

Métodos de Codificación para QML

DOI: 10.29236/sistemas.n173a7

Samuel Lopera Torres, Daniel Sierra Sosa, Juan G. Lalinde Pulido

1. Introducción

La fusión de la computación cuántica con el aprendizaje automático está allanando el camino para el Aprendizaje Automático Cuántico (QML, por sus siglas en inglés), un campo con el potencial de redefinir las capacidades computacionales en sectores como la criptografía, el descubrimiento de fármacos y las finanzas. En el núcleo de QML se encuentra la codificación de datos clásicos en estados cuánticos, una tarea tanto fundamental como compleja. El proceso de codificación

permite representar la información clásica aprovechando las propiedades únicas de los sistemas cuánticos, como la superposición y el entrelazamiento, lo que facilita la ejecución de tareas intensivas en computación con una rapidez y eficiencia potencialmente sin precedentes.

No obstante, los desafíos de la codificación de datos en QML son numerosos. La codificación clásica representa los datos en el sistema binario, lo que la hace simple en el sentido de que solo cambia los

símbolos utilizados para representar los números enteros, pero no cambia sus propiedades. Los otros tipos de datos, como números de punto flotante, texto, imágenes y sonidos se codifican a partir de los enteros mediante algoritmos definidos para interpretar y transformar esas secuencias binarias. La codificación cuántica representa los datos mediante un estado cuántico, por lo que debe considerar las características específicas del entorno cuántico, tales como la superposición, el entrelazamiento y, especialmente, el hecho de que un estado cuántico es un vector unitario en un espacio de Hilbert.

Adicionalmente, las limitaciones en el número de qubits, los tiempos de coherencia y la fidelidad de las compuertas en el hardware cuántico actual afectan la viabilidad práctica de las estrategias de codificación. Estos desafíos hacen que sea esencial desarrollar métodos de codificación que no solo sean teóricamente sólidos, sino también resistentes al ruido, eficientes en términos de hardware y escalables.

Este estudio es motivado tanto por el potencial teórico del QML como de sus aplicaciones prácticas. Algoritmos cuánticos, como el de Shor y de Grover, muestran el potencial la computación cuántica. Sin embargo, materializarlo exige una comprensión profunda de cómo los datos clásicos pueden codificarse eficientemente dentro de sistemas cuánticos. Por ello, este

trabajo se centra los métodos de codificación de datos.

2. Principios Matemáticos y Físicos

Para entender los principios de representación de datos clásicos como estados cuánticos, hay que comprender algunos de los conceptos básicos de la mecánica cuántica [1]. En primer lugar, es importante entender la *Notación de Dirac*, también conocida como *bra-ket*. Esta se utiliza para representar vectores en el espacio de Hilbert. Un estado cuántico es un vector columna y se representa mediante el *ket* $|\varphi\rangle = (a_1, a_2, \dots, a_n)^T$. El *bra* es un vector fila que es el transpuesto conjugado del *ket*, $\langle\varphi| = |\varphi\rangle^\dagger = (a_1^*, a_2^*, \dots, a_n^*)$. Adicionalmente, el producto interno entre $|\varphi\rangle$ y $|\psi\rangle$, en la notación de Dirac¹, es $\langle\varphi|\psi\rangle$, el producto externo es $|\varphi\rangle\langle\psi|$ y la norma de un vector es $\langle\varphi|\varphi\rangle$.

El equivalente cuántico al bit clásico, entendido como la unidad fundamental para representar la información, es el *qubit*. Matemáticamente hablando, un qubit es un vector unitario en el espacio de Hilbert \mathbb{C}^2 . Esto significa que se puede representar cualquier qubit genérico $|\varphi\rangle$ como un vector en \mathbb{C}^2 , de la siguiente manera:

$$|\varphi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

¹ La notación utilizada para el producto interno es un braket. La parte izquierda es el bra y la derecha el ket.

Todo espacio vectorial tiene infinitas bases. En la computación cuántica, cuando se tiene un qubit, se utiliza la base constituida por los estados $|0\rangle$ y $|1\rangle$ y se denomina la base computacional², donde:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ y } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Como cualquier vector en \mathbb{C}^2 , $|\varphi\rangle$ se puede expresar como una combinación lineal de los elementos de la base de dicho espacio de Hilbert. Dado que cualquier qubit $|\varphi\rangle$ puede expresarse como una combinación lineal de los elementos de la base:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Esto es lo que se conoce como el principio de superposición. Adicionalmente, se debe cumplir que $|\alpha|^2 + |\beta|^2 = 1$, ya que, como se verá a continuación, todo estado cuántico es un vector con norma 1. Finalmente, cuando se mide un qubit, su estado se altera y los únicos resultados posibles son $|0\rangle$ o $|1\rangle$. El resultado de la medición es aleatorio donde la probabilidad de que el resultado sea $|0\rangle$ ó $|1\rangle$ es $|\alpha|^2$ y $|\beta|^2$ respectivamente.

Cuando se tiene un sistema con múltiples qubits $|\varphi_s\rangle$, el sistema compuesto es el resultado del producto tensorial de cada uno de los estados.

$$|\varphi\rangle = |\varphi_0\rangle \otimes |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \dots \otimes |\varphi_{n-2}\rangle \otimes |\varphi_{n-1}\rangle$$

y la base del nuevo espacio es el producto tensorial de las bases de cada qubit y, por lo tanto, es una base ortonormal donde los elementos de la base se numeran³ de $|0\rangle$ a $|2^n - 1\rangle$. Esto implica que si son n qubits, el espacio es \mathbb{C}^{2^n} . En general, el estado cuántico de un sistema de n qubits es vector unitario en \mathbb{C}^{2^n} y se representa como

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

donde los $\alpha_i \in \mathbb{C}$ y deben cumplir que

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

Aplicar una operación a un sistema cuántico es equivalente a multiplicar una matriz unitaria⁴; estas matrices unitarias se llaman compuertas. En sistema de n qubits la matriz unitaria es una matriz cuadrada de $2^n \times 2^n$. Si se aplican operaciones

-
- 2 La base computacional tiene un sentido físico muy claro: El estado $|0\rangle$ corresponde al estado de reposo. Esto es importante porque uno de los aspectos misteriosos de la mecánica cuántica es que es imposible conocer un estado cuántico porque al medirlo se produce el llamado colapso de la función de onda y cambia de estado. Adicionalmente, cuando un qubit está en el estado $|0\rangle$ o en el estado $|1\rangle$ se comporta como un bit clásico. Una muy buena introducción a la teoría de la información cuántica se puede encontrar en [10]
 - 3 Dado que el espacio de Hilbert se construye mediante productos tensoriales, la representación binaria del número asociado con la base describe cuáles es el componente de cada qubit que aporta a ese estado.
 - 4 La matriz debe ser unitaria porque el resultado debe ser un estado cuántico válido y por lo tanto la matriz no puede alterar la norma del vector. Esto tiene dos implicaciones importantes: toda operación cuántica es invertible y, por lo tanto, no disipa calor.

que afectan a algunos qubits, de todas maneras la operación afecta a todo el sistema y la compuerta se puede modelar matemáticamente como el producto tensorial de aplicar la operación al subespacio generado por los qubits afectados y aplicar la identidad al subespacio generado por los otros qubits. Uno de los resultados más importantes es que existe un conjunto de compuertas básicas tales que cualquier compuerta cuántica puede ser representada como aplicación sucesiva, es decir multiplicación, de las compuertas básicas. Por eso se puede hablar de computación cuántica universal.

Cuando se trabaja con sistemas compuestos, representar puertas cuánticas como multiplicaciones de matrices puede resultar complicado. Para resolver ese problema, generalmente se usan circuitos cuánticos para representar estas operaciones. En esta representación, el tiempo transcurre en el algoritmo de izquierda a derecha. El estado inicial de cada qubit es casi

siempre $|0\rangle$, que es el estado de reposo. Las compuertas se representan por rectángulos donde el texto del rectángulo identifica la compuerta. Las líneas horizontales simples corresponden a qubits y las dobles a bits clásicos. Avanzar de izquierda a derecha implica multiplicar la matriz que representa el operador por el vector que representa el estado del sistema. En principio, toda operación debe afectar a todos los qubits. Si hay varias compuertas en la misma línea vertical, la compuerta a aplicar es el producto tensorial de estas. Si hay algún qubit al que no se le esté aplicando ninguna compuerta, se utiliza la matriz identidad para representar este hecho y se incluye en el producto tensorial.

La Figura 1 presenta un ejemplo de un circuito cuántico con dos qubits. El estado inicial es $|0\rangle \otimes |0\rangle$. Como la compuerta H se aplica únicamente al primer qubit, entonces la operación que realmente se aplica es $H \otimes I_{2 \times 2}$, donde $I_{2 \times 2}$ es la matriz identidad con dos filas y dos columnas.

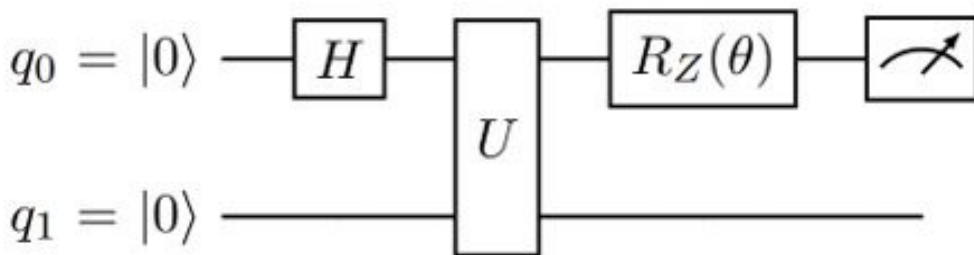


Figura 1 Ejemplo de un circuito cuántico

El estado después de haber aplicado $H \otimes I_{2 \times 2}$ es $(H \otimes I_{2 \times 2})(|0\rangle \otimes |0\rangle)$.

La Tabla 1 presenta las compuertas cuánticas más comunes y su representación en forma de matriz.

3. Algoritmos de Codificación

En el aprendizaje cuántico de máquinas (QML), la forma en que codi-

ficamos datos clásicos en estados cuánticos está relacionada con métodos de Kernel [2]. Formalmente, la codificación puede considerarse como un mapeo de características desde el espacio de entrada original X a un espacio de Hilbert \mathbb{C}^{2^n} . A diferencia de lo que pasa en la representación clásica en binario, este mapeo implica una transformación no trivial de los datos [3], [4]. La


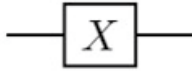
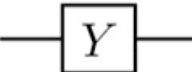

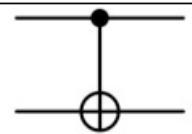
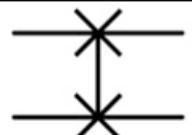
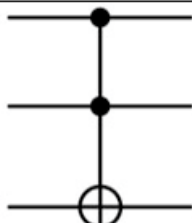
Compuerta	Representación en circuito	Representación en matriz
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Negación controlada		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Swap		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Tabla 1 Compuertas cuánticas más comunes

forma como se codifiquen va a influir no solo en qué modelos se pueden utilizar sino en también en su capacidad expresiva [5]. La codificación se lleva a cabo mediante un circuito cuántico parametrizado por los datos de entrada y, posiblemente, por otros parámetros. aplicados a un estado inicial que generalmente es $|0\rangle^{\otimes n}$. Formalmente, el mapa de características cuántico se define como una transformación:

$$\psi: X \rightarrow \mathbb{C}^{2^n}$$

donde n es el número de qubits necesarios para representar los datos y depende no solo del número de datos sino también de su naturaleza y su rango. La codificación, también conocida como preparación del estado, se realiza mediante una transformación unitaria parametrizada. En otras palabras, para codificar $x \in X$ como $|\psi(x)\rangle \in \mathbb{C}^{2^n}$, se utiliza una transformación paramétrica $U_\psi(x)$ tal que

$$|\psi(x)\rangle = U_\psi(x)|0\rangle^{\otimes n}$$

Todos los algoritmos en esta sección construyen un circuito cuántico desde cero, y como tal, no tienen la necesidad de instrucciones de retorno; ya que, en resumen, son una serie de operaciones que utilizan el vector de características $x \in X$ como parámetro. Se presentarán tres tipos de codificación básicos: en base, en amplitud y en fase. Estas son las formas de codifi-

cación naturales que surgen a partir de la estructura de un estado cuántico.

Antes de entrar en cada una de ellas, es importante presentar de dónde se derivan. El estado cuántico de un sistema de n qubits es vector unitario en \mathbb{C}^{2^n} se representa como

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

donde los α_i y deben cumplir que

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

En esta representación, los $|i\rangle$ son 2^n las bases del espacio y se pueden utilizar para representar datos en binario. Los $\alpha_i \in \mathbb{C}$, como son números complejos, tienen una magnitud y una fase. Hay que recordar que si $z = a + ib$ es un número complejo, puede ser escrito como $z = re^{i\theta}$ donde $r = \sqrt{a^2 + b^2}$ y $\theta = \tan^{-1}(b/a)$. Así α_i , puede ser utilizado para representar los datos mediante su magnitud r o su fase θ .

3.1 Codificación en Base

La codificación de base, como su nombre lo indica, tiene como objetivo codificar los datos realizando una superposición uniforme de todos los estados base involucrados en el conjunto de datos.

Por ejemplo, dado un conjunto de datos $D = [2, 5, 9]$ o, en binario: $D = [0010, 0101, 1001]$, se necesita la siguiente superposición:

$$|\psi\rangle = \frac{|0010\rangle + |0101\rangle + |1001\rangle}{\sqrt{3}}$$

En general, se busca construir el estado:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} I_D(i) |i\rangle$$

donde $I_D(i)$ es una función que devuelve 1 ó 0 dependiendo de la presencia del estado i en el conjunto de datos D . El número de qubits necesarios, n , está determinado por el rango del conjunto de datos, dado que un sistema cuántico de n qubits tiene 2^n estados base distintos. Se puede observar también que, dado que este esquema de codificación ignora la multiplicidad de los datos pues todos los datos iguales se representan con el mismo estado.

La manera más sencilla de implementar este tipo de codificación es utilizando el algoritmo de Grover utilizando $I_D(i)$ como función oráculo. Para el algoritmo de Grover se construye una compuerta unitaria que cumpla que

$$U_{I_D} |x\rangle = (-1)^{I_D(x)} |x\rangle$$

y luego de \sqrt{N} iteraciones se obtiene una muy buena aproximación

del estado deseado. Este tipo de codificación es fácil de entender pues, si bien utiliza la superposición cuántica, representa los datos clásicos mediante el sistema binario [6]

3.2 Codificación en Amplitud

La codificación en amplitud busca codificar los datos en el vector de estado haciendo cada dato sea codificado en la amplitud un estado diferente. En este caso, se Schuld [1] propone un algoritmo para codificar datos en tiempo lineal $O(n)$, que consiste en usar varias rotaciones controladas de los qubits en el sistema compuesto. En otras palabras, si $x = \{x_0, x_1, \dots, x_{N-1}\} \in X$ y se va a representar mediante el estado cuántico

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

entonces el objetivo es que $\alpha_i = x_i$. Como $|\varphi\rangle$ es un estado cuántico, debe cumplir la condición de normalidad $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$ así que x debe ser normalizando antes de poder representarlo. La forma de construir los α_i es mediante rotaciones aplicadas a cada qubit.

Sin embargo, antes de entender el algoritmo detrás de la codificación de amplitud, es necesario comprender cómo rotar un qubit para lograr un resultado deseado. Una forma es utilizar una compuerta que rota el qubit sobre un eje

específico. Una compuerta⁵ utilizada para es fin es la compuerta $R_y(\theta)$, que se define como

$$R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Como el espacio \mathbb{C}^{2^n} es el resultado del producto tensorial de los espacios asociados a cada qubit, entonces en cada uno de los vectores de la base computacional de \mathbb{C}^{2^n} están representados todos los qubits simultáneamente. Por ejemplo, si $n = 4$ se tiene el espacio $\mathbb{C}^{2^4} = \mathbb{C}^{16}$. La base computacional de este espacio tiene 16 vectores. Como se representa cada elemento de la base en binario, el noveno (9°) vector de la base sería $|1001\rangle$, lo que quiere decir que este vector corresponde al estado en el cuál es primer y el último qubit tienen estado 1 y los dos del medio tiene estado 0.

Con el fin de simplificar la explicación del algoritmo, se asumirá que el número de datos a codificar $N = 2^n$. Si esta situación no se da, se completa la colección de datos con ceros (0) hasta que el número esa una potencia de dos (2). El algoritmo procede de la siguiente manera:

1. Se determina cuál es la rotación que debe aplicarse al primer qubit. Si se tienen $N = 2^n$ datos, entonces el espacio es \mathbb{C}^{2^n} , y la primera mitad de los estados tienen el primer qubit en cero (0)

mientras que la otra mitad lo tiene en uno (1). Como se cumple la condición de normalidad, se suma el cuadrado de las amplitudes de los primeros estados y con base en este valor se determina cuál es la rotación que se debe aplicar al primer qubit para que

$$P(0) = \sum_{i=0}^{(2^n-1)/2} |x_i|^2$$

donde $P(0)$ es la probabilidad de que al medir el primer qubit dé $|0\rangle$. El qubit se rota aplicando la compuerta $R_y(\theta)$.

2. Se determina cuál es la rotación que debe aplicarse al segundo qubit. Se divide el análisis en dos casos. Los estados para los cuales el primer qubit es cero (0) y los estados para los cuales el primer qubit es uno (1). Para cada caso se consideran solamente los estados que cumplen con la condición, o sea que para cada caso se analiza la mitad de los estados.
 - a. Para los casos en que el primer qubit es cero (0), se calcula $P(00)$, la probabilidad de que los dos primeros qubits sean cero (0) teniendo presente que ya se conoce la probabilidad $P(0)$ de que el primer qubit sea cero (0).

⁵ La compuerta Pauli Y es un caso particular de la compuerta $R_y(\theta)$ donde $\theta = \pi$.

En este caso

$$P(00) = \frac{\sum_{i=0}^{(2^n-1)/4} |x_{2i}|^2}{P(0)}$$

Con base en $P(00)$ se calcula la rotación que hay que aplicar al segundo qubit cuando el primero es cero (0), y se implementa con una compuerta $R_y(\theta)$ controlada⁶ por el primer qubit.

- b. Para los casos en que el primer qubit es cero (1), se calcula $P(10)$, la probabilidad de que el primer qubit sea uno (1) y el segundo qubit sea cero (0) teniendo presente que ya se conoce la probabilidad $P(0)$ de que el primer qubit sea cero (0). En este caso

$$P(10) = \frac{\sum_{i=0}^{(2^n-1)/4} |x_{2i+1}|^2}{P(1)}$$

Con base en $P(10)$ se calcula la rotación θ que hay que aplicar al segundo qubit cuando el primero es uno (1), y se implementa con una compuerta $R_y(\theta)$ controlada⁷ por el primer qubit.

3. En general, se debe determinar para cada qubit cuál es la rotación que se debe aplicar. Para calcular la rotación que debe aplicarse al q -ésimo qubit, se 2^{q-1} casos. sigue el mismo proceso de los pasos anteriores. Cada caso representa un estado con $q-1$ qubits. Sea $0 \leq t < 2^{q-1}$,

caso asociado con t calcula cuál es la probabilidad de que el estado de q bits $2t$ tenga el q -ésimo qubit en cero (0). En general

$$P(2t) = \frac{\sum_{i=0}^{(2^n-1)/2^q} |x_{2^{q-1}i+t}|^2}{P(t)}$$

Con base en $P(2t)$ se calcula la rotación θ que hay que aplicar al q -ésimo qubit cuando los $q-1$ primeros qubits son t , y se implementa con una compuerta $R_y(\theta)$ controlada por los primeros $q-1$ qubits.

Ahora bien, el método presentado no es el único, pero si presenta claramente cómo se puede codificar en amplitud. Existen más propuestas de métodos de codificación en amplitud, como por ejemplo [2] que plantea una estrategia “dividir y conquistar”. Otras propuestas incluyen la aplicación de la cascada, realizándola en el orden inverso, como se propone en [7], o utilizar una cascada de compuertas R_z previas a la cascada R_y .

Esta codificación es bastante útil para representar problemas físicos. [8] y en codificación de datos para aprendizaje de máquina cuántico, lo que lleva a que la investigación en este campo sea muy activa.

6 Una compuerta controlada realiza la operación específica solo si el qubit controlador está en un estado específico.

7 Una compuerta controlada realiza la operación específica solo si el qubit controlador está en un estado específico.

3.3 Codificación en Fase (Angulo)

Finalmente, la codificación en fase codifica los valores de la entrada en cada uno de los ángulos de cada qubit, resultando en un estado no entrelazado (separable). El algoritmo propuesto por Weigold [34], recibe un vector de entrada $X = (x_0, x_1, \dots, x_n)^T$ y produce un estado cuántico donde cada x_i de la entrada está representado en el ángulo del i -ésimo qubit.

Es importante tener en cuenta que $R(X_i)$ es una rotación arbitraria, lo que significa que se puede utilizar, por simplicidad, cualquiera de las puertas R_X , R_Y ó R_Z . Lo importante es que todas las rotaciones sean del mismo tipo. En su mayoría, la literatura y artículos utilizan la puerta R_Y , ya que generalmente, el eje X se suele utilizar para negaciones, y el eje Z para la medición. No obstante, en caso de utilizar este método de codificación, la rotación más apropiada depende única y exclusivamente del problema a trabajar.

La construcción del estado cuántico correspondiente a esta forma de codificar es

$$|\varphi\rangle = R_Y(x_0)|0\rangle \otimes R_Y(x_1)|0\rangle \otimes \dots \otimes R_Y(x_n)|0\rangle$$

Este no suele ser utilizado generalmente para aspectos de ciencia de datos, es más orientado (debido a su alta complejidad en memoria),

sino para algoritmos de optimización y búsqueda, como el algoritmo de Shor. [9]

4. Conclusión

La exploración y el desarrollo de métodos de codificación cuántica son fundamentales para el procesamiento de información y la comunicación cuánticas. Esta revisión ha examinado tres estrategias de codificación cuántica presentando sus fundamentos teóricos, implementaciones prácticas y los desafíos que enfrentan.

Los métodos que se presentaron tienen una sólida fundamentación teórica, pero su implementación en entornos reales enfrenta obstáculos adicionales como la decoherencia de qubits, el ruido cuántico y la necesidad de operaciones cuánticas de alta fidelidad. La investigación futura debería centrarse en superar estos desafíos mediante el desarrollo de algoritmos de corrección de errores cuánticos más robustos y escalables, mejorando la fidelidad de las operaciones cuánticas y optimizando la integración de sistemas cuánticos con tecnologías clásicas. Asimismo, explorar materiales y arquitecturas cuánticas novedosas, como qubits topológicos y sistemas cuánticos fotónicos, podría proporcionar nuevas vías para una codificación cuántica eficiente y resiliente.

En resumen, los métodos de codificación cuántica están a la vanguar-

dia del desarrollo de tecnologías cuánticas, pero continuar su desarrollo es un requisito para explotar el potencial transformador de la computación cuántica. La investigación interdisciplinaria y la colaboración continua serán esenciales para desbloquear el potencial completo de la codificación cuántica, lo que finalmente llevará a la realización de sistemas prácticos y poderosos de información cuántica.

Referencias

- 1 M. Schuld y F. Petruccione, *Supervised Learning with Quantum Computers*, 1st ed., Springer, 2018, p. 287.
- 2 M. Schuld, «Supervised quantum machine learning models are kernel methods,» *arXiv preprint arXiv:2101.11020*, 2021.
- 3 M. Schuld y N. Killoran, «Quantum machine learning in feature hilbert spaces,» *Physical review letters*, vol. 122, p. 040504, 2019.
- 4 S. Lloyd, M. Schuld, A. Ijaz, J. Izaac y N. Killoran, «Quantum embeddings for machine learning,» *arXiv preprint arXiv:2001.03622*, 2020.
- 5 M. Schuld, R. Sweke y J. J. Meyer, «Effect of data encoding on the expressive power of variational quantum-machine-learning models,» *Physical Review A*, vol. 103, p. 032430, 2021.
- 6 C. Wetterich, «Entanglement in quantum mechanics,» *Physical Review A*, vol. 99, p. 022112, 2019.
- 7 M. Mangin-Brinet, J. Zhang, D. Lacroix y E. A. Ruiz Guzman, «Efficient solution of the non-unitary time-dependent Schrodinger equation on a quantum computer with complex absorbing potential,» *Quantum*, vol. 8, p. 1311, April 2024.
- 8 W. J. Munro y others, «Weak measurements with superconducting qubits,» *Physical Review Letters*, vol. 86, p. 4992–4995, 2001.
- 9 D. B. Szombati y others, «Josephson ϕ 0-junction in nanowire quantum dots,» *Nature Physics*, vol. 16, p. 568–573, 2020.
- 10 M. A. Nielsen y I. L. Chuang, *Quantum Computation and Quantum Information*, 10th anniv ed., Cambridge University Press, 2010, p. 676. 