

Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores unidad Juriquilla

Identificación de Fuentes de Luz usando aprendizaje automático cuántico

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Licenciado en Tecnología

 $P \quad R \quad E \quad S \quad E \quad N \quad T \quad A \quad :$

José Diego Guerrero Morales

TUTORES

Dr. Mario Alan Quiroz Juárez Dr. Roberto de Jesús León Montiel







Universidad Nacional Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor. Para mi carnalote, con quien he enfrentado este mundo desde hace 23 años.

Agradecimientos

En toda mi vida nunca me ha faltado nada; estoy consciente de que la razón es el sacrificio de mis padres, Beatriz y José. Este trabajo no hubiera sido posible sin su apoyo. Sé que en su vida hubo diferentes carencias, sin embargo, se aseguraron de que no nos faltara nada a mi hermano y a mí. Mamá, gracias por enseñarme a poner el corazón en todo lo que hago, que cada momento es único y que cada aspecto de la vida es bello. Papá, gracias a ti aprendí que todo en la vida tiene un costo justo, que cada decisión solo sirve para tomar la siguiente y que cada persona ama a su manera.

A mi hermano Rodrigo, gracias por acompañarme todos estos años a darle sentido a esta vida y al cómo vivirla. Hemos tenido tantas aventuras juntos que, en algunos de mis recuerdos, no me es claro quién soy yo. Si nuestros caminos se separan, sé que no debemos parar nuestra búsqueda, ya que el final de esta será solo para llegar al lugar donde empezamos y conocerlo por primera vez. Por esto mismo, sé que si recorremos el mundo en direcciones opuestas, como el monstruo sin nombre, tarde o temprano regresaré a mi otra mitad.

Séneca, en sus cartas a Lucilio, habla de cómo a quien se le llama amigo se debe de confiar como uno confía en sí mismo, pues ya lo has juzgado antes de llamarlo tal. En esta proceso he estado rodeado de personas maravillosas que me han mostrado a qué se refería Séneca. Máximo, mi amistad más larga y constante, juntos hemos aprendido que las equivocaciones son parte de lo que somos; hemos pasado por diferentes temporadas, pero nuestra amistad sigue vigente. Gabo, a quien he considerado mi complemento, gracias por enseñarme que el cambio nunca es cómodo y que debe realizarse todos los días. Martín, te he visto sufrir incontables veces por un bien mayor; gracias a tu ejemplo sé que es posible cambiar al mundo mediante esfuerzo y cooperación. Iván, quiero agradecerte por mostrarme lo trivial que resulta la vida cuando tienes confianza en ti mismo.

Las asignaturas de la licenciatura se caracterizaron por su carácter introductorio; sin embargo, la clase del Dr. Miguel fue una de las pocas excepciones. Doctor, gracias por introducirme al maravilloso mundo de la mecánica cuántica y presentarme a sus extraordinarios fundadores. Nuestras discusiones, dentro y fuera del salón de clases, forjaron al Tecnólogo que soy ahora. En nuestras últimas clases mencionó que su único objetivo era que nosotros lo superásemos; quiero dejar aquí escrito que pondré todo de mi parte para que así sea. Agradezco a la Dra. Consuelo por siempre confiar en mis capacidades y alentarme a soñar en grande; al Dr. Iván, por no ceder conmigo y ayudarme a encontrar mi mejor versión; a su vez, a mis profesores y amigos de la Facultad de Ciencias CU, quienes me recibieron con los brazos abiertos, en especial a Jesús y Eduardo.

No puedo terminar sin agradecer a mis asesores, Dr. Mario Alan y Dr. Roberto, gracias por su invaluable guía y, sobre todo, su paciencia durante el desarrollo de este trabajo. Sé que ha sido un trabajo largo; de verdad aprecio todo el tiempo que me han dedicado.

Para terminar, quisiera agradecer a una persona que me acompañó durante varios años y que, a pesar de haber tomado caminos diferentes, me es imposible no mencionarla. Lizbeth, gracias por tu apoyo incondicional durante el tiempo que coincidimos; siempre tendré presente tus enseñanzas y tu calidez como persona.

Agradecimientos a programas e instituciones

Agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnologías por el apoyo otorgado a través del proyecto CF-2023-I-1496, así como a la Dirección General de Asuntos del Personal Académico (DGAPA) de la UNAM por el respaldo brindado mediante el Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica, en el marco del proyecto IA103325.

Agradezco al Centro de Física Aplicada y Tecnología Avanzada, así como al Instituto de Ciencias Nucleares, por asignarme un lugar de trabajo y recursos para la realización de esta tesis. A su vez a la empresa IBM, por aportar créditos equivalentes a un total de 12 horas de tiempo de cómputo en sus computadoras cuánticas.

Índice general

Agradecimientos		5	
Resumen			13
Introd	ntroducción		
1. Justificación		17	
1.1.	Métod	los clásicos y nuevas alternativas	17
1.2.	Comp	utación cuántica	18
1.3.	Identi	ficación de fuentes coherentes: Un problema cuántico	18
1.4.	Impac	to y perspectivas	18
2. Hip	ótesis j	y objetivos	21
2.1.	Hipót	\hat{c} esistence of the second secon	21
2.2.	Objet	ivo	21
2.3.	Objet	ivos específicos	21
3. Mar	·co teó	rico	23
3.1.	Nocio	nes básicas de mecánica cuántica	23
	3.1.1.	Superposición	23
	3.1.2.	Operadores	24
	3.1.3.	Operador de densidad	25
3.2.	Cuantización del campo electromagnético		26
	3.2.1.	Segunda cuantización del campo electromagnético	27
	3.2.2.	Estados de número	29
	3.2.3.	Estados coherentes	30
	3.2.4.	Estados termales	32
3.3.	Bases	del cómputo cuántico \hdots	34
	3.3.1.	Qubits	35
	3.3.2.	Transformaciones unitarias	35
	3.3.3.	Compuertas control	37
	3.3.4.	Circuitos cuánticos	38
	3.3.5.	Obtención e interpretación de resultados	38
	3.3.6.	Estados cuánticos equiprobables	39
	3.3.7.	Estados de Hipergráficas	40

	3.3.8. Simulación de circuitos cuánticos	41	
3.4.	Aprendizaje automático en cómputo clásico y cuántico	45	
	3.4.1. La neurona artificial	45	
	3.4.2. Perceptrón cuántico	45	
	3.4.3. Algoritmos genéticos	49	
4. Dise	ño, implemetación y evaluación de algoritmos	53	
4.1.	Estructura general	53	
4.2.	Obtención de los datos	53	
4.3.	Representación de los datos en estados cuánticos	53	
4.4.	Preparación de los circuitos cuánticos	57	
4.5.	Entrenamiento y evaluación	57	
	4.5.1. Generación de la población inicial	59	
	4.5.2. Selección	59	
	4.5.3. Selección de padres y mutantes	59	
	4.5.4. cruce y reemplazo	61	
	4.5.5. Evaluación	61	
4.6.	Ejecución en una QPU real	63	
5. Rest	iltados	65	
5.1.	Separabilidad de los datos	65	
	5.1.1. Perceptrón de dos qubits	67	
	5.1.2. Perceptrón de seis qubits	72	
5.2.	Generación de los circuitos cuánticos	73	
5.3.	Entrenamiento del perceptrón	74	
5.4.	Evaluación del perceptrón	75	
	5.4.1. Evaluación del entrenamiento exhaustivo	77	
	5.4.2. Evaluación del entrenamiento ligero	79	
6. Disc	usión	87	
U. Disc		0.	
7. Con	clusiones	89	
Bibliog	Bibliografía		
8 Bibli	8 Bibliografía		
6.D 1011	o.Dibilografia		

Índice de figuras

3.1.	Distribuciones de probabilidad de luz coherente y termal	34
3.2.	Esfera de Bloch	36
3.3.	Representación de compuertas cuánticas comunes en circuitos cuánticos	38
3.4.	Circuito de una compuerta multicontrol Z y multicontrol X, a) y b) son representaciones comúnes de la compuerta control Z y c) es la	
	representación de la compuerta control X	38
3.5.	Un circuito cuántico con una medición del un qubit anciliar, este qubit está enredado con a_0 y a_1 por medio de una compuerta multi control X	39
3.6	Representación de un estado cuántico de gráfica con tres vértices y tres	00
0.0.	aristas	40
3.7.	Representación de un estado cuántico de hipergráfica 3-uniforme con	10
	tres hiperaristas.	40
3.8.	Representación de un estado cuántico de hipergráfica general con hi- peraristas de diferente orden.	41
42fig	gure.caption.21	
3.10	. Perceptrón en su más simple realización	45
3.11	. Una función de activación sigmoide: $\sigma(x) = \frac{1}{1+e^{-x}}$	46
3.12	. Representación gráfica de un perceptrón cuántico (Tacchino, Macchia- vello, Gerace, y Bajoni, 2019)	48
3.13	. Circuito del perceptrón cuántico, esencialmente de dos transformacio- nes unitarias U_i , U_w , de una compuerta MCX y una medición en un	
	qubit anciliar	48
3.14 3.15	. Un cromosoma, gen y alelo en una representación binaria	50
	de multipunto, d) cruce uniforme	52
4.16	. Proceso general. Todo el proceso es híbrido, el tipo de cómputo se detona con distintos colores, dependiendo si fue totalmente clásico, simulado en qiskit, simulado con ruido o ejecutado en un procesador	
	cuántico real.	54
4.17	. Preparación de datos clásicos	56
4.18	. Flujo de trabajo para entrenamiento de un algoritmo genético	58
4.19	. Funcional overlap entre dos distribuciones gaussianas, el overlap es la	
	zona sombreada de color amarillo.	60

4.20. Caso óptimo, dos distribuciones con un overlap nulo y un umbral entre las distribuciones
4.21. Proceso cruzado de entrenamiento y evaluación del algoritmo genético
5.22. Espacio de características para los datos de luz termal y coherente reescalados a resolución de 8 bits
5.23. Histograma de todos los vectores binarios en su representación decimal, se puede observar que las dos clases se separan. Se utilizaron datos reescalados a 8 bits, con un número promedio de fotónes igal a 0.77 y 160 datapoints.
5.24. Histogramas de las proyecciones de vectores binarios, en el lado iz- quierdo cada clase se proyecta con su vector representativo y en el lado derecho las proyecciones se hacen con el vector de la otra clase. Los vec- tores representativos, en base decimal son $w_{coh} = 35000895515131904$
$y w_{th} = 40004547942994000$
derecho las proyecciones se hacen con el vector de la otra clase
5.26. Perceptrón cuántico de dos qubits evaluado clásicamente
5.27. Perceptrón cuántico de dos qubits evaluado en una simulación ideal.
5.28. Discrepancia entre los perceptrones clásico y cuánico simulado ideal-
mente.
5.29. Perceptrón de dos qubits simulado con ruido
5.30. Discrepancia de un perceptrón cuántico simulado con ruido y uno im-
plementado clásicamente
5.31. Error medio entre las proyecciones de datos coherentes, ejecutadas en
5.22 Error modio entre les provessiones de detes termoles, ciecutades en
una simulación realista (con ruido), con respecto a una ejecución clásica
5.33 Preparación del circuito cuántico
5.33. Curva de aprendizaje del algoritmo genético
5.35 Provecciones de los datos de entrenamiento con el vector de pesos en-
contrado
5.36 Resultados de entrenamiento para la configuración ligera, en el lado
izquierdo es el entrenamiento clásico y el derecho el simulado idealmente
5.37. Prueba de error tipo I v II para el entrenamiento clásico en configura-
ción exhaustiva
5.38. Provecciones clásicas y simuladas idealmente con el entrenamiento exhaus-
tivo
5.39. Proyecciones simuladas con ruido en el set de evaluación del entrena-
miento exhaustivo
5.40. Eficiencias del perceptrón entrenado clásicamente, evaluado en una
computadora clásica (naranja), cuántica simulada idealmente (azul),
cuántica simulada con ruido (azul oscuro)

5.41.	Eficiencias del entrenamiento largo usando simulación ideal, cada barra	
	representa la eficiencia del perceptrón evaluado en cómputo clásico	
	(naranja), cuántico simulado (azul) y simulado con ruido (azul oscuro).	80
5.42.	Eficiencias del entrenamiento ligero entrenado en una computadora	
	cuántica simulada con ruido	81
5.43.	Proyecciones simuladas con ruido del set de evaluación	81
5.44.	Proyecciones clásicas y simuladas de los datos de entrada con el vector	
	obtenido del entrenamiento clásico	82
5.45.	Proyecciones simuladas con ruido de los datos de entrada con el vector	
	de pesos entrenado clásicamente	82
5.46.	Eficiencias en los conjuntos de evaluación de todos los entrenamientos	
	para la configuración ligera. Cada color indica en que tipo de cómputo	
	se realizó la evaluación. El eje horizontal denota en que tipo de cómputo	
	se realizó el entrenamiento.	83
5.47.	Eficiencias de todos los entrenamientos para la configuración exhaus-	
	tiva. Cada color indica en que tipo de cómputo se realizó la evaluación.	
	El eje horizontal denota en que tipo de cómputo se realizó el entrena-	0.4
	miento.	84

Resumen

La identificación de fuentes de luz es fundamental para el desarrollo de múltiples tecnologías fotónicas. Tradicionalmente, esta tarea ha dependido de métodos estadísticos que requieren una cantidad impráctica de observaciones. Sin embargo, en años recientes se ha demostrado que el uso de aprendizaje automático (machine learning) permite distinguir fuentes de luz hasta diez mil veces más rápido en comparación con los métodos convencionales. Este avance es prometedor y abre la puerta al desarrollo de nuevas tecnologías fotónicas. No obstante, existe otro paradigma computacional que ha generado gran interés en las últimas décadas: la computación cuántica. Este campo emergente aprovecha los principios de la mecánica cuántica para superar las limitaciones de las computadoras clásicas. En este contexto, el problema de la identificación de fuentes de luz tiene una naturaleza intrínsecamente cuántica, lo que hace que la computación cuántica sea una herramienta idónea para procesar este tipo de datos. En el presente trabajo de tesis, se diseñó e implementó un perceptrón cuántico para clasificar fuentes de luz coherentes y térmicas. Este modelo de aprendizaje automático fue entrenado y evaluado en diferentes esquemas de cómputo, tanto clásicos como cuánticos. Los resultados muestran que la versión cuántica presenta una ligera ventaja en comparación con su contraparte clásica. Es importante mencionar que la empresa IBM concedió créditos de cómputo al proyecto de la tesis presente, bajo el nombre de proyecto: "Identification of light sources using quantum machine learning" con el número 2803709, se proporcionó el equivalente a un total de 12 horas de cómputo en sus computadoras cuánticas.

Introducción

La identificación de fuentes de luz ha sido una problemática fundamental en la óptica cuántica y clásica, debido a sus aplicaciones en áreas como la comunicación cuántica, la metrología y los sistemas de sensado avanzado (Huttner, Imoto, Gisin, y Mor, 1995). En particular, diferenciar entre luz coherente y termal es esencial para garantizar la precisión en tecnologías como los sistemas de detección y alcance de luz (LIDAR, por sus siglas en inglés), si bien, para los LIDAR comerciales, la detección de coherencia cuántica no es necesaria, en años recientes se ha presentado el concepto de LIDAR cuántico, el cual ofrece ventajas sobre su contraparte clásica (Slepyan, Vlasenko, Mogilevtsev, y Boag, 2022), sin embargo, resulta necesaria una técnica efectiva para detectar este tipo de coherencia. A su vez, los protocolos de criptografía e imagen cuántica, utilizan este concepto (Howard y cols., 2019; Liu y Shih, 2009). Sin embargo, las técnicas tradicionales, basadas en métodos clásicos de análisis espectral, enfrentan limitaciones cuando se aplican a entornos ruidosos o altamente complejos (Zhang y cols., 2019).

En los últimos años, el desarrollo del cómputo cuántico ha abierto nuevas posibilidades para abordar problemas computacionalmente intensivos (Sun y cols., 2025; Dubey, Gantala, Ray, Prabhakar, y Rajagopal, 2025; Sharma y cols., 2021). La capacidad de los algoritmos cuánticos para procesar grandes cantidades de datos espectrales y modelar interacciones cuánticas ha demostrado ser una herramienta prometedora en este contexto (Bauer, Bravyi, Motta, y Chan, 2020; Bhatia y Ramkumar, 2020). Sin embargo, hasta ahora, no se ha explorado ampliamente su potencial para la identificación de fuentes de luz.

En este trabajo, se propone un método basado en aprendizaje automático cuántico para la identificación de fuentes de luz; se propone el entrenamiento híbrido de un perceptrón cuántico, capaz de diferenciar fuentes de luz. Para esto, se expone un esquema de codificación de datos clásicos a estados cuánticos y se verifica su separabilidad. Además, se explora la equivalencia de la implementación del perceptrón en los esquemas de cómputo clásico y cuántico. Este documento se organiza en siete capítulos. En los capítulos 1 y 2 describen la justificación de la investigación, así como sus objetivos e hipótesis. El capítulo 3 presenta el marco teórico. Por su parte, el capítulo 4 detalla los algoritmos y el tipo de entrenamiento utilizado. Los últimos capítulos, 5 , 6 y 7 abarcan los resultados obtenidos, su discusión y las conclusiones.

1. Justificación

La coherencia es un recurso físico fundamental que permite caracterizar las propiedades cuánticas de un sistema. Este concepto es central en muchas áreas de la física y tiene aplicaciones prácticas de gran relevancia (Streltsov, Adesso, y Plenio, 2017). En el caso de la luz, la coherencia nos permite diferenciar entre una fuente coherente, como un láser, y una fuente termal, como la luz ambiental. Esta distinción es esencial para el desarrollo de tecnologías ópticas avanzadas. Por ejemplo, sistemas como la localización por LiDAR, técnicas de tomografía, interferometría, y prácticamente cualquier tecnología que utilice luz para extraer información sobre un objeto físico, en esencia, aquellas tecnologías que dependen de enviar un haz de luz al objeto y recibir su señal de retorno.

El principal desafío radica en la presencia de numerosas fuentes de luz en el entorno, lo que dificulta la identificación de la luz enviada inicialmente (Zhang y cols., 2019). Por esta razón, es crucial contar con métodos robustos para distinguir y caracterizar la coherencia de las señales ópticas en escenarios con ruido ambiental elevado.

1.1. Métodos clásicos y nuevas alternativas

Existen técnicas clásicas basadas en las propiedades estadísticas de la luz para determinar su grado de coherencia (Glauber, 1963). Sin embargo, estas metodologías suelen ser poco prácticas debido a la gran cantidad de mediciones necesarias y al procesamiento analítico complejo y lento que requieren (Hloušek, Dudka, Straka, y Ježek, 2019). Este desafío ha motivado la exploración de enfoques alternativos, entre ellos, el uso de aprendizaje automático (machine learning) (Zambra y cols., 2005; Kudyshev y cols., 2020).

Recientemente, se han desarrollado modelos de aprendizaje automático para abordar el problema de la identificación y caracterización de la coherencia de la luz. Estos enfoques han demostrado ser prometedores, ofreciendo un procesamiento más eficiente y resultados que sugieren su potencial como una solución práctica para superar las limitaciones de los métodos tradicionales (You y cols., 2020).

1.2. Computación cuántica

Por otro lado, la computación cuántica se perfila como la próxima gran revolución en el cómputo, con la promesa de superar significativamente las capacidades del cómputo clásico. Aunque aún no se ha logrado una demostración concluyente de la llamada "supremacía cuántica", existen casos aislados donde el cómputo cuántico ha mostrado ventajas claras sobre su contraparte clásica (AbuGhanem y Eleuch, 2024). Actualmente, la investigación en este campo se centra en identificar problemas específicos donde los algoritmos cuánticos puedan ofrecer un desempeño superior.

Dado que la computación cuántica se basa en las leyes de la mecánica cuántica para procesar información, su potencial es especialmente relevante en la simulación de sistemas cuánticos. Como propuso Richard Feynman en 1982 (Feynman, 1982), este enfoque parecería ideal para abordar problemas intrínsecamente cuánticos, como las dinámicas cuánticas y la química cuántica.

1.3. Identificación de fuentes coherentes: Un problema cuántico

La identificación de fuentes coherentes de luz es, en esencia, un problema cuántico. Este reto involucra directamente las propiedades fundamentales de la luz, como su coherencia y entrelazamiento, lo que lo convierte en un candidato natural para ser abordado mediante computación cuántica.

Recientemente, se han propuesto metodologías que utilizan algoritmos cuánticos para simular sistemas físicos complejos, y los resultados obtenidos han sido prometedores (Wiebe, Berry, Høyer, y Sanders, 2011; Ollitrault, Miessen, y Tavernelli, 2021). Aplicar computación cuántica a la identificación de fuentes coherentes no solo permitiría resolver este problema de manera más eficiente, sino que también abriría nuevas oportunidades para la aplicación de esta tecnología en problemas reales y concretos.

1.4. Impacto y perspectivas

Resolver este problema tendría un impacto significativo en varios ámbitos. Por un lado, impulsaría el desarrollo y la adopción de la computación cuántica en la industria fotónica, mejorando tecnologías ópticas avanzadas como LiDAR y comunicaciones cuánticas (Khan, Elser, Dirmeier, Marquardt, y Leuchs, 2017; Gallego Torromé y Barzanjeh, 2024). Por otro lado, representaría una herramienta novedosa para el estudio de las propiedades cuánticas de la luz, facilitando el diseño de nuevas tecnologías basadas en estas propiedades fundamentales. Además, el éxito en este tipo de aplicaciones acercaría a la computación cuántica a escenarios industriales y prácticos, demostrando su valor en problemas aterrizados y fortaleciendo su papel en la siguiente generación de herramientas tecnológicas.

2. Hipótesis y objetivos

2.1. Hipótesis

La tarea de identificación de fuentes de luz coherente y termal, puede ser resuelta por medio de un perceptrón implementado por medio de algoritmos cuánticos en un procesador cuántico.

2.2. Objetivo

Desarrollar un algoritmo de aprendizaje automático implementado en un procesador cuántico real para discriminar fuentes de luz coherentes y termales.

2.3. Objetivos específicos

- 1. Revisar bibliografía relacionada con la teoría de cómputo cuántico.
- 2. Desarrollar un modelo computacional que simule un perceptrón cuántico en una computadora clásica para un problema multiclase.
- 3. Implementar un perceptrón cuántico en un procesador cuántico real utilizando la teoría de hipergrafos.
- 4. Entrenar un perceptrón cuántico para el problema de identificación binaria de fuentes de luz.
- 5. Realizar pruebas de funcionamiento y depuración del perceptrón en un entorno de simulación del procesador cuántico.

3. Marco teórico

3.1. Nociones básicas de mecánica cuántica

3.1.1. Superposición

La mecánica clásica y la mecánica cuántica comparten algunas similitudes a nivel superficial. Sin embargo, difieren de manera fundamental, como señala Paul M. Dirac:

"En el corazón de la mecánica cuántica yace el principio de superposición" (Dirac, 1931).

La formulación de la mecánica cuántica depende esencialmente del principio de superposición, mientras que en la mecánica clásica es inusual encontrar sistemas que lo sigan. Esto se debe a que la mecánica clásica se rige por la segunda ley de Newton (ver Ecuación 3.1), mientras que la mecánica cuántica se fundamenta en la ecuación de Schrödinger (ver Ecuación 3.2), la cual es una ecuación de onda y, por lo tanto, lineal. Dado que es una ecuación de onda, se asume que los objetos que describe son simultáneamente ondas y partículas. Este comportamiento dual ha sido demostrado experimentalmente en diversas ocasiones, como en el experimento de difracción de electrones, donde, a pesar de ser partículas, los electrones son difractados como si fueran ondas (Thomson y Reid, 1927; Eibenberger, Gerlich, Arndt, Mayor, y Tüxen, 2013).

La validez de la función de onda como una representación fiel de los objetos cuánticos ha sido, y sigue siendo, un tema de debate entre los físicos (Norsen, 2017).

$$\frac{d\mathbf{p}(\mathbf{r},t)}{dt} = F(\mathbf{r},t),\tag{3.1}$$

$$-\frac{\hbar}{2m} \left[\nabla^2 \psi + V(\mathbf{r}, t) \right] = i\hbar \frac{d\psi}{dt}.$$
(3.2)

En la mecánica clásica, la evolución temporal de un sistema se obtiene resolviendo la ecuación diferencial correspondiente al modelado físico del sistema. La solución puede ser cualquier función suave, y basta con una única función para definir esta evolución en todo el espacio y en sus niveles de energía, que son continuos. En cambio, en la mecánica cuántica, al tratarse de una ecuación de onda, las soluciones son sumas de eigenfunciones ponderadas, y los niveles energéticos se encuentran discretizados. Por lo tanto, todo estado cuántico se describe mediante la superposición de todas sus eigenfunciones. Dado que cada eigenfunción es una solución de su ecuación de onda debido a la linealidad, es habitual referirse a un estado cuántico como una superposición de varios estados cuánticos.

Las eigenfunciones de un sistema cuántico dependen completamente de su ecuación de onda. Sin embargo, dado que conocemos la forma general de las soluciones, y que las eigenfunciones forman un espacio vectorial, es posible representar la solución de un sistema de manera indefinida, utilizando un formalismo vectorial. Esta representación, conocida como la notación 'bra-ket', fue introducida por Paul M. Dirac (Dirac, 1939). En este formalismo, un vector ψ se representa con un ket $|\psi\rangle$, y su conjugado complejo se denota como un bra $\langle \psi |$ y sus vectores base u_n (eigenfunciones) como $|u_n\rangle$

$$|\psi\rangle = \eta_1 |u_1\rangle + \eta_2 |u_2\rangle + \eta_3 |u_3\rangle \dots \tag{3.3}$$

3.1.2. Operadores

La teoría cuántica comparte varias cantidades físicas con la teoría clásica, tales como la posición, el momento y el momento angular, entre otras. Sin embargo, existen cantidades cuánticas que no tienen una contraparte clásica, como el espín. Siguiendo el principio de correspondencia de Heisenberg, que establece que la mecánica cuántica debe coincidir con la clásica en el límite de grandes números cuánticos. En este contexto, podemos decir que las cantidades físicas de la mecánica clásica se heredan de la cuántica, mientras que aquellas que no tienen equivalente clásico se podrían considerar cantidades puramente cuánticas.

Para identificar estas cantidades en mecánica cuántica se utilizan operadores en lugar de fórmulas. Dado que las soluciones de los sistemas cuánticos son vectores, los operadores tienen una representación matricial o diferencial. Aunque la teoría cuántica impone pocas restricciones a la forma de estos operadores, para que sus efectos sean observables deben cumplir ciertas propiedades clave. Sea O un operador, este debe ser unitario (3.4), hermitiano (3.5) y sus eigenvectores deben formar un conjunto completo ortonormal, es decir, deben satisfacer la llamada relación de completez (3.6).

$$OO^{\dagger} = \mathbf{I}, \tag{3.4}$$

$$O^{\dagger} = O, \tag{3.5}$$

$$\sum_{i} |u_i\rangle \langle u_i| = 1. \tag{3.6}$$

La relación de completez asegura que cada vector base es único y que podemos utilizarlos como una base válida para describir el sistema. Para ello, se emplea el llamado operador proyector $|u_i\rangle \langle u_i|$. Entre los operadores cuánticos, el más importante es el Hamiltoniano, ya que la ecuación de onda de un sistema esta construida con este operador. Si el potencial es independiente del tiempo, la ecuación de Schrodinger se puede separar en el producto de una solución dependiente del espacio y otra dependiente del tiempo $\psi = \phi(\mathbf{r})\beta(\mathbf{t})$, en estos casos especiales, la solución de la parte temporal es una exponencial $\beta(\mathbf{t}) = e^{iEt/\hbar}$ y ϕ debe cumplir con la ecuación de Schrodinger independiente del tiempo (3.7). El Hamiltoniano es el operador que nos proporciona la energía total del sistema, lo que da sentido a la ecuación. Dado que el Hamiltoniano describe la energía del sistema, es habitual referirse a los niveles de energía como niveles del Hamiltoniano, o bien, como eigenfunciones del Hamiltoniano.

$$\hat{H} \left| \psi \right\rangle = E \left| \psi \right\rangle.$$
 (3.7)

3.1.3. Operador de densidad

Existen casos en los que las variables de un estado cuántico no están totalmente determinadas, de manera que no es posible representarlo con un vector de estado. Normalmente en estos sistemas, las variables sólo pueden ser conocidas estadísticamente, como el caso de un ensamble de un conjunto de partículas. Es aquí donde surge el problema de interpretación de la información disponible para describir el estado cuántico de la mejor manera posible. Para estudiar estos sistemas, es necesario utilizar un operador equivalente al vector de estado llamado operador de densidad.

Concepto de una mezcla estadística de estados

El problema de información incompleta siempre ha estado presente en la mecánica. La rama de la física que se encarga de estos sistemas es llamada mecánica estadística, la cual se resguarda en el concepto de probabilidad. Por ejemplo, un sistema en equilibrio termodinámico a temperatura T tiene una probabilidad proporcional a $e^{E_n/kT}$ de estar en el estado de energía E_n . En mecánica cuántica se puede hacer lo equivalente, a cada estado compuesto por una superposición de eigenfunciones se puede asignar una probabilidad p_n de que el sistema sea descrito por ese estado en particular, y por supuesto, la suma de estas probabilidades debe ser 1,

$$p_1 + p_2 + \dots = \sum_k p_k = 1. \tag{3.8}$$

En este caso se dice que el sistema es una mezcla estadística de los estados $|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_n\rangle$ con probabilidades $p_1, p_2, ..., p_n$. Resulta importante no confundir estas probabilidades con las probabilidades de colapso una base de un estado en específico. Las probabilidades p corresponden a todo el estado, mientras que las probabilidades $|\langle u_n|u_n\rangle|^2$ (Cohen-Tannoudji, Diu, y Laloë, 2019) se refiere a la probabilidad de que el estado al ser medido colapse en la base $|u_n\rangle$.

Para describir al operador de densidad es necesario tener en cuenta las siguiente propiedad de los operadores observables:

$$\langle u_n | A | u_p \rangle = A_{np}, \tag{3.9}$$

donde A_{np} es el elemento (n, p) de la matriz que representa el operador. La siguiente propiedad proporciona una forma de calcular el valor medio de un observable aplicado a un estado $|\psi\rangle$

$$\langle A \rangle = \langle \psi(t) | A | \psi(t) \rangle = \sum_{n,p} c_n^*(t) c_p(t) A_{np}.$$
(3.10)

Es evidente que el valor medio depende cuadráticamente de los coeficientes $c_n^*(t)$ y $c_p(t)$, pero estos coeficientes son simplemente los elementos de matriz del proyector $|\psi(t)\rangle \langle \psi(t)|$, pues

$$\langle u_p | \psi(t) \rangle \langle \psi(t) | u_k \rangle = c_n^*(t) c_p(t).$$
(3.11)

Ahora podemos definir el operador de densidad como:

$$\hat{\rho}(t) = |\psi(t)\rangle \langle \psi(t)|. \qquad (3.12)$$

El operador de densidad es representado en la base $\{|u_n\rangle\}$ por la llamada matriz de densidad, cuyos elementos son:

$$\rho_{pn}(t) = \langle u_p | \hat{\rho}(t) | u_n \rangle = c_n^*(t) c_p(t).$$
(3.13)

Es importante mencionar que se puede demostrar que el operador de densidad constituye una representación válida de un estado cuántico, y además proporciona un enfoque conveniente para encontrar propiedades de un observable.

Usando 3.13 y la ecuación 3.8 sabemos que la traza de esta matriz debe ser 1,

$$\sum_{n} |c_n(t)|^2 = \sum_{n} \rho_{nn} = \text{Tr}\{\hat{\rho}(t)\} = 1.$$
(3.14)

Con el operador de densidad tenemos otra forma de encontrar el valor medio de un observable

$$\langle A \rangle(t) = \sum_{n,p} \langle u_p | \hat{\rho}(t) | u_n \rangle \langle u_n | A | u_p \rangle, \qquad (3.15)$$

$$=\sum_{p}\left\langle u_{p}\right|\hat{\rho}(t)A\left|u_{p}\right\rangle , \tag{3.16}$$

$$= \operatorname{Tr}\{\rho(t)A\},\tag{3.17}$$

y con $\rho(t)$ es posible construir una ecuación de Schrödinger totalmente válida,

$$i\hbar \frac{d}{dt}\hat{\rho}(t) = [H(t), \hat{\rho}(t)], \qquad (3.18)$$

donde $[H(t), \rho(t)]$ es el conmutador de H(t) y $\hat{\rho}(t)$,

$$[H(t), \hat{\rho}(t)] = H(t)\hat{\rho}(t) - \hat{\rho}(t)H(t).$$
(3.19)

3.2. Cuantización del campo electromagnético

Históricamente, el campo electromagnético ha sido cuantizado en dos ocasiones. La primera fue por Max Planck en 1905 con su distribución, en la cual describió la energía de una onda electromagnética en función de su longitud de onda como una suma de cuantos de energía. Gracias a esta primera cuantización, se desarrolló la mecánica cuántica, la cual resultó ser muy útil para describir partículas individuales. Sin embargo, cuando se intentaba estudiar fenómenos colectivos de partículas, como una fuente de luz que consiste en un gran número de fotones, no se disponía de una teoría adecuada. Fue en este contexto que se propuso una segunda cuantización del campo electromagnético, basada en las ecuaciones clásicas del electromagnetismo y el tratamiento cuántico de la luz.

3.2.1. Segunda cuantización del campo electromagnético

Una forma de entender el desarrollo de la cuantización del campo electromagnético es la presentada en el libro de Daniel Walls (Walls y Milburn, 2008). El tratamiento comienza a partir de las ecuaciones de Maxwell para el campo electromagnético:

$$\boldsymbol{\nabla} \cdot \mathbf{B} = 0, \tag{3.20}$$

$$\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t},\tag{3.21}$$

$$\boldsymbol{\nabla} \cdot \mathbf{D} = 0, \tag{3.22}$$

$$\boldsymbol{\nabla} \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t},\tag{3.23}$$

donde $\mathbf{B} = \mu_0 \mathbf{H}$ y $\mathbf{D} = \epsilon_0 \mathbf{E}$, siendo μ_0 y ϵ_0 la permeabilidad magnética y la permitividad eléctrica del vacío, respectivamente. Por lo tanto, $\mu_0 \epsilon_0 = c^{-2}$. En el estudio de fenómenos electromagnéticos, normalmente se elige una norma. Las ecuaciones de Maxwell son invariantes ante cambios de norma en ausencia de fuentes. Una elección conveniente es la norma de Coulomb, donde tanto \mathbf{B} como \mathbf{E} están determinados por un potencial vectorial $\mathbf{A}(\mathbf{r}, t)$.

$$\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A},\tag{3.24}$$

$$\mathbf{E} = \frac{\partial \mathbf{A}}{\partial t}.\tag{3.25}$$

Con la condición de la norma de Coulomb

$$\boldsymbol{\nabla} \cdot \mathbf{A} = 0. \tag{3.26}$$

Sustituyendo en 3.24 podemos encontrar que $\mathbf{A}(\mathbf{r},t)$ satisface la ecuación de onda

$$\nabla^{2} \mathbf{A}(\mathbf{r}, \mathbf{t}) = \frac{1}{c^{2}} \frac{\partial^{2} \mathbf{A}(\mathbf{r}, t)}{\partial t^{2}}.$$
 (3.27)

Separando el potencial en dos términos complejos tenemos,

$$\mathbf{A}(\mathbf{r},t) = \mathbf{A}^{(+)}(\mathbf{r},t) + \mathbf{A}^{(-)}(\mathbf{r},t).$$
(3.28)

Donde $\mathbf{A}^{(+)}(\mathbf{r},t)$ contiene todas las amplitudes que varían como $e^{-i\omega t}$ para toda $\omega > 0$ y $\mathbf{A}^{(-)}(\mathbf{r},t)$ contiene todas las amplitudes que varían como $e^{i\omega t}$, además $\mathbf{A}^{(-)} = (\mathbf{A}^{(+)})^*$.

La solución de la ecuación de onda normalmente se expresa como una suma de funciones ortogonales y un propagador temporal:

$$\mathbf{A}^{(+)} = \sum_{k} c_k \mathbf{u}_k(r) e^{-i\omega_k t}, \qquad (3.29)$$

donde los coeficientes de Fourier c_k son constantes. El conjunto de funciones vectoriales $\mathbf{u}_k(\mathbf{r})$ que corresponden a las frecuencias ω_k deben satisfacer la ecuación de onda y la condición de transversalidad:

$$\left(\nabla^2 + \frac{\omega_k^2}{c^2}\right)\mathbf{u}_k(\mathbf{r}) = 0, \qquad (3.30)$$

$$\boldsymbol{\nabla} \cdot \mathbf{u}_k(\mathbf{r}) = 0. \tag{3.31}$$

Como resultado, estas funciones modales forman un conjunto completo ortogonal de funciones:

$$\int_{V} \mathbf{u}_{k}^{*}(\mathbf{r}) \mathbf{u}_{k'}(\mathbf{r}) d\mathbf{r} = \delta_{kk'}.$$
(3.32)

Las funciones dependen de la geometría del problema y de sus condiciones de frontera. Para el caso particular de un volumen cúbico de lados L, las funciones modales son:

$$\mathbf{u}_k(\mathbf{r}) = L^{-3/2} \hat{\mathbf{e}}^{(\lambda)} \exp\{i\mathbf{k} \cdot \mathbf{r}\},\tag{3.33}$$

donde $\hat{\mathbf{e}}^{(\lambda)}$ es el vector de polarización. El índice de modo k describe una configuración específica de las variables, índice de polarización (λ) = 1, 2 y las componentes del vector de propagación k. Cada componente del vector de onda toma los valores,

$$k_x = \frac{2\pi n_x}{L}, k_x = \frac{2\pi n_y}{L}, k_z = \frac{2\pi n_x}{L}, \ n_x, n_y, n_z = 0, \pm 1, \pm 2, \dots$$
(3.34)

El vector $\hat{\mathbf{e}}^{\lambda}$ es obligado a ser perpendicular a \mathbf{k} por la condición de transversalidad. Ahora el potencial puede escribirse de la siguiente manera:

$$\mathbf{A}(\mathbf{r},t) = \sum_{k} \left(\frac{\hbar}{2\omega_k \epsilon_0}\right)^{1/2} \left[\phi_k \mathbf{u}_k(\mathbf{r}) e^{-i\omega_k t} + \phi_k^* \mathbf{u}_k^*(\mathbf{r}) e^{i\omega_k t}\right].$$
 (3.35)

Normalmente el propagador temporal se contrae en las funciones modales haciéndolas dependientes del tiempo,

$$\mathbf{A}(\mathbf{r},t) = \sum_{k} \left(\frac{\hbar}{2\omega_k \epsilon_0}\right)^{1/2} \left[\phi_k \mathbf{u}_k(\mathbf{r},t) + \phi_k^* \mathbf{u}_k^*(\mathbf{r},t)\right].$$
(3.36)

Y en consecuencia el campo eléctrico:

$$\mathbf{E}(\mathbf{r},t) = i \sum_{k} \frac{\hbar \omega_k}{2\epsilon_0}^{1/2} \left[\phi_k \mathbf{u}_k(\mathbf{r},t) - \phi_k^* \mathbf{u}_k^*(\mathbf{r},t) \right].$$
(3.37)

donde los factores de normalización se escogieron para que las amplitudes ϕ_k y ϕ_k^* sean adimensionales. En la teoría electromagnética clásica, estas amplitudes son números complejos. El campo es cuantizado cuando se eligen ϕ_k y ϕ_k^* como los operadores adjuntos a, a^{\dagger} . Dado que los fotones son bosones las relaciones de conmutación adecuadas para los operadores son:

$$[a_k, a_{k'}] = \left[a_k^{\dagger}, a_{k'}^{\dagger}\right] = 0, \left[a_k, a_k^{\dagger}\right] = \delta_{kk'}.$$
(3.38)

En este contexto, se puede encontrar el Hamiltoniano del sistema:

$$H = \frac{1}{2} \int (\epsilon_0 \mathbf{E}^2 + \mu_0 \mathbf{H}^2) d\mathbf{r}.$$
 (3.39)

Sustituyendo las expresiones para \mathbf{E} y \mathbf{H} y usando las condiciones 3.28 y 3.29, el Hamiltoniano se convierte en la forma reducida siguiente:

$$H = \sum_{k} \hbar \omega_k \left(a_k^{\dagger} a_k + \frac{1}{2} \right).$$
(3.40)

Este Hamiltoniano coincide con el Hamiltoniano del oscilador armónico cuántico. Por lo tanto, el comportamiento dinámico de las amplitudes del campo eléctrico puede ser descrito por un ensamblaje de osciladores armónicos independientes. De manera intuitiva, se puede afirmar que representa la suma del número de fotones en cada modo, multiplicado por la energía de un fotón en dicho modo, más $\frac{1}{2}\hbar\omega_k$, que representa las fluctuaciones del vacío.

3.2.2. Estados de número

El Hamiltoniano 3.40 tiene eigenvalores $h\omega(n_k + \frac{1}{2})$, donde n_k es un entero $(n_k = 0, 1, 2, ..., \infty)$. Los eigenestados son escritos como $|n_k\rangle$ y son conocidos como **estados de número**. También, estos son eigenestados del operador de número $N_k = a_k^{\dagger} a_k$,

$$a_k^{\dagger} a_k \left| n_k \right\rangle = n_k \left| n_k \right\rangle. \tag{3.41}$$

El estado basal del oscilador armónico (o el estado del vacío) está definido por:

$$a_k \left| 0 \right\rangle = 0, \tag{3.42}$$

y su energía está dada por:

$$\langle 0|H|0\rangle = \frac{1}{2} \sum_{k} \hbar \omega_k. \tag{3.43}$$

Los operadores a_k^{\dagger} y a_k elevan y disminuyen los niveles de energía de las eigenfunciones del oscilador armónico. En términos de fotones, estos operadores representan la creación y aniquilación de un fotón con un vector de onda **k** y polarización $\hat{\mathbf{e}}_k$. De aquí surge la terminología de operadores de creación y aniquilación. Al aplicar estos operadores a los estados de número, obtenemos:

$$a_k |n_k\rangle = n_k^{1/2} |n_k - 1\rangle, a_k^{\dagger} |n_k\rangle = (n_k + 1)^{1/2} |n_k + 1\rangle.$$
(3.44)

Los estados de número pueden generarse aplicando el operador de creación sucesivamente en el vacío

$$|n_k\rangle = \frac{(a_k^{\mathsf{T}})_k^n}{(n_k!)^{1/2}} |0\rangle, n_k = 0, 1, 2...$$
 (3.45)

Estos estados son ortogonales y forman un set completo:

$$\langle n_k | m_k \rangle = \delta_{mn}. \tag{3.46}$$

$$\sum_{n_k=0}^{\infty} |n_k\rangle \langle n_k| = 1.$$
(3.47)

Por lo tanto, son una base válida del espacio de Hilbert. Los estados de número se forman contando fotón por fotón, haciéndolos adecuados para sistemas de fotones de altas energías, donde el número de fotones es muy pequeño, como en el caso de los rayos γ . Sin embargo, no son viables para representaciones de campos en los que el número de fotones es muy alto, ya que sería impráctico contarlos. Para solucionar este problema, típicamente, se utiliza una superposición o mezcla de estados de número.

3.2.3. Estados coherentes

Una base más apropiada para los campos ópticos son los estados coherentes. Un estado coherente tiene un número indefinido de fotones, y el producto de las incertidumbres en amplitud y fase para estos estados es el mínimo permitido por la mecánica cuántica. Estos estados son los más cercanos a una descripción clásica del campo.

En lugar de ser representados por operadores de aniquilación y creación, los estados coherentes utilizan los operadores de desplazamiento:

$$D(\alpha) = e^{\alpha a^{\dagger} - \alpha^* a}.$$
(3.48)

donde α es un número complejo arbitrario. Usando el teorema de los operadores tenemos que:

$$e^{A+B} = e^A e^B e^{[A,B]/2}, (3.49)$$

el cuál también es válido para:

$$[A, [A, B]] = [B, [A, B]] = 0.$$
(3.50)

Entonces el operador de desplazamiento puede escribirse de la forma equivalente siguiente:

$$D(\alpha) = e^{-|\alpha|^2/2} e^{\alpha a^{\dagger}} e^{\alpha^* a}.$$
 (3.51)

Cabe resaltar las propiedades del operador de desplazamiento

$$D^{\dagger}(\alpha) = D^{-1}(\alpha) = D(-\alpha), \ D^{\dagger}(\alpha)aD(\alpha) = a + \alpha D^{\dagger}(\alpha)a^{\dagger} + \alpha^{*}.$$
(3.52)

El estado coherente $|\alpha\rangle$ se genera al operar con $D(\alpha)$ en el estado del vacío. Los estados coherentes resultan ser eigenestados del operador de aniquilación, lo cual se puede comprobar utilizando las propiedades del operador D,

$$D^{\dagger}(\alpha)a |\alpha\rangle = D^{\dagger}(\alpha)aD(\alpha) |0\rangle = (a+\alpha) |0\rangle = \alpha |0\rangle.$$
(3.53)

Entonces tenemos que,

$$D^{\dagger}(\alpha)a \left| \alpha \right\rangle = \alpha \left| 0 \right\rangle, \tag{3.54}$$

operando en ambos lados $D(\alpha)$

$$a \left| \alpha \right\rangle = \alpha \left| \alpha \right\rangle.$$
 (3.55)

Como se mencionó anteriormente, el problema con los estados de número es que tienen un número finito de fotones. Por otro lado, los estados coherentes poseen un número indefinido de fotones. Esto se puede comprobar realizando una expansión de los estados coherentes en la base de los estados de número. Al tomar el producto escalar en ambos lados de la ecuación 3.55 con $|n\rangle$, se obtiene la relación recursiva:

$$(n+1)^{1/2} \langle n+1|\alpha \rangle = \alpha \langle n|\alpha \rangle.$$
(3.56)

Y por lo tanto,

$$\langle n|\alpha\rangle = \frac{\alpha^n}{(n!)^{1/2}} \langle 0|\alpha\rangle, \qquad (3.57)$$

se puede realizar la expansión de $|\alpha\rangle$ en términos de los estados de número $|n\rangle$ con coeficientes de expansión $\langle n | \alpha \rangle$.

$$|\alpha\rangle = \sum |n\rangle \langle n|\alpha\rangle = \langle 0|\alpha\rangle \sum_{n} \frac{\alpha^{n}}{(n!)^{1/2}} |n\rangle.$$
(3.58)

Para que esta expansión sea válida, es necesario que $|\alpha\rangle$ esté normalizado. Calculando su norma, obtenemos:

$$|\langle \alpha | \alpha \rangle|^{2} = |\langle 0 | \alpha \rangle|^{2} \sum_{n} \frac{|\alpha|^{2n}}{n!} = |\langle 0 | \alpha \rangle|^{2} e^{-|\alpha|^{2}}.$$
(3.59)

pero usando el operador de desplazamiento,

$$\langle 0|\alpha\rangle = \langle 0|D(\alpha)|0\rangle = e^{-|\alpha|^2/2},$$
(3.60)

entonces $|\langle \alpha | \alpha \rangle|^2 = 1$ y los estados coherentes están normalizados.

Recapitulando, los estados coherentes se pueden expandir en estados de número como:

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum \frac{\alpha^n}{(n!)^{1/2}} |n\rangle.$$
 (3.61)

Calculando la probabilidad de un estado coherente podemos encontrar la distribución de probabilidad que siguen estos estados.

$$P(n) = |\langle n | \alpha \rangle|^{2} = \frac{|\alpha|^{2n} e^{-|\alpha|^{2}}}{n!}.$$
(3.62)

Es evidente que la distribución de probabilidad que rigen a los estados coherentes es la distribución de Possion.

3.2.4. Estados termales

La mayoría de la luz natural es radiación térmica. La radiación térmica es el estado del campo electromagnético en equilibrio térmico. (Leonhardt, 2010)

Cuando el campo electromagnético interactúa con un material, se produce un intercambio de fotones. De esta manera, tanto la energía como el número de fotones de una fuente térmica varían en función de este intercambio (Leonhardt, 2010).

Para encontrar una expresión de la distribución de probabilidad para estos estados se seguirá un enfoque mecánico estadístico. Aquí, el problema se trata de una mezcla de estados.Bajo este esquema es imposible asignar un vector de estados único a este ensamble de fotones. El estado estacionario más general de una mezcla estadística de eigenestados de un Hamiltoniano \hat{H} , es una mezcla de estados de número:

$$\hat{\rho} = \sum_{n} \rho_n \left| n \right\rangle \left\langle n \right|, \qquad (3.63)$$

donde se ha usado el operador de densidad para representar la mezcla de estados. Dado que los estados termales son estados de máximo desorden, deben maximizar la entropía S para una energía E dada. También, debe satisfacer la conservación de la probabilidad total, la suma de las probabilidades ρ_n debe ser 1. Estas constricciones se le pueden restar a la entropía total del sistema:

$$S = -k_B \sum_{n} \rho_n \ln \rho_n - a \left(\sum_{n} \rho - 1 \right) - b \left(\sum_{n} \rho_n E_n - E \right), \qquad (3.64)$$

donde E es la energía promedio del modo a analizar. Como se mencionó, se trata de un problema de optimización, entonces es posible utilizar los multiplicadores de Lagrange para encontrar su solución. Diferenciando con respecto a la probabilidad e igualando a cero, obtenemos:

$$\frac{\partial S}{\partial \rho_n} = 0 = -k_B (\ln \rho_n + 1) - a - bE_n.$$
(3.65)

Esta ecuación tiene la solución:

$$\rho_n = \frac{1}{Z} \exp\left(-\frac{E_n}{k_B T}\right). \tag{3.66}$$

donde

$$T = \frac{1}{b}; Z = \sum_{n} \exp(1 + a/k_b).$$
 (3.67)

con T la temperatura y Z la función de partición o suma estadística:

$$Z = \sum_{n} \exp\left(-\frac{E_n}{k_B T}\right).$$
(3.68)

Para que la probabilidad total se conserve, cuando el equilibrio térmico es alcanzado, la entropía debe asumir el valor,

$$S = -k_B \sum_{n} \rho_n \ln \rho = k_B \ln Z + \frac{E}{T}.$$
 (3.69)

donde E es la energía promedio o energía interna:

$$E = \sum_{n} \rho_n E_n. \tag{3.70}$$

Suponiendo que T es variable, usando 3.68 como la definición de suma estadística, encontramos que $k_B T^2 dZ = ZE dT$. Por lo tanto, la derivada de la entropía 3.69 es la expresión:

$$\frac{dS}{dE} = \frac{1}{T}.\tag{3.71}$$

Esta expresión es la definición termodinámica de temperatura (Kondepudi, 2008) . Usando la expresión de Boltzmann para la distribución estadística de estados con energías E_n 3.63 en equilibrio térmico con temperatura T y combinando las ecuaciones 3.63, 3.66 y 3.68 en la expresión compacta, tenemos:

$$\hat{\rho} = \frac{1}{Z} \exp\left(-\frac{\hat{H}}{k_B T}\right), Z = \operatorname{tr}\left[\exp\left(\frac{\hat{H}}{k_B T}\right)\right].$$
(3.72)

Podemos introducir la constante β para el caso específico de un solo modo de luz,

$$\beta = \frac{\hbar\omega}{k_B T}.\tag{3.73}$$

Para el oscilador electromagnético del modo de luz, la suma estadística es la serie geométrica:

$$Z = \sum_{n}^{\infty} e^{-n\beta} = \frac{1}{1 - e^{-\beta}}.$$
(3.74)

Usando la matriz de densidad y la definición de \hat{H} , se encuentra la descripción del estado de luz termal,

$$\hat{\rho} = \left(1 - e^{-\beta}\right) \sum_{n=0}^{\infty} |n\rangle \langle n|. \qquad (3.75)$$

Para un número promedio de fotones, se llega al espectro de Plank del oscilador armónico en equilibrio térmico,

$$\langle n \rangle = \frac{1}{Z} \sum_{n=0}^{\infty} e^{-n\beta} n = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = \frac{1}{e^{\beta} - 1}, \qquad (3.76)$$

reescribiendo

$$e^{\beta} = \frac{1 + \langle n \rangle}{\langle n \rangle}.$$
(3.77)

Usando esta última ecuación junto con 3.75, se obtiene el operador de densidad de la luz termal:

$$\hat{\rho} = \sum_{n}^{\infty} \frac{\langle n \rangle^n}{\left(1 + \langle n \rangle\right)^{n+1}} \left| n \right\rangle \langle n \right|, \qquad (3.78)$$

con una probabilidad

$$P(n) = \frac{\langle n \rangle^n}{\left(1 + \langle n \rangle\right)^{n+1}}.$$
(3.79)

En la figura 3.1 se comparan las distribuciones de probabilidad de la luz coherente y termal para un $\langle n \rangle = 5$.



Figura 3.1: Distribuciones de probabilidad de luz coherente y termal

3.3. Bases del cómputo cuántico

El cómputo cuántico se perfila como una de las promesas tecnológicas de la actualidad, con la capacidad de superar los límites de la computación clásica en ciertos problemas específicos. Sin embargo, el único algoritmo que ha demostrado una ventaja significativa sobre los métodos clásicos es el algoritmo de Grover, que permite realizar búsquedas no estructuradas de manera más eficiente. A pesar de esto, se han propuesto diversas aplicaciones en áreas como la optimización de problemas (e.g. el cálculo de arg max), simulaciones químicas y la simulación de dinámicas Hamiltonianas (Manenti y Motta, 2023).

3.3.1. Qubits

El qubit es la unidad fundamental del cómputo cuántico, y puede describirse como el estado de un sistema físico con un Hamiltoniano de dos niveles, es decir, con dos energías accesibles. Aunque los valores energéticos y sus eigenfunciones dependen del hardware específico, como se mencionó en secciones anteriores, es común referirse a estos niveles como los estados de número $|0\rangle y |1\rangle$. El qubit puede entenderse como un análogo cuántico de un bit clásico, pero con la peculiaridad de que antes de ser medido no se encuentra en un estado definido.

La diferencia clave entre un qubit y un bit radica en uno de los postulados más controvertidos de la mecánica cuántica: el colapso de la función de onda, propuesto por Max Born en 1926 (Norsen, 2017). Este postulado establece que, al medir un sistema cuántico, su función de onda, que originalmente representa una superposición de todos los estados posibles, colapsa a un único estado definido (ver Figura 3.2). En el caso de los qubits, este colapso ocurre hacia uno de los dos estados posibles, $|0\rangle$ o $|1\rangle$; podría decirse que un qubit es el equivalente cuántico de un interruptor, pero con la capacidad de estaren una superposición de ambos estados hasta ser observado.

3.3.2. Transformaciones unitarias

Así como la mecánica cuántica tiene similitudes superficiales con la mecánica clásica, así también, existen similitudes entre el cómputo cuántico y el clásico. Matemáticamente, el estado de un qubit es representado como un estado cuántico $|\psi\rangle$ y sus vectores base son llamados bases computacionales. En la computación cuántica se utiliza la representación vectorial de kets. En esta representación, las amplitudes son guardadas en vectores columna, asignándolas a $|0\rangle$ o $|1\rangle$ dependiendo su posición.

En el caso de un solo qubit, su estado puede describirse como:

$$\begin{aligned} |\psi\rangle &= \gamma \left|0\right\rangle + \chi \left|1\right\rangle, \\ |\psi\rangle &= \begin{bmatrix} \gamma \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \chi \end{bmatrix}. \end{aligned} \tag{3.80}$$

Estos estados son equivalentes a los estados de espín $|\uparrow\rangle + |\downarrow\rangle$.

Cuando se opera con más de un qubit, el estado del sistema se describe mediante un producto tensorial de los qubits individuales:

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle.$$
(3.81)

Como los qubits son sistemas cuánticos, para alterar su estado es necesario utilizar operadores cuánticos, que normalmente se representan en notación matricial.


Figura 3.2: Esfera de Bloch

En el campo de la computación cuántica, estas transformaciones se conocen como compuertas cuánticas, análogas a las compuertas lógicas en la computación clásica.

Al igual que en la mecánica clásica, algunas compuertas cuánticas, como la compuerta X o NOT, tienen equivalentes en la lógica clásica, aunque la mayoría no los tiene. Entre las más importantes se encuentran las transformaciones de Pauli (X, Y, Z), que corresponden a rotaciones del qubit alrededor de distintos ejes en la esfera de Bloch. La esfera de Bloch es una representación gráfica de la orientación de un estado de espín; el eje z representa los números reales, mientras que los ejes x y y representan números complejos, como se muestra en la Figura 3.2. Esta representación es comúnmente utilizada en problemas de un solo qubit.

Una de las compuertas más importantes en el cómputo cuántico, y que se usa en el presente trabajo, es la compuerta Haddamard, esta compuerta crea estados equiprobables si se actúa sobre un estado $|1\rangle$ o $|0\rangle$,

$$H |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}},$$

$$H |1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$
(3.82)

Las representaciones matriciales de las compuertas suelen ser difíciles de manejar, pues las dimensiones de estas matrices son $2^n \times 2^n$ donde *n* es el número de qubits. Sin embargo, para el caso particular de un solo qubit son bastante ilustrativas:

1. Compuerta X o NOT:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

2. Compuerta Z:

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

3. Compuerta Y:

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

4. Compuerta Hadamard (H):

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

En esta representación, es evidente que las compuertas cuánticas constituyen un caso particular de las transformaciones de rotación sobre el propio eje de un sistema de espín. Esta relación no es accidental, ya que los sistemas de dos niveles fueron específicamente definidos para modelar estos fenómenos. En el contexto de la computación cuántica, las matrices de rotación que describen estas transformaciones son comúnmente conocidas como compuertas cuánticas generales (Vogel, 2011).

3.3.3. Compuertas control

Las compuertas cuánticas controladas son compuertas que pueden activar o desactivar su efecto en un qubit, dependiendo del valor de un qubit de control. Los dos casos más comunes de este tipo de compuertas son las compuertas control-X y control-Z, abreviadas como CX y CZ. En ambas, el qubit se rota sobre un eje de la esfera de Bloch, dependiendo del valor del qubit de control; específicamente, se activan cuando el qubit de control es 1.

1. Compuerta Control X:

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

2. Compuerta Control Z:

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

a) Compuerta X.

$$|q_0\rangle - X - |q'_0\rangle$$
(c) Compuerta Z.
 $|q_0\rangle - Z - |q'_0\rangle$
(c) Compuerta Z.
 $|q_0\rangle - Z - |q'_0\rangle$
(c) Compuerta CX.
 $|q_0\rangle - Z - |q'_0\rangle$
(c) Compuerta H.
 $|q_0\rangle - Y - |q'_0\rangle$
(c) Compuerta H.
 $|q_0\rangle - Y - |q'_0\rangle$
(c) Compuerta H.
 $|q_0\rangle - H - |q'_0\rangle$
(c) Compuerta CZ.
 $|q_0\rangle - H - |q'_0\rangle$
(c) Compuerta CZ.
 $|q_0\rangle - H - |q'_0\rangle$
(c) Compuerta CZ.
 $|q_0\rangle - H - |q'_0\rangle$
(c) Compuerta CZ.

Figura 3.3: Representación de compuertas cuánticas comunes en circuitos cuánticos



Figura 3.4: Circuito de una compuerta multicontrol Z y multicontrol X, a) y b) son representaciones comúnes de la compuerta control Z y c) es la representación de la compuerta control X

3.3.4. Circuitos cuánticos

Los circuitos cuánticos representan la contraparte cuántica de los circuitos lógicos digitales utilizados en la computación clásica. De manera similar a los circuitos clásicos, el principio fundamental consiste en aplicar una secuencia de operaciones (compuertas cuánticas) sobre qubits de entrada para obtener un resultado de salida. La mayoría de las compuertas se representan como cuadrados (ver Figura 3.3), excepto aquellas que realizan algún tipo de entrelazamiento entre qubits, como las compuertas controladas o multicontroladas.

Una generalización de las compuertas controladas son las compuertas multicontrol, que, en lugar de tener un solo qubit de control, pueden tener tantos como sean necesarios. Un ejemplo de un circuito cuántico con compuertas multicontrol puede ser observado en la Figura 3.4.

3.3.5. Obtención e interpretación de resultados

La teoría cuántica establece que al medir un estado cuántico, este colapsa en una de sus eigenfunciones. Este colapso ocurre con una probabilidad igual al cuadrado del



Figura 3.5: Un circuito cuántico con una medición del un qubit anciliar, este qubit está enredado con $q_0 \ge q_1$ por medio de una compuerta multi control X

valor absoluto de la amplitud de cada uno de los vectores base. En el cómputo cuántico, los resultados obtenidos son estas probabilidades, ya que se prepara y colapsa el sistema un número determinado de veces, conocido como el número de "shots". Para colapsar el sistema se utiliza la compuerta de medición, que almacena el resultado de la medición en un bit clásico. En la mayoría de los casos, se requiere que el estado cuántico no sea destruido; para ello, se emplea un qubit anciliar (auxiliar), que se entrelaza con los qubits que se desean medir, y se mide directamente este qubit. De este modo, se conserva el circuito original (véase Figura 3.5).

3.3.6. Estados cuánticos equiprobables

Una gran parte de algoritmos cuánticos utilizan estados particulares llamados estados REW (Real Equally Weighted). Estos estados se caracterizan por que todas sus bases son equiprobables, lo que significa que sus amplitudes sólo pueden cambiar en signo y deben ser reales. Un estado REW puede ser descrito de la siguiente forma:

$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{n}^{m} = (-1)^{f(n)} |\psi_n\rangle,$$
 (3.83)

donde f(n) es una función binaria. Para describir estos estados normalmente se utilizan circuitos con compuertas CZ. Sin embargo, esta representación resulta poco práctica, pues es necesario indicar la secuencia completa de compuertas, sus qubits de control y objetivo. Una alternativa prometedora son los estados de hipergráfos. Estos estados se basan en codificar sus relaciones en una hipergráfica, que resulta en una mejor representación y facilita su análisis (Rossi, Huber, Bruß, y Macchiavello, 2013).

Estados de Gráficas

Un estado de gráfica se basa en una gráfica matemática $g_2 = \{V, E\}$, compuesta por un conjunto de vértices V y un conjunto de aristas E, donde cada arista conecta exactamente dos vértices. El estado cuántico asociado se construye asignando a cada vértice un qubit inicializado como $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, y aplicando puertas C^2Z entre pares de vértices conectados:

$$|g_2\rangle = \prod_{\{i_1,i_2\}\in E} C^2 Z_{i_1i_2}|+\rangle^{\otimes n}$$

Un ejemplo de un estado de gráfica se muestra en la Figura 3.6.



Figura 3.6: Representación de un estado cuántico de gráfica con tres vértices y tres aristas.

Estados de Hipergráficas k-Uniformes

Una k-hipergráfica uniforme $g_k = \{V, E\}$ es una generalización de las gráficas, donde cada hiperarista conecta exactamente k vértices. Su estado cuántico asociado se define como:

$$|g_k\rangle = \prod_{\{i_1, i_2, \dots, i_k\} \in E} C^k Z_{i_1 i_2 \cdots i_k} |+\rangle^{\otimes n}.$$

En la Figura 3.7 se muestra una hipergráfica 3-uniforme con tres hiperaristas.



Figura 3.7: Representación de un estado cuántico de hipergráfica 3-uniforme con tres hiperaristas.

3.3.7. Estados de Hipergráficas

Una hipergráfica general $g_{\leq n} = \{V, E\}$ permite hiperaristas de cualquier orden k, donde $1 \leq k \leq n$. Su estado cuántico asociado se define como:

$$|g_{\leq n}\rangle = \prod_{k=1}^{n} \prod_{\{i_1, i_2, \dots, i_k\} \in E} C^k Z_{i_1 i_2 \cdots i_k} |+\rangle^{\otimes n}.$$



Figura 3.8: Representación de un estado cuántico de hipergráfica general con hiperaristas de diferente orden.

En la Figura 3.8 se muestra una hipergráfica general con hiperaristas de diferente orden. En la Ref. (Rossi y cols., 2013) han demostrado que estos estados coinciden con los estados REW.

3.3.8. Simulación de circuitos cuánticos

A diferencia del cómputo clásico, para diseñar algoritmos cuánticos, no es posible seguir la metodología que se adopta comúnmente, en la que una vez el programa está escrito se puede ejecutar en el hardware y así validar los resultados. En el caso de la computación cuántica, aunque los QPU están al alcance de cualquier persona que cuente con un correo electrónico, normalmente carece de disponibilidad con altos tiempos de espera para ejecutar un sólo circuito, incluso si la tuviera, es necesario tener en cuenta la sensibilidad al ruido y los errores de la computadora. Por lo tanto, resulta conveniente tener una metodología que permita trabajar de una forma eficiente.

Como se ha explicado, el comportamiento de una computadora cuántica es descrito por las leyes de la mecánica cuántica y estas leyes pueden ser codificadas en una computadora clásica para simular dinámicas. Por lo tanto, una computadora cuántica puede ser simulada utilizando una computadora clásica. Hay una gran variedad de paquetes de software que realizan estas simulaciones como lo son Qiskit, Pennylane, Amazon Bracket, Cirq, entre otros. En este trabajo se utilizó el paquete de Qiskit, pues es la utilizada por los equipos de IBM, además de ser la más extensamente documentada y utilizada en el área de investigación.

La librería Qiskit

Qiskit es una librería de código abierto que permite el uso del lenguaje de programación Python para el desarrollo, simulación y ejecución de algoritmos cuánticos



📃 Qiskit SDK 📃 Qiskit Runtime Service 📃 Qiskit Transpiler Service

Figura 3.9: Pasos para el desarrollo de un programa en una computadora cuántica según el grupo de desarrollo de Qiskit (*Introduction to QiSkit / IBM Quantum Documentation*, 2024)

(Javadi-Abhari y cols., 2024). Actualmente, es mantenida y utilizada por la compañía IBM, que emplea esta librería como su principal entorno de desarrollo para cómputo cuántico. Qiskit contiene todas las compuertas comúnmente utilizadas y se rige por circuitos cuánticos. Según Qiskit, la metodología para resolver un problema en cómputo cuántico se ilustra en la Figura 3.9, que consiste en primero encontrar una manera de plantear el problema en un circuito cuántico, seguido de la optimización del circuito planteado, para posteriormente ejecutarlo en una QPU y, finalmente, procesar los resultados.

Para ejecutar un circuito cuántico en Qiskit es necesario correr los siguentes pasos (Get started with primitives / IBM Quantum Documentation, 2024):

Inicializar la cuenta

1 2

3

4

5

Qiskit maneja los objetos llamados primitivas, que son los que permiten ejecutar códigos en una computadora cuántica real. La primitiva utilizada en el presente trabajo es la llamada SamplerV2(). Dado que esta primitiva es un servicio administrado. Primero es necesario inicializar una cuenta. Posteriormente, se puede elegir el QPU, a través del objeto **backend**, el cual contiene toda la información de la computadora objetivo.

Listing 3.1: Inicialización de una cuenta en qiskit y generación de un backend

Crear un circuito

Enseguida, es necesario crear un circuito. Para esto se utiliza el objeto quantum circuit En el ejemplo mostrado, se utiliza una transformación hermitiana cualquiera, al final se agrega la medición de todos los circuitos.

	Listing 3.2	2: Cre	ación	de	un	circuito	cuántico	alea	torio
--	-------------	--------	-------	----	----	----------	----------	------	------------------------

```
from giskit_ibm_runtime import QiskitRuntimeService
1
2
  import numpy as np
3
  from qiskit.circuit.library import IQP
4
  from qiskit.transpiler.preset_passmanagers import
5
        generate_preset_pass_manager
6
  from qiskit.quantum_info import random_hermitian
\overline{7}
8
  n_qubits = 127
9
10
  mat = np.real(random_hermitian(n_qubits, seed=1234))
11
  circuit = IQP(mat)
12
  circuit.measure_all()
13
```

Transpilar el circuito

Posteriormente, se transpila el circuito cuántico, ya que las computadoras cuánticas no poseen las compuertas convencionales de forma nativa. Para hacer esto, es necesario transformar estas compuertas en circuitos equivalentes que contengan las compuertas disponibles en el QPU objetivo. En este contexto, se crea un objeto parse manager, el cual contiene toda la información sobre la arquitectura de una QPU dada su backend. Una vez creado, se ejecuta su método run para generar el circuito transpilado. Actualmente, a este circuito se le denomina ISA por Instruction Set Architecture.

Listing 3.3: Transpilación de un circuito

Inicializar el Sampler

1

 $\mathbf{2}$

3

Para inicializar el objeto Sampler, es necesario especificar un modo de ejecución. Los modos de ejecución disponibles son batch, session o job. El modo más simple de ejecución es el de job. En este modo se abre una sesión en la cuenta de IBM y se envía un solo circuito. El modo session permite dejar una sesión abierta, lo que habilita el envío de varios circuitos a la vez y la ejecución en paralelo. Por otro lado, el modo batch es un administrador de varios jobs que ejecuta de manera eficiente los circuitos. Sin embargo, estos circuitos deben ser independientes. Para especificar un modo de ejecución en el sampler, basta con pasar al constructor un objeto batch, session o backend, correspondientes a los modos de batch, session y job, respectivamente.

Listing 3.4: Transpilación de un circuito

```
1 from qiskit_ibm_runtime import SamplerV2 as Sampler
2
3 sampler = Sampler(mode=backend)
```

Resultados

Para obtener los resultados es necesario correr el método run del objeto sampler. El objeto que tiene los resultados se llama PUB (Primitive Unified Bloc)

Listing 3.5: Obtener detalles de un trabajo de ejecución

```
1 job = sampler.run([isa_circuit])
2 print(f">>>_JOb_ID:_J{job.job_id()}")
3 print(f">>>_JOb_Status:_J{job.status()}")
```

Cada job tiene un ID único. Con este ID se puede obtener los resultados de la ejecución en cualquier momento.

```
Listing 3.6: Salida
```

```
1 >>> Job ID: 58223448-5100-4dec-a47a-942fb30edced
2 >>> Job Status: JobStatus.RUNNING
```

Los resultados están en el PUB, el cual se encuentra en la primera entrada del vector resultante de llamar al job.results(), específicamente, es necesario llamar al método get_counts() del objeto meas que es abreviación de measurement.

Listing 3.7: Obtener resultados del PUB

```
1 result = job.result()
2
3 # Obtener resultados del primer (y unico) PUB
4 pub_result = result[0]
5 print(f"Cuentas_del_registro_de_salida:
6 UUUUUUUu{pub_result.data.meas.get_counts()}")
```

Al final, se genera un diccionario con un registro de todas las mediciones que se realizaron. Cada base computacional tiene las frecuencias en las que se observaron.

Listing 3.8: Salida

		0
1	Cuentas	<pre>del registro de salida: {'0111': 50,</pre>
2	,0000,:	243, '0001': 101, '0101': 93, '0100': 188,
3	' 0011 ' :	128, '1011': 22,'0110': 13, '1100': 10,
4	'1000' :	24, '0010': 29, '1010': 26, '1101': 20,
5	1110 ::	45, '1001' : 16, '1111' : 16}



Figura 3.10: Perceptrón en su más simple realización

3.4. Aprendizaje automático en cómputo clásico y cuántico

3.4.1. La neurona artificial

Una neurona biológica, en términos generales, es una célula compuesta por dendritas, axón y un cuerpo, también llamado soma. El soma integra un núcleo, como cualquier otra célula. Las dendritas son las ramas que conectan el cuerpo con el exterior, permitiendo la recepción de señales provenientes de otras neuronas. El axón es el transmisor de la neurona, encargado de enviar señales a otras neuronas. La conexión entre neuronas se denomina sinapsis, que es la unidad de comunicación entre dos neuronas. Cuando el total de señales recibidas por una neurona supera un umbral, esta se activa (Zou, Han, y So, 2009).

La neurona artificial es un modelo computacional inspirado en una neurona biológica (ver Figura 3.10). El modelo más simple de una neurona artificial es el perceptrón. Este puede considerarse como un discriminador que, dada una entrada, clasifica el resultado como positivo o negativo. El perceptrón se compone de un vector de valores reales \vec{i} de dimensión m, que representa las señales de entrada, junto con un vector de pesos \vec{w} .

La clasificación se obtiene evaluando el producto interno de ambos vectores. El resultado del producto interno es procesado mediante una función de activación (por ejemplo la función de activación sigmoide mostrada en la Figura 3.11), que definirá la salida del la neurona. Si esta función de activación es una función Heaviside, el modelo se conoce como **perceptrón** (F., 1957). La interconexión de un conjunto de neuronas se conoce como Red Neuronal Artificial.

3.4.2. Perceptrón cuántico

La implementación de un perceptrón en un procesador cuántico ha sido abordada por (Tacchino y cols., 2019), quienes propusieron un modelo viable de perceptrón



Figura 3.11: Una función de activación sigmoide: $\sigma(x) = \frac{1}{1+e^{-x}}$

cuántico en el que los vectores de entrada \vec{i} y pesos \vec{w} son convertidos en estados cuánticos $|\psi_i\rangle$ y $|\psi_w\rangle$. Los datos de entrada deben estar restringidos al conjunto $\{1, -1\}$ para que la información pueda representarse adecuadamente. Esto se logra utilizando estados REW, donde cada una de las amplitudes del estado cuántico representa una entrada del vector de datos o pesos. En otras palabras, dichos estados siguen la estructura de las ecuaciones 3.85 y 3.86, donde cada base computacional está identificada por la representación decimal del índice del estado en superposición.

$$\vec{i} = \begin{bmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{bmatrix}, \ \vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{bmatrix}, \ i_j, w_j = 1, -1,$$
(3.84)

$$\left|\psi_{i}\right\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_{j} \left|j\right\rangle, \qquad (3.85)$$

$$\psi_w \rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j \left| j \right\rangle.$$
(3.86)

En este enfoque, es necesario generar un circuito cuántico específico para cada vector de entrada. Para cada par de vetores i y w es necesario un circuito cuántico. Todo el perceptrón se descompone en la acción de dos transformaciones unitarias U_i y U_w

Definición de las transformaciones unitarias

Se elige una transformación U_i tal que tome un estado inicial en el que todos los qubits estén en cero y los convierta al estado $|\psi_i\rangle$,

$$U_i \left| 0 \right\rangle^{\otimes N_q} = \left| \psi_i \right\rangle. \tag{3.87}$$

Posteriormente se selecciona una transformación U_w que transforme el estado $|\psi_w\rangle$ a la base computacional en el espacio de Hilbert, en otras palabras lo transforme al estado $|11...11\rangle$,

$$U_w |\psi_w\rangle = |1\rangle^{\otimes N_q} = |m-1\rangle.$$
(3.88)

El producto punto se realiza de la siguiente manera: Se empieza con todos los qubits en cero,

$$|0\rangle^{\otimes Nq}.$$
 (3.89)

Posteriormente, se usa U_i para obtener el estado de entrada:

$$U_i \left| 0 \right\rangle^{N_q} = \left| \psi_i \right\rangle. \tag{3.90}$$

En el siguiente paso es donde se realiza el producto interno. Al estado de entrada, se aplica U_w para rotarlo el ángulo θ que tendría que rotarse el estado de pesos $|\psi_w\rangle$ para alinearse con el eje vertical. Cabe recalcar que en ningún momento se construye el estado de pesos.

$$U_w |\psi_i\rangle = |\phi_{i,w}\rangle. \tag{3.91}$$

Una vez construido el nuevo estado $|\phi_{i,w}\rangle$ es posible encontrar el producto interno proyectando con el eje vertical $|1\rangle^{\otimes N_q} = |m-1\rangle$. El estado $|\phi_{i,w}\rangle$ tiene la siguiente estructura:

$$|\phi_{i,w}\rangle = c_0 |00..00\rangle + c_1 |00..01\rangle + \dots + c_{m-1} |m-1\rangle.$$
(3.92)

Es evidente que al recorrer el estado de entrada el mismo ángulo que tendría que recorrerse $|\psi_w\rangle$ la proyección $\langle \psi_w | \psi_i \rangle$ se vuelve equivalente a la proyección de $|\phi_{i,w}\rangle$ con el eje vertical, en otras palabras, el resultado que se está buscando está contenido en la amplitud c_{m-1} . Esto se puede demostrar utilizando las definiciones de los operadores 3.87 3.88 y la definición de $|\phi_{i,w}\rangle$ 3.91:

$$\langle \psi_w | \psi_i \rangle = \langle \psi_w | U_w^{\dagger} U_w | \psi_i \rangle, \qquad (3.93)$$

$$\langle \psi_w | \psi_i \rangle = \langle m - 1 | \phi_{i,w} \rangle,$$
 (3.94)

$$\langle \psi_w | \psi_i \rangle = c_{m-1}. \tag{3.95}$$

Esto se puede observar gráficamente en la Figuras 3.12 y 3.13.

Por lo tanto el perceptrón cuántico es un circuito variacional, o anzat (Javadi-Abhari y cols., 2024), pues este cambia su estructura dependiendo de los valores de entrada.



Figura 3.12: Representación gráfica de un perceptrón cuántico (Tacchino y cols., 2019)



Figura 3.13: Circuito del perceptrón cuántico, esencialmente de dos transformaciones unitarias U_i , U_w , de una compuerta MCX y una medición en un qubit anciliar

Entrenamiento

El perceptrón por si solo cae en la clasificación de métodos de aprendizaje automático supervisado, lo que significa que es necesario definir un conjunto de datos de entrenamiento y otro de evaluación. El perceptrón es entrenado al cambiar el vector de pesos siguiendo una regla de aprendizaje que se deriva de un proceso de optimización. Este proceso de ajuste de pesos se repite iterativamente hasta hallar un vector que minice la función de costo.

Se han desarrollado diversos esquemas de aprendizaje para redes neuronales artificiales (ANNs) con el objetivo de alcanzar diferentes metas de aprendizaje. Los enfoques más utilizados son los métodos de corrección de errores.

Los métodos de corrección de errores generalmente emplean un mecanismo de propagación hacia atrás (Zou y cols., 2009). Sea $y_{k,n}$ la salida del nodo (neurona) k-ésimo en el paso n de entrenamiento, y t_k la salida objetivo para ese nodo. Se puede definir una función de error como la diferencia entre la salida del nodo y la salida objetivo:

$$e_k = y_{k,n} - t_k,$$

donde e_k es el error para el nodo k-ésimo.

Sea λ una constante positiva que modula la tasa de ajuste de los pesos. Si se utiliza el método del descenso del gradiente y el error cuadrático medio como la función de

costo, el nuevo peso para la entrada x_i se calcula mediante la expresión:

$$w_{kj,n+1} = w_{kj,n} - \lambda e_k \cdot x_j.$$

El vector de pesos se actualiza en cada paso hasta que el sistema converge. Hay muchas formas de minimizar el error e_k .. En algunos problemas, el método del descenso del gradiente no resulta efectivo. En estos casos se puede recurrir a técnicas metaheurísticas, como por ejemplo algoritmos genéticos, algoritmos de enjambre o recocido simulado.

3.4.3. Algoritmos genéticos

Los algoritmos evolutivos (AE) son los metaheurísticos más influyentes de los últimos años en el ámbito de la optimización. Estos algoritmos representan una clase de métodos de optimización de propósito general que simulan un paradigma neodarwiniano y la genética de Mendel. Los algoritmos evolutivos consisten en un generador de poblaciones y un selector, así como en una función estimadora de aptitud y tres operadores genéticos básicos: cruce, mutación y selección. Los individuos en una población compiten e intercambian información entre sí (Du y Swamy, 2016). Los algoritmos genéticos son la forma más popular de algoritmos evolutivos. Este paradigma representa y codifica información utilizando conceptos de la teoría genética, tales como cromosomas y alelos. Los algoritmos genéticos pueden variar entre sí, pero en general, todos comparten la misma estructura:

- 1. Inicializar en t = 0
- 2. Generar una población aleatoria $\mathcal{P}(0)$
- 3. Repetir hasta que un criterio de término sea alcanzado :
 - a) Evaluar aptitud de cada individuo en $\mathcal{P}(t)$
 - b) Seleccionar de $\mathcal{P}(t)$ los individuos aptos para ser padres.
 - c) Aplicar operadores genéticos (mutación y cruce) a los padres y generar $\mathcal{P}(t+1)$
 - d) asignar t = t + 1

Terminologías de los algoritmos genéticos

Los algoritmos evolutivos hacen uso de terminologías tomadas de teorías biológicas. A continuación, se enlistan las más fundamentales:

- 1. **Población**: Un conjunto de individuos en una generación es llamado una población $\mathcal{P}(t) = \{\mathbf{x}_1, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{Np}, \}$ donde N_p es el tamaño de la población
- 2. Cromosoma. Cada individuo \mathbf{x}_i en una población es un cromosoma. Un cromosoma (o genoma) es un conjunto de parámetros que definen una solución del problema a optimizar. Debido a esto, se puede decir que una población en realidad es una población de soluciones.



Figura 3.14: Un cromosoma, gen y alelo en una representación binaria

- 3. Gen. Un cromosoma \mathbf{x} es una cadena de elementos x_i llamados genes $\mathbf{x} = (x_1, x_2, x_3, ..., x_n)$ donde *n* es el número de genes en el cromosoma. Cada gen codifica un parámetro del problema en un cromosoma. Un gen es usualmente codificado como una cadena binaria o un número real.
- 4. Alelo Los alelos son la unidad más pequeña de información en un cromosoma. Las diferencias entre gen, cromosoma y alelo se pueden observar en la figura 3.14
- 5. Genotipo. En biología, un genotipo se refiere a la codificación genética de un organismo, usualmente en la forma de ADN. En AE, un genotipo representa la codificación de una solución, en otras palabras, es otra forma de referirse a un cromosoma.
- 6. Fenotipo. En biología, el fenotipo de un organismo se refiere a la constitución física de un organismo o la manifestación de cierto rasgo. En el contexto de AE, un fenotipo es alguna característico de algún cromosoma, un ejemplo de fenotipo es su aptitud.
- 7. Aptitud. En biología se refiere a la habilidad de un individuo de cierto genotipo para reproducirse. Saliendo del aspecto biológico, la aptitud es una función que cuantifica la optimalidad de una solución (cromosoma). La función de aptitud es usada como función $f(\mathbf{x}_i)$ objetivo para un cromosoma \mathbf{x}_i .
- 8. Selección Natural. La selección natural altera poblaciones, es un proceso de selección del más apto al ambiente, en otras palabras a las restricciones del problema.
- 9. **Deriva genética**. Es un proceso por el cual los cromosomas son alterados de forma estocástica, este proceso puede ser a nivel de genes o alelos, un ejemplo de un operador de deriva genética es la mutación.

Selección

La selección en la mayoría de implementaciones de algoritmos genéticos trata de imitar el principio de la "supervivencia del más apto". Este proceso parte de una población inicial $\mathcal{P}(t)$ y forma la siguiente población $\mathcal{P}(t+1)$ con individuos suficientemente aptos para reproducirse. Existen varias formas de realizar la selección, todas basadas en la función objetivo de aptitud. Usualmente se dividen en tres tipos selección de ruleta, selección por ranking y selección por torneo.

La selección por ruleta consiste en seleccionar al azar a los padres, la probabilidad que tiene cada individuo de ser seleccionado es igual al valor normalizado de su aptitud.

$$P_{i} = \frac{f(\mathbf{x}_{i})}{\sum_{i=0}^{N_{p}} f(\mathbf{x}_{i})}.$$
(3.96)

Por otro lado, en la selección por ranking, se ordena a los individuos de acuerdo a su valor de aptitud. El mejor individuo obtiene el rango más alto N_p y el peor obtiene el rango más bajo 1. La probabilidad de selección se asigna linealmente de acuerdo a su rangos

$$P_i = \frac{1}{N_p} \left(\beta - 2(\beta - 1)\frac{i - 1}{N_p - 1} \right), i = 1, 2, ..., N_p,$$
(3.97)

donde β puede ser cualquier valor en el intervalo cerrado [0, 2]. Finalmente, la selección por torneo ocupa un número h de individuos a la vez. Los h cromosomas son comparados y una copia de los mejores individuos pasan a formar parte de los padres. El torneo se realiza un número N_p de veces hasta que se lleguen al número deseado de padres. Escoger un tipo de selección es crucial para el problema de optimización, pues cada uno de estas opciones tiene sus ventajas y desventajas. Por ejemplo, la selección uniforme es muy propensa a seleccionar individuos no aptos y la selección por torneo normalmente presenta sesgos ya que sólo usa información local.

cruce

Una vez los padres son seleccionados, tiene que haber un intercambio genético entre ellos para generar la nueva población, en otras palabras, que tengan hijos. Los tipos más comunes de cruce son, cruce de un sólo punto, de dos puntos, multipunto y cruce uniforme (ver Figura 3.15). El cruce de uno y dos puntos se basa en selecciona una región del cromosoma en ambos padres para posteriormente intercambiar los genes de ésa zona uno con el otro. En el cruce multipunto se seleccionan varias zonas para intercambiar y finalmente el cruce uniforme le asigna una probabilidad a cada gen para ser intercambiado.

Mutación, explotación vs exploración

La mutación a diferencia del cruce, requiere de un sólo padre para generar hijos. Se trata de un operador el cual selecciona aleatoriamente un alelo en un cromosoma y lo reemplaza con otra información, este cambio normalmente se trata de voltear el alelo. Una práctica común es la mutación uniforme, en la cual cada alelo tiene una probabilidad de presentar una mutación igual a 1/L donde L es el largo del cromosoma. Ahora la pregunta que queda es cuántos mutantes generar y cuántos padres.

Para contestar estas preguntas se definen las probabilidades P_c , P_m y N_p las cuales son la probabilidad de cruce, probabilidad de mutación y el número de individuos



Figura 3.15: Tipos de cruce, a) Crosover un punto, b) cruce de doble punto, c) cruce de multipunto, d) cruce uniforme

en la población. Cada uno de estos se ajusta de modo que cada cromosoma tiene una probabilidad P_c se ser candidato a ser padre y P_m de tener una mutación. Normalmente hay un juego entre estos parámetros.

- Incrementar P_c resulta en una exploración rápida al costo de probablemente destruir buenos cromosomas
- Incrementar P_m tiende a convertir un algoritmo genético en una búsqueda aleatoria.
- Incrementar N_p incrementa la diversidad de la población y reduce la probabilidad de convergencia prematura, aumentando el tiempo de convergencia.

4. Diseño, implemetación y evaluación de algoritmos

4.1. Estructura general

Para la implementación del perceptrón cuántico, fue necesario utilizar varios esquemas de cómputo. El proceso se puede resumir en cinco pasos mostrados en la Figura 4.16. Cada paso es ejecutado de manera 1) clásica, 2) cuántica simulada idealmente, 3) cuántica simulada con ruido o cuánticamente; exceptuando el entrenamiento y evaluación. En esta tesis se busca ejecutar todos los esquemas para establecer una comparación.

4.2. Obtención de los datos

El conjunto de datos del numero de fotones para una fuente termal y una coherente fueron obtenidos experimentalmente en la universidad de Louisiana y proporcionados en archivos CSV. Los datos incluyen las mediciones del numero de fotones con el tiempo de ambas fuentes de luz, termal y coherente, para diferentes números promedio de fotones (0.4, 0.53, 0.67, 0.77 y 0.735). Adicionalmente, los datos están organizados en términos del número de mediciones que usan para construir las distribuciones de probabilidad (llamados datapoints) de cada observación. Cada subcarpeta incluye 100,000 observaciones para cada fuente de luz.

En la Tabla 4.1 se muestra un ejemplo de un archivo de la base de datos para un numero promedio de fotones y numero de datapoints especificos. Los renglones corresponden a las probabilidades de obtener n fotones iniciando con n = 0 y finalizando con n = 6, es decir, el renglón 1 es la probabilidad de obtener cero fotones, P(0), el segundo renglón corresponde con P(1) y las columnas son la observaciones, es decir, tenemos 100,000 ejemplos de distribuciones de probabilidad para cada fuente.

4.3. Representación de los datos en estados cuánticos

La codificación de datos clásicos a cuánticos se refiere a convertir un dato clásico (binario) a un estado cuántico. Sin embargo, para realizar esta conversión, primero



Figura 4.16: Proceso general. Todo el proceso es híbrido, el tipo de cómputo se detona con distintos colores, dependiendo si fue totalmente clásico, simulado en qiskit, simulado con ruido o ejecutado en un procesador cuántico real.

	Observación 1	Observación 100K
P(0)	0.3	 0.5
P(1)	0.2	 0.1
P(2)	0.01	 0.05
P(3)	0.1	 0.03
P(4)	0.25	 0.12
P(5)	0.006	 0.004
P(6)	0.078	 0.001

Tabla 4.1: Estructura de la base de datos experimentales

es necesario realizar una pre-codificación (también clásica) de los datos que sea compatible con la codificación cuántica.

Esta primera parte se centrará en la etapa de pre-codificación y la subsecuente en la codificación cuántica. Como se presentó en el marco teórico, para la construcción del perceptrón cuántico son exclusivamente necesarios vectores con valores -1,1. Sin embargo, los datos experimentales consisten de 100k observaciones donde cada una de ellas es un arreglo de números flotantes, por lo tanto, fue necesario encontrar una transformación M que permitiera pasar de un vector de números flotantes en un vector de números enteros restringidos a $\{-1,1\}$, en otras palabras, $M: \mathbb{R}^7 \to P^{2^{N_q}} | P = \{-1,1\}$ donde N_q es el número de qubits del circuito.

En la presentación del perceptrón, se recalcó la importancia de los estados REW, pues los datos a procesar están codificados en las amplitudes de estos estados. Los estados REW, al restringir los valores de sus amplitudes a 1 y -1 nos permite verlos como otro sistema de numeración binaria, pues se codifican son sólo dos elementos. Por lo tanto, existe una correspondencia uno a uno con el sistema de numeración binario.

Haciendo uso de esta correspondencia es posible idear un sistema de pre-codificación para los datos, donde por medio de el escalamiento y binarización, se encuentra una metodología para representar cada observación en un vector compatible con la representación en estado cuántico REW.

Esta metodología se resume en los siguientes pasos:

Dado un vector \mathbf{i}^n dónde n es un número de observación

- Para cada entrada i_i^n cuantizar el valor flotante a un entero de 8 bits
- Convertir el entero a su representación binaria y guardarlo en un arreglo
- Repetir para cada n y concatenarlos en un sólo arreglo binario N
- Transformar los valores binarios 0, 1 a 1,-1 siguiendo la siguiente regla
 - $N_j = (-1)^{N_j}$.

De esta manera, si hablamos del arreglo de probabilidades como un histograma, cada distribución de probabilidad de una observación dada, es codificada en un número de 56 bits para ser transformado en un arreglo de 1 y -1 que cumple con los requerimientos para ser transformado en estado cuántico. En otras palabras, un arreglo de dimensiones $7 \times 32 = 224$ bits es convertido a un sólo número de 56 bits, para codificar estos datos a datos cuánticos basta con 6 qubits, pues el valor es codificado en la amplitud de cada base y dado que el número de bases para este sistema es $2^6 = 64$, el espacio es suficiente para representar las 56 amplitudes.

El procesamiento de los datos se encuentra resumido en la Figura 4.17



Figura 4.17: Preparación de datos clásicos

4.4. Preparación de los circuitos cuánticos

Como se ve explicó en el marco teórico, la estructura general del perceptrón cuántico es bastante simple, se puede observar en la Figura 3.13 que sólo se necesitan definir dos transformaciones unitarias U_i y U_w . Sin embargo, el problema recae en que estas transformaciones son variantes, para poder construirlas (Tacchino y cols., 2019) se ha recurrido a la definición de estados de hipergráficas. Como se vió en el marco teórico, estos estados son construidos por una serie de compuertas multicontrol Z. En Ref. (Tacchino y cols., 2019) se ha propuesto un algoritmo iterativo para construir estos estados. Sin embargo, para el presente trabajo, se utilizó una representación matricial de estas transformaciones unitarias.

4.5. Entrenamiento y evaluación

Para el entrenamiento del perceptrón cuántico se utilizaron las estrategias de algoritmos genéticos presentadas en el marco teórico. Para asegurar consistencia, en esta sección se utilizarán los conceptos de la teoría de algoritmos evolutivos presentada en secciones anteriores, llamando así a un vector de pesos \mathbf{w} como cromosoma y un conjunto de varios vectores de pesos como una población. Este entrenamiento se realizó de tres formas, un entrenamiento totalmente clásico, uno cuántico ideal simulado con qiskit y otro cuántico con ruido simulado. El diagrama de flujo del entrenamiento se puede observar en la figura 4.18

En el trabajo presente se abordan cuatro tipos de cómputo distintos, clásico, cuántico simulado idealmente, cuántico simulado con ruido y cuántico ejecutado. Por lo tanto, para poder evaluar el perceptrón, es necesario realizar las evaluaciones y entrenamientos en estos cuatro esquemas, sin embargo, cuanto más nos alejamos del cómputo clásico, el costo computacional sube, pues como se ha dicho anteriormente, la simulación de cómputo cuántico es una simulación de dinámicas cuánticas y para simularlas el espacio en memoria crece exponencialmente con el número de qubits, posteriermente al incluir el ruido de la computadora, este costo crece nuevamente. Cuando se llega a la computadora real, además del costo del circuito, es necesario permanecer el lista de espera para poder ejecutar los programas.

Dado que la etapa de entrenamiento es la más demandante computacionalmente, esta se acotó a cómputo clásico, cuántico simulado idealmente y cuántico simulado con ruido, dejando fuera una etapa de entrenamiento en la QPU real.

En principio, las proyecciones en los distintos esquemas son totalmente equivalentes, esto se explicó en la sección del marco teórico, por lo tanto el único elemento a cambiar es el tipo de cómputo que se usa para calcular las proyecciones entre los vectores $\mathbf{i} \ \mathbf{y} \ \mathbf{w}$. Para el caso clásico, se usó unicamente la librería de *numpy* para calcular el producto escalar entre ambos vectores. En la simulación ideal, basta con construir los circuitos cuánticos para cada dato de entrada y simularlos. Para la simulación con



Figura 4.18: Flujo de trabajo para entrenamiento de un algoritmo genético

ruido, es necesario transpilar el circuito usando el backend de la QPU a mimicar y simular el circuito transpilado.

58

4.5.1. Generación de la población inicial

Para la preparación de la población inicial, se creó una matriz de $l \times N_p$ donde *l* es la logitud del cromosoma, en el caso del perceptrón cuántico $l = m = 2^{N_q}$, los valores esta matriz tienen son 1, -1 en un orden aleatorio.

$$P_{1} = \begin{bmatrix} -1 & 1 & \dots & 1 & -1 \\ -1 & -1 & \dots & 1 & 1 \\ \vdots & \vdots & \vdots \\ 1 & -1 & \dots & 1 & 1 \\ -1 & 1 & \dots & -1 & -1 \end{bmatrix}.$$
 (4.98)

De esta forma, cada columna cumple con la estructura del vector \mathbf{w} 3.84. Cada una de las columnas de esta matriz es un cromosoma candidato o una solución al problema.

4.5.2. Selección

Función de aptitud

Para el entrenamiento se seleccionó a la funcional *overlap* 4.99 como función de aptitud. Esta funcional tiene de entrada dos vectores y regresa un número real entre el intervalo [0,1], este número real representa que tanto se parecen estas vectores (ver Figuras 4.19 y 4.20).

De este modo, el vector de pesos óptimo es el que minimice el overlap entre los histogramas de proyecciones de vectores de entrada coherentes y termales.

De este modo, se puede colocar un umbral en medio de ambas distribuciones para separarlas con un perceptrón.

$$\Omega[I_E, I_T] = \frac{\left[\int I_E^{1/2}(x) I_T^{1/2}(x) dx\right]^2}{\left[\int I_E(x) dx\right] \left[\int I_T(x) dx\right]}.$$
(4.99)

4.5.3. Selección de padres y mutantes

Una vez definida la función de aptitud y calculadas las aptitudes de cada individuo, se definió un número N_{par} de individuos que podrán se seleccionados como padre por cada generación. La selección de padres, en un principio, se realizaba por el método de ruleta. Sin embargo, este propiciaba a la selección de individuos no aptos y por lo tanto prolongaba el tiempo de convergencia. Entonces se decidió realizar la selección por medio de una estrategia elitista, donde se seleccionan a los N_{par} individuos con mayor aptitud para ser padres.

Primero, se definió un par de **vectores selectores**, los cuales indican que individuos pueden ser candidatos a padres y/o mutantes. Estos vectores selectores contienen



Figura 4.19: Funcional overlap entre dos distribuciones gaussianas, el overlap es la zona sombreada de color amarillo.

los índices de los individuos a ser seleccionados.

Para construir los vectores, es necesario ordenar las aptitudes de menor a mayor. Una vez ordenados, se pueden obtener los índices de los individuos más aptos y guardarlos en el vector selector de padres.

Para realizar las mutaciones se construyó un vector equivalente de índices para seleccionar mutantes. Sin embargo, se realiza un muestreo de individuos, y a cada uno se le asigna la misma probabilidad de ser mutante. Una vez ubicados los mutantes se realizó una mutación uniforme, donde a cada alelo se le da, nuevamente, la misma probabilidad de ser volteado.

A estos vectores se le denominó \mathbf{p}_{ind} y \mathbf{m}_{ind} . Un ejemplo se muestra a continuación:

$$\mathbf{p}_{ind} = \begin{bmatrix} 34 & 3 & \dots & 11 & 56 \end{bmatrix}, \tag{4.100}$$

$$\mathbf{m}_{ind} = \begin{bmatrix} 2 & 78 & \dots & 5 & 0 \end{bmatrix}. \tag{4.101}$$

Cada individuo tiene su aptitud, esta se guarda en un vector de aptitud \mathbf{f} de longitud N_p donde el indice del vector corresponde al del individuo en la matriz de población.

$$\mathbf{f} = [0.84, 0.56, \dots, 0.99, 0.19]. \tag{4.102}$$



Figura 4.20: Caso óptimo, dos distribuciones con un overlap nulo y un umbral entre las distribuciones

4.5.4. cruce y reemplazo

Una vez seleccionado los padres, se utiliza el vector selector para realizar un cruce uniforme, donde a cada alelo de la pareja de padres se asigna un cambio con la misma probabilidad 3.15. Para el reemplazo de los individuos, así como la selección de padres, esta también fue elitista. Se definió un número de hijos a considerar N_{child} , donde este número puede ser igual al número de padres. Posteriormente, se usa el vector \mathbf{p}_{ind} para seleccionar a los N_{child} individuos menos aptos de la población y se les reemplaza con los hijos para la siguiente generación.

4.5.5. Evaluación

La estrategia que se siguió se puede observar en la Figura 4.21, donde por cada etapa de entrenamiento se cambió el tipo de cómputo, para posteriormente a cada una de estas variantes evaluarlas de la misma manera. Por ejemplo, al entrenar clásicamente, el vector de pesos resultante es evaluado con un conjunto de evaluación en una computadora clásica, cuántica simulada ideal y cuántica simulada con ruido.

Para encontrar el umbral correspondiente a cada vector de pesos, se utilizaron las metricas error Tipo-I(α) y Error Tipo-II(β) (Quiroz-Juárez, Torres-Gómez, Hoyo-Ulloa, De J León-Montiel, y U'Ren, 2021) definidas por las ecuaciones 4.103 y 4.104

$$pr(\alpha) = pr(Z > p_{\alpha}|H_0); \qquad (4.103)$$



Figura 4.21: Proceso cruzado de entrenamiento y evaluación del algoritmo genético

$$pr(\beta) = pr(Z \le p_{\alpha}|H_0). \tag{4.104}$$

Donde Z es el estimador, p_{α} es el umbral y H_0 es la hipótesis nula. Se consideró que los datos son coherentes como hipótesis nula.

4.6. Ejecución en una QPU real

Una vez que se obtuvieron los resultados de los entrenamientos para los diferentes esquemas de computo, se eligió el umbral que mejor separaba los datos en la simulación ruidosa, debido a que esta simulación es la más cercana a la computación cuántica real. En vista de que el acceso a las computadoras cuánticas de IBM se encuentra restringido a 10 minutos por mes, se realizó una petición de créditos para cómputo cuántico a la empresa bajo este proyecto, y estos fueron concedidos, habilitando un poco más de 12 horas de computo.

Con el acceso a las computadoras cuánticas, se subieron los circuitos correspondientes a las proyecciones del vector de pesos óptimo y el conjunto de evaluación, estos resultados totalmente cuánticos fueron comparados con la evaluación del perceptrón clásico. 64

5. Resultados

5.1. Separabilidad de los datos

Es importante recalcar la importancia de la representación de un vector de caracterisicas en punto flotante y su codificación en una base de 1 y -1. En primer lugar, se debe notar que estamos pasando de números con resolución de 64 bits (punto flotante) a números de 8 bits (enteros). Además de eso, estamos concatenando estos números para formar un nuevo número, que integra cada componente del vector de caracteristicas pero que no asegura la conservación de la información premisa que deseamos obtener de estos datos. En el caso de la presente tésis, nos centramos en lograr la discriminación de los datos etiquetados como luz coherente y de los datos clasificados como luz termal.

Para comprobar esto se realizaron dos métodos, primeramente, se probó si el reescalamiento de los datos permitía su separabilidad. Para esto se graficó un espacio de características con tres ejes, cada uno correspondiendo al valor de una barra del histograma original. De esta forma, cada observación corresponde a un punto en el volumen. Posteriormente, se graficaron los puntos para ambos casos, termales y coherentes obteniendo dos nubes de puntos separables como se muestra en la Figura 5.22.

El siguiente paso fue comprobar que la transformación de vectores a estados cuánticos conserve separabilidad. Para ello se usaron las representaciones en vectores binario y de estados cuánticos. Con la representación binaria de los vectores, se realizó un histograma de los datos coherentes y termales usando su representación decimal, es decir a cada vector binario de la base de datos se convirtió en un número de 56 bits y se construyeron dos histogramas para ambas clases. Esto se puede observar en la Figura 5.23.

Una vez que observó que en representación binaria los datos son separables, el siguiente paso fue ver si con la representación en estados también se lograba esto. Sin embargo, no era posible seguir exactamente la misma estrategia que con los vectores binarios.

Lo que se realizó fue lo siguiente: se tomó el número con la frecuencia más alta de las clases; se obtuvo la representación binaria y en estados cuánticos de estos números y se proyectaron con todos los datos. Estas proyecciones se realizaron de forma direc-



Figura 5.22: Espacio de características para los datos de luz termal y coherente reescalados a resolución de 8 bits



Distribución de datos binarios $\langle n \rangle = 0.77$; dp = 160; bits = 8

Figura 5.23: Histograma de todos los vectores binarios en su representación decimal, se puede observar que las dos clases se separan. Se utilizaron datos reescalados a 8 bits, con un número promedio de fotónes igal a 0.77 y 160 datapoints.

ta, es decir los datos de cada clase se proyectaron con su vector más representativo o de forma cruzada, donde la clase se proyectó con el vector más representativo de la otra.

Como se puede observar en la Figura 5.24, al proyectar, la representación binaria parece no ser capaz de separar ambas clases. Por otro lado, en la representación de vector de estado, al usar una proyección cruzada, se puede observar que los datos son separables (ver Figura 5.25).

5.1.1. Perceptrón de dos qubits

Primeramente, fue necesario implementar el perceptrón de dos qubits de (Tacchino y cols., 2019), y se obtuvieron resultados muy similares a los presentados en el artículo. Como el perceptrón del artículo es de dos qubits, es posible explorar todo el espacio de combinaciones posibles de vectores **i** y **w** que permite este sistema. Este número es $2^m = 16$ elementos, donde $m = 2^{N_q}$.

En el artículo mencionado se propone un etiquetado de estos vectores para un manejo más sencillo. Pasando de su representación en las amplitudes $\{1,-1\}$ a su representación binaria, se puede obtener un par de números decimales k y w los cuales etiquetan a estos vectores, iterando de 0 a 15 en ambos números y ejecutando las proyecciones de estas combinaciones se obtuvó un mapa de calor de todos los valores



Figura 5.24: Histogramas de las proyecciones de vectores binarios, en el lado izquierdo cada clase se proyecta con su vector representativo y en el lado derecho las proyecciones se hacen con el vector de la otra clase. Los vectores representativos, en base decimal son $w_{coh} = 35000895515131904$ y $w_{th} = 40884347942994688$



Figura 5.25: Histogramas de las proyecciones con estados cuánticos, en el lado izquierdo cada clase se proyecta con su vector representativo y en el lado derecho las proyecciones se hacen con el vector de la otra clase.



Figura 5.26: Perceptrón cuántico de dos qubits evaluado clásicamente

posibles del perceptrón y se repitió este proceso para las proyecciones en cómputo clásico (ver Figura 5.26) y cuántico simulado ideal (ver Figura 5.27)

El artículo originalmente usa una métrica, definida como discrepancia, la cuál esta descrita por la ecuación 5.105, con el fin de comparar un perceptrón generado por una sucesión de compuertas Z en un circuito y uno generado por hipergráficas. Sin embargo, en el presente trabajo no se utilizó ninguno de esos métodos, aún así se usó la misma métrica para comparar el perceptrón cuántico simulado con el perceptrón clásico. Se puede observar la discrepancia individual de cada proyección en la Figura 5.28

$$\mathcal{D} = \frac{\sum_{\kappa,\gamma} |O_q(\kappa,\gamma) - O_c(\kappa,\gamma)|}{2^{2^{Nqbits+1}}}.$$
(5.105)

El siguiente paso fue hacer lo mismo en una simulación ruidosa, aquí es donde los resultados difieren. Para esta simulación se obtenían datos de discrepancias más altas que en el de la simulación ideal.

Se puede observar que el valor máximo no es 1, dando un indicio de que la norma se está perdiendo de alguna manera, como se muestra en la Figura 5.29.

A diferencia del caso de la simulación ideal, las discrepancias del perceptron con ruido, afectan los resultados de manera contundente y no pueden ser ignoradas (ver Figura 5.30).



Figura 5.27: Perceptrón cuántico de dos qubits evaluado en una simulación ideal.



Figura 5.28: Discrepancia entre los perceptrones clásico y cuánico simulado idealmente.



Figura 5.29: Perceptrón de dos qubits simulado con ruido



Figura 5.30: Discrepancia de un perceptrón cuántico simulado con ruido y uno implementado clásicamente


Figura 5.31: Error medio entre las proyecciones de datos coherentes, ejecutadas en una simulación realista (con ruido), con respecto a una ejecución clásica

5.1.2. Perceptrón de seis qubits

Las pruebas realizadas anteriormente sirvieron para saber que resultados esperar en la implementación real para el problema que se aborda en esta tesis. Los datos del histograma demandan un perceptrón cuántico de al menos 6 qubits. El perceptrón se implementó y probó; esperado que la simulación con ruido también presentara discrepancias altas y se confirmó con los resultados. Para probar que tanto se pueden reducir estas discrepancias se graficó la media del valor absoluto de la diferencia entre las proyecciones realizadas clásicamente y por el perceptrón ruidoso.

Aquí, se iteró el número de shots en el rango [10, 12000]. Los resultados se pueden observar en las figuras 5.31 y 5.32, donde se muestra que en el caso de los datos coherentes, hay una pendiente negativa y para los termales, una positiva. En ambos casos, el error no baja de 0.4, mucho más alto que en el problema de dos qubits que se mencionó anteriormente. Se intentó realizar la misma prueba para la QPU real brisbane de IBM, sin embargo, los tiempos de espera de cada circuito rondaban entre 1 a 3 horas, por lo tanto, sólo se mandaron casos clave; cuando el vector de pesos y entrada son ortogonales y cuando son iguales, en ambos casos no se obtuvieron los resultados esperados (0 y 1), indicando que la normalización de los estados se pierde en el proceso, por lo tanto, en la computadora cuántica real el ruido tampoco puede ser ignorado.

Esta es la razón por la cuál es necesario realizar el entrenamiento con los diferentes tipos de cómputo.



Figura 5.32: Error medio entre las proyecciones de datos termales, ejecutadas en una simulación realista (con ruido), con respecto a una ejecución clásica

5.2. Generación de los circuitos cuánticos

Basta con definir unas transformaciones unitarias como matrices de $N_q \times N_q$, donde la diagonal es igual al vector de entrada y al vector de pesos. Se definen $U'_i \ge U'_w$ como:

$$U_i' = \operatorname{diag}\{\mathbf{i}^T\}; \tag{5.106}$$

$$U'_w = \operatorname{diag}\{\mathbf{w}^T\}.$$
(5.107)

Se puede demostrar con algebra lineal que la transformación U'_i transforma el estado $|+\rangle^{\otimes m}$ en el estado $|\psi_i\rangle$, y en el caso de U'_w transforma el estado $|\psi_w\rangle$ a $|+\rangle^{\otimes m}$. Estas transformaciones primadas, resultan muy útiles para definir a las compuertas U.

Para el caso de U_i , es necesaria una compuerta que lleve $|0\rangle$ a $|+\rangle^{\otimes m}$. Sin embargo, encontrar esta compuerta es trivial, pues se trata de la compuerta Haddamard

$$U_i = H^{\otimes \ m} U'_i. \tag{5.108}$$

Para U_w , se tiene que encontrar una transformación que pase el estado $|+\rangle^{\otimes m}$ a $|1\rangle^{\otimes m}$, Esto se puede lograr en dos pasos, primeramente, aplicando la compuerta de Haddamard

$$H \left| + \right\rangle^{\otimes m} = \left| 0 \right\rangle^{\otimes m}. \tag{5.109}$$

Una vez aplicada la compuerta H, se puede voltear el estado con una compuerta X,

$$X|0\rangle^{\otimes m} = |1\rangle^{\otimes m}. \tag{5.110}$$



Figura 5.33: Preparación del circuito cuántico



Tabla 5.2: Configuraciones para el entrenamiento

Por lo tanto, podemos definir la transformación U_w de la siguiente manera:

$$U_w = X H^{\otimes \ m} U'_w \tag{5.111}$$

El proceso de preparación se puede observar en la figura 5.33.

5.3. Entrenamiento del perceptrón

Siguiendo la metodología de la Figura 4.21. Fue necesario definir una configuracion de entrenamiento poco costosa computacionalmente para realizar el entrenamiento en el entorno de simulación ruidoso. Los parámetros sobre el tamaño de los sets de datos es aplicado a ambos tipos de datos, coherentes y termales, de manera que se debe tener en cuenta que los parámetros reportados son multiplicados por dos. Las configuraciones pueden ser apreciadas en la tabla 5.2.



Entrenamiento con GA con overlap mínimo de 0.1255992656655885

Figura 5.34: Curva de aprendizaje del algoritmo genético

En la configuración exhaustiva se obtuvo un overlap mínimo de 0.1256

El proceso de entrenamiento se puede observar en la Figura 5.34, comenzando con un overlap de 0.36 y terminando con un overlap de 0.1255. La proyecciones del vector de pesos obtenido con el conjunto de entrenamiento fueron ploteados y se pueden observar en la Figura 5.35.

Para la configuración ligera de entrenamiento, se obtuvieron overlaps mucho más bajos, pues son muy pocos datos, véase la Figura 5.36.

5.4. Evaluación del perceptrón

Las eficiencias se obtuvieron para ambas configuraciones. En el caso de la configuración ligera, se obtuvieron todas las combinaciones dadas. Por otro lado, para la configuración exhaustiva, el tiempo que tomaría realizarla en la simulación ruidosa es no computable. Una proyección de un dato coherente y uno termal con el vector de pesos toma aproximadamente 30 segundos. Asumiendo que la multiplicación de este tiempo por el número total de proyecciones es igual al tiempo total de cómputo, se necesitaría por lo menos de 2000 × 100 × 150 × 30s = 9 × 10⁸s , o en otras palabras, se necesitaría más de 28 años y medio. Para generar el set de evaluación se tomó el 20 % del conjunto de datos tomados aleatoriamente de la muestra, excluyendo a los datos que se utilizaron para el entrenamiento. Para la simulación ruidosa se usó el backend de la computadora IBM brisbane de 127 qubits.



Figura 5.35: Proyecciones de los datos de entrenamiento con el vector de pesos encontrado



Figura 5.36: Resultados de entrenamiento para la configuración ligera, en el lado izquierdo es el entrenamiento clásico y el derecho el simulado idealmente



Figura 5.37: Prueba de error tipo I y II para el entrenamiento clásico en configuración exhaustiva

5.4.1. Evaluación del entrenamiento exhaustivo

Para el entrenamiento exhaustivo se obtuvo un umbral de 0.0755859375 por medio de la prueba de errores tipo I y tipo II (véase Figura 5.37). Las proyecciones en el test de evaluación para los casos de las proyecciones clásicas, simulación ideal y con ruido se encuentran en las figuras 5.38, donde se puede apreciar que las proyecciones del simulador ruidoso son las más alejadas de las clásicas, tanto el umbral, como el vector de pesos obtenido son totalmente incapaces de separar los datos; incluso moviendo el umbral es imposible diferenciarlos pues las proyecciones se encuentran en el mismo rango. Esto se puede ver ya que ambos histogramas, coherente y termal, se encuentran sobrepuestos, véase la figura 5.39.

Se evaluó el perceptrón clásico en los tres tipos de cómputo y se obtuvieron los resultados de la Figura 5.40. En esa figura se puede observar que en el caso del entrenamiento clásico, la evaluación con mayor eficiencia es la clásica con un 92.084 % de eficiencia, seguida por la evaluación del cuántico simulado ideal (91.68 %) y por último la del cuántico simulado con ruido (50 %).

Las eficiencias de esta configuración en la simulación ideal se encuentran en la Figura 5.41, donde se puede observar que, de nuevo, las mejores eficiencias son obtenidas por las proyecciones clásicas y simuladas idealmente, con la diferencia que esta vez la simulada idealmente es la que tiene mayor eficiencia con un 93.33 %. Los resultados se pueden ver en la Tabla 5.3.



Figura 5.38: Proyecciones clásicas y simuladas idealmente con el entrenamiento exhaustivo



Figura 5.39: Proyecciones simuladas con ruido en el set de evaluación del entrenamiento exhaustivo



Figura 5.40: Eficiencias del perceptrón entrenado clásicamente, evaluado en una computadora clásica (naranja), cuántica simulada idealmente (azul), cuántica simulada con ruido (azul oscuro)

Entrenamiento	Eficiencias		
	Clásica	Simulación ideal	Simulación realista
Clásico	0.92084	0.91683	0.50
Simulación ideal	0.92168	0.93373	0.50
Simulación realista	0.50	0.50	0.625

Tabla 5.3: Eficiencias de la configuración ligera

5.4.2. Evaluación del entrenamiento ligero

Como se ha dicho anteriormente, la utilidad del entrenamiento ligero radica en que es una forma de proyectar los resultados obtenidos por el exhaustivo. Usando los parámetros de la Tabla 5.2 y evaluando en los tres tipos de cómputo se obtuvieron las eficiencias mostradas en la Figura 5.42, es posible observar en la figuras 5.44 y 5.45 que, así como en el entrenamiento exhaustivo, los vectores obtenidos por un entrenamiento clásico y/o simulado idealmente, son incapaces de separar los datos de una simulación con ruido. Como se puede observar, el entrenamiento simulado con ruido es el único que puede separar los datos evaluados en este tipo de cómputo, pues es donde se obtuvo la eficiencia más alta (62.5%). La separación es más evidente si se observa el histograma de las proyecciones del conjunto de evaluación en la Figura 5.43. En la figura se puede apreciar que este vector de pesos es capaz de separar los



Figura 5.41: Eficiencias del entrenamiento largo usando simulación ideal, cada barra representa la eficiencia del perceptrón evaluado en cómputo clásico (naranja), cuántico simulado (azul) y simulado con ruido (azul oscuro).

Entrenamiento	Eficiencias		
	Clásica	simulación ideal	simulación realista
Clásico	0.625	0.75	0.50
simulación ideal	0.625	0.75	0.50
simulación realista	0.50	0.50	0.625

Tabla 5.4: Eficiencias de la configuración ligera

datos coherentes y termales de una simulación ruidosa.

Finalmente, todas las eficiencias se pueden observar en la Tabla 5.5 y graficadas en las figuras 5.47 y 5.46.



Figura 5.42: Eficiencias del entrenamiento ligero entrenado en una computadora cuántica simulada con ruido



Figura 5.43: Proyecciones simuladas con ruido del set de evaluación



Figura 5.44: Proyecciones clásicas y simuladas de los datos de entrada con el vector obtenido del entrenamiento clásico



Figura 5.45: Proyecciones simuladas con ruido de los datos de entrada con el vector de pesos entrenado clásicamente



Figura 5.46: Eficiencias en los conjuntos de evaluación de todos los entrenamientos para la configuración ligera. Cada color indica en que tipo de cómputo se realizó la evaluación. El eje horizontal denota en que tipo de cómputo se realizó el entrenamiento.



Figura 5.47: Eficiencias de todos los entrenamientos para la configuración exhaustiva.Cada color indica en que tipo de cómputo se realizó la evaluación. El eje horizontal denota en que tipo de cómputo se realizó el entrenamiento.

Configuración	Entrenamiento	Evaluación	Eficiencia
Ligera		Clásica	0.625
	Clásica	simulación ideal	0.625
		simulación realista	0.50
		Clásica	0.75
	simulación ideal	simulación ideal	0.75
		simulación realista	0.50
		Clásica	0.50
	simulación realista	simulación ideal	0.50
		simulación realista	0.625
Exhaustiva		Clásica	0.9208
	Clásica	simulación ideal	0.9168
		simulación realista	0.50
		Clásica	0.9216
	simulación ideal	simulación ideal	0.9337
		simulación realista	0.50
		Clásica	NA
	simulación realista	simulación ideal	NA
		simulación realista	NA

Tabla 5.5: Tabla configurada de acuerdo con la imagen proporcionada.

5. RESULTADOS

6. Discusión

El primer resultado relevante de esta investigación consiste en la transformación de la representación de los datos clásicos a un formato compatible con el cómputo cuántico.

Se observó que el tratamiento cuántico de los datos preserva la separabilidad de las clases. En las figuras 5.23 y 5.25 se muestra que, aunque la distribución binaria es separable (5.24), las proyecciones resultantes al utilizar otros vectores binarios pierden diferenciabilidad. En contraste, al emplear la representación en amplitudes, es posible observar que ambas clases permanecen separables.

Es importante señalar que para implementar la codificación discreta (cuántica) de los datos, a diferencia de la Ref. (Tacchino y cols., 2019), donde se utilizan estados de hipergráficas, el presente trabajo encontró una forma más sencilla y totalmente compatible con la librería de Qiskit tanto para simulaciones ideales como para simulaciones realistas (ruidosas). Este método demostró ser equivalente, pues con este método se implementó el perceptrón de dos qubits que proponen los autores y se obtuvieron resultados similares.

Se implementaron algoritmos genéticos para el entrenamiento del perceptrón cuántico, logrando entrenarlo en tres de los cuatro tipos de cómputo considerados. Donde en la evaluación de estos, el mejor tipo de entrenamiento fue el de qiskit, superando en la evaluación cuántica ideal y en la clásica. Específicamente, se obtuvo una eficiencia del 93.37 % en el entrenamiento y evaluación cuántica, a diferencia del 92.16 % usando entrenamiento cuántico y evaluación clásica. Esto contraste con el 91.68 % y 92.08 % del entrenamiento puramente clásico.

Los resultados obtenidos son comparables con los resultados de la Ref. (You y cols., 2020), haciendo uso de la codificación en amplitudes se redujo el espacio en memoria de los datos. En el trabajo mencionado se usan los 100,000 datos, mientras que el trabajo presente hace uso de 2,500. Visto en términos de memoria, cada dato es una columna de siete flotantes de 32 bits resultando en 2.8 Mb. Por otro lado, usando la codificación propuesta y la configuración exhaustiva del entrenamiento, el conjunto de datos ocupa 140 kb, usando un entero de 8 bits para guardar los 1 y -1, en otras palabras, se usa solamente el 5% de la memoria utilizada por los autores. Por lo tanto, inclusive sin tener en cuenta la parte del cómputo cuántico, resulta en

una mejora en cómputo clásico por sí solo el usar este tipo de codificación. Este resultado podría motivar la idea de implementar el algoritmo propuesto en esta tesis en sistemas embebidos de bajos recursos.

Regresando al cómputo cuántico, más específicamente a la simulación ruidosa. El ruido es un problema latente en el campo del cómputo cuántico, actualmente hay muchos esfuerzos en disminuirlo; sin embargo, en el presente trabajo se ha mostrado que, al menos utilizando la estructura de perceptrón cuántico presentada, es posible entrenarlo teniendo en cuenta el ruido inherente del sistema físico, realizando el entrenamiento en el simulador ruidoso o ejecutándolo directamente en la computadora. La dificultad radica en la facilidad de simular la computadora o la disponibilidad de la QPU real, por lo tanto, resultan necesarios paqueterías que sean capaces de simularlas en tiempos cortos o en su defecto aumentar la disponibilidad del hardware.

Una posible solución es la propuesta por (Huang y cols., 2024) donde crearon, verificaron y procesaron 27 clases de estados de hipergráficas de cuatro qubits, logrando altos niveles de fidelidad (> 90 %) en los estados generados. Esta propuesta de procesador cuántico específico para generación de estados de hipergráficas podría hacer posible el entrenar el perceptron directamente en el hardware cuántico.

7. Conclusiones

En la presente tesis se obtuvieron resultados relevantes para el reconocimiento de fuentes de luz utilizando aprendizaje automático cuántico, destacando que el mejor desempeño correspondió a un entrenamiento cuántico simulado bajo condiciones ideales con una eficiencia del 93.37 % de eficiencia según el método 'accuracy_score()' de la librería sklearn de python, definido por $\frac{1}{n} \sum_{i=0}^{n-1} 1(\hat{y}_i == y_i)$ (scikit-learn developers, 2025). Este entrenamiento se implementó utilizando Python, específicamente mediante la librería Qiskit.

Los algoritmos genéticos demostraron ser compatibles con el esquema cuántico, lo que valida su potencial en este contexto. Además, se exploró una metodología para entrenar en presencia de ruido, aunque la implementación en una unidad de procesamiento cuántico (QPU) real permanece como trabajo futuro, el echo de que el perceptrón entrenado en una simulación ruidosa, haya obtenido una eficiencia del 62.50 % indica que es posible entrenar el perceptrón directamente en la QPU.Las limitaciones enfrentadas en este trabajo son comunes en esta línea de investigación, especialmente la dificultad para probar los algoritmos y las consideraciones adicionales necesarias para su ejecución. Estos factores complican tanto el desarrollo como la validación de algoritmos cuánticos. No obstante, se identificó un caso en el que la formulación cuántica parece ofrecer ventajas sobre la clásica en el problema de identificación de fuentes de luz. Asimismo, gracias a la codificación cuántica, incluso operando de manera clásica, se logró reducir la complejidad del problema. De hecho, con un entrenamiento completamente clásico, se obtuvo una eficiencia del 92.08 %.

Es importante destacar que se propuso una estrategia sencilla para codificar datos clásicos en un sistema cuántico. En particular, el uso de los estados REW permite transformar cualquier vector binario en un estado cuántico mediante dos transformaciones unitarias, donde la diagonal de la matriz resultante es igual al vector que se desea codificar. Este esquema facilita una transición gradual del cómputo clásico al cuántico, permitiendo la computación de operaciones vectoriales, además del producto escalar.

Dado que la simulación de circuitos cuánticos puede ejecutarse en una GPU (development team, s.f.), una alternativa al entrenamiento en una QPU real sería paralelizar la gran cantidad de circuitos necesarios para el aprendizaje utilizando una GPU. Además, resultaría interesante explorar un esquema completamente cuántico para el entrenamiento del perceptrón. Como se mencionó en secciones anteriores, los algoritmos genéticos parecen ser compatibles con el cómputo cuántico, principalmente porque ambos son de naturaleza probabilística. Adicionalmente, el paralelismo inherente al cómputo cuántico permitiría explorar múltiples configuraciones de circuitos de manera simultánea (Vogel, 2011), lo que podría conducir a una implementación enteramente cuántica con una posible ventaja significativa.

Desde una perspectiva tecnológica, este algoritmo podría implementarse en sistemas embebidos de muy bajo consumo energético utilizando entrenamiento clásico. Asimismo, podría aplicarse en sistemas de microscopía avanzada, tales como la de tomografía de coherencia óptica cuántica (QOCT por sus siglas en inglés) (Yepiz-Graciano y cols., 2022), el trabajo presente podría usarse para la optimización de la detección de pares de fotones, permitiendo su uso fuera del laboratorio: la configuración clásica del algoritmo podría usarse para dispositivos portables, mientras que su configuración cuántica para arreglos totalmente cuánticos.

En el presente trabajo, el cómputo cuántico parece ofrecer una ligera ventaja sobre el clásico. Sin embargo, para evaluar con mayor precisión esta ventaja, sería necesario comparar los esquemas de cómputo clásico y cuántico en trabajos futuros. Esto incluiría la realización de pruebas como validación cruzada y análisis estadísticos para determinar si dicha ventaja es significativa o no.

Bibliografía

- AbuGhanem, M., y Eleuch, H. (2024). Nisq computers: A path to quantum supremacy. IEEE Access, 12, 102941–102961. doi: 10.1109/ACCESS.2024.3432330
- Bauer, B., Bravyi, S., Motta, M., y Chan, G. K.-L. (2020). Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22), 12685–12717. Descargado de https://doi.org/10.1021/acs.chemrev.9b00829 (PMID: 33090772) doi: 10.1021/acs.chemrev.9b00829
- Bhatia, V., y Ramkumar, K. (2020, 10). An efficient quantum computing technique for cracking rsa using shor's algorithm. 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), 89–94. Descargado de https://doi.org/10.1109/iccca49541.2020.9250806 doi: 10.1109/iccca49541.2020.9250806
- Cohen-Tannoudji, C., Diu, B., y Laloë, F. (2019). *Quantum mechanics, volume 1.* John Wiley Sons.
- development team, T. C.-Q. (s.f.). *Cuda-q.* Descargado de https://github.com/ NVIDIA/cuda-quantum
- Dirac, P. A. M. (1931). The principles of quantum mechanics. Clarendon Press.
- Dirac, P. A. M. (1939, 7). A new notation for quantum mechanics. Mathematical Proceedings of the Cambridge Philosophical Society, 35(3), 416–418. Descargado de https://doi.org/10.1017/s0305004100021162 doi: 10.1017/ s0305004100021162
- Du, K.-L., y Swamy, M. N. S. (2016). Search and optimization by metaheuristics. Springer. Descargado de https://doi.org/10.1007/978-3-319-41192-7 doi: 10.1007/978-3-319-41192-7
- Dubey, A., Gantala, T., Ray, A., Prabhakar, A., y Rajagopal, P. (2025). Quantum machine learning for recognition of defects in ultrasonic imaging. NDT E International, 150, 103262. Descargado de https://www.sciencedirect.com/ science/article/pii/S0963869524002275 doi: https://doi.org/10.1016/ j.ndteint.2024.103262
- Eibenberger, S., Gerlich, S., Arndt, M., Mayor, M., y Tüxen, J. (2013, 1). Matter-wave interference of particles selected from a molecular library with masses exceeding 10 000 amu. *Physical Chemistry Chemical Physics*, 15(35), 14696. Descargado de https://doi.org/10.1039/c3cp51500a doi: 10.1039/ c3cp51500a
- F., R. (1957). The perceptron: A perceiving and recognizing automaton,. Tech. Rep. Inc. Report No. 85-460-1, 021404.

- Feynman, R. P. (1982, 6). Simulating physics with computers. International Journal of Theoretical Physics, 21(6-7), 467–488. Descargado de https://doi.org/ 10.1007/bf02650179 doi: 10.1007/bf02650179
- Gallego Torromé, R., y Barzanjeh, S. (2024). Advances in quantum radar and quantum lidar. Progress in Quantum Electronics, 93, 100497. Descargado de https://www.sciencedirect.com/science/article/ pii/S0079672723000460 doi: https://doi.org/10.1016/j.pquantelec.2023 .100497
- Get started with primitives / ibm quantum documentation. (2024). Descargado de https://docs.quantum.ibm.com/guides/get-started-with-primitives
- Glauber, R. J. (1963, 6). The quantum theory of optical coherence. *Physical Review*, 130(6), 2529-2539. Descargado de https://doi.org/10.1103/physrev.130 .2529 doi: 10.1103/physrev.130.2529
- Hloušek, J., Dudka, M., Straka, I., y Ježek, M. (2019, Oct). Accurate detection of arbitrary photon statistics. *Phys. Rev. Lett.*, 123, 153604. Descargado de https://link.aps.org/doi/10.1103/PhysRevLett.123.153604 doi: 10.1103/PhysRevLett.123.153604
- Howard, L. A., Gillett, G. G., Pearce, M. E., Abrahao, R. A., Weinhold, T. J., Kok, P., y White, A. G. (2019, Sep). Optimal imaging of remote bodies using quantum detectors. *Phys. Rev. Lett.*, 123, 143604. Descargado de https://link.aps .org/doi/10.1103/PhysRevLett.123.143604 doi: 10.1103/PhysRevLett.123 .143604
- Huang, J., Li, X., Chen, X., Zhai, C., Zheng, Y., Chi, Y., ... Wang, J. (2024, 3). Demonstration of hypergraph-state quantum information processing. *Nature Communications*, 15(1). Descargado de https://doi.org/10.1038/s41467 -024-46830-7 doi: 10.1038/s41467-024-46830-7
- Huttner, B., Imoto, N., Gisin, N., y Mor, T. (1995, Mar). Quantum cryptography with coherent states. *Phys. Rev. A*, 51, 1863–1869. Descargado de https://link .aps.org/doi/10.1103/PhysRevA.51.1863 doi: 10.1103/PhysRevA.51.1863
- Introduction to qiskit / ibm quantum documentation. (2024). Descargado de https://docs.quantum.ibm.com/guides
- Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C. J., Lishman, J., Gacon, J., ... Gambetta, J. M. (2024). Quantum computing with qiskit. doi: 10.48550/ arXiv.2405.08810
- Khan, I., Elser, D., Dirmeier, T., Marquardt, C., y Leuchs, G. (2017, 6). Quantum communication with coherent states of light. *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences*, 375(2099), 20160235. Descargado de https://doi.org/10.1098/rsta.2016.0235 doi: 10.1098/rsta.2016.0235
- Kondepudi, D. (2008). Introduction to modern thermodynamics. John Wiley and Sons.
- Kudyshev, Z. A., Bogdanov, S. I., Isacsson, T., Kildishev, A. V., Boltasseva, A., y Shalaev, V. M. (2020, 9). Rapid classification of quantum sources enabled by machine learning. *Advanced Quantum Technologies*, 3(10). Descargado de https://arxiv.org/abs/1908.08577 doi: 10.1002/qute.202000067

- Leonhardt, U. (2010). Essential quantum optics. Cambridge University Press.
- Liu, J., y Shih, Y. (2009, Feb). Nth-order coherence of thermal light. Phys. Rev. A, 79, 023819. Descargado de https://link.aps.org/doi/10.1103/PhysRevA .79.023819 doi: 10.1103/PhysRevA.79.023819
- Manenti, R., y Motta, M. (2023). *Quantum information science*. Oxford University Press.
- Norsen, T. (2017). Foundations of quantum mechanics. Springer.
- Ollitrault, P. J., Miessen, A., y Tavernelli, I. (2021, 11). Molecular quantum dynamics: A quantum computing perspective. Accounts of Chemical Research, 54(23), 4229–4238. Descargado de https://doi.org/10.1021/acs.accounts .1c00514 doi: 10.1021/acs.accounts.1c00514
- Quiroz-Juárez, M. A., Torres-Gómez, A., Hoyo-Ulloa, I., De J León-Montiel, R., y U'Ren, A. B. (2021, 9). Identification of high-risk covid-19 patients using machine learning. *PLoS ONE*, 16(9), e0257234. Descargado de https://doi .org/10.1371/journal.pone.0257234 doi: 10.1371/journal.pone.0257234
- Rossi, M., Huber, M., Bruß, D., y Macchiavello, C. (2013, nov). Quantum hypergraph states. New Journal of Physics, 15(11), 113022. Descargado de https:// dx.doi.org/10.1088/1367-2630/15/11/113022 doi: 10.1088/1367-2630/15/ 11/113022
- scikit-learn developers. (2025). 3.4. metrics and scoring: quantifying the quality
 of predictions. Descargado de https://scikit-learn.org/stable/modules/
 model_evaluation.html#accuracy-score
- Sharma, M., Choudhary, V., Bhatia, R. S., Malik, S., Raina, A., y Khandelwal, H. (2021). Leveraging the power of quantum computing for breaking rsa encryption. *Cyber-Physical Systems*, 7(2), 73–92. Descargado de https://doi.org/ 10.1080/23335777.2020.1811384 doi: 10.1080/23335777.2020.1811384
- Slepyan, G., Vlasenko, S., Mogilevtsev, D., y Boag, A. (2022). Quantum radars and lidars: Concepts, realizations, and perspectives. *IEEE Antennas and Propagation Magazine*, 64(1), 16–26. doi: 10.1109/MAP.2021.3089994
- Streltsov, A., Adesso, G., y Plenio, M. B. (2017, 10). Colloquium: Quantum coherence as a resource. *Reviews of Modern Physics*, 89(4). Descargado de https://doi .org/10.1103/revmodphys.89.041003 doi: 10.1103/revmodphys.89.041003
- Sun, Y., Li, D., Xiang, Q., Yuan, Y., Hu, Z., Hua, X., ... Fu, Y. (2025). Scalable quantum convolutional neural network for image classification. *Physica A: Statistical Mechanics and its Applications*, 657, 130226. Descargado de https:// www.sciencedirect.com/science/article/pii/S0378437124007350 doi: https://doi.org/10.1016/j.physa.2024.130226
- Tacchino, F., Macchiavello, C., Gerace, D., y Bajoni, D. (2019, 3). An artificial neuron implemented on an actual quantum processor. *npj quantum information*, 5(1). Descargado de https://doi.org/10.1038/s41534-019-0140-4 doi: 10.1038/s41534-019-0140-4
- Thomson, G. P., y Reid, A. (1927, 6). Diffraction of cathode rays by a thin film. *Nature*, *119*(3007), 890. Descargado de https://doi.org/10.1038/119890a0 doi: 10.1038/119890a0
- Vogel, M. (2011, 11). Quantum computation and quantum information, by m.a.

nielsen and i.l. chuang. Contemporary physics, 52(6), 604–605. Descargado de https://doi.org/10.1080/00107514.2011.587535 doi: 10.1080/00107514 .2011.587535

- Walls, D., y Milburn, G. J. (2008). *Quantum optics*. Springer Science Business Media.
- Wiebe, N., Berry, D. W., Høyer, P., y Sanders, B. C. (2011, 10). Simulating quantum dynamics on a quantum computer. Journal of Physics A Mathematical and Theoretical, 44 (44), 445308. Descargado de https://doi.org/10.1088/1751 -8113/44/44/445308 doi: 10.1088/1751-8113/44/44/445308
- Yepiz-Graciano, P., Ibarra-Borja, Z., Ramírez Alarcón, R., Gutiérrez-Torres, G., Cruz-Ramírez, H., Lopez-Mago, D., y U'Ren, A. B. (2022, Sep). Quantum optical coherence microscopy for bioimaging applications. *Phys. Rev. Appl.*, 18, 034060. Descargado de https://link.aps.org/doi/10.1103/ PhysRevApplied.18.034060 doi: 10.1103/PhysRevApplied.18.034060
- You, C., Quiroz-Juárez, M. A., Lambert, A., Bhusal, N., Dong, C., Pérez-Leija, A., ... Magaña-Loaiza, O. S. (2020, 6). Identification of light sources using machine learning. *Applied physics reviews*, 7(2), 021404. Descargado de https://doi .org/10.1063/1.5133846 doi: 10.1063/1.5133846
- Zambra, G., Andreoni, A., Bondani, M., Gramegna, M., Genovese, M., Brida, G., ... Paris, M. G. A. (2005, Aug). Experimental reconstruction of photon statistics without photon counting. *Phys. Rev. Lett.*, 95, 063602. Descargado de https://link.aps.org/doi/10.1103/PhysRevLett.95.063602 doi: 10.1103/PhysRevLett.95.063602
- Zhang, C., Bromley, T. R., Huang, Y.-F., Cao, H., Lv, W.-M., Liu, B.-H., ... Adesso, G. (2019, Nov). Demonstrating quantum coherence and metrology that is resilient to transversal noise. *Phys. Rev. Lett.*, 123, 180504. Descargado de https://link.aps.org/doi/10.1103/PhysRevLett.123.180504 doi: 10.1103/PhysRevLett.123.180504
- Zou, J., Han, Y., y So, S.-S. (2009). Overview of artificial neural networks. En D. J. Livingstone (Ed.), Artificial neural networks: Methods and applications (pp. 14-22). Totowa, NJ: Humana Press. Descargado de https://doi.org/ 10.1007/978-1-60327-101-1_2 doi: 10.1007/978-1-60327-101-1_2