

Una introducción práctica al quantum machine learning y a los algoritmos variacionales cuánticos

Elías F. Combarro (Universidad de Oviedo)
efernandezca@uniovi.es

Universidad de Almería - Noviembre 2021



UNIVERSIDAD
DE ALMERÍA



Universidad de Oviedo

Parte I

Introducción y objetivos

¿Qué es la computación cuántica?

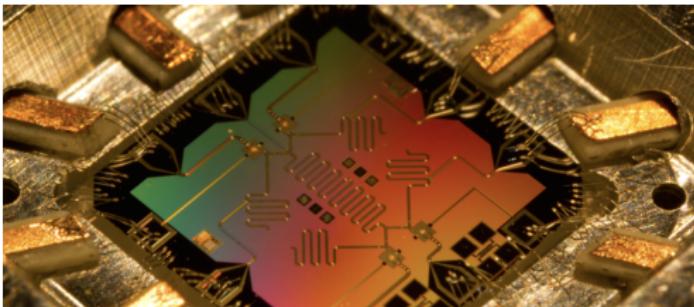
Computación cuántica

La computación cuántica es un paradigma probabilista de computación que utiliza las propiedades de la mecánica cuántica para realizar cálculos



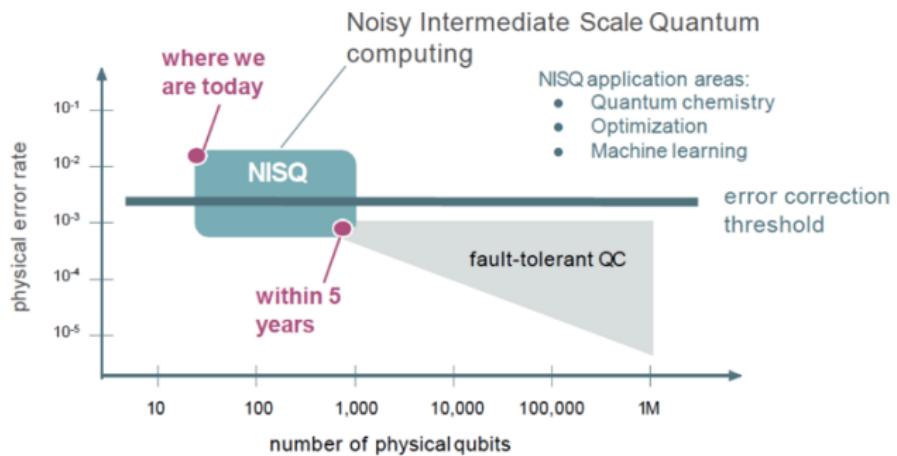
Aplicaciones de la computación cuántica

- Aceleración de tareas de búsqueda (algoritmo de Grover)
- Factorización de números (algoritmo de Shor)
- Resolución de sistemas de ecuaciones lineales (algoritmo HHL)
- Criptografía y comunicaciones cuánticas
- Simulación de procesos químicos y físicos
- Problemas de optimización
- Quantum Machine Learning



QC en los tiempos del NISQ

- Noisy Intermediate-Scale Quantum computers
 - Afectados por el ruido
 - Número de qubits limitado (50-100)
 - Sin conexión completa
- A pesar de ello, estos ordenadores podrían ser útiles en ciertas tareas

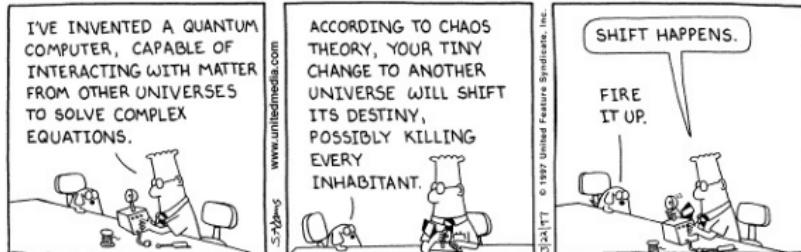


"Quantum computing in the NISQ era and beyond" Preskill, 2018 <https://arxiv.org/abs/1801.00862>



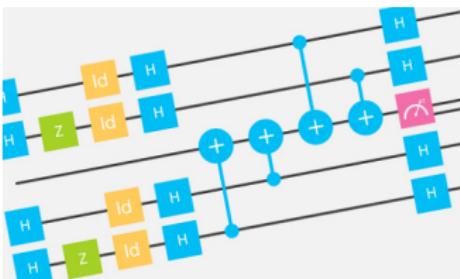
Objetivos de este curso

- Recordar los conceptos fundamentales de la computación cuántica
- Estudiar algunos de los algoritmos principales de la era NISQ
 - Variational Quantum Eigensolver (VQE)
 - Quantum Approximate Optimization Algorithm (QAOA) y Quantum Annealing
 - Quantum Support Vector Machines (QSVMs)
 - Variational Quantum Classifiers (VQCs)
 - Quantum Generative Adversarial Networks (QGANs)
- Ejecutar estos algoritmos en algunos casos sencillos
 - En simuladores
 - En ordenadores cuánticos reales



Recursos y herramientas

- Jupyter Notebooks
 - Entorno para la creación y ejecución de documentos que combinan código, texto enriquecido e imágenes
 - Se puede ejecutar localmente (Anaconda) o en la nube (mybinder.org, Google Colab...)
- IBM Quantum Experience
 - Acceso online gratuito a simuladores (hasta 32 qubits) y **ordenadores cuánticos reales** (1, 5, 7 y 15 qubits)
 - Diferentes topologías y lenguajes (python, qasm, Jupyter Notebooks)
 - Lanzado en mayo de 2016
 - <https://quantum-computing.ibm.com/>

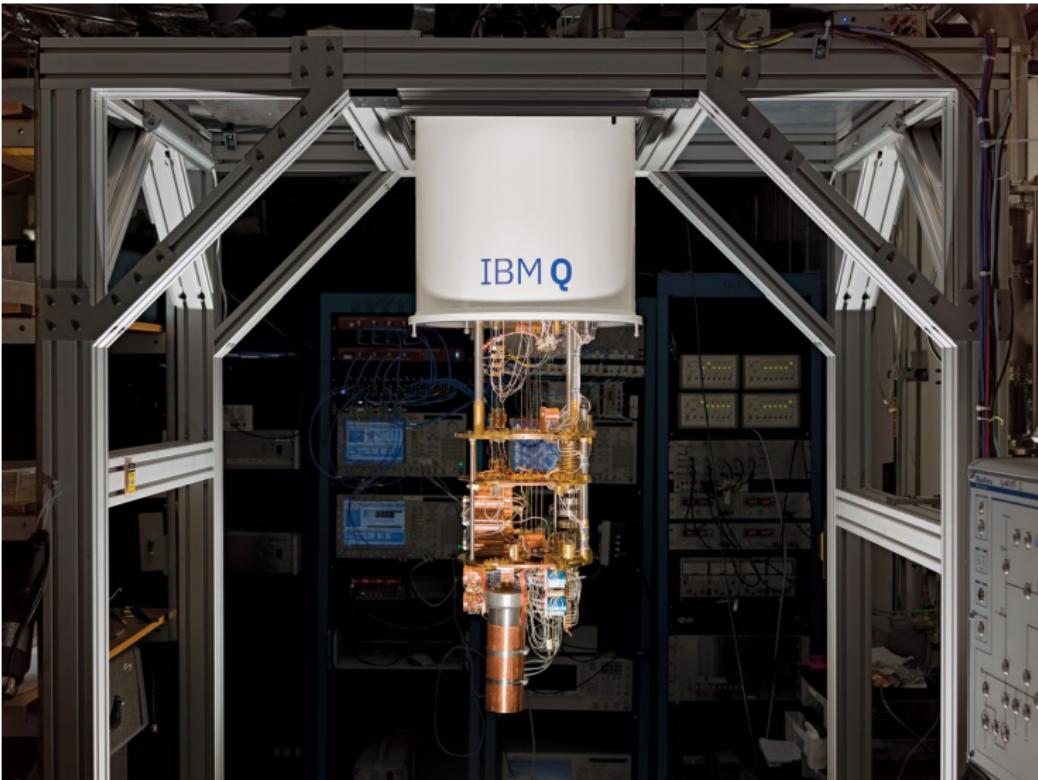


Recursos y herramientas (2)

- Quirk
 - Simulador online (hasta 16 qubits)
 - Gran número de puertas y opciones de visualización
 - <http://algassert.com/quirk>
- D-Wave Leap
 - Ocean: Librería en python para computación cuántica adiabática
 - Orientada a problemas específicos (QUBO, Quantum annealing...)
 - Leap: Acceso a *quantum annealers*
 - <https://ocean.dwavesys.com/>



Así es (y así suena) el IBM Q



The Sounds of IBM: IBM Q

Parte II

Conceptos básicos de computación
cuántica

Elementos de la computación cuántica

- Toda computación tiene tres elementos: datos, operaciones y resultados.
- En la computación cuántica, estos elementos se corresponden con los siguientes conceptos:
 - Datos = **qubits**
 - Operaciones = **puertas cuánticas** (transformaciones unitarias)
 - Resultados = **mediciones**
- Todos ellos se rigen por las leyes de la mecánica cuántica, por lo que pueden ser contrarios a la intuición



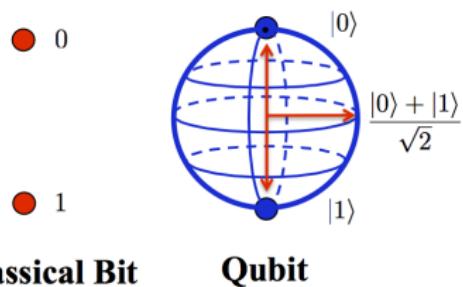
Qubits

- Un bit clásico es un elemento que puede tomar dos valores distintos (0 ó 1). Es discreto.
- Un qubit puede “tener” **infinitos** valores. Es continuo.
- Los qubits viven en un **espacio vectorial de Hilbert** que tiene por base dos elementos que denotamos $|0\rangle$ y $|1\rangle$.
- Un qubit genérico tiene la forma de una **superposición**

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

donde α y β son **números complejos** que cumplen

$$|\alpha|^2 + |\beta|^2 = 1$$



Medida de un qubit

- La única forma de conocer el estado de un qubit es realizar una medida. Sin embargo:
 - El resultado de la medida es aleatorio
 - Al medir, solo obtenemos un bit (clásico) de información
- Si medimos el estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ obtendremos 0 con probabilidad $|\alpha|^2$ y 1 con probabilidad $|\beta|^2$.
- Además, el nuevo estado de $|\psi\rangle$ después de realizar la medida será $|0\rangle$ o $|1\rangle$ según el resultado que se haya obtenido (colapso de la función de onda)
- Es más, no podemos realizar varias medidas de $|\psi\rangle$ porque no se puede copiar el estado (**teorema de no clonación**)



La esfera de Bloch

- Una forma habitual de representar el estado de un qubit es mediante la llamada esfera de Bloch
- Si $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ con $|\alpha|^2 + |\beta|^2 = 1$ podemos encontrar ángulos γ, δ, θ tales que

$$\alpha = e^{i\gamma} \cos \frac{\theta}{2}$$

$$\beta = e^{i\delta} \sin \frac{\theta}{2}$$

- Como las fases globales son físicamente irrelevantes, podemos reescribir

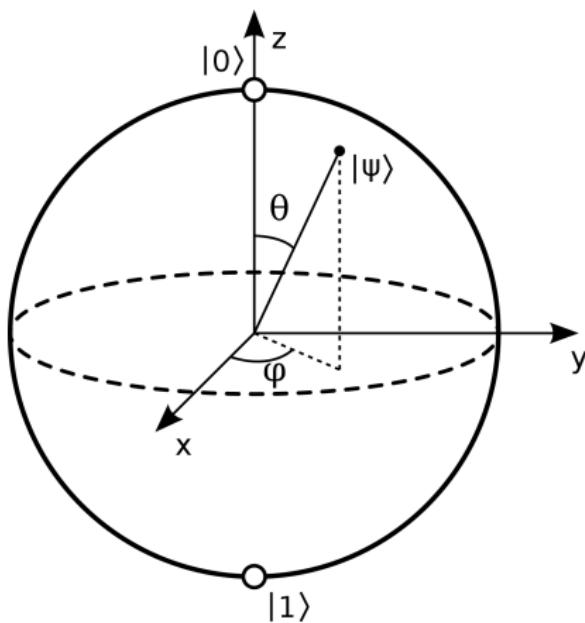
$$|\psi\rangle = \cos \frac{\theta}{2}|0\rangle + e^{i\varphi} \sin \frac{\theta}{2}|1\rangle$$

con $0 \leq \theta \leq \pi$ y $0 \leq \varphi < 2\pi$.

La esfera de Bloch (2)

- De los ángulos en $|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$ podemos obtener coordenadas esféricas para un punto en \mathbb{R}^3

$$(\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$$



Producto escalar, notación de Dirac y esfera de Bloch

- El producto escalar de dos estados $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ viene dado por

$$\langle\psi_1|\psi_2\rangle = (\overline{\alpha_1} \ \overline{\beta_1}) \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2$$

- Nótese que $\langle 0|0\rangle = \langle 1|1\rangle = 1$ and $\langle 0|1\rangle = \langle 1|0\rangle = 0$
- Esto nos permite calcular del siguiente modo

$$\begin{aligned}\langle\psi_1|\psi_2\rangle &= (\overline{\alpha_1}\langle 0| + \overline{\beta_1}\langle 1|)(\alpha_2|0\rangle + \beta_2|1\rangle) \\ &= \overline{\alpha_1}\alpha_2\langle 0|0\rangle + \overline{\alpha_1}\beta_2\langle 0|1\rangle + \overline{\beta_1}\alpha_2\langle 1|0\rangle + \overline{\beta_1}\beta_2\langle 1|1\rangle \\ &= \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2\end{aligned}$$

- Puntos antipodales en la esfera de Bloch se corresponden con estados ortogonales

Puertas cuánticas

- Las leyes de la mecánica cuántica nos dicen que la evolución de un sistema responde a la ecuación de Schrödinger (si no se realiza una medida).

$$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle$$

- En el caso de la computación cuántica, esto implica que las operaciones que se pueden realizar son transformaciones lineales que vienen dadas por matrices unitarias. Es decir, matrices U de números complejos que verifican

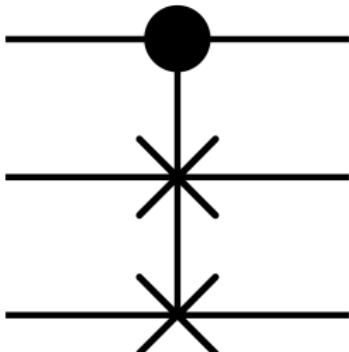
$$UU^\dagger = U^\dagger U = I$$

donde U^\dagger es la transpuesta conjugada de U .

- Cada matriz de este tipo es una posible puerta cuántica en un circuito cuántico

Computación reversible

- Como consecuencia, todas las operaciones tienen una inversa: **computación reversible**
- Todas las puertas tienen el mismo número de entradas que de salidas
- No podemos implementar directamente operaciones como *or*, *and*, *nand*, *xor*...
- Teóricamente, podríamos realizar cualquier computación sin gastar energía



Puertas cuánticas de un qubit

- Si tenemos un solo qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, habitualmente lo representamos como un vector columna $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
- Entonces, una puerta cuántica de un qubit se corresponderá con una matriz $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ que verifica
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
siendo $\bar{a}, \bar{b}, \bar{c}, \bar{d}$ los conjugados de los números complejos a, b, c, d .

Acción de una puerta cuántica de un qubit

- Un estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ es transformado en

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix}$$

es decir, en el estado $|\psi\rangle = (a\alpha + b\beta)|0\rangle + (c\alpha + d\beta)|1\rangle$

- Como U es unitaria, se cumple que

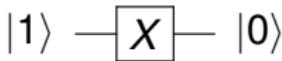
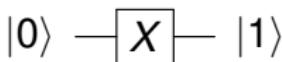
$$|(a\alpha + b\beta)|^2 + |(c\alpha + d\beta)|^2 = 1$$

La puerta X o NOT

- La puerta X viene definida por la matriz (unitaria)

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

- Su acción es (notación del modelo de circuitos)



es decir, actúa como un NOT

- Su acción sobre un qubit general sería

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{\boxed{X}} \beta |0\rangle + \alpha |1\rangle$$

La puerta H

- La puerta H o puerta de Hadamard viene definida por la matriz (unitaria)

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

- Su acción es

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

$$|1\rangle \xrightarrow{H} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

- Se suele denotar

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

y

$$|-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

La puerta Z

- La puerta Z viene definida por la matriz (unitaria)

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

- Su acción es

$$|0\rangle \xrightarrow{\boxed{Z}} |0\rangle$$

$$|1\rangle \xrightarrow{\boxed{Z}} -|1\rangle$$

Otras puertas

- Puerta Y

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

- Puerta T

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- Puerta S

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

- La puerta $R(\alpha)$ o puerta de fase, que depende de un parámetro (el ángulo α)

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

Puertas de rotación

- Podemos definir las siguientes puertas de rotación

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}X = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Y = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

- Se cumple que $R_X(\pi) \equiv X$, $R_Y(\pi) \equiv Y$, $R_Z(\pi) \equiv Z$,
 $R_Z(\frac{\pi}{2}) \equiv S$, $R_Z(\frac{\pi}{4}) \equiv T$

Usando las puertas de rotación para generar puertas de un qubit

- Para cada puerta U de un qubit, existe un vector $r = (r_x, r_y, r_z)$ de longitud 1 y un ángulo θ tal que

$$U \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} (r_x X + r_y Y + r_z Z)$$

- Por ejemplo, eligiendo $\theta = \pi$ y $r = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$ se puede ver que

$$H \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = -i \frac{1}{\sqrt{2}} (X + Z)$$

- Además, se puede demostrar que existen ángulos α , β y γ tales que

$$U \equiv R_Z(\alpha)R_Y(\beta)R_Z(\gamma)$$

Trabajando con dos qubits

- Cada qubit puede estar en los estados $|0\rangle$ y $|1\rangle$
- Así que para dos qubits tenemos cuatro posibilidades:

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

que también se denotan

$$|0\rangle |0\rangle, |0\rangle |1\rangle, |1\rangle |0\rangle, |1\rangle |1\rangle$$

o

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

- Como podemos tener superposiciones, un estado genérico del sistema será

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

donde los α_{xy} son números complejos que cumplen

$$\sum_{x,y=0}^1 |\alpha_{xy}|^2 = 1$$

Medida de un estado de dos qubits

- Tenemos un estado

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

- Si medimos ambos qubits, obtendremos:
 - 00 con probabilidad $|\alpha_{00}|^2$ y el nuevo estado será $|00\rangle$
 - 01 con probabilidad $|\alpha_{01}|^2$ y el nuevo estado será $|01\rangle$
 - 10 con probabilidad $|\alpha_{10}|^2$ y el nuevo estado será $|10\rangle$
 - 11 con probabilidad $|\alpha_{11}|^2$ y el nuevo estado será $|11\rangle$
- Es una situación análoga a la que teníamos con un solo qubit, pero ahora con cuatro posibilidades

Medida de un qubit en un estado de dos qubits

- Sobre un estado

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

también podemos medir solo un qubit

- Si medimos el primer qubit (el segundo es análogo):
 - Obtendremos 0 con probabilidad $|\alpha_{00}|^2 + |\alpha_{01}|^2$
 - En ese caso, el nuevo estado de $|\psi\rangle$ será

$$\frac{\alpha_{00} |00\rangle + \alpha_{01} |01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

- Obtendremos 1 con probabilidad $|\alpha_{10}|^2 + |\alpha_{11}|^2$
- En ese caso, el nuevo estado de $|\psi\rangle$ será

$$\frac{\alpha_{10} |10\rangle + \alpha_{11} |11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$$

Puertas cuánticas de dos qubits

- Un estado de dos qubits es

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

- Se representa mediante el vector columna

$$\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

- Podemos calcular productos escalares teniendo en cuenta que

$$\langle 00|00\rangle = \langle 01|01\rangle = \langle 10|10\rangle = \langle 11|11\rangle = 1$$

$$\langle 00|01\rangle = \langle 00|10\rangle = \langle 00|11\rangle = \cdots = \langle 11|00\rangle = 0$$

- Una puerta cuántica de dos qubits es una matriz unitaria U de tamaño 4×4

Productos tensoriales de puertas de un qubit

- Podemos obtener una puerta de dos qubits haciendo actuar dos puertas de un qubit, A y B , simultáneamente sobre cada uno de ellos
- En este caso, la matriz de la puerta de dos qubits es el producto tensorial $A \otimes B$
- Se verifica que

$$(A \otimes B)(|\psi_1\rangle \otimes |\psi_2\rangle) = (A|\psi_1\rangle) \otimes (B|\psi_2\rangle)$$

- Por supuesto, A o B podrían ser la identidad
- NO todas las puertas de dos qubits son productos tensoriales de puertas de un qubit

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix}$$

La puerta *CNOT*

- La puerta *CNOT* (controlled-NOT) viene definida por la matriz (unitaria)

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

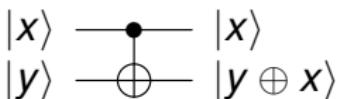
- Si el primer qubit es $|0\rangle$, no se hace nada. Si es $|1\rangle$, se invierte el segundo qubit (y el primero queda igual)
- Es decir:

$$|00\rangle \rightarrow |00\rangle \quad |01\rangle \rightarrow |01\rangle$$

$$|10\rangle \rightarrow |11\rangle \quad |11\rangle \rightarrow |10\rangle$$

La puerta *CNOT*

- Su acción con elementos $x, y \in \{0, 1\}$ es, por tanto:



- Es una puerta muy importante, puesto que nos permite:
 - Realizar entrelazamientos
 - Copiar información clásica, ya que:

$$|00\rangle \rightarrow |00\rangle$$

$$|10\rangle \rightarrow |11\rangle$$

Sistemas de n qubits

- Cada uno de los n qubits puede estar en los estados $|0\rangle$ y $|1\rangle$
- Así que para el conjunto de los n qubits tenemos 2^n posibilidades:

$$|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle$$

o simplemente

$$|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$$

- Un estado genérico del sistema será

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n - 1} |2^n - 1\rangle$$

donde los α_i son números complejos que cumplen

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

Medida de un estado de n qubits

- Supongamos que tenemos un estado genérico de n qubits

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

- Si medimos todos los qubits, obtendremos:
 - 0 con probabilidad $|\alpha_0|^2$ y el nuevo estado será $|0\dots 00\rangle$
 - 1 con probabilidad $|\alpha_1|^2$ y el nuevo estado será $|0\dots 01\rangle$
 - ...
 - $2^n - 1$ con probabilidad $|\alpha_{2^n-1}|^2$ y el nuevo estado será $|1\dots 11\rangle$
- Es una situación análoga a la que teníamos con un solo qubit, pero ahora con 2^n posibilidades

Medida de un qubit en un estado de n qubits

- Tenemos un estado

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

- Si medimos el qubit j -ésimo
 - Obtendremos 0 con probabilidad

$$\sum_{i \in I_0} |\alpha_i|^2$$

donde I_0 es el conjunto de números i cuyo j -ésimo bit es 0

- En ese caso, el nuevo estado de $|\psi\rangle$ será

$$\frac{\sum_{i \in I_0} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_0} |\alpha_i|^2}}$$

- El caso en el que se obtiene 1 es análogo

Puertas cuánticas de n qubits

- Un estado de n qubits es

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

- Se representa mediante el vector columna

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix}$$

- Para calcular productos escalares en notación de Dirac basta notar que

$$\langle i|j\rangle = \delta_{ij}$$

- Una puerta cuántica de dos qubits es una matriz unitaria U de tamaño $2^n \times 2^n$

Puertas universales en la computación cuántica

- El número de puertas cuánticas (incluso para un solo qubit) es infinito no numerable. Por tanto, ningún conjunto finito de puertas es universal en el sentido tradicional del término
- Lo que sí se puede conseguir son familias de puertas que **aproximan** cualquier puerta cuántica tanto como queramos

Teorema

Las puertas X, H, T y CNOT son universales para la computación cuántica

- Y podemos tener conjuntos de puertas infinitos en uno y dos qubits que son universales

Teorema

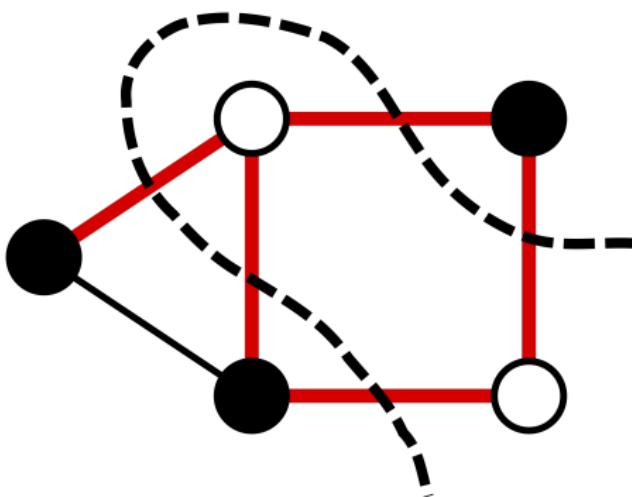
Las puertas de rotación de un qubit junto con la puerta CNOT son universales para la computación cuántica

Parte III

Optimización en ordenadores
cuánticos: Quantum Annealing

El problema del corte máximo

- Consideraremos el problema de dividir los vértices de un grafo en dos grupos maximizando los ejes cortados



- Es un problema NP-hard (si podemos resolverlo en tiempo polinomial, también podemos resolver en tiempo polinomial cualquier otro problema que esté en NP)

Planteando el problema del corte máximo con *spins*

- Identificamos cada vértice i del grafo con una variable Z_i que puede tomar valor 1 o -1
- Asignamos valor 1 a los vértices de un grupo y -1 a los del otro
- Entonces, si E es el conjunto de ejes, el problema se puede plantear como

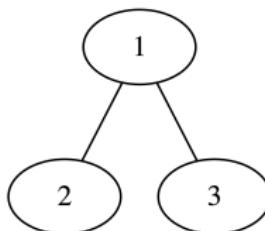
$$\text{Minimizar} \sum_{(i,j) \in E} Z_i Z_j$$

ya que vértices en distintos grupos aportan -1 a la suma y vértices del mismo grupo aportan 1

Ejemplo de corte máximo

- Para el grafo de la figura se trata de minimizar

$$H = Z_1Z_2 + Z_1Z_3$$



- Por inspección (o enumerando todas las posibilidades) se ve que las soluciones óptimas son 011 y 100

¿Y dónde metemos la computación cuántica en todo esto?

- Recordemos que la puerta Z tiene como matriz

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

y que el vector $|0\rangle$ tiene como coordenadas

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

- Entonces

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

- Podemos ver el anterior producto de matrices y vectores como

$$\langle 0 | Z | 0 \rangle = 1$$

¿Y dónde metemos la computación cuántica en todo esto?

- Análogamente

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Así que

$$\langle 1 | Z | 1 \rangle = (0 \quad 1) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$$

- Si tenemos más qubits, podemos utilizar productos tensoriales. Por ejemplo, denotando $Z_1 Z_2 = Z_1 \otimes Z_2$, tenemos

$$\langle 01 | Z_1 Z_2 | 01 \rangle = (\langle 0 | Z_1 | 0 \rangle) \cdot (\langle 1 | Z_2 | 1 \rangle) = 1 \cdot (-1) = -1$$

y, denotando $Z_1 Z_3 = Z_1 \otimes Z_3$, tenemos

$$\langle 101 | Z_1 Z_3 | 101 \rangle = (\langle 1 | Z_1 | 1 \rangle) \cdot (\langle 1 | Z_3 | 1 \rangle) = (-1) \cdot (-1) = 1$$

Volviendo al ejemplo de corte máximo

- Teníamos el problema de corte representado por

$$H = Z_1Z_2 + Z_1Z_3$$

- Podemos identificar un posible corte con $|011\rangle$ (tomar los vértices 2 y 3 y dejar fuera el 1) y evaluar su coste mediante

$$\begin{aligned}\langle 011 | H | 011 \rangle &= \langle 011 | (Z_1Z_2 + Z_1Z_3) | 011 \rangle \\ &= \langle 011 | Z_1Z_2 | 011 \rangle + \langle 011 | (Z_1Z_3) | 011 \rangle = -1 + (-1) = -2\end{aligned}$$

- Del mismo modo

$$\begin{aligned}\langle 010 | H | 010 \rangle &= \langle 010 | (Z_1Z_2 + Z_1Z_3) | 010 \rangle \\ &= \langle 010 | Z_1Z_2 | 010 \rangle + \langle 010 | (Z_1Z_3) | 010 \rangle = -1 + 1 = 0\end{aligned}$$

El maravilloso mundo de los hamiltonianos

- Entonces, lo que nos interesa es hallar un estado cuántico $|x\rangle$ de forma que

$$\langle x| H |x\rangle$$

sea mínimo, con $H = \sum_{(i,j) \in E} Z_i Z_j$ la función de coste del problema del corte máximo

- Se trata de un caso particular de un problema muy importante en física: hallar el estado de energía mínima (**ground state**) de un hamiltoniano
- Un hamiltoniano es una matriz H hermitiana ($H = H^\dagger$)
- Físicamente, puede representar fuerzas, potenciales... en la ecuación de Schrödinger
- La energía de un estado $|\psi\rangle$ es

$$\langle \psi | H | \psi \rangle$$

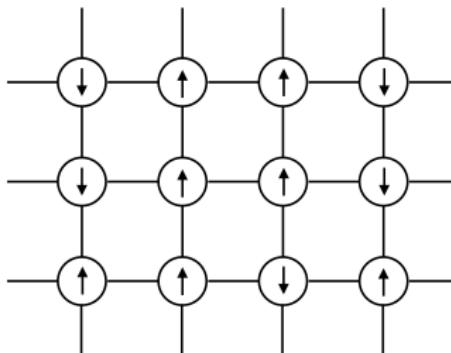
Ejemplo: el modelo de Ising

- Se tienen n partículas con spin, que interactúan entre sí con ciertas fuerzas de acoplamiento
- Su hamiltoniano es

$$H = \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j + \sum_{i=1}^n h_i Z_i$$

con J_{ij} y h_i coeficientes reales

- Queremos encontrar una asignación de valores de spins (1 o -1) que minimice la suma
- El problema general es NP-hard



Computación cuántica adiabática

- ¿Cómo obtener el *ground state* de H ?
- Una solución natural es aplicar el propio hamiltoniano H para llegar a la solución
- El **teorema adiabático** nos asegura que si comenzamos en el estado de mínima energía de un hamiltoniano y lo vamos variando lentamente, nos mantendremos siempre en el estado de mínima energía
- La idea de la computación cuántica adiabática es:
 - Comenzar en el estado de mínima energía de un hamiltoniano sencillo H_i
 - Evolucionar el sistema hacia el estado de mínima energía del hamiltoniano del problema H_f
 - Para ello se aplica el hamiltoniano dependiente del tiempo

$$H(t) = \left(1 - \frac{t}{T}\right)H_i + \frac{t}{T}H_f$$

durante tiempo T

Computación cuántica adiabática (2)

- Para garantizar la adiabaticidad, T debe crecer como el inverso del cuadrado del *spectral gap* de $H(t)$ (diferencia entre el primer y segundo nivel de energía)
- El spectral gap es **difícil** de calcular
- En la práctica, se usa el *quantum annealing*:
 - Se toma $H_i = - \sum_{i=1}^n X_i$ (con ground state $\sum_{x=0}^{2^n-1} |x\rangle$)
 - Como H_f se toma un hamiltoniano de Ising
 - Se deja evolucionar durante un tiempo T (no necesariamente adiabático)
 - Se mide para obtener una solución
 - Se repite un cierto número de veces y se devuelve la mejor solución obtenida
- Es la base de los ordenadores cuánticos de D-Wave

Los ordenadores cuánticos de D-Wave

- Son ordenadores de propósito específico: resolver el modelo de Ising
- Accesibles gratuitamente (1 minuto/mes) a través de <https://www.dwavesys.com/take-leap>



Parte IV

Optimización en ordenadores
cuánticos: QUBO y otros problemas

El problema del particionado de números

- Tenemos una lista de números (s_1, s_2, \dots, s_n)
- Queremos encontrar una partición de la lista en dos conjuntos de suma tan parecida como sea posible
- Es un problema NP-duro (aunque en casos prácticos se suele poder resolver eficientemente)
- Usamos variables Z_i que indican si el elemento i -ésimo pertenece a un conjunto (1) o al otro (-1)
- Podemos formular el problema como

$$\text{Minimizar} \left(\sum_{i=1}^n s_i Z_i \right)^2$$

y expandir y simplificar teniendo en cuenta que $Z_i^2 = 1$.

Un ejemplo de particionado de números

- Consideramos la lista $(2, 5, 4, 7)$
- Queremos minimizar

$$(2Z_1 + 5Z_2 + 4Z_3 + 7Z_4)^2$$

es decir

$$\begin{aligned} 4Z_1^2 + 20Z_1Z_2 + 16Z_1Z_3 + 28Z_1Z_4 + 25Z_2^2 \\ + 40Z_2Z_3 + 70Z_2Z_4 + 16Z_3^2 + 56Z_3Z_4 + 49Z_4^2 \end{aligned}$$

- Es equivalente a minimizar

$$20Z_1Z_2 + 16Z_1Z_3 + 28Z_1Z_4 + 40Z_2Z_3 + 70Z_2Z_4 + 56Z_3Z_4$$

QUBO: Quadratic Unconstrained Binary Optimization

- Los problemas QUBO se plantean como

$$\text{Minimizar} \sum_{1 \leq i \leq j \leq n}^n w_{ij} x_i x_j$$

donde cada x_i es una variable binaria y los w_{ij} son coeficientes reales

- También se puede plantear como minimizar

$$x^T Q x$$

con x el vector columna de elementos (x_i), Q una matriz simétrica tal que $Q_{ij} = Q_{ji} = \frac{1}{2}w_{ij}$ y $Q_{ii} = w_{ii}$

- Se puede reescribir como un modelo de Ising con la transformación

$$x_i = \frac{1 - Z_i}{2}$$

y volver a QUBO con

$$Z_i = 1 - 2x_i$$

Ejemplo: De QUBO a Ising

- Consideramos el problema de minimizar

$$3x_1^2 - 4x_1x_2 + 4x_2^2$$

con x_1 y x_2 variables binarias

- Hacemos el cambio

$$x_1 = \frac{1 - Z_1}{2}$$

$$x_2 = \frac{1 - Z_2}{2}$$

- Obtenemos

$$\frac{3Z_1^2}{4} - Z_1Z_2 - \frac{Z_1}{2} + Z_2^2 - Z_2 + \frac{3}{4}$$

- Es equivalente a minimizar

$$-Z_1Z_2 - \frac{Z_1}{2} - Z_2$$

Problemas enteros con restricciones lineales

- Las restricciones se añaden como penalizaciones en la función a minimizar
- Una restricción de igualdad

$$\sum_{i=1}^n a_i x_i = b$$

se añade como el término de penalización

$$M \left(\sum_{i=1}^n a_i x_i - b \right)^2$$

donde M es una constante suficientemente grande

- Podemos aprovechar que x_i^2 para simplificar

Problemas enteros con restricciones lineales (2)

- Una restricción de desigualdad

$$\sum_{i=1}^n a_i x_i \leq b$$

se transforma en una de igualdad añadiendo una variable de holgura entera $c \geq 0$

$$\sum_{i=1}^n a_i x_i + c = b$$

- Posteriormente, c (y las variables x_i , si son enteras) se transforma de acuerdo a

$$c = \sum_{k=0}^m c_k 2^k$$

donde c_i son variables binarias y m es suficientemente grande para poder expresar todos los valores válidos de c (el último valor podría ser menor que 2^m si hiciera falta)

Ejemplo: De problema con restricciones lineales a QUBO

- Minimizar

$$2xy - y - 4zy$$

sujeto a

$$0 \leq x, y, z \leq 1$$

$$x + 2y + 3z \leq 5$$

- Añadimos $0 \leq c \leq 5$ y reescribimos

$$x + 2y + 3z + c = 5$$

- Sustituimos $c = c_0 + 2c_1 + 2c_2$ e incorporamos la penalización. Ahora queremos minimizar

$$2xy - y - 4zy + 100(x + 2y + 3z + c_0 + 2c_1 + 2c_2 - 5)^2$$

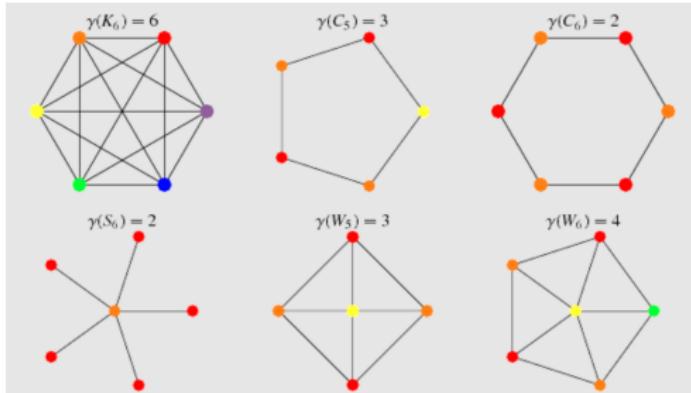
que es un problema QUBO (hemos elegido $M = 100$).

- También podríamos haber usado $c = c_0 + 2c_1 + 4c_2$

Ejercicios

- Formular usando QUBO los siguientes problemas:

- 1 Problema de la mochila:** Tenemos n objetos de pesos $p_i \geq 0$ y valores $v_i \geq 0$. Elegir un conjunto de objetos de valor total máximo sin sobrepasar un peso P dado.
- 2 Problema del viajante de comercio:** Tenemos n ciudades con distancias $d_{ij} \geq 0$. Encontrar un recorrido de todas las ciudades que minimice la distancia recorrida.
- 3 Coloreado de un grafo:** Dado un grafo, determinar si con k colores podemos asignar colores diferentes a todos los vértices adyacentes del grafo



Quantum Annealing y Machine Learning (1)

Published: 19 October 2017

Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar & Maria Spiropulu 

Nature 550, 375–379(2017) | Cite this article

2461 Accesses | 183 Altmetric | Metrics

Abstract

The discovery of Higgs-boson decays in a background of standard-model processes was assisted by machine learning methods^{1,2}. The classifiers used to separate signals such as these from background are trained using highly unerring but not completely perfect simulations of the physical processes involved, often resulting in incorrect labelling of background processes or signals (label noise) and systematic errors. Here we use quantum^{3,4,5,6} and classical^{7,8} annealing (probabilistic techniques for approximating the global maximum or minimum of a given function) to solve a Higgs-signal-versus-background machine learning optimization problem, mapped to a problem of finding the ground state of a corresponding Ising spin model. We build a set of weak classifiers based on the kinematic observables of the Higgs decay photons, which we then use to construct a strong classifier. This strong classifier is highly resilient against overtraining and against

Quantum Annealing y Machine Learning (2)

- Se consideran 36 clasificadores sencillos $c_i(x)$ que se combinan para formar un clasificador mejor

$$O(x) = \sum_i w_i c_i(x)$$

con $w_i \in \{0, 1\}$

- Se minimiza

$$\sum_x (y(x) - \sum_i w_i c_i(x))^2$$

que, con un parámetro de regularización λ , es equivalente a minimizar

$$\sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_i) w_i$$

where $C_{ij} = \sum_x c_i(x)c_j(x)$ and $C_i = \sum_x c_i(x)y(x)$.



OPEN

QUBO formulations for training machine learning models

Prasanna Date¹✉, Davis Arthur² & Lauren Pusey-Nazzaro³

Training machine learning models on classical computers is usually a time and compute intensive process. With Moore's law nearing its inevitable end and an ever-increasing demand for large-scale data analysis using machine learning, we must leverage non-conventional computing paradigms like quantum computing to train machine learning models efficiently. Adiabatic quantum computers can approximately solve NP-hard problems, such as the quadratic unconstrained binary optimization (QUBO), faster than classical computers. Since many machine learning problems are also NP-hard, we believe adiabatic quantum computers might be instrumental in training machine learning models efficiently in the post Moore's law era. In order to solve problems on adiabatic quantum computers, they must be formulated as QUBO problems, which is very challenging. In this paper, we formulate the training problems of three machine learning models—linear regression, support vector machine (SVM) and balanced k-means clustering—as QUBO problems, making them conducive to be trained on adiabatic quantum computers. We also analyze the computational complexities of our formulations and compare them to corresponding state-of-the-art classical approaches. We show that the time and space complexities of our formulations are better (in case of SVM and balanced k-means clustering) or equivalent (in case of linear regression) to their classical counterparts.

La formulación del problema es importante

Performance of Domain-Wall Encoding for Quantum Annealing

Jie Chen,¹ Tobias Stollenwerk,² and Nicholas Chancellor¹

¹*Department of Physics; Joint Quantum Centre (JQC) Durham-Newcastle
Durham University, South Road, Durham, DH1 3LE, UK*

²*German Aerospace Center (DLR), Linder Höhe, 51147 Cologne, Germany*

(Dated: December 2020)

In this paper we experimentally test the performance of the recently proposed domain-wall encoding of discrete variables from [Chancellor Quantum Sci. Technol. 4 045004] on Ising model flux qubit quantum annealers. We compare this encoding with the traditional one-hot methods and find that they outperform the one-hot encoding for three different problems at different sizes both of the problem and of the variables. From these results we conclude that the domain-wall encoding yields superior performance against a variety of metrics furthermore, we do not find a single metric by which one hot performs better. We even find that a 2000Q quantum annealer with a drastically less connected hardware graph but using the domain-wall encoding can outperform the next generation Advantage processor if that processor uses one-hot encoding.

I. INTRODUCTION

Quantum annealing is a subject of much recent interest, because of recent advances in both theory and experimental implementations. After the initial numerical studies which pointed to quantum annealing as a potential tool for optimization [1], focus was mostly on relatively simple closed systems in the adiabatic limit [2, 3]. However, a wide variety of advances have now taken place, for example better understanding of the role noise plays [4], more rapid quenches [5–8], and how to incorporate quantum annealing into hybrid protocols [9–12]. Experimentally this field is exciting because it allows

strength. An alternative approach to mapping is to use parity constraints [26–29], but we use minor embedding methods for this paper because they are more commonly used, and because quantum Monte-Carlo studies have suggested that this is a better method for the kind of devices we study here [30].

Since solving a QUBO is known to be NP-hard, all other optimization problems can be mapped to them with only a polynomial overhead. One particular mapping which is common is a discrete-to-binary mapping, where discrete variables with greater than two values are mapped to binary variables. The traditional way to do this is to use a kind of constraint known as a one-hot

Parte V

Optimización en ordenadores
cuánticos: QAOA

Quantum Approximate Optimization Algorithm

- El Quantum Approximate Optimization Algorithm (QAOA) fue propuesto por Farhi, Goldstone y Gutmann (2014) para obtener soluciones aproximadas del problema de minimizar (o maximizar) funciones de la forma

$$C(x) = \sum_a w_a C_a(x)$$

donde x es una cadena de n bits, los w_a son números reales y cada C_a es una función booleana

- El problema Max-Cut es de este tipo, con cada w_a igual a 1 y cada C_a de la forma

$$x_i \oplus x_j$$

MAX-SAT y weighted MAX-SAT

- El problema MAX-SAT es una generalización del problema de satisfabilidad (SAT)
- Dada una fórmula booleana en forma normal conjuntiva, se trata de determinar el máximo número de cláusulas que pueden ser satisfechas simultáneamente
- Por ejemplo, la fórmula

$$(x_0 \vee x_1) \wedge (\neg x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee \neg x_1)$$

no es satisfacible, pero podemos satisfacer tres cláusulas de las cuatro

- Si entendemos cada cláusula como una función que devuelve 0 (falso) o 1 (verdadero), podemos expresar MAX-SAT como maximizar

$$C(x) = \sum_a C_a(x)$$

con cada $C_a(x)$ una cláusula

- Añadiendo pesos $w_a \geq 0$, tenemos weighted MAX-SAT

Funciones booleanas y hamiltonianos

- Para cada función booleana C_a podemos encontrar H_a de la forma

$$a_0 I + \sum_i a_i Z_i + \sum_{i < j} a_{ij} Z_i Z_j + \sum_{i < j < k} a_{ijk} Z_i Z_j Z_k + \dots$$

tal que para cada x se verifica que $C_a(x) = \langle x | H_a | x \rangle$

- Así, minimizar $C(x)$ es equivalente a encontrar el estado de mínima energía de

$$H_f = \sum_a w_a H_a$$

ya que H_f es diagonal y $\langle x | H_f | x \rangle = C(x)$.

Funciones booleanas y hamiltonianos

- Para pasar una fórmula C_a booleana a hamiltoniano podemos
 - Transformar la fórmula para que sólo tenga \neg y \wedge
 - Transformar $\neg f$ en $(1 - f)$
 - Transformar $f \wedge g$ en $f \cdot g$
 - Simplificar potencias de x_i (sabemos que $x_i^2 = x_i$)
 - Hacer los cambios $x_i = \frac{1}{2}(I - Z_i)$
- Podrían aparecer hamiltonianos con términos de grado 3 o superior

x	$\frac{1}{2}I - \frac{1}{2}Z$	\bar{x}	$\frac{1}{2}I + \frac{1}{2}Z$
$x_1 \oplus x_2$	$\frac{1}{2}I - \frac{1}{2}Z_1 Z_2$	$\bigoplus_{j=1}^k x_j$	$\frac{1}{2}I - \frac{1}{2}Z_1 Z_2 \dots Z_k$
$x_1 \wedge x_2$	$\frac{1}{4}I - \frac{1}{4}(Z_1 + Z_2 - Z_1 Z_2)$	$\bigwedge_{j=1}^k x_j$	$\frac{1}{2^k} \prod_j (I - Z_j)$
$x_1 \vee x_2$	$\frac{3}{4}I - \frac{1}{4}(Z_1 + Z_2 + Z_1 Z_2)$	$\bigvee_{j=1}^k x_j$	$I - \frac{1}{2^k} \prod_j (I + Z_j)$
$\overline{x_1 x_2}$	$\frac{3}{4}I + \frac{1}{4}(Z_1 + Z_2 - Z_1 Z_2)$	$x_1 \Rightarrow x_2$	$\frac{3}{4}I + \frac{1}{4}(Z_1 - Z_2 + Z_1 Z_2)$

Los estados parametrizados en el QAOA

- QAOA se puede ver como una adaptación de la computación adiabática a los circuitos cuánticos
- Recordemos que el hamiltoniano adiabático era

$$H(t) = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$$

with $H_i = -\sum_{j=1}^n X_j$

- Como aproximación de la evolución del sistema, consideramos estados parametrizados o variacionales de la forma

$$|\beta, \gamma\rangle = e^{-i\beta_p H_i} e^{-i\gamma_p H_f} \dots e^{-i\beta_2 H_i} e^{-i\gamma_2 H_f} e^{-i\beta_1 H_i} e^{-i\gamma_1 H_f} |s\rangle$$

donde $p \geq 1$ y

$$|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$$

Optimización con QAOA

- QAOA es un método híbrido en el que se usan tanto un ordenador clásico como uno cuántico
- Los pasos son:
 - 1 Elegir un valor para p y unos ángulos iniciales β, γ
 - 2 Preparar el estado $|\beta, \gamma\rangle$
 - 3 Estimar la energía $E(\beta, \gamma) = \langle\beta, \gamma| H_f |\beta, \gamma\rangle$ de $|\beta, \gamma\rangle$
 - 4 Si se cumplen los criterios de parada, terminar. Si no, variar β y γ para minimizar $E(\beta, \gamma)$ y volver a 2
- El paso 2 se realiza en el ordenador cuántico. El resto, en uno clásico.

Cómo preparar el estado $|\beta, \gamma\rangle$

- El estado $|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$ se puede preparar con puertas de Hadamard
- Cada $e^{-i\beta_k X_j}$ es una rotación $R_X(2\beta_k)$ o equivalentemente



- Para implementar $e^{-i\gamma_k H_f}$ sólo necesitamos considerar casos de la forma

$$e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$$

porque

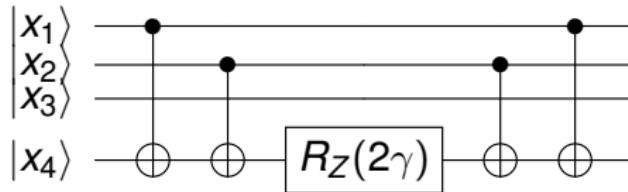
- Todos los términos de la forma $Z_{i_1} \cdots Z_{i_j}$ commutan
- Los pesos en $\sum_a w_a H_a$ se “absorben” en los ángulos γ

Implementando $e^{-i\gamma_k Z_{i_1} \dots Z_{i_j}}$

- Nótese que $e^{-i\gamma_k Z_{i_1} \dots Z_{i_j}}$ es diagonal en la base computacional
- De hecho, para una cadena binaria $x = x_1 \dots x_n$ su acción en $|x\rangle$ es
 - $|x\rangle \rightarrow e^{-i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \dots \oplus x_{i_j} = 0 \pmod{2}$
 - $|x\rangle \rightarrow e^{i\gamma_k} |x\rangle$ if $x_{i_1} \oplus \dots \oplus x_{i_j} = 1 \pmod{2}$
- Como se ve, es muy similar a una rotación R_Z
- Entonces, podemos:
 - Computar la paridad de $x_{i_1} \oplus \dots \oplus x_{i_j}$ con puertas CNOT
 - Aplicar $R_Z(2\gamma_k)$ en el qubit sobre el que hemos computado la paridad
 - Descomputar la paridad

Un ejemplo

- Imaginemos que estamos trabajando con 4 qubits y queremos implementar $e^{-i\gamma Z_1 Z_2 Z_4}$
- Podemos usar el siguiente circuito:



Estimando la energía

- Estimar la energía en el QAOA es muy sencillo
- Repetimos el siguiente proceso un número fijo de veces
 - 1 Preparamos el estado $|\beta, \gamma\rangle$
 - 2 Lo medimos para obtener un resultado x
 - 3 Calculamos $C(x)$y obtenemos la media de los resultados
- Este proceso funciona porque como

$$|\beta, \gamma\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$$

entonces

$$\langle \beta, \gamma | H_f | \beta, \gamma \rangle = \sum_{x \in \{0,1\}^n} |a_x|^2 C(x)$$

- También es interesante guardar la x con el valor de $C(x)$ mínimo entre las que hemos medido

Algunos comentarios sobre el QAOA

- Para que el proceso sea eficiente, H_f debe tener un número de términos $e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$ que sea polinomial en el número de qubits n y en el número m de términos $C_a(x)$ de $C(x)$
- Si un término $C_a(x)$ sólo involucra k bits, su traducción H_a involucrará sólo k matrices Pauli Z_i
- Por tanto, si p es constante e independiente de n y m , y todas los términos involucran a lo sumo k bits (constante también independiente de n y m) entonces el número de puertas será polinomial en n y m
- Es el caso, por ejemplo, de Max-Cut y de Max 3-SAT
- Cuando $p \rightarrow \infty$, el estado de mínima energía de $|\beta, \gamma\rangle$ tiende al estado de mínima energía de H_f
- En algunos casos, se han obtenido resultados interesantes incluso para p
- La elección del optimizador clásico también es importante

Parte VI

Variational Quantum Eigensolver

VQE: Variational Quantum Eigensolver

- QAOA puede verse como un caso particular de un algoritmo más general: el Variational Quantum Eigensolver (VQE)
- Ahora, tenemos un hamiltoniano general H_f (con un número polinomial de términos) y queremos aproximar su estado de mínima energía
- En lugar del estado parametrizado $|\beta, \gamma\rangle$ del QAOA usaremos
 - Un estado inicial $|\psi\rangle$ fácil de preparar (podría ser simplemente $|0\rangle$)
 - Una transformación unitaria parametrizada $U(\theta)$ que se suele llamar **forma variacional**
- Con ellos, creamos un ansatz

$$|\psi(\theta)\rangle = U(\theta) |\psi\rangle$$

y tratamos de minimizar su energía con respecto a H_f variando los parámetros

El principio variacional

- Como H_f es una matriz hermitiana, tiene autovalores reales λ_i y una base asociada de autovectores ortonormales $|\phi_i\rangle$
- Así, podemos escribir $|\psi(\theta)\rangle$ como una combinación lineal

$$|\psi(\theta)\rangle = \sum_i \alpha_i |\phi_i\rangle$$

- La energía de $|\psi(\theta)\rangle$ es

$$\langle \psi(\theta) | H_f | \psi(\theta) \rangle = \sum_i |\alpha_i|^2 \lambda_i$$

- Si λ_{min} es el menor de esos autovalores, entonces

$$\min_{\theta} \langle \psi(\theta) | H_f | \psi(\theta) \rangle \geq \lambda_{min}$$

Aproximando el estado de mínima energía con VQE

- VQE es también un algoritmo híbrido en el que se usa el ordenador cuántico en colaboración con un ordenador clásico
- Los pasos son:
 - ① Elegir un estado inicial $|\psi\rangle$, una forma variacional $U(\theta)$ y un vector de parámetros inicial θ
 - ② Preparar el estado $|\psi(\theta)\rangle = U(\theta) |\psi\rangle$
 - ③ Estimar la energía $E(\theta) = \langle\psi(\theta)| H_f |\psi(\theta)\rangle$ de $|\psi(\theta)\rangle$
 - ④ Si se cumplen los criterios de parada, terminar. Si no, variar θ para minimizar $E(\theta)$ y volver a 2
- El paso 2 se realiza en el ordenador cuántico. El resto, en uno clásico.

Estimando la energía de un estado

- El hamiltoniano siempre puede ser expresado como una combinación lineal de productos tensoriales de Paulis
- Por ejemplo

$$H_f = \frac{1}{4} Z_1 Z_3 - 3 X_1 Y_3 Z_4$$

- Dado $|\psi\rangle$, podemos usar la linearidad y evaluar

$$\langle\psi| H_f |\psi\rangle = \frac{1}{4} \langle\psi| Z_1 Z_3 |\psi\rangle - 3 \langle\psi| X_1 Y_3 Z_4 |\psi\rangle$$

- Para estimar $\langle\psi| Z_1 Z_3 |\psi\rangle$ podemos medir $|\psi\rangle$ en la base computacional y promediar las energías de los resultados (que serán 1 o -1 para cada resultado individual).

Estimando la energía de un estado (2)

- Para estimar $\langle \psi | X_1 Y_3 Z_4 | \psi \rangle$ podemos tener en cuenta que

$$X = HZH$$

and

$$Y = SHZHS^\dagger$$

- Entonces $\langle \psi | X_1 Y_3 Z_4 | \psi \rangle$ es igual

$$\langle \psi | (H \otimes I \otimes SH \otimes I) Z_1 Z_3 Z_4 (H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle$$

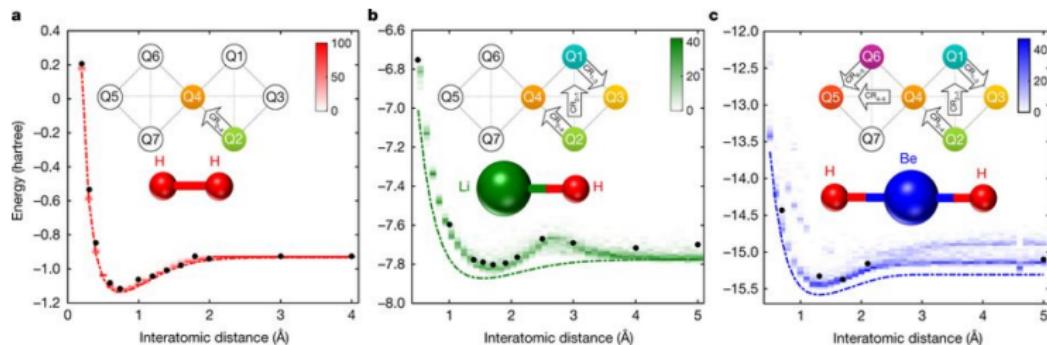
- Por tanto, podemos medir la energía en $Z_1 Z_3 Z_4$ de $(H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle$ porque

$$\left((H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle \right)^\dagger = \langle \psi | (H \otimes I \otimes SH \otimes I)$$

- Esto es equivalente a una medida en una base diferente

Estudiando moléculas con VQE

- VQE ha sido utilizado para estudiar los estados de energía mínima de varias moléculas
- El hamiltoniano químico (fermiónico) tiene que traducirse en un hamiltoniano de qubits (Jordan-Wigner, Bravyi-Kitaev...)
- Se usa información del problema para:
 - El estado inicial (estado del vacío $|0\rangle$, Hartree-Fock...)
 - La forma variacional (Unitary Coupled-Cluster Single and Double excitations...)



Encontrando estados excitados

- También podemos usar VQE para encontrar estados excitados (vectores propios que no son de mínima energía)
- Una vez que tenemos el estado de mínima energía $|\psi_0\rangle = U(\theta_0) |\psi\rangle$, consideramos el hamiltoniano

$$H'_f = H_f + C |\psi_0\rangle \langle \psi_0|$$

- Así, tenemos que $\langle \varphi | H'_f | \varphi \rangle$ es

$$\langle \varphi | H_f | \varphi \rangle + C \langle \varphi | \psi_0 \rangle \langle \psi_0 | \varphi \rangle = \langle \varphi | H_f | \varphi \rangle + C |\langle \psi_0 | \varphi \rangle|^2$$

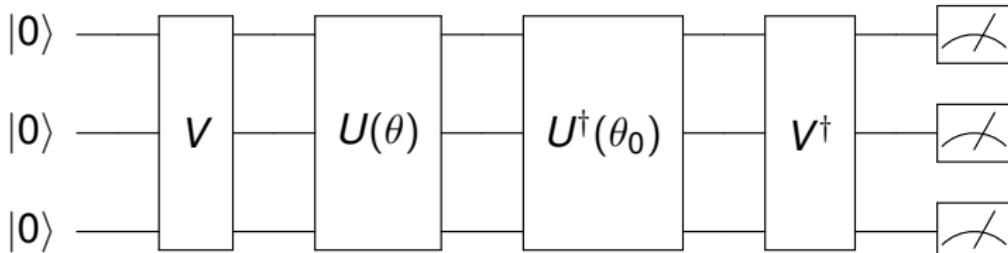
- Si C es mayor que la diferencia entre la energía mínima y el siguiente nivel de energía de H_f , entonces $|\psi_0\rangle$ ya no es estado de mínima energía de H'_f

Calculando productos escalares de estados parametrizados

- Para calcular el producto escalar en la nueva expresión de la energía podemos tener en cuenta que $|\psi_0\rangle = U(\theta_0) |\psi\rangle$ y que el nuevo estado será de la forma $|\varphi\rangle = U(\theta) |\psi\rangle$ para algún θ
- Entonces, es fácil estimar $|\langle\psi_0|\varphi\rangle|^2$ ejecutando el circuito de la figura y calculando la frecuencia relativa de $|0\rangle$ porque

$$|\langle\psi_0|\varphi\rangle|^2 = |\langle 0| V^\dagger U(\theta_0)^\dagger U(\theta) V |0\rangle|^2$$

donde V es una unitaria tal que $V|0\rangle = |\psi\rangle$



Una aplicación del VQE en física de altas energías

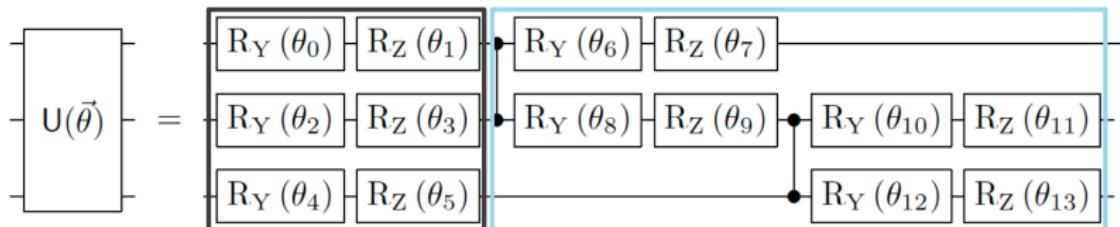
- Trabajo de Li, Macridin, Spentzouris - Fermilab (2019)
- Hamiltoniano de Rabi: sistema de dos niveles (TLS) acoplado a un modo fotónico

$$H = \omega a a^\dagger + \frac{\Omega}{2} Z + g(a^\dagger + a) X$$

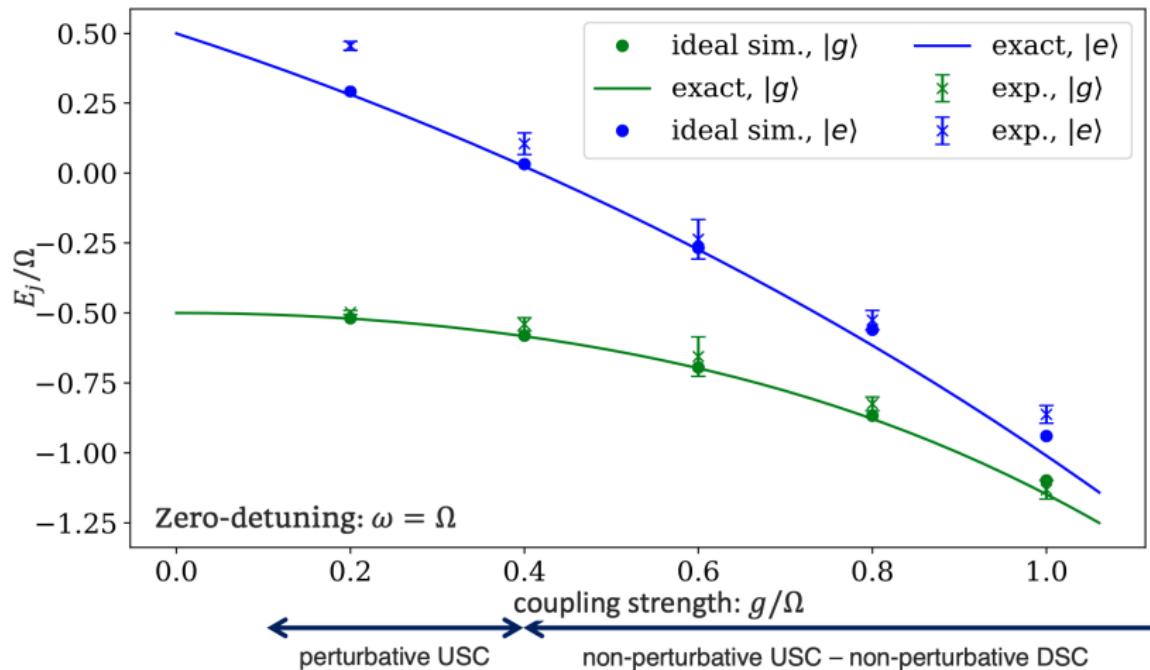
- Codificación en binario en la base: el modo fotónico se trunca a un máximo de 3 fotones

$$H = \omega Z_0 + \frac{\omega}{2} Z_1 + \frac{\Omega}{2} Z_2 + g \sqrt{Z_0 + 2} X_1 X_2$$

$$+ \frac{g}{\sqrt{2}} X_0 X_1 X_2 + Y_0 Y_1 X_2 + \frac{3\omega}{2}$$



Resultados en simulación y en ordenadores de Rigetti



Buscando ansatzes adaptados al hardware

npj Quantum Information

www.nature.com/npjqi

ARTICLE

OPEN



Gate-free state preparation for fast variational quantum eigensolver simulations

Oinam Romesh Meitei¹, Bryan T. Gard¹, George S. Barron¹, David P. Pappas^{3,4}, Sophia E. Economou^{1,2}, Edwin Barnes² and Nicholas J. Mayhall¹ 

The variational quantum eigensolver is currently the flagship algorithm for solving electronic structure problems on near-term quantum computers. The algorithm involves implementing a sequence of parameterized gates on quantum hardware to generate a target quantum state, and then measuring the molecular energy. Due to finite coherence times and gate errors, the number of gates that can be implemented remains limited. In this work, we propose an alternative algorithm where device-level pulse shapes are variationally optimized for the state preparation rather than using an abstract-level quantum circuit. In doing so, the coherence time required for the state preparation is drastically reduced. We numerically demonstrate this by directly optimizing pulse shapes which accurately model the dissociation of H₂ and HeH⁺, and we compute the ground state energy for LiH with four transmons where we see reductions in state preparation times of roughly three orders of magnitude compared to gate-based strategies.

npj Quantum Information (2021)7:155; <https://doi.org/10.1038/s41534-021-00493-0>

INTRODUCTION

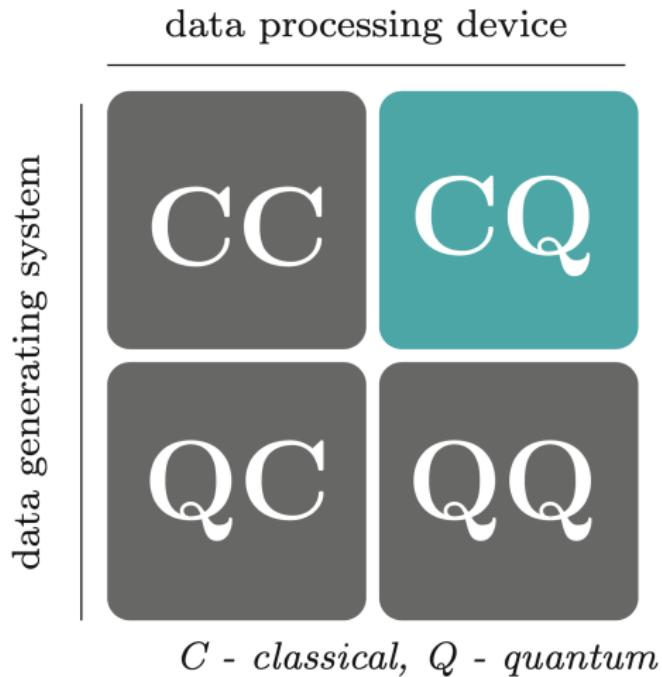
Molecular modeling stands in the juncture of key advances in many important fields including and not limited to energy storage, material design, and drug discovery. For more than half

This ability to perform a one-to-one mapping between vectors in the Hilbert space containing the molecule's electronic wavefunction and those in the state space accessible to a QPU means that with enough control over the QPU it should be possible to take

Parte VII

Quantum Machine Learning

De qué hablamos cuando hablamos de Quantum Machine Learning

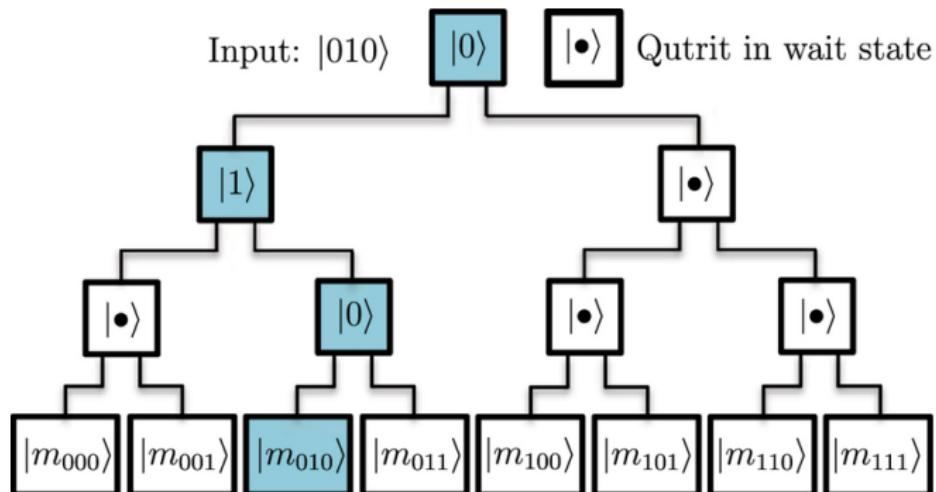


QBLAS: The Quantum Basic Linear Algebra Subroutines

- Varios algoritmos de Quantum Machine Learning (QML) se basan en el speedup exponencial de
 - La transformada cuántica de Fourier
 - La estimación cuántica de fase
 - El algoritmo HHL
- Llamamos a estos métodos Quantum Basic Linear Algebra Subroutines (QBLAS)
- Otras subrutinas usadas en QML incluyen la amplificación de la amplitud y el quantum annealing
- Cargar el **input**, leer el **output**, y el gran tamaño de los circuitos pueden suponer un problema

QRAM: El elefante en la habitación

- Una Quantum Random Access Memory (QRAM) debería permitir cargar datos en superposición
- Se han propuesto algunas arquitecturas (por ejemplo, la “bucket brigade”) pero se necesita más investigación
- La carga de datos puede ser un cuello de botella para muchos algoritmos de QML



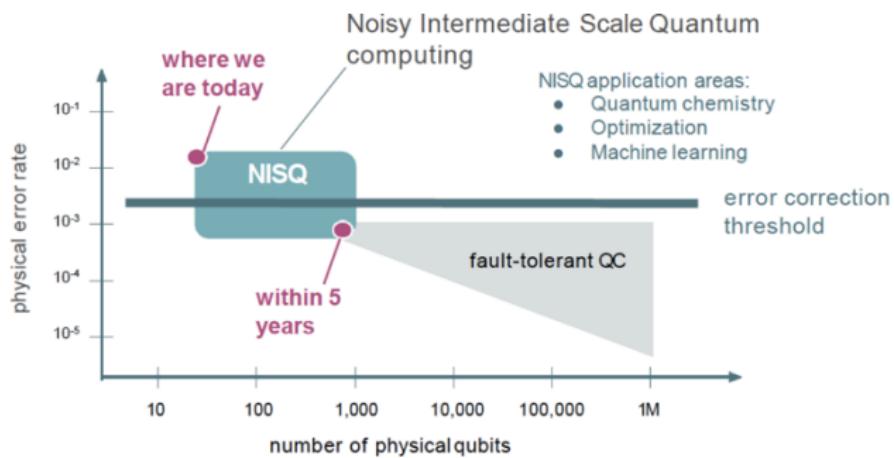
QML transductiva y speedups

Method	Speedup	Amplitude amplification	HHL	Adiabatic	qRAM
Bayesian inference ^{106,107}	$O(\sqrt{N})$	Yes	Yes	No	No
Online perceptron ¹⁰⁸	$O(\sqrt{N})$	Yes	No	No	Optional
Least-squares fitting ⁹	$O(\log N)^*$	Yes	Yes	No	Yes
Classical Boltzmann machine ²⁰	$O(\sqrt{N})$	Yes/No	Optional/ No	No/Yes	Optional
Quantum Boltzmann machine ^{22,61}	$O(\log N)^*$	Optional/No	No	No/Yes	No
Quantum PCA ¹¹	$O(\log N)^*$	No	Yes	No	Optional
Quantum support vector machine ¹³	$O(\log N)^*$	No	Yes	No	Yes
Quantum reinforcement learning ³⁰	$O(\sqrt{N})$	Yes	No	No	No

*There exist important caveats that can limit the applicability of the method⁵¹.

QML en los tiempos del NISQ

- Los ordenadores de la era NISQ (Noisy Intermediate-Scale Quantum) son
 - Ruidosos (sin tolerancia a fallos)
 - De número limitado de qubits (50-100)
 - Sin conexión total entre sus qubits
- A pesar de ello, pueden llegar a ser útiles para hacer QML

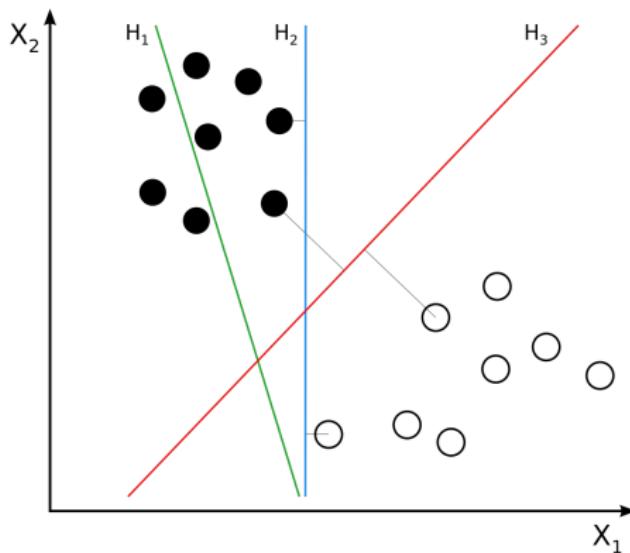


Parte VIII

Quantum Support Vector Machines

Support Vector Machines

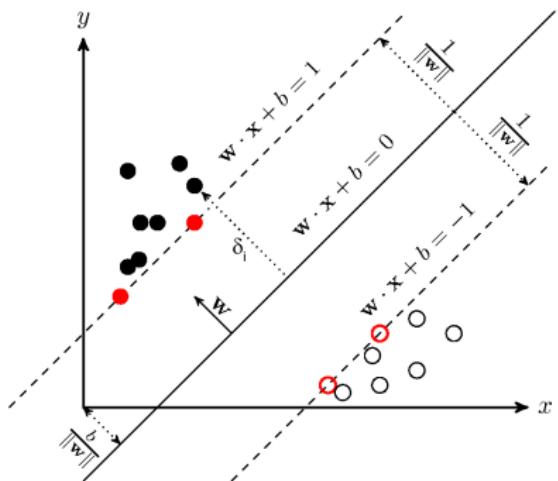
- Las Support Vector Machines (SVM) son un algoritmo (clásico) de clasificación muy popular
- La idea en la que se basan es encontrar un hiperplano separador con el mayor margen posible



Encontrando el hiperplano

- Tenemos puntos de entrenamiento (x_i, y_i) donde los x_i son vectores de números reales y $y_i \in \{1, -1\}$
- El problema de encontrar el hiperplano separador con el mayor margen se puede formular como

$$\text{Minimizar } \frac{1}{2} \|w\|^2 \text{ sujeto a } y_i(w \cdot x_i + b) \geq 1$$



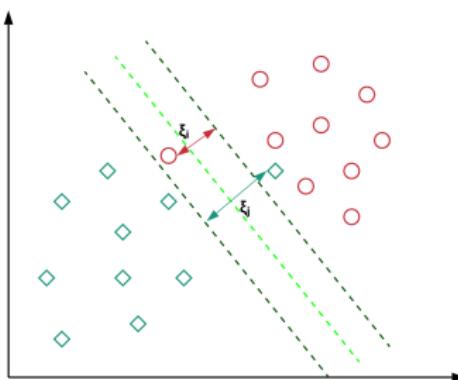
El caso del “margen blando”

- En el caso del “margen blando”, introducimos un hiperparámetro $C \geq 0$ y reformulamos el problema como

$$\text{Minimizar } \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

sujeto a

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$



Formulación dual del SVM

- Una formulación equivalente del problema de optimización del SVM es esta forma dual

$$\text{Maximizar} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$$

sujeto a

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- De los valores α_i podemos recuperar b y w . De hecho

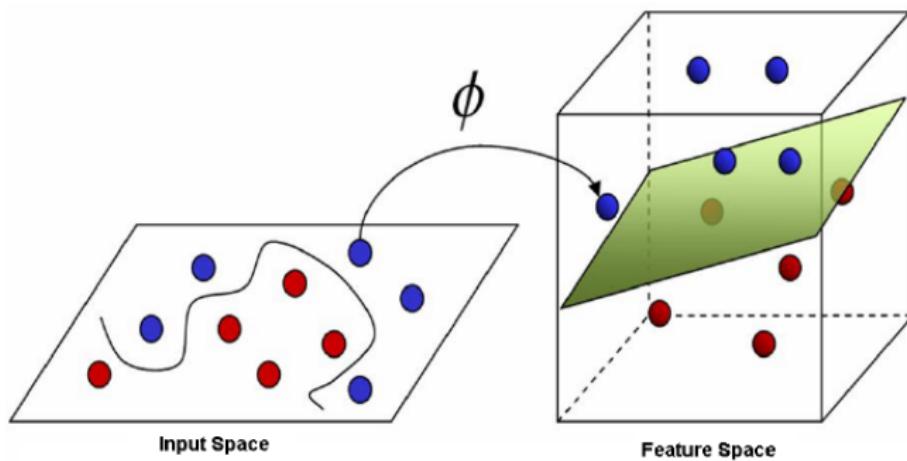
$$w = \sum_i \alpha_i y_i x_i$$

y para clasificar un punto x podemos calcular

$$w \cdot x + b = \sum_i \alpha_i y_i (x_i \cdot x) + b$$

Separación no lineal

- Una técnica muy utilizada para mejorar la clasificación con SVM es trasladar los puntos x_i a un espacio de dimensión superior usando una *feature map* $\phi(x_i)$



El truco del kernel

- Podemos incorporar fácilmente la feature map en nuestra formulación dual del SVM

$$\text{Maximizar} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j (\phi(x_i) \cdot \phi(x_j))$$

sujeto a

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- De nuevo, podemos obtener w como

$$w = \sum_i \alpha_i y_i \phi(x_i)$$

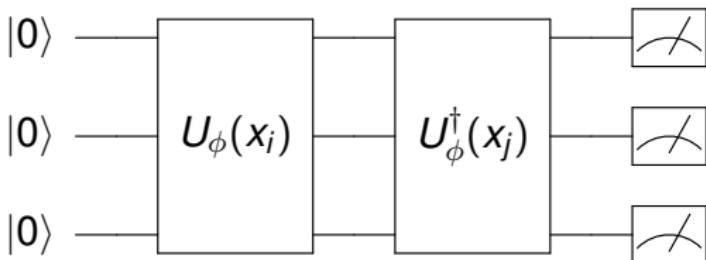
y para clasificar un punto x sólo necesitamos calcular

$$w \cdot x + b = \sum_i \alpha_i y_i (\phi(x_i) \cdot \phi(x)) + b$$

- La función $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ es llamada kernel

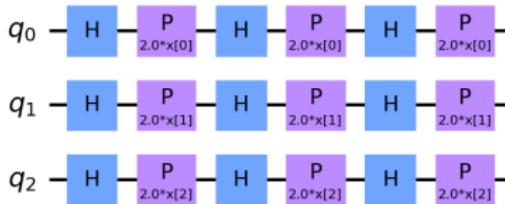
Calculando kernels con ordenadores cuánticos

- En 2019, Havlíček, Cároles, Temme et al. propusieron la idea de usar ordenadores cuánticos para estimar kernels
- Cada punto x_i se traslada a un espacio de Hilbert con un circuito variacional $U_\phi(x_i)$ such that $U_\phi(x_i) |0\rangle = |\phi(x_i)\rangle$
- Podemos estimar $|\langle\phi(x_j)|\phi(x_i)\rangle|^2$ ejecutando el circuito de la figura y calculando la frecuencia relativa de $|0\rangle$
- La resolución del problema de optimización se puede realizar en un ordenador clásico

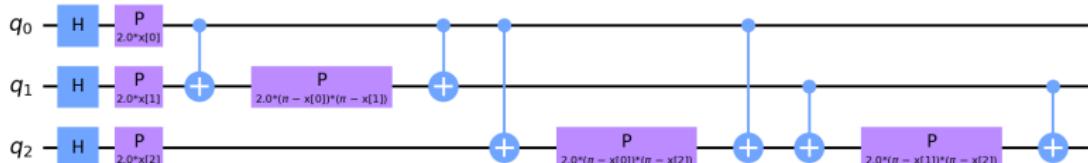


Ejemplos de feature maps

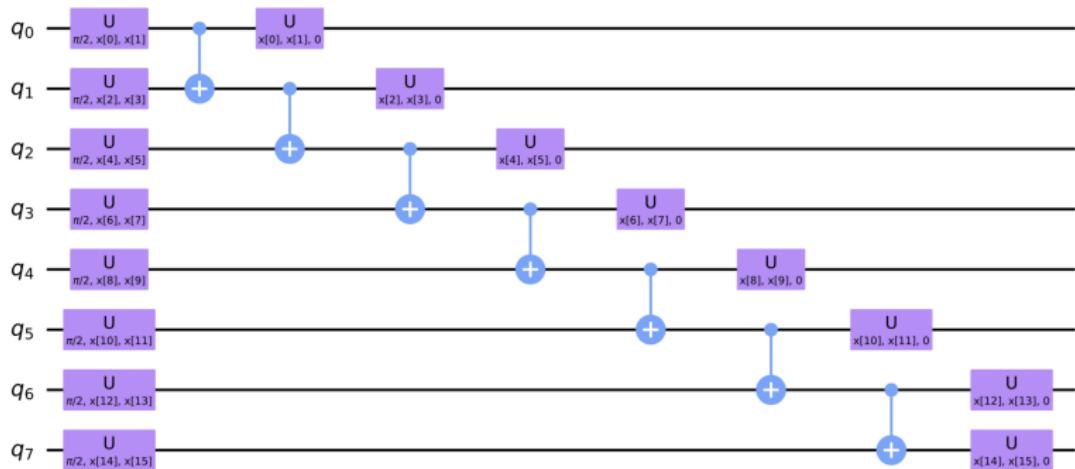
- Codificación en amplitud:
 - Partimos de un vector de números reales (x_1, x_2, \dots, x_m)
 - Puede ser necesario normalizarlo y “rellenarlo” hasta tamaño $2^n \geq m$
 - La codificación se hace mediante un circuito polinomial en m que prepara $\sum_{i=1}^{2^n} x_i |i\rangle$
- Z-feature map (codificación en ángulo)



- ZZ-feature map

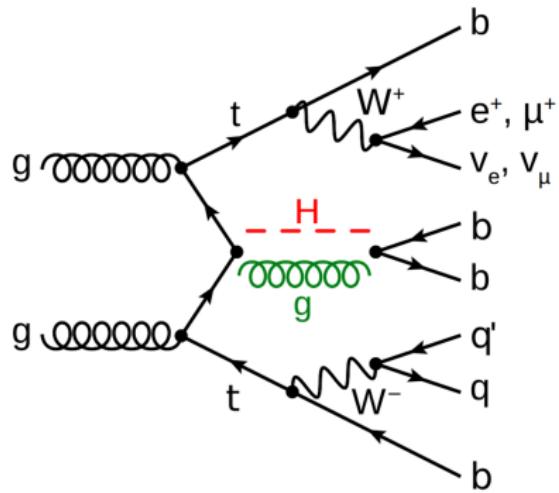


Otro ejemplo de feature map



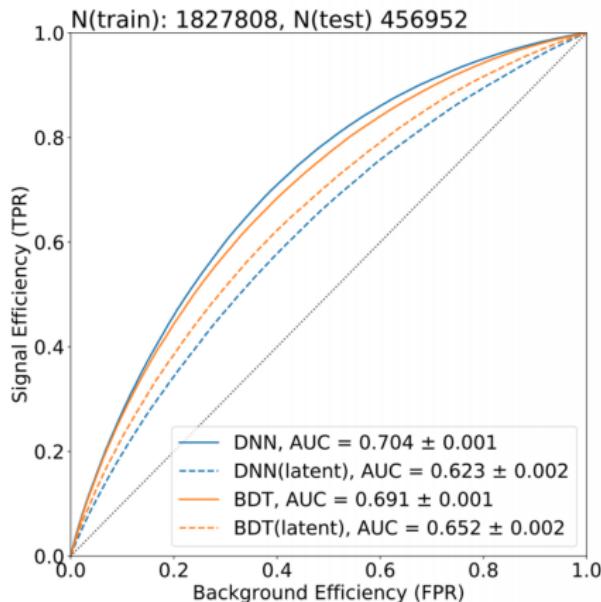
Un problema de clasificación en física de altas energías

- Producción del bosón de Higgs
- 67 variables físicas



Resultados con métodos de machine learning clásico

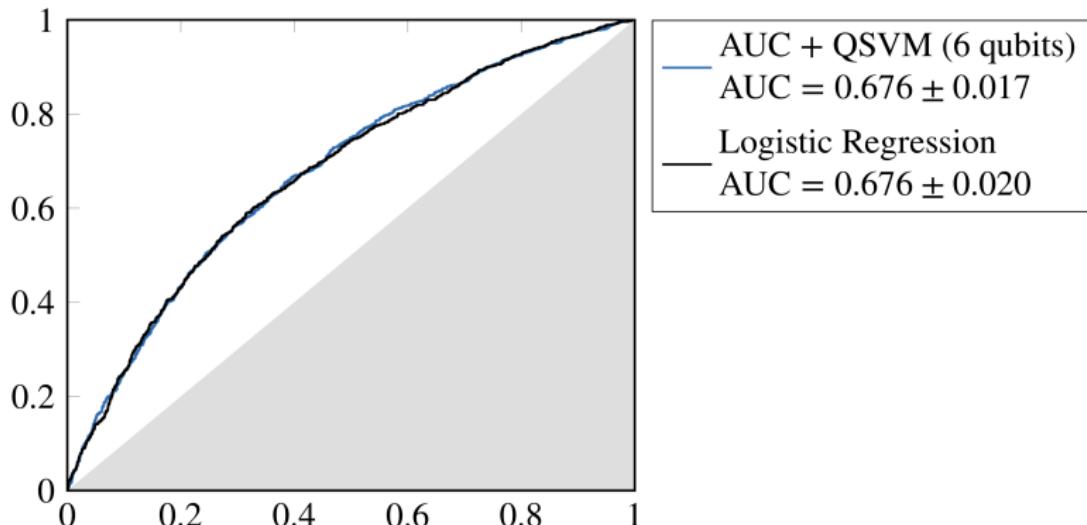
- Boosted decision trees (BDT) y deep neural networks (DNN)
- Entrenamiento con las 67 variables y con técnicas de reducción de dimensionalidad (autoencoders)
- Conjuntos de entrenamiento grandes



Resultados con QSVMs

- 64 variables seleccionadas con AUC
- Forma variacional: amplitude encoding

$N_{\text{train}} = 576$



Higgs analysis with quantum classifiers

Vasilis Belis^{1,*}, Samuel González-Castillo², Christina Reissel¹, Sofía Vallecorsa³, Elías F. Combarro⁴, Günther Dissertori¹, and Florentin Reiter⁵

¹Institute of Particle Physics and Astrophysics, ETH Zürich, Zürich, Switzerland

²Faculty of Sciences, University of Oviedo, Oviedo, Spain

³CERN, 1, Esplanade des Particules, Geneva, CH 1211

⁴Department of Computer Science, University of Oviedo, Oviedo, Spain

⁵Institute for Quantum Electronics, ETH Zürich, Zürich, Switzerland

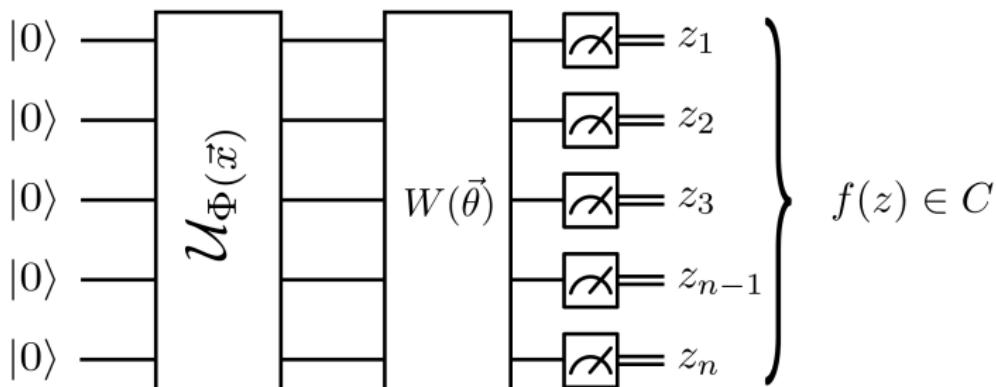
Abstract. We have developed two quantum classifier models for the $t\bar{t}H(b\bar{b})$ classification problem, both of which fall into the category of hybrid quantum-classical algorithms for Noisy Intermediate Scale Quantum devices (NISQ). Our results, along with other studies, serve as a proof of concept that Quantum Machine Learning (QML) methods can have similar or better performance, in specific cases of low number of training samples, with respect to conventional machine learning (ML) methods even with a limited number of qubits available in current hardware. To utilise algorithms with a low number of qubits — to accommodate for limitations in both simulation hardware and real quantum hardware — we investigated different feature reduction methods. Their impact on the performance of both the classical and quantum models was assessed. We addressed different implementations of two QML models, representative of the two main approaches to supervised quantum machine learning today: a Quantum Support Vector Machine (QSVM), a kernel-based method, and a Variational Quantum Circuit (VQC), a variational approach.

Parte IX

Redes neuronales cuánticas

¿Qué es una red neuronal cuántica?

- Las redes neuronales cuánticas (QNNs) son circuitos cuánticos parametrizados que se pueden “entrenar” con datos y ser utilizadas en tareas de clasificación, regresión...
- La arquitectura más habitual es la que se muestra en la figura: un “feature map”, que traslada los datos de entrada a un espacio de Hilbert, seguido de una forma variacional con parámetros libres que se entrena



Entrenando y clasificando con una QNN

- Una QNN prepara un estado $|\psi(x, \theta)\rangle$ que depende de los datos de entrada x y de los parámetros θ
- Medimos el estado y estimamos un valor esperado sobre un hamiltoniano, por ejemplo

$$f(x, \theta) = \langle \psi(x, \theta) | Z_1 \cdots Z_n | \psi(x, \theta) \rangle$$

- Para cada ejemplo de entrenamiento x_i tenemos una clase y_i
- Elegimos una función de pérdida L y buscamos θ que minimice

$$\sum_i L(y_i, f(x, \theta))$$

- Una vez encontrado θ_{min} podemos predecir la clase de x usando $f(x, \theta_{min})$

Gradientes y la regla del desplazamiento de parámetros

- Para obtener θ_{min} , podemos usar un minimizador clásico
- Si necesitamos calcular gradientes de f podemos usar la regla de desplazamiento de parámetros
- Supongamos que

$$U(\theta) = e^{-i\theta H}$$

con H una matriz hermitiana de autovalores $\pm r$ (r real)

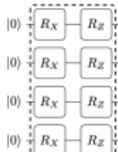
- Este es el caso, por ejemplo, si U es una rotación de un qubit
- Entonces, se verifica que

$$\frac{\partial f(x, \theta)}{\partial \theta} = r \cdot [f(x, \theta + s) - f(x, \theta - s)]$$

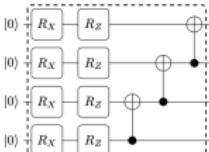
donde $s = \frac{\pi}{4r}$

- Simplemente necesitamos dos evaluaciones adicionales del mismo circuito con un desplazamiento de parámetros

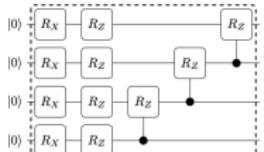
Cómo elegir formas variacionales



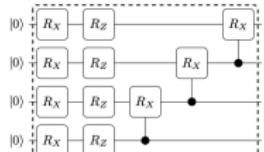
Circuit 1



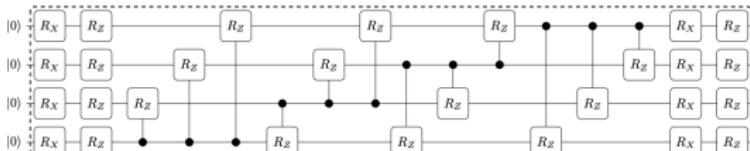
Circuit 2



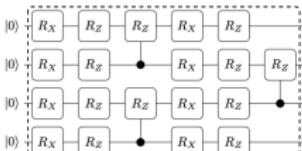
Circuit 3



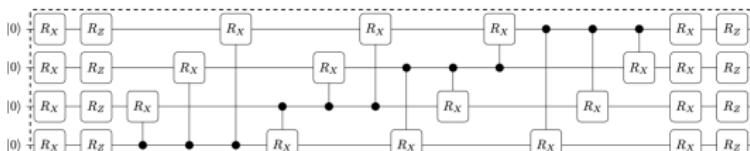
Circuit 4



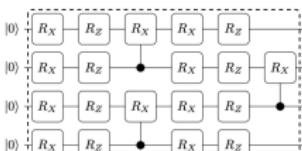
Circuit 5



Circuit 7



Circuit 6



Circuit 8

Figura: Sim, Johnson, Aspuru-Guzik. Adv. Quantum Tech. 2(12) (2019)

La capacidad de las redes neuronales cuánticas

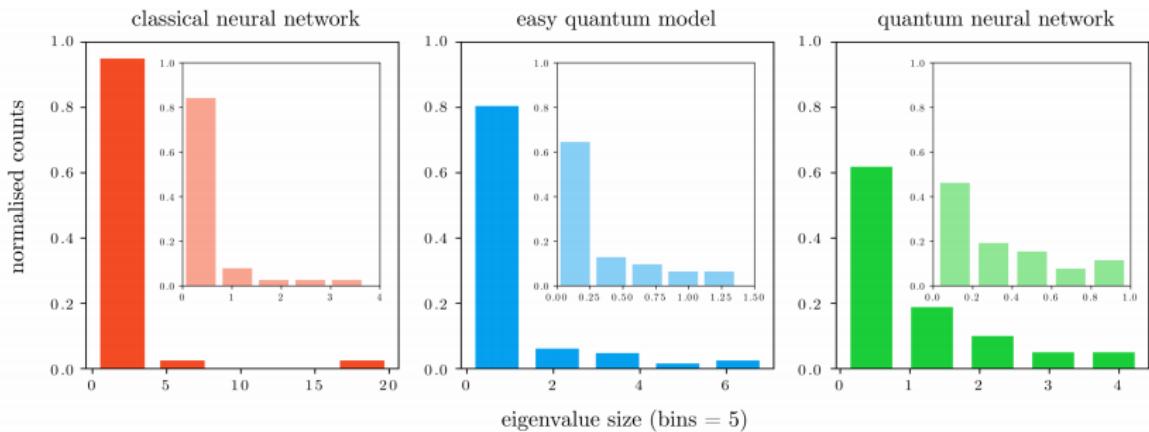


Figura: Amira Abbas et al.

<https://arxiv.org/pdf/2011.00027.pdf>

El problema de los barren plateaus

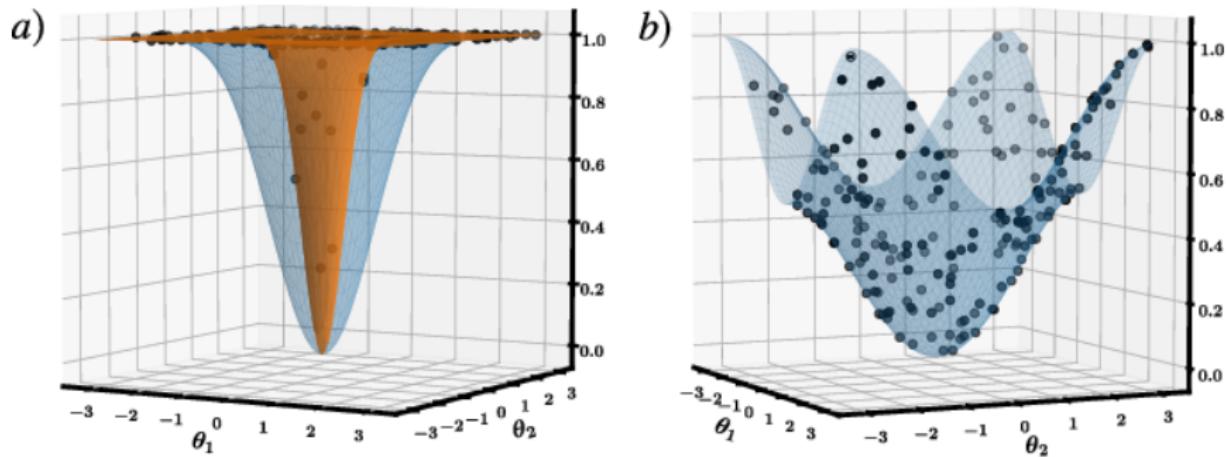
ARTICLE

DOI: 10.1038/s41467-018-07090-4

OPEN

Barren plateaus in quantum neural network training landscapes

Jarrod R. McClean¹, Sergio Boixo¹, Vadim N. Smelyanskiy¹, Ryan Babbush¹ & Hartmut Neven¹



El problema de los barren plateaus: posibles soluciones

Cost Function Dependent Barren Plateaus in Shallow Parametrized Quantum Circuits

M. Cerezo,^{1,2,*} Akira Sone,^{1,2} Tyler Volkoff,¹ Lukasz Cincio,¹ and Patrick J. Coles^{1,*}

¹Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA.

²Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA

Variational quantum algorithms (VQAs) optimize the parameters θ of a parametrized quantum circuit $V(\theta)$ to minimize a cost function C . While VQAs may enable practical applications of noisy quantum computers, they are nevertheless heuristic methods with unproven scaling. Here, we rigorously prove two results, assuming $V(\theta)$ is an alternating layered ansatz composed of blocks forming local 2-designs. Our first result states that defining C in terms of global observables leads to exponentially vanishing gradients (i.e., barren plateaus) even when $V(\theta)$ is shallow. Hence, several VQAs in the literature must revise their proposed costs. On the other hand, our second result states that defining C with local observables leads to at worst a polynomially vanishing gradient, so long as the depth of $V(\theta)$ is $\mathcal{O}(\log n)$. Our results establish a connection between locality and trainability. We illustrate these ideas with large-scale simulations, up to 100 qubits, of a quantum autoencoder implementation.

PHYSICAL REVIEW X 11, 041011 (2021)

Absence of Barren Plateaus in Quantum Convolutional Neural Networks

Arthur Pesah^{1,2} M. Cerezo,^{1,3} Samson Wang,^{1,4} Tyler Volkoff,¹ Andrew T. Sornborger,⁵ and Patrick J. Coles¹

¹Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

²Department of Physics and Astronomy, University College London, London WC1E 6BT, United Kingdom

³Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87544

⁴Imperial College London, London, United Kingdom

⁵Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA

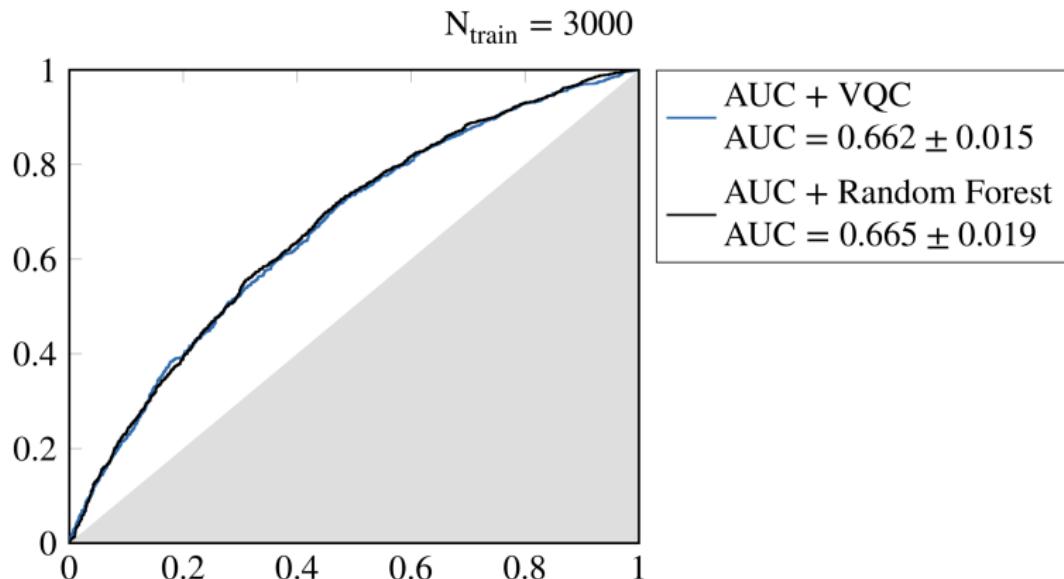


(Received 12 March 2021; revised 13 July 2021; accepted 2 August 2021; published 15 October 2021)

Quantum neural networks (QNNs) have generated excitement around the possibility of efficiently analyzing quantum data. But this excitement has been tempered by the existence of exponentially vanishing gradients, known as barren plateau landscapes, for many QNN architectures. Recently, quantum convolutional neural networks (QCNNs) have been proposed, involving a sequence of convolutional and pooling layers that reduce the number of qubits while preserving information about relevant data

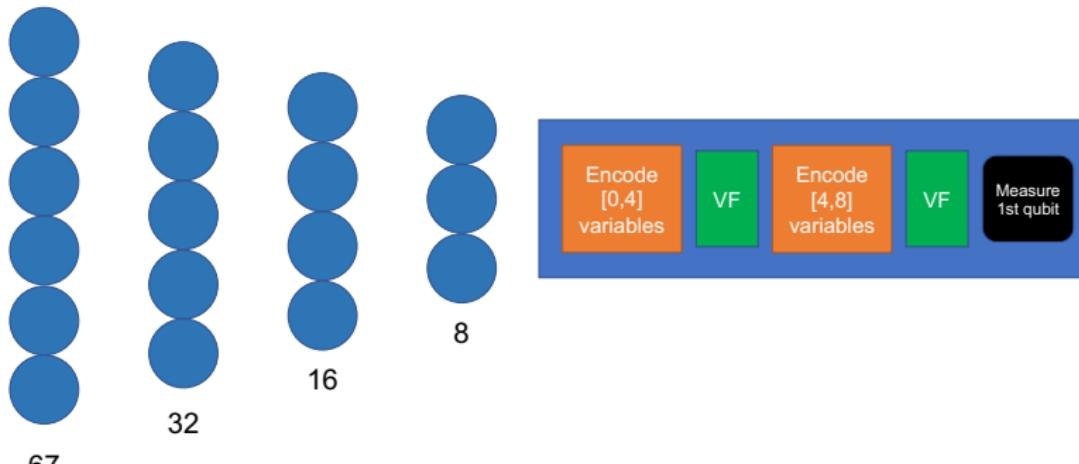
Resultados en el problema de clasificación de HEP

- 4 qubits, 8 variables (data-reuploading)
- ZZ feature map
- Forma variacional 2-local



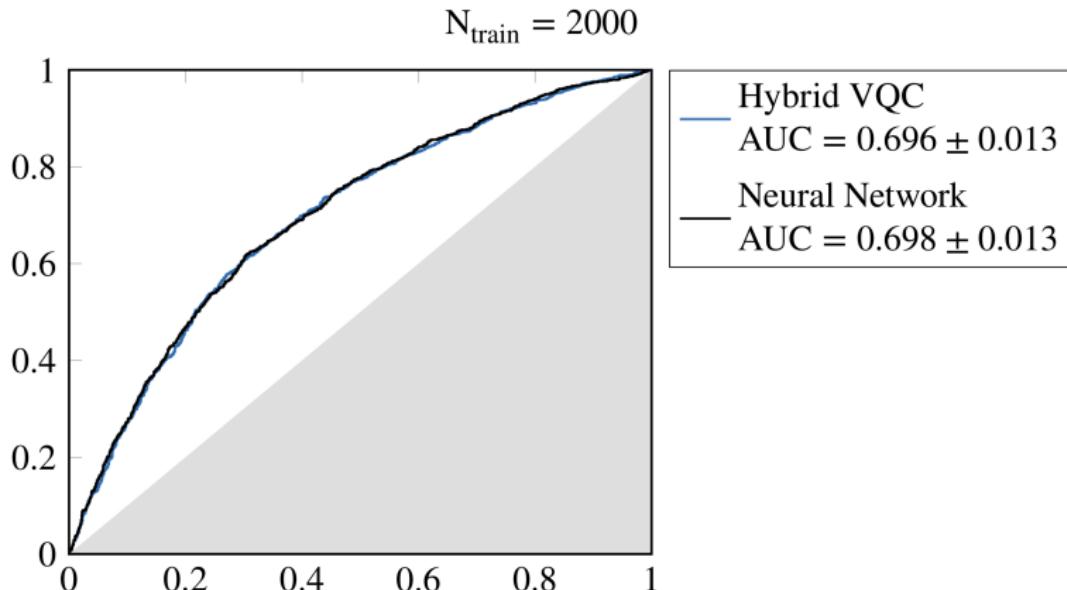
Arquitectura híbrida

- Usamos una red neuronal clásica para reducir la dimensionalidad
- La red neuronal cuántica se usa para clasificar
- Se entrena conjuntamente



Resultados con la arquitectura híbrida

- 4 qubits, 8 variables (data-reuploading)
- ZZ feature map
- Forma variacional 2-local



Parte X

Quantum Generative Adversarial
Networks

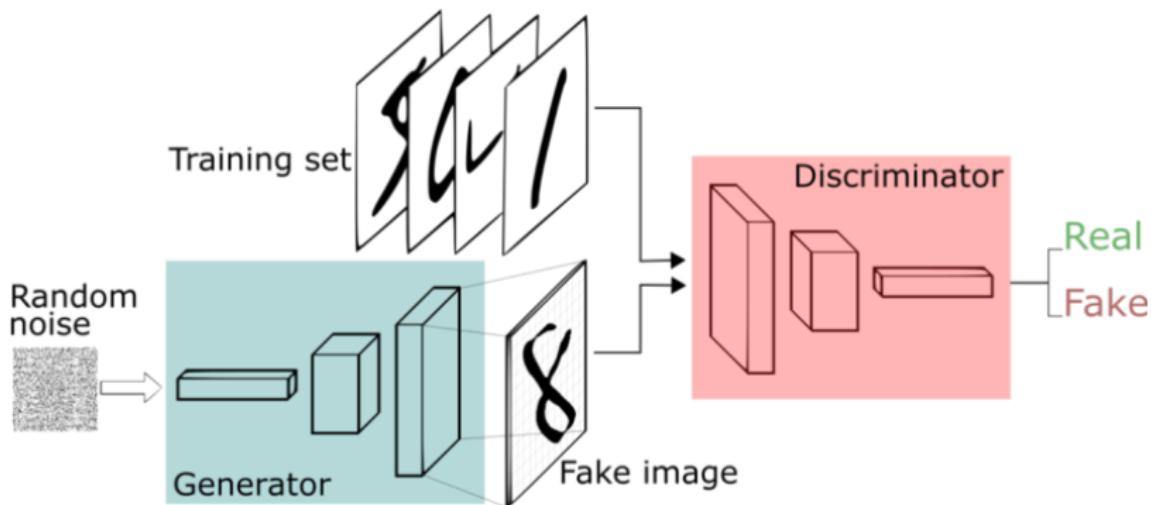
GANs: Generative Adversarial Networks

- Las Generative Adversarial Networks (GANs) fueron propuestas por Ian Goodfellow y sus colaboradores en 2014
- Su objetivo es is, dado un conjunto de datos, aprender a generar nuevos ejemplos que sigan la misma distribución
- Se han conseguido resultados impresionantes en muchas aplicaciones



Arquitectura de una GAN

- Dos redes neuronales: generador y discriminador
- El generador trata de “engañar” al discriminador
- El discriminador intenta distinguir entre datos reales y falsos



Entrenando una GAN

- El generador y el discriminador se entrena en fases alternas
- El discriminador trata de maximizar

$$E_x [\log D(x)] + E_z [\log (1 - D(G(z)))]$$

- El generador trata de minimizar

$$E_z [\log (1 - D(G(z)))]$$

o (en la práctica) de maximizar

$$E_z [\log D(G(z))]$$

GANs cuánticas

- Una GAN cuántica (QGAN) reemplaza el generador o el discriminador (o ambos) por un circuito cuántico

PHYSICAL REVIEW LETTERS **121**, 040502 (2018)

Editors' Suggestion

Featured in Physics

Quantum Generative Adversarial Learning

Seth Lloyd¹ and Christian Weedbrook²

¹*Massachusetts Institute of Technology, Department of Mechanical Engineering,
77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*
²*Xanadu, 372 Richmond Street W, Toronto, Ontario M5V 1X6, Canada*



(Received 30 April 2018; published 26 July 2018)

PHYSICAL REVIEW A **98**, 012324 (2018)

Quantum generative adversarial networks

Pierre-Luc Dallaire-Demers^{*} and Nathan Killoran

Xanadu, 372 Richmond Street W, Toronto, Ontario M5V 1X6, Canada



(Received 7 May 2018; published 23 July 2018)

Usando QGANs para cargar distribuciones de probabilidad

npj Quantum Information

www.nature.com/npjqi

ARTICLE OPEN

Quantum Generative Adversarial Networks for learning and loading random distributions

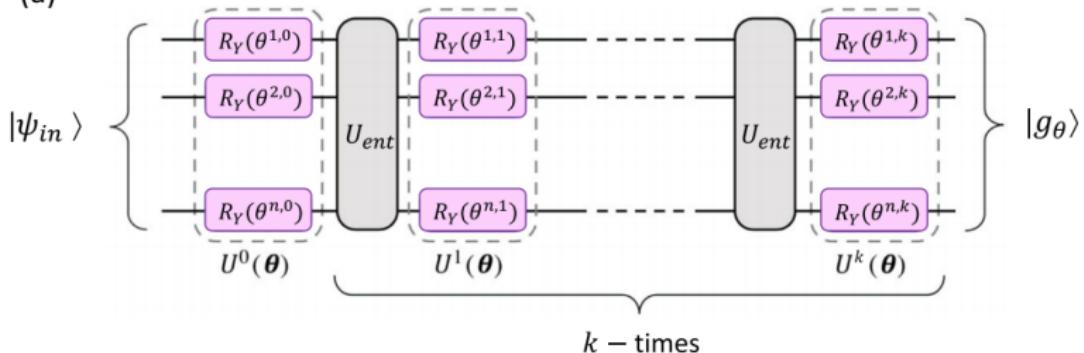
Christa Zoufal^{1,2*}, Aurélien Lucchi² and Stefan Woerner¹

Quantum algorithms have the potential to outperform their classical counterparts in a variety of tasks. The realization of the advantage often requires the ability to load classical data efficiently into quantum states. However, the best known methods require $\mathcal{O}(2^n)$ gates to load an exact representation of a generic data structure into an n -qubit state. This scaling can easily predominate the complexity of a quantum algorithm and, thereby, impair potential quantum advantage. Our work presents a hybrid quantum-classical algorithm for efficient, approximate quantum state loading. More precisely, we use quantum Generative Adversarial Networks (qGANs) to facilitate efficient learning and loading of generic probability distributions - implicitly given by data samples - into quantum states. Through the interplay of a quantum channel, such as a variational quantum circuit, and a classical neural network, the qGAN can learn a representation of the probability distribution underlying the data samples and load it into a quantum state. The loading requires $\mathcal{O}(\text{poly}(n))$ gates and can thus enable the use of potentially advantageous quantum algorithms, such as Quantum Amplitude Estimation. We implement the qGAN distribution learning and loading method with Qiskit and test it using a quantum simulation as well as actual quantum processors provided by the IBM Q Experience. Furthermore, we employ quantum simulation to demonstrate the use of the trained quantum channel in a quantum finance application.

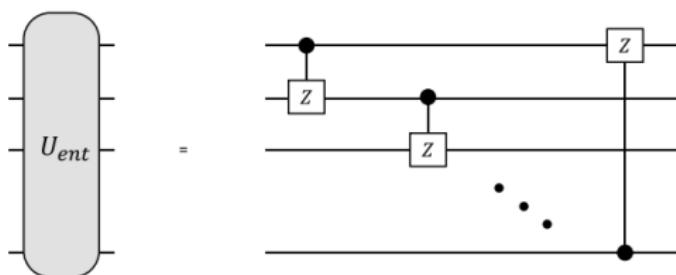
npj Quantum Information (2019)5:103 ; <https://doi.org/10.1038/s41534-019-0223-2>

El generador cuántico en la QGAN de IBM

(a)

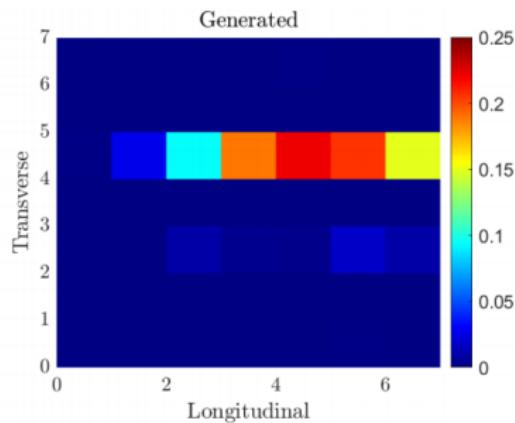
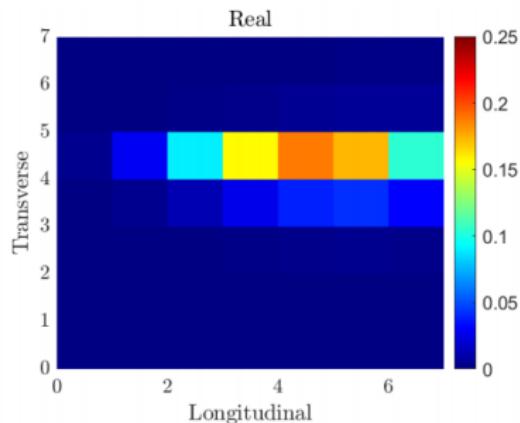


(b)

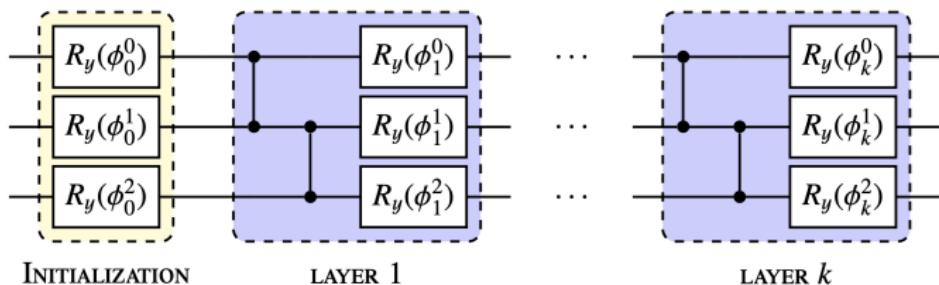
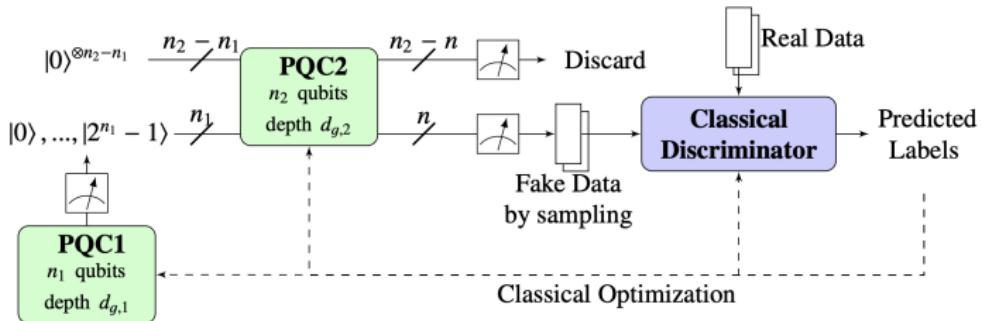


Aplicación de QGANs en HEP: Imágenes de calorímetros

- Proyección bidimensional del depósito de energía tras colisión de partículas

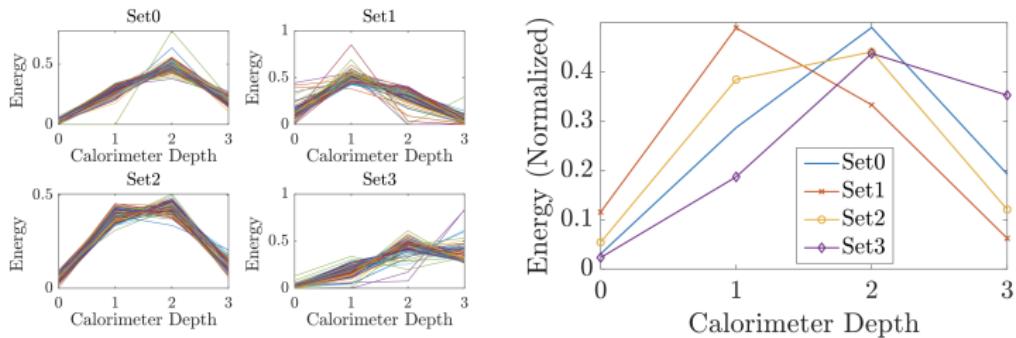


El modelo Dual-PQC GAN



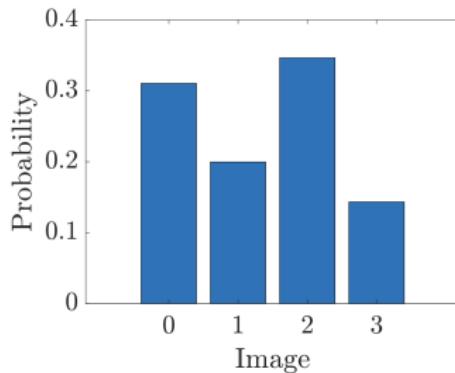
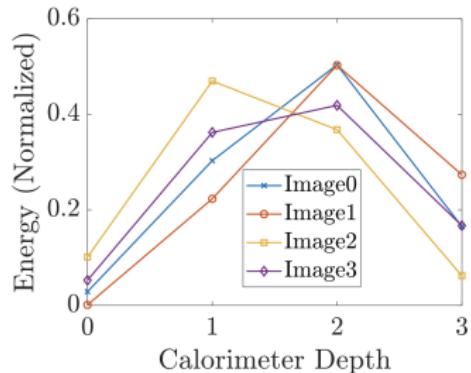
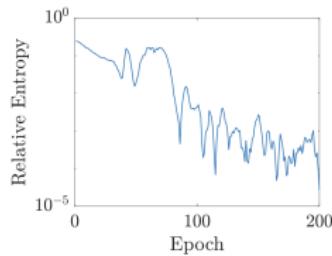
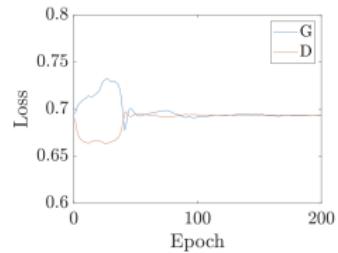
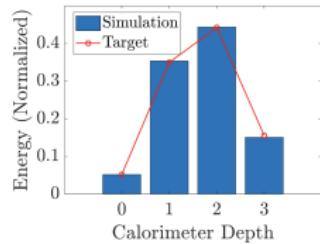
Entrenando el modelo

- Simplificación a una dimensión, 4 pixels
- 20,000 imágenes (agrupadas para evaluación)
- $n_1 = n_2 = 4, n = 2$
- Discriminador implementando en PyTorch (4,256,128,1)



Algunos resultados

- $d_{g,1} = 2$
- $d_{g,1} = 16$



Dual-Parameterized Quantum Circuit GAN Model in High Energy Physics

Su Yeon Chang^{1,2,}, Steven Herbert^{3,4,**}, Sofia Vallecorsa^{1,***}, Elías F. Combarro⁵, and Ross Duncan^{3,6,7}*

¹OpenLab, CERN, Geneva, Switzerland

²Department of Physics, EPFL, Lausanne, Switzerland

³Cambridge Quantum Computing Ltd, Cambridge, UK

⁴Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

⁵Department of Computer Science, University of Oviedo, Oviedo, Spain

⁶Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK

⁷Department of Physics, University College London, London, UK

Abstract. Generative models, and Generative Adversarial Networks (GAN) in particular, are being studied as possible alternatives to Monte Carlo simulations. It has been proposed that, in certain circumstances, simulation using GANs can be sped-up by using quantum GANs (qGANs).

We present a new design of qGAN, the dual-Parameterized Quantum Circuit (PQC) GAN, which consists of a classical discriminator and two quantum generators which take the form of PQCs. The first PQC learns a probability distribution over N -pixel images, while the second generates normalized pixel intensities of an individual image for each PQC input.

With a view to HEP applications, we evaluated the dual-PQC architecture on the task of imitating calorimeter outputs, translated into pixelated images. The results demonstrate that the model can reproduce a fixed number of images with a reduced size as well as their probability distribution and we anticipate it should allow us to scale up to real calorimeter outputs.

Parte XI

Conclusiones

Conclusiones

- Los ordenadores cuánticos de los que disponemos hoy en día son o de propósito específico (quantum annealers) o NISQ
- Aún así, se están estudiando sus aplicaciones en:
 - Optimización
 - Simulación química y física
 - Quantum Machine Learning
- Las herramientas principales son:
 - Modelos QUBO/Ising
 - Circuitos variacionales
- Campo fascinante pero aún no entendido por completo

El futuro está por construir

