Programación Python para Big Data - Tarea lección 9

Kevin Martínez García

6 de julio de 2022

Índice

1. Introducción

En la actividad correspondiente a la lección 9 se nos pidió realizar una serie de ejercicios en relación con la computación cuántica. En la primera actividad realizaremos un resumen en términos generales sobre qué es la comutación cuántica y en las siguientes trataremos aspectos máas concretos como puertas lógicas, algoritmos etc.

2. Actividad 1 - Resumen de computación cuántica

En términos generales podríamos definir la computación cuántica como un tipo de computación que hace uso de las propiedades de los estados cuánticos tales como el entrelazamiento, la superposición y la interferencia para realizar cálculos. Un computador convencional utiliza bits como unidad básica de información y, como sabemos, los bits únicamente pueden tener dos estados opuestos (0 ó 1). En computación cuántica se hace uso de bits cuánticos o qubits, es decir, unidades básicas de información que pueden estar en una superposición de estados (una superposición de 0 ó 1). Adjuntamos en la Figura 1 [1] una tabla con algunas implementaciones físicas de los qubits.

Physical support	Name	Information support	$\|0\rangle$	$\|1\rangle$
Photon	Polarization encoding	Polarization of light	Horizontal	Vertical
	Number of photons	Fock state	Vacuum	Single photon state
	Time-bin encoding	Time of arrival	Early	Late
Coherent state of light	Squeezed light	Quadrature	Amplitude-squeezed state	Phase-squeezed state
Electrons	Electronic spin	Spin	Up	Down
	Electron number	Charge	No electron	One electron
Nucleus	Nuclear spin addressed through NMR	Spin	Up	Down
Optical lattices	Atomic spin	Spin	Up	Down
Josephson junction	Superconducting charge qubit	Charge	Uncharged superconducting island (Q=0)	Charged superconducting island (Q=2e, one extra Cooper pair)
	Superconducting flux qubit	Current	Clockwise current	Counterclockwise current
	Superconducting phase qubit	Energy	Ground state	First excited state
Singly charged quantum dot pair	Electron localization	Charge	Electron on left dot	Electron on right dot
Quantum dot	Dot spin	Spin	Down	Up
Gapped topological system	Non-abelian anyons	Braiding of Excitations	Depends on specific topological system	Depends on specific topological system
Vibrational qubit[10]	Vibrational states	Phonon/vibron	$\|01\rangle$ superposition	$\|10\rangle$ superposition
van der Waals heterostructure[11]	Electron localization	Charge	Electron on bottom sheet	Electron on top sheet

Figura 1: Implementación física de los qubits

Una de las primeras cuestiones que puede surgir cuando hablamos de computación cuántica es el concepto de qubit y como se materializa en la realidad. Es decir, podemos preguntarnos cómo es un qubit en realidad o cómo podemos generar uno. Un ejemplo de qubit podría ser el *espín* de un electrón [2]. Un electrón es una partícula subatómica que orbita alrededor de los núcleos de los átomos y que posee carga eléctrica negativa. Una de las propiedades de los electrones es su espín, que es una medida de su momento angular intrínseco (lo podríamos entender como la cantidad de movimiento rotacional del electrón, aunque en realidad es más complicado nos sirve de momento). El espín de los electrones puede ser $\frac{1}{2}$ o $-\frac{1}{2}$ pero, antes de ser observado este estado será una superposición de ambos por lo que podría ser empleado como qubit. Algunos experimentos [3] proponen usar los electrones de la capa de valencia de un átomo de fósforo como qubits y el uso de campos magnéticos para forzar la interacción entre dichos electrones (y por tanto entre los qubits).

Sin embargo, el usar partículas como los electrones como qubits plantea una problemática muy clara y es el lograr mantener el estado del qubit durante un intervalo de tiempo lo suficientemente prolongado como para que pueda resultar útil en algún calculo. El *espín* de un electrón está sujeto a

variaciones constantes por el simple hecho de interaccionar con partículas cercanas o por las vibraciones que se producen al encontrarse a temperaturas elevadas. Esta problemática no es exclusiva de los electrones y podría extenderse a cualquier entidad cuántica sujeta a variar su estado por las interacciones con su entorno. Por tanto, una de las principales problemáticas que surgen con la creación y mantenimiento de qubits es la necesidad de evitar este tipo de alteraciones en su estado.

Uno de los procedimientos más utilizados consiste en someter al soporte físico del qubit a temperaturas extremadamente bajas (en concreto todo lo cerca que sea posible del 0 Kelvin) para tratar de reducir la energía interna de dicho soporte. Evidentemente todo lo que hemos explicado hasta el momento implica que un ordenador cuántico completamente funcional no estaría disponible para cualquiera y muchísimo menos en un entorno doméstico. Es por eso que, en la actualidad, únicamente las grandes élites tecnológicas cuentan con ordenadores cuánticos con un número considerable de qubits (Google, IBM, etc.).

Si bien puede resultar excesivamente teórico, también resulta interesante examinar los computadores cuánticos desde el punto de vista de la computabilidad y la complejidad. En cuanto a la computabilidad, los computadores cuánticos son capaces de resolver los mismos problemas que los computadores clásicos y pueden ser simulados mediante una máquina de Turing. Es decir, el modelo de los computadores cuánticos es equivalente al de las máquinas de Turing y, por tanto, no pueden resolver problemas no decidibles como el problema de parada.

Por otra parte, en cuanto a la cuestión de la complejidad, se trata de una área con un desarrollo muy temprano en cuanto a la computación cuántica, sin embargo, existen algunas conclusiones y sospechas generales que resulta interesante comentar. Se denota como **BQP** *bounded error, quantum, polynomial time* a la clase de los problemas que pueden resolverse en tiempo polinómico utilizando una máquina cuántica de Turing. En concreto, se sospecha que $\mathbf{P} \subseteq \mathbf{BQP} \subseteq \mathbf{PSPACE}$. En otras palabras, todos los problemas que pueden ser resueltos eficientemente por un ordenador clásico determinista también pueden ser resueltos eficientemente por un ordenador cuántico, y todos los problemas que pueden ser resueltos eficientemente por un ordenador cuántico también pueden ser resueltos por un ordenador clásico determinista con recursos espaciales polinomiales. Además, se sospecha que **BQP** es un superconjunto estricto de **P**, lo que significa que hay problemas que pueden ser resueltos de forma eficiente por los ordenadores cuánticos y que no pueden ser resueltos de forma eficiente por los ordenadores clásicos deterministas [4].

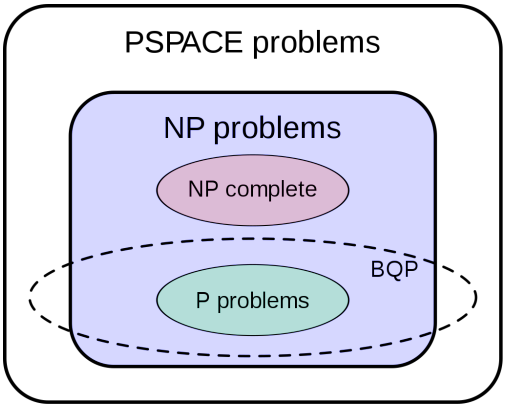

Figura 2: BQP en el mapa de complejidad

Por ejemplo, la factorización de números enteros (un problema de especial interés en la criptografía y la ciberseguridad) puede resolverse de forma más eficiente (con menor coste temporal) mediante computadores cuánticos y sus propiedades. En resumen, los computadores cuánticos pueden resolver de forma eficiente problemas que los computadores clásicos no son capaces, pero no aportan nada nuevo en cuanto a los problemas no decidibles.

Para terminar con esta primera sección, simplemente comentar que por lo que hemos visto hasta el momento, parece que la computación cuántica se mantendrá, al menos durante un tiempo, como una tecnología accesible a unos pocos y con unas aplicaciones muy concretas. Algunos científicos se mantienen dudosos y escépticos frente a la denominada *supremacía cuántica*, es decir, llegar a demostrar que un dispositivo cuántico puede resolver problemas que un computador clásico no podría resolver en un tiempo asumible [5]. Otros científicos apuntan a los problemas derivados de la corrección de errores que requieren los computadores cuánticos y señalan como, al escalar a sistemas con números elevados de qubits, se podría convertir en algo lo suficientemente significativo como para impedir su progreso. En definitiva, la computación cuántica requiere de más investigación y trabajo por parte de la comunidad científica para descrubrir si verdaderamente tiene el potencial de sustituir al computador clásico. En posteriores secciones mostraremos algunos algoritmos que presentan resultados prometedores en este aspecto.

3. Actividad 2 - Cuenta en IBM Quantum Computing

Adjunto una captura con la página principal de IBM después de crear mi cuenta.

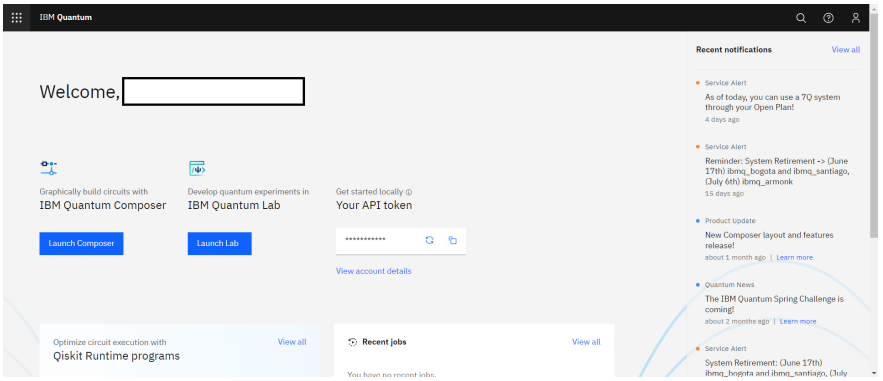

Figura 3: Página principal IBM

4. Actividad 3 - Puertas lógicas digitales

Las puertas lógicas son dispositivos electrónicos que tienen aplican sobre los bits de un circuito funciones de tipo booleano (como la negación, la conjunción o la disyunción). Las puertas lógicas componen los circuitos de conmutación en los chips y son esenciales para el funcionamiento de la electrónica actual. A continuación, vamos a explicar brevemente el funcionamiento de las puertas lógicas más comunes, incluyendo una imagen de cada uno así como su tabla de verdad.

4.1. Puerta lógica NOT

La puerta lógica NOT calcula la negación lógica de un bit, es decir, si está a 1 lo pone a 0 y viceversa.

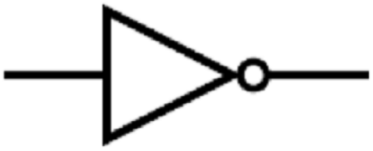

Figura 4: Puerta lógica NOT

Input	Output
0	1
1	0

Cuadro 1: Tabla de verdad puerta NOT

4.2. Puertas lógicas OR y NOR

La puerta lógica OR calcula la disyunción lógica $\vee$ de dos bits, es decir, esta puerta si que toma dos inputs en lugar de uno.

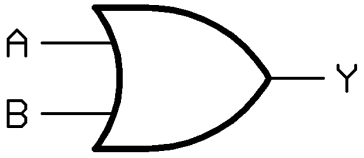

Figura 5: Puerta lógica OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Cuadro 2: Tabla de verdad puerta OR

Por otra parte, la puerta lógica NOR es la negación en la salida de la puerta lógica OR. A nivel práctico, se podría entender como una puerta OR que, después de su salida tiene una puerta NOT.

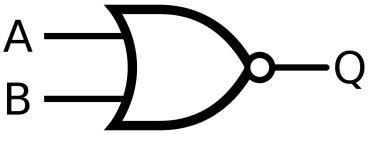

Figura 6: Puerta lógica NOR

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

Cuadro 3: Tabla de verdad puerta NOR

4.3. Puertas lógicas AND y NAND

La puerta lógica AND calcula la conjunción lógica $\wedge$ de dos bits, es decir, esta puerta si que toma dos inputs en lugar de uno.

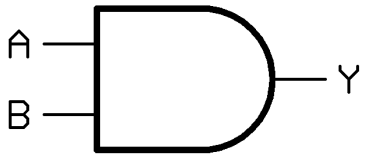

Figura 7: Puerta lógica AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Cuadro 4: Tabla de verdad puerta AND

Por otra parte, la puerta lógica NAND es la negación en la salida de la puerta lógica AND. A nivel práctico, se podría entender como una puerta AND que, después de su salida tiene una puerta NOT.

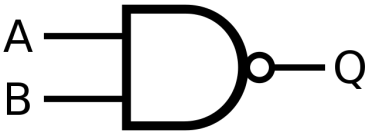

Figura 8: Puerta lógica NAND

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

Cuadro 5: Tabla de verdad puerta NAND

4.4. Puerta lógica XOR

La puerta lógica XOR implementa la operación "o exclusivo" es decir, la salida de esta puertaresulta cierta sí uno y sólo uno de sus inputs es verdadero.

Figura 9: Puerta lógica XOR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Cuadro 6: Tabla de verdad puerta XOR

5. Actividad 4 - Puertas lógicas cuánticas

Las puertas lógicas cuánticas son circuitos cuánticos que actúan sobre un número reducido de bits cuánticos. Al igual que las puertas lógicas digitales, constituyen los bloques básicos para construir cualquier circuito cuántico. Antes de proceder a explicar las puertas cuánticas más comunes es conveniente comprender dos conceptos. Por una parte, a nivel formal las puertas cuánticas se representan mediante matrices unitarias de orden $2^n \times 2^n$ siendo n el número de bits cuánticos sobre los que actúan.

Por otra parte, para obtener el resultado de aplicar una puerta cuántica sobre un bit cuántico, hay que calcular el producto matriz-vector del portón cuántico con el estado del bit (vamos a empezar con un bit y luego veremos que ocurre con puertas que actúan sobre más de uno). Como hemos visto en sesiones de teoría, un bit cuántico es una superposición de estados (decimos que antes de ser observado se encuentra en una superposición de 0 y 1, para colapsar cuando es observado en alguno de los dos estados). Estos estados se representan con notación *bra-ket* como aparece en la ecuación (1) a continuación [6].

$$|q_0\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

Siendo α y β coeficientes complejos. Vemos que en función de los valores de α y β el estado del bit cuántico q_0 puede ser "más cercano" a 0 o a 1. Aplicar una puerta cuántica sobre un qubit no es más que calcular el siguiente producto. Sea A una puerta cuántica cuañquiera y $|q\rangle$ un estado cuántico arbitrario, el resultado de aplicar A sobre $|q\rangle$ es:

$$A|q\rangle = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix} = \begin{pmatrix} aq_0 + bq_1 \\ cq_0 + dq_1 \end{pmatrix} \tag{2}$$

Otro aspecto importante a tener en cuenta es que ocurre cuando tenemos puertas que actúan sobre más de un bit cuántico. En ese caso, debemos tener en cuenta el estado que surge por

el *entrelazamiento cuántico* entre ambos. Si tenemos dos qubits con estados $|a\rangle$ y $|b\rangle$ el estado resultante de su entrelazamiento se calcula mediante el *producto tensorial* de ambos [7].

$$|ab\rangle = |a\rangle \otimes |b\rangle = \begin{pmatrix} a_0 \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \\ a_1 \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{pmatrix} \tag{3}$$

Para calcular la acción de una puerta sobre este estado simplemente calcularíamos el producto matriz-vector como hemos visto en la ecuación (2). Una vez vistos estos conceptos podemos comenzar a examinar los diferentes portones.

5.1. Hadamard gate

El portón de Hadamard actúa sobre un único bit cuántico y crea un estado de superposición con igual probabilidad para 0 o 1. La matriz de este portón es la que aparece en la ecuación a continuación.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

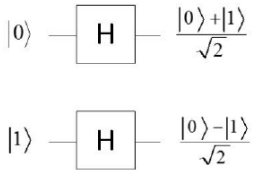

Figura 10: Puerta cuántica Hadamard

5.2. Pauli gates

Los portones de Pauli actúan sobre un único qubit y distinguimos los tipos Pauli-X, Pauli-Y y Pauli-Z. A continuación mostramos sus respectivas matrices junto con la representación de cada uno en un circuito [8].

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

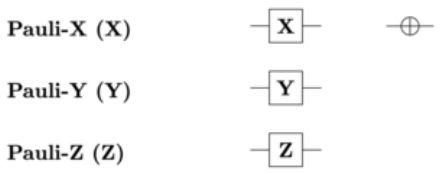

Figura 11: Puertas de Pauli

5.3. CNOT

El portón CNOT actúa sobre dos qubits y simplemente realiza la negación del primero siempre que el segundo se encuentre en estado $|1\rangle$ (en cualquier otro caso no altera su estado). Adjuntamos la matriz (que al actuar sobre dos qubits será de dimensión 4×4 como hemos visto) y la ilustración del portón en un circuito.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

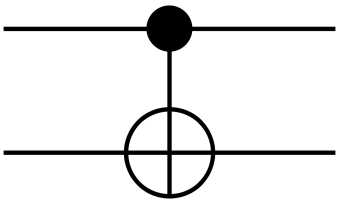

Figura 12: Puerta CNOT

5.4. Toffoli gate

La puerta Toffoli (CCNOT) actúa sobre tres bits cuánticos. Si los dos primeros bits se encuentran en estado $|1\rangle$ entonces invierte el tercero, en caso contrario no altera su estado. La matriz resultante de esta puerta es de dimensiones 9×9 por lo que no la adjuntaremos, si que incluimos una imágen con la represnetación del portón en un circuito cuántico.

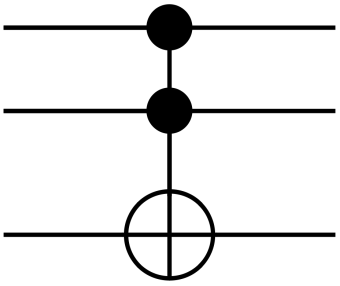

Figura 13: Puerta de Toffoli

6. Actividad 5 - La esfera de Bloch

La *esfera de Bloch* es una representación geométrica del estado de un qubit. Una visualización de esta esfera es la que aparece en la Figura a continuación. Como puede verse los polos de la esfera se corresponden con los estados base $|0\rangle$ y $|1\rangle$. Cualquier estado intermedio entre ambos puede representarse como un vector apuntando a cualquier otro punto dentro de la esfera. Resulta también interesante comentar que el hecho de que las puertas cuánticas vistas en el ejercicio anterior suponen alterar el estado de un qubit o lo que es lo mismo, rotar y/o desplazar la representación de dicho qubit en la esfera de Bloch. Por ejemplo, un qubit en estado $|0\rangle$ apunta hacia "arriba" en la

esfera, si le aplicásemos una puerta Pauli-X, invertiríamos su estado transformándolo en el vector $|1\rangle$ que apuntaría hacia "abajo".

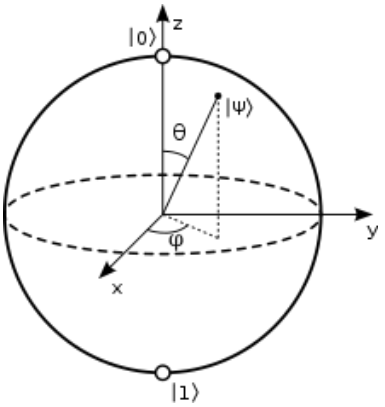

Figura 14: Esfera de Bloch

7. Actividad 6 - Números Complejos

El cuerpo de los complejos es una extensión de los números reales que se caracteriza en esencia por contener todas las raíces de los polinomios. Un número complejo $z \in \mathbb{C}$ está formado por una parte real (que denotaremos en general como a) y una parte imaginaria (que denotaremos en general como ib) tal y como aparece en la ecuación (4) a continuación.

$$z = a + ib$$

Siendo i el número imaginario que se define como $i = \sqrt{-1}$ En el campo de la mecánica cuántica resultan de especialidad utilidad para abstraer ciertas realidades y, como ya hemos visto, un vector cuántico de estados de dimensión n puede pertenecer a $\mathbb{C}^n$.

8. Actividad 7 - Algoritmos cuánticos

Un algoritmo llamativo dentro de la computación cuántica es el denominado algoritmo de Grover. El algoritmo de Grover es un algoritmo de búsqueda no estructurada que toma como entrada una colección de N datos y devuelve como resultado un elemento ω denominado "ganador", es decir, el dato que cumple cierta propiedad que buscamos. No entraremos en los detalles de la implementación del algoritmo, pero si competiremos que tiene cota $\mathbf{O}(\sqrt{N})$ lo que supone cierta mejora con respecto a la cota $\mathbf{O}(n)$ propia de los computadores clásicos [9].

Otro algoritmo interesante es el Algorithmo de Shor que permite factorizar un número en sus factores primos (lo cual resulta de gran interés en criptografía y ciberseguridad). De nuevo, no entraremos en detalles del funcionamiento pero, para cierto número N con d números decimales, el algoritmo de Shor factoriza dicho N en un tiempo polinomial con d, más información en la fuente biblográfica incluida [10].

Referencias

[1] Wikipedia Contributors (2019). Qubit. [online] Wikipedia. Available at: `https://en.wikipedia.org/wiki/Qubit`.

[2] Wikipedia.org. (2017). Electrón. [online] Available at: https://es.wikipedia.org/wiki/Electr%C3%B3n.

[3] MIT Technology Review. (n.d.). The Phosphorous Atom Quantum Computing Machine. [online] Available at: https://www.technologyreview.com/2013/05/22/178359/the-phosphorous-atom-quantum-computing-machine/.

[4] Wikipedia Contributors (2019). Quantum computing. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Quantum_computing.

[5] Wikipedia. (2020). Quantum supremacy. [online] Available at: https://en.wikipedia.org/wiki/Quantum_supremacy.

[6] community.qiskit.org. (n.d.). Representing Qubit States. [online] Available at: https://qiskit.org/textbook/ch-states/representing-qubit-states.html.

[7] community.qiskit.org. (n.d.). Multiple Qubits and Entangled States. [online] Available at: https://qiskit.org/textbook/ch-gates/multiple-qubits-entangled-states.html.

[8] community.qiskit.org. (n.d.). Single Qubit Gates. [online] Available at: https://qiskit.org/textbook/ch-states/single-qubit-gates.html.

[9] community.qiskit.org. (n.d.). Grover's Algorithm. [online] Available at: https://qiskit.org/textbook/ch-algorithms/grover.html.

[10] IBM Quantum. (n.d.). Shor's algorithm. [online] Available at: https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm.