



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



Agencia
Nacional de
Investigación
y Desarrollo

Ministerio de Ciencia,
Tecnología, Conocimiento
e Innovación

Gobierno de Chile

Aprendizaje Automático Cuántico

Ariel Norambuena

ariel.norambuena@usm.cl

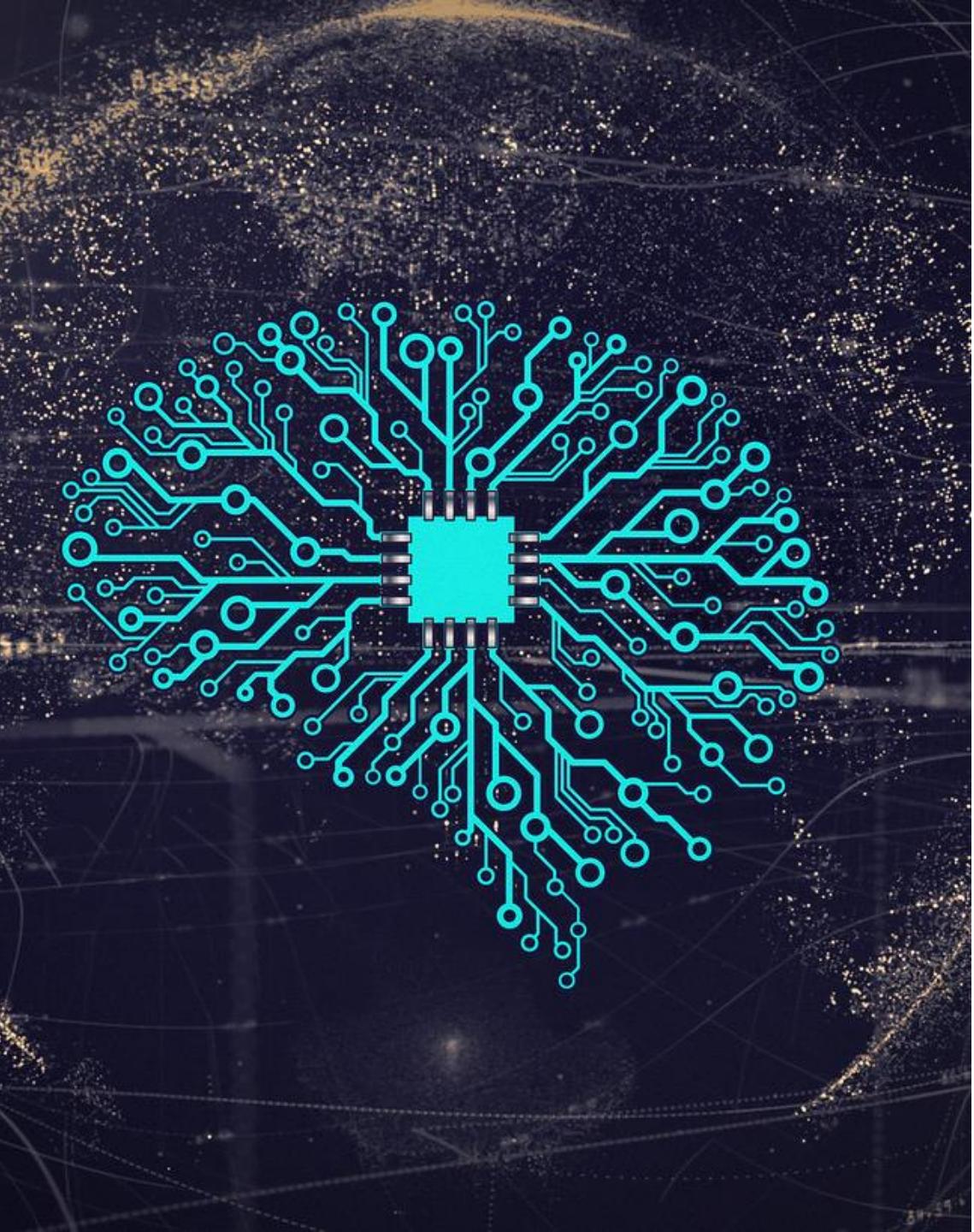
Profesor Asistente

Departamento de Física

Universidad Técnica Federico Santa María

III Escuela de Computación Cuántica, 6-10 enero, 2025





Índice



Inteligencia Artificial



Aprendizaje automático cuántico



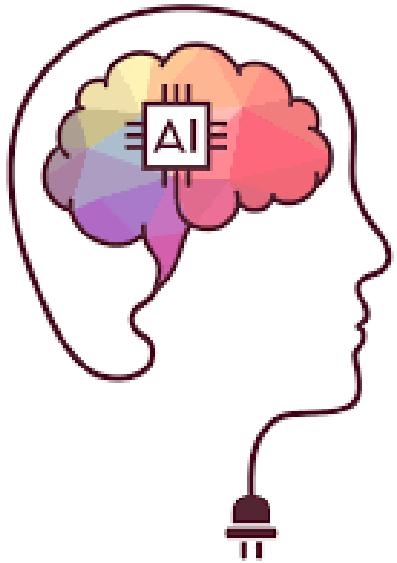
Circuitos cuánticos variacionales



Redes neuronales en mecánica cuántica



Máquinas de vectores de soporte

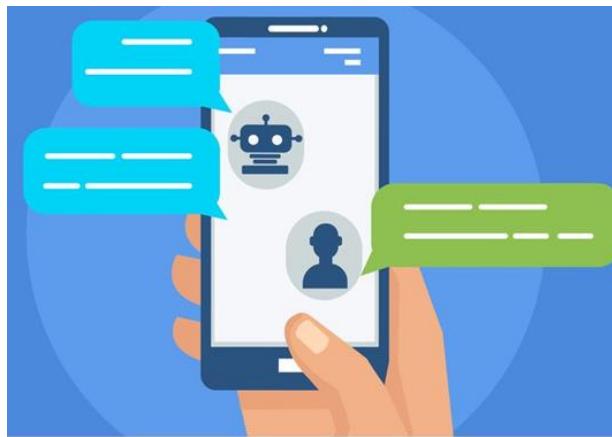


Inteligencia Artificial

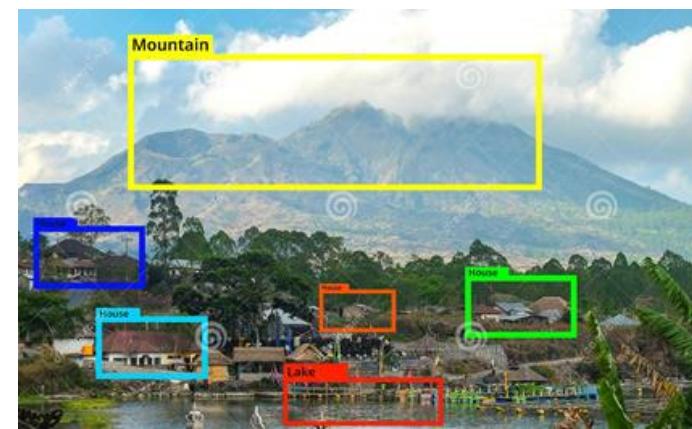
Inteligencia artificial

- La **inteligencia artificial (IA)** se focaliza en construir sistemas capaces de realizar tareas similares a la inteligencia humana, como el procesamiento del lenguaje natural, el reconocimiento de imágenes, y la toma de decisiones.

Chatbots



Procesamiento de imágenes

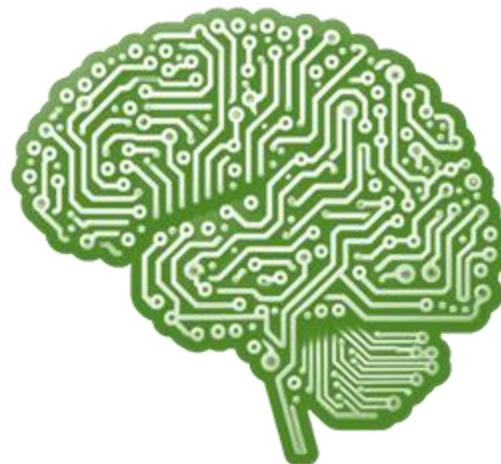


Vehículos autónomos



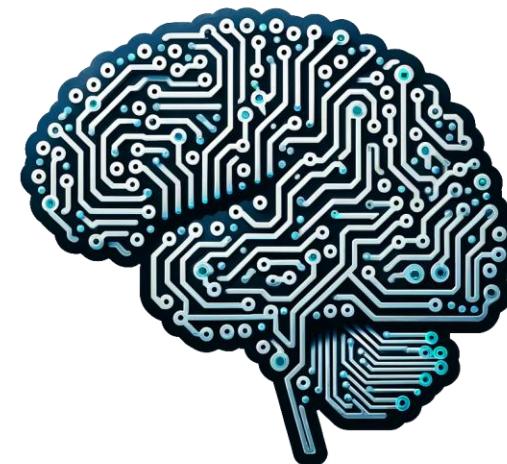
Inteligencia artificial débil y fuerte

- Actualmente, los **sistemas de IA** carecen de una comprensión y conciencia similares a las capacidades humanas: **no tienen una verdadera inteligencia**



IA débil

Sistemas que sólo pueden realizar un conjunto limitado de tareas

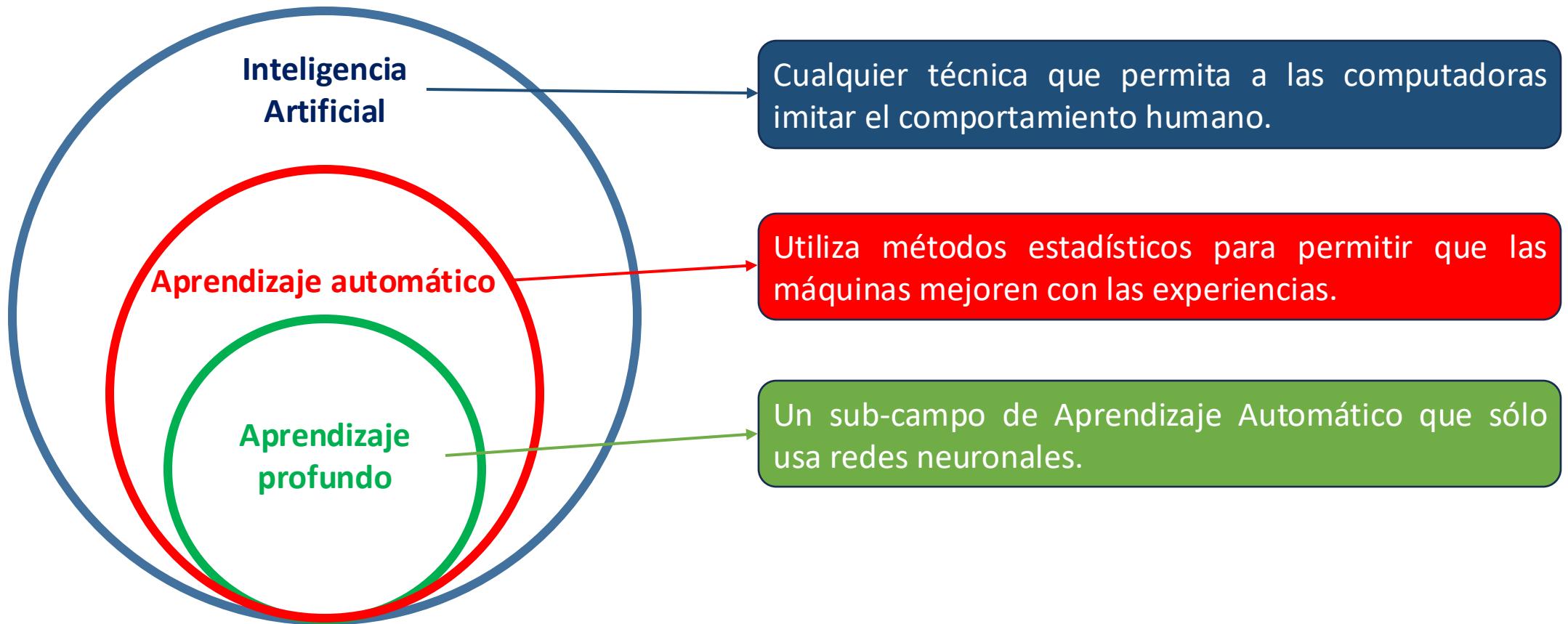


IA fuerte

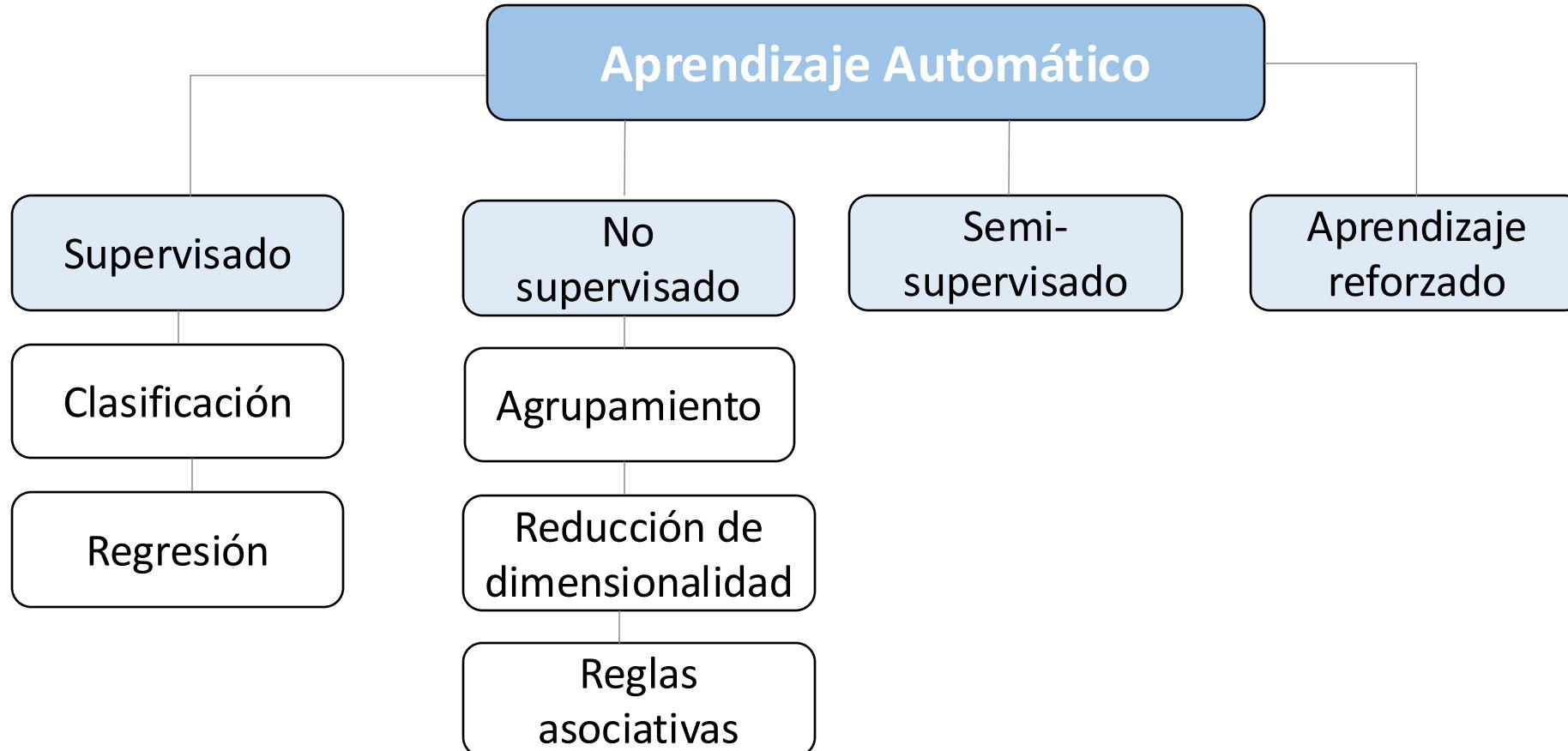
Sistemas que pueden realizar multitud de tareas con un nivel alto de complejidad

Inteligencia Artificial y subcampos

- La IA contiene varios subcampos como **Aprendizaje Automático** y **Aprendizaje Profundo**



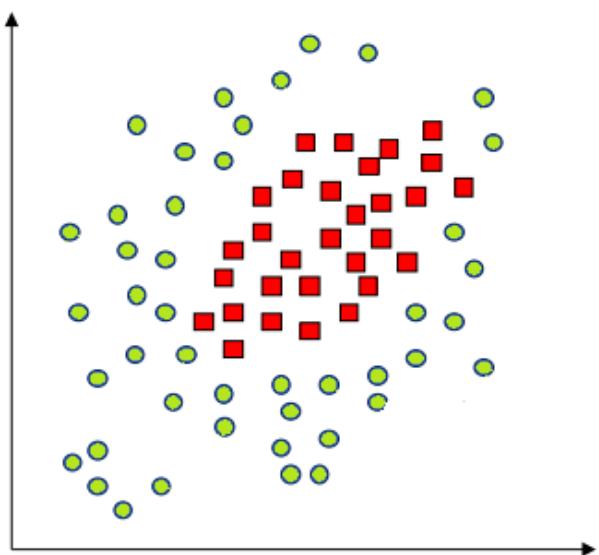
Ramas del Aprendizaje Automático



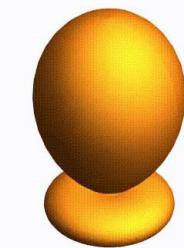
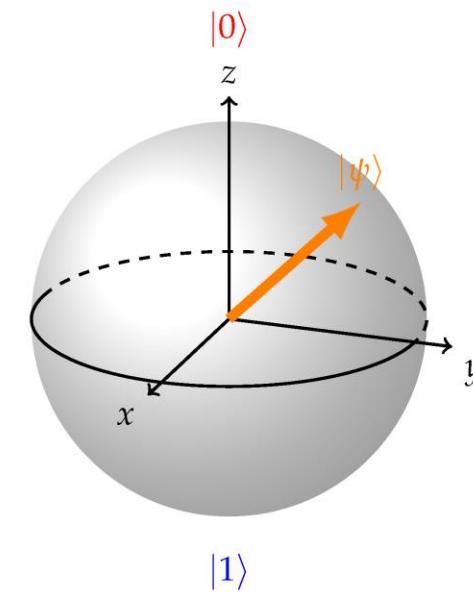
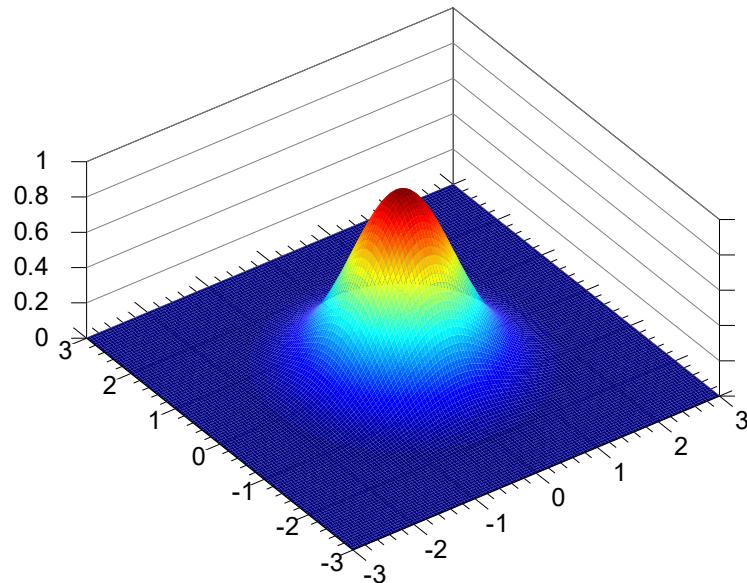
Aprendizaje automático

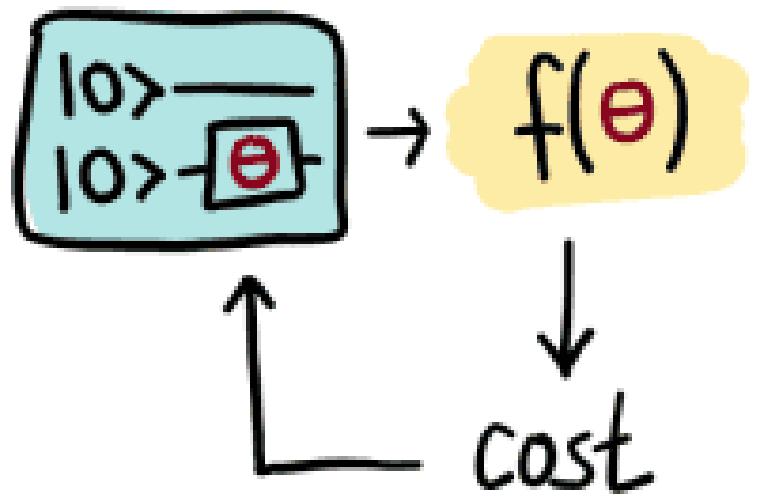
Mecánica cuántica

Geometría



Probabilidades





Aprendizaje Automático Cuántico

Aprendizaje Automático Cuántico

		Tipo de Algoritmo
		Classical
		Quantum
Classical		CC
Quantum		CQ
QC		QC
QQ		QQ

CC: dato clásico procesado en un computador clásico

QC: dato cuántico procesado en un computador clásico

CQ: dato clásico procesado en un computador cuántico

QQ: dato cuántico procesado en un computador cuántico

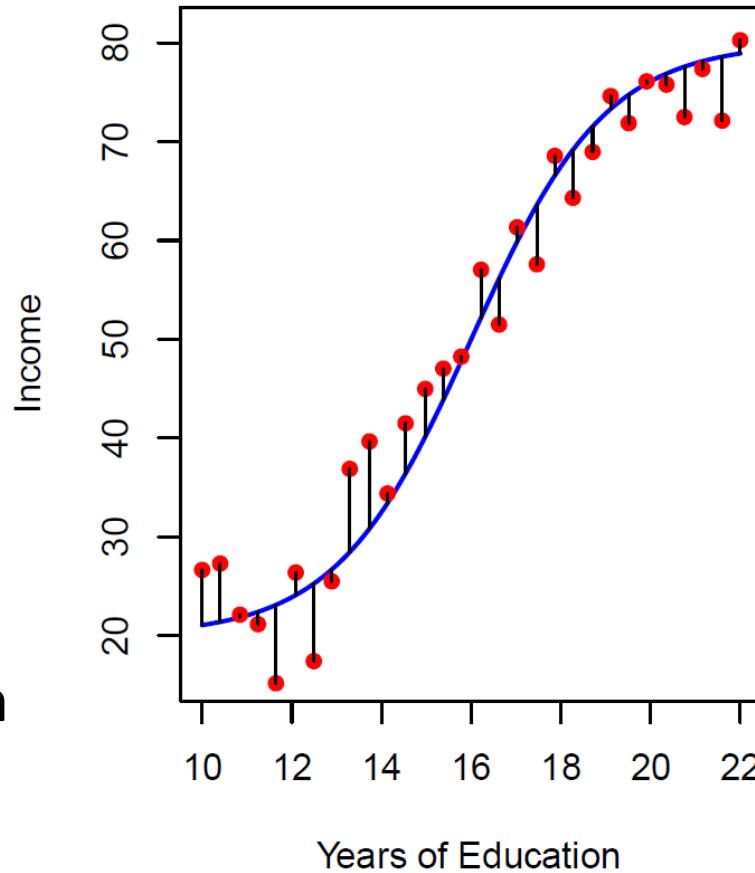
Codificación de la información

Información clásica

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Métodos de codificación

- Base computacional
- Amplitud
- Hamiltoniana
- Fase



Relación fundamental

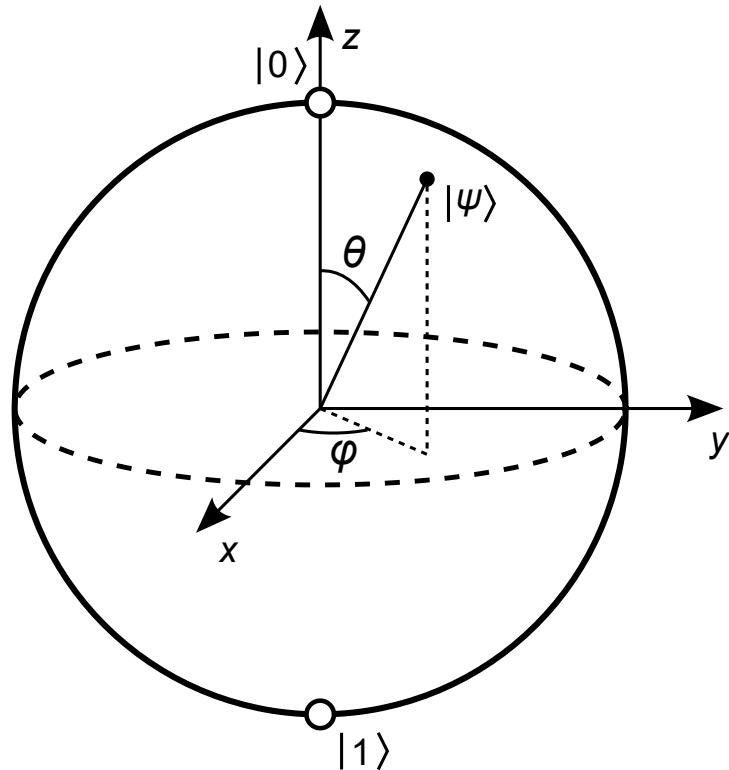
$$y = f(\mathbf{x}) + \epsilon$$

Predicción

$$\hat{y} = \hat{f}(\mathbf{x})$$

ML: se buscan modelos tales que $\hat{y} \approx y$

Elementos de la Computación Cuántica



- Implementación nativa de operadores unitarios y hermíticos
$$O = e^{iU}, \quad U = U^\dagger$$
- Medidas (colapso de la función de onda)
- Suposición y entrelazamiento

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

$$\theta \in [0, \pi], \quad \varphi \in [0, 2\pi]$$

Codificación de base computacional

“hello”

	Bit clásico	Bit cuántico	
“h”:104	1101000	$ 1101000\rangle$	▪ Ventaja: fácil de preparar
“e”:101	1100101	$ 1100101\rangle$	$ 0\rangle \xrightarrow{X} 1\rangle$
“l”:108	1101100	$ 1101100\rangle$	$ 0\rangle \xrightarrow{X} 1\rangle$
“l”:108	1101100	$ 1101100\rangle$	▪ Desventaja: requiere
“o”:111	1101111	$ 1101111\rangle$	muchos bits cuánticos

¡7 bits cuánticos!

Codificación de amplitud

Dato clásico

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad \rightarrow \quad |x\rangle = \frac{1}{\mathcal{N}} (x_0|0\rangle + x_1|1\rangle)$$

$$\mathcal{N} = \sqrt{x_0^2 + x_1^2}$$

Dato cuántico

- **Ventaja:** requiere menos bit cuánticos

Número de datos para N bits cuánticos

$$\dim(\mathcal{H}^{\otimes N}) = 2^N$$

- **Desventaja:** difícil de preparar y leer

Codificación Hamiltoniana

Modelo de Ising

$$H = - \sum_{ij} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i$$

- **Ventaja:** fácil (natural) de implementar

Problemas tipo QUBO

$$f_Q(x) = \sum_{i,j=1}^n Q_{ij} x_i x_j, \quad \mathbb{B} = \{0, 1\}$$

- **Desventaja:** limitado para ciertos problemas

$$f_Q(x) : \mathbb{B}^n \mapsto \mathbb{R}, \quad x \in \mathbb{B}^n$$

Codificación de fase

Dato clásico

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \xrightarrow{\hspace{1cm}} \quad U(\mathbf{x}) = R_y(x_4)R_x(x_3)R_z(x_2)R_y(x_1)$$

Operador cuántico

$$R_\sigma(\theta) = \exp\left(-i\frac{\theta}{2}\sigma\right)$$

- **Ventaja:** fácil (natural) de implementar
- **Desventaja:** requiere muchas compuertas (profundidad)

Conjunto de testeo y entrenamiento

Datos clásicos

(X_{test}, Y_{test})

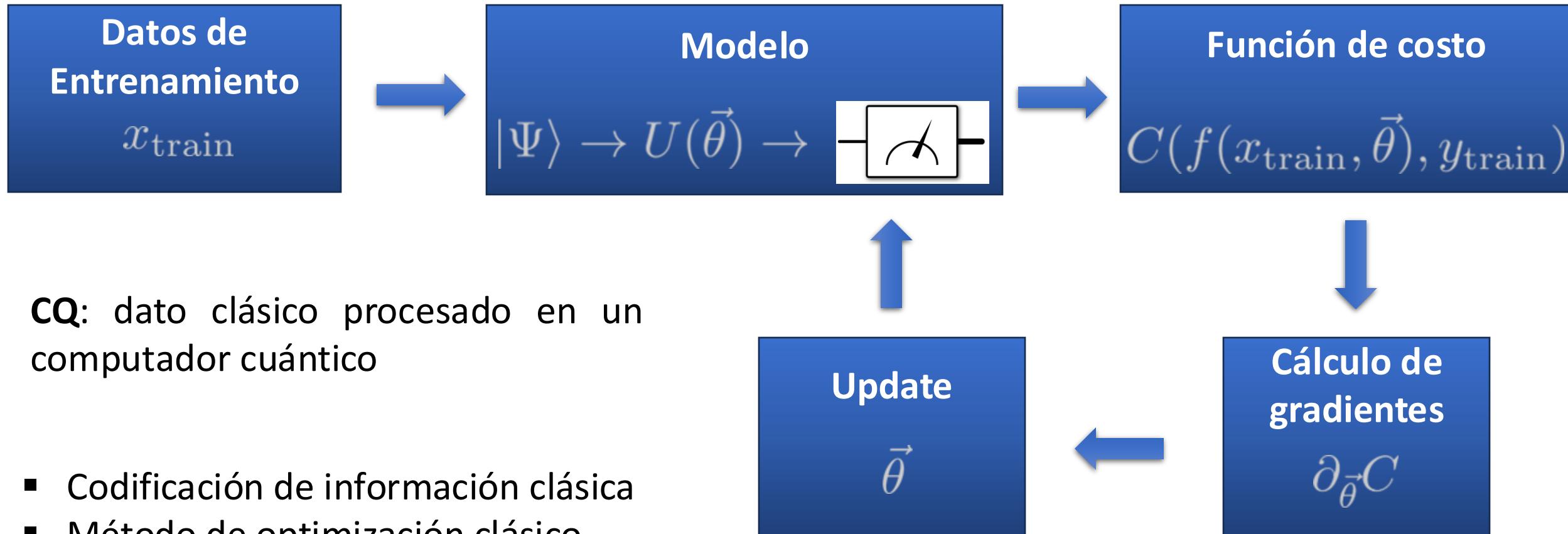
Datos de testeo

(X_{train}, Y_{train})

Datos de
entrenamiento

- El proceso de aprendizaje consiste en usar una fracción de los datos originales para ajustar los parámetros del modelo.

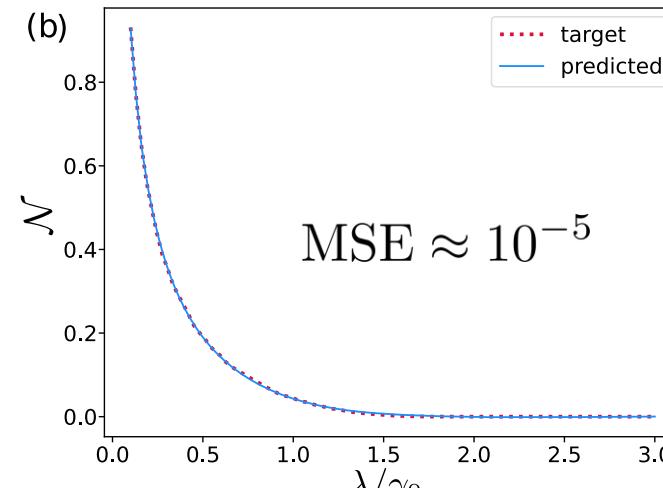
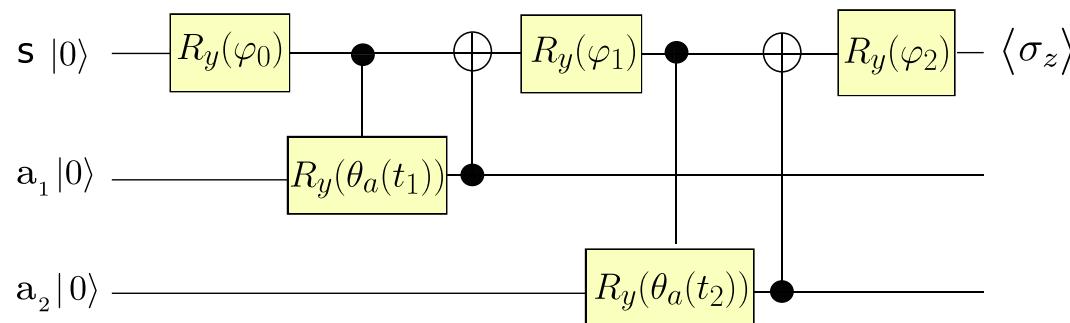
Circuitos cuánticos variacionales (VQCs)



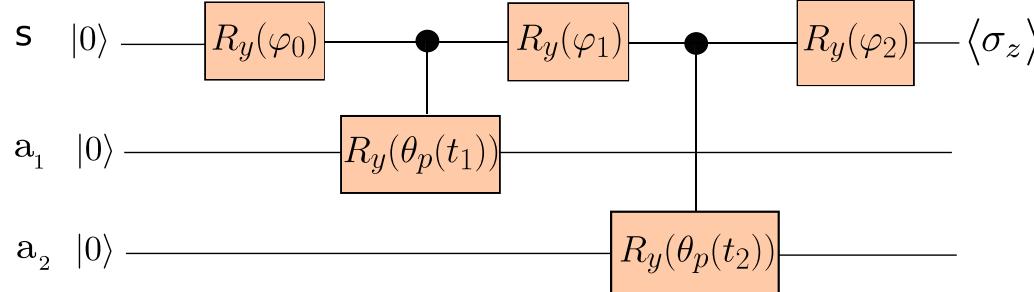
- Codificación de información clásica
- Método de optimización clásico
- Modelo cuántico para codificar y procesar información

VQCs y grado de no-Markovianidad

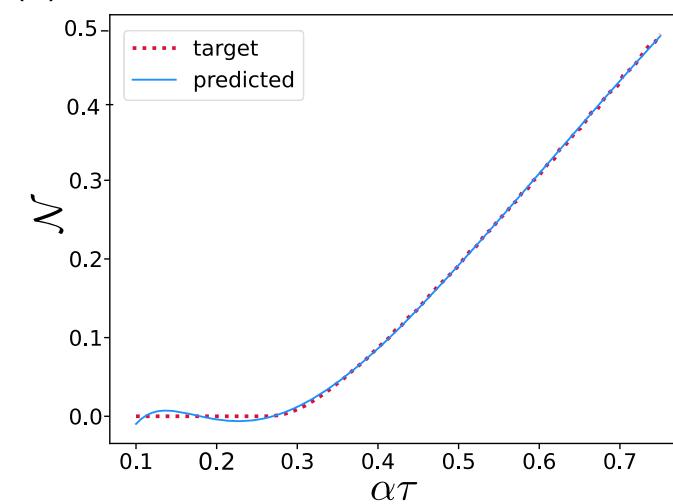
(a)



(c)



(d)



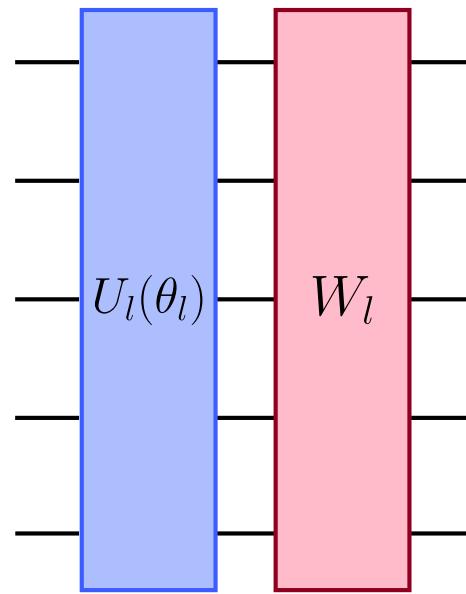
Hossein Dinani

Problema de Barren Plateaus

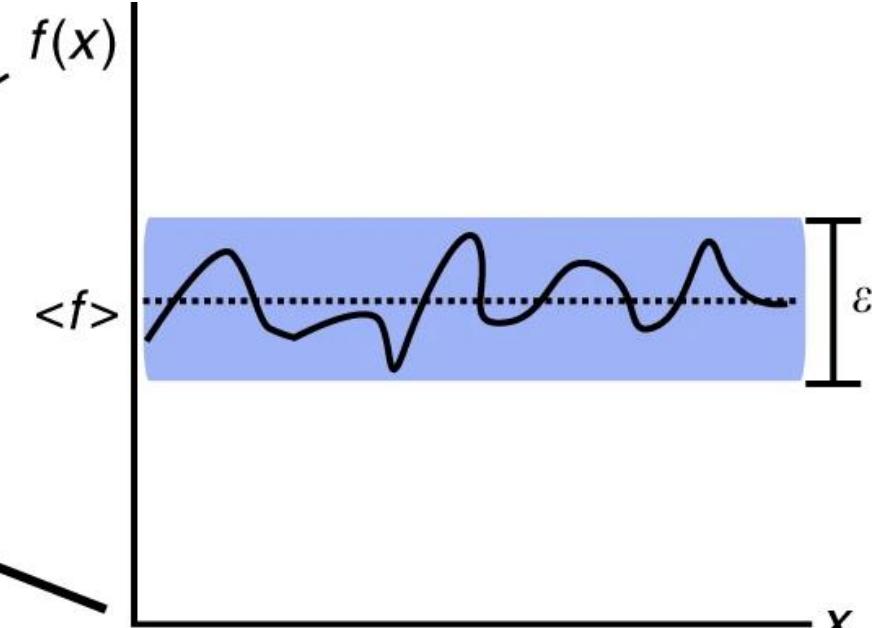
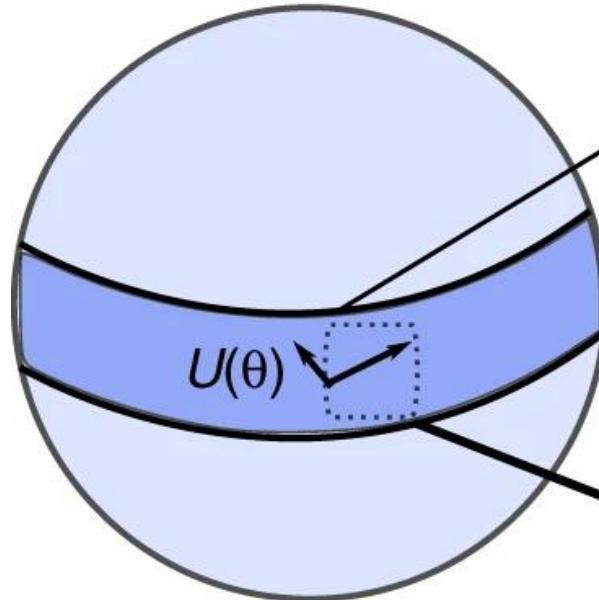


Número de qbits y problema de Barren Plateau

$$U(\vec{\theta}) = \prod_{l=1}^L U_l(\theta_l) W_l$$

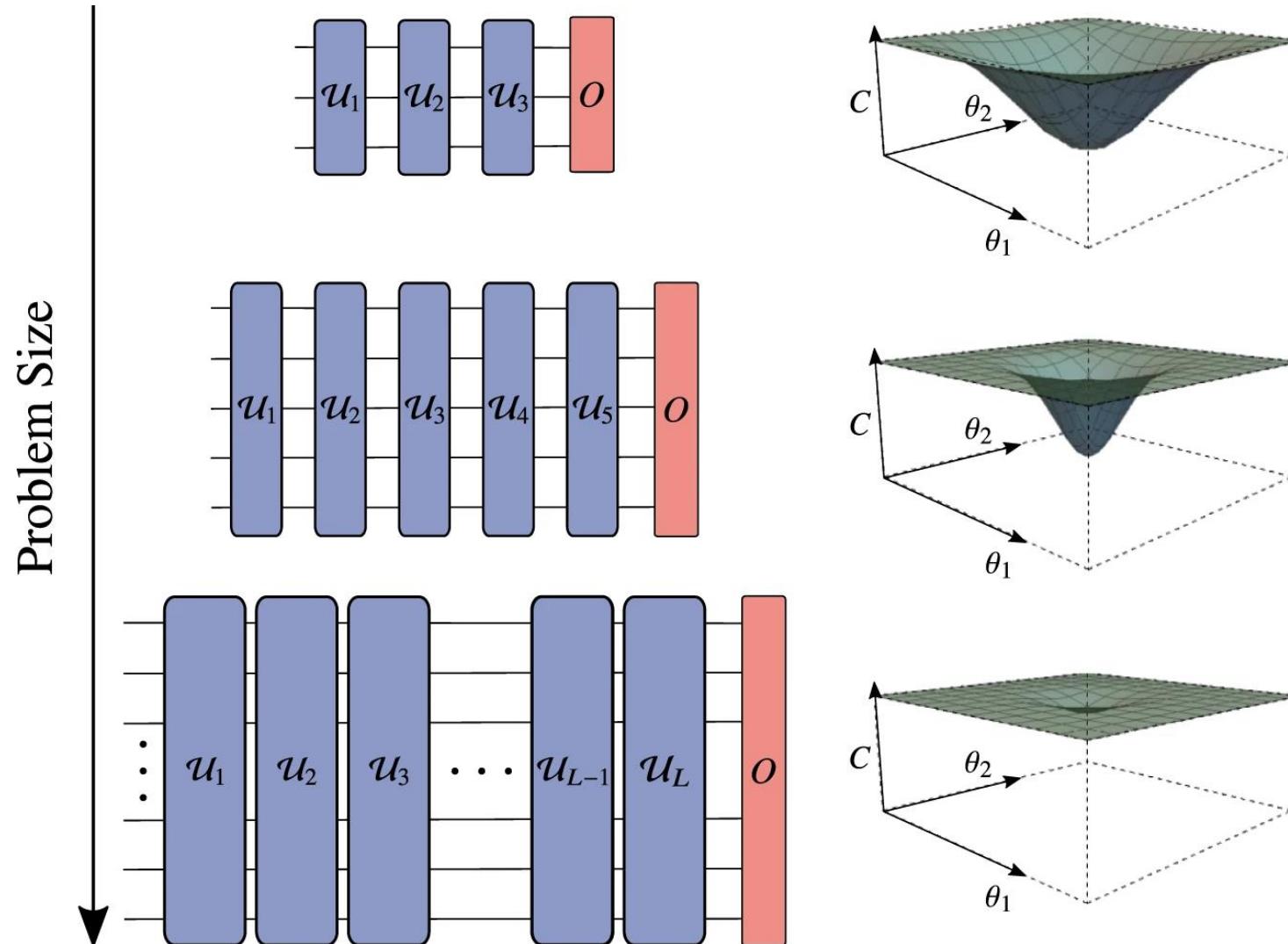


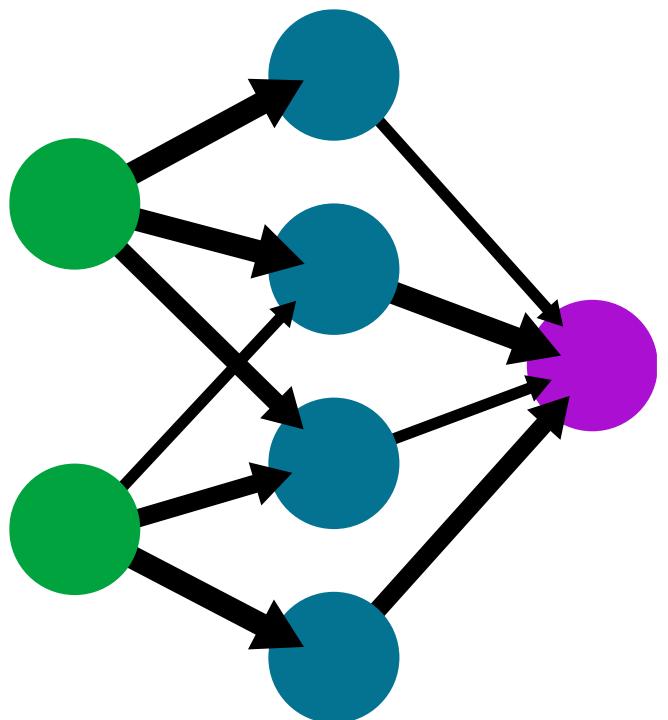
$$U_l(\theta_l) = e^{-i\theta_l V_l}$$



$$E(\vec{\theta}) = \langle 0 | U^\dagger(\vec{\theta}) H U(\vec{\theta}) | 0 \rangle, \quad \text{Var}[\partial_{\theta_k} E] \approx \frac{1}{2^{(3N-1)}} \text{Tr}[H^2] \text{Tr}[\rho^2] \text{Tr}[V^2]$$

Barren plateau inducido por ruido

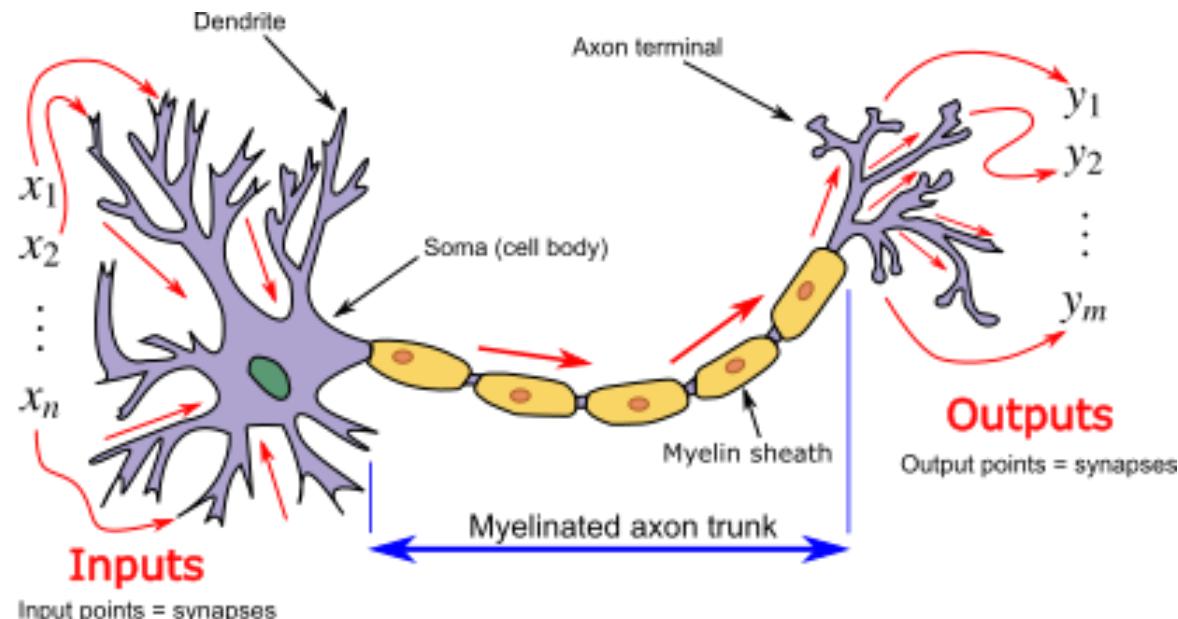




Redes neuronales clásicas y cuánticas

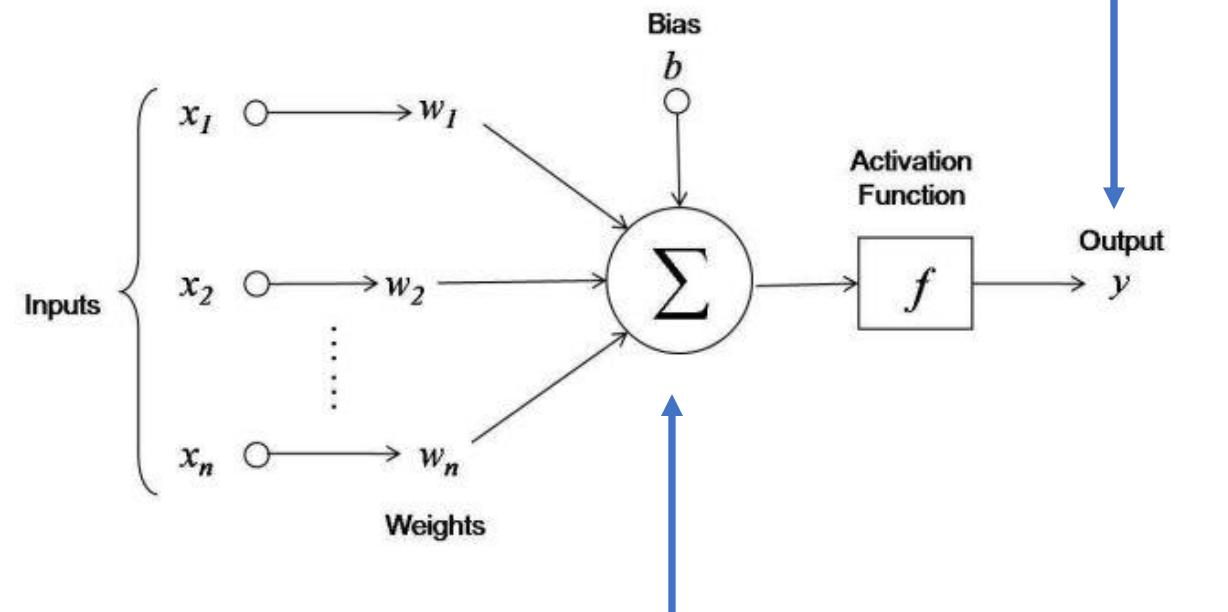
Concepto de neurona bio-inspirada

Neurona biológica



Attribution: this work is derived from [Neuron.svg](#) by [Quasar Jarosz](#) at [English Wikipedia](#).

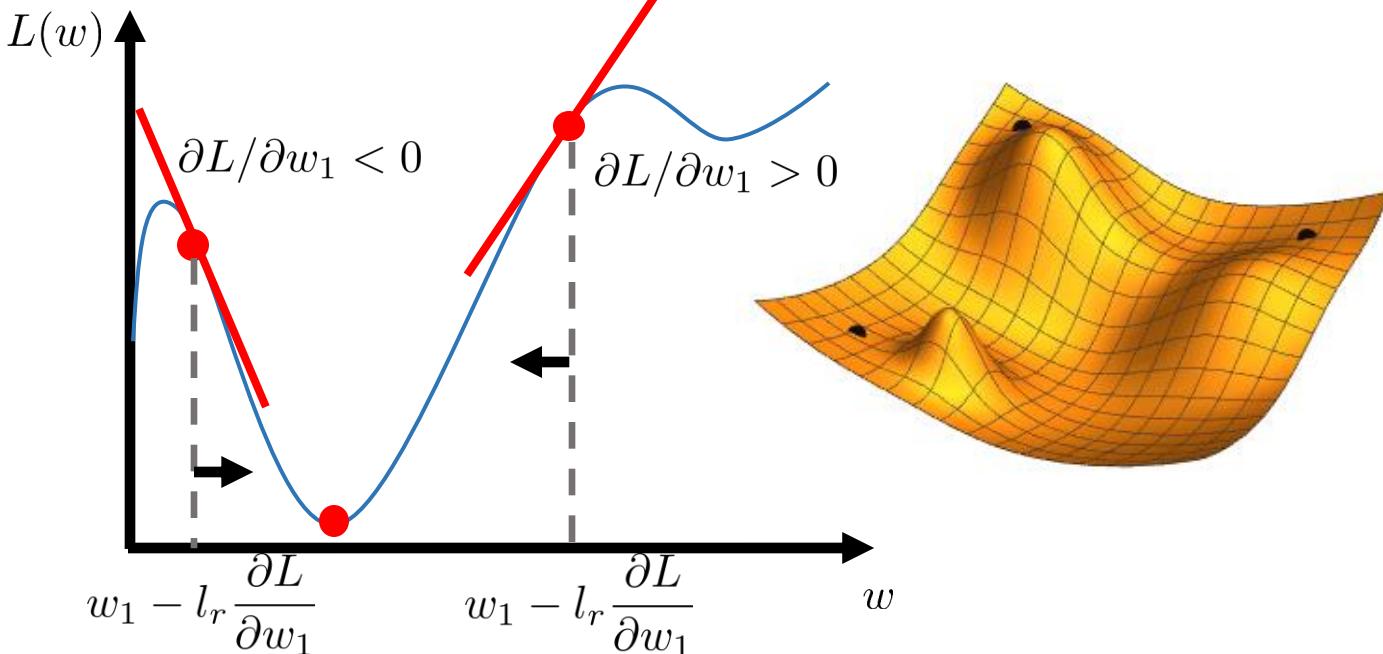
Neurona artificial



$$y = f(\Sigma)$$

¿Qué es una red neuronal?

Red neuronal



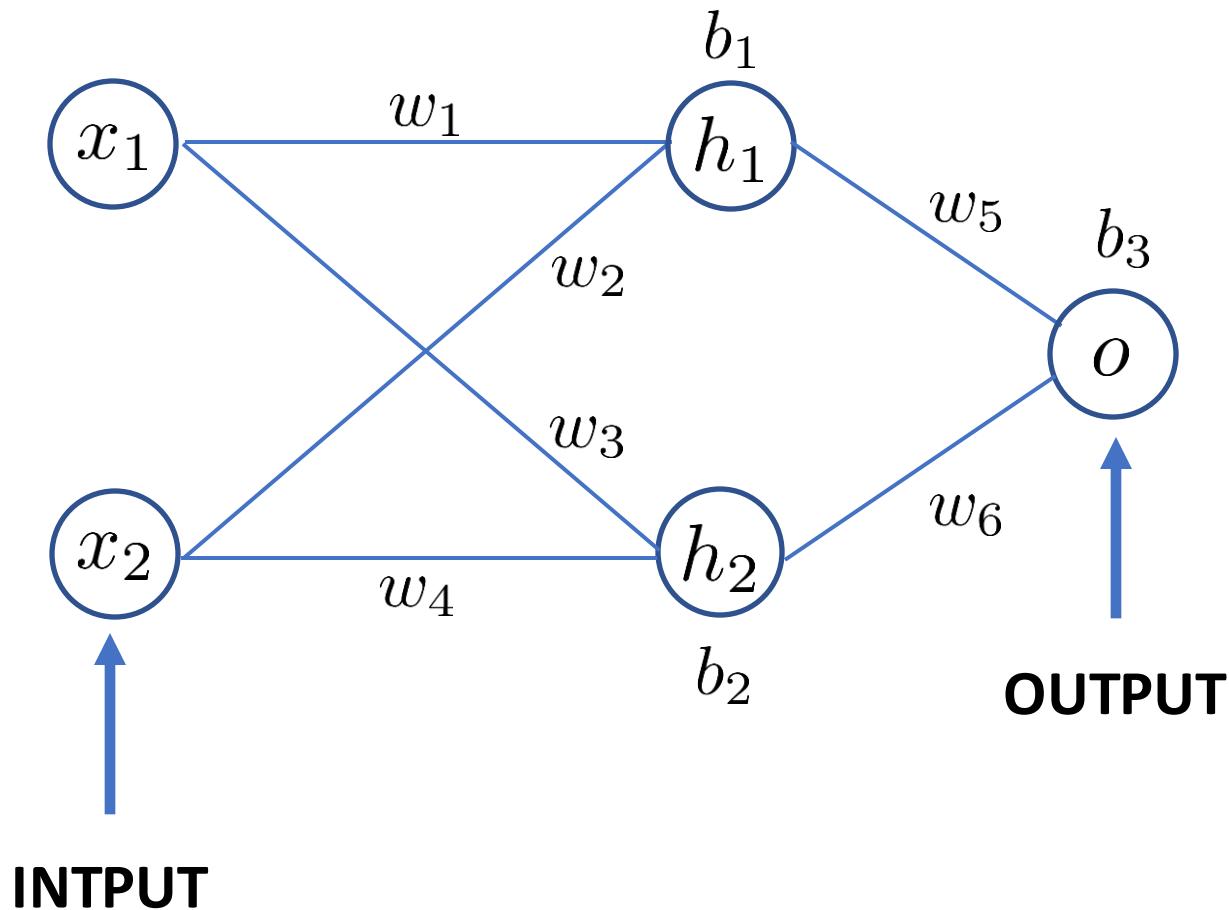
$$\omega_i \rightarrow \omega_i - l_r \frac{\partial L}{\partial \omega_i} \quad l_r : \text{learning rate}$$

Propiedades

- Teorema de aproximación universal
Hornik K, Neural Networks 4 251–257, 1991.
Siempre existe una red neuronal que puede aproximar a una función arbitraria $f(x)$
- Toda red neuronal tiene asociada una función de costo (loss function)
$$L = L(\omega_i, b_i)$$
- Usando métodos de optimización se buscan valores (ω_i, b_i) que **minimicen** $L(\omega_i, b_i)$

Una red neuronal puede ser entrenada (aprendizaje) usando datos, modelos o una combinación de ambos. Una vez entrenada, puede resolver problemas nuevos.

Red neuronal simple



Capas ocultas

$$h_1 = f(w_1x_1 + w_2x_2 + b_1)$$

$$h_2 = f(w_3x_1 + w_4x_2 + b_2)$$

Predicción

$$o = f(w_5h_1 + w_6h_2 + b_3)$$

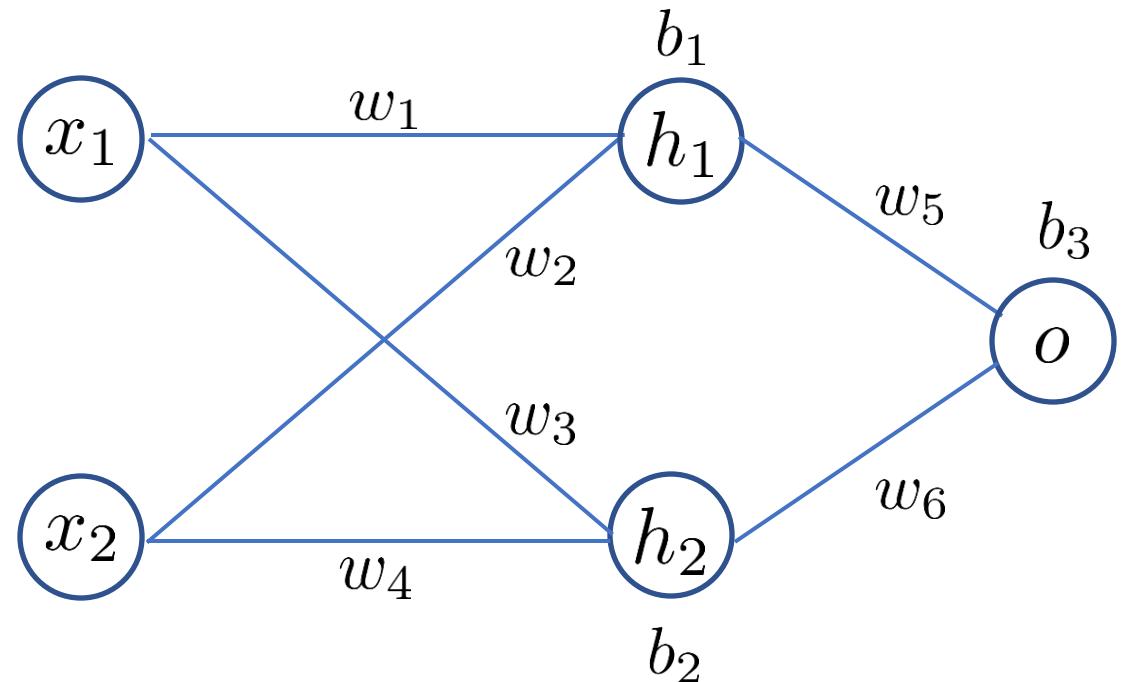
Derivadas en redes neuronales

Función de costo

$$L = \frac{1}{N} \sum_{i=1}^N \left(y_i^{true} - y_i^{pred} \right)^2$$

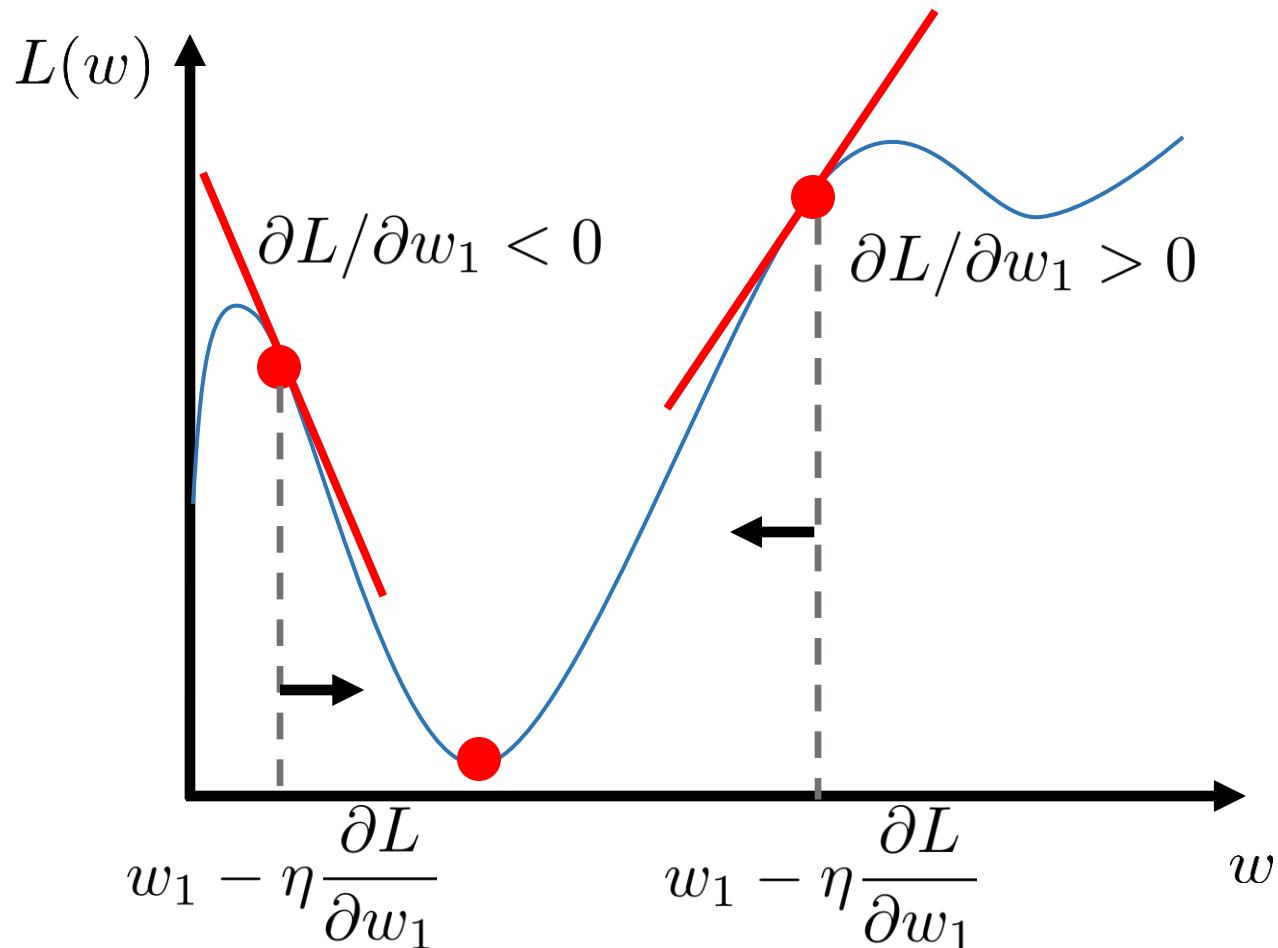
Derivadas exactas

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y^{pred}} \frac{\partial y^{pred}}{\partial h_1} \frac{\partial h_1}{\partial w_1}$$



$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial y^{pred}} \frac{\partial y^{pred}}{\partial h_2} \frac{\partial h_2}{\partial w_4}$$

Métodos de gradientes



$$w_i^{n+1} = w_i^n - \eta \frac{\partial L}{\partial w_i^n}$$

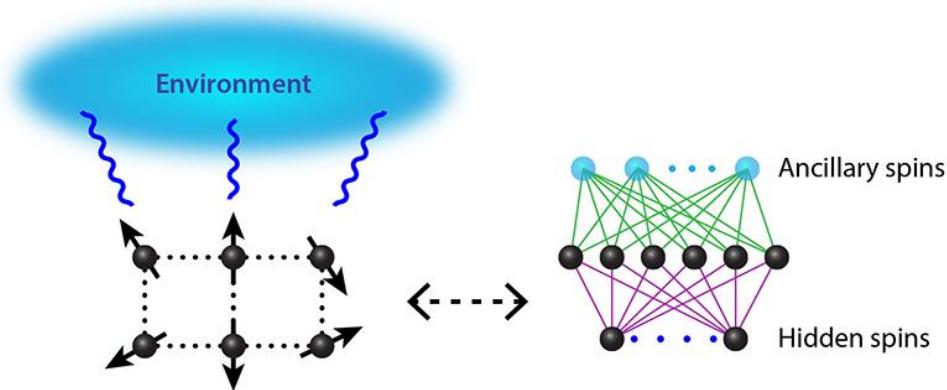
$$b_i^{n+1} = b_i^n - \eta \frac{\partial L}{\partial b_i^n}$$

n : épocas (iteraciones)

η : tasa de aprendizaje

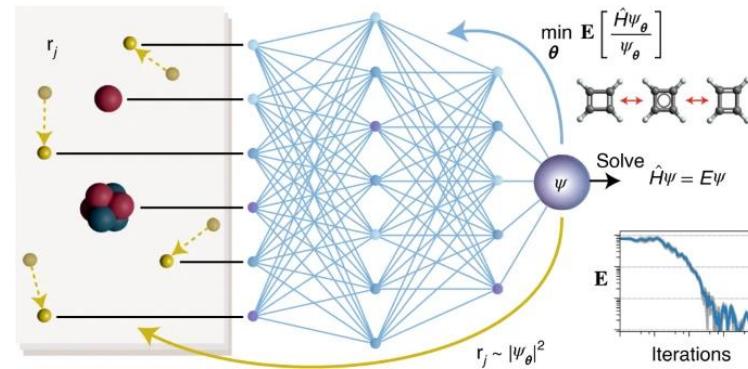
Uso de aprendizaje profundo en cuántica

Steady-state in many-body systems



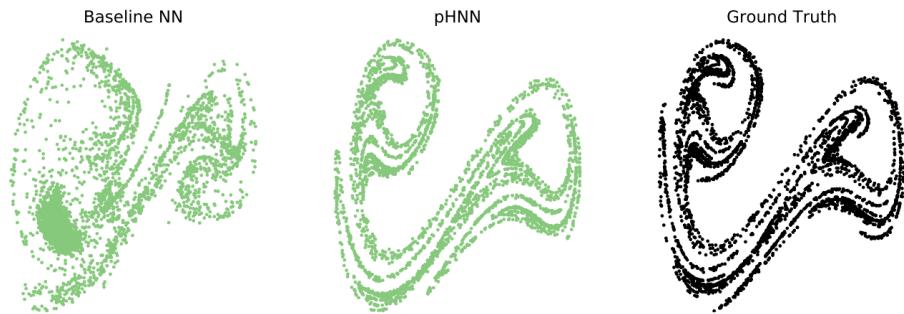
Phys. Rev. Lett. **122**, 250501 (2019)

Estimation of ground state in complex molecules



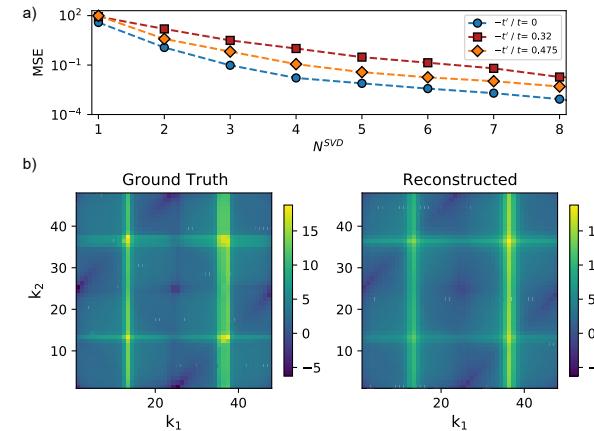
Nat. Chem. **12**, 891–897 (2020)

Learning chaotic dynamical systems



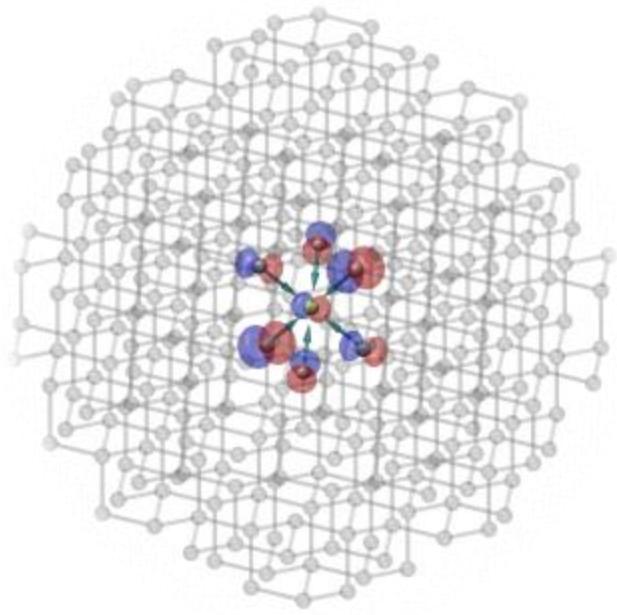
Phys. Rev. E **104**, 034312 (2021)

Deep learning data-driven reduction

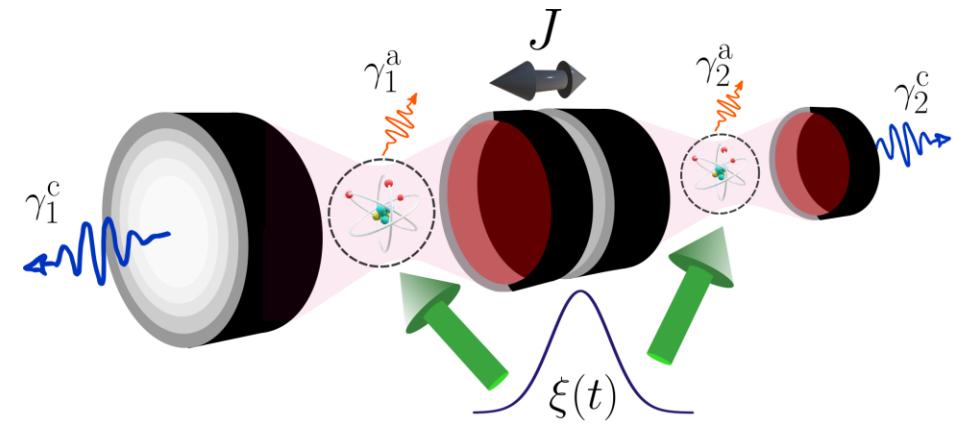


Phys. Rev. Lett. **129**, 136402 (2022)

Centros de color en diamante



Cavidades ópticas



control

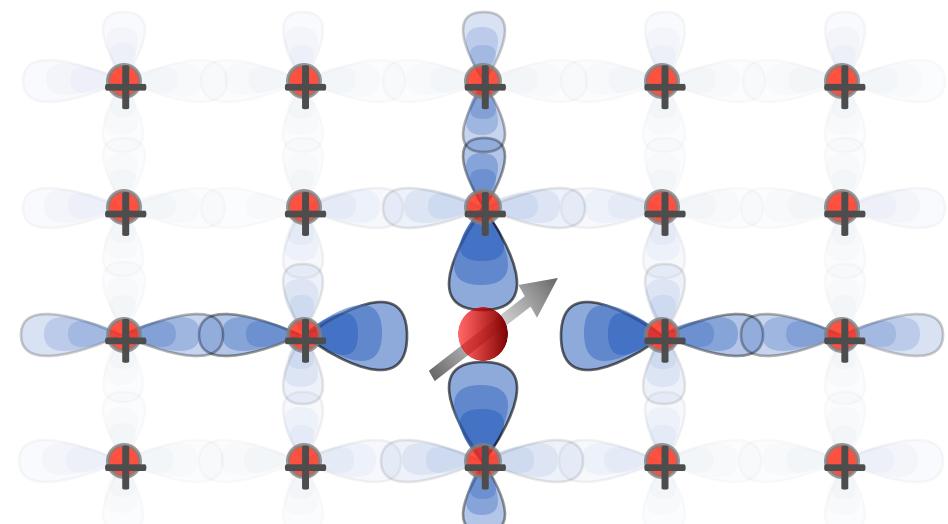
$$H(t) = H_S + H_B + V + \boxed{H_c(t)}$$

Espín en materia condensada

Redes de espín



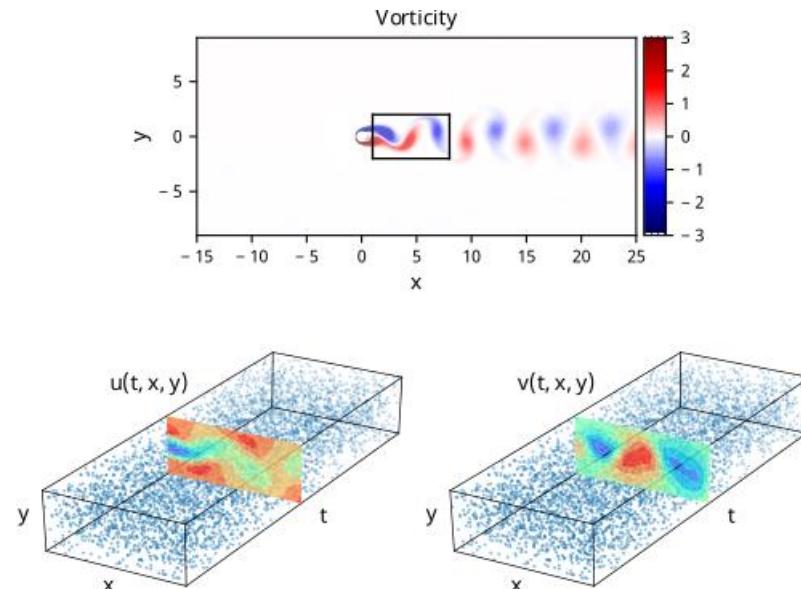
$$\mathbf{S}^{(j-2)} \mathbf{S}^{(j-1)} \mathbf{S}^{(j)} \mathbf{S}^{(j+1)} \mathbf{S}^{(j+2)}$$



Redes neuroanales informadas por la física

Data driven model

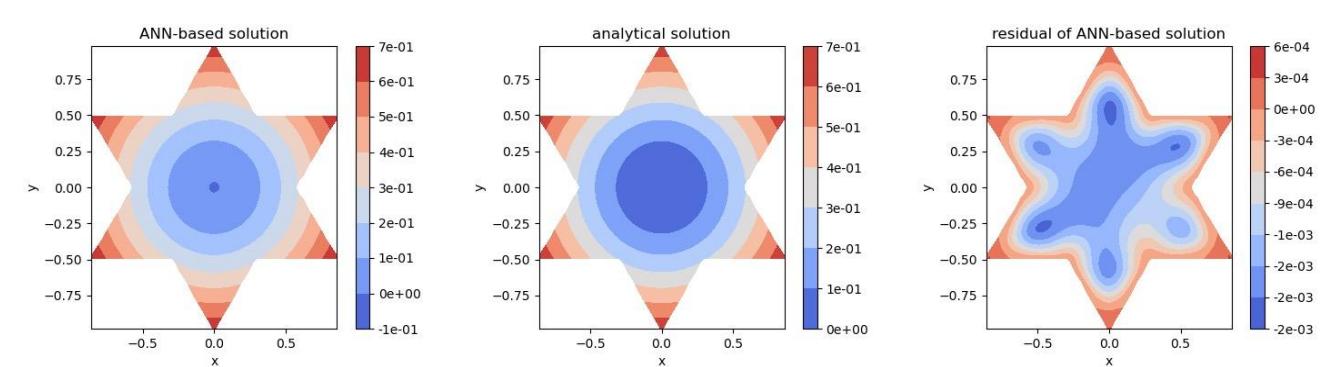
Aprendizaje, identificación de sistemas,
Descubrimiento de ecuaciones diferenciales



$$f := u_t + \lambda_1(uu_x + vu_y) + p_x - \lambda_2(u_{xx} + u_{yy})$$
$$g := v_t + \lambda_1(uv_x + vv_y) + p_y - \lambda_2(v_{xx} + v_{yy})$$

Equation driven model

Solución de ecuaciones diferenciales ordinarias/parciales
y modelos no lineales



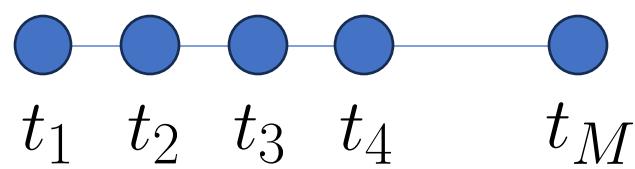
$$\nabla^2 u(x, y) + e^{u(x, y)} = 1 + x^2 + y^2 + \frac{4}{(1 + x^2 + y^2)^2}$$

NeuroDiffEq: Open-source Python library;
F. Chen, M. Mattheakis et al JOSS **15** (2020)

Redes neuronales para control cuántico

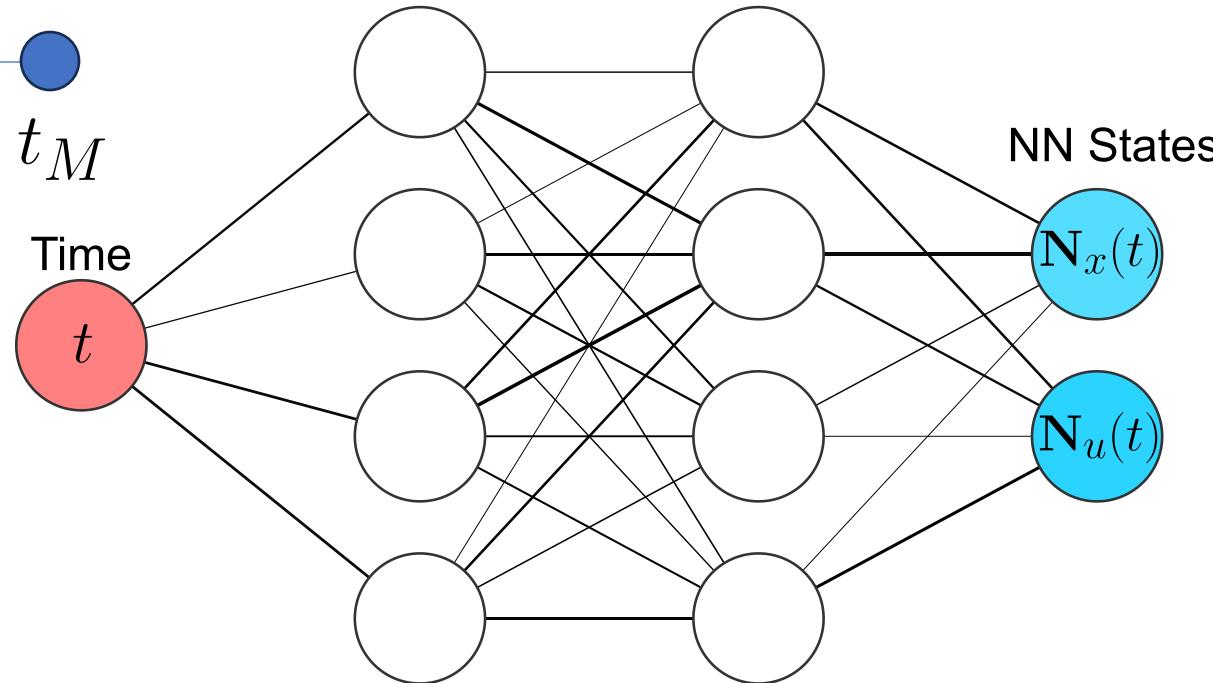
Input: time array

$$t = [t_1, t_2, \dots, t_M]$$



QC: dato cuántico
procesado en un
computador clásico

Hidden layers
with activation function $\sin(x)$



Loss function
 $L = L_{\text{model}} + L_{\text{control}} + L_{\text{const}} + L_{\text{reg}}$

System dynamics

$$\dot{\mathbf{x}}(t) = A(\lambda, \mathbf{u}(t))\mathbf{x}(t)$$

$$(t, \lambda, \mathbf{x}(0), \mathbf{u}(0))$$

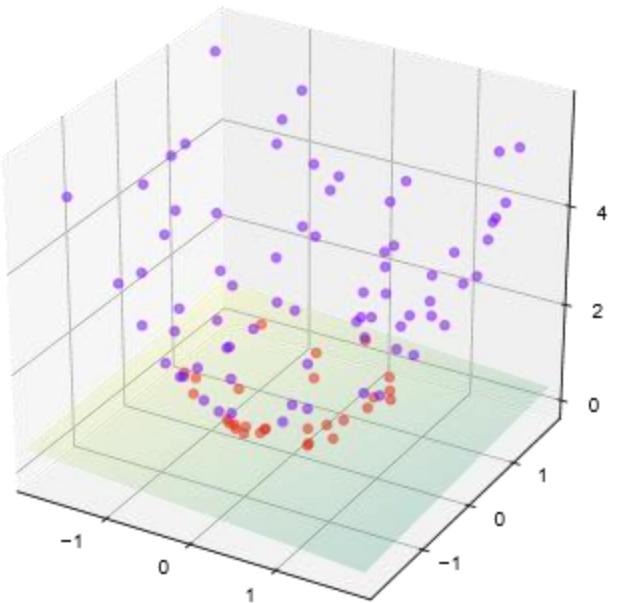
Neural Network

$$\mathbf{N}_x(t), \mathbf{N}_u(t)$$

Parametrization

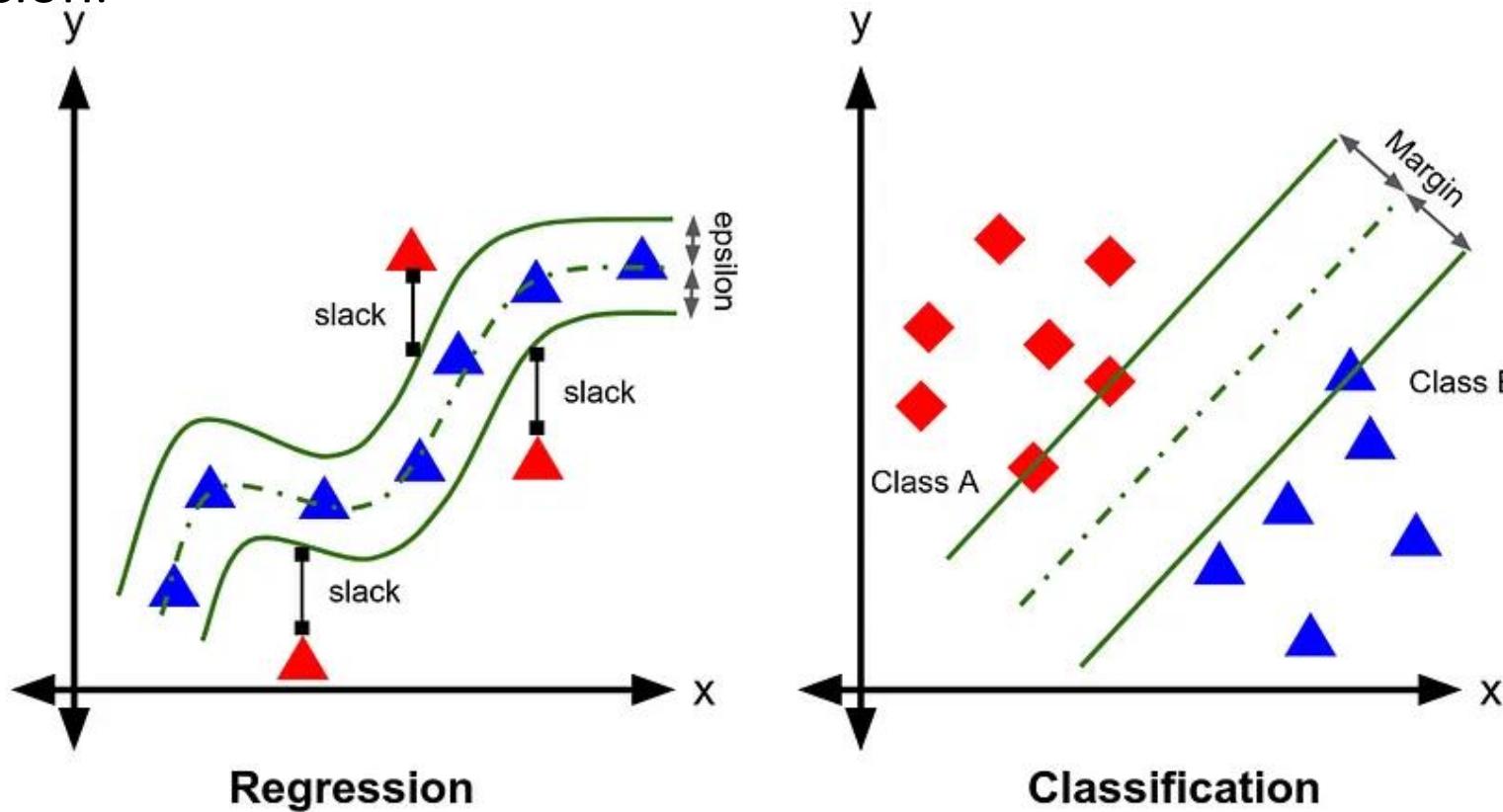
$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(0) + f(t)\mathbf{N}_x(t) \\ \mathbf{u}(t) &= \mathbf{u}(0) + f(t)\mathbf{N}_u(t)\end{aligned}$$

Máquinas de Vectores de Soporte



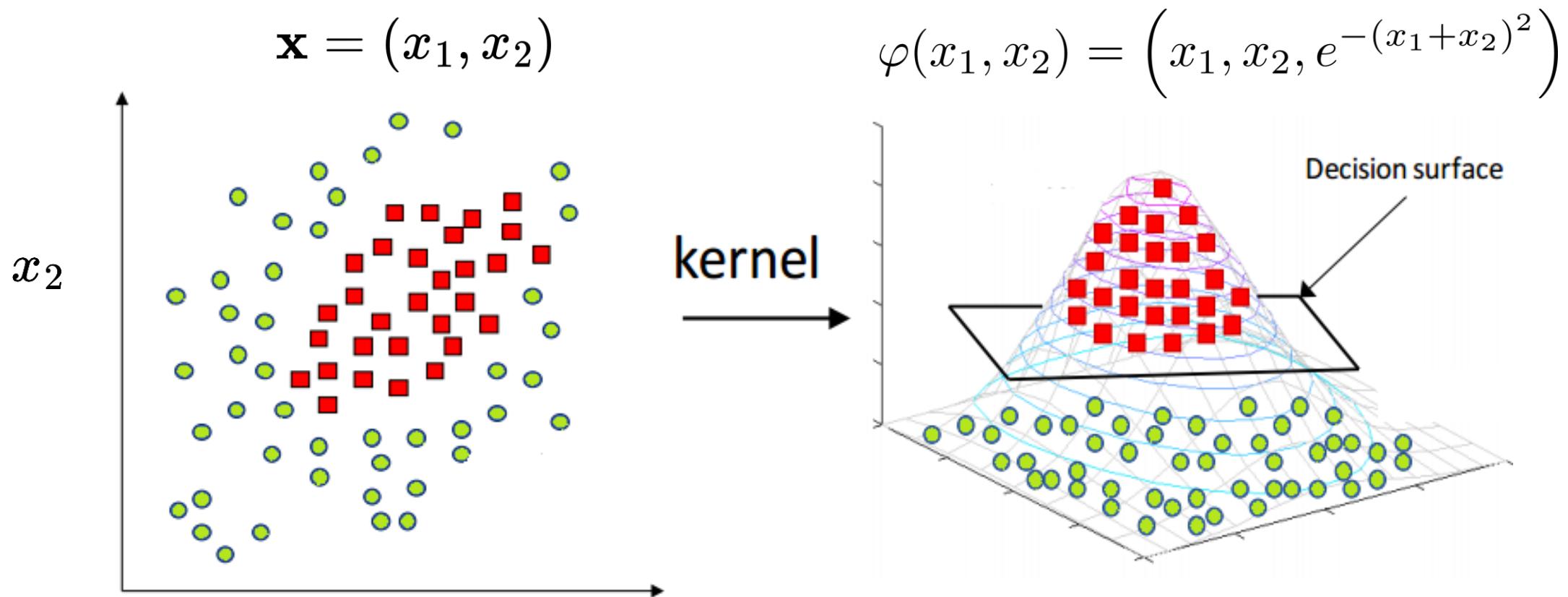
Máquinas de vectores de soporte

- Las máquinas de vectores de soporte (Support Vector Machine, SVM) son un conjunto de algoritmos de aprendizaje supervisado.
- La familia de algoritmos de SVM pueden ser utilizados tanto para problemas de regresión como clasificación.



Truco del kernel

- El truco del kernel consiste en mapear los datos a una dimensión superior mediante una función φ que llamaremos mapa de características.



Función kernel

- Sea V el espacio donde pertenece la función φ , el producto interno se calcula como:

Mapa de características $\mathbf{x} \longmapsto \varphi(\mathbf{x})$ $\varphi(\mathbf{x}) = (x_1, x_2, x_1^2 + x_2^2)$

Producto interno

$$\begin{aligned}\langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle &= x_1 y_1 + x_2 y_2 + (x_1^2 + x_2^2) (y_1^2 + y_2^2) \\ &= \mathbf{x} \cdot \mathbf{y} + \|\mathbf{x}\|^2 \|\mathbf{y}\|^2 \\ &= k(\mathbf{x}, \mathbf{y}) \quad \text{Función kernel}\end{aligned}$$

Propiedades de la función kernel

1. Las funciones kernel son simétricas: $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$
2. Semi definida positiva: $k(\mathbf{x}, \mathbf{y}) \geq 0$
3. Suma de kernels es una función kernel: $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) + k_2(\mathbf{x}, \mathbf{y})$
4. Producto de dos kernels es un kernel: $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y})k_2(\mathbf{x}, \mathbf{y})$
5. Escalamiento por ambos lados es un kernel: $k(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})k(\mathbf{x}, \mathbf{y})f(\mathbf{y})$
6. Es común normalizar las funciones kernel de modo que:
$$0 \leq k(\mathbf{x}, \mathbf{y}) \leq 1$$

$$k(\mathbf{x}, \mathbf{x}) = 1$$

Interpretación y tipos de kernels

- Las funciones Kernel más utilizadas son: lineal, polinómica homogéneo/no-homogéneo, sigmoide y función de base radial

**Kernel polinomial
homogéneo**

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$$

**Kernel función de
base radial**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

**Kernel polinomial
no-homogéneo**

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + r)^d$$

Kernel sigmoide

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$$

Matriz de Gram

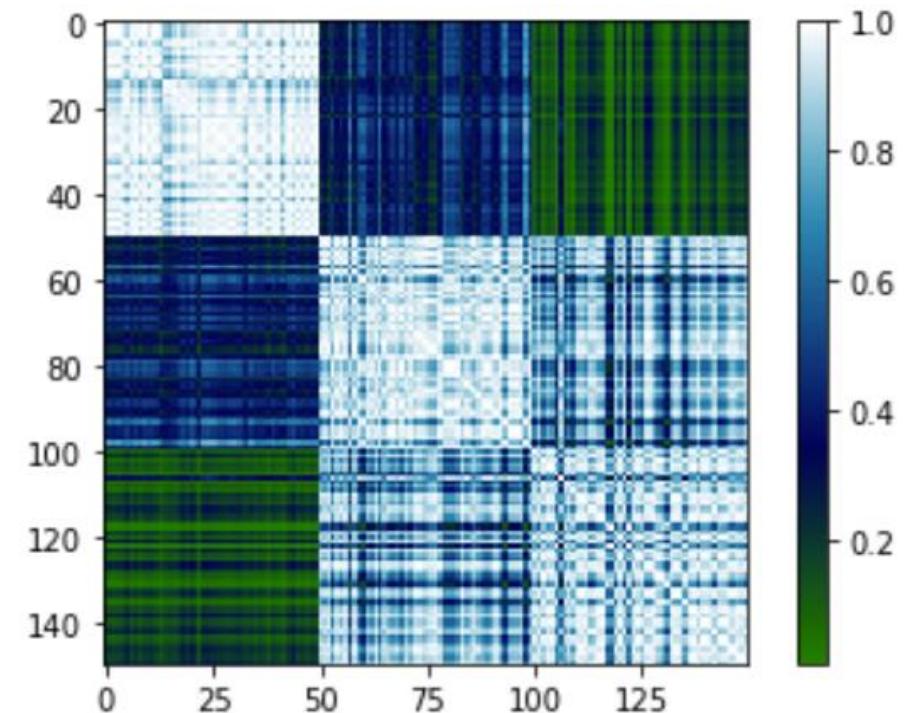
- La función Kernel entre dos datos puede ser entendida como una matriz. A tal matriz se le llama matriz de Gram (similitud de los datos).

Ejemplo: Función de base radial

```
display(X)
array([[5.1, 3.5, 1.4],
       [4.9, 3. , 1.4],
       xi [4.7, 3.2, 1.3],
       [4.6, 3.1, 1.5],
       [5. , 3.6, 1.4],
       [5.4, 3.9, 1.7],
       [4.6, 3.4, 1.4],
       [5. , 3.4, 1.5],
       [4.4, 2.9, 1.4],
       [4.9, 3.1, 1.5],
       [5.4, 3.7, 1.5],
       xj [4.8, 3.4, 1.6],
       [4.8, 3. , 1.4],
       [4.3, 3. , 1.1],
       [5.8, 4. , 1.2],
       [5.7, 4.4, 1.5],
       [5.4, 3.9, 1.3],
       [5.1, 3.5, 1.4],
```

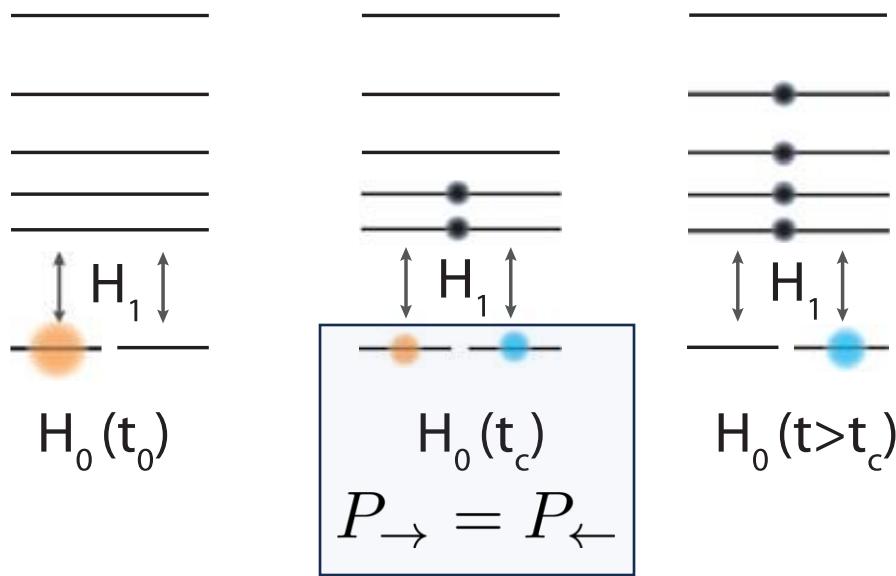
Matriz de Gram

$$K_{ij} = \exp\left(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2\right)$$

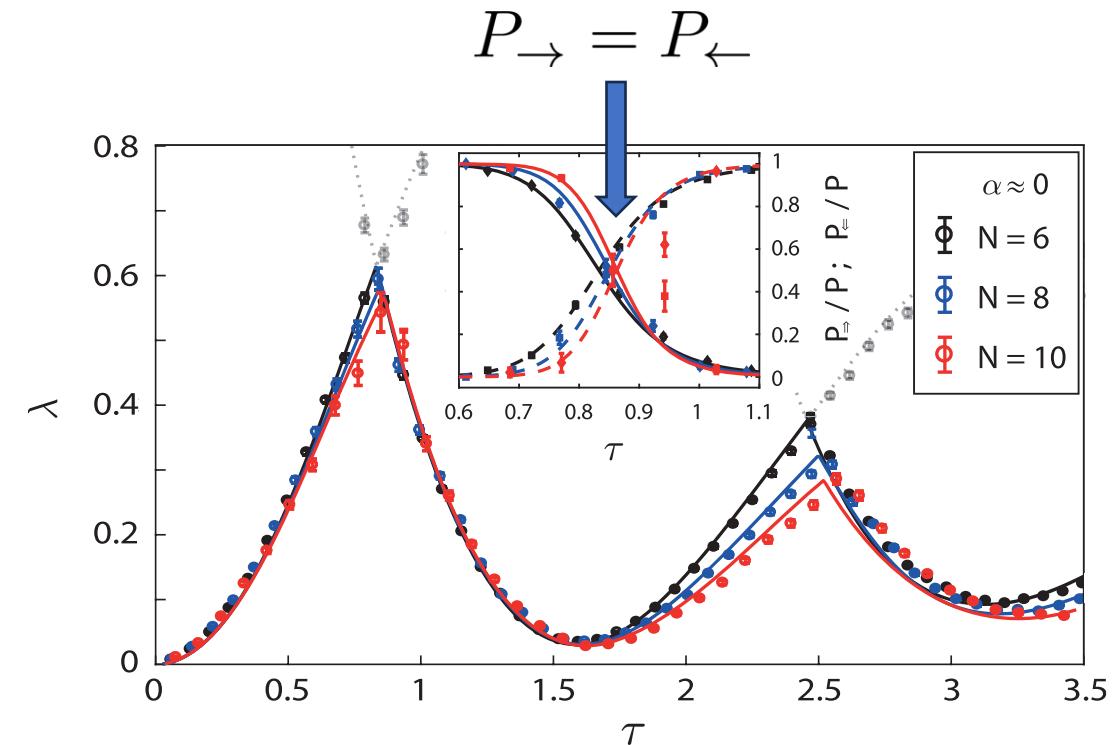


Transiciones de fase dinámica cuántica

- Es un **fenómeno dinámico** en sistemas cuánticos, donde se observa un cambio particular en el **estado de mínima energía** al activar un parámetro de control (campo magnético).
- Ha sido **confirmado experimentalmente** usando iones atrapados en el año 2017, pero a la fecha, no se ha propuesto un método de inteligencia artificial para predecir dicho fenómeno.

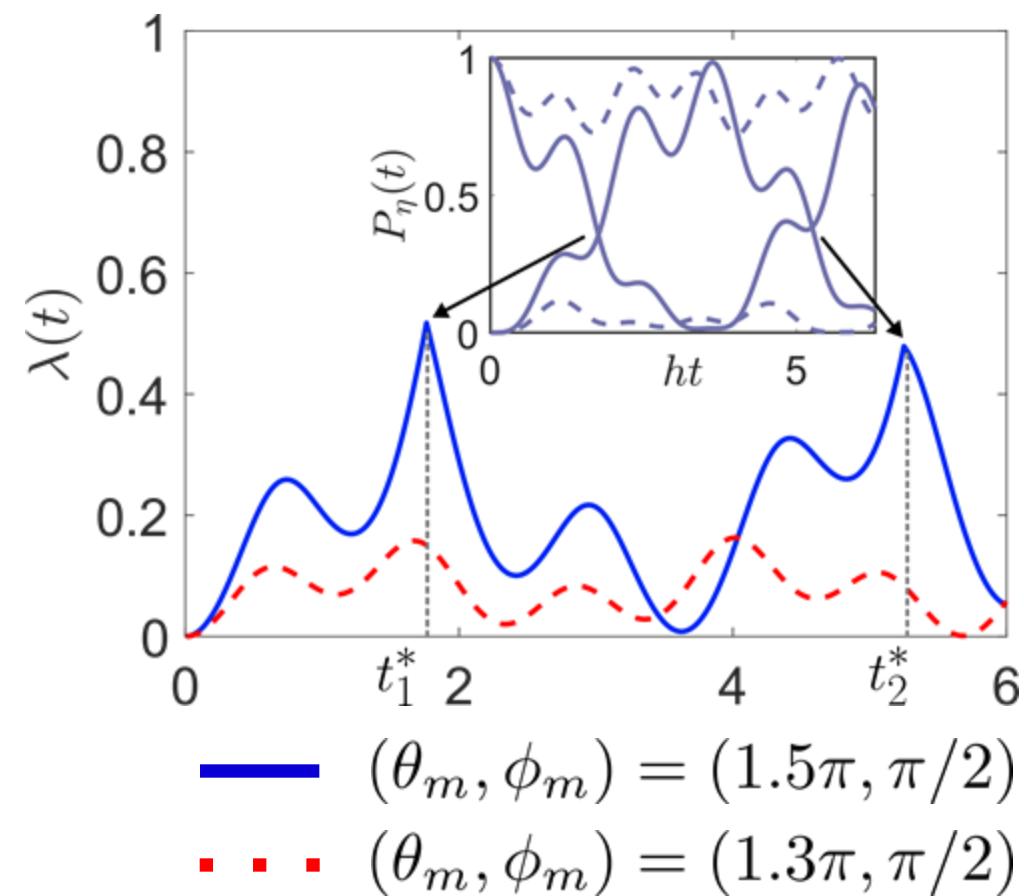
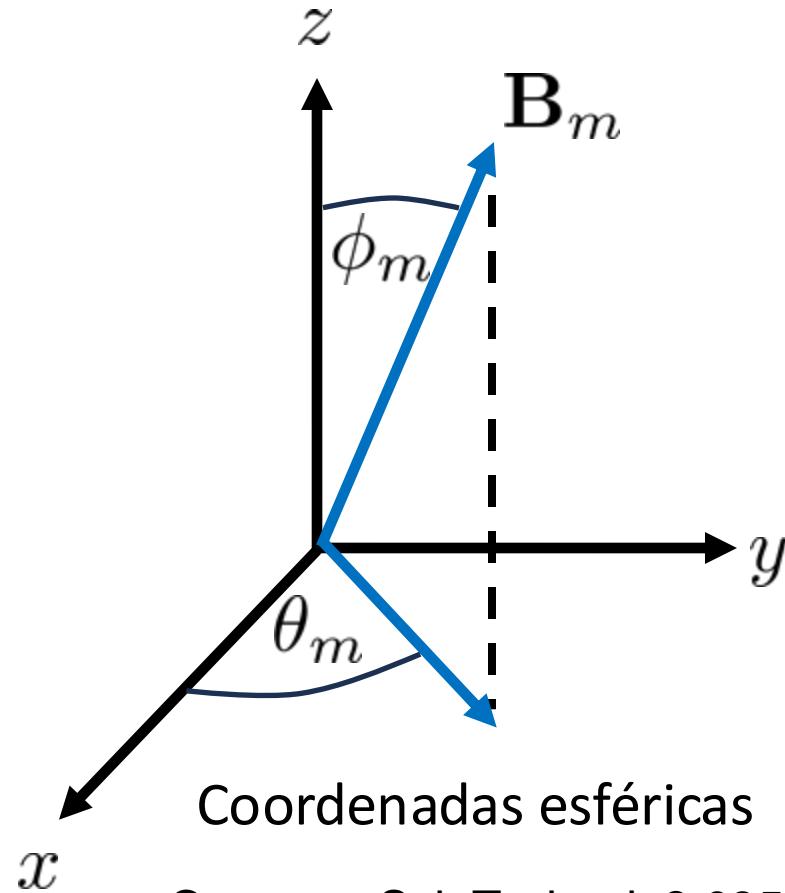


Phys. Rev. Lett. 119, 080501 (2017).



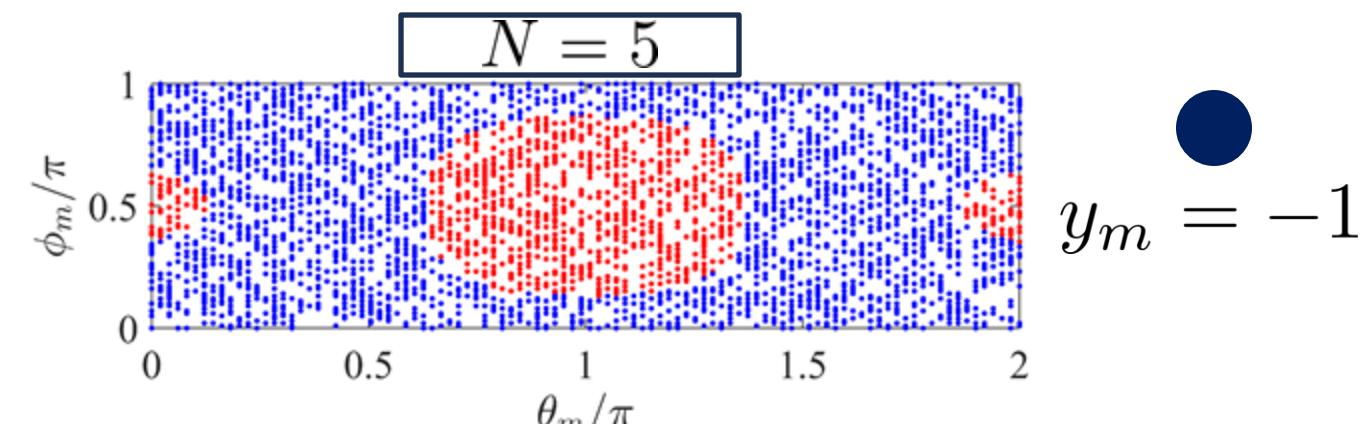
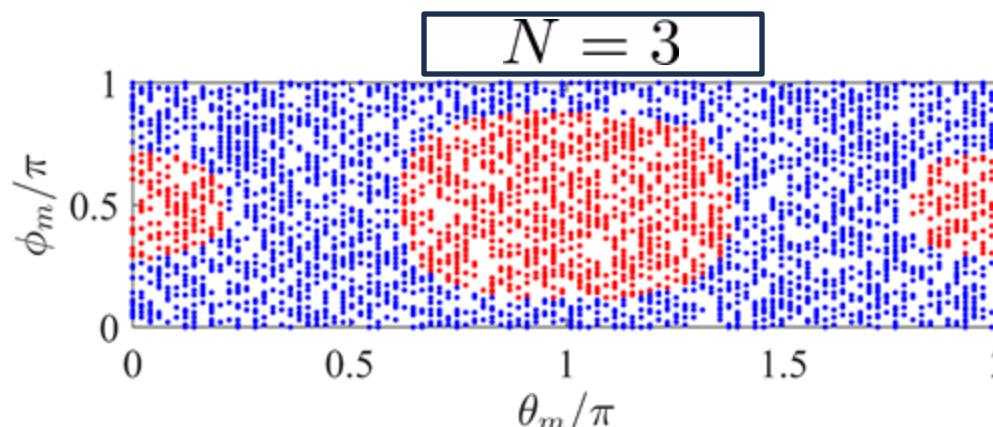
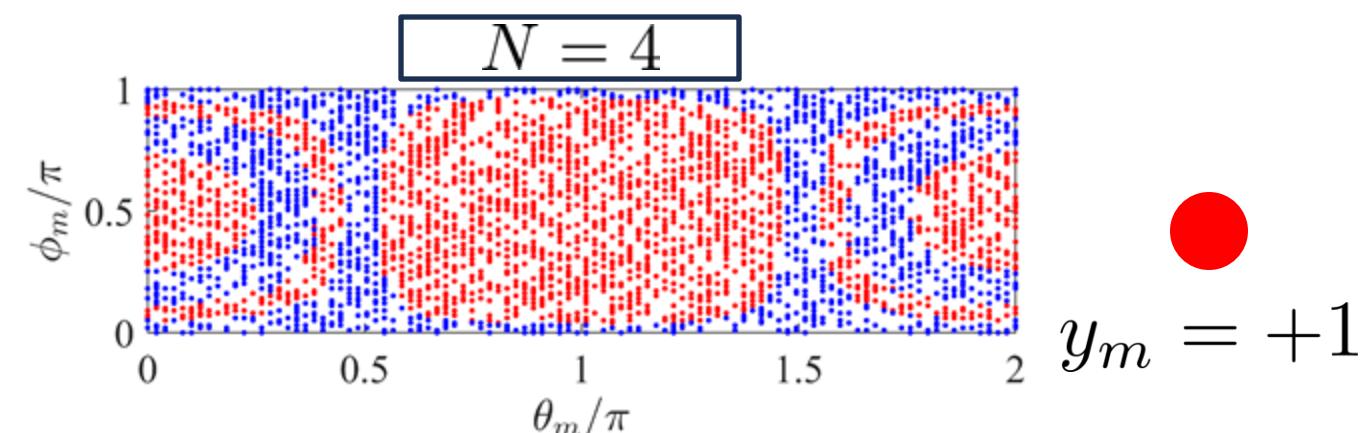
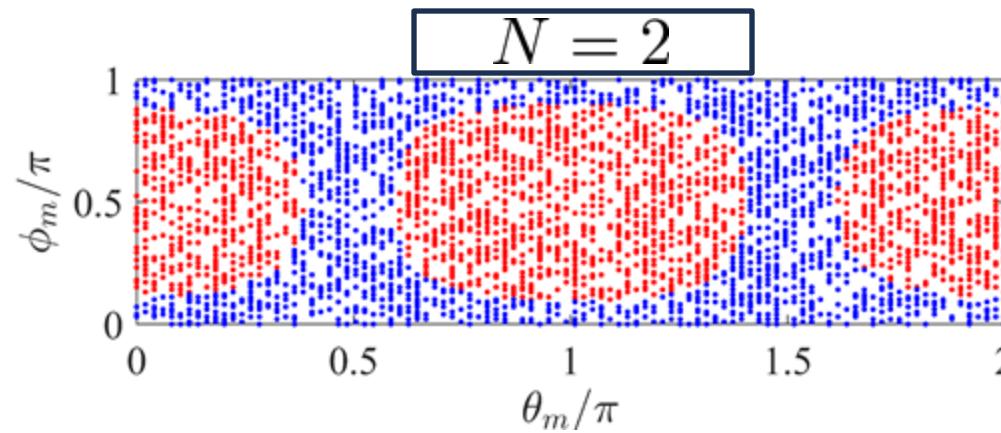
Ejemplo sobre uso de SVM en cuántica

- Fijamos un **campo magnético** para un par de ángulos (θ_m, ϕ_m) y amplitud $h_m = |\mathbf{B}_m|$
- Evolucionamos el sistema y nos fijamos **si las probabilidades son iguales** $P_+(t) = P_-(t)$
- Si las probabilidades son iguales $y_m = +1$, en caso contrario $y_m = -1$



Datos generados para modelo de clasificación

- Datos: $\mathbf{x}_m = (\theta_m, \phi_m)$ $y_m \in \{+1, -1\}$, 40.000 datos distintos para cada N



$y_m = +1$

$y_m = -1$

Kernel de base radial clásico

Método Clásico

Dato	θ_m	ϕ_m	y_m
:	:	:	:
\mathbf{x}_i	4.199315	0.789345	-1
	4.230889	0.789345	-1
	4.262462	0.789345	-1
	4.294036	0.789345	-1
\mathbf{x}_j	4.325610	0.789345	1
	4.357184	0.789345	1
	4.388758	0.789345	1
	4.420331	0.789345	1
:	:	:	:

Función de base radial

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

- El kernel se calcula directamente de los datos
- El parámetro γ se calcula por defecto

$$\gamma = \frac{1}{n_{\text{features}}}$$

Kernel de base radial cuántico

Problema de autovalores

$$\hat{H}(\mathbf{h}_m)|\Psi_n\rangle = E_n|\Psi_n\rangle$$

$$|G_m\rangle = |G_{m'}\rangle$$

$$\xrightarrow{\hspace{1cm}}$$

$$I_{m'm}^Q = 1$$

$$\begin{array}{c} \uparrow \\ |G_m\rangle \perp |G_{m'}\rangle \\ \downarrow \end{array}$$

$$I_{m'm}^Q = 0$$

Función de base radial cuántica

$$K(\mathbf{x}_m, \mathbf{x}_{m'}) = e^{-\gamma \sqrt{1 - I_{mm'}^Q}}$$

$$I_{mm'}^Q = |\langle G_m | G_{m'} \rangle|^2$$

- El kernel se calcula directamente de la función de onda del sistema dado un campo magnético.
- El parámetro $\gamma = 1$ en el caso cuántico.
- El método cuántico es **lento**, pero **si captura la física** del fenómeno cuántico.

Método clásico vs. Método cuántico

Función de base radial tradicional

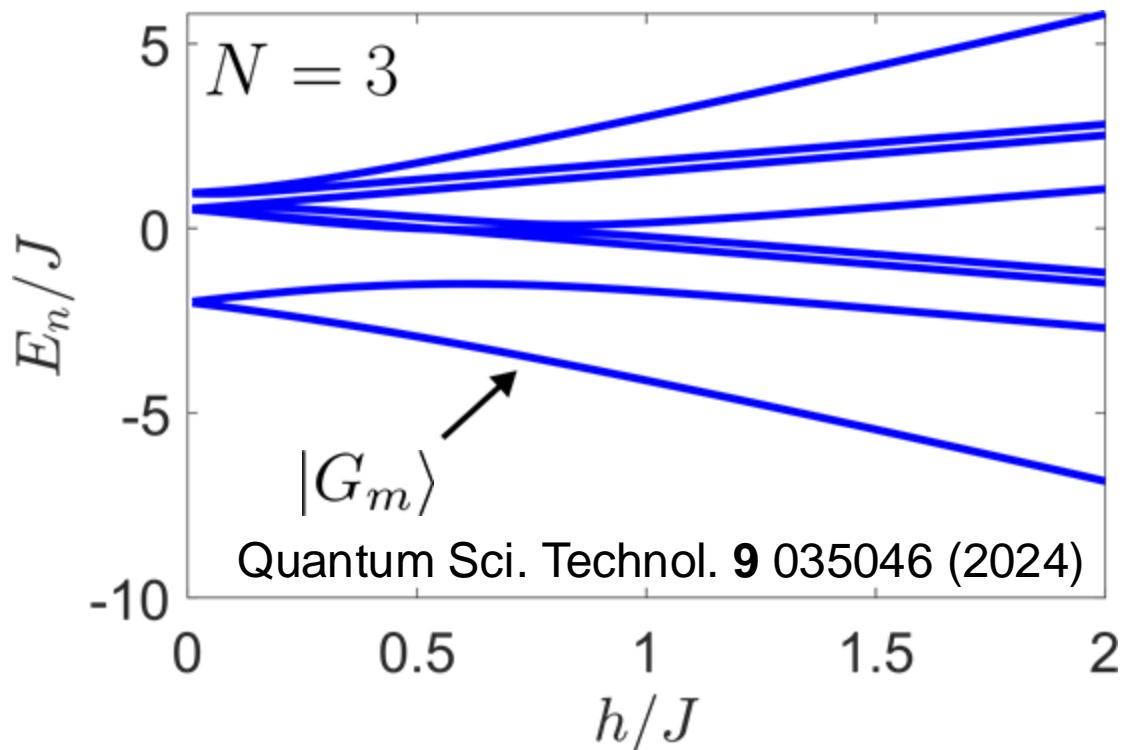
$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2}$$

Dato	θ_m	ϕ_m	y_m
:	:	:	:
\mathbf{x}_i	4.199315	0.789345	-1
	4.230889	0.789345	-1
	4.262462	0.789345	-1
	4.294036	0.789345	-1
	4.325610	0.789345	1
	4.357184	0.789345	1
	4.388758	0.789345	1
	4.420331	0.789345	1
:	:	:	:

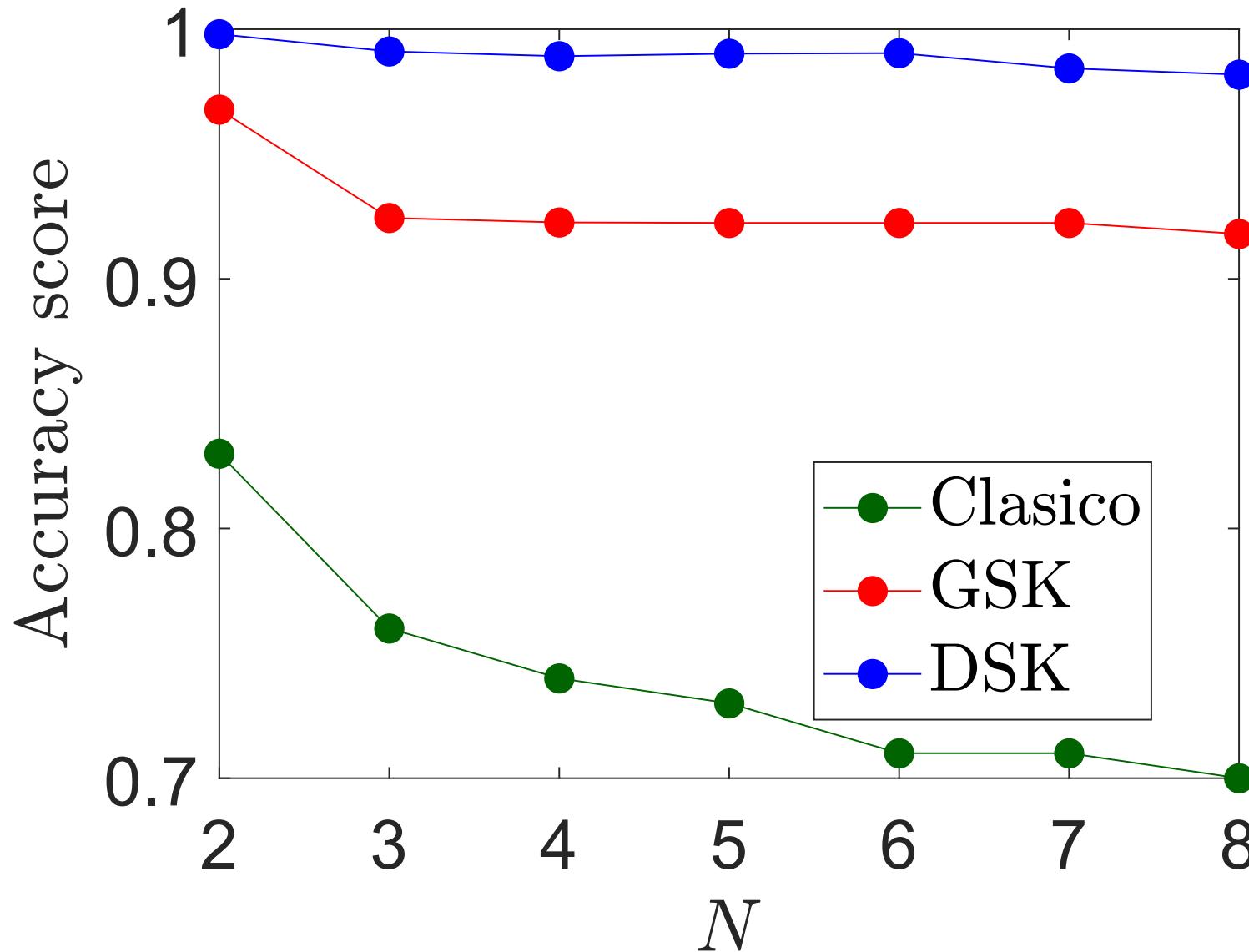
Función de base radial cuántica

$$K(\mathbf{x}_m, \mathbf{x}_{m'}) = e^{-\gamma \sqrt{1 - I_{mm'}^Q}}$$

$$I_{mm'}^Q = |\langle G_m | G_{m'} \rangle|^2$$



Problema de clasificación



Métodos cuánticos
DSK y GSK





Desafíos

El problema de entrada

“Aunque los algoritmos cuánticos pueden acelerar el procesamiento de datos, **el tiempo para leer esos datos** de manera clásica puede llegar a ser tan elevado que anule parte de la ventaja cuántica”

LOADING.....


$$X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \rightarrow \quad |\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle$$

El problema de salida

“Para **extraer la solución completa en forma de bits**, a veces se requieren resultados de tamaño exponencial. Como alternativa, podría bastar con obtener solo estadísticas resumidas del estado cuántico”

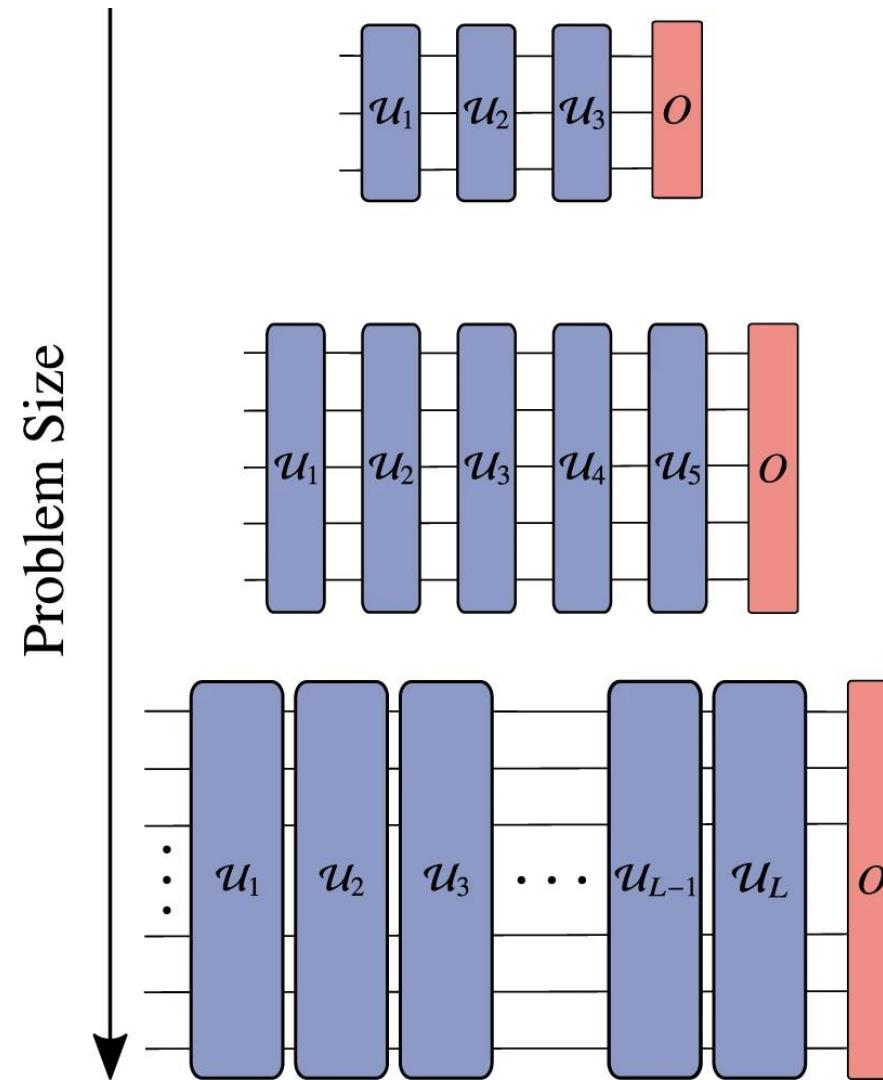
$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle$$



$$[s_1, s_2, \dots, s_N]$$

El problema de estimación de costos

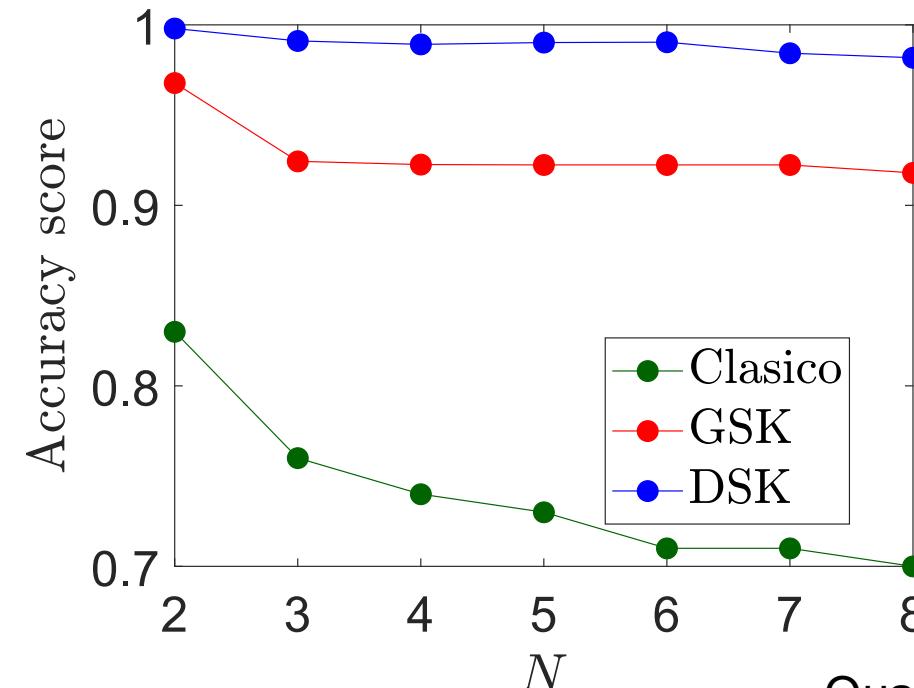
“Se sabe muy poco sobre el **número real de puertas cuánticas necesarias** para los algoritmos de aprendizaje cuántico. Las estimaciones sugieren grandes ventajas para problemas suficientemente grandes, pero no está claro cuándo ocurre ese punto de inflexión.”



El problema de evaluación

“Demostrar que un algoritmo cuántico supera a todos los métodos clásicos conocidos implica comparaciones exhaustivas. Resultados teóricos que establezcan límites inferiores podrían ayudar para abordar esta dificultad”

Biamonte, J., Wittek, P., Pancotti, N. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017)





Muchas gracias