al ámbito de la experimentación cuántica a nivel de laboratorio. Cabe decir que grandes progresos se han conseguido en esta dirección gracias a la incorporación de la inteligencia artificial al campo de la física cuántica, espcialmente en aspectos relacionados con algunos de los problemas que ofrece la construcción de un ordenador cuántico funcional: procesado de señales cuánticas, metrología cuántica, estimación de Hamiltonianos, problemas relacionados con el control cuántico, reducción del ruido, etc. Aunque también ha llevado a avances en ciertos ámbitos científicos concretos como el de la física de la materia condensada a través de por ejemplo el problema de varios cuerpos, o como el del diseño de experimentos ópticos cuánticos [2].

Por plantear un mayor abanico de posibilidades, los esfuerzos se han centrado espe-

cialmente en buscar la forma de acelerar y optimizar la ejecución de tareas de aprendizaje sobre elementos clásicos a través de la computación cuántica. Esto requiere cifrar datos clásicos que nos ofrecerá el medio de tal forma que puedan ser interpretados por sistemas cuánticos, desde los que unos algoritmos cuánticos puedan procesarlos para finalmente ser descodificados de nuevo hacia una forma clásica para que el agente clásico pueda ejecutar la acción estimada como óptima [7].

Es en este tipo de tareas en las que hemos centrado este trabajo y, si bien como ya hemos dicho, es en esta dirección hacia la que se concentra la investigación del aprendizaje por refuerzo cuántico, la realidad de este ámbito es que los recursos que son necesarios hoy en día para realizar la tarea de codificación y descodificación de información entre sistemas clásicos y cuánticos son tan elevados que llega a ponerse en cuestión la aceleración cuántica que pueda conseguirse en la práctica [7]. No obstante, esto no impide que el campo del aprendizaje cuántico por refuerzo haya recibido una creciente atención en los últimos años, especialmente para combatir el limitante de resolver la *ecuación de Bellman* para sistemas con muchos estados que, como ya dijimos en secciones anteriores, tardarían miles de años en ser resueltos a través de ordendores clásicos.

Por todo lo discutido, podemos llegar a la conclusión de que incluso pese al inconveniente que representa la construcción de ordenadores cuánticos por todas las dificultades que plantea su puesta en funcionamiento, merece la pena emplear esfuerzos y recursos en progresar hacia una adaptación de la inteligencia artificial en ordenadores cuánticos por las múltiples ventajas que presenta: no estamos hablando tan solo de un aumento de la velocidad en la ejecución de ciertas tareas [2] que por su propia naturaleza necesitarían una respuesta en un tiempo mínimo, como podría ser el tiempo de reacción de un coche autónomo ante un obstáculo imprevisto, sino también en un aumento de la velocidad que puede permitir que se resuelvan tareas que hoy en día son irresolubles únicamente por la cantidad de tiempo que requiere ejecutarlas [1].

Bibliografía

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, jun 2018.

[3] Shang Yu, Francisco Albarrán-Arriagada, Juan Carlos Retamal, Yi-Tao Wang, Wei Liu, Zhi-Jin Ke, Yu Meng, Zhi-Peng Li, Jian-Shun Tang, Enrique Solano, Lucas Lamata, Chuan-Feng Li, and Guang-Can Guo. Reconstruction of a photonic qubit state with reinforcement learning. *Advanced Quantum Technologies*, 2(7-8):1800074, mar 2019.

[4] William K. Wootters and Wojciech H. Zurek. The no-cloning theorem. *Physics Today*, 62(2):76–77, feb 2009.

[5] Andrew M. Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1–52, jan 2010.

[6] MinnaLearn and the University of Helsinki. Elements of ai. 2018.

[7] F. Albarrán-Arriagada, J. C. Retamal, E. Solano, and L. Lamata. Measurement-based adaptation protocol with quantum reinforcement learning. *Physical Review A*, 98(4):042315, oct 2018.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[9] V. Saggio. Quantum-enhanced reinforcement learning. Vienna Center for Quantum Science and Technology (VCQ), University of Vienna, Austria, 2021. Online Seminar. `https://youtu.be/wLYmrJtXneY`.

[10] Giuseppe Davide Paparo, Vedran Dunjko, Adi Makmal, Miguel Angel Martin-Delgado, and Hans J. Briegel. Quantum speedup for active learning agents. *Physical Review X*, 4(3):031002, jul 2014.

[11] V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund, S. Wölk, H. J. Briegel, and P. Walther. Experimental quantum speed-up in reinforcement learning agents. *Nature*, 591(7849):229–233, mar 2021.

[12] Daoyi Dong, Chunlin Chen, Hanxiong Li, and Tzyh-Jong Tarn. Quantum reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(5):1207–1220, oct 2008.