



Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Física

UNA INTRODUCCIÓN A LA COMPUTACIÓN CUÁNTICA APLICADA AL APRENDIZAJE AUTOMÁTICO

Autora: Clara Georgina Stampa Guilarte

Tutores: Luis Miguel Nieto Calzada | Mariano Caruso

Octubre 2023

Índice general

Lista de figuras	5
Lista de tablas	7
Introducción	9
1. Computación cuántica	11
1.1. Fundamentos de computación cuántica	11
1.1.1. Qubits y superposición	11
1.1.2. Puertas cuánticas	13
1.1.3. Circuitos cuánticos	16
1.1.4. Medidas	17
1.2. Ejemplos de fenómenos cuánticos	17
1.2.1. Paralelismo cuántico	17
1.2.2. Entrelazamiento	18
1.2.3. Teleportación cuántica	19
1.3. Aplicaciones de la computación cuántica	20
2. Aprendizaje automático	23
2.1. Fundamentos de machine learning clásico	23
2.1.1. Los datos	23
2.1.2. El modelo	25
2.1.3. El aprendizaje	26
2.2. Ejemplos de modelos de machine learning	27
2.2.1. Regresión lineal	27
2.2.2. Support vector machines (SVM)	28
3. El problema de optimización	31
3.1. Descripción del problema	31
3.2. Máquina analógica clásica	32
3.3. Máquinas cuánticas	35
3.3.1. Marco de trabajo	35
3.3.2. El algoritmo	37
3.3.3. Algunas observaciones	39
Conclusiones	41
Bibliografía	43

Índice de figuras

1.1.	Representación de un qubit mediante la esfera de Bloch [2].	12
1.2.	Representación de la puerta CNOT para dos qubits [2].	15
1.3.	Representación de la puerta cuántica <i>controlled-U</i> para n qubits [2].	16
1.4.	Circuito cuántico que intercambia dos qubits (izquierda) y su notación equivalente habitual [2].	17
1.5.	Representación de la operación de medida en un circuito cuántico [2].	17
1.6.	Circuito cuántico para visualizar el fenómeno del paralelismo cuántico [2].	18
1.7.	Circuito cuántico generador de los estados de Bell [2].	18
1.8.	Circuito cuántico para la teleportación de un qubit. Las dos líneas superiores se corresponden con el sistema de Alice y la inferior, al de Bob [2].	19
2.1.	Representación de los datos de entrenamiento (x_n, y_n) , mostrando el salario en función de la edad para los datos conocidos de la Tabla 2.2. En este ejemplo, el objetivo sería predecir el sueldo de una persona de 60 años ($x = 60$), marcado mediante la línea discontinua roja [3].	25
3.1.	Representación del circuito eléctrico que se utiliza para la optimización analógica. Éste se compone de m dipolos eléctricos \mathcal{N}_k interconectados a través de la red de interacción \mathcal{N} [4].	33
3.2.	Esquema del i -ésimo dipolo conectado a la red central de interacción \mathcal{N} .	33
3.3.	Circuito multiplicador matricial de un vector de m componentes a otro de n componentes, sintetizado mediante una matriz de conductancias $m \times n$.	34
3.4.	Circuito cuántico para resolver el problema de optimización de (3.1) cuando el operador $\hat{\Lambda}$ es unitario [5].	37
3.5.	Fragmento del circuito cuántico para resolver el problema de optimización de (3.1) cuando el operador $\hat{\Lambda}$ es unitario. Esta parte se corresponde con la fase de codificación para preparar el estado $ \psi_1\rangle$ [5].	38

Índice de tablas

1.1. Tablas lógicas de la puerta XOR (izquierda) y la puerta NAND (derecha)	15
2.1. Ejemplo ficticio de datos que no se encuentran en formato numérico [3]	24
2.2. Ejemplo ficticio de datos (véase Tabla 2.1) convertidos en formato numérico [3]	24
3.1. Resumen de la notación en los problemas de optimización de la regresión lineal y de SVM según la ecuación general (3.1)	31
3.2. Resumen de la identificación entre los elementos de las ecuaciones de evolución de los modelos de regresión y SVM (ML) con los de redes eléctricas (RE), (3.7) y (3.2), respectivamente.	34
3.3. Resumen de la identificación entre los elementos de las ecuaciones de evolución de los modelos de regresión y SVM (ML) con los de redes eléctricas (RE), (3.7) y (3.2), respectivamente, cuando todos los condensadores son iguales.	35

Introducción

En este trabajo, se aborda el estudio de uno de los campos de la física más novedosos y que, desde las últimas décadas, se encuentra en el foco de investigación de la comunidad científica: la *computación cuántica*.

La idea de aprovechar la *mecánica cuántica* al servicio de la computación surge de la necesidad de simular sistemas cuánticos complejos. A medida que el tamaño de estos sistemas aumenta, la simulación en computadoras clásicas se vuelve inabordable, según señaló Feynman en 1982 [1]. En 1985, Deutsch mostró que la computación cuántica, al basarse en partículas cuánticas, presenta un paradigma radicalmente diferente: las computadoras cuánticas no sólo son capaces de simular sistemas cuánticos, sino que también poseen la capacidad intrínseca de realizar ciertas tareas probabilísticas más rápido que cualquier medio clásico [2].

En este contexto, nuestro enfoque se centra en la aplicación concreta de la computación cuántica en el ámbito del *aprendizaje automático* o *machine learning*. Este campo se encuentra actualmente en pleno auge, impulsado por los notables avances que la inteligencia artificial ha experimentado durante los últimos años. A grandes rasgos, su propósito fundamental es la construcción de máquinas capaces de llevar a cabo la extracción de patrones y la toma de decisiones basándose en el análisis de datos [3].

Más allá de los conceptos abstractos, en este trabajo proponemos utilizar la física para llevar a cabo el *entrenamiento* de dos modelos concretos de machine learning. Recientemente, se ha publicado un resultado que formaliza la equivalencia entre redes neuronales y circuitos eléctricos, partiendo del paradigma de red tipo Hopfield [4]. Este resultado establece un diccionario entre ambos mundos y permite construir una *máquina analógica clásica* basada en la teoría de circuitos eléctricos para entrenar los modelos de manera más eficiente. Además, veremos que es posible construir un *algoritmo cuántico* que resuelva el mismo problema [5] desde un enfoque que no sólo representa una aplicación práctica de los principios cuánticos, sino que también sugiere la posibilidad de mejoras significativas en la eficiencia y el rendimiento del aprendizaje automático.

El primer capítulo ofrece una visión general de la computación cuántica desde su nivel más elemental y se divide en tres secciones. La primera aborda los fundamentos de este campo, incluyendo los conceptos clave para su comprensión, como qubit, superposición, puertas cuánticas...

La segunda sección ofrece algunos ejemplos de fenómenos cuánticos relevantes para la computación cuántica a la hora de la elaboración de algoritmos. En particular, se analiza el paralelismo cuántico, el entrelazamiento y la teleportación cuántica.

La última sección del capítulo se centra en algunas aplicaciones conocidas de la computación cuántica en la actualidad, teniendo en cuenta que se trata de un campo muy reciente y cuyo verdadero potencial sigue investigándose a día de hoy.

El segundo capítulo ofrece una introducción al machine learning con el fin de contextualizar al lector y que éste tenga a su disposición la información necesaria para comprender el problema a resolver en el siguiente capítulo. En la primera sección, vemos los fundamentos del machine learning clásico desde un formalismo matemático, analizando los conceptos de vectores, modelos y aprendizaje.

La segunda sección muestra dos ejemplos concretos de modelos de machine learning: regresión lineal y *support vector machines*. Ambos modelos tienen mucha relevancia dentro del mundo del aprendizaje automático y su entrenamiento será el objeto del siguiente capítulo.

El tercer capítulo aborda el problema de optimización que surge al entrenar los modelos de machine learning explicados anteriormente, dividiéndose en tres secciones. La primera ofrece una descripción detallada del problema de optimización analógica que constituye la fase de entrenamiento de los modelos, junto con la notación que se va a utilizar.

En la segunda sección, se describe una solución clásica a partir de la construcción de una máquina analógica basada en circuitos eléctricos, de forma que resuelve el problema en su evolución temporal.

La tercera y última sección del capítulo muestra una máquina alternativa para resolver el problema de optimización utilizando la computación cuántica como herramienta. Para ello, se debe establecer previamente un marco de trabajo que permita definir bien el algoritmo cuántico utilizando los conceptos estudiados en el primer capítulo.

Para finalizar, se ofrecen algunas de las conclusiones a las que se ha llegado a través de la realización de este trabajo y las posibles futuras líneas de investigación que surgen a partir de las mismas.

Capítulo 1

Computación cuántica

La computación clásica utiliza la lógica binaria para el tratamiento de datos y almacena la información en *bits* (acrónimo de *binary digit*, dígito binario). Éstos tienen una conexión muy importante con la física pues, en la mayoría de las tecnologías de la información, los bits se representan mediante la presencia o ausencia de carga eléctrica. En particular, se utilizan los electrones de determinadas regiones de una clase concreta de materiales: los semiconductores (por ejemplo, el silicio). Éstos desempeñan un papel clave en este contexto ya que, a través de transistores y sus bandas de energía, ofrecen la plataforma física donde los ceros y unos de los bits encuentran su existencia tangible, lo que revela la fuerte conexión entre física y computación desde su nivel más elemental.

En general, los ordenadores que se construyen hoy día son de este tipo y son capaces de abordar una gran cantidad de problemas. Sin embargo, la mecánica cuántica ha abierto un nuevo camino en la computación: la *computación cuántica*. Este nuevo campo se sirve de algunos de los principios fundamentales de la física cuántica para la resolución de problemas mediante el uso de *qubits* (acrónimo de *quantum bit*, bit cuántico) como unidad fundamental. Este capítulo servirá como introducción a los conceptos fundamentales de la computación cuántica, sabiendo que es un campo que actualmente sigue en desarrollo y cuyo alcance tiene una complejidad mucho mayor que la que se puede resumir en este trabajo.

El esquema que se ha seguido a lo largo de este capítulo ha sido, fundamentalmente, el proporcionado por [2], pues para muchos es considerada la *Biblia* en este ámbito y se ha completado con algunas de las ideas que aparecen en [6], [7], [8] y [9].

1.1. Fundamentos de computación cuántica

1.1.1. Qubits y superposición

Comenzaremos dando la definición de *qubit*, visto como un objeto matemático abstracto. Al igual que en el caso clásico, donde un bit define un estado que puede ser 0 o 1, en la computación cuántica se utilizan los qubits. Siguiendo la notación de Dirac, cuyas propiedades damos por conocidas (véase [6], Secciones 1 y 2), éstos pueden definir un estado $|0\rangle$ o $|1\rangle$, que podríamos considerar como el análogo al estado 0 o 1, respectivamente, de un bit clásico. La diferencia fundamental entre bits y qubits radica en el concepto de *superposición*. Es decir, un qubit $|\psi\rangle$ puede definir un estado resultante de la combinación lineal de los estados $|0\rangle$ y $|1\rangle$, siendo de la forma

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1.1)$$

con $\alpha, \beta \in \mathbb{C}$ y bajo la condición de normalización $|\alpha|^2 + |\beta|^2 = 1$. Alternativamente, podemos considerar que cada qubit es un vector en \mathbb{C}^2 y $\{|0\rangle, |1\rangle\}$ es una base ortonormal del espacio de estados computacionales. En adelante, representaremos vectorialmente dicha base de la forma

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1.2)$$

De este modo, ya hemos visto la primera diferencia entre computación clásica y cuántica: en el caso clásico, cada unidad de información contiene un único estado de los dos posibles (0 o 1), mientras que en cuántica, cada qubit puede representar uno de los infinitos estados intermedios entre $|0\rangle$ y $|1\rangle$, generados a partir de combinaciones lineales de los mismos. Sin embargo, la mecánica cuántica restringe la cantidad de información disponible del estado cuántico $|\psi\rangle$. Cuando medimos un qubit descrito por (1.1), el estado *colapsa* y obtenemos como resultado $|0\rangle$ o $|1\rangle$, con probabilidades $|\alpha|^2$ o $|\beta|^2$, respectivamente (por lo que se debe verificar la condición de normalización $|\alpha|^2 + |\beta|^2 = 1$). Es decir, la medición del estado de un qubit provoca un cambio en el mismo, de forma que pasa de un estado de superposición a convertirse en el resultado medido, $|0\rangle$ o $|1\rangle$. Se trata de uno de los postulados fundamentales de la mecánica cuántica.

Tenemos entonces que un qubit puede existir en un espacio continuo de estados intermedios entre $|0\rangle$ y $|1\rangle$ hasta que es medido. En la literatura, es habitual encontrar la esfera de Bloch como representación geométrica del espacio que conforman los estados que pueden aparecer en un qubit (véase la Figura 1.1), de modo que cualquiera de ellos puede escribirse como

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

con $\theta, \varphi \in \mathbb{R}$. Como apunte, notemos que la base $\{|0\rangle, |1\rangle\}$ no es la única posible. Otra base ortonormal que conviene conocer es $\{|+\rangle, |-\rangle\}$, donde $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ y $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

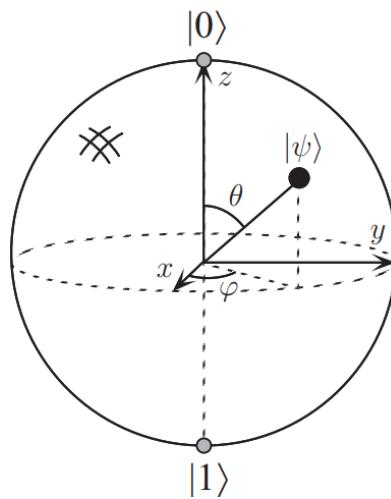


Figura 1.1: Representación de un qubit mediante la esfera de Bloch [2].

Supongamos ahora que tenemos un sistema formado por dos qubits. En el caso clásico, con dos bits tendríamos cuatro estados posibles: 00, 01, 10 y 11. Análogamente, en mecánica cuántica tenemos una base computacional que viene dada por $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$,

donde

$$|xy\rangle = |x\rangle \otimes |y\rangle, \quad x, y = 0, 1,$$

y \otimes denota el producto tensorial. Igual que en el caso anterior, un par de qubits puede existir en estado de superposición a partir de los cuatro estados anteriores de forma que el vector que describe el sistema es

$$|\varphi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle,$$

con $\alpha_{ij} \in \mathbb{C}$, $i, j = 0, 1$, y bajo la condición de normalización $\sum_{i,j} |\alpha_{ij}|^2 = 1$. Del mismo modo, la medición del estado del sistema dará como resultado uno de los estados de la base ($|x\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$) con probabilidad dada por la amplitud correspondiente $|\alpha_x|^2$ y el estado del sistema justamente después de la medición es $|x\rangle$.

Ahora, supongamos que no medimos los dos qubits simultáneamente sino que lo hacemos primero sobre el de la izquierda. Tendremos entonces que dicho qubit puede valer $|0\rangle$ con probabilidad $|\alpha_{00}|^2 + |\alpha_{01}|^2$ y el estado justamente tras la medición sería

$$|\psi'\rangle = \frac{\alpha_{00} |00\rangle + \alpha_{01} |01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}},$$

donde se ha tenido en cuenta de nuevo que los vectores deben estar normalizados. El caso en el que el primer qubit da $|1\rangle$ se deduce de forma análoga.

Un ejemplo importante de sistemas formados por dos qubits son los *estados de Bell* o *pares EPR*, que toman la forma

$$\begin{aligned} |\beta_{00}\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, & |\beta_{01}\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\ |\beta_{10}\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}, & |\beta_{11}\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \end{aligned}$$

Éstos tienen la propiedad de *entrelazamiento*, concepto que aparecerá posteriormente en este trabajo y al que, de momento, no haremos más referencia.

Para terminar, consideremos un sistema formado por n qubits. De forma análoga al caso anterior, la base computacional viene dada por estados de la forma

$$|x_1 x_2 \dots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle,$$

con $x_i = 0, 1$, $i = 1, \dots, n$, y \otimes el producto tensorial. Es decir, cada estado cuántico del sistema está definido por 2^n coeficientes.

1.1.2. Puertas cuánticas

Los ordenadores clásicos están formados a partir de circuitos eléctricos constituidos por *cables* y *puertas lógicas*, de modo que los primeros se encargan del transporte de la información y las segundas la manipulan. En cuántica, también encontramos estos conceptos y, en este apartado, nos centraremos en el análisis de las puertas cuánticas. Suponiendo que estamos trabajando con la base computacional de un único qubit dada por (1.2), las puertas cuánticas pueden representarse en forma matricial. Por ejemplo, una puerta NOT para un bit, cuya definición clásica viene dada por las relaciones $0 \rightarrow 1$ y $1 \rightarrow 0$, toma su forma cuántica para un qubit mediante la matriz

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{1.3}$$

Es decir, dado un estado $|\psi\rangle$ definido como (1.1), teniendo en cuenta la notación vectorial del mismo en la base (1.2) y haciendo actuar la matriz X sobre éste, obtenemos como resultado el estado

$$X|\psi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle.$$

Así pues, las puertas cuánticas para un único qubit vienen dadas por matrices 2×2 , lo cual implica un comportamiento lineal de las mismas. Además, con el fin de preservar la condición de normalización sobre los estados finales, toda matriz U que represente una puerta cuántica para un qubit debe ser unitaria, es decir, debe cumplir $U^\dagger U = I$, donde U^\dagger representa el adjunto de U e I es la matriz identidad. Esta será la única restricción que encontraremos sobre las puertas cuánticas.

Otros ejemplos importantes de este tipo de objetos son la puerta Z

$$Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (1.4)$$

y la puerta de Hadamard, H , que viene dada por

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (1.5)$$

cuyas acciones sobre un estado cuántico $|\psi\rangle$ dado por (1.1) son

$$\begin{aligned} Z|\psi\rangle &= Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle, \\ H|\psi\rangle &= H(\alpha|0\rangle + \beta|1\rangle) = \alpha \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned}$$

Observemos que, además, se verifica $H^2 = I$.

Se puede demostrar (véase [2]), que toda puerta cuántica U para un único qubit se corresponde con un número finito de rotaciones y reflexiones sobre la esfera de Bloch. Es decir, la matriz que representa a U puede descomponerse como un producto de rotaciones de la forma

$$\begin{pmatrix} \cos \frac{\gamma}{2} & -\sin \frac{\gamma}{2} \\ \sin \frac{\gamma}{2} & \cos \frac{\gamma}{2} \end{pmatrix},$$

y de la puerta que representa la rotación de ángulo β sobre el eje \hat{z} ,

$$\begin{pmatrix} e^{-i\frac{\beta}{2}} & 0 \\ 0 & e^{i\frac{\beta}{2}} \end{pmatrix},$$

multiplicadas globalmente por la fase $e^{i\alpha}$, con $\alpha, \beta, \gamma \in \mathbb{R}$. No obstante, no será necesario construir estas puertas para valores α, β y γ arbitrarios, sino que sólo se construirán aproximaciones para ciertos valores concretos de ellos. De este modo, se puede obtener cualquier puerta cuántica para un único qubit mediante un conjunto finito de puertas cuánticas. El análogo de esta afirmación para puertas cuánticas de múltiples qubits también es cierto y es fundamental para la computación cuántica.

Por otro lado, es conocido que existen puertas clásicas, como las de tipo NAND (véase la Tabla 1.1), que se denominan *universales* pues toda función sobre bits puede obtenerse como composición de puertas de este tipo (el análisis de los objetos clásicos se sale de los objetivos del trabajo pero se puede encontrar más información sobre las puertas lógicas clásicas en [7]).

A	B	$A \oplus B$	A	B	$A \bar{A} B$
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

Tabla 1.1: Tablas lógicas de la puerta XOR (izquierda) y la puerta NAND (derecha).

En el caso cuántico, el ejemplo más importante y similar a la universalidad para varios qubits es la puerta CNOT (*controlled-NOT*). Ésta tiene dos qubits como input, el qubit de control y el qubit objetivo. Podemos ver su representación como circuito en la Figura 1.2, donde la línea de arriba representa al qubit de control y la de abajo, al qubit objetivo. Notemos que, en este caso, el símbolo \oplus es el de suma en módulo 2. Su representación matricial en la base $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, que vectorialmente toma la forma

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (1.6)$$

es la que se muestra a continuación

$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (1.7)$$

donde U_{CN} es unitaria, luego se verifica la única condición sobre este tipo de operadores. Así, la acción de este objeto puede resumirse como

$$U_{CN} |00\rangle = |00\rangle, \quad U_{CN} |01\rangle = |01\rangle, \quad U_{CN} |10\rangle = |11\rangle, \quad U_{CN} |11\rangle = |10\rangle. \quad (1.8)$$

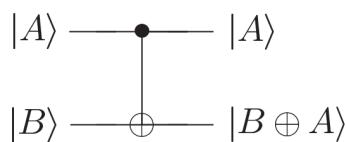


Figura 1.2: Representación de la puerta CNOT para dos qubits [2].

Como se ha mencionado, se considera que este tipo de puertas son el caso análogo a las puertas universales clásicas gracias a un resultado que no vamos a demostrar (véase [2], Sección 4.5): toda puerta cuántica para varios qubits puede descomponerse en puertas CNOT y de un único qubit.

Además, comparando la tabla lógica de la puerta clásica XOR (véase la Tabla 1.1) con (1.8), podría considerarse que la puerta CNOT es una generalización cuántica de la puerta XOR, pese a que esta última no es universal. No es posible encontrar este análogo para todos los ejemplos clásicos pues existen casos en los que el efecto de una puerta clásica es irreversible y conlleva una pérdida de información. Mientras tanto, las puertas cuánticas son siempre invertibles (recordemos que vienen dadas por matrices unitarias), lo cual pone

de manifiesto una vez más el potencial de la mecánica cuántica dentro de la rama de la computación.

Por último, supongamos que U es una matriz unitaria arbitraria que actúa sobre n qubits, de forma que U puede interpretarse como una puerta cuántica sobre éstos. Es posible definir la puerta *controlled-U* como una generalización de la puerta CNOT que ya conocemos. Tal y como puede observarse en la Figura 1.3, dicha puerta tiene un sólo qubit de control (la línea de arriba con el punto negro) y n qubits objetivo (líneas que atraviesan la caja U). En particular, si $U = X$ y $n = 1$, se obtiene la puerta CNOT, ya analizada anteriormente.

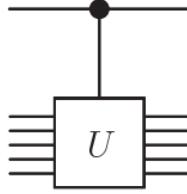


Figura 1.3: Representación de la puerta cuántica *controlled-U* para n qubits [2].

1.1.3. Circuitos cuánticos

Aunque ya hemos dado los primeros pasos dentro de los circuitos cuánticos (por ejemplo, el de la Figura 1.2), vamos a ver con más detalle cómo interpretarlos y su utilidad. En primer lugar, los circuitos deben leerse de izquierda a derecha. Cada línea del circuito representa un *cable*, pero no en el sentido físico de la palabra, sino que simboliza el paso del tiempo o un cambio de posición en el espacio. Además, por convenio, se asume que el estado inicial del circuito en la base de estados computacionales es aquel que está formado únicamente por qubits en el estado $|0\rangle$.

Otro requisito de los circuitos cuánticos es que no pueden ser cíclicos, restricción que no encontramos en el caso clásico. Tampoco está permitida la operación FANIN, que consiste en la unión de dos cables de entrada y, por tanto, no es reversible. Lo mismo ocurre con la operación análoga para la salida, FANOUT, pues la mecánica cuántica no permite realizar una copia idéntica de un qubit (la demostración de esta afirmación puede encontrarse en [2], Sección 1.3.5.).

En la Figura 1.4, podemos observar un ejemplo sencillo de un circuito cuántico con dos qubits. En particular, éste utiliza tres puertas CNOT sucesivas, dando lugar al intercambio de los estados de los dos qubits. Paso a paso, el efecto del circuito sobre un estado $|a, b\rangle$ de la base de estados computacionales (1.6) es

$$\begin{aligned} U_{CN}\hat{U}_{CN}U_{CN}|a, b\rangle &= U_{CN}\hat{U}_{CN}|a, a \oplus b\rangle = U_{CN}|a \oplus (a \oplus b), a \oplus b\rangle \\ &= |a \oplus (a \oplus b), (a \oplus (a \oplus b)) \oplus (a \oplus b)\rangle = |b, a\rangle, \end{aligned}$$

donde U_{CN} está definida en (1.7), \hat{U}_{CN} es la expresión matricial de una puerta CNOT como la de la Figura 1.2 en la que se han invertido los papeles de qubit de control y qubit objetivo, y recordemos que \oplus denota la suma en módulo 2.

Llegados a este punto, cabe destacar que, dentro del contexto de la computación, es posible simular todo circuito clásico mediante un circuito cuántico. Esta afirmación puede

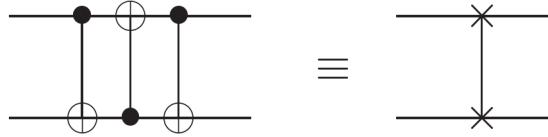


Figura 1.4: Circuito cuántico que intercambia dos qubits (izquierda) y su notación equivalente habitual [2].

resultar sorprendente al lector puesto que, como ya hemos mencionado anteriormente, los elementos cuánticos son siempre reversibles mientras que en el caso clásico puede no ser así. La clave de esta cuestión es la existencia de una puerta lógica clásica denominada *puerta de Toffoli* que, para cualquier circuito clásico, permite obtener un circuito equivalente tal que sólo se componga de elementos reversibles y su versión cuántica es la que resuelve el problema. Es decir, un ordenador cuántico puede simular la acción de uno clásico. Para más detalles sobre esta cuestión se recomienda consultar [2], Sección 1.4.1.

1.1.4. Medidas

La última operación que hemos considerado relevante para este apartado introductorio es la de medida, que se representa como en la Figura 1.5. Su acción consiste en convertir un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ en un bit clásico $M = 0$ ó 1 con probabilidades $|\alpha|^2$ y $|\beta|^2$, respectivamente. Gráficamente, se distingue de un qubit por tener como cable una línea doble.

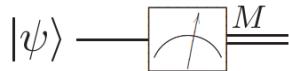


Figura 1.5: Representación de la operación de medida en un circuito cuántico [2].

1.2. Ejemplos de fenómenos cuánticos

En esta sección, veremos algunos de los fenómenos más relevantes en computación cuántica. Éstos son fundamentales para los *algoritmos cuánticos*, sobre los que no haremos referencia de momento pues en los próximos capítulos analizaremos uno de ellos en concreto, pero al lector interesado se le recomienda leer [2], [6] y [9], donde aparecen desarrollados algunos algoritmos notables como el de Deutsch, el de Deutsch-Josza o el de Grover.

1.2.1. Paralelismo cuántico

Un aspecto fundamental en la elaboración de algoritmos cuánticos es el llamado *paralelismo cuántico*. Esta propiedad permite que los ordenadores cuánticos evalúen una misma función $f(x)$ en distintos puntos x de forma simultánea, sin necesidad de construir varios circuitos. Veámoslo con un ejemplo: supongamos que tenemos el circuito que se muestra en la Figura 1.6, con $f : \{0, 1\} \rightarrow \{0, 1\}$ (un sólo bit). El estado resultante será

$$|\psi\rangle = \frac{|0, 0 + f(0)\rangle + |1, 0 + f(1)\rangle}{\sqrt{2}} = \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}.$$

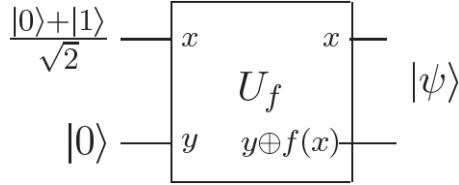


Figura 1.6: Circuito cuántico para visualizar el fenómeno del paralelismo cuántico [2].

Podemos observar que, gracias a la superposición, se está evaluando f para dos valores de x distintos, al mismo tiempo y mediante un único circuito. Para la generalización al caso de funciones con un número arbitrario de bits, se utiliza la transformada de Hadamard y la de Walsh-Hadamard. No entraremos en detalles (véase [2]), salvo decir que la primera da lugar a la superposición de 2^n estados mediante el uso de n puertas.

Ahora, notemos que al medir el estado final, éste colapsa y, por tanto, se necesita dar un paso más allá para extraer toda la información que nos ofrece el paralelismo cuántico. Aquí entran en juego los algoritmos cuánticos.

1.2.2. Entrelazamiento

Uno de los fenómenos cuánticos más relevantes para la computación es el *entrelazamiento*. Desde un punto de vista matemático, el espacio de estados formado por N qubits se genera mediante el producto tensorial de los N espacios de estados de cada qubit por separado. Se dice que un sistema formado por N qubits y definido por el estado $|\psi\rangle$ es *separable* cuando éste puede escribirse como producto tensorial de estados para un único qubit de la forma

$$|\psi\rangle = |\psi_1 \cdots \psi_N\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_N\rangle,$$

y se dice que los estados están *entrelazados* cuando no es posible llevar a cabo esta factorización. Se trata de una propiedad muy importante a la hora de utilizar algoritmos cuánticos y puede profundizarse en el tema en [9], Sección 3.3.

Un ejemplo habitual en la literatura para ilustrar el concepto de estados entrelazados es el de los ya mencionados *estados de Bell* o *pares EPR*. Consideremos el circuito de dos qubits de la Figura 1.7, formado por una puerta de Hadamard y una puerta CNOT.

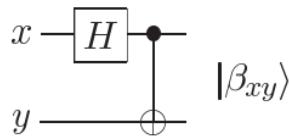


Figura 1.7: Circuito cuántico generador de los estados de Bell [2].

Teniendo en cuenta (1.5) y (1.8), se puede comprobar que el estado resultante $|\beta_{xy}\rangle$ para cada input $|xy\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ es el siguiente

$$\begin{aligned} |\beta_{00}\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, & |\beta_{01}\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\ |\beta_{10}\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}, & |\beta_{11}\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \end{aligned}$$

que son los conocidos estados de Bell o pares EPR. Como podemos observar estos estados no son factorizables, de modo que están entrelazados. El valor de esta propiedad proviene de que las mediciones realizadas en una de las partículas sobre cualquiera de estos estados, instantáneamente, afectarán el estado de la otra partícula, sin importar la distancia que las separe, desafiando la intuición clásica. Por ejemplo, si medimos la primera partícula de $|\beta_{11}\rangle$ y encontramos que está en el estado $|0\rangle$, entonces la otra partícula se encuentra, instantáneamente, en el estado $|1\rangle$, a pesar de que ambas posibilidades eran igualmente probables antes de la medición. Esta correlación intrínseca entre qubits aparentemente independientes permite ciertas ventajas a la hora de realizar cálculos simultáneos.

1.2.3. Teleportación cuántica

La teleportación cuántica es un fenómeno que permite el traslado de estados cuánticos sin necesidad de un canal de comunicación cuántico, aunque sí se requiere un canal clásico. Habitualmente, esta técnica se ilustra mediante el siguiente ejemplo (para verlo con más detalle, se recomienda leer [2], Sección 1.3.7 y [6], Sección 6.4).

Supongamos que Alice y Bob son dos conocidos, cada uno de ellos con uno de los qubits que componen un estado de Bell, pero ahora se encuentran separados, no pueden verse. El objetivo de Alice es conseguir enviar un qubit $|\psi\rangle$ a Bob, con la restricción de que ni siquiera ella sabe cuál es la forma del estado del qubit en cuestión y, además, sólo puede enviar información clásica a Bob.

Aunque parezca complicado, la teleportación cuántica permite que Alice logre su objetivo. El proceso es el que se detalla a continuación: Alice hace interactuar al qubit $|\psi\rangle$ con el que tiene procedente del estado de Bell y mide ambos, obteniendo uno de los cuatro resultados clásicos posibles: 00, 01, 10 o 11. Ahora, ella envía esta información a Bob y, según el mensaje que éste reciba, Bob llevará a cabo una operación de cuatro posibles sobre su qubit correspondiente del estado de Bell. Concretamente, la operación es $Z^{M_1}X^{M_2}$, donde X y Z son las puertas definidas en (1.3) y (1.4), respectivamente, y M_1 y M_2 son las medidas obtenidas sobre el qubit $|\psi\rangle$ y el de Alice, respectivamente. De este modo, es posible que Bob recupere el estado original $|\psi\rangle$. El circuito correspondiente sería el que se muestra en la Figura 1.8.

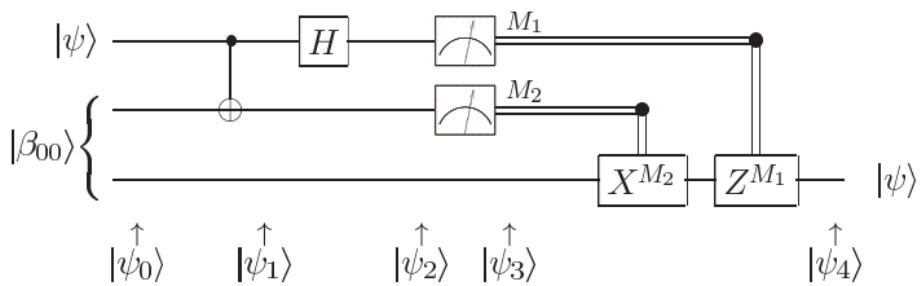


Figura 1.8: Circuito cuántico para la teleportación de un qubit. Las dos líneas superiores se corresponden con el sistema de Alice y la inferior, al de Bob [2].

Sin entrar en más detalle sobre este fenómeno, conviene recalcar que la teleportación cuántica no puede producirse a mayor velocidad que la de la luz (lo que implicaría la posibilidad de enviar información hacia el pasado), pues para concluir la teleportación, Alice debe enviar a Bob el resultado que ha obtenido tras su medida mediante un canal

de comunicación clásico, el cual se encuentra limitado por la velocidad de la luz. Otro aspecto importante es que, aunque parezca que se está creando una copia del estado $|\psi\rangle$ (ya mencionamos anteriormente que esto no estaba permitido), en realidad, el único qubit que queda en el estado $|\psi\rangle$ tras la teleportación es el de Bob, mientras que el original queda como $|0\rangle$ o $|1\rangle$.

La teleportación cuántica pone de manifiesto la equivalencia de diferentes recursos cuánticos mostrando que, en algunos casos, una combinación específica de entrelazamiento cuántico y comunicación clásica puede ser tan efectiva como el envío directo de información cuántica. En particular, el uso de una pareja EPR junto con dos bits clásicos de comunicación es al menos tan efectivo como utilizar un qubit de comunicación. Este tipo de observaciones tiene implicaciones importantes en el diseño de algoritmos cuánticos y en la comprensión de cómo se pueden utilizar los recursos cuánticos de manera eficiente.

1.3. Aplicaciones de la computación cuántica

Existen ejemplos de algoritmos cuánticos como el de Deutsch–Jozsa que ponen en evidencia la mayor eficiencia de la computación cuántica para resolver ciertos problemas computacionales frente a los límites que encontramos en los ordenadores clásicos, lo que podría llevar a pensar en una *supremacía cuántica*. Sin embargo, es común que los problemas que se pueden resolver mediante estos algoritmos no sean de verdadero interés. Aquí mostraremos tres tipos de algoritmos cuánticos que suponen una ventaja frente a las alternativas clásicas y que son realmente útiles en sus aplicaciones:

- La primera clase de algoritmos que vamos a considerar es la de aquellos basados en versiones cuánticas de la *transformada de Fourier*. Como sabemos, la transformada de Fourier clásica tiene un gran número de aplicaciones en las distintas ramas de la ciencia, dando una versión de los problemas que a menudo es más sencilla de resolver que la original. Del mismo modo, es posible utilizar la transformada de Fourier cuántica para resolver de manera eficiente problemas que no pueden ser atacados eficazmente por una computadora clásica. Un ejemplo sería el algoritmo de Shor para la resolución de problemas de factorización.
- En segundo lugar, consideramos los algoritmos cuánticos de búsqueda o *quantum search algorithms*. Éstos resuelven el siguiente problema: dado un espacio de búsqueda de tamaño N del que no sabemos nada sobre la estructura de la información que hay en él, queremos encontrar en dicho espacio un elemento que satisface una propiedad conocida. Mientras que una computadora clásica requiere N operaciones aproximadamente para la resolución de este problema, un algoritmo cuántico de búsqueda puede hacerlo con unas \sqrt{N} operaciones, lo cuál tiene gran interés de aplicación en una importante variedad de problemas.
- Otra rama de la computación cuántica de interés en las aplicaciones es la *simulación de sistemas cuánticos* que no pueden ser simulados de forma eficiente por ordenadores clásicos. Este es un problema muy importante en varios campos, como la química cuántica, donde las restricciones de los ordenadores clásicos dificultan la simulación adecuada del comportamiento de moléculas con un tamaño relativamente moderado y la complicación se amplifica notablemente a la hora de tratar con moléculas grandes como las que aparecen en multitud de procesos biológicos.

Es importante entender que la computación cuántica es un campo que aún se encuentra en desarrollo y queda mucho por investigar, por lo que no podemos dar una respuesta clara

sobre el verdadero potencial que tiene. Actualmente, existen varias líneas de investigación orientadas a la aplicación de la computación cuántica sobre el aprendizaje automático. Esta nueva rama de la ciencia recibe el nombre de *quantum machine learning*.

En lo que resta de trabajo, el objetivo es mostrar la utilidad de la computación cuántica (y, en general, la física en su totalidad) como herramienta a la hora de resolver problemas matemáticos en los que, a priori, no vemos que pueda existir una interpretación física de los mismos. Siendo más concretos, en los siguientes capítulos, vamos a centrarnos en resolver un problema de optimización común dentro del contexto del aprendizaje automático clásico y veremos que es posible abordarlo mediante algoritmos cuánticos.

Capítulo 2

Aprendizaje automático

En este capítulo, vamos a definir algunos de los conceptos clave en torno a otro campo que también se considera de los más relevantes del panorama científico actual: el *aprendizaje automático* o *machine learning*. El objetivo no es profundizar en esta rama de la ciencia, sino comprender las nociones básicas de la misma para contextualizar el problema que se va a resolver en el siguiente capítulo utilizando técnicas basadas en principios físicos.

La estructura del capítulo se basa en la que encontramos en [3], pues ofrece las nociones básicas sobre el aprendizaje automático formalizadas matemáticamente. Para profundizar más en el tema, también se recomienda leer [10] y [11].

2.1. Fundamentos de machine learning clásico

El aprendizaje automático es una de las ramas de la inteligencia artificial que trata el desarrollo de técnicas y algoritmos para que un ordenador sea capaz de aprender patrones y tomar decisiones, sin estar programado explícitamente para ello. Así pues, éstos no deben seguir una serie de instrucciones concretas, sino que los sistemas resultantes deben utilizar modelos matemáticos parametrizados que se ajusten a un conjunto de datos conocido, de forma que puedan predecir resultados sobre datos futuros. Decimos que se trata de un aprendizaje, entendiéndolo como una forma de encontrar patrones y estructuras en los datos automáticamente, mediante la optimización de parámetros. La cantidad de aplicaciones que se pueden encontrar para el machine learning es innumerable, desde predicciones financieras hasta diagnósticos médicos para la predicción de tumores pasando por las recomendaciones de Netflix para la próxima película que podría gustarte.

Dentro del contexto del aprendizaje automático, existen tres aspectos fundamentales: los datos, el modelo y el aprendizaje.

2.1.1. Los datos

En primer lugar, asumiremos que los datos con los que trabajamos ya se encuentran representados numéricamente y que son legibles para un ordenador clásico. Por ejemplo, en el caso de un conjunto de datos (o *dataset*) como el de la Tabla 2.1, cada fila representa un caso particular y cada columna es una característica concreta.

La elección de las características que van a determinar el modelo y la representación numérica de éstas son de vital importancia, por lo que el conocimiento del área para el

Nombre	Género	Nivel universitario	Código postal	Edad	Salario anual (€)
Aditya	M	MSc	W21BG	36	89563
Bob	M	PhD	EC1A1BA	47	123543
Chloé	F	BEcon	SW1A1BH	26	23989
Daisuke	M	BSc	SE207AT	68	138769
Elisabeth	F	MBA	SE10AA	33	113888

Tabla 2.1: Ejemplo ficticio de datos que no se encuentran en formato numérico [3].

que se está trabajando suele ser fundamental para lograr resultados fiables. En la Tabla 2.2, se pueden ver los datos de la Tabla 2.1 en un formato numérico.

Género	Nivel univ.	Latitud (grados)	Longitud (grados)	Edad	Salario anual (miles €)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

Tabla 2.2: Ejemplo ficticio de datos (véase Tabla 2.1) convertidos en formato numérico [3].

El siguiente paso es el de normalización y escalado de los datos (por columnas), de forma que tengan media 0 y varianza 1. En adelante, supondremos que los datos ya se encuentran en este formato y cada input, es decir, cada fila, la denotaremos como $\mathbf{x}_n \in X \subseteq \mathbb{R}^D$, $n = 1, \dots, N$, siendo N el número total de ejemplos o puntos de los que se dispone y D el número de características de las que se tiene información.

Siguiendo con el ejemplo de la Tabla 2.2, consideremos ahora el problema concreto de predecir el salario anual en función de la edad. En este caso, cada punto $x_n \in X \subseteq \mathbb{R}$ (la edad) lleva asociada una etiqueta $y_n \in Y \subseteq \mathbb{R}$ (el sueldo). Así pues, en este caso, el dataset viene dado por un conjunto de pares etiquetados $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset X \times Y$. Esta representación vectorial permite el uso del álgebra y la geometría lineales para facilitar la resolución de este tipo de problemas.

Llegados a este punto, podemos distinguir tres tipos de problemas de aprendizaje automático: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. El primero trabaja con datos etiquetados $(\mathbf{x}_n, \mathbf{y}_n) \in X \times Y \subseteq \mathbb{R}^D \times \mathbb{R}^K$ y su objetivo es encontrar un modelo tal que sea capaz de predecir un output \mathbf{y} para un nuevo input desconocido $\mathbf{x} \notin X$. El caso anterior de predecir el salario en función de la edad es un ejemplo de este tipo.

El aprendizaje no supervisado trata con conjuntos datos no etiquetados, es decir, el dataset se compone de vectores $\mathbf{x}_n \in X$ que no llevan asociada ninguna categoría como las que anteriormente hemos denotado por $\mathbf{y}_n \in Y$. Bajo estas circunstancias, la finalidad de los modelos es encontrar patrones o estructuras intrínsecas (por ejemplo, *clusters*) a partir de los datos conocidos. Un ejemplo de este tipo de problemas podría ser la agrupación automática de fotos similares en un teléfono móvil.

Por último, el aprendizaje por refuerzo es una variante del aprendizaje supervisado que, en lugar de una etiqueta, recibe un *feedback* o refuerzo. Este caso es habitual cuando el modelo creado debe tomar decisiones que pueden afectar al siguiente input \mathbf{y} , mediante un proceso de prueba-error, el modelo aprende qué acción debe realizar y cuándo para lograr ciertos objetivos. Por ejemplo, un ordenador programado para aprender a jugar al ajedrez mediante partidas reales.

2.1.2. El modelo

Una vez ya tenemos los datos en el formato vectorial adecuado, el siguiente paso consiste en la construcción del modelo o *predictor* que dé resultados para datos de entrada desconocidos. Es decir, que al recibir un input concreto (bajo la notación anterior, un nuevo vector de características $\mathbf{x} \notin X$, $\mathbf{x} \in \mathbb{R}^D$), éste sea capaz de producir un output $y \in Y \subset \mathbb{R}^K$ adecuado. En adelante, supondremos que los outputs son escalares, es decir, $K = 1$. Siguiendo con el ejemplo dado por las Tablas 2.1 y 2.2, véase la Figura 2.1, donde necesitamos un mecanismo para hallar el sueldo de una persona de 60 años sin tener ningún ejemplo conocido de la misma edad.

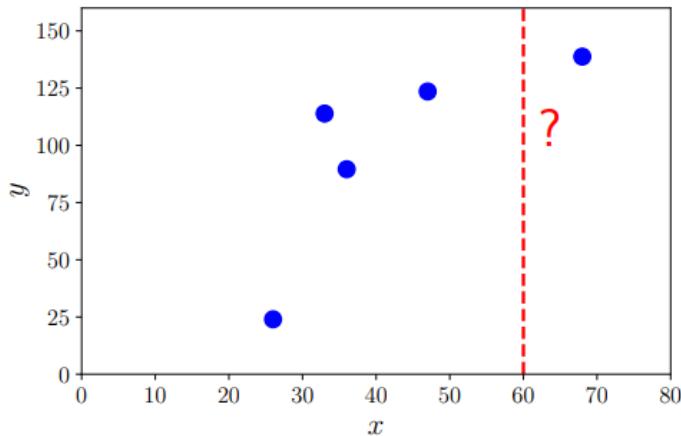


Figura 2.1: Representación de los datos de entrenamiento (x_n, y_n) , mostrando el salario en función de la edad para los datos conocidos de la Tabla 2.2. En este ejemplo, el objetivo sería predecir el sueldo de una persona de 60 años ($x = 60$), marcado mediante la línea discontinua roja [3].

Existen dos enfoques a la hora de trabajar con un predictor: determinista o probabilista.

- En el caso determinista, se utiliza un modelo paramétrico tal que se predice el output y a partir de una función $f : \mathbb{R}^D \times \mathbb{R}^L \rightarrow \mathbb{R}$ con $y = f(\mathbf{x}, \boldsymbol{\omega})$, siendo $\boldsymbol{\omega} \in \mathbb{R}^L$ el vector de parámetros. Dicha función puede aproximarse por su desarrollo en serie de Taylor en torno a $\boldsymbol{\omega} = \mathbf{0}$ y, truncando a su primer orden no trivial, da lugar a la regresión lineal múltiple

$$y = \boldsymbol{\omega} \cdot \phi(\mathbf{x}) + b, \quad (2.1)$$

donde $b \in \mathbb{R}$ es el denominado sesgo o *bias* y la función $\phi : X \rightarrow Z$, $Z \subset \mathbb{R}^L$ se denomina *mapa de características*. Notemos que el apellido *lineal* se refiere a la relación entre y y $\boldsymbol{\omega}$. En lo que sigue, tomaremos $b = 0$ sin pérdida de generalidad puesto que podemos extender el vector de características mapeadas de la forma $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \dots, \phi_L(\mathbf{x}))$ y el de parámetros como $\boldsymbol{\omega} = (\omega_0, \omega_1, \dots, \omega_L)$, con $\phi_0(\mathbf{x}) = 1$ y $\omega_0 = b$.

- Para un predictor probabilista, se tiene un vector aleatorio \mathbf{X} y una variable aleatoria Y , de modo que las características \mathbf{x} y los outputs y son realizaciones de los mismos, respectivamente. En este caso, el modelo paramétrico viene dado por la distribución de probabilidad condicionada con función de densidad $P(y|\mathbf{x}, \boldsymbol{\omega})$, donde aparecen los parámetros $\boldsymbol{\omega} \in \mathbb{R}^L$. Es habitual considerar que el vector aleatorio está formado por variables aleatorias independientes e idénticamente distribuidas.

2.1.3. El aprendizaje

El aprendizaje en el contexto del machine learning consiste en encontrar un modelo y sus parámetros correspondientes de forma que el predictor resultante actúe adecuadamente sobre datos desconocidos. Para ello, en primer lugar tomaremos un subconjunto del dataset del que se dispone y, a partir del mismo, realizaremos el *entrenamiento* del modelo con el fin de estimar los parámetros más apropiados. A continuación, utilizaremos el subconjunto de datos que aún no ha sido procesado por el modelo (conjunto *test*) y servirán para comprobar la eficacia del mismo. Por último, se lleva a cabo la selección del modelo que se considera mejor entre los que se han podido obtener en las anteriores fases. En este trabajo, nos vamos a enfocar en el entrenamiento de los modelos pero se recomienda la lectura de [3] para profundizar en el resto de etapas.

Vamos a centrarnos en el caso de tener un dataset de entrenamiento etiquetado dado por $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\} \subset X \times Y \subset \mathbb{R}^D \times \mathbb{R}$. Como ya se ha mencionado previamente, entrenar un modelo equivale a estimar los parámetros $\boldsymbol{\omega}$ de f o P , según si lo hacemos desde una perspectiva determinista o probabilista. Para ello, se debe llevar a cabo un proceso de optimización de forma que los valores que se obtienen para los parámetros $\boldsymbol{\omega}$ minimicen la distancia entre el resultado dado por el predictor para las características $\mathbf{x}_i \in X$, $i = 1, \dots, N$, y su etiqueta correspondiente y_i , $i = 1, \dots, N$.

Dentro del caso probabilista, se puede definir la función de probabilidad L como

$$L(\boldsymbol{\omega}) = \prod_{i=1}^N P(y_i | \mathbf{x}_i, \boldsymbol{\omega}), \quad (2.2)$$

y sirve para medir la verosimilitud de realizaciones concretas de la variable aleatoria condicionada $Y|X$. El proceso de optimización en este caso consiste en maximizar (2.2) o, alternativamente (véase [3]), se puede definir el problema de minimización equivalente

$$\min_{\boldsymbol{\omega}} (-\log L(\boldsymbol{\omega})). \quad (2.3)$$

Si nos situamos en un contexto determinista, podemos definir la función error cuadrático medio a partir de los elementos de \mathcal{D} como

$$\epsilon(\boldsymbol{\omega}) = \frac{1}{N} \sum_{i=1}^N (f(\mathbf{x}_i, \boldsymbol{\omega}) - y_i)^2, \quad (2.4)$$

de forma que el entrenamiento consiste de nuevo en un proceso de minimización

$$\min_{\boldsymbol{\omega}} \epsilon(\boldsymbol{\omega}). \quad (2.5)$$

Así pues, a partir de ahora podemos continuar sin especificar el enfoque desde el que estamos considerando el predictor pues, tanto en el caso probabilista como en el determinista, la fase de entrenamiento consiste en el proceso de minimización ((2.3) o (2.5), respectivamente) de una función $\varphi(\boldsymbol{\omega})$ definida por (2.2) o por (2.4), según el caso.

A continuación, presentamos el *algoritmo de descenso del gradiente* como método para resolver el problema de minimización de $\varphi(\boldsymbol{\omega})$. Se construye tomando un primer vector $\boldsymbol{\omega}_0$ de forma aleatoria y, para $s \geq 0$, se tiene la relación de recurrencia

$$\boldsymbol{\omega}_{s+1} = \boldsymbol{\omega}_s - \gamma_s \nabla \varphi(\boldsymbol{\omega}_s), \quad (2.6)$$

donde $\gamma_s \in \mathbb{R}$ es la denominada *learning rate* y puede ser la misma en todos los pasos. Como podemos observar, este algoritmo genera una secuencia de vectores de parámetros de manera que los modifica avanzando hacia un mínimo local de la función $\varphi(\boldsymbol{\omega})$ por su pendiente más abrupta y γ_s ajusta la intensidad de la actualización en dicha dirección (véase [3]).

Observemos que no siempre llegaremos a un mínimo local mediante este procedimiento (por ejemplo, si γ_s es demasiado grande podríamos estar oscilando infinitamente alrededor de un mínimo local sin llegar a alcanzarlo nunca). También puede ocurrir que el tamaño de \mathcal{D} sea demasiado grande y que se produzcan problemas de eficiencia a nivel computacional. Para solventarlo, se utilizan variaciones del método como el algoritmo de descenso del gradiente estocástico, que toma subconjuntos aleatorios de \mathcal{D} . Por último, notemos que siempre que la función $\varphi(\boldsymbol{\omega})$ sea convexa, todos los mínimos locales también serán globales, de modo que el algoritmo puede llegar a converger a la solución global (véase [3]). Los casos que vamos a estudiar en este trabajo cumplen esta condición.

Podemos considerar también una versión en tiempo continuo de (2.6) que consiste en el problema de valor inicial de una ecuación diferencial de primer orden para una función $\boldsymbol{\omega} : \mathcal{T} \subseteq \mathbb{R} \rightarrow \mathcal{W} \subseteq \mathbb{R}^L$ de la forma

$$\frac{d\boldsymbol{\omega}(t)}{dt} = -\gamma(t)\nabla\varphi(\boldsymbol{\omega}(t)), \quad (2.7)$$

con valor inicial $\boldsymbol{\omega}(0)$ un vector aleatorio. Este algoritmo recibe el nombre de *algoritmo de flujo del gradiente*. Notemos que la versión discreta del algoritmo descrita en (2.6) puede ser implementada en un computador digital usual. Sin embargo, no puede hacerse directamente para el caso continuo (2.7), salvo si se recupera (2.6) discretizando el problema. En este trabajo, desarrollaremos distintas métodos para construir una máquina analógica que resuelva el problema de optimización en tiempo continuo (2.7), al que nos referiremos en adelante como *problema de optimización analógica*. En primer lugar, lo haremos mediante la teoría clásica de circuitos y, a continuación, utilizaremos los conceptos del Capítulo 1 para armar un algoritmo cuántico que resuelva el problema en cuestión.

2.2. Ejemplos de modelos de machine learning

En este apartado, analizamos dos modelos fundamentales dentro del aprendizaje automático: *regresión lineal* y *support vector machines* (SVM). En ambos casos, lo hacemos desde una perspectiva determinista.

2.2.1. Regresión lineal

Dado un conjunto de datos etiquetados $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\} \subset X \times Y \subset \mathbb{R}^D \times \mathbb{R}$, el método de regresión lineal estima el vector de parámetros $\boldsymbol{\omega} \in \mathbb{R}^D$ de forma que para un nuevo valor $\mathbf{x} \notin X$, la predicción correspondiente es

$$y = f(\mathbf{x}, \boldsymbol{\omega}) = \boldsymbol{\omega} \cdot \mathbf{x}. \quad (2.8)$$

Como podemos observar, (2.8) es un caso particular de (2.1) con $L = D$ pero podría generalizarse el método intercambiando \mathbf{x} por $\phi(\mathbf{x})$ y, como hemos explicado anteriormente, añadiendo un sesgo b .

Para entrenar este modelo, utilizamos el error cuadrático medio (2.4) para $f(\mathbf{x}, \boldsymbol{\omega}) = \boldsymbol{\omega} \cdot \mathbf{x}$, dando lugar a una función cuadrática positiva. Tenemos entonces que resolver el siguiente

problema de optimización

$$\min_{\omega} \left(\frac{1}{N} \sum_{i=1}^N (\omega \cdot \mathbf{x}_i - y_i)^2 \right), \quad (2.9)$$

cuya solución analítica es

$$\omega = (\mathbf{X}\mathbf{X}^t)^{-1}\mathbf{X}\mathbf{y}, \quad (2.10)$$

donde $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N) \in \mathbb{R}^{D \times N}$ es la matriz cuyas columnas son los vectores de características ordenados de \mathcal{D} e $\mathbf{y} = (y_1, \dots, y_N)^t \in \mathbb{R}^N$ es el vector columna formado por las etiquetas de \mathcal{D} . Es claro que $\mathbf{X}\mathbf{X}^t$ debe ser invertible.

La solución (2.10) puede generalizarse para el caso de un modelo que viene dado por $y = f(\mathbf{x}, \omega) = \omega \cdot \phi(\mathbf{x})$. La expresión de la solución en este caso es

$$\omega = (\Phi\Phi^t)^{-1}\Phi\mathbf{y},$$

donde $\Phi = (\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_N)) \in \mathbb{R}^{D \times N}$ es la matriz cuyas columnas son los vectores de salida del mapa de características correspondiente. Igual que antes, se requiere que $\Phi\Phi^t$ sea invertible.

2.2.2. Support vector machines (SVM)

Este modelo sirve para tratar con problemas de clasificación binaria donde el conjunto de datos $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\} \subset X \times Y \subset \mathbb{R}^D \times \mathbb{R}$ tiene la peculiaridad de que $Y = \{-1, +1\}$. El objetivo es encontrar un plano $\pi = \{\mathbf{x} \in \mathbb{R}^D : \omega \cdot \mathbf{x} + b = 0\} \subset \mathbb{R}^D$ que sirva como frontera entre los datos etiquetados con $y = -1$ y los que tienen la etiqueta $y = +1$. Notemos que aquí ω representa el vector normal al plano π .

Ahora, suponiendo que ya tenemos dicho plano, como se trata de una superficie orientable, podemos utilizar la siguiente función de clasificación como predictor para una nueva muestra desconocida $\mathbf{x} \notin X$

$$y = \text{sgn}(\omega \cdot \mathbf{x} + b).$$

Sin entrar en detalles, el problema de optimización a resolver es el denominado problema primigenio

$$\min_{\omega, b} \left(\frac{1}{2} \|\omega\|^2 : y_i \cdot (\omega \cdot \mathbf{x}_i + b) \geq 1, \forall (\mathbf{x}_i, y_i) \in \mathcal{D} \right), \quad (2.11)$$

donde no sólo hay que minimizar una función sino que, además, se tienen N restricciones. El teorema de Karush-Kuhn-Tucker ([3]) garantiza la existencia de constantes no negativas $\alpha_1, \dots, \alpha_N$ tales que el problema (2.11) equivale al siguiente

$$\min_{\omega, b} \left(\frac{1}{2} \|\omega\|^2 + \sum_{i=1}^N \alpha_i (1 - y_i \cdot (\omega \cdot \mathbf{x}_i + b)) \right). \quad (2.12)$$

Se puede considerar una versión dual del problema (2.12) de la forma

$$\min_{\alpha} \left(\frac{1}{2} \sum_{i,j=1}^N Q_{ij} \alpha_i \alpha_j - \sum_{i=1}^N \alpha_i : \alpha \cdot \mathbf{y} = 0, \alpha_i \geq 0 \right), \quad (2.13)$$

con $Q_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$, $i, j = 1, \dots, N$. Una vez resuelto el problema dual (2.13), la solución del problema primigenio viene dada por

$$\omega = \sum_{i \in S} \alpha_i y_i \mathbf{x}_i, \quad b = \frac{1}{|S|} \sum_{i \in S} (y_i - \omega \cdot \mathbf{x}_i),$$

donde $S = \{i \in \{1, \dots, N\} : \alpha_i \neq 0\}$.

De forma análoga a lo que se hizo en el modelo anterior, podemos generalizar este resultado al caso de buscar un plano $\pi = \{\mathbf{x} \in \mathbb{R}^D : \boldsymbol{\omega} \cdot \phi(\mathbf{x}) + b = 0\} \subset \mathbb{R}^D$. Este caso puede darse cuando el conjunto de características $\{\mathbf{x}_i\}_{i=1}^N \subset X$ no sea linealmente separable pero $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ sí lo sea. El problema de optimización dual es entonces

$$\min_{\boldsymbol{\alpha}} \left(\frac{1}{2} \sum_{i,j=1}^N Q'_{ij} \alpha_i \alpha_j - \sum_{i=1}^N \alpha_i : \boldsymbol{\alpha} \cdot \mathbf{y} = 0, \alpha_i \geq 0 \right),$$

con $Q'_{ij} = y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, $i, j = 1, \dots, N$.

No vamos a entrar en más detalles sobre el machine learning pues se sale de los objetivos de este trabajo, pero la información expuesta y los recursos recomendados son suficientes para comprender el valor que tiene la fase de entrenamiento en la construcción de modelos. En el siguiente capítulo, veremos cómo la física puede jugar un papel crucial en la búsqueda de caminos que nos permitan llevar a cabo esta tarea de manera eficiente pese a que, en principio, no veamos una relación directa entre ambas ramas de la ciencia.

Capítulo 3

El problema de optimización

En este capítulo, mostraremos una alternativa para la resolución de un problema de machine learning clásico utilizando la física como instrumento. En concreto, vamos a centrarnos en los problemas de optimización del modelo de regresión lineal (2.9) y el de SVM (2.13), incluyendo sus versiones utilizando mapa de características descritas en la Sección 2.2. Es habitual que la fase de entrenamiento sea la que requiera más tiempo en su ejecución y la búsqueda de caminos más eficientes para llevarla a cabo es un tema que actualmente se está investigando. En este trabajo, el objetivo es construir un sistema físico cuya dinámica se rija por (2.7) de forma que, tras un tiempo lo suficientemente largo, éste nos proporcione los valores de los parámetros adecuados ω tras el entrenamiento del modelo en cuestión. Comenzaremos con un breve análisis del problema de optimización y será resuelto en las secciones sucesivas a través de dos alternativas, una clásica y otra cuántica.

Las principales fuentes que se han utilizado para el desarrollo del capítulo son [3], [4], [12], [13], [14] y [5].

3.1. Descripción del problema

En primer lugar, generalizamos las optimizaciones (2.9) y (2.13) a partir de una función $\varphi : Q \subset \mathbb{R}^m \rightarrow \mathbb{R}$ de la forma

$$\varphi(\mathbf{q}) = \frac{1}{2} \sum_{i,j=1}^m \Lambda_{ij} q_i q_j - \sum_{i=1}^m p_i q_i, \quad (3.1)$$

donde la matriz $\Lambda = (\cdots \Lambda_{ij} \cdots) \in \mathbb{R}^{m \times m}$ y el vector columna $\mathbf{p} = (\cdots p_i \cdots)^t \in \mathbb{R}^m$ están fijos. En la Tabla 3.1, se encuentra el resumen de la notación de cada modelo, utilizando en ambos casos la versión general con mapa de características. Observemos que en el modelo de regresión se tiene $m = D$ mientras que para SVM, $m = N$. También se utiliza la notación $\mathbf{e} = (1, \dots, 1)^t \in \mathbb{R}^N$ y $\Psi = \Phi \mathbf{Y}$ con $\mathbf{Y} = \text{diag}(\mathbf{y})$.

$\min \varphi$	\mathbf{q}	Λ	\mathbf{p}
Regresión	ω	$\Phi \Phi^t$	$\Phi \mathbf{y}$
SVM	α	$(\Phi \mathbf{Y})^t \Phi \mathbf{Y}$	\mathbf{e}

Tabla 3.1: Resumen de la notación en los problemas de optimización de la regresión lineal y de SVM según la ecuación general (3.1).

Ahora, podemos abordar estos problemas en su versión analógica mediante el algoritmo del flujo del gradiente (2.7) aplicado a la función φ . Según (3.1), dicho algoritmo toma la forma

$$\frac{d\mathbf{q}(t)}{dt} = -\gamma (\mathbf{\Lambda}\mathbf{q}(t) - \mathbf{p}), \quad (3.2)$$

con valor inicial $\mathbf{q}(0)$ un vector aleatorio y learning rate $\gamma > 0$ constante. La solución analítica de (3.2) viene dada por

$$\mathbf{q}(t) = \exp(-\gamma\mathbf{\Lambda}t)\mathbf{q}(0) + (I - \exp(-\gamma\mathbf{\Lambda}t))\mathbf{\Lambda}^{-1}\mathbf{p}, \quad (3.3)$$

y, tomando el límite cuando $t \rightarrow \infty$ en (3.3) y teniendo en cuenta que $\mathbf{\Lambda}$ es una matriz definida positiva, obtenemos la solución estacionaria \mathbf{q}^* de (3.2)

$$\mathbf{q}^* = \mathbf{\Lambda}^{-1}\mathbf{p}. \quad (3.4)$$

En la práctica, se puede dar una estimación de un tiempo finito para el que la máquina podría entrenar el modelo en cuestión. Dicho tiempo viene dado por el máximo tiempo propio de decaimiento $\tau = (\gamma\lambda_\mu)^{-1}$, donde λ_μ es el menor autovalor de $\mathbf{\Lambda}$.

Observemos que \mathbf{q}^* y τ no dependen en ningún caso del vector inicial $\mathbf{q}(0)$ y, además, \mathbf{q}^* también es independiente del valor γ . Sin embargo, el tiempo τ es inversamente proporcional al learning rate, de modo que cuanto mayor sea dicha constante, más rápido se producirá el entrenamiento.

En las siguientes secciones, analizaremos las alternativas que ofrece la física para resolver este problema siguiendo un camino clásico y otro cuántico.

3.2. Máquina analógica clásica

Veamos ahora cómo construir una máquina analógica que codifique el problema de optimización analógico obtenido a partir del algoritmo de flujo de gradiente resumido en (3.2).

En [4], se trata con un modelo de tipo red neuronal recurrente (RN), cuya versión en tiempo continuo es implementada por una red eléctrica (RE). Es decir, existe una equivalencia $\text{RN} \simeq \text{RE}$. Aquí, desarrollaremos cómo utilizar esa arquitectura circuital para codificar la ecuación diferencial (3.2), por lo que nos respaldaremos en [12] y [13] para el desarrollo de esta sección.

Es importante entender que las redes neuronales recurrentes son modelos de machine learning distintos a los que hemos visto hasta ahora y su análisis se sale de los objetivos del trabajo. La clave es que la ecuación de evolución de estas redes tiene la misma forma que (3.2), por lo que podemos seguir la metodología desarrollada en [4] para estudiar el caso que nos ocupa.

La Figura 3.1 muestra la topología general de la red eléctrica que se va a utilizar. Se trata de m dipolos $\{\mathcal{N}_k\}_{k=1}^m$ conectados a una red central \mathcal{N} , denominada red de interacción.

Para analizar la arquitectura de los dipolos, utilizaremos la representación en tensiones de los mismos (véase la Figura 3.2), aunque también podría hacerse mediante su versión dual, que utiliza corrientes e inductancias ([12], [13]). Así, cada dipolo \mathcal{N}_i se compone de un capacitor C_i y un generador de corriente J_i , conectados a una red de m puertas. En cada una de ellas, entra una corriente I_i , de forma que todos los terminales b_i se conectan

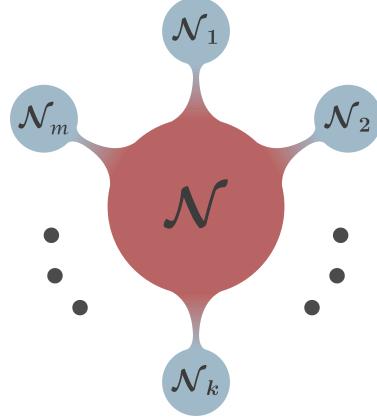


Figura 3.1: Representación del circuito eléctrico que se utiliza para la optimización analógica. Éste se compone de m dipolos eléctricos \mathcal{N}_k interconectados a través de la red de interacción \mathcal{N} [4].

a la misma referencia (es decir, el potencial nulo). La diferencia de potencial entre el par de terminales a_i y b_i se denota por V_i , que es a su vez la diferencia de potencial del condensador C_i del dipolo \mathcal{N}_i .

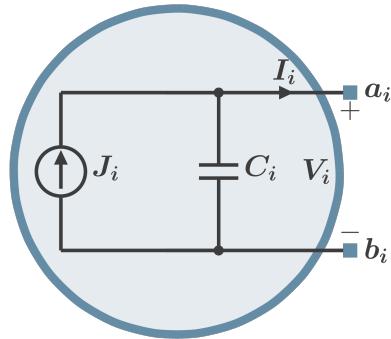


Figura 3.2: Esquema del i -ésimo dipolo conectado a la red central de interacción \mathcal{N} .

Aplicando la segunda ley de Kirchoff (véase [12], [13]) sobre el nodo a_i , se tiene que $I_{C_i} + I_i = J_i$, con $I_{C_i} = C_i \frac{dV_i}{dt}$. Es decir, para cada $i = 1, \dots, m$ se verifica

$$C_i \frac{dV_i}{dt} = -I_i + J_i. \quad (3.5)$$

Ahora, nuestro objetivo es que cada tensión V_i se corresponda con la componente i -ésima del vector de parámetros \mathbf{q} del problema de optimización analógico, por lo que la expresión (3.5) deberá adoptar la forma de (3.2). Para ello, lo único que falta es tener una relación lineal entre I_i y cada una de las diferencias de potencial $\{V_1, \dots, V_m\}$, es decir, que se cumpla

$$I_i = \sum_{j=1}^m G_{ij} V_j, \quad (3.6)$$

para ciertos valores constantes G_{ij} , $i, j = 1, \dots, m$. Esto puede sintetizarse mediante un circuito multiplicador como el que se muestra en la Figura 3.3, donde G_{ij} denota cada conductancia. Éste se corresponde con la red de m puertos denotada por \mathcal{N} en la Figura 3.1. Recordemos que la matriz de conductancias contiene los valores de la resistencia inversa entre los nodos de entrada y salida, de forma que cada elemento de la matriz representa la conductancia entre un par de nodos específicos. Así, cada señal de salida es una combinación lineal de las señales de entrada ponderadas por las conductancias correspondientes (véase [12], [13]).

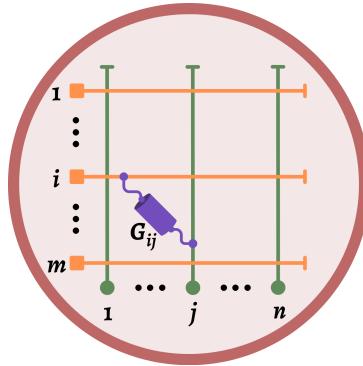


Figura 3.3: Circuito multiplicador matricial de un vector de m componentes a otro de n componentes, sintetizado mediante una matriz de conductancias $m \times n$.

Uniendo (3.5) y (3.6) en un formato matricial, obtenemos la siguiente ecuación diferencial

$$\frac{d\mathbf{v}}{dt} = -\mathbf{c}^{-1}\mathbf{G}\mathbf{v} + \mathbf{c}^{-1}\mathbf{J}, \quad (3.7)$$

con $\mathbf{v} = (V_1, \dots, V_m)^t \in \mathbb{R}^m$, $\mathbf{c} = \text{diag}(C_1, \dots, C_m) \in \mathbb{R}^{m \times m}$, $\mathbf{G} = (\dots G_{ij} \dots) \in \mathbb{R}^{m \times m}$ y $\mathbf{J} = (J_1, \dots, J_m) \in \mathbb{R}^m$. Si comparamos (3.7) con (3.2), obtenemos la relación entre los elementos asociados al modelo de ML a entrenar (SVM o regresión lineal, en nuestro caso) y los de una red eléctrica RE en la que puede implementarse el algoritmo de flujo del gradiente (véase la Tabla 3.2).

ML	\mathbf{q}	$\gamma\mathbf{\Lambda}$	$\gamma\mathbf{p}$
RE	\mathbf{v}	$\mathbf{c}^{-1}\mathbf{G}$	$\mathbf{c}^{-1}\mathbf{J}$

Tabla 3.2: Resumen de la identificación entre los elementos de las ecuaciones de evolución de los modelos de regresión y SVM (ML) con los de redes eléctricas (RE), (3.7) y (3.2), respectivamente.

Si, además, todos los condensadores se toman iguales, es decir, $C_i = c$ para $i = 1, \dots, m$ con c una constante estrictamente positiva, entonces obtenemos la asignación que se muestra en la Tabla 3.3.

Observemos que la relación entre el learning rate γ y la capacidad inversa de los condensadores c^{-1} puede ser interpretada físicamente. Tal y como se explica en la Sección 3.1, para cierto tiempo finito $\tau = (\gamma\lambda_\mu)^{-1}$, el sistema gobernado por la ecuación (3.2) se relaja hacia la solución estacionaria \mathbf{q}^* dada por (3.4). Es decir, transcurrido ese tiempo, podremos afirmar que el sistema está entrenado, ha aprendido. Ocurre un proceso similar

ML	\mathbf{q}	$\mathbf{\Lambda}$	\mathbf{p}	γ
RE	\mathbf{v}	\mathbf{G}	\mathbf{J}	c^{-1}

Tabla 3.3: Resumen de la identificación entre los elementos de las ecuaciones de evolución de los modelos de regresión y SVM (ML) con los de redes eléctricas (RE), (3.7) y (3.2), respectivamente, cuando todos los condensadores son iguales.

con la red eléctrica para un tiempo dado por cg_μ^{-1} (donde g_μ denota el mínimo autovalor de la matriz de conductancias \mathbf{G}). Aquí, reducir el valor de c acelera el aprendizaje, haciendo que el tiempo de entrenamiento sea menor, pero notemos que no podemos tomar $c = 0$ pues perderíamos la evolución temporal del circuito.

La codificación que revela la Tabla 3.3 permite que el circuito en cuestión sea inicializado colocando diferencias de potencial aleatorias en cada condensador $C_i = c$, es decir, el vector inicial $\mathbf{v}(0)$ se puede elegir al azar. A su vez, la matriz de conductancias \mathbf{G} debe estar sintonizada con los datos de entrenamiento según la Tabla 3.1. Así, tras un tiempo cg_μ^{-1} , la lectura de las diferencias de potencial en cada puerta de la red \mathcal{N} se corresponderá con los valores de los parámetros del modelo de machine learning ya entrenado.

3.3. Máquinas cuánticas

Igual que se ha construido una máquina analógica capaz de llevar a cabo la fase de entrenamiento de un modelo de aprendizaje automático utilizando circuitos eléctricos, cabe preguntarse si es posible hacer lo mismo mediante un algoritmo cuántico. En este apartado, veremos que, en efecto, existen alternativas cuánticas para la optimización de (3.1).

Recordemos que el algoritmo de flujo de gradiente (3.2) sirve para obtener los parámetros que definen un modelo de machine learning y, a la hora de implementarlo mediante un algoritmo cuántico, existen dos posibilidades para abordarlo: la estática y la dinámica.

La opción estática se basa en la convergencia de (3.3) hacia la solución estacionaria \mathbf{q}^* dada por (3.4). En particular, se trata de resolver el sistema de ecuaciones $\mathbf{\Lambda q} = \mathbf{p}$. Existen algoritmos cuánticos como el de HHL (véase [14]) que implementan la solución de ecuaciones lineales algebraicas, pero en este trabajo nos centraremos en la alternativa dinámica.

El camino dinámico para la resolución del problema se lleva a cabo mediante la codificación de la ecuación diferencial (3.2) en un sistema cuántico, de forma que el mismo nos permitirá llegar a la solución (3.3) y una medida posterior en este sistema dará lugar a (3.4). El artículo [5] ofrece un algoritmo cuántico para la resolución de ecuaciones diferenciales lineales como la de nuestro problema y lo vamos a tomar como alternativa cuántica para el entrenamiento de los modelos de regresión lineal y SVM.

3.3.1. Marco de trabajo

En primer lugar, consideraremos la aproximación a la solución (3.3) dada por los desarrollos en serie de Taylor de orden n de las exponenciales que aparecen en dicha expresión, que toma la forma

$$\mathbf{q}(t) \approx \left(\sum_{k=0}^n \frac{(-\gamma t)^k}{k!} \mathbf{\Lambda}^k \right) \mathbf{q}(0) - \left(\sum_{l=1}^n \frac{(-\gamma t)^l}{l!} \mathbf{\Lambda}^{l-1} \right) \mathbf{p}. \quad (3.8)$$

A continuación, construimos los estados asociados a un sistema cuántico teniendo en cuenta la estructura de (3.8). Para ello, tomamos una base ortonormal \mathcal{B} de un espacio de Hilbert \mathcal{H}

$$\mathcal{B} = \{|j\rangle : j = 1, \dots, m\},$$

donde m es la dimensión de \mathbf{q} . Los estados a considerar codifican cada una de las componentes de los vectores que aparecen en (3.8) como sus amplitudes de probabilidad y la matriz Λ se identifica con el operador lineal en \mathcal{H} , $\hat{\Lambda}$. Explícitamente, tenemos que

$$|\mathbf{q}(0)\rangle = \sum_{j=1}^m \frac{q_j(0)}{\|\mathbf{q}(0)\|} |j\rangle, \quad (3.9)$$

$$|\mathbf{p}\rangle = \sum_{j=1}^m \frac{p_j}{\|\mathbf{p}\|} |j\rangle, \quad (3.10)$$

$$\hat{\Lambda} = \sum_{i,j=1}^m \frac{\Lambda_{ij}}{\|\Lambda\|} |i\rangle \langle j|, \quad (3.11)$$

donde $\|\cdot\|$ es la norma vectorial euclídea (si $\mathbf{v} \in \mathbb{R}^m$, $\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_m^2}$) y $\|\Lambda\|$ es la norma matricial de Λ inducida por la anterior, que es la raíz cuadrada del autovalor máximo de $\Lambda^t \Lambda$. Combinando las expresiones (3.9)-(3.11) con (3.8), obtenemos una aproximación a la solución del problema, que toma la forma del vector correspondiente a un sistema cuántico que evoluciona en el tiempo como

$$\begin{aligned} |\mathbf{q}(t)\rangle &\approx \sum_{k=1}^n \|\mathbf{q}(0)\| \frac{(-\gamma t)^k}{k!} \left(\|\Lambda\| \hat{\Lambda} \right)^k |\mathbf{q}(0)\rangle \\ &\quad - \sum_{l=1}^n \|\mathbf{p}\| \frac{(-\gamma t)^l}{l!} \left(\|\Lambda\| \hat{\Lambda} \right)^{l-1} |\mathbf{p}\rangle. \end{aligned} \quad (3.12)$$

Tal y como se explica en [5], el algoritmo presenta dos variantes según si el operador $\hat{\Lambda}$ es unitario o no. Por lo general, el problema de optimización que estamos estudiando, dentro del contexto del aprendizaje automático, no verificará esta condición. Pese a ello, en adelante, nos restringimos al caso en el que el operador sí es unitario pues ofrece la posibilidad de estudiar el algoritmo de una forma más sencilla para su comprensión y es suficiente para cumplir con los objetivos de este trabajo. El caso no unitario se basa en que $\hat{\Lambda}$ puede descomponerse como una combinación lineal de operadores unitarios y requiere más qubits, por lo que se vuelve algo más complejo.

Como $\hat{\Lambda}$ lo estamos suponiendo unitario, sus potencias también lo serán. Ahora, bajo la notación $U_k = (-\hat{\Lambda})^k$, $k \in \mathbb{R}$, la expresión (3.12) toma la forma

$$|\mathbf{q}(t)\rangle \approx \frac{1}{A^2} \left(\sum_{k=1}^n C_k U_k |\mathbf{q}(0)\rangle + \sum_{l=1}^n D_l U_{l-1} |\mathbf{p}\rangle \right), \quad (3.13)$$

con

$$\begin{aligned} C_k &= \frac{\|\mathbf{q}(0)\| \|\Lambda\|^k (\gamma t)^k}{k!}, \quad \text{para } k = 0, \dots, n, \\ D_l &= \frac{\|\mathbf{p}\| \|\Lambda\|^{l-1} (\gamma t)^l}{l!}, \quad \text{para } l = 1, \dots, n, \end{aligned}$$

y, siguiendo [5], $A^2 = \mathcal{C}^2 + \mathcal{D}^2$ ($\mathcal{C} = \sqrt{\sum C_k}$ y $\mathcal{D} = \sqrt{\sum D_l}$) es la constante de normalización del estado, de modo que la componente j -ésima de la solución $\mathbf{q}(t)$ es $q_j(t) = A^2 \langle j | \mathbf{q}(t) \rangle$.

3.3.2. El algoritmo

El diagrama del circuito cuántico que se va a utilizar es el que se muestra en la Figura 3.4. Dividiremos el algoritmo en cuatro fases: codificación, generación de estados entrelazados, decodificación y medición.

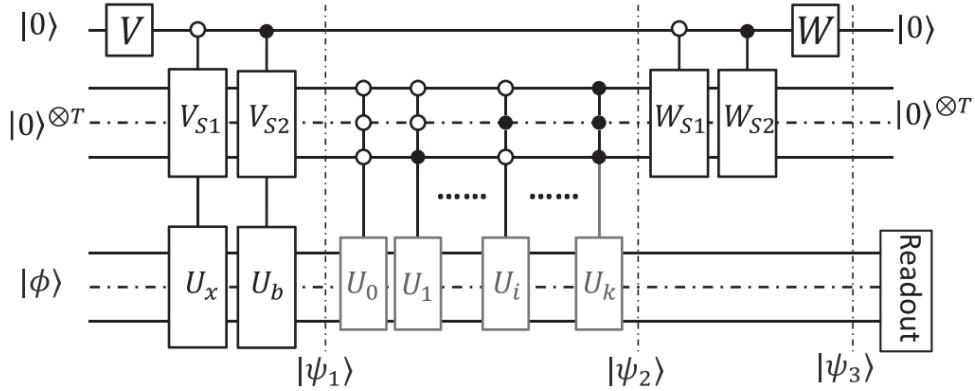


Figura 3.4: Circuito cuántico para resolver el problema de optimización de (3.1) cuando el operador $\hat{\Lambda}$ es unitario [5].

Fase de codificación

El circuito cuántico que se muestra en la Figura 3.4 está formado por un sistema principal de qubits y dos auxiliares (también denominados *ancillas*). Por un lado, para codificar vectores de m componentes, necesitamos $\log_2(m)$ qubits principales. Los estados $|\mathbf{q}(0)\rangle$ y $|\mathbf{p}\rangle$ se preparan y almacenan en los qubits principales y se encuentran etiquetados por el primer ancilla como $|0\rangle|\mathbf{q}(0)\rangle$ y $|1\rangle|\mathbf{p}\rangle$, respectivamente. Es decir, el primer ancilla está formado por un sólo qubit. El segundo ancilla está formado por $T = \log_2(n+1)$ qubits, donde recordemos que n es el orden de truncamiento del desarrollo de Taylor generado al aproximar la solución (3.3).

Supongamos que el estado de entrada del sistema de qubits principales es $|\phi\rangle$ y que todos los qubits de los ancillas están en el estado inicial $|0\rangle$. La primera puerta cuántica que aparece, V , viene dada por la matriz

$$V = \frac{1}{A} \begin{pmatrix} \mathcal{C} & \mathcal{D} \\ \mathcal{D} & -\mathcal{C} \end{pmatrix},$$

de modo que el primer sistema ancilla formado por un sólo qubit evoluciona hacia un estado de superposición.

Los estados $|\mathbf{q}(0)\rangle$ y $|\mathbf{p}\rangle$ se forman haciendo actuar las puertas controladas U_x y U_b , respectivamente, sobre el estado de entrada $|\phi\rangle$, dependiendo del estado del qubit del primer ancilla. De este modo, el estado inicial $|0\rangle|0\rangle^{\otimes T}|\phi\rangle$ pasa a ser

$$\frac{\mathcal{C}}{A}|0\rangle|0\rangle^{\otimes T}|\mathbf{q}(0)\rangle + \frac{\mathcal{D}}{A}|1\rangle|0\rangle^{\otimes T}|\mathbf{p}\rangle.$$

A continuación, definimos las puertas controladas V_{S1} y V_{S2} mediante su expresión

matricial

$$V_{S1} = \frac{1}{\mathcal{C}} \begin{pmatrix} \sqrt{C_0} & Q & \cdots & Q \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{C_n} & Q & \cdots & Q \end{pmatrix}_{(n+1) \times (n+1)},$$

$$V_{S2} = \frac{1}{\mathcal{D}} \begin{pmatrix} \sqrt{D_1} & Q & \cdots & Q \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{D_n} & Q & \cdots & Q \\ 0 & Q & \cdots & Q \end{pmatrix}_{(n+1) \times (n+1)},$$

donde los valores Q indican elementos arbitrarios en las matrices bajo la condición de hacer que V_{S1} y V_{S2} sean unitarias. Estas puertas operan sobre los qubits del segundo ancilla, de forma que el estado total que se obtiene tras su actuación es

$$|\psi_1\rangle = \frac{1}{A} \left(|0\rangle \sum_{k=0}^n \sqrt{C_k} |k\rangle |\mathbf{q}(0)\rangle + |1\rangle \sum_{l=1}^n \sqrt{D_l} |l-1\rangle |\mathbf{p}\rangle \right).$$

Resumiendo, para obtener el estado $|\psi_1\rangle$ a partir del estado inicial $|0\rangle |0\rangle^{\otimes T} |\phi\rangle$, la operación a realizar es $|0\rangle \langle 0| \otimes V_{S1} \otimes U_x + |1\rangle \langle 1| \otimes V_{S2} \otimes U_b$. Observemos que esta parte del circuito puede verse desde dos perspectivas que generan el mismo resultado, tal y como se indica en la Figura 3.5.

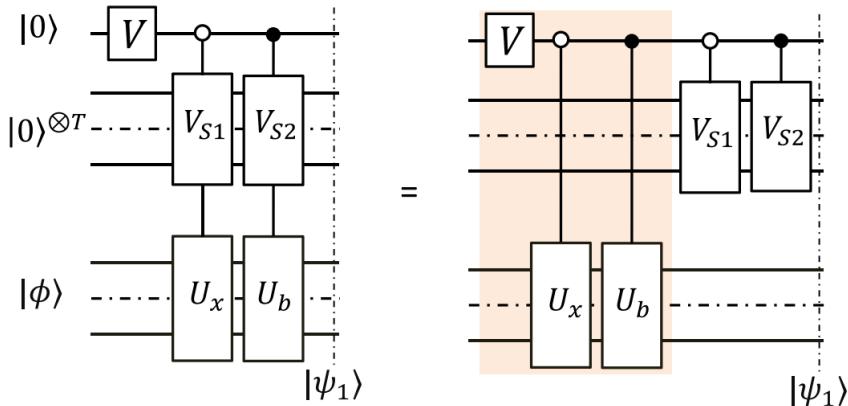


Figura 3.5: Fragmento del circuito cuántico para resolver el problema de optimización de (3.1) cuando el operador $\hat{\Lambda}$ es unitario. Esta parte se corresponde con la fase de codificación para preparar el estado $|\psi_1\rangle$ [5].

Fase de generación de estados entrelazados

El siguiente paso consiste en aplicar una serie de operaciones controladas por el segundo ancilla sobre el conjunto de qubits principal de la forma $U_c = \sum_{j=0}^n |j\rangle \langle j| \otimes U_j$, donde recordemos que $U_k = (-\hat{\Lambda})^k$. Así, el sistema principal y el segundo ancilla quedan entrelazados, dando lugar al estado total

$$|\psi_2\rangle = \frac{1}{A} \left(|0\rangle \sum_{k=0}^n \sqrt{C_k} |k\rangle U_k |\mathbf{q}(0)\rangle + |1\rangle \sum_{l=1}^n \sqrt{D_l} |l-1\rangle U_{l-1} |\mathbf{p}\rangle \right).$$

Fase de decodificación

Dado que las puertas cuánticas son siempre reversibles, todas las operaciones llevadas a cabo durante la fase de codificación pueden aplicarse de forma inversa. Para ello, se debe aplicar $|0\rangle\langle 0| \otimes W_{S1} + |1\rangle\langle 1| \otimes W_{S2}$ sobre el segundo ancilla controlado por el primero, donde $W_{S1} = V_{S1}^\dagger$ y $W_{S2} = V_{S2}^\dagger$. El estado resultante en el subespacio donde todos los qubits pertenecientes al segundo ancilla son $|0\rangle^{\otimes T}$ es

$$\frac{1}{A} \left(|0\rangle|0\rangle^{\otimes T} \sum_{k=0}^n \frac{C_k}{\mathcal{C}} U_k |\mathbf{q}(0)\rangle + |1\rangle|0\rangle^{\otimes T} \sum_{l=1}^n \frac{D_l}{\mathcal{D}} U_{l-1} |\mathbf{p}\rangle \right).$$

Por último, se opera con la puerta $W = V^\dagger$ sobre el primer ancilla. Llegados a este punto, el único subespacio relevante es aquel en el que todos los qubits de los ancillas son $|0\rangle$, donde el estado del sistema completo en este subespacio es

$$|\psi_3\rangle = \frac{1}{A^2} |0\rangle|0\rangle^{\otimes T} \left(\sum_{k=0}^n C_k U_k |\mathbf{q}(0)\rangle + \sum_{l=1}^n D_l U_{l-1} |\mathbf{p}\rangle \right).$$

Fase de medición

El último paso consiste en medir el estado final del conjunto principal de qubits en el subespacio en el que todos los qubits de los ancillas son $|0\rangle$. Observando (3.13), es claro que se obtiene la solución directamente sobre un factor A^2 y con una probabilidad de acierto de, aproximadamente, $\frac{1}{A^4}$.

Para obtener la solución estacionaria (3.4), que es lo que nos interesa, bastará con medir el estado final para varias realizaciones transcurrido al menos un tiempo $\tau = (\gamma\lambda_\mu)^{-1}$, que recordemos que era el tiempo propio de decaimiento según veíamos en la Sección 3.1. De este modo, obtendremos la distribución de probabilidad de los distintos posibles estados finales y concluiremos que los parámetros del modelo dados por (3.4) son aquellos cuya probabilidad se corresponde con $\frac{1}{A^4}$.

3.3.3. Algunas observaciones

Resumimos brevemente el análisis que se hace en [5] sobre la complejidad del algoritmo descrito en el apartado anterior. La rapidez de este algoritmo proviene de la libertad para procesar información adquirida gracias a la presencia de los sistemas de qubits auxiliares, que permiten la creación de estados en superposición y la aplicación de operaciones controladas sobre el conjunto principal de qubits.

Desde un punto de vista físico, tenemos que se están implementando varias puertas cuánticas de forma simultánea sobre el sistema principal, pero en diferentes subespacios. A continuación, todas las operaciones de la primera fase se aplican en su forma inversa. Es decir, se produce una combinación de la información que hay en los diferentes subespacios para obtener un resultado final dado por el subespacio en el que los dos ancillas son $|0\rangle$. Tal y como se puede deducir de la Figura 3.4, los fenómenos cuánticos que juegan un papel fundamental en este proceso son la superposición y el entrelazamiento.

Por otro lado, para comprender el coste del algoritmo, se debe tener en cuenta la complejidad del sistema de puertas utilizado, que para el caso que hemos estudiado es $O(n \log_2(n) \log_2(m) + \log_2(m) + n^2)$, frente al tiempo requerido por un ordenador clásico

para aproximar la solución, que es de al menos $O(m^3)$. Es claro que únicamente depende del orden del desarrollo de Taylor utilizado para la aproximación de la solución y de la dimensión del vector \mathbf{q} , que en nuestro caso es el número de parámetros del modelo que se quiere entrenar. En general, un orden n relativamente bajo será suficiente para llegar a un buen resultado en un tiempo razonable.

Otro factor importante en la complejidad del algoritmo es la cantidad de qubits a utilizar. Recordemos que necesitamos un total de $1 + \log_2(n + 1)$ qubits para los ancillas y, además, se verifica $n = \ln\left(\frac{C_0}{\epsilon}\right)$, donde C_0 es una constante y ϵ es el error entre la solución ideal y la aproximada. Igual que antes, un orden n relativamente bajo será suficiente para obtener buenos resultados en este sentido.

Conclusiones

Este trabajo presenta una muestra de la importancia de la física para la resolución de problemas que, aparentemente, no tienen que ver con ella. Comenzando por la base teórica de la computación cuántica, se han estudiado principios cuánticos y conceptos como la superposición cuántica y el entrelazamiento, que se revelan no sólo como curiosidades sino como las bases para abordar problemas computacionales complejos.

Por otro lado, con el fin de ilustrar una de las aplicaciones de la computación cuántica, hemos tenido la oportunidad de conocer el campo del aprendizaje automático o *machine learning*, tratándose de uno de los temas de mayor relevancia dentro del panorama científico actual por sus múltiples aplicaciones. Con este trabajo, hemos podido resumir sus puntos clave, llegando a ver dos modelos tan importantes como la regresión lineal y el SVM.

Al dar nuestros primeros pasos dentro del ámbito del aprendizaje automático, hemos podido comprobar la importancia de llevar a cabo el entrenamiento de los modelos de manera eficiente, pues esta fase es la que más tiempo y energía exige y es determinante para obtener los parámetros adecuados. Es un aspecto vital para la construcción de modelos, por lo que la búsqueda de la eficiencia en su ejecución tiene mucho interés para la comunidad científica.

En particular, hemos analizado el problema de optimización analógica para los modelos de regresión lineal y de SVM, que surge a partir de la versión continua del algoritmo de descenso del gradiente y no puede implementarse digitalmente de forma directa. Por ello, hemos querido dar una primera alternativa física desde una perspectiva clásica que, mediante la teoría de circuitos, nos ha permitido construir una máquina analógica capaz de llevar a cabo esta fase de entrenamiento. A continuación, hemos podido comprobar que existen algoritmos cuánticos capaces de resolver este mismo problema y su descripción ha sido un ejemplo de aplicación de los conceptos teóricos que estudiamos al principio del trabajo.

Así pues, consideramos que se han alcanzado los objetivos propuestos puesto que no sólo hemos dado una introducción teórica de la computación cuántica sino que hemos podido ver una aplicación de la misma que, a su vez, nos ha permitido ampliar nuestros conocimientos en torno a un ámbito tan sustancial como el aprendizaje automático. Además, este trabajo sirve como motivación para continuar estudiando este tema en el futuro, llevando a cabo la implementación del algoritmo desarrollado sobre algún problema concreto. De hecho, el algoritmo en cuestión sirve para resolver cualquier ecuación diferencial lineal de primer orden, presentes en multitud de ámbitos y que no tienen por qué tratarse únicamente de aquellas que se obtienen a partir del entrenamiento de modelos de aprendizaje automático, por lo que el análisis del mismo es de gran interés. Del mismo modo, el algoritmo HHL, que no se ha analizado en profundidad pero que sí hemos mencionado, sirve para obtener matrices inversas mediante circuitos cuánticos, por lo que la investigación más a fondo del mismo presenta oportunidades prometedoras en cuanto a futuras exploraciones

para resolver este problema computacional tan conocido.

Para cerrar el trabajo, cabe destacar el papel fundamental de la física como una herramienta impulsora para diversos campos científicos, ejemplificado en su aplicación dentro del aprendizaje automático. La física, con sus principios fundamentales que rigen el comportamiento del mundo que nos rodea, no sólo establece los cimientos de su propia disciplina, sino que también actúa como un propulsor esencial para la innovación en otras áreas científicas. Esta relación con otras ramas de la ciencia refleja la naturaleza colaborativa de la investigación. Reconocer y fomentar esta interconexión, no sólo amplia nuestras capacidades científicas, sino que también nos permite llegar a territorios aún por descubrir.

Bibliografía

- [1] R. P. Feynman, *Simulating Physics with Computers*, Int. J. Theor. Phys. **21** (1982), pp. 467-488.
- [2] M. A. Nielsen & I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2010).
- [3] M. P. Deisenroth, A. A. Faisal & Cheng Soon Ong, *Mathematics for Machine Learning*, Cambridge University Press (2020).
- [4] M. Caruso & C. Jarne, *Recurrent Neural Networks as Electrical Networks, a Formalization*, DCAI 2022. Lecture Notes in Networks and Systems, Springer (2023), pp. 105-114.
- [5] T. Xin, et al. *Quantum algorithm for solving linear differential equations: Theory and experiment*, Phys. Rev. A **101**, 032307 (2020).
- [6] B. Zygelman, *A First Introduction to Quantum Computing and Information*, University of Nevada, Springer (2018).
- [7] A. Elahi, *Computer Systems Digital Design, Fundamentals of Computer Architecture and Assembly Language*, Springer (2018).
- [8] C. Bauckhage, et al. *Quantum Machine Learning: State of the Art and Future Directions*, Federal Office for Information Security (2022).
- [9] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*, University of Borås, Elsevier (2014).
- [10] T. Hastie, R. Tibshirani, & J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer (2009).
- [11] G. James, D. Witten, T. Hastie, & R. Tibshirani, *An Introduction to Statistical Learning*, Springer (2013).
- [12] N. Balabanian & T. A. Bickart, *Linear Network Theory: Analysis, Properties, Design and Synthesis*, Weber Systems (1982).
- [13] H. Carlin, et al. *Network Theory: An Introduction to Reciprocal and Nonreciprocal Circuits. Prentice-Hall series in electrical engineering*, Prentice-Hall (1964).
- [14] A. W. Harrow, A. Hassidim & S. Lloyd, *Quantum m Algorithm for Linear Systems of Equations*, Phys. Rev. Lett. **103**, 150502 (2009).