

Aplicaciones de la optimización cuántica y el quantum machine learning

Elías F. Combarro (Universidad de Oviedo)

efernandezca@uniovi.es

<https://github.com/EliasCombarro/UAL2024>

Universidad de Almería - Mayo 2024



UNIVERSIDAD
DE ALMERÍA



Universidad de Oviedo

Parte I

Introducción y objetivos

The Shape of Things to Come

Quantum computing breakthrough could happen with just hundreds, not millions, of qubits using new error-correction system

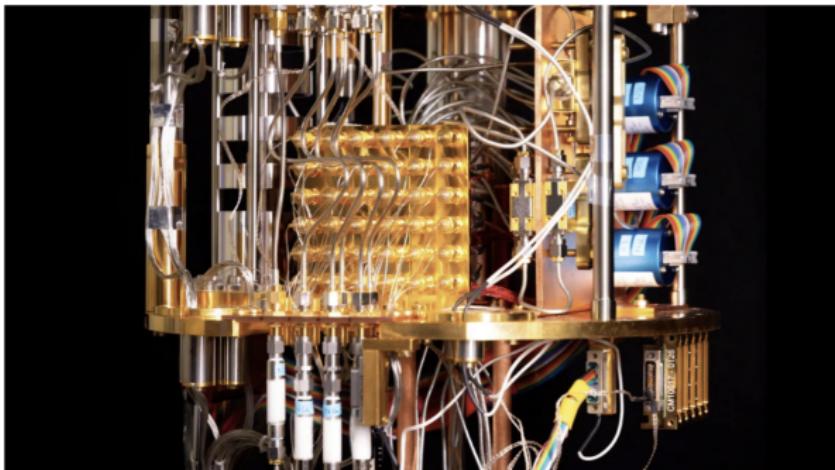
News

By Keumars Afifi-Sabet published 23 hours ago

Scientists have designed a physical qubit that behaves as an error-correcting "logical qubit," and now they think they can scale it up to make a useful quantum computer using a few hundred.



Comments (1)

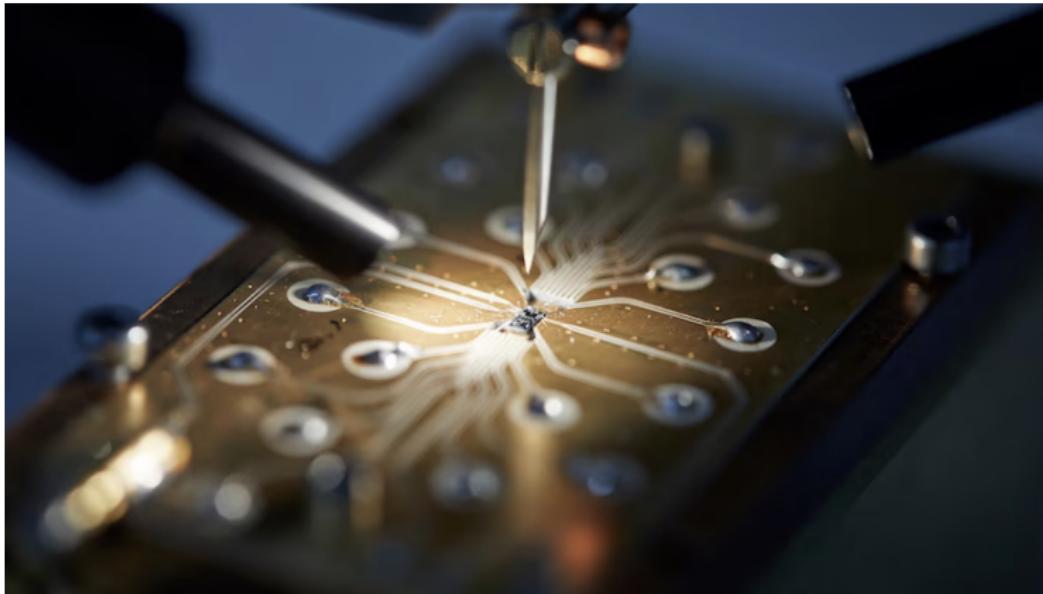


The Shape of Things to Come

Australia signs deal worth almost \$1b with PsiQuantum to build world's first 'useful' quantum computer

ABC Radio Brisbane / By [Kenji Sato](#), [Craig Zonca](#) and [Loretta Ryan](#), and [Steve Austin](#)

Posted 8h ago



Computación cuántica y optimización

RESEARCH ARTICLE | PHYSICS

f X in G Q E M

An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems via computational learning theory

NIKLAS PIRNAY , VINCENT ULITZSCH , FREDERIK WILDE , JENS EISERT , AND JEAN-PIERRE SEIFERT  Authors Info & Affiliations

SCIENCE ADVANCES • 15 Mar 2024 • Vol 10, Issue 11 • DOI: 10.1126/sciadv.ad5170

4,473



Abstract

It is unclear to what extent quantum algorithms can outperform classical algorithms for problems of combinatorial optimization. In this work, by resorting to computational learning theory and cryptographic notions, we give a fully constructive proof that quantum computers feature a super-polynomial advantage over classical computers in approximating combinatorial optimization problems. Specifically, by building on seminal work by Kearns and Valiant, we provide special instances that are hard for classical computers to approximate up to polynomial factors. Simultaneously, we give a quantum algorithm that can efficiently approximate the optimal solution within a polynomial factor. The quantum advantage in this work is ultimately borrowed from Shor's quantum algorithm for factoring. We introduce an explicit and comprehensive end-to-end construction for the advantage bearing instances. For these instances, quantum computers have, in principle, the power to approximate combinatorial optimization solutions beyond the reach of classical efficient algorithms.



Objetivos de este curso

- Recordar los conceptos fundamentales de la computación cuántica
- Profundizar en las aplicaciones de la computación cuántica en problemas de optimización y quantum machine learning
- Estudiar en detalle los siguientes métodos:
 - Quantum Annealing
 - Quantum Approximate Optimization Algorithm (QAOA)
 - Variational Quantum Eigensolver (VQE)
 - Quantum Support Vector Machines
 - Quantum Neural Networks



Recursos y herramientas

- Jupyter Notebooks
 - Entorno para la creación y ejecución de documentos que combinan código, texto enriquecido e imágenes
 - Trabajaremos principalmente en Google Colab
- Librerías de programación cuántica
 - Quantum Annealers: D-Wave Leap y Ocean
<https://cloud.dwavesys.com/leap/>
 - Ordenadores basados en puertas: IBM Quantum y Qiskit
<https://quantum.ibm.com/>
 - Software para QML: PennyLane
<https://www.xanadu.ai/products/pennylane/>
- Notebooks del curso en
<https://github.com/EliasCombarro/UAL2024>

D:wave



Para saber más

A Practical Guide to Quantum Machine Learning and Quantum Optimization

This book provides deep coverage of modern quantum algorithms, including machine learning and optimization, to help you solve real-world problems. You'll be introduced to quantum computing using a hands-on approach that requires minimal mathematical and physical knowledge to understand the topics. You'll discover many algorithms, tools, and methods to model optimization problems with QUBO and Ising formalisms and find out how to solve optimization problems with quantum annealing, QAOA, Grover Adaptive Search, and VQE. The book also shows you how to train quantum machine learning models such as quantum support vector machines, quantum neural networks, and quantum generative adversarial networks. The book takes a straightforward path to helping you learn about algorithms through chapters illustrating them with code that's ready to be run on quantum simulators and actual quantum computers. You'll also see how to utilize programming languages such as IBM's Qiskit, Xanadu's PennyLane, and D-Wave's Leap.

By the end of this book, you'll have built a solid foundation in the fundamentals of quantum computing, along with a wide variety of modern quantum algorithms and programming skills that'll enable you to start applying quantum methods to solve practical problems right away.

WHAT YOU WILL LEARN

- Review the basics of quantum computing
- Gain a solid understanding of modern quantum algorithms
- Understand how to formulate optimization problems with QUBO
- Solve optimization problems with quantum annealing, QAOA, GAs, and VQE
- Find out how to create quantum machine learning models
- Explore how quantum support vector machines and quantum neural networks work using Qiskit and PennyLane
- Discover how to implement hybrid architectures using Qiskit and PennyLane and its PyTorch interface

www.packtpub.com

Get a free PDF of this book

9781839210001

SAMUEL GONZÁLEZ-CASTILLO
ELÍAS F. COMBARRO

A Practical Guide to Quantum Machine Learning and Quantum Optimization

Leverage the power of quantum computing by applying quantum methods to solve practical problems

ELÍAS F. COMBARRO
SAMUEL GONZÁLEZ-CASTILLO
Foreword by Alberto Di Meglio,
Head of Innovation - Coordinator CERN Quantum Technology Initiative

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



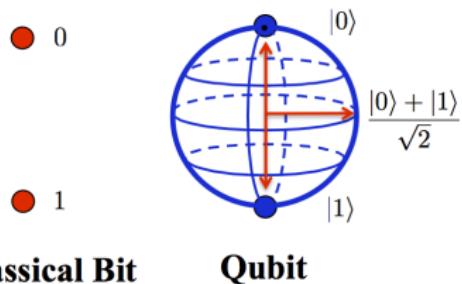
Qubits

- 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

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



Pequeño recordatorio de números complejos

- Un número complejo tiene la forma

$$a + bi$$

donde a y b son números reales e i cumple $i^2 = -1$

- La suma de números complejos se define como

$$(a_1 + b_1 i) + (a_2 + b_2 i) = (a_1 + a_2) + (b_1 + b_2)i$$

- La multiplicación de números complejos se define como

$$(a_1 + b_1 i) \cdot (a_2 + b_2 i) = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2)i$$

- El módulo de un número complejo es

$$|a + bi| = \sqrt{a^2 + b^2}$$

Pequeño recordatorio de números complejos (2)

- El conjugado de un número complejo es

$$\overline{a + bi} = a - bi$$

- Se cumple que

$$(a + bi)(\overline{a + bi}) = a^2 + b^2 = |a + bi|^2$$

- Fórmula de Euler:

$$e^{i\alpha} = \cos(\alpha) + \sin(\alpha)i$$

- Se verifica:

$$e^{i\alpha} e^{i\beta} = e^{i(\alpha+\beta)}$$

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

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

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

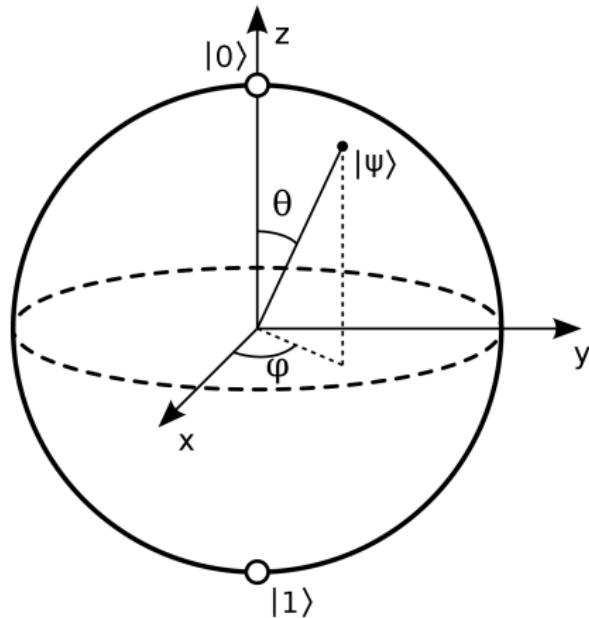
- Como las fases globales son físicamente irrelevantes, podemos reescribir

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

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

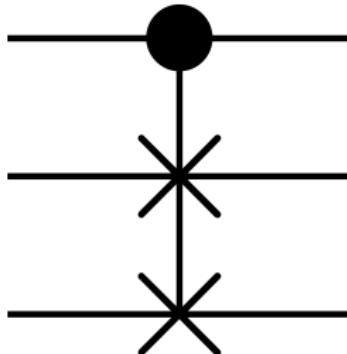
$$UU^\dagger = U^\dagger U = I$$

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

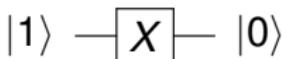
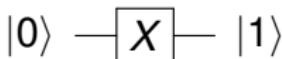
$$|(a\alpha + b\beta)|^2 + |(c\alpha + d\beta)|^2 = 1$$

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

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

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 es hacer 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:

$$\begin{array}{c} |x\rangle \xrightarrow{\bullet} |x\rangle \\ |y\rangle \xrightarrow{\oplus} |y \oplus x\rangle \end{array}$$

- Es una puerta muy importante, puesto que nos permite:
 - Realizar entrelazamientos (más sobre ello enseguida)
 - 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 $|0\dots00\rangle$
 - 1 con probabilidad $|\alpha_1|^2$ y el nuevo estado será $|0\dots01\rangle$
 - ...
 - $2^n - 1$ con probabilidad $|\alpha_{2^n-1}|^2$ y el nuevo estado será $|1\dots11\rangle$
- 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 de forma exacta**

Teorema

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

Parte III

Brevísima introducción a la
complejidad computacional

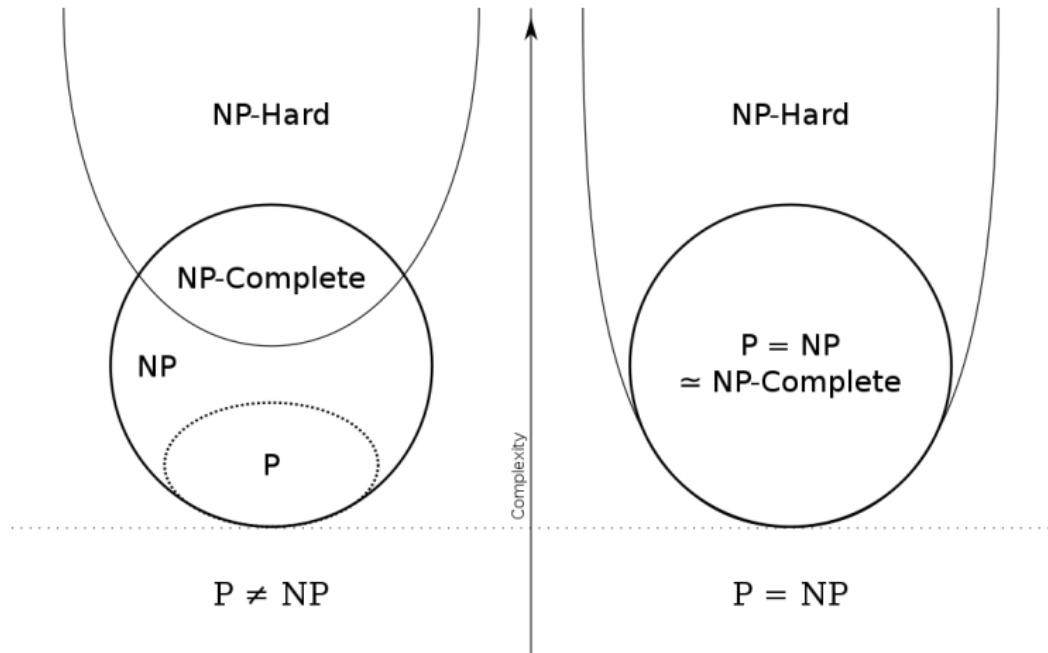
Las clases P y NP

- Las clases P y NP contienen problemas de decisión (respuesta sí y no)
- La clase P incluye los problemas de decisión que pueden ser resueltos en tiempo polinomial (en el tamaño de la entrada)
- Ejemplos de problemas en P :
 - Comprobar si una cadena binaria es un palíndromo
 - Comprobar si un número es primo o no
 - Comprobar si un grafo tiene un camino euleriano
- La clase NP incluye los problemas de decisión cuya solución (positiva) puede ser comprobada en tiempo polinomial (en el tamaño de la entrada)
- Ejemplos de problemas en NP :
 - Todos los de P
 - Comprobar si un número tiene un factor menor que k
 - Comprobar si un grafo tiene un camino hamiltoniano

Reducciones y problemas *NP*-duros y completos

- Un problema *A* se puede reducir a otro problema *B* si podemos usar un algoritmo que resuelva *B* para resolver *A*
- Un problema es *NP*-duro si cualquier problema de *NP* es reducible a él en tiempo polinomial
- Un problema es *NP*-completo si es *NP*-duro y está en *NP*
- Ejemplos de problemas *NP*-duros
 - El problema del viajante de comercio
 - El problema del corte máximo
 - Comprobar si un grafo tiene un camino hamiltoniano
- Ejemplos de problemas *NP*-completos
 - *SAT*
 - Dado un grafo, determinar si tiene un corte de tamaño *k* o mayor
 - Comprobar si un grafo tiene un camino hamiltoniano

Relaciones entre P y NP



Parte IV

QUBO: Quadratic Unconstrained
Binary Optimization

Problemas de optimización combinatoria

- Los problemas en los que nos vamos a centrar principalmente provienen de la optimización combinatoria
- Esto incluye problemas muy famosos:
 - El problema del viajante de comercio
 - El problema de la mochila
 - Hallar un coloreado mínimo en un grafo
 - Determinar si una fórmula es satisfacible
 - ...
- Se trata de problemas NP-hard (en algunos casos, NP-completos)



Reformulando problemas de optimización combinatoria

- Para usar métodos como QAOA o quantum annealing, necesitamos reformular nuestros problemas
- Los dos “formatos” principales que usaremos son:
 - QUBO (Quadratic Unconstrained Binary Optimization): Minimizar

$$\sum_{i,j} q_{ij} x_i x_j$$

con $x_i \in \{0, 1\}$ y q_{ij} coeficientes reales

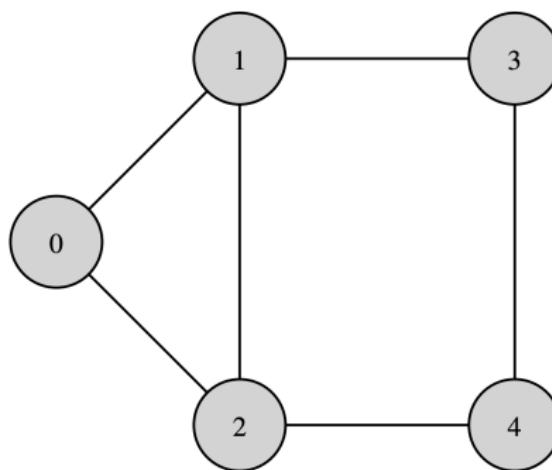
- Modelo de Ising: Minimizar

$$-\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j$$

con $Z_i \in \{-1, 1\}$ y J_{jk}, h_j coeficientes reales

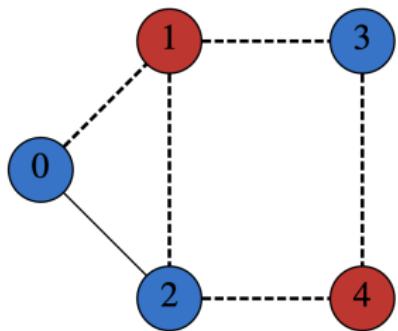
El problema del corte máximo

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

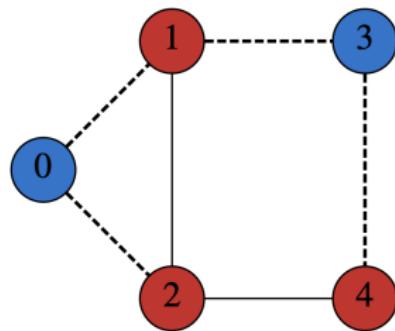


- Es un problema NP-hard

Dos posibles cortes



(a) Optimal cut.



(b) Non-optimal cut.

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

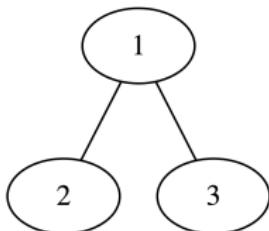
$$\text{Minimizar} \sum_{(i,j) \in E} Z_i Z_j$$

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

$$H = Z_1 Z_2 + Z_1 Z_3$$



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

El modelo de Ising

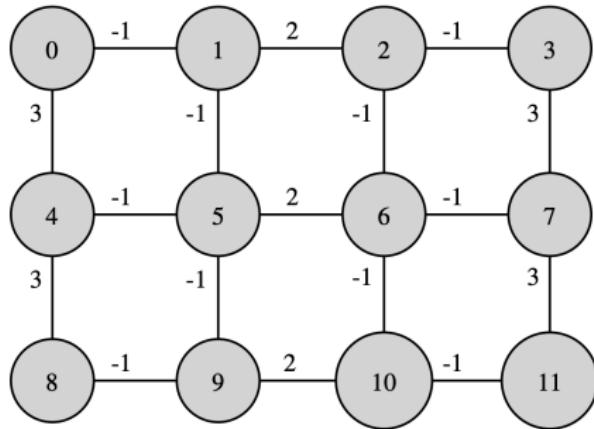
- Se tienen n partículas con spin, que interactúan entre sí con ciertas fuerzas de acoplamiento
- Su hamiltoniano es

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

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

Un ejemplo de modelo de Ising



Minimizar
$$\begin{aligned} & z_0z_1 - 2z_1z_2 + z_2z_3 - 3z_0z_4 + z_1z_5 + z_2z_6 - 3z_3z_7 \\ & + z_4z_5 - 2z_5z_6 + z_6z_7 - 3z_4z_8 + z_5z_9 + z_6z_{10} - 3z_7z_{11} \\ & + z_8z_9 - 2z_9z_{10} + z_{10}z_{11} - z_0 - z_1 - z_2 - z_3 - z_4 - z_5 \\ & - z_6 - z_7 - z_8 - z_9 - z_{10} - z_{11} \end{aligned}$$

sujeto a
$$z_j \in \{-1, 1\}, \quad j = 0, \dots, 11.$$

¿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

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

Extendiendo la búsqueda a todos los estados cuánticos

- Un estado cuántico genérico se puede escribir como

$$|\psi\rangle = \sum_x a_x |x\rangle$$

- Para cada x se verifica

$$Z_j Z_k |x\rangle = \pm |x\rangle$$

- Tenemos que

$$\begin{aligned}\langle\psi| Z_j Z_k |\psi\rangle &= \left(\sum_y a_y^* \langle y| \right) Z_j Z_k \left(\sum_x a_x |x\rangle \right) \\ &= \sum_y \sum_x a_y^* a_x \langle y| Z_j Z_k |x\rangle \\ &= \sum_x |a_x|^2 \langle x| Z_j Z_k |x\rangle\end{aligned}$$

Extendiendo la búsqueda a todos los estados cuánticos

- Por linearidad

$$\begin{aligned}\langle \psi | (Z_1 Z_2 + Z_1 Z_3) | \psi \rangle &= \langle \psi | Z_1 Z_2 | \psi \rangle + \langle \psi | Z_1 Z_3 | \psi \rangle \\&= \sum_x |a_x|^2 \langle x | Z_1 Z_2 | x \rangle \\&\quad + \sum_x |a_x|^2 \langle x | Z_1 Z_3 | x \rangle \\&= \sum_x |a_x|^2 \langle x | (Z_1 Z_2 + Z_1 Z_3) | x \rangle\end{aligned}$$

Extendiendo la búsqueda a todos los estados cuánticos

- Se sigue que

$$\begin{aligned}& \sum_x |a_x|^2 \langle x | (Z_1 Z_2 + Z_1 Z_3) |x\rangle \\& \geq \sum_x |a_x|^2 \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle \\& = \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle \sum_x |a_x|^2 \\& = \langle x_{\min} | (Z_1 Z_2 + Z_1 Z_3) |x_{\min}\rangle\end{aligned}$$

donde x_{\min} es el estado para el que se alcanza el mínimo.

El principio variacional

- Podemos extender el resultado a una matriz de Ising de la forma

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

ya que $Z|x\rangle = \pm|x\rangle$.

- Tendremos que

$$\langle \psi | H | \psi \rangle = \sum_x |a_x|^2 \langle x | H | x \rangle$$

- Y también que

$$\langle \psi | H | \psi \rangle \geq \langle x_{min} | H | x_{min} \rangle$$

- A veces se denomina “principio variacional” a este resultado

El maravilloso mundo de los hamiltonianos

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

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

sea mínimo, con

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

- 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 ($H = H^\dagger$)
- La energía de un estado $|\psi\rangle$ es

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

Ejemplo: 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-hard (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 caso 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

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

El problema Subset Sum

- El problema Subset Sum consiste en, dado un conjunto de números enteros $S = \{a_0, \dots, a_m\}$ y un valor objetivo T , hallar un subconjunto de S cuya suma sea lo más cercana posible a T .
- Es NP-hard
- Se podría escribir como un caso de minimizar un modelo de Ising
- Es más sencillo escribirlo como minimizar

$$c(x_0, x_1, \dots, x_m) = (a_0x_0 + a_1x_1 + \dots + a_mx_m - T)^2$$

donde $x_i \in \{0, 1\}$.

Ejemplo de Subset Sum

- $S = \{1, 4, -2\}$ y $T = 2$
- Tenemos que minimizar

$$(x_0 + 4x_1 - 2x_2 - 2)^2$$

- Expandiendo, tenemos

$$x_0^2 + 8x_0x_1 - 4x_0x_2 - 4x_0 + 16x_1^2 - 16x_1x_2 - 16x_1 + 4x_2^2 + 8x_2 + 4$$

un polinomio de grado dos en variables binarias

- Podríamos simplificar: eliminar el término independiente y tener en cuenta que $x_i^2 = x_i$
- Es un caso de un problema QUBO: Quadratic Unconstrained Binary Optimization

QUBO: Quadratic Unconstrained Binary Optimization

- Los problemas QUBO se plantean como

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

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

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

Binary Linear Programming

- Los problemas BLP (Binary Linear Programming) son del tipo

$$\text{Minimizar } c_0x_0 + c_1x_1 + \dots + c_mx_m$$

$$\text{sujeto a } Ax \leq b,$$

$$x_j \in \{0, 1\}, \quad j = 0, \dots, m,$$

- BLP es NP-hard
- ¡Tiene restricciones!

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 $x_i^2 = x_i$ 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 BLP 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

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

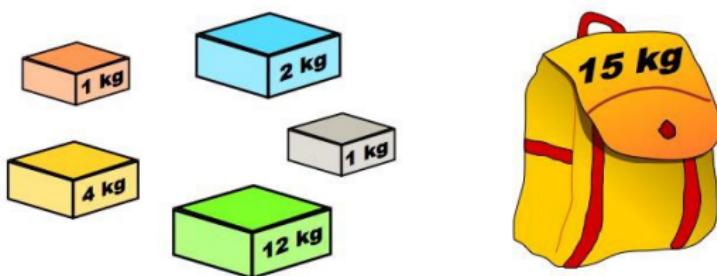
que es un problema QUBO (hemos elegido $M = 100$).

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

El problema de la mochila

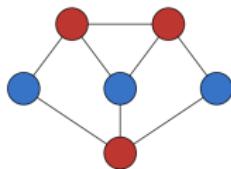
- Nos dan objetos $j = 0, \dots, m$, cada uno con peso w_j y valor c_j
- Queremos elegir objetos, maximizando el valor total, sin pasarnos de un peso máximo W
- Es un problema NP-hard
- Claramente, es un BLP:

$$\begin{aligned} & \text{Minimizar} && -c_0x_0 - c_1x_1 - \dots - c_mx_m \\ & \text{sujeto a} && w_0x_0 + w_1x_1 + \dots + w_mx_m \leq W, \\ & && x_j \in \{0, 1\}, \quad j = 0, \dots, m, \end{aligned}$$

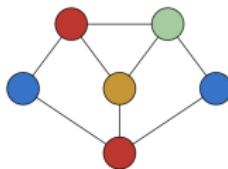


Coloreado de grafos

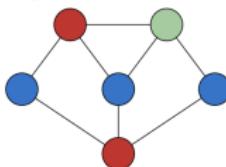
- Dado un grafo, queremos saber cuál es el menor número de colores necesarios para colorear sus vértices sin que dos vértices adyacentes reciban el mismo color
- Es NP-hard
- Decidir si un grafo se puede colorear con k colores es NP-completo (para k al menos 3)
- Tiene múltiples aplicaciones



(a) Invalid colouring



(b) Non-optimal colouring



(c) Optimal colouring

Versión QUBO del coloreado de grafos

- Queremos saber si un grafo de m vértices es coloreable con k colores
- Definimos variables binarias x_{jl} with $j = 0, \dots, m$ and $l = 0, \dots, k - 1$
- x_{jl} es 1 si y solo si el vértice j recibe el color l
- Restricción por cada vértice:

$$\sum_{l=0}^{k-1} x_{jl} = 1$$

- Restricción por cada par de vértices j y h adyacentes en el grafo:

$$\sum_{l=0}^{k-1} x_{jl} x_{hl} = 0$$

Versión QUBO del coloreado de grafos

- Versión QUBO:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{j=0}^m \left(\sum_{l=0}^{k-1} x_{jl} - 1 \right)^2 + \sum_{(j,h) \in E} \sum_{l=0}^{k-1} x_{jl} x_{hl} \\ \text{sujeto a} \quad & x_{jl} \in \{0, 1\}, \quad j = 0, \dots, m, l = 0, \dots, k-1, \end{aligned}$$

- Si el mínimo es 0, el grafo es coloreable con k colores. Si no, no lo es.

El problema del viajante de comercio

- Hay una serie de ciudades $j = 0, \dots, m$ que tenemos que recorrer, minimizando el coste total
- El coste de viajar de j a l es w_{jl}
- Es un problema NP-hard



Versión QUBO del problema del viajante de comercio

- Variables binarias x_{jl}
- x_{jl} es 1 si y solo si la ciudad j se visita en la posición l
- Restricción por cada ciudad:

$$\sum_{l=0}^m x_{jl} = 1$$

- Restricción por cada posición:

$$\sum_{j=0}^m x_{jl} = 1$$

- Coste del viaje

$$\sum_{l=0}^{m-1} \sum_{j=0}^m \sum_{k=0}^m w_{jk} x_{jl} x_{kl+1}$$

Versión QUBO del problema del viajante de comercio

- Formulación completa:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{l=0}^{m-1} \sum_{j=0}^m \sum_{k=0}^m w_{jk} x_{jl} x_{kl+1} \\ & + B \left(\sum_{l=0}^m x_{jl} - 1 \right)^2 + B \left(\sum_{j=0}^m x_{jl} - 1 \right)^2 \\ \text{sujeto a} \quad & x_{jl} \in \{0, 1\}, \quad j, l = 0, \dots, m, \end{aligned}$$

- Elegimos B suficientemente grande. Por ejemplo:

$$B = 1 + \sum_{j,k=0}^m w_{jk}$$

Otros problemas

frontiers in
PHYSICS

REVIEW ARTICLE
published: 12 February 2014
doi: 10.3389/fphy.2014.00005



Ising formulations of many NP problems

Andrew Lucas*

Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA

Edited by:

Jacob Biamonte, ISI Foundation,
Italy

Reviewed by:

Mauro Faccin, ISI Foundation, Italy
Ryan Babbush, Harvard University,
USA

Bryan A. O'Gorman, NASA, USA

***Correspondence:**

Andrew Lucas, Lyman Laboratory of
Physics, Department of Physics,
Harvard University, 17 Oxford St.,
Cambridge, MA 02138, USA
e-mail: lucas@fas.harvard.edu

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering, and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

Keywords: spin glasses, complexity theory, adiabatic quantum computation, NP, algorithms

Quantum Information Processing (2022) 21:67

<https://doi.org/10.1007/s11128-021-03405-5>



Unconstrained binary models of the travelling salesman problem variants for quantum optimization

Özlem Salehi¹ · Adam Glos¹ · Jarosław Adam Miszczak¹

Received: 28 June 2021 / Accepted: 26 December 2021 / Published online: 22 January 2022
© The Author(s) 2022

Ejercicios

- Escribir la versión QUBO/Ising de los siguientes problemas:
 - Determinar si un grafo tiene un **camino Hamiltoniano** o no
 - **Vertex cover**: dado un grafo, encontrar un conjunto de vértices del menor tamaño posible tal que cada eje sea incidente en, al menos, uno de los vértices elegidos.
 - **Maximum clique**: dado un grafo, encontrar un conjunto de vértices de tamaño máximo tal que todos estén conectados entre sí
- ¿Cómo se podría utilizar un algoritmo para el problema de decisión del coloreado de grafos para determinar el **número cromático** de un grafo?

Parte V

Quantum Annealing

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>



Resolviendo problemas de optimización con D-Wave

- Utilizar los quantum annealers de D-Wave es muy sencillo
 - Por ejemplo, tomemos el hamiltoniano

$$Z_0 Z_1 + Z_0 Z_2$$

- Lo podemos definir con el siguiente código:

Resolviendo problemas de optimización con D-Wave

- Para ejecutarlo, podemos usar las siguientes instrucciones:

```
from dwave.system import DWaveSampler  
from dwave.system import EmbeddingComposite  
sampler = EmbeddingComposite(DWaveSampler())  
result = sampler.sample(problem, num_reads=10)  
print(result)
```

- Obtendremos algo como:

```
    0   1   2   energy   num_oc.   chain_.  
0  +1  -1  -1    -2.0        6      0.0  
1  -1  +1  +1    -2.0        4      0.0  
['SPIN', 2 rows, 10 samples, 3 variables]
```

Trabajando con QUBO en D-Wave

- Podemos ejecutar problemas QUBO en D-Wave usando `dimod.BINARY` (las soluciones vendrán dadas como ceros y unos)
- También podemos definir directamente problemas declarando:
 - Variables
 - Función objetivo
 - Restricciones
- Por ejemplo, un problema como

$$\begin{aligned} \text{Minimize} \quad & -5x_0 + 3x_1 - 2x_2 \\ \text{subject to} \quad & x_0 + x_2 \leq 1, \\ & 3x_0 - x_1 + 3x_2 \leq 4 \\ & x_j \in \{0, 1\}, \quad j = 0, 1, 2 \end{aligned}$$

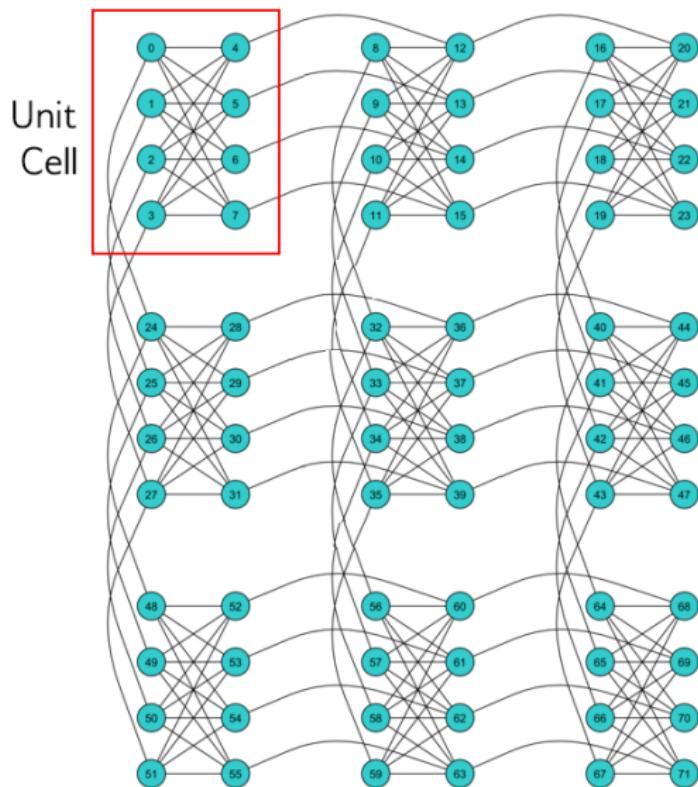
- Se aplican automáticamente las transformaciones que hemos estudiado

Los annealers de D-Wave

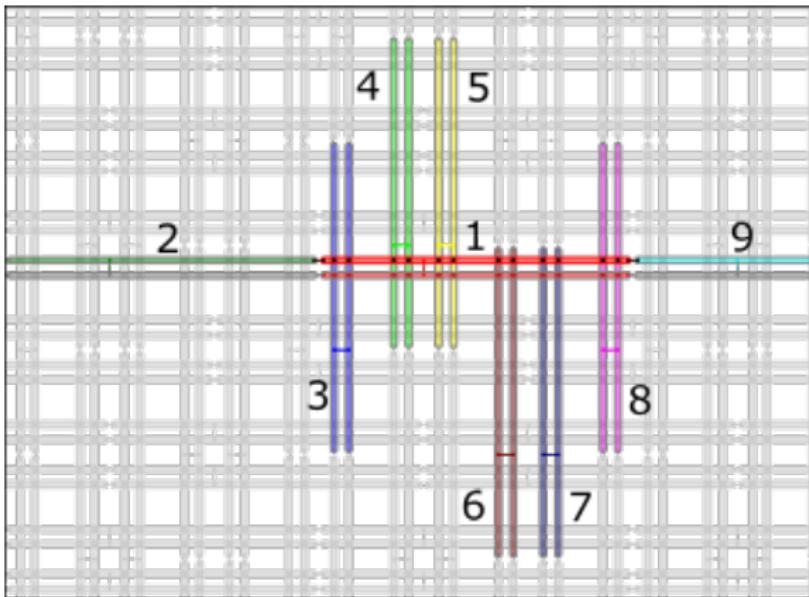
- D-Wave ofrece distintos annealers a través de Leap:
 - DWAVE 2000Q: 2048 qubits, topología Chimera (retirado)
 - Advantage System: 5760 qubits, topología Pegasus
 - Advantage 2 (prototipo): 1248 qubits, topología Zephyr
 - Hybrid solvers
- La topología es extremadamente importante:
- **NO** hay conexiones completas
- En algunos casos, no hay conexiones 3 a 3 (Chimera)
- Se necesitan cadenas para representar variables lógicas



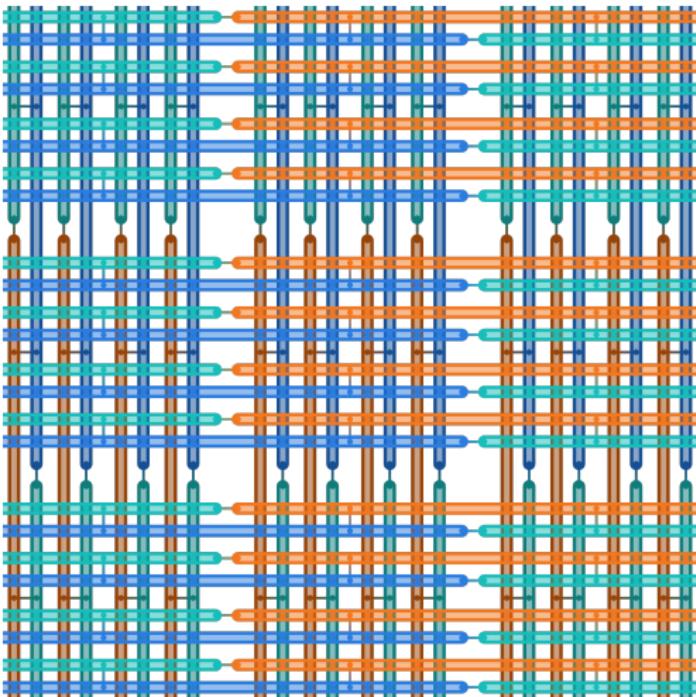
La topología Chimera



La topología Pegasus



La topología Zephyr

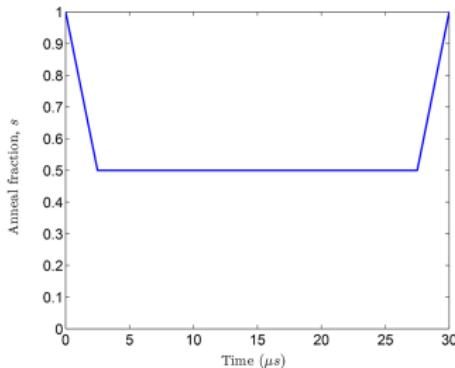
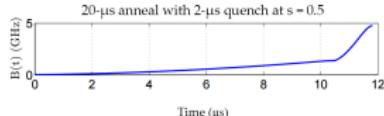
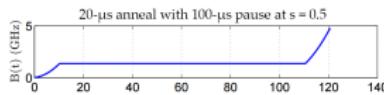
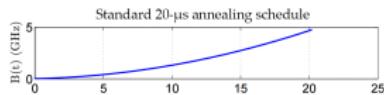


Embeddings

- Si la topología no soporta directamente las conexiones de las variables lógicas del problema, hay que usar un embedding
- Encontrar un embedding es un problema NP-hard
- El software de D-Wave proporciona distintos métodos (heurísticos) para encontrar embeddings:
 - EmbeddingComposite (calcula un embedding en cada ejecución)
 - AutoEmbeddingComposite (primero intenta NO utilizar embedding)
 - FixedEmbeddingComposite (utiliza un embedding pasado por parámetro)
 - LazyEmbeddingComposite (calcula un embedding solo la primera vez y lo almacena)

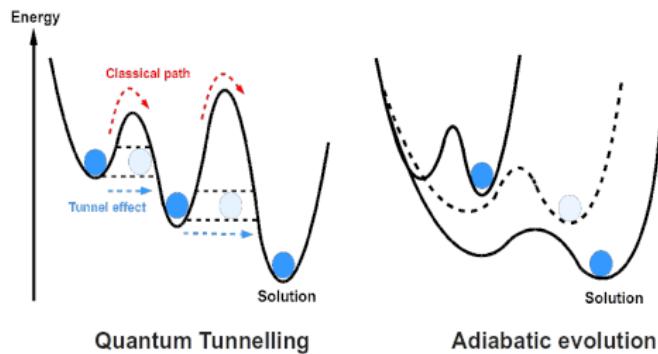
Parámetros del annealing

- Para realizar el annealing, se pueden tener en cuenta los siguientes parámetros:
 - Annealing time
 - Annealing schedule
 - Forward
 - Reverse
 - Coupling strength
 - Lagrange multiplier (en problemas con restricciones)



Hybrid solvers

- El software de D-Wave ofrece solvers clásicos:
 - Exact Solver
 - Simulated Annealing
 - Steepest Descent Solver
 - Tabu Solver
- Y solvers híbridos, que combinan recursos clásicos y cuánticos
- Para los solvers híbridos, es interesante la propiedad `quota_conversion_rate`



Ejercicios

- Crear un problema QUBO o Ising (por ejemplo, una instancia de MaxCut) con un número grande de variables (100 o más)
- Ejecutarlo en:
 - Advantage System
 - Advantage System 2
 - Simulated annealing y otros solvers clásicos e híbridos
- Comparar:
 - Calidad de las soluciones
 - Tiempo de ejecución
- Repetir, incrementando el número de variables

Parte VI

QAOA: Quantum Approximate
Optimization Algorithm

Discretizando la computación adiabática

- En el modelo adiabático, partimos de un ground state y llevamos a cabo una evolución con

$$H(t) = A(t)H_0 + B(t)H_1$$

- H_1 es, por ejemplo, un hamiltoniano de Ising
- Podemos intentar discretizar la evolución mediante

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

en el intervalo $[t_c, t_c + \Delta t]$

- Tendríamos

$$\left(\prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle,$$

con $t_m = m\frac{\Delta t}{T}$ y $p = \frac{T}{\Delta t}$

Discretizando la computación adiabática

- Realizamos una aproximación más:

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)} \approx e^{i\Delta tA(t_c)H_0} e^{i\Delta tB(t_c)H_1}$$

- Finalmente, tendremos

$$\prod_{m=0}^p e^{i\Delta tA(t_m)H_0} e^{i\Delta tB(t_m)H_1} |\psi_0\rangle$$

- Esta es la inspiración para el QAOA, que considera estados parametrizados de la forma

$$|\beta, \gamma\rangle = e^{i\beta_p H_0} e^{i\gamma_p H_1} \dots e^{i\beta_2 H_0} e^{i\gamma_2 H_1} e^{i\beta_1 H_0} e^{i\gamma_1 H_1} |\psi_0\rangle$$

con $p \geq 1$

QAOA: el algoritmo

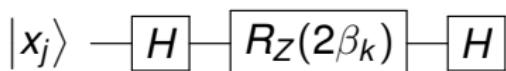
- QAOA es un método híbrido en el que se usan tanto un ordenador clásico como uno cuántico
- Los pasos son:
 - 1 Elegir un valor para p y unos ángulos iniciales β, γ
 - 2 Preparar el estado $|\beta, \gamma\rangle$
 - 3 Estimar la energía $E(\beta, \gamma) = \langle\beta, \gamma| H_1 |\beta, \gamma\rangle$ 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 $|\beta, \gamma\rangle$

- Tomamos $H_0 = \sum X_i$
- El estado $|\psi_0\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} |x\rangle$ se puede preparar con puertas de Hadamard
- Se cumple que

$$e^{i\beta_k H_0} = e^{-i\beta_k \sum_{j=0}^{n-1} X_j} = \prod_{j=0}^{n-1} e^{-i\beta_k X_j}$$

- Cada $e^{-i\beta_k X_j}$ es una rotación $R_X(2\beta_k)$ o equivalentemente



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

- Tomamos $H_1 = -\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j$
- Entonces, tenemos

$$e^{i\gamma_l H_1} = e^{-i\gamma_l (\sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j)} = \prod_{j,k} e^{-i\gamma_l J_{jk} Z_j Z_k} \prod_j e^{-i\gamma_l h_j Z_j}.$$

- Podemos generalizar y considerar casos de la forma

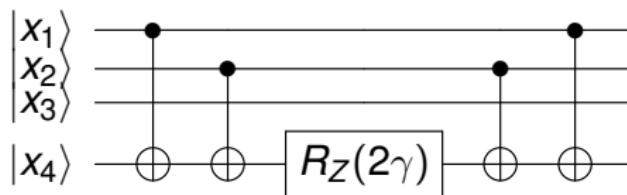
$$e^{-i\gamma_k Z_{i_1} \cdots Z_{i_j}}$$

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:

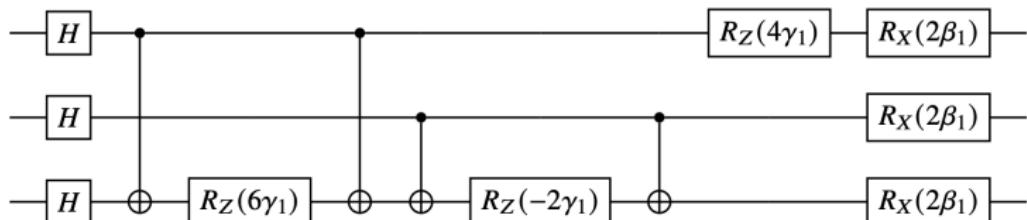


Ejemplo de circuito QAOA completo

- Consideramos el caso

$$3Z_0Z_2 - Z_1Z_2 + 2Z_0$$

- Obtenemos el siguiente circuito para $p = 1$



Estimando la energía

- Estimar la energía en el QAOA es muy sencillo
- Suponemos que partimos de un problema QUBO con función de coste $C(x)$, de donde obtenemos H_1
- 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

$$\langle \beta, \gamma | H_1 | \beta, \gamma \rangle = \sum_{x \in \{0,1\}^n} |a_x|^2 C(x)$$

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

QUBO y HOBO

- Consideramos la fórmula booleana

$$(x_0 \vee \neg x_1 \vee x_2) \wedge (\neg x_0 \vee x_1 \vee \neg x_2) \wedge (x_0 \vee x_1 \vee x_2)$$

- A partir de ella, podemos obtener el polinomio

$$(1 - x_0)x_1(1 - x_2) + x_0(1 - x_1)x_2 + (1 - x_0)(1 - x_1)(1 - x_2)$$

- La fórmula original es satisfacible si y solo si el mínimo del polinomio es 0
- Es un polinomio de grado 3
- Tenemos un problema HOBO (Higher Order Binary Optimization) o PUBO (Polynomial Unconstrained Binary Optimization)
- Lo podemos resolver también con QAOA (nuestros desarrollos se aplican también a estos casos)

Algunos comentarios sobre el QAOA

- Para que el proceso sea eficiente, H_1 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 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
- Cuando $p \rightarrow \infty$, el estado de mínima energía de $|\beta, \gamma\rangle$ tiende al estado de mínima energía de H_1
- En algunos casos, se han obtenido resultados interesantes incluso para p pequeño

Nuevos resultados sobre la eficacia del QAOA

The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model

✉ Joao Basso¹, Edward Farhi^{1,2}, ✉ Kunal Marwaha³,
✉ Benjamin Villalonga¹, and ✉ Leo Zhou⁴

¹ Google Quantum AI, Venice, CA 90291

² Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139

³ Department of Computer Science, University of Chicago, Chicago, IL 60637

⁴ Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, CA 91125

October 26, 2021

Abstract

The Quantum Approximate Optimization Algorithm (QAOA) finds approximate solutions to combinatorial optimization problems. Its performance monotonically improves with its depth p . We apply the QAOA to MaxCut on large-girth D -regular graphs. We give an iterative formula to evaluate performance for any D at any depth p . Looking at random D -regular graphs, at optimal parameters and as D goes to infinity, we find that the $p = 11$ QAOA beats all classical algorithms (known to the authors) that are free of unproven conjectures. While the iterative formula for these D -regular graphs is derived by looking at a single tree subgraph, we prove that it also gives the ensemble-averaged performance of the QAOA on the Sherrington-Kirkpatrick (SK) model. Our iteration is a compact procedure, but its computational complexity grows as $O(p^2 4^p)$.

Variantes de QAOA

PHYSICAL REVIEW RESEARCH 4, 033029 (2022)

www.nature.com/scientificreports/

Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer

Linghua Zhu^{1,2}, Ho Lun Tang,¹ George S. Biamonte³, F. A. Calderon-Vargas,¹

Nicholas J. Mayhall,¹ Edwin Barnes,¹ and Sophia E. Economou,^{1,4}

¹Department of Physics, Virginia Tech, Blacksburg, Virginia 24061, USA

²Department of Chemistry, Virginia Tech, Blacksburg, Virginia 24061, USA

(Received 25 May 2020; revised 22 December 2020; accepted 17 June 2022; published 11 July 2022)

The quantum approximate optimization algorithm (QAOA) is a hybrid variational quantum-classical algorithm that solves combinatorial optimization problems. While there is evidence suggesting that the fixed form of the standard QAOA Ansatz is not optimal, there is no systematic approach for finding better Ansätze. We address this problem by developing an iterative version of QAOA that performs gradient descent. This Ansatz can be applied to specific combinatorial problems. We simulate the algorithm on a class of Max-Cut graph problems and show that it converges much faster than the standard QAOA, while simultaneously reducing the required number of CNOT gates and optimization parameters. We provide evidence that this speedup is connected to the concept of short-cut adiabaticity.

DOI: 10.1103/PhysRevResearch.4.033029

I. INTRODUCTION

Many important computationally hard combinatorial optimization problems such as Max-Cut, graph coloring, traveling salesman, and scheduling problems [1–3] are known to be NP-hard to bring quantum computers ground state provide the solutions. One can in principle solve these optimization problems on a quantum computer by initializing the quantum device in the ground state of a Hamiltonian that is easy to prepare and adiabatically tuning the latter into the problem Hamiltonian. In a digital quantum computer, this translates into a Trotterized version of the adiabatic evolution operator, which is the alternating product of two evolution operators corresponding to the Trotterizer and the problem (Ising) Hamiltonians. In the limit of an infinite product, this Trotterized form becomes exact.

QAOA is a hybrid quantum-classical variational algorithm that uses a finite-order version of this evolution operator to prepare wavefunction on a quantum processor [4–6]. QAOA is performed by iteratively minimizing the expectation value of the Ising Hamiltonian with respect to the parameters in the Ansatz. The quantum processor is also used to measure energy expectation values, while the optimization is done on a classical computer. There has been a lot of

QAOA may provide a significant quantum advantage over classical algorithms [14–16], and that it is computationally universal [17,18].

Despite these advances, there are limitations and potential issues with this algorithm. The performance improves with the number of alternating layers in the Ansatz, but the latter is limited by coherence times in existing and near-term quantum processors. Moreover, more layers implies more operations, which increases the cost function evaluated by the classical optimizer [19]. Furthermore, Ref. [20] points out that the locality and symmetry of QAOA can also severely limit its performance. These issues can be attributed to, or are at least exacerbated by, the form of the QAOA Ansatz. In particular, short-depth QAOA is not really the digitized version of the Trotterized problem Hamiltonian at all. The Ansatz, and its cost, should not be expected to perform well, usually, and even well. A short-depth Ansatz that is further taskified to a given combinatorial problem could therefore address the issues with the standard QAOA Ansatz. However, identifying such an alternative is a highly nontrivial problem given the vast space of possible Ansätze.

In this work, we propose an iterative version of QAOA termed adaptive derivative assembled problem tailored-

scientific reports

OPEN

Multi-angle quantum approximate optimization algorithm

Rebekah Herrman^{1,2}, Phillip C. Lotshaw^{2,3}, James Ostrowski¹, Travis S. Humble², and George Siopsis^{1*}

The quantum approximate optimization algorithm (QAOA) generates an approximate solution to combinatorial optimization problems using a variational circuit circuit defined by parameterized layers of quantum evolution. In theory, the algorithm can find an exact ground state with just two gate sets and one circuit depth. Under ideal performance in practice, however, we investigate a multi-angle ansatz for QAOA that reduces circuit depth and improves the approximation ratio by increasing the number of classical parameters. Even though the number of parameters increases, our results indicate that good parameters can be found in polynomial time for a test dataset we consider. This new ansatz is able to find an approximation to the ground state with fewer layers and fewer parameters over QAOA. The theoretical performance is lower bounded by the conventional ansatz, and we present empirical results for graphs on eight vertices that one layer of the multi-angle ansatz is comparable to three layers of the traditional ansatz on MaxCut problems. Similarly, multi-angle QAOA yields a higher approximation ratio than QAOA for a collection of MaxCut instances on fifty and one-hundred vertex graphs. Most of the optimal parameters are forced to be zero, so their associated circuit can be removed from the circuit, further decreasing the circuit depth. These results indicate that multi-angle QAOA requires shallower circuits to solve problems than QAOA, making it more viable for near-term intermediate-scale quantum devices.

Among several quantum algorithms implemented on noisy intermediate-scale quantum (NISQ) devices^{1,2}, the quantum approximate optimization algorithm (QAOA) offers a promising route to approximately solve combinatorial optimization problems such as Max-Cut, Max-Satisfiability, and Max-k-Clique. QAOA takes a set of classical parameters to optimize the cost function and an initial state for a quantum state prepared by a sequence of operators acting on a known initial state. Variations to the original algorithm include alternative operators and initial states³, while many classical aspects such as the parameter optimization and problem encoding are shared. However, the original QAOA has been shown to have poor performance, especially for QAOA requiring large numbers of qubits and deep circuits⁴. For example, a recent study has developed a systematic set of parameters that are argued to require $p = 30$ layers of QAOA to reach performance comparable to the conventional Goemans-Williamson algorithm on MaxCut⁵, while another paper has argued that hundreds of qubits and many more layers are required to reach similar performance⁶. Not only does the circuit depth and affects the fidelity of the prepared quantum state so the performance that can be achieved from noise-term quantum computers at these depths is questionable.

One way to improve the performance is to increase the number of classical parameters introduced in each layer, a variation that we term multi-angle QAOA (ma-QAOA). This approach was originally briefly introduced^{7,8}. Increasing the number of classical parameters allows for finer-grain control over the optimization of the cost function and the approximation ratio, which measures optimality relative to the known best solution. The main idea behind ma-QAOA is that the additional parameters, which correspond to a responding reduction in circuit depth preserves the critical resource of the quantum state. In addition, finding the absolute optimal angles is not necessary in order to see an improvement over QAOA.

Here, we quantify the advantages of using multiple parameters for each layer of QAOA. First, we prove that

Check for updates

Variantes de QAOA

A Review on Quantum Approximate Optimization Algorithm and its Variants

Kostas Blekos^{*1}, Dean Brand², Andrea Ceschini³, Chiao-Hui Chou⁴, Rui-Hao Li⁵, Komal Pandya⁶, and Alessandro Summa⁷

¹*Department of Physics, School of Natural Sciences, University of Patras, Patras 26504, Greece; email: mplekos@upatras.gr*

²*Department of Physics, Stellenbosch University, Stellenbosch 7600, South Africa; email: deanbrand@protonmail.com*

³*Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome 00184, Italy; email: andrea.ceschini@uniroma1.it*

⁴*Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei 10617, Taiwan; email: s110112020@gmail.com*

⁵*Department of Physics, Case Western Reserve University, Cleveland, Ohio 44106, USA; email: rzd527@case.edu*

⁶*Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801106, India; email: komal.lame@gmail.com*

⁷*School of Physics, Trinity College Dublin, Dublin 2, Ireland; email: summera@tcd.ie*

June 28, 2023

Abstract

The Quantum Approximate Optimization Algorithm (QAOA) is a highly promising variational quantum algorithm that aims to solve combinatorial optimization problems that are classically intractable. This comprehensive review offers an overview of the current state of QAOA, encompassing its performance analysis in diverse scenarios, its applicability across various problem instances, and considerations of hardware-specific challenges such as error susceptibility and noise resilience. Additionally, we conduct a comparative study of selected QAOA extensions and variants, while exploring future prospects and directions for the algorithm. We aim to provide insights into key questions about the algorithm, such as whether it can outperform classical algorithms and under what circumstances it should be used. Towards this goal, we offer specific practical points in a form of a short guide. **Keywords:** Quantum Approximate Optimization Algorithm (QAOA), Variational Quantum Algorithm (VQA), Quantum Optimization, Combinatorial Optimization Problem, NSO, Algo-

Ejercicios

- Formular como QUBO/HOBO el problema del **Set Cover**: Dada una colección de conjuntos $S = \{S_1, \dots, S_n\}$, encontrar el menor número de conjuntos cuya unión incluye todos los elementos incluidos en al menos uno de los S_i .
- Tomar un problema sencillo (por ejemplo, una instancia de Max Cut en un grafo no muy grande)
- Ejecutarlo
 - Con QAOA y distintos valores de p en simuladores
 - Con quantum annealing
- Comparar:
 - Calidad de las soluciones (en función de p)
 - Tiempos (solo QAOA)

Parte VII

VQE: 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

$$X = HZH$$

y

$$Y = SHZHS^\dagger$$

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

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

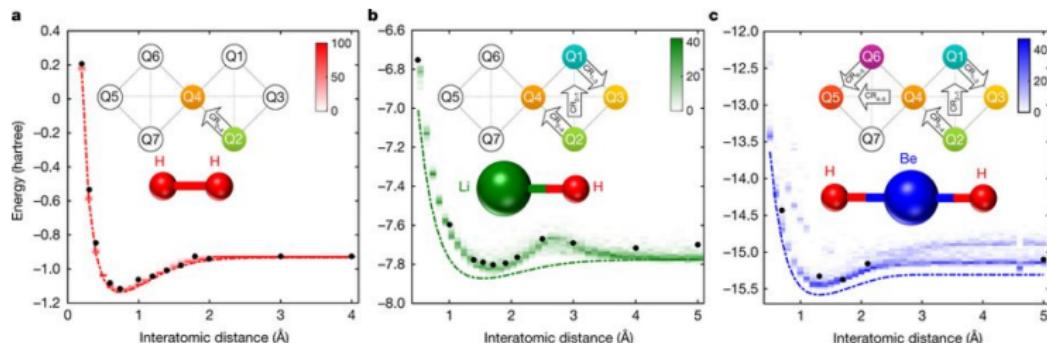
- Por tanto, podemos medir la energía en $Z_1 Z_3 Z_4$ de $(H \otimes I \otimes HS^\dagger \otimes I) | \psi \rangle$ porque

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

- 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

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

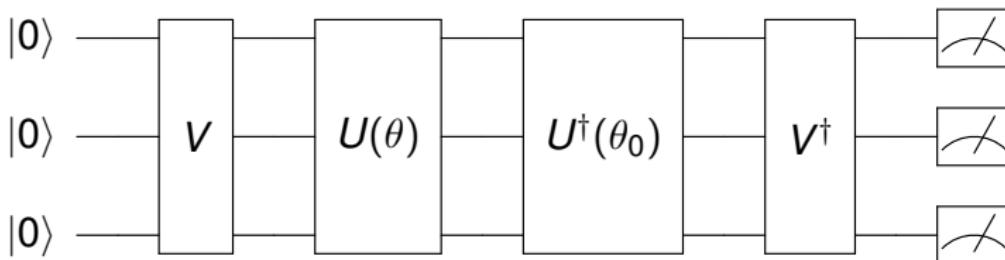
- Si C es mayor que la diferencia entre la energía mínima y el siguiente nivel de energía de H_f , entonces $|\psi_0\rangle$ ya no es estado de mínima energía de H'_f

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

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

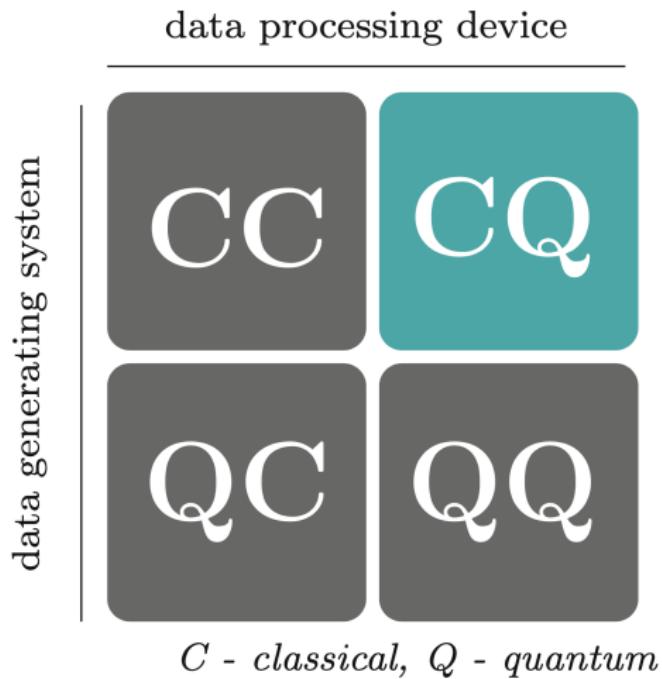
donde V es una unitaria tal que $V|0\rangle = |\psi\rangle$



Parte VIII

Quantum Machine Learning

De qué hablamos cuando hablamos de Quantum Machine Learning

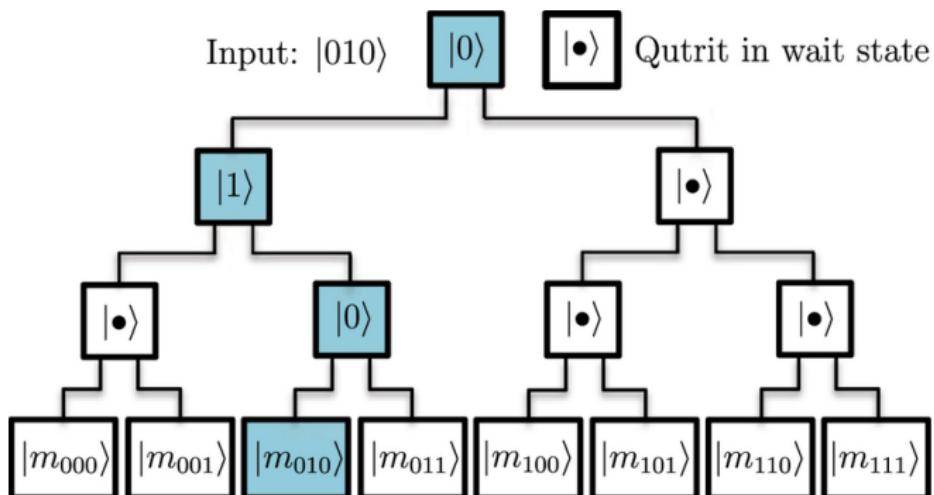


QBLAS: The Quantum Basic Linear Algebra Subroutines

- 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



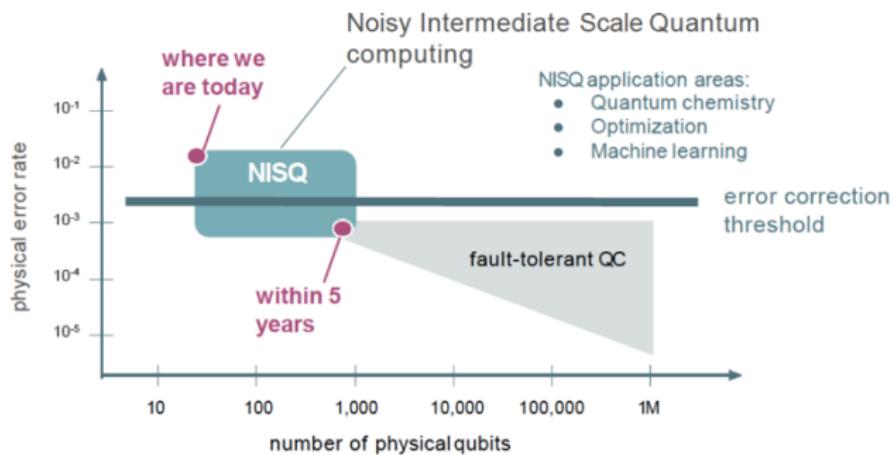
QML translacional y speedups

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



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

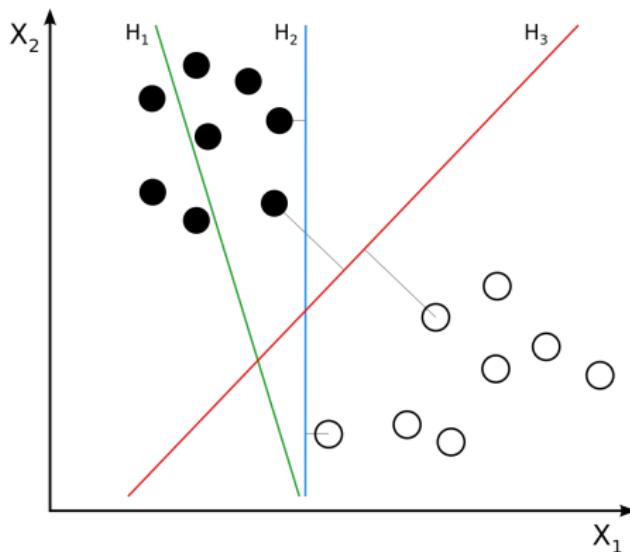


Parte IX

Quantum Support Vector Machines

Support Vector Machines

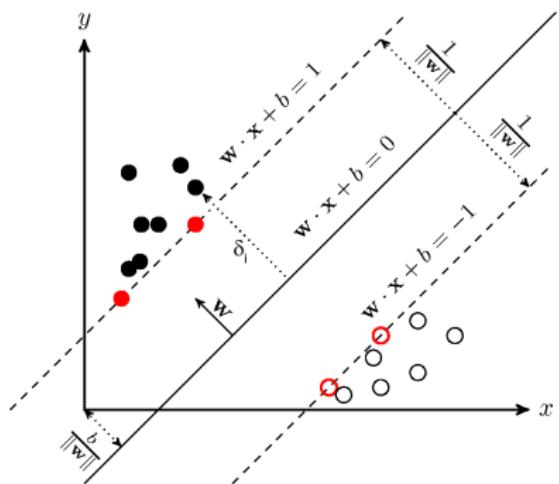
- Las Support Vector Machines (SVM) son un algoritmo (clásico) de clasificación muy popular
- La idea en la que se basan es encontrar un hiperplano separador con el mayor margen posible



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

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



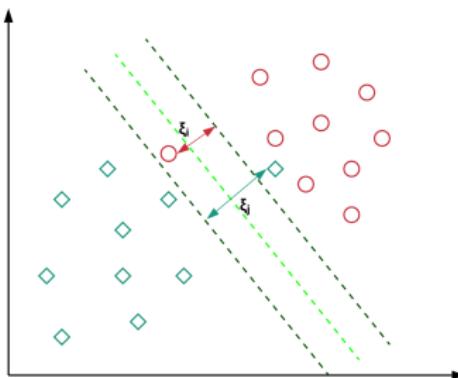
El caso del “margen blando”

- 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

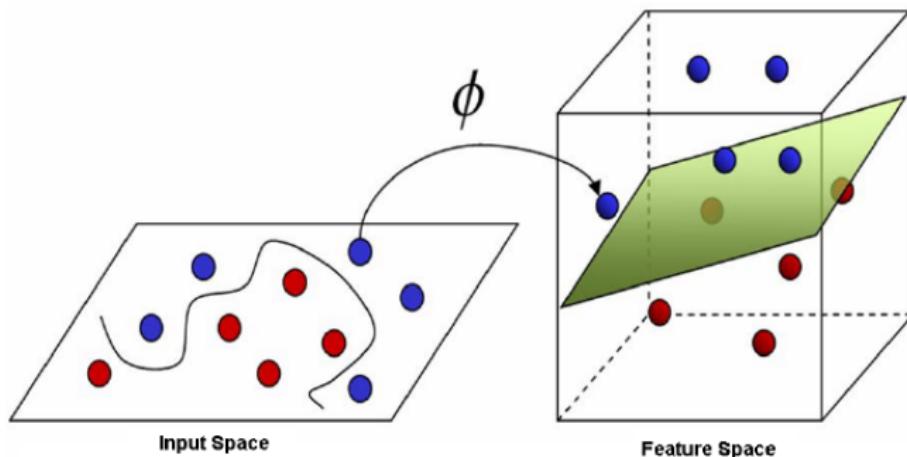
$$w = \sum_i \alpha_i y_i x_i$$

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

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

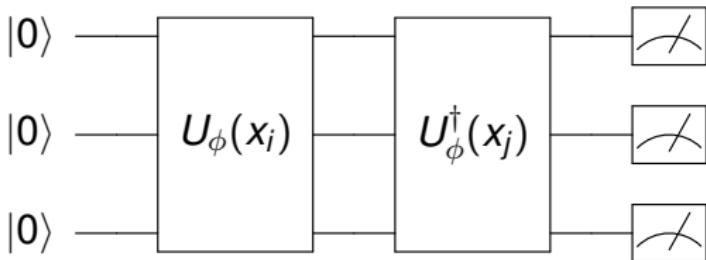
y para clasificar un punto x sólo necesitamos calcular

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

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

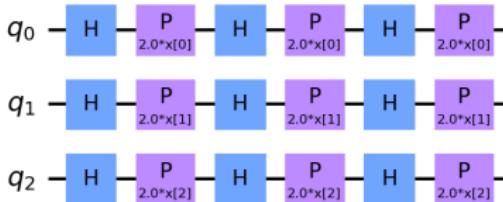
Calculando kernels con ordenadores cuánticos

- En 2019, Havlíček, Cárcoles, 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 $U_\phi(x_i)$ tal que $U_\phi(x_i)|0\rangle = |\phi(x_i)\rangle$
- 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

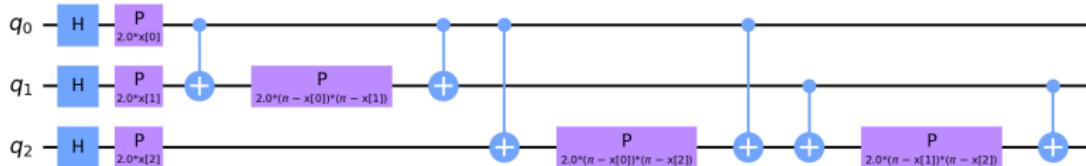


Ejemplos de feature maps

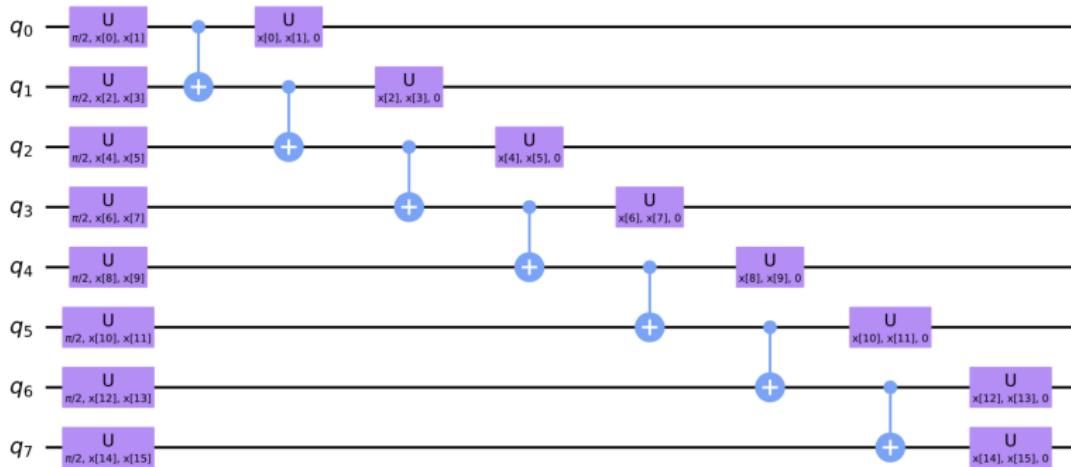
- Codificación en amplitud:
 - Partimos de un vector de números reales $(x_0, x_2, \dots, x_{m-1})$
 - Puede ser necesario normalizarlo y “rellenarlo” hasta tamaño $2^n \geq m$
 - La codificación se hace mediante un circuito polinomial en m que prepara $\sum_{i=0}^{2^n-1} x_i |i\rangle$
- Z-feature map (codificación en ángulo)



- ZZ-feature map

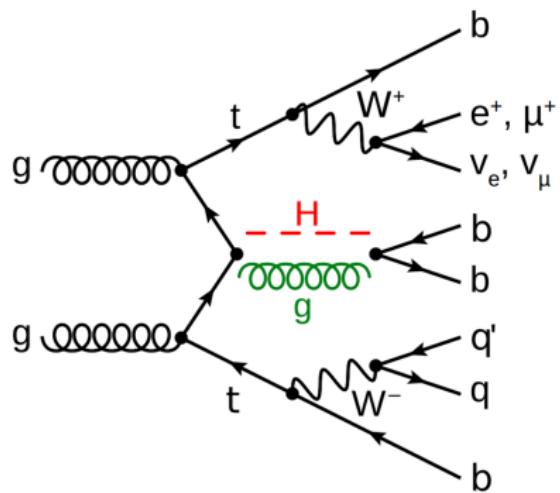


Otro ejemplo de feature map



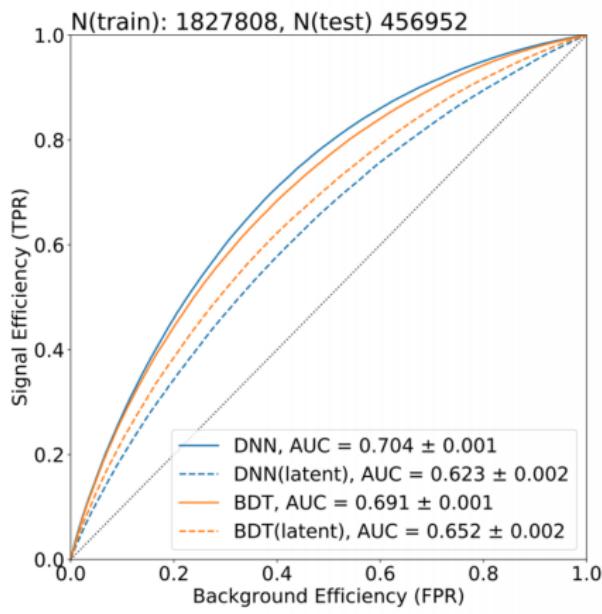
Un problema de clasificación en física de altas energías

- Producción del bosón de Higgs
- 67 variables físicas



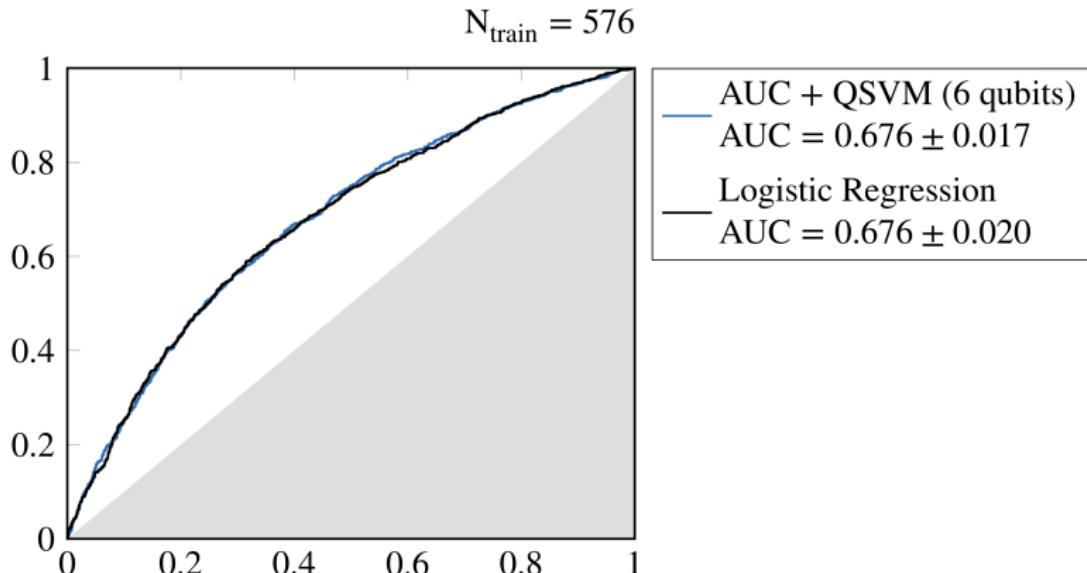
Resultados con métodos de machine learning clásico

- 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



Higgs analysis with quantum classifiers

Vasilis Belis^{1,}, Samuel González-Castillo², Christina Reissel¹, Sofia 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

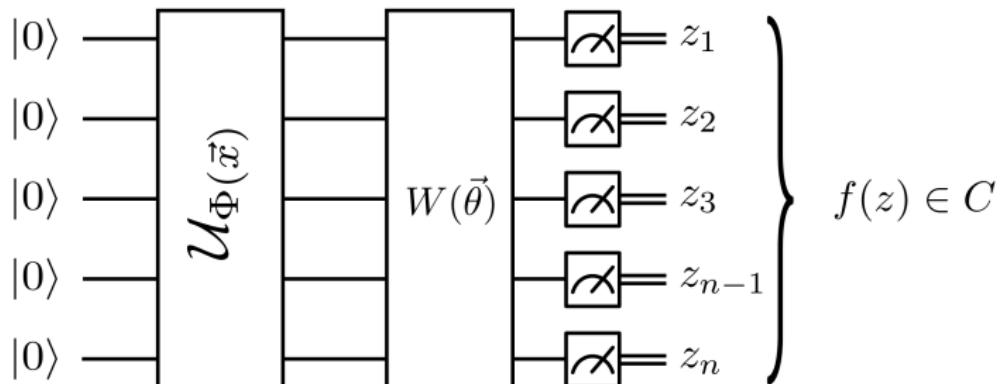
Abstract. We have developed two quantum classifier models for the $t\bar{t}H$ classification problem, both of which fall into the category of hybrid quantum-classical algorithms for Noisy Intermediate Scale Quantum devices (NISQ). Our results, along with other studies, serve as a proof of concept that Quantum Machine Learning (QML) methods can have similar or better performance, in specific cases of low number of training samples, with respect to conventional 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 X

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: una “feature map”, que traslada los datos de entrada a un espacio de Hilbert, seguida 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 $f(x, \theta_{min})$ (y, quizás, un umbral)

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)

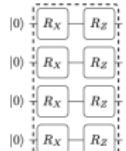
- Este es el caso, por ejemplo, si U es una rotación de un qubit
- Entonces, se verifica que

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

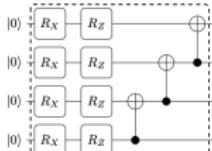
donde $s = \frac{\pi}{4r}$

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

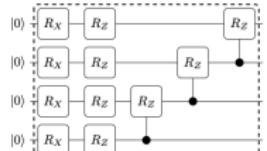
Cómo elegir formas variacionales



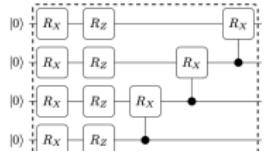
Circuit 1



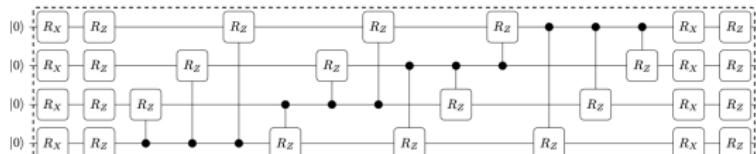
Circuit 2



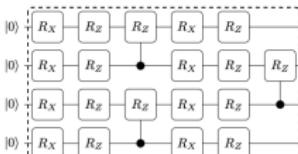
Circuit 3



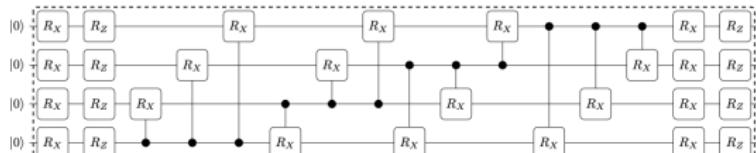
Circuit 4



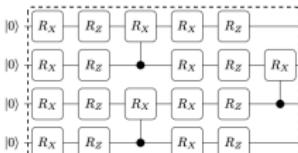
Circuit 5



Circuit 7



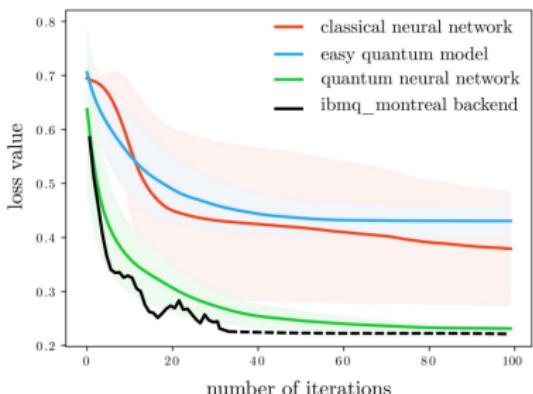
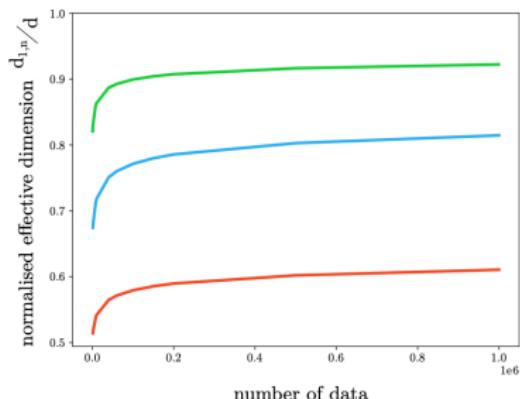
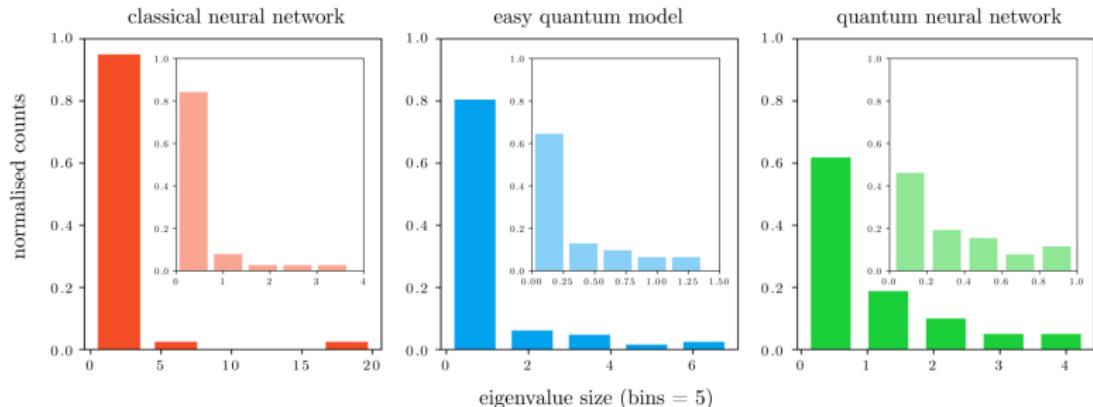
Circuit 6



Circuit 8

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

The power of Quantum Neural Networks



Aprendiendo con pocos datos

nature communications



Article

<https://doi.org/10.1038/s41467-022-32550-3>

Generalization in quantum machine learning from few training data

Received: 12 April 2022

Matthias C. Caro^{1,2}, Hsin-Yuan Huang ^{3,4}, M. Cerezo^{5,6}, Kunal Sharma⁷,

Accepted: 4 August 2022

Andrew Sornborger^{5,8}, Lukasz Cincio⁹ & Patrick J. Coles ⁹

Published online: 22 August 2022

Check for updates

Modern quantum machine learning (QML) methods involve variationally optimizing a parameterized quantum circuit on a training data set, and subsequently making predictions on a testing data set (i.e., generalizing). In this work, we provide a comprehensive study of generalization performance in

RESEARCH

RESEARCH ARTICLE

QUANTUM COMPUTING

Quantum advantage in learning from experiments

Hsin-Yuan Huang^{1,2*}, Michael Broughton³, Jordan Cotler^{4,5}, Sitan Chen^{6,7}, Jerry Li⁸, Masoud Mohseni³, Hartmut Neven³, Ryan Babbush³, Richard Kueng⁹, John Preskill^{8,2,30}, Jarrod R. McClean^{1,6}

Quantum technology promises to revolutionize how we learn about the physical world. An experiment that processes quantum data with a quantum computer could have substantial advantages over conventional experiments in which quantum states are measured and outcomes are processed with a classical computer. We proved that quantum machines could learn from exponentially fewer experiments than the number required by conventional experiments. This exponential advantage is shown for predicting properties of physical systems, performing quantum principal component analysis, and learning about physical dynamics. Furthermore, the quantum resources needed for achieving an exponential advantage are quite modest in some cases. Conducting experiments with 40 superconducting qubits and 1300 quantum gates, we demonstrated that a substantial quantum advantage is possible with today's quantum processors.

quantum-enhanced experiments have a similar exponential advantage in a related scenario shown in Fig. 1C, in which the goal is to learn about a quantum process \mathcal{E} rather than a quantum state p . Advantages of entangling measurements over single-copy measurements have been noticed previously (*11, 12*), but our work goes much further by establishing an advantage that scales exponentially with system size.

Building on previous observations (*8, 13*), we proved that for a task that entails acquiring information about a large number of noncommuting observables, quantum-enhanced experiments could have an exponential advantage even when the measured quantum state is unentangled. Our work substantially reduces the complexity of the required quantum-enhanced experiments, improving the prospects for near-term implementation. By performing experiments with up to 40 superconducting qubits, we showed that this

El problema de los barren plateaus

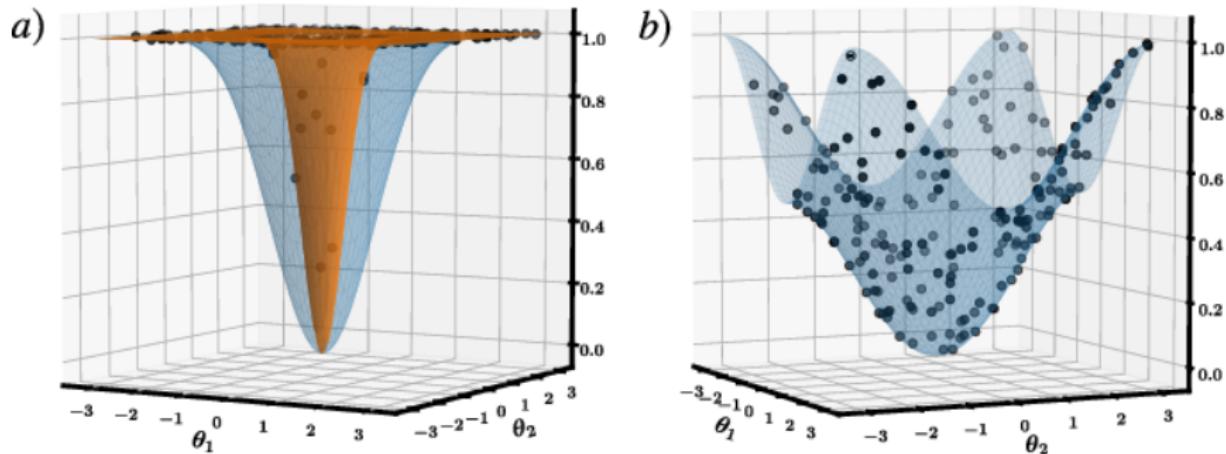
ARTICLE

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 (2)

Subtleties in the trainability of quantum machine learning models

Supanut Thanansip^{1,2}, Samson Wang^{1,3}, Nhat A. Nghiem^{1,4}, Patrick J. Coles¹ and M. Cerezo^{5,6}

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

²*Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2 117543, Singapore*

³*Imperial College London, London, UK*

⁴*Department of Physics and Astronomy, State University of New York at Stony Brook, Stony Brook, New York 11794-3800, USA*

⁵*Information Sciences, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

⁶*Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

A new paradigm for data science has emerged, with quantum data, quantum models, and quantum computational devices. This field, called Quantum Machine Learning (QML), aims to achieve a speedup over traditional machine learning for data analysis. However, its success usually hinges on efficiently training the parameters in quantum neural networks, and the field of QML is still lacking theoretical scaling results for their trainability. Some trainability results have been proven for a closely related field called Variational Quantum Algorithms (VQAs). While both fields involve training a parametrized quantum circuit, there are crucial differences that make the results for one setting not readily applicable to the other. In this work we bridge the two frameworks and show that gradient scaling results for VQAs can also be applied to study the gradient scaling of QML models. Our results indicate that features deemed detrimental for VQA trainability can also lead to issues such as barren plateaus in QML. Consequently, our work has implications for several QML proposals in the literature. In addition, we provide theoretical and numerical evidence that QML models exhibit further trainability issues not present in VQAs, arising from the use of a training dataset. We refer to these as dataset-induced barren plateaus. These results are most relevant when dealing with classical data, as here the choice of embedding scheme (i.e., the map between classical data and quantum states) can greatly affect the gradient scaling.

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

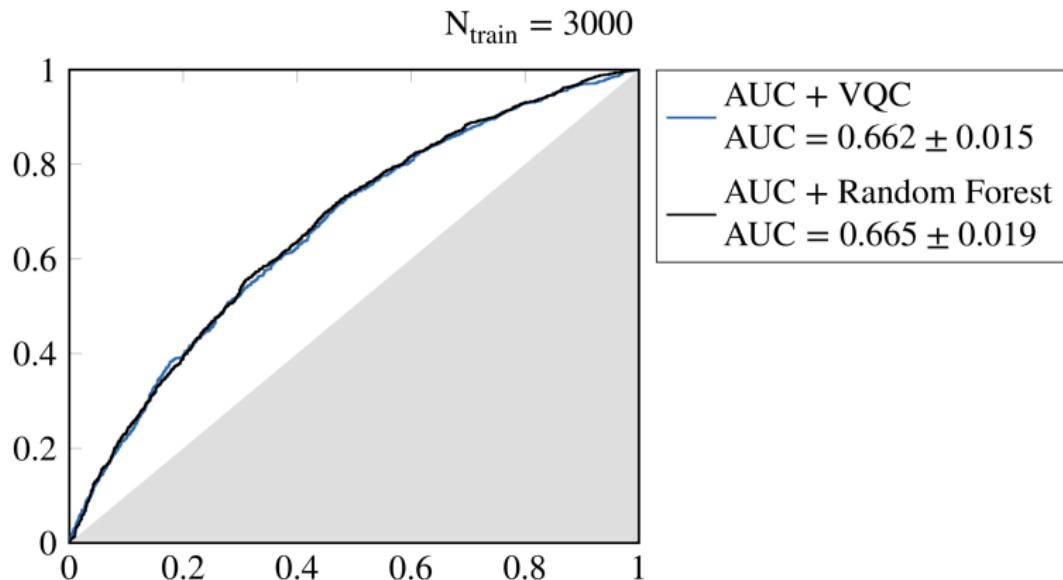


(Received 12 March 2021; revised 13 July 2021; accepted 2 August 2021; published 15 October 2021)

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 layers followed by a fully connected layer. For this type of architecture, it is not clear whether the barren plateau problem is avoided or mitigated. In this work, we show that the barren plateau problem is avoided for QCNNs with a single convolutional layer. Specifically, we prove that the cost function is bounded away from zero for all parameter values, which implies that the gradients are bounded away from zero. This result holds for QCNNs with a single convolutional layer and a single fully connected layer. The proof relies on a combination of mathematical analysis and numerical simulations. The results are also supported by theoretical arguments based on the properties of the convolutional layer and the fully connected layer. The findings are relevant for the development of QCNNs and their potential applications in quantum computing.

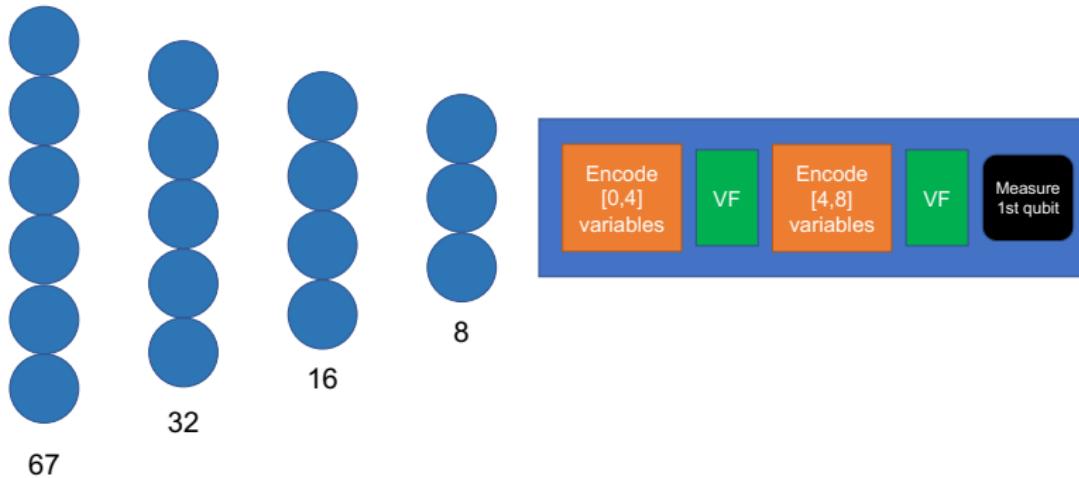
Resultados en el problema de clasificación de HEP

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



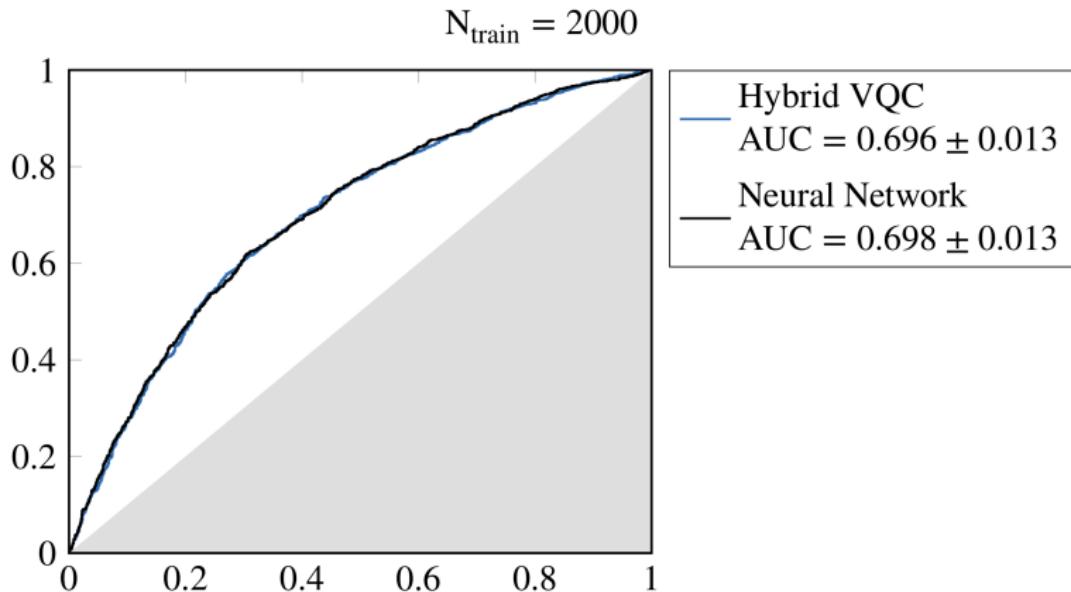
Arquitectura híbrida

- 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 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

