

# **Una pequeña introducción a la optimización cuántica y al quantum machine learning**

**Elías F. Combarro (Universidad de Oviedo)**  
[efernandezca@uniovi.es](mailto:efernandezca@uniovi.es)

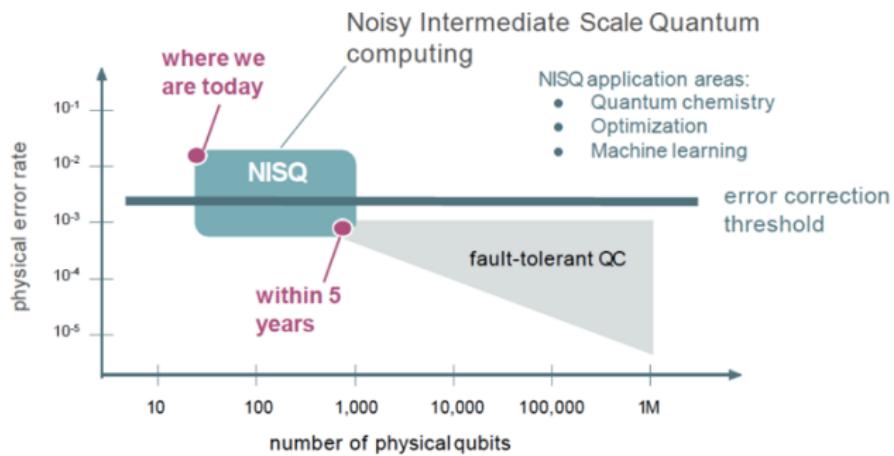
XIV Seminario de Invierno CAPAP-H, La Laguna (Tenerife)  
2 de febrero de 2023

# Parte I

Introducción: la era NISQ

# QC en los tiempos del NISQ

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



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



# Objetivos

- Estudiar algoritmos cuánticos para optimización combinatoria
  - Quantum annealing
- Estudiar algoritmos cuánticos para quantum machine learning
  - Quantum Support Vector Machines
  - Quantum Neural Networks
- Ejecutar algunos de estos algoritmos en:
  - Quantum annealers
  - Simuladores de circuitos cuánticos



# 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](http://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**

1st EDITION

**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**

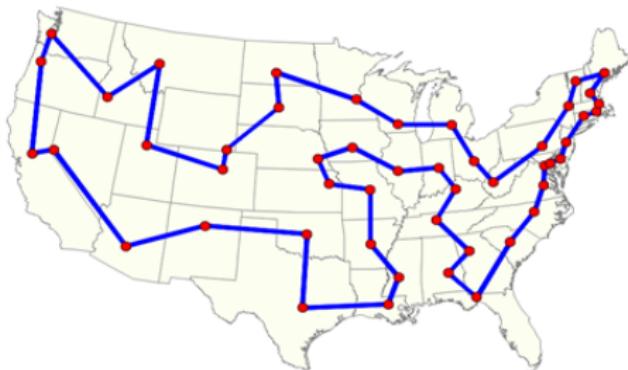
Foreword by Alberto Di Meglio,  
Head of Innovation - Coordinator CERN Quantum Technology Initiative

## Parte II

Optimización combinatoria:  
problemas QUBO

# 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

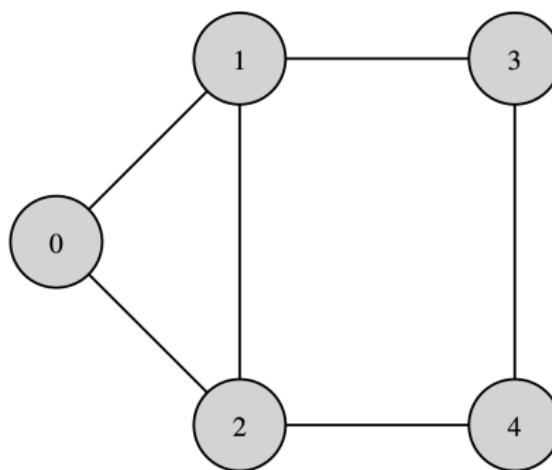
- Modelo de Ising: Minimizar

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

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

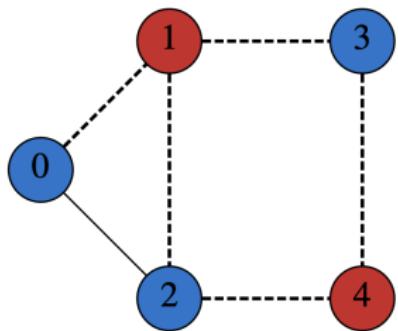
# El problema del corte máximo

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

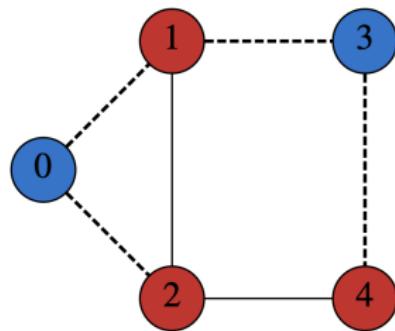


- Es un problema NP-hard

# 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

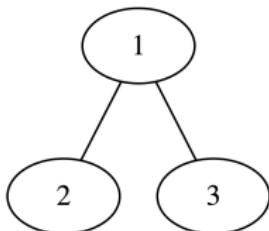
$$\text{Minimizar} \sum_{(i,j) \in E} Z_i Z_j$$

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

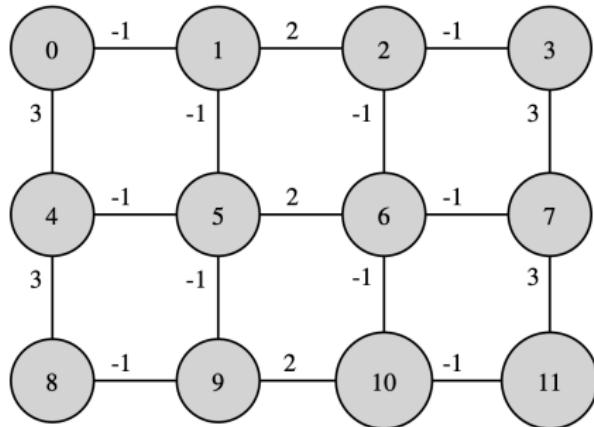
- Se tienen  $n$  partículas con spin, que interactúan entre sí con ciertas fuerzas de acoplamiento
- Su hamiltoniano es

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

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

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

# El maravilloso mundo de los hamiltonianos

- 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

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

- Nos apoyamos en el **principio variacional**

# 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

$$Z_i = 1 - 2x_i$$

# Binary Linear Programming

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

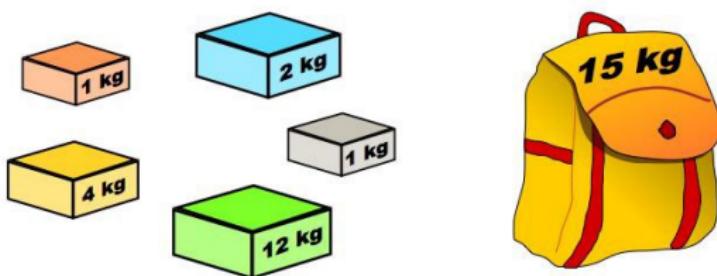
$$\begin{aligned} \text{Minimizar } & c_0x_0 + c_1x_1 + \dots + c_mx_m \\ \text{sujeto a } & Ax \leq b, \\ & x_j \in \{0, 1\}, \quad j = 0, \dots, m, \end{aligned}$$

- BLP es NP-hard
- ¡Tiene restricciones!
- Hay un método sistemático para pasar a QUBO (añadiendo variables auxiliares)

# 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:

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



# 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

Mathematical Methods  
in the Applied Sciences



RESEARCH ARTICLE

Open Access



## Solving the Kemeny ranking aggregation problem with quantum optimization algorithms

Elías F. Combarro, Raúl Pérez-Fernández, José Ranilla, Bernard De Baets

First published: 13 July 2023 | <https://doi.org/10.1002/mma.9489>

## Parte III

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

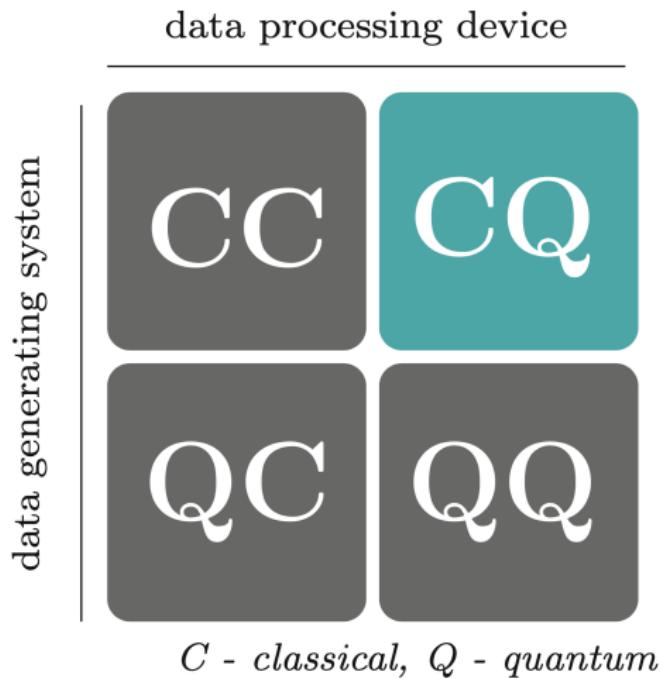
$$\begin{aligned} \text{Minimize} \quad & -5x_0 + 3x_1 - 2x_2 \\ \text{subject to} \quad & x_0 + x_2 \leq 1, \\ & 3x_0 - x_1 + 3x_2 \leq 4 \\ & x_j \in \{0, 1\}, \quad j = 0, 1, 2 \end{aligned}$$

- Se aplican automáticamente las transformaciones necesarias

## Parte IV

Quantum Machine Learning

# De qué hablamos cuando hablamos de Quantum Machine Learning



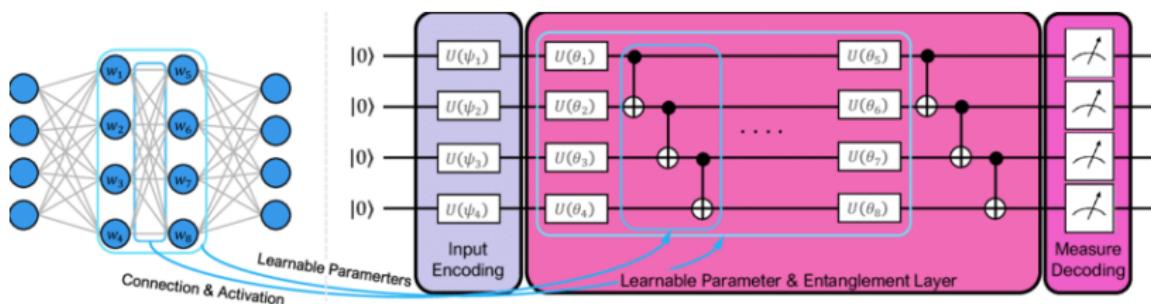
# QML translacional y speedups

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

\*There exist important caveats that can limit the applicability of the method<sup>51</sup>.

# Circuitos variacionales: la navaja suiza del QML

- Circuitos cuánticos con parámetros libres (ángulos de rotación)
- Varios usos:
  - Introducción de datos de entrada
  - Parámetros optimizables

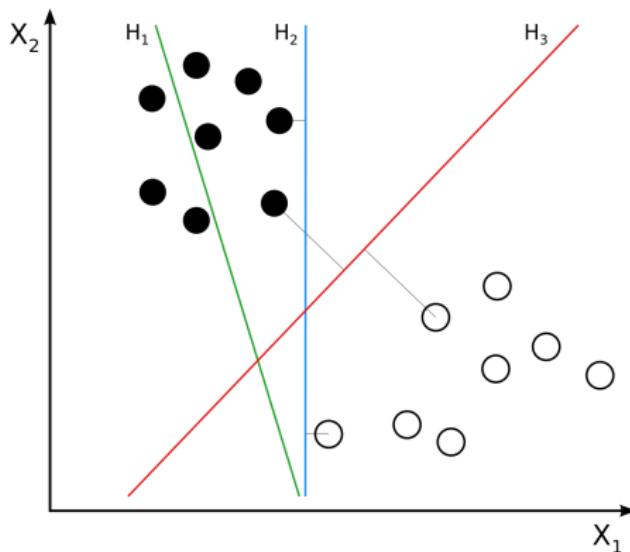


## Parte V

Quantum Support Vector Machines

# Support Vector Machines

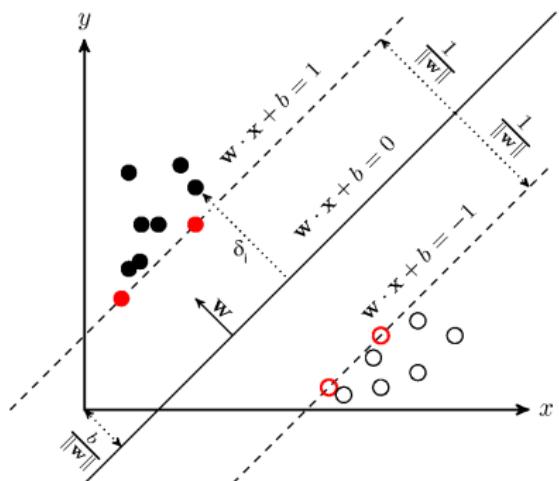
- Las Support Vector Machines (SVM) son un algoritmo (clásico) de clasificación muy popular
- La idea en la que se basan es encontrar un hiperplano separador con el mayor margen posible



# 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

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



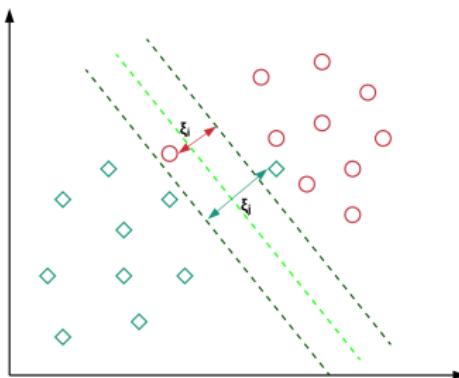
# El caso del “margen blando”

- 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  $\alpha_i$  podemos recuperar  $b$  y  $w$ . De hecho

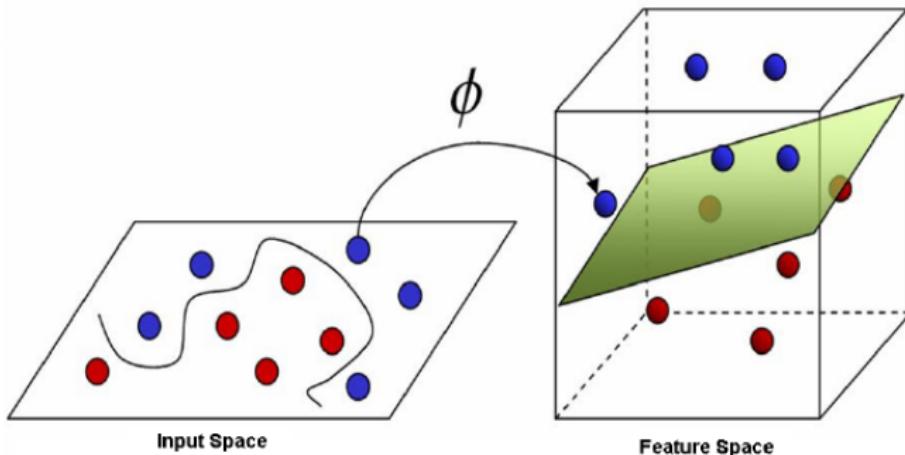
$$w = \sum_i \alpha_i y_i x_i$$

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

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

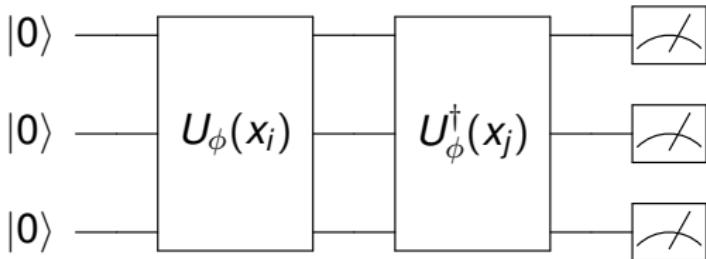
y para clasificar un punto  $x$  sólo necesitamos calcular

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

- 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

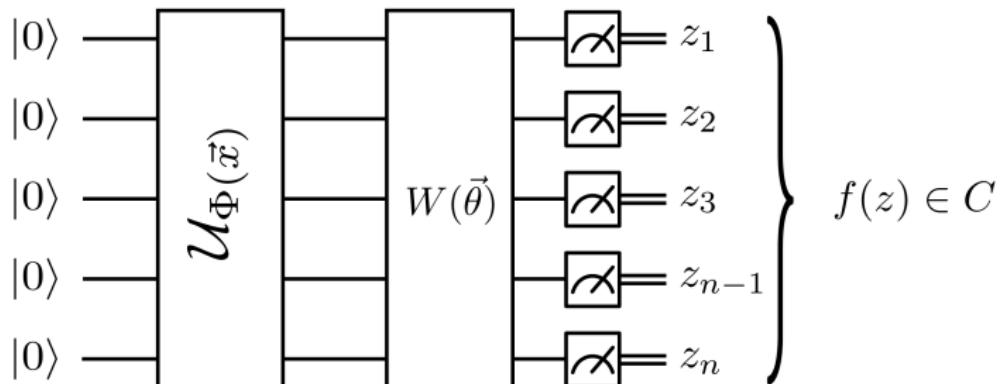


## Parte VI

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  $\theta$
- Medimos el estado y estimamos un valor esperado sobre un hamiltoniano, por ejemplo

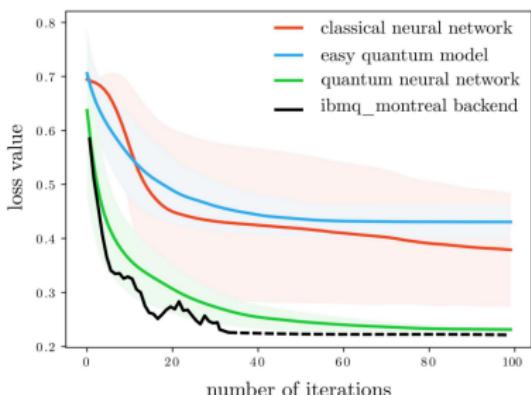
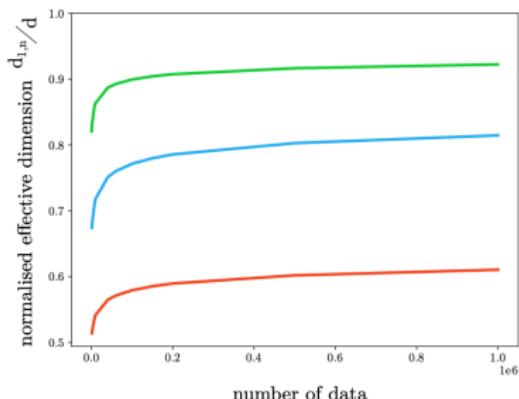
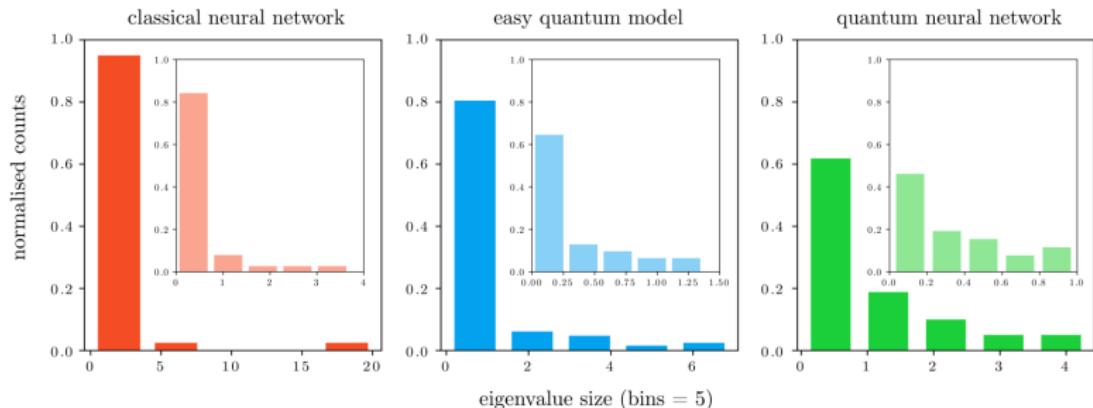
$$f(x, \theta) = \langle \psi(x, \theta) | Z_1 \cdots Z_n | \psi(x, \theta) \rangle$$

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

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

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

# 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<sup>1,2</sup>, Hsin-Yuan Huang <sup>3,4</sup>, M. Cerezo<sup>5,6</sup>, Kunal Sharma<sup>7</sup>,

Accepted: 4 August 2022

Andrew Sornborger<sup>5,8</sup>, Lukasz Cincio<sup>9</sup> & Patrick J. Coles <sup>9</sup>

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<sup>1,2\*</sup>, Michael Broughton<sup>3</sup>, Jordan Cotler<sup>4,5</sup>, Sitan Chen<sup>6,7</sup>, Jerry Li<sup>8</sup>, Masoud Mohseni<sup>3</sup>, Hartmut Neven<sup>3</sup>, Ryan Babbush<sup>3</sup>, Richard Kueng<sup>9</sup>, John Preskill<sup>8,2,30</sup>, Jarrod R. McClean<sup>1,6</sup>

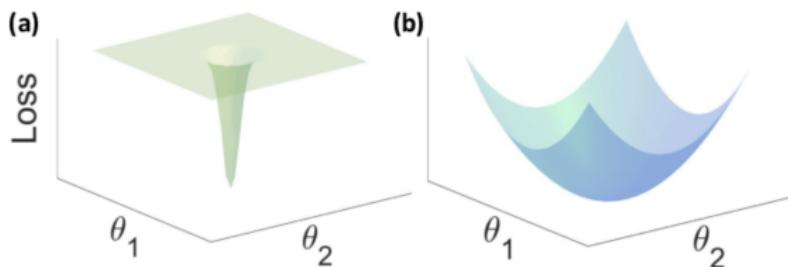
Quantum technology promises to revolutionize how we learn about the physical world. An experiment that processes quantum data with a quantum computer could have substantial advantages over conventional experiments in which quantum states are measured and outcomes are processed with a classical computer. 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

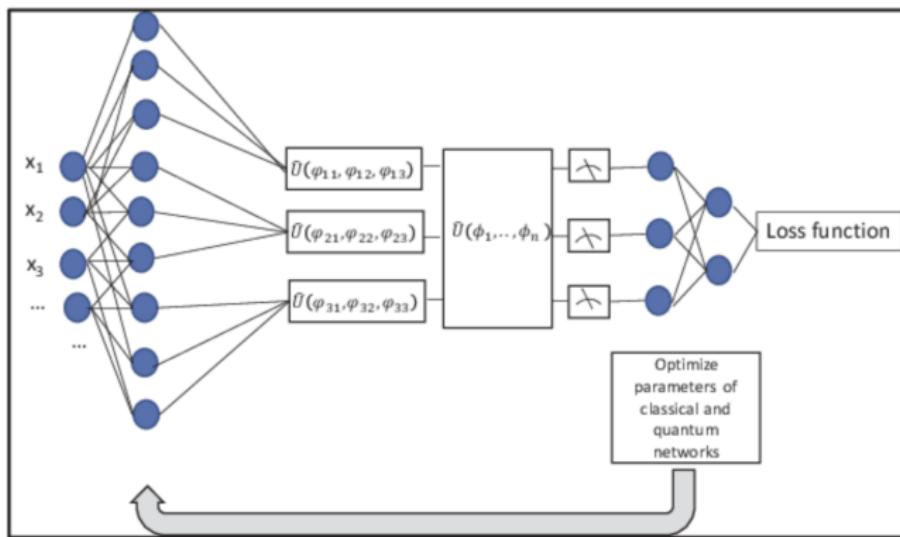
# Limitaciones del QML

- Tamaño de los dispositivos
  - Selección de parámetros e instancias
  - Pocos parámetros entrenables
- Ruido y errores
- Acceso a los dispositivos
- Problemas inherentes (barren plateaus, selección de formas variacionales, arquitecturas...)



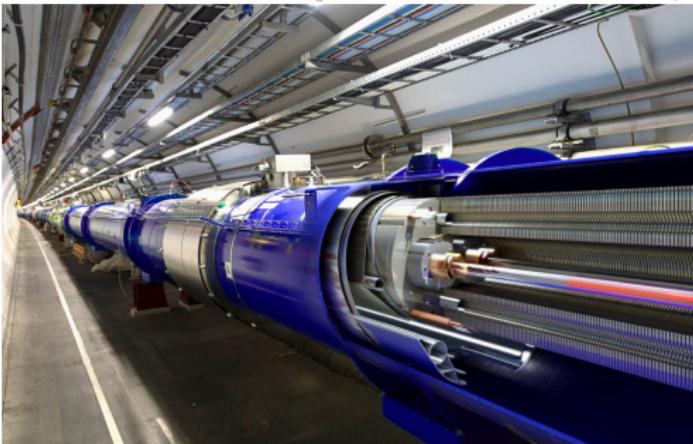
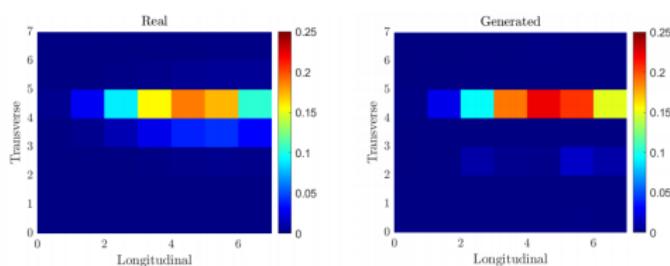
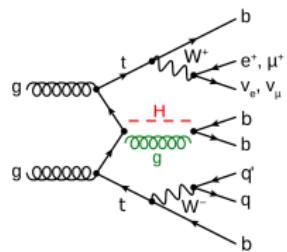
# Redes híbridas

- Pueden paliar algunas de las limitaciones
  - Selección de parámetros “entrenable”
  - Salida configurable
- ¿Cuánto trabajo hace la parte clásica y cuánto la cuántica?



# Algunas aplicaciones del Quantum Machine Learning

- Aplicaciones en ML supervisado, no supervisado y por refuerzo



Gracias por su atención

