Aprendizaje Automático Cuántico II: Redes Neuronales

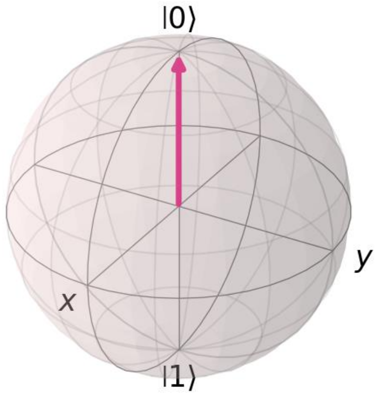
Luciano Pereira

ICFO - Institut de Ciencies Fotoniques, Spain

La **Computación Cuántica** es un paradigma de computación que emplea **sistemas cuánticos** para procesar información.

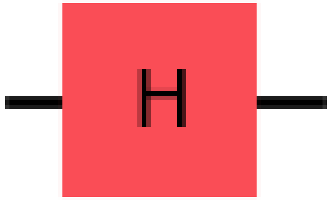
Qubit

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



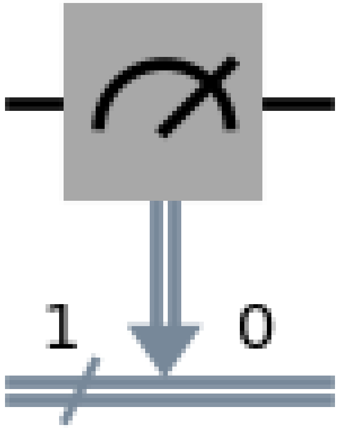
Unitarias

$$U$$

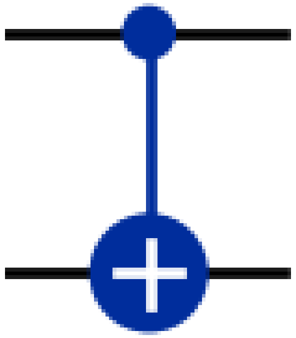


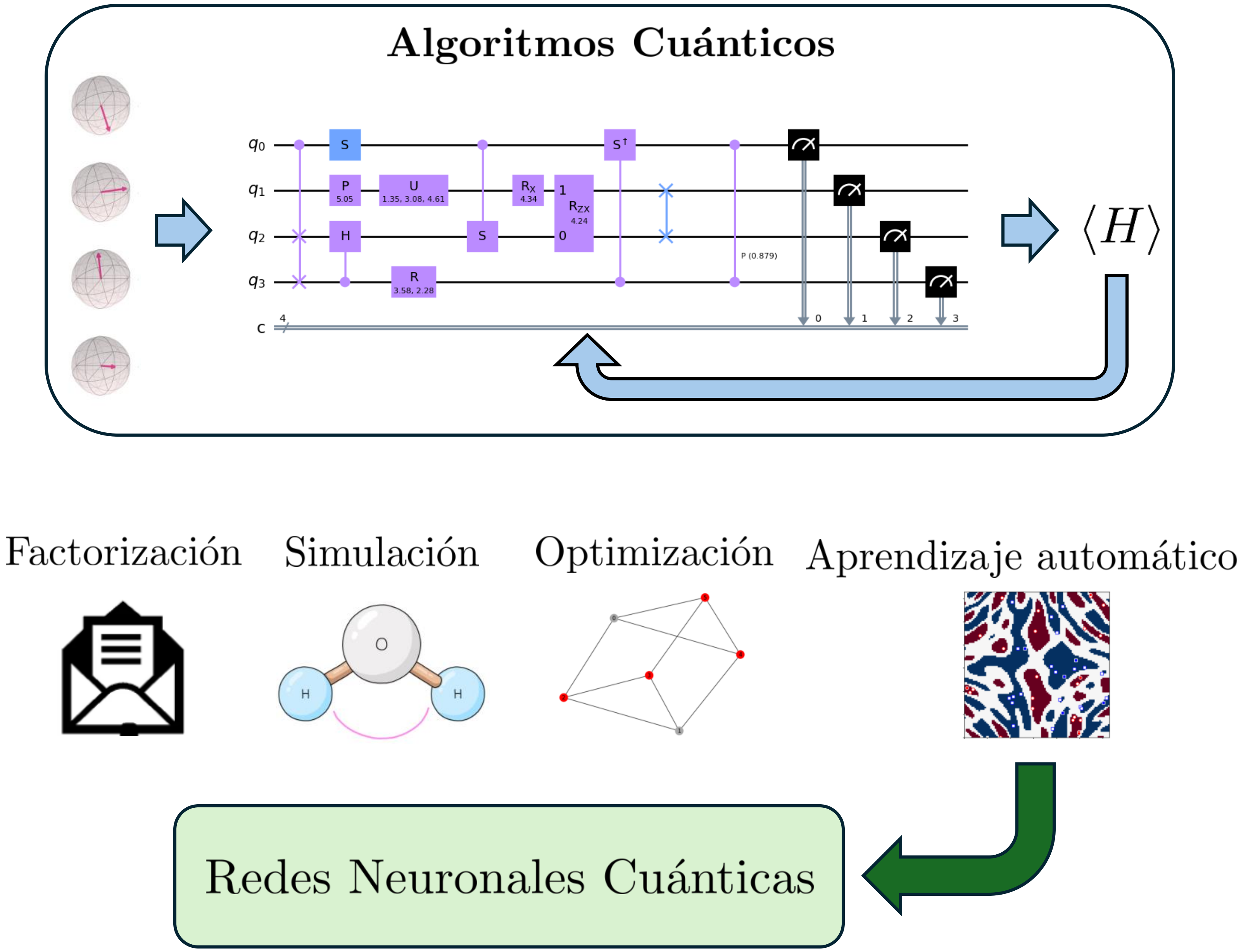
Mediciones

$$\{|m\rangle\}$$



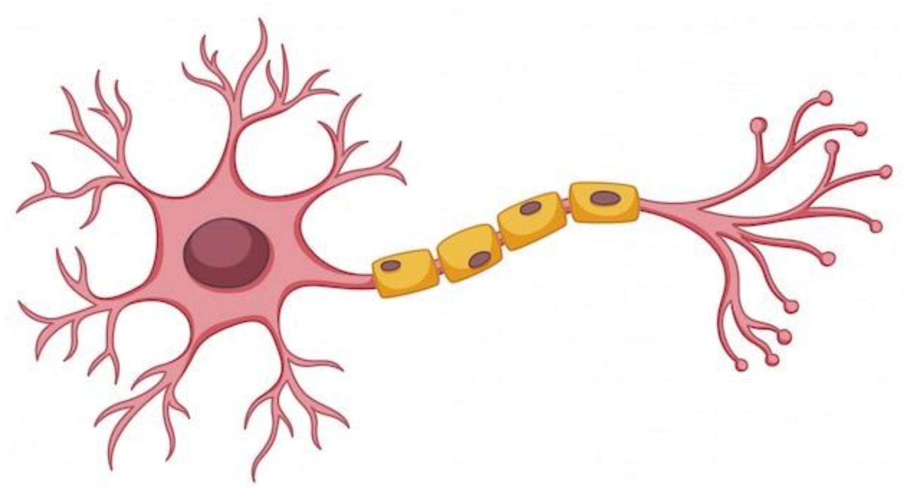
Entrelazamiento

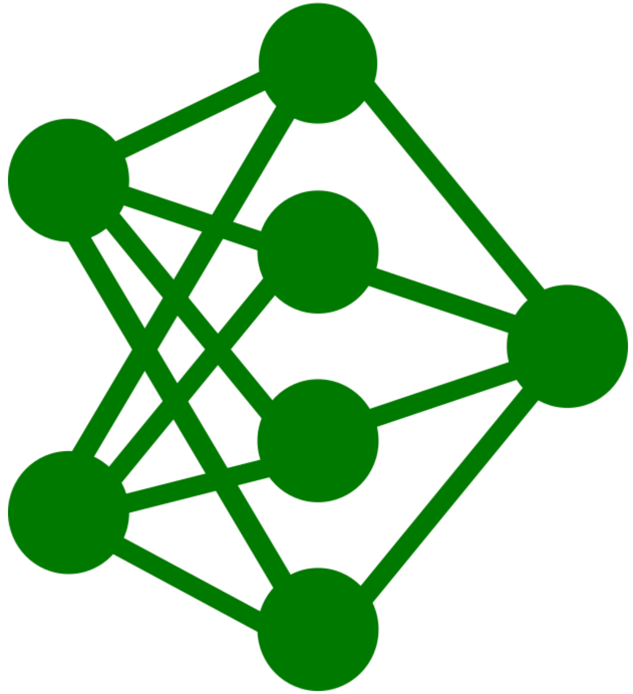



Algoritmos Cuánticos
q_0
q_1
q_2
q_3
c
4
S
P
5.05
U
1.35, 3.08, 4.61
H
R
3.58, 2.28
S
R_X
4.34
1
R_{ZX}
4.24
0
$S^\dagger$
P (0.879)
0
1
2
3
$\langle H \rangle$
Factorización
Simulación
Optimización
Aprendizaje automático
O
H
H
Redes Neuronales Cuánticas

Redes neuronales artificiales

Modelo de aprendisaje automático compuesto por un conjunto de nodos interconectados que imitan una red neuronal cerebral. Estos nodos son llamados neuronas.





Neurona Artificial: Perceptron

Función con entrada $\vec{x} = (x_1, \cdots, x_n)$
y con salida y tal que,

$$y = K\left(\sum_i \omega_i x_i + b_i\right)$$

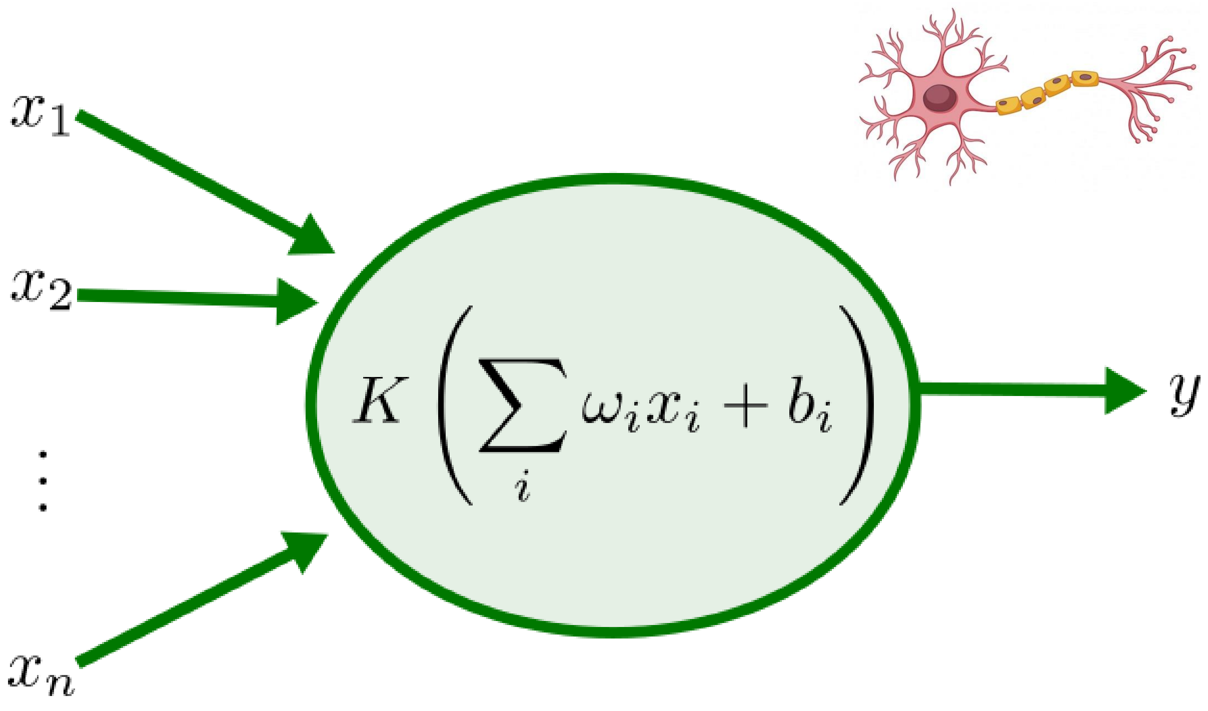
donde $\{w_i, b_i\}$ son coeficientes reales
y K es la función de activación.

Neurona Artificial: Perceptron

Función con entrada $\vec{x} = (x_1, \cdots, x_n)$
y con salida y tal que,

$$y = K\left(\sum_i \omega_i x_i + b_i\right)$$

donde $\{w_i, b_i\}$ son coeficientes reales
y K es la función de activación.

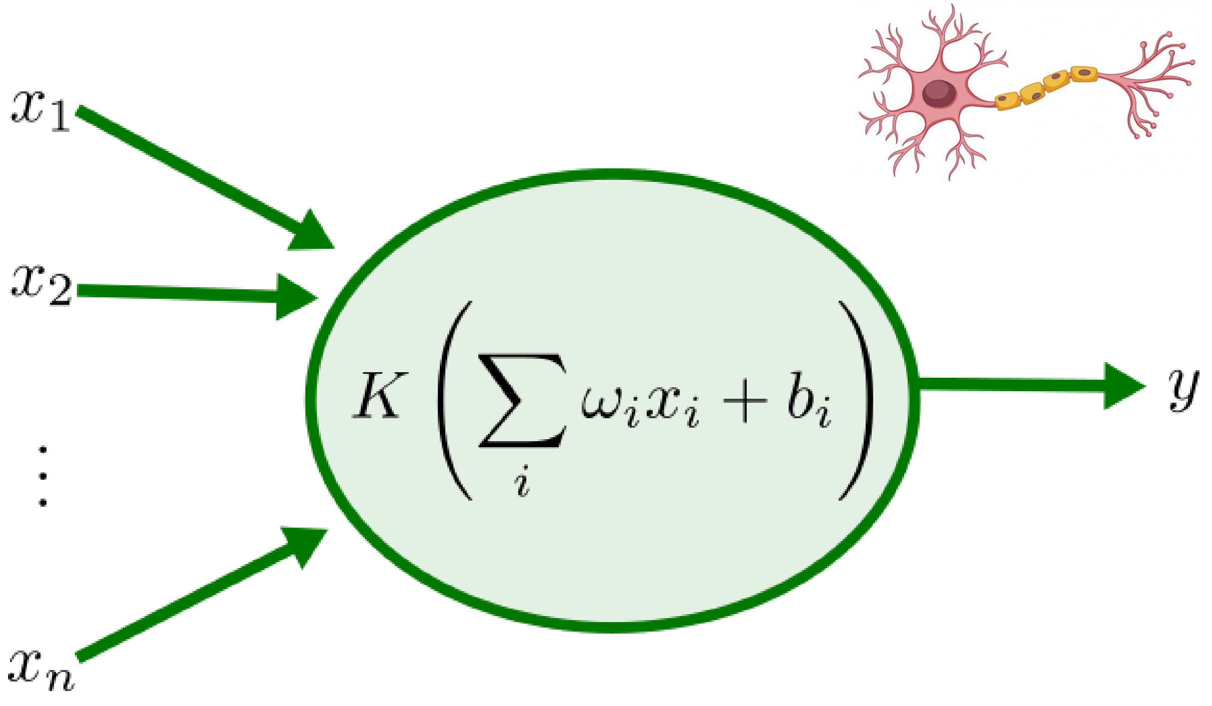


Neurona Artificial: Perceptron

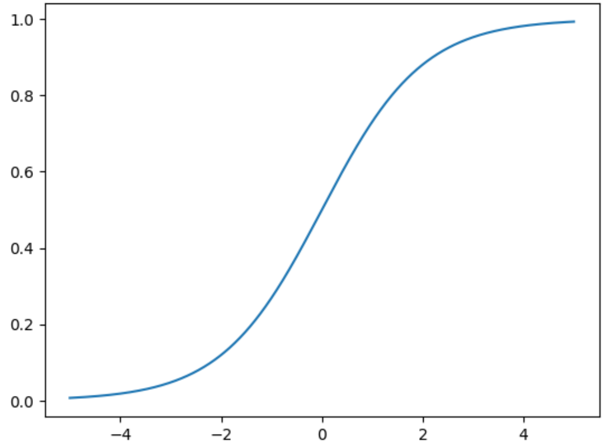
Función con entrada $\vec{x} = (x_1, \cdots, x_n)$
y con salida y tal que,

$$y = K\left(\sum_i \omega_i x_i + b_i\right)$$

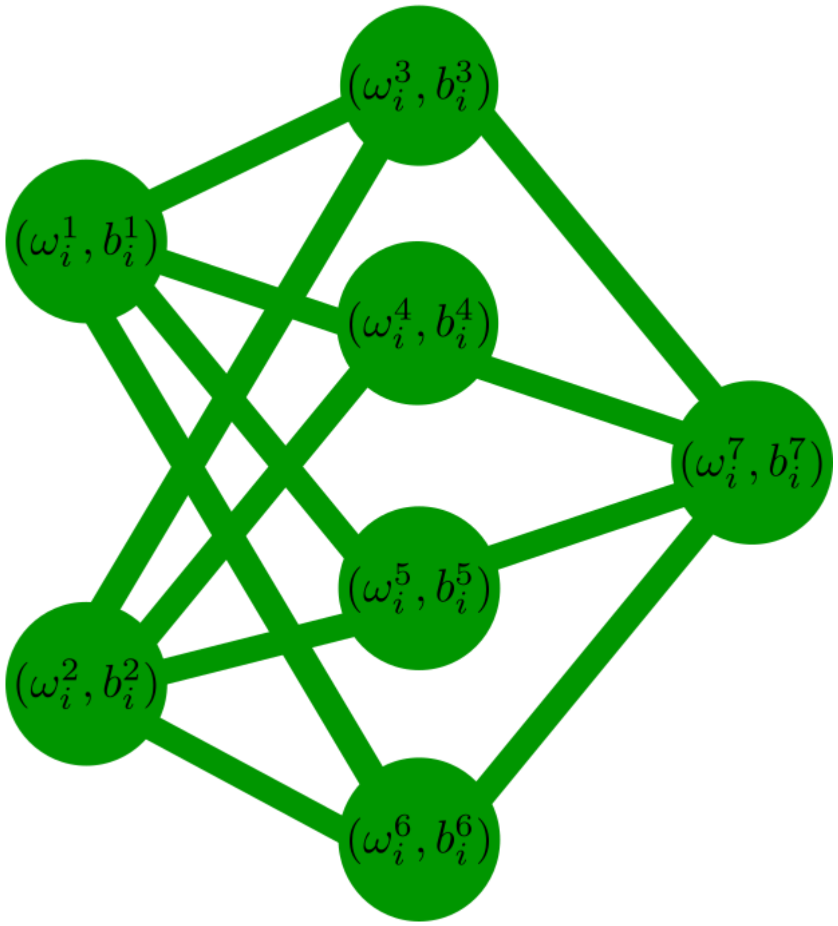
donde $\{w_i, b_i\}$ son coeficientes reales
y K es la función de activación.



Sigmoid: $K(x) = \frac{1}{1+e^{-x}}$



Red neuronal artificial



$$\omega = (\omega_i^1, \cdots, \omega_i^m), \quad b = (b_i^1, \cdots, b_i^m)$$

Consideremos un conjunto de datos de entrada x_i y predicciones y_i. Sea $\mathcal{R}_{\omega,b}$ la acción de la red neuronal sobre los datos de entrada.

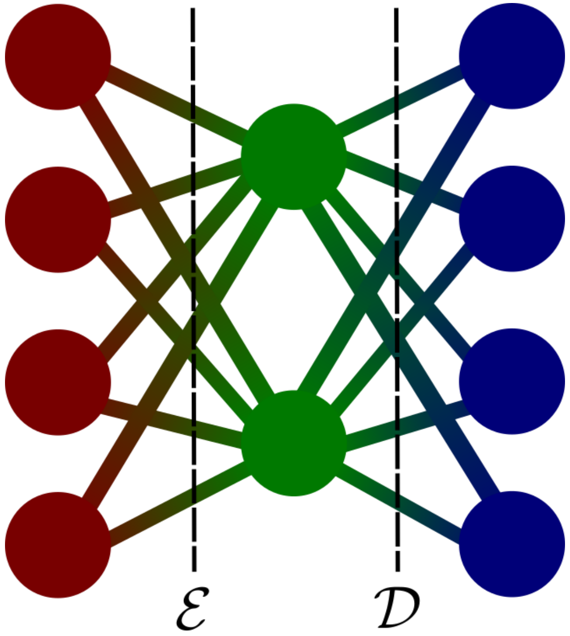
El **entrenamiento** de la red consiste en minimizar alguna función objetivo de los datos de salida sobre los parámetros de la red ω y b. Un ejemplo es el error cuadrático

$$L_{\omega,b} = \sum_i |y_i - \mathcal{R}_{\omega,b}(x_i)|^2.$$

Autoencoders

Arquitertura de red neuronal para reducir dimensión de los datos de entrada.

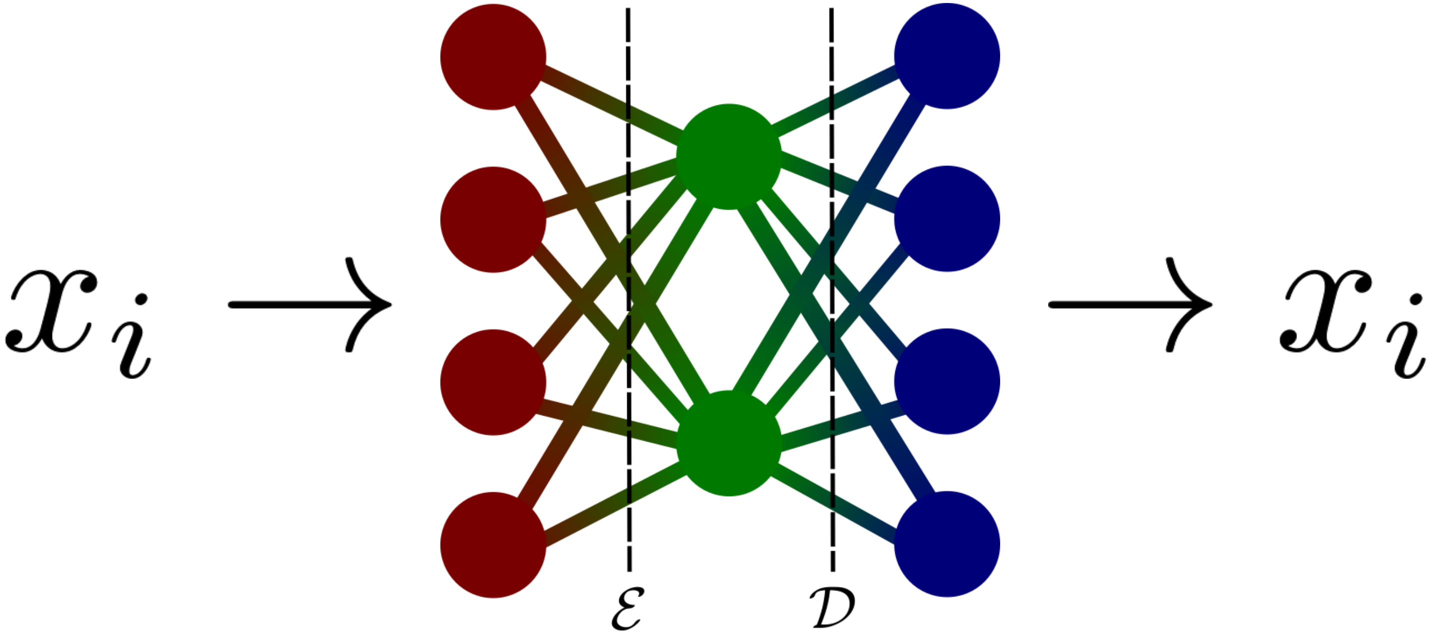
$x_i \rightarrow$



$\rightarrow x_i$

Autoencoders

Arquitertura de red neuronal para reducir dimensión de los datos de entrada.



Función objetivo para entrenar[1]:

$$L_{\omega,b} = \sum_i |x_i - \mathcal{R}_{\omega,b}(x_i)|^2.$$

Aplicaciones[1]:

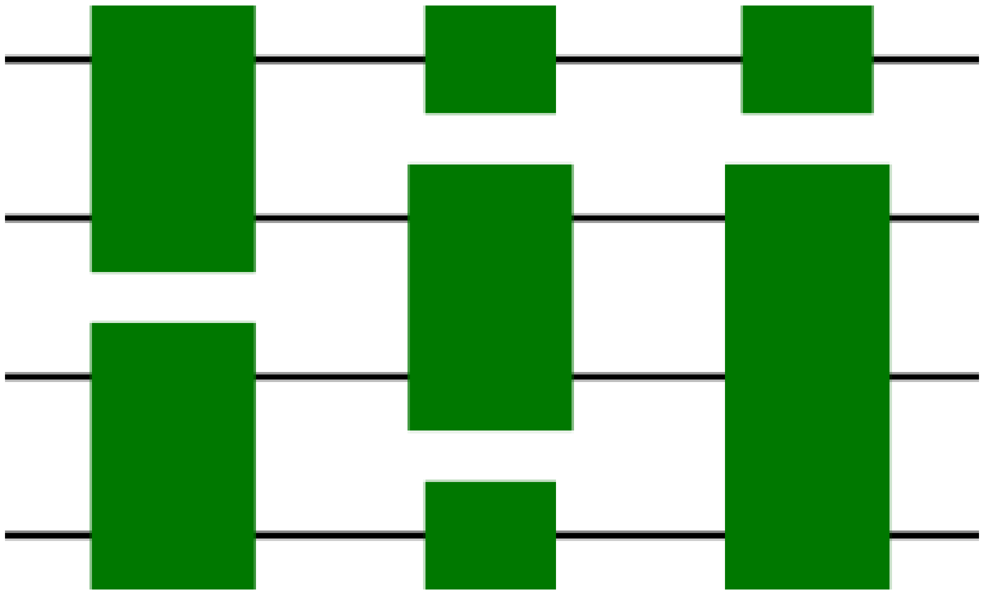
procesamiento de imagenes y lenguaje, analisis de redes, sistemas de recomendación, detección de anomalías, información cuántica[2].

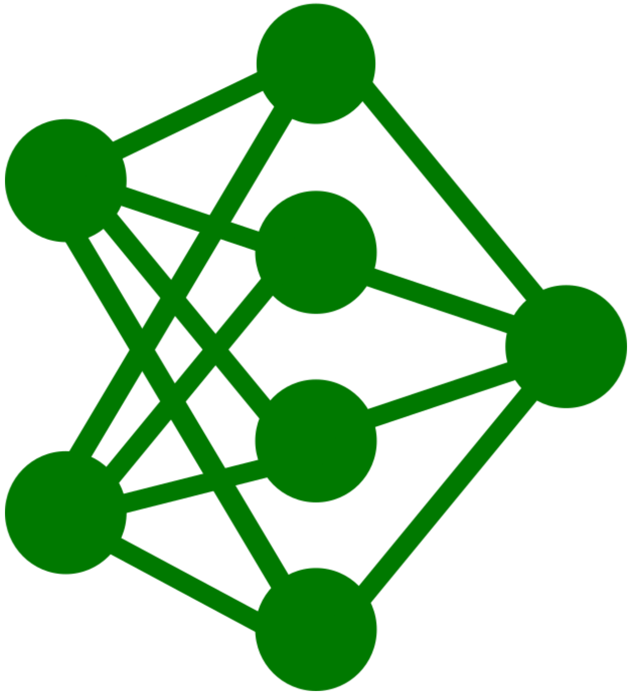
[1] Berahmand, K., Daneshfar, F., Salehi, E.S. et al. Autoencoders and their applications in machine learning: a survey. Artif Intell Rev 57, 28 (2024).

[2] D. Uzcategui-Contreras, A. Guerra, S. Niklitschek, A. Delgado, Machine Learning approach to reconstruct Density Matrices from Quantum Marginals, arXiv:2410.11145 (2024).

Redes Neuronales Cuánticas

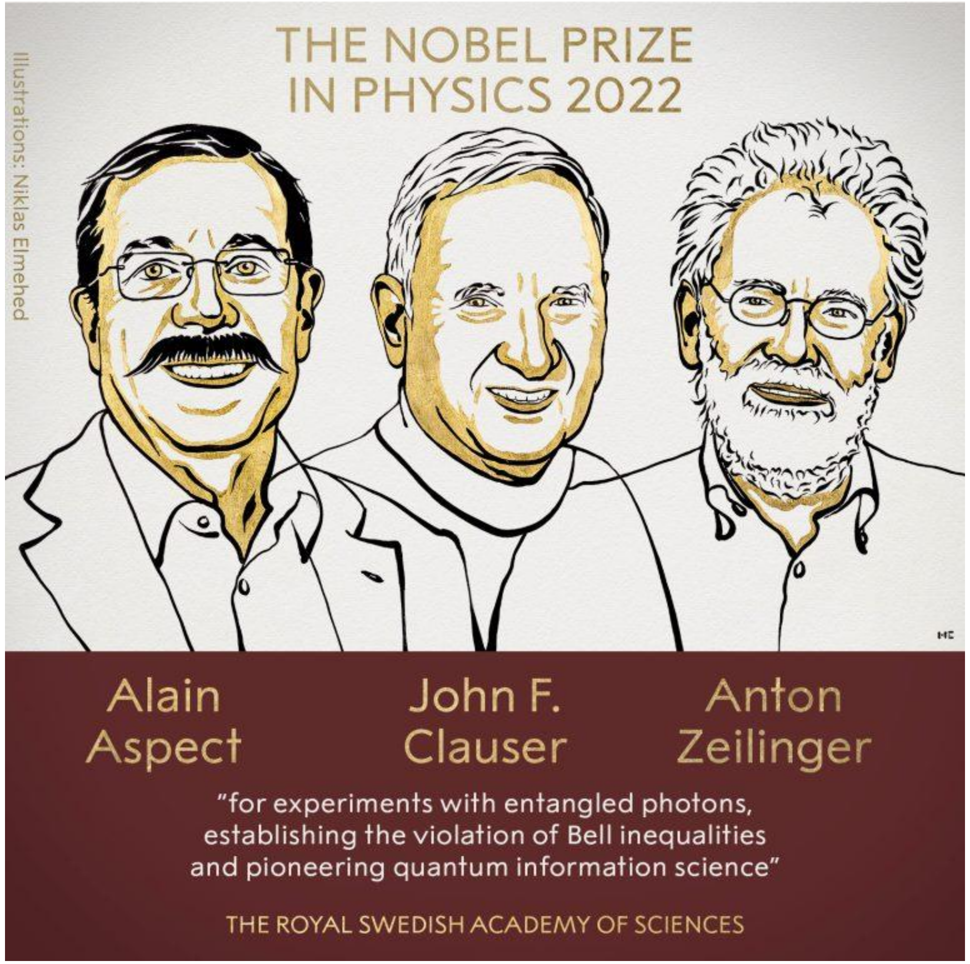
Modelo de aprendizaje automático que combina conceptos de computación cuántica y redes neuronales artificiales.

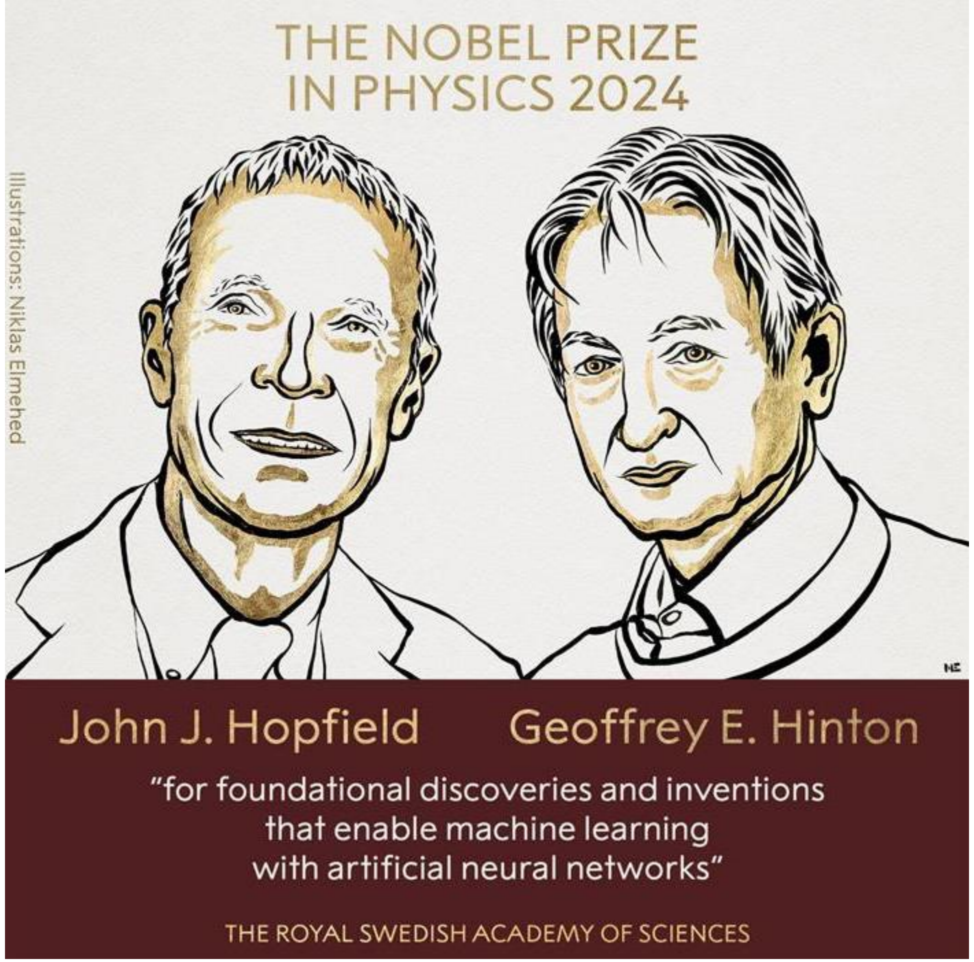




Redes Neuronales Cuánticas

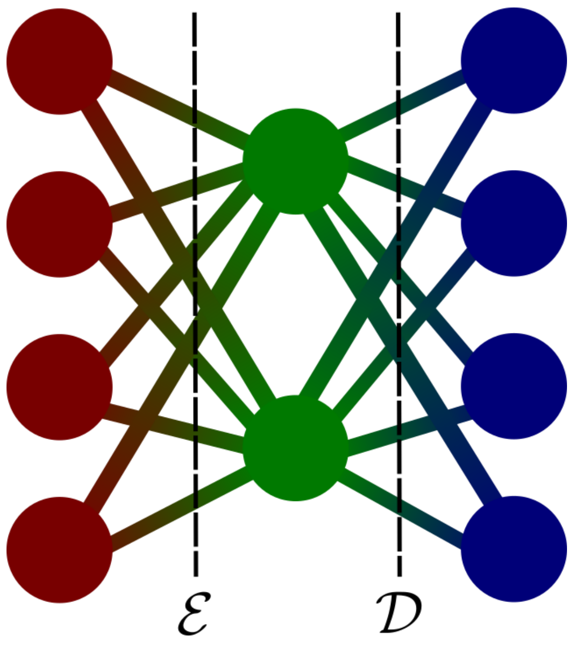
Modelo de aprendisaje automático que combina conceptos de computación cuántica y redes neuronales artificiales.



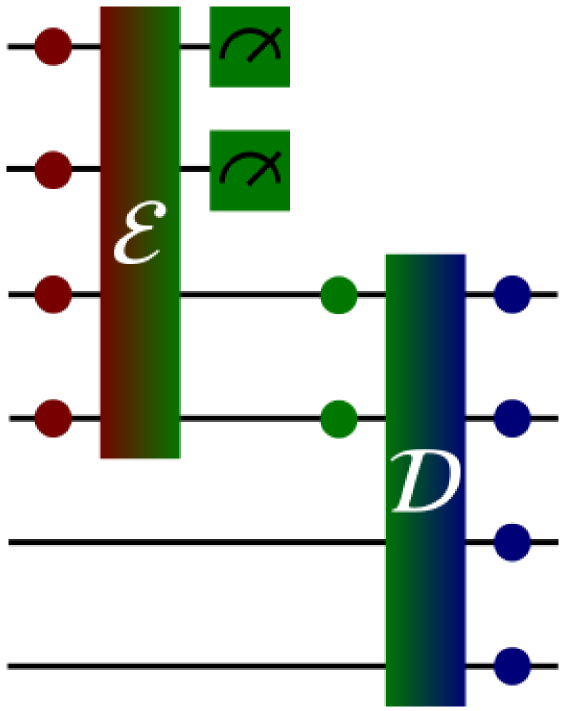


Autoencoder cuánticos

Arquitectura de red neuronal para comprimir estados cuánticos.



Clásico	**Cuántico**
Neurona $(x_1, \cdots, x_n)$	Qubit $\|\psi\rangle$
Combinación lineal y Función de activación	Operaciones unitarias y Mediciones



Matrices Densidad

Consideremos un sistema cuántico cuyo estado es $|\psi_i\rangle$ con probabilidad p_i. Decimos que el estado de este sistema es la matriz densidad

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

Esta matriz es semi-definida positiva $\rho \geq 0$ y de traza uno $\mathrm{Tr}(\rho) = 1$.

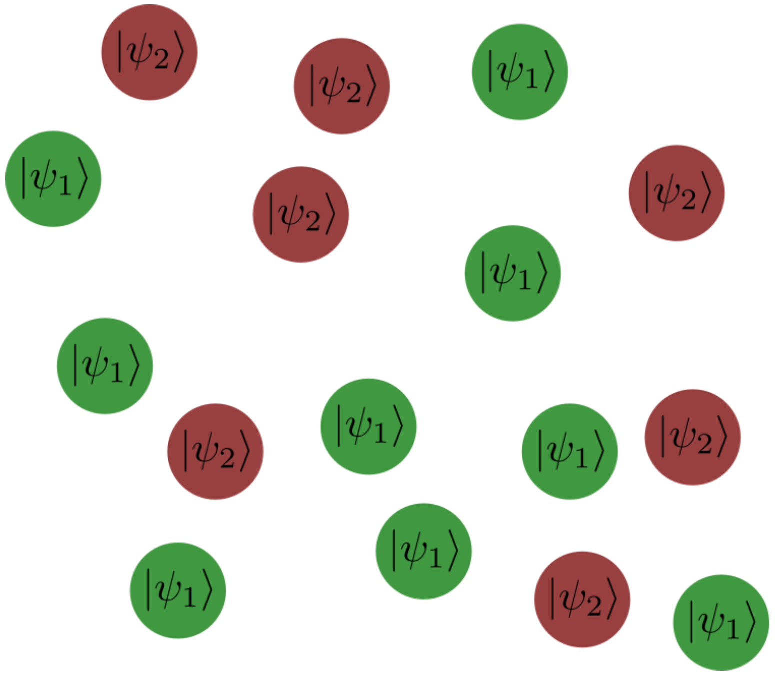
Matrices Densidad

Consideremos un sistema cuántico cuyo estado es $|\psi_i\rangle$ con probabilidad p_i.
Decimos que el estado de este sistema es la matriz densidad

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

Esta matriz es semi-definida positiva $\rho \geq 0$ y de traza uno $\mathrm{Tr}(\rho) = 1$.

$$|\psi\rangle = |0\rangle, \qquad |\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$



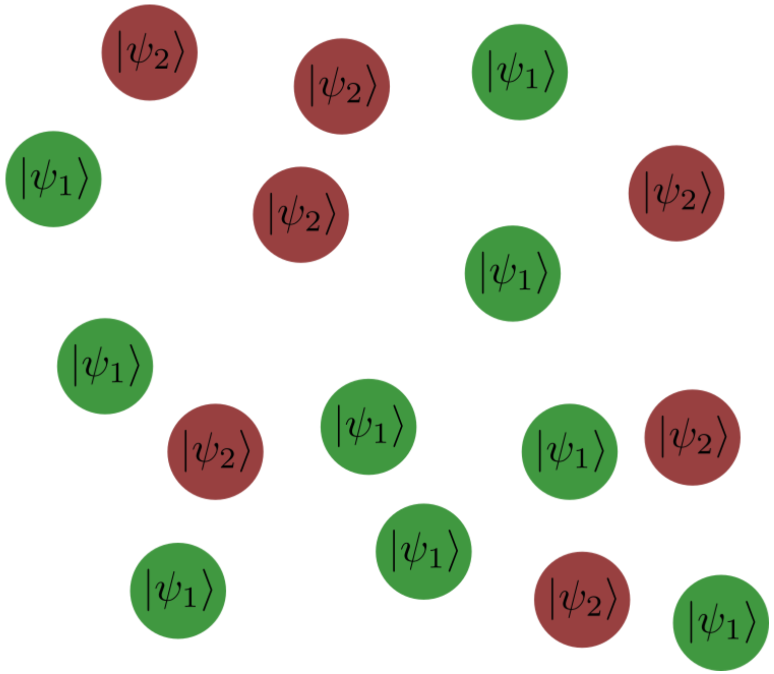
Matrices Densidad

Consideremos un sistema cuántico cuyo estado es $|\psi_i\rangle$ con probabilidad p_i. Decimos que el estado de este sistema es la matriz densidad

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

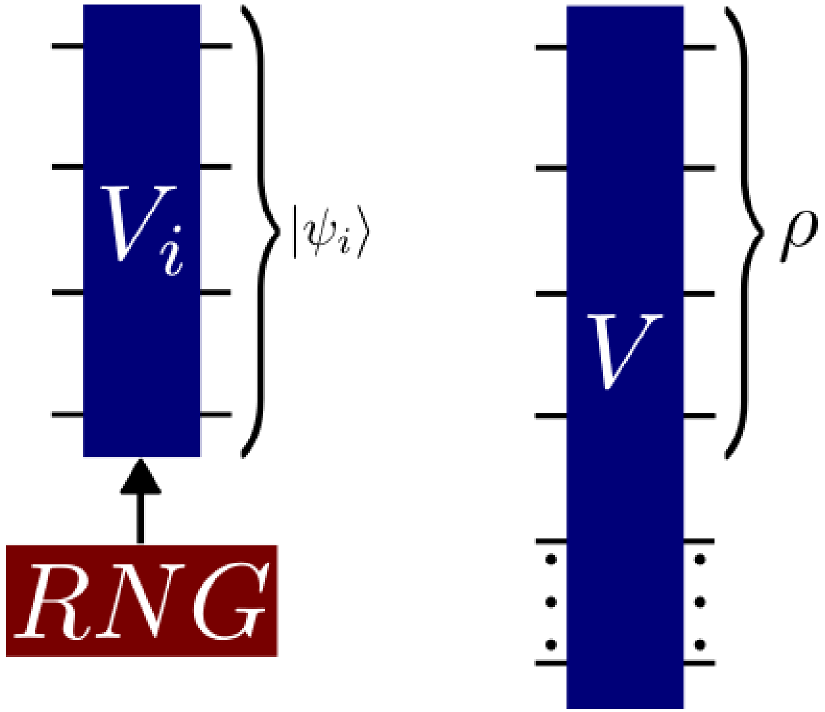
Esta matriz es semi-definida positiva $\rho \geq 0$ y de traza uno $\mathrm{Tr}(\rho) = 1$.

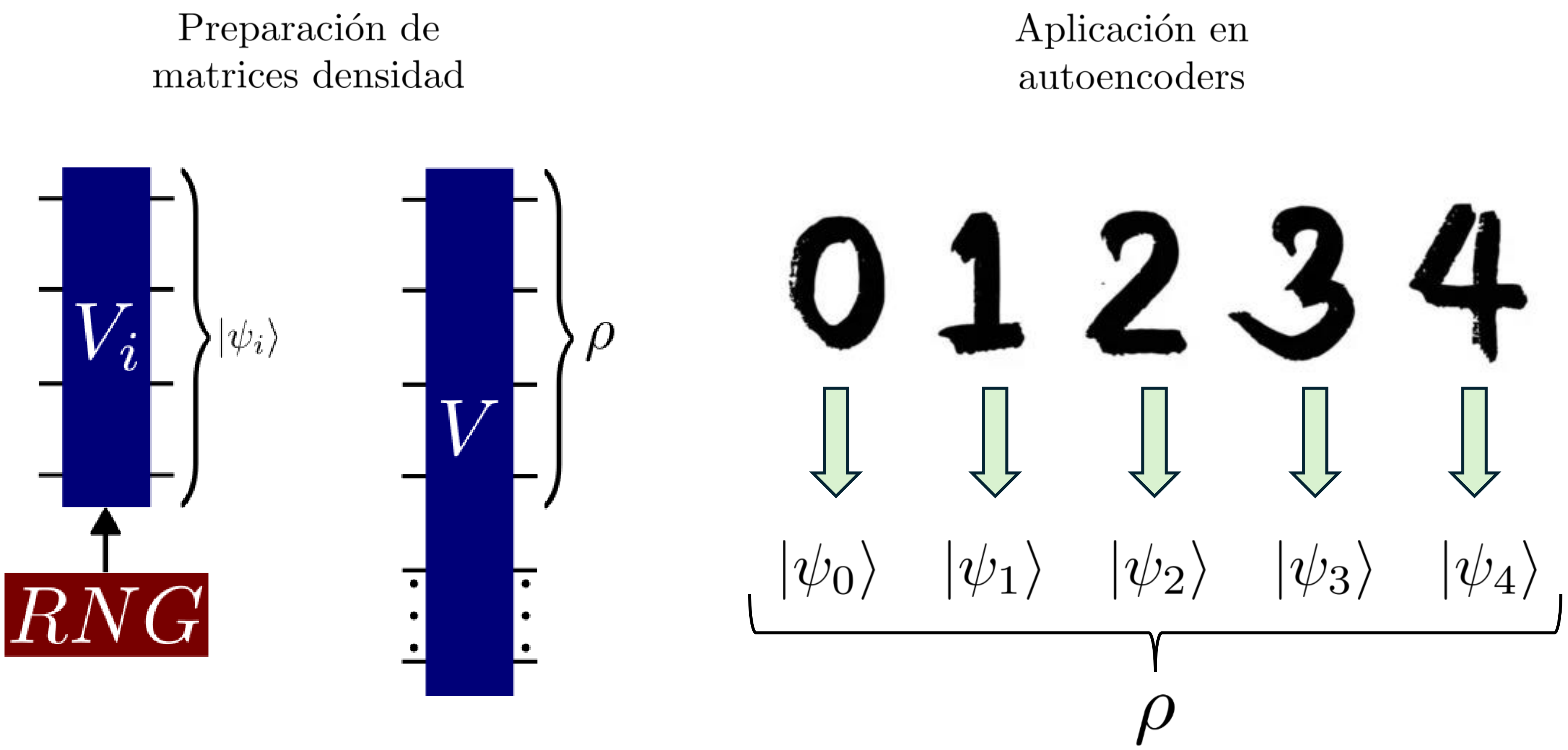
$$|\psi\rangle = |0\rangle, \qquad |\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$



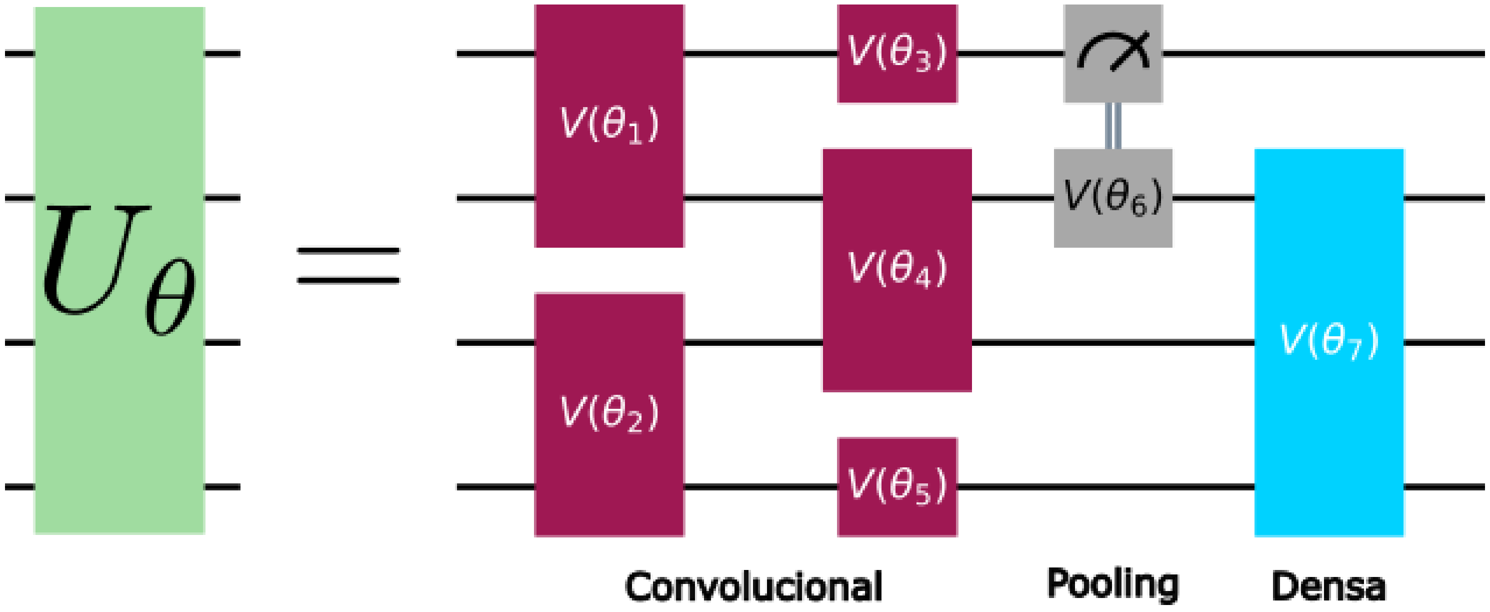
$$\begin{aligned}
\rho &= \frac{9}{16}|\psi_1\rangle\langle\psi_1| + \frac{7}{16}|\psi_2\rangle\langle\psi_2| \\
&= \frac{9}{16}|0\rangle\langle 0| + \frac{7}{32}(|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 1|) \\
&= \frac{25}{32}|0\rangle\langle 0| + \frac{7}{32}(|1\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 1|) \\
&= \begin{pmatrix} \frac{25}{32} & \frac{7}{32} \\ \frac{7}{32} & \frac{7}{32} \end{pmatrix}
\end{aligned}$$

Preparación de
matrices densidad




Preparación de matrices densidad
V_i
$|\psi_i\rangle$
RNG
V
ρ
Aplicación en autoencoders
0 1 2 3 4
$|\psi_0\rangle$ $|\psi_1\rangle$ $|\psi_2\rangle$ $|\psi_3\rangle$ $|\psi_4\rangle$
ρ

Para el encoding $\mathcal{E}$ utilizamos una transformación unitaria parametrica $U(\theta)$.
Como U_θ es invertible, el decoding $\mathcal{D}$ viene dado por $U_\theta^\dagger$.



I. Cong, S. Choi, and M.D. Lukin, Nat. Phys. 15, 1273–1278 (2019).

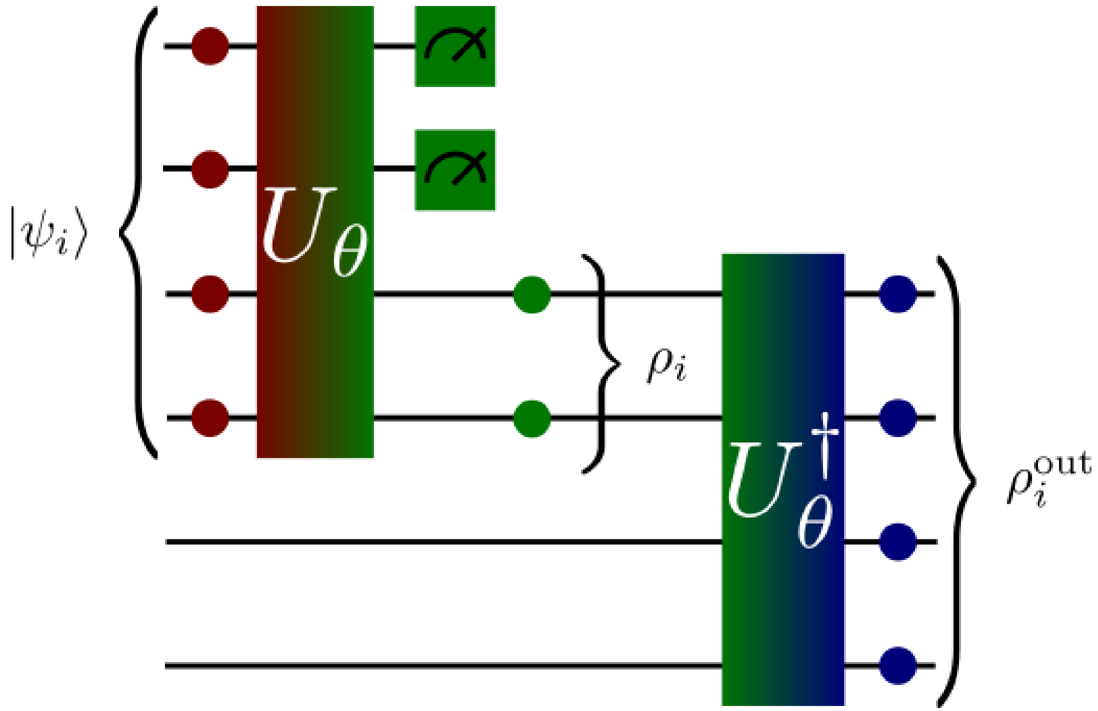
El entrenamiento consiste en maximizar la **fidelidad** entre el estado de entrada $|\psi_i\rangle$ y el estado de salida ρ_i^{out},

$$F(|\psi_i\rangle, \rho_i^{\text{out}}) = \langle\psi_i|\rho_i^{\text{out}}|\psi_i\rangle.$$

Si $F = 1$ entonces $|\psi_i\rangle\langle\psi_i| = \rho_i^{\text{out}}$.

Si queremos aplicar el autoencoder cuántico a un ensamble de estados $\{p_i, |\psi_i\rangle\}$, debemos maximizar la **fidelidad promedio**,

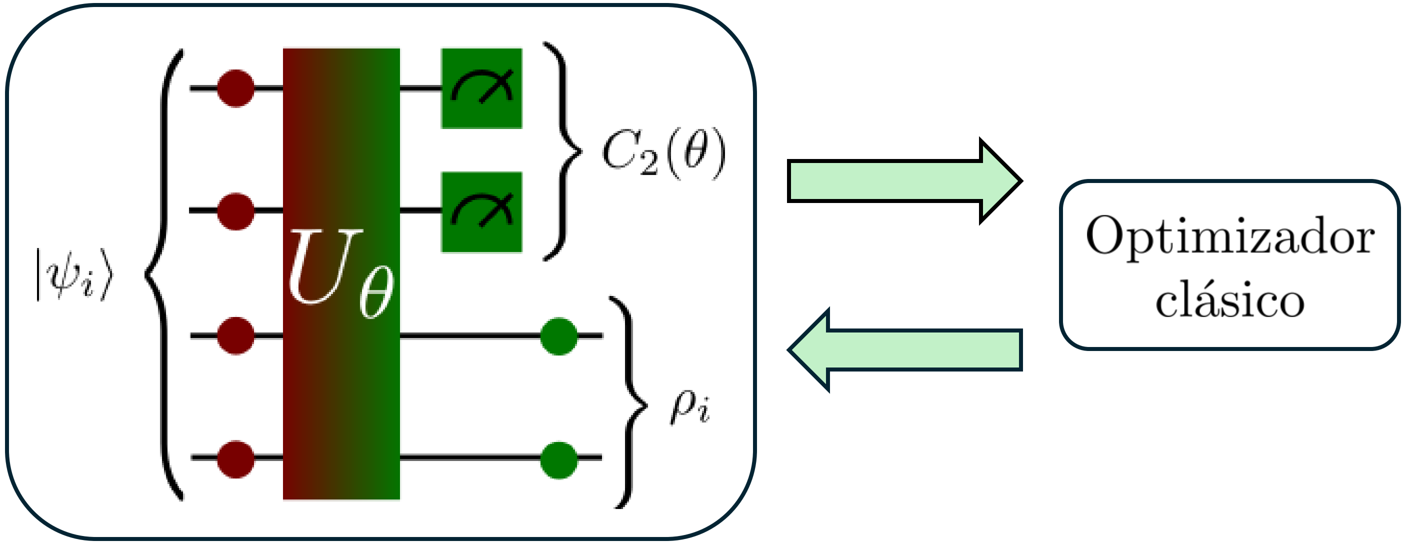
$$C(\theta) = \sum_i p_i F(|\psi_i\rangle, \rho_i^{\text{out}}).$$



Equivalentemente, podemos optimizar la siguiente función objetivo,

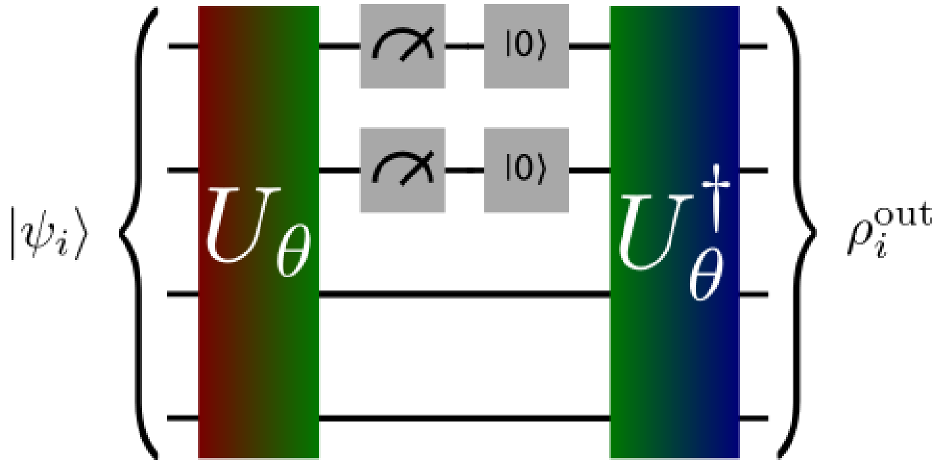
$$C_2(\theta) = \sum_i p_i \mathrm{Tr}\left[(I \otimes |0\cdots 0\rangle\langle 0\cdots 0|)(U|\psi_i\rangle\langle\psi_i|U^\dagger)\right].$$

Esto es quivalente a medir $|0\cdots 0\rangle$ en los qubits descartados en la capa oculta para cada estado $|\psi_i\rangle$.

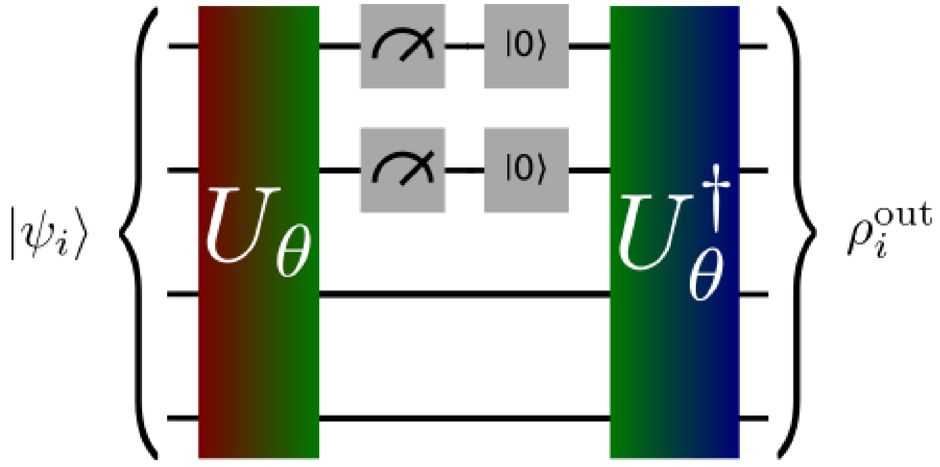


J. Romero, J. P. Olson and A. Aspuru-Guzik, Quantum Sci. Technol. 2, 045001 (2017)

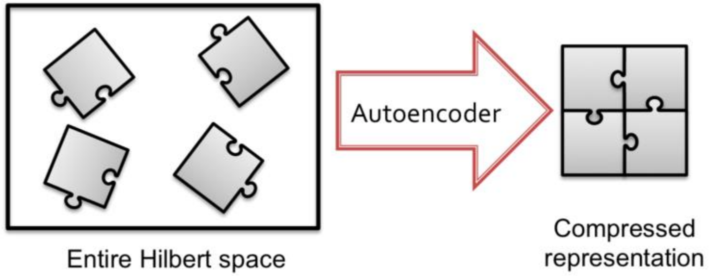
Podemos ahorrarnos qubits en la **decodificación** través del siguiente circuito.



Podemos ahorrarnos qubits en la **decodificación** través del siguiente circuito.

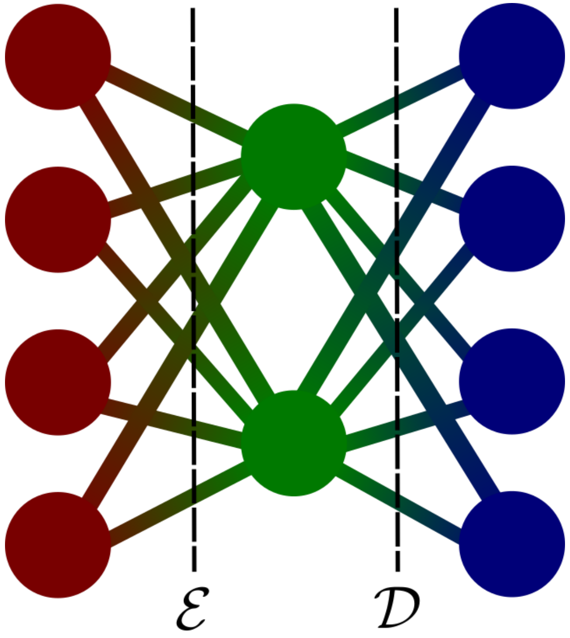


Aplicaciones:
Compresión de estados cuántico.

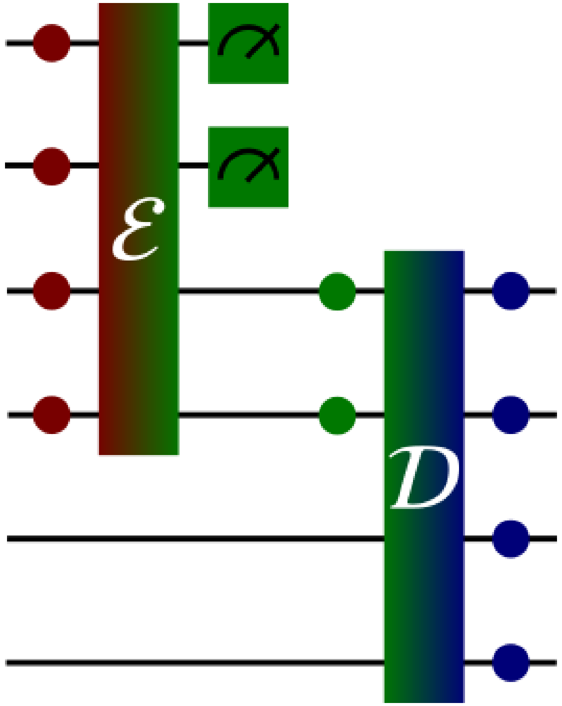


Estados cuánticos de atómos con M electrones en N modos son generado por un subespacio propio de dimensión $d = \binom{N}{M}$.

Estos estados pueden ser reducidos a $\log_2(d)$ qubits



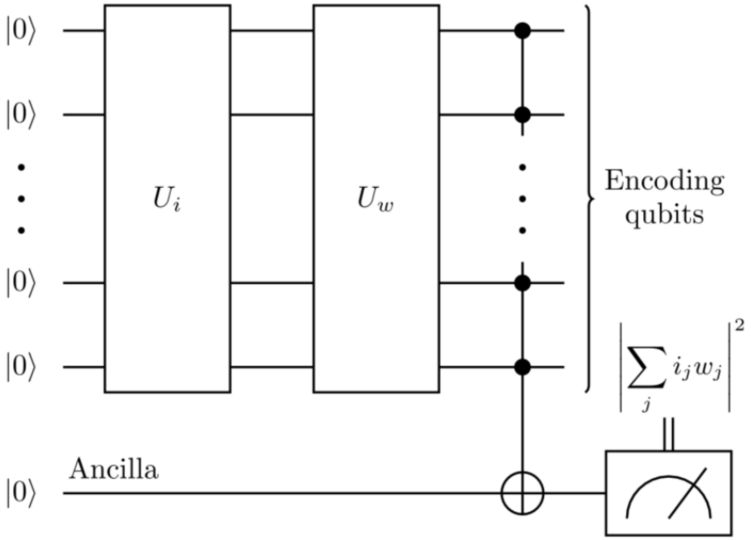
Clásico	**Cuántico**
Neurona $(x_1, \cdots, x_n)$	Qubit $\lvert\psi\rangle$
Combinación lineal y Función de activación	Operaciones unitarias y Mediciones
Entrenar $\mathcal{E}$ y $\mathcal{D}$	Entrenar U
Error Cuadratico	Fidelidad



Otras propuestas para neuronas cuánticas

Otras propuestas para neuronas cuánticas

Circuitos multiqubit[1,2]

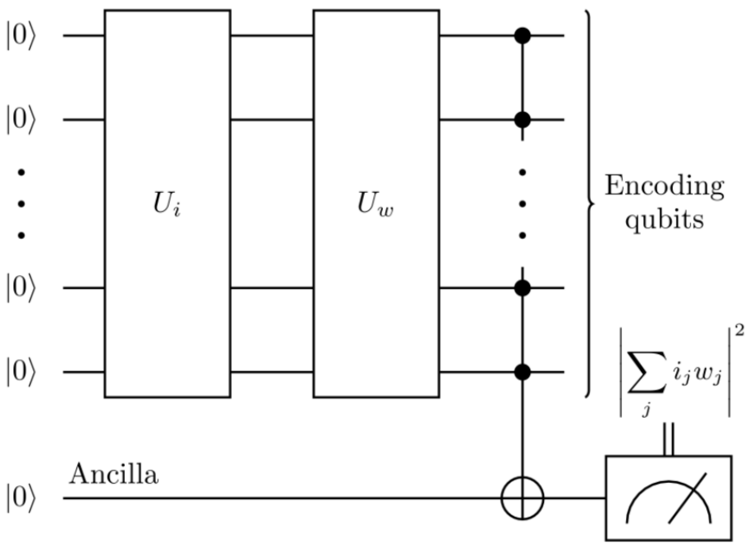


[1] A. Kapoor, N. Wiebe, and K. Svore,
Adv Neural Inf Process Syst, 29 (2016).

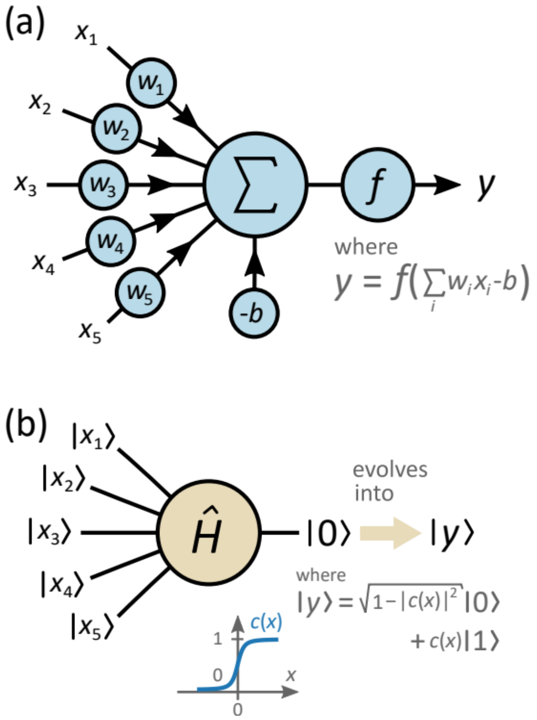
[2] A. Hathidara and L. Pandey,
arXiv:2412.02083 (2024)

Otras propuestas para neuronas cuánticas

Circuitos multiqubit[1,2]



Control cuántico[3,4]



[1] A. Kapoor, N. Wiebe, and K. Svore, Adv Neural Inf Process Syst, 29 (2016).

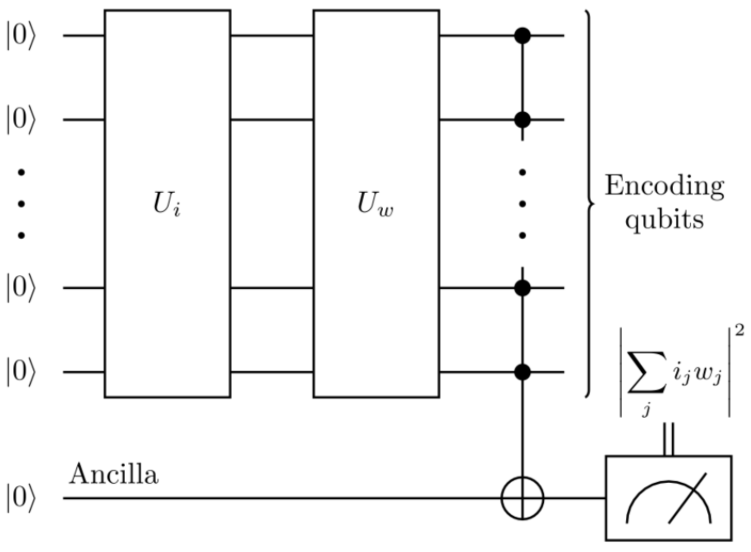
[2] A. Hathidara and L. Pandey, arXiv:2412.02083 (2024)

[3] E. Torrontegui and J. J. García-Ripoll, EPL 125 30004 (2019)

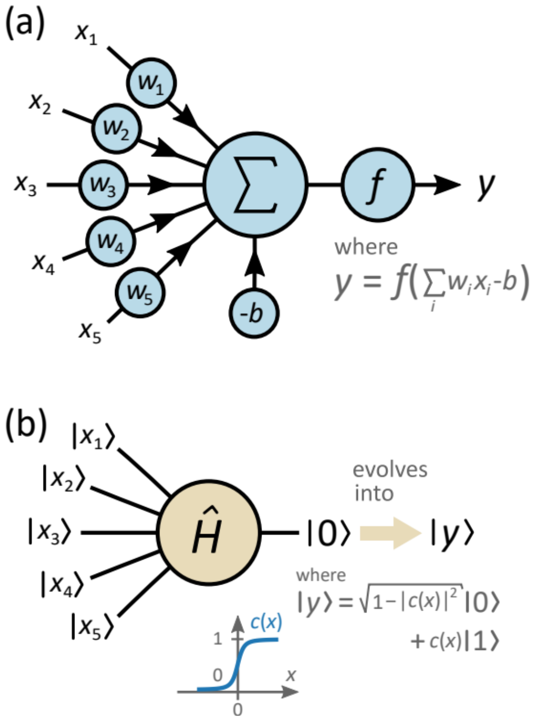
[4] M. Pechal F. Roy, S. A. Wilkinson, et.al., Phys. Rev. Research 4, 033190 (2022)

Otras propuestas para redes neuronales cuánticas

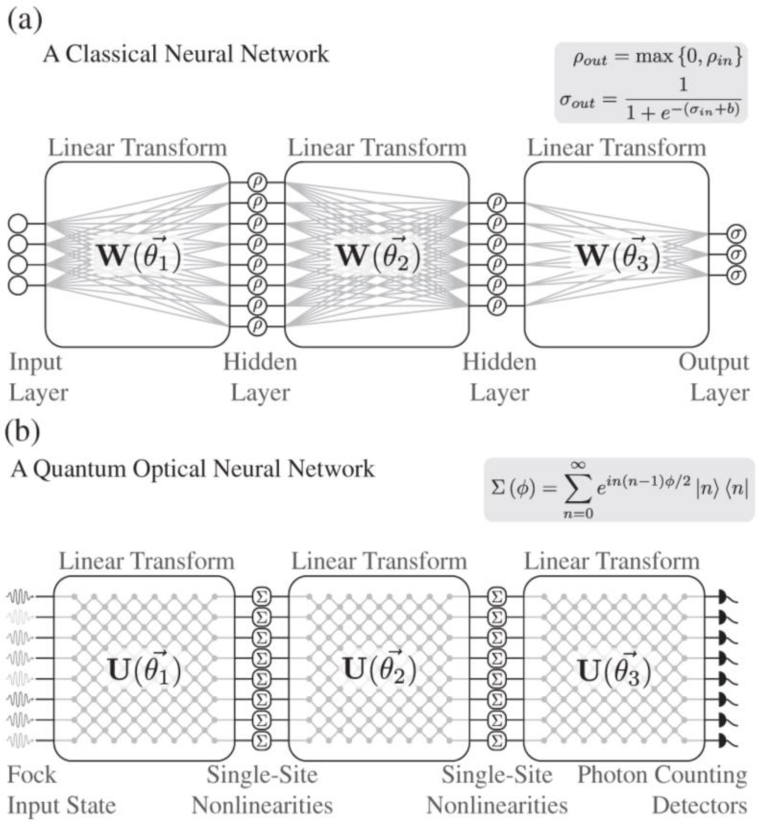
Circuitos multiqubit[1,2]



Control cuántico[3,4]



Redes neuronales ópticas[5]



[1] A. Kapoor, N. Wiebe, and K. Svore, Adv Neural Inf Process Syst, 29 (2016).

[2] A. Hathidara and L. Pandey, arXiv:2412.02083 (2024)

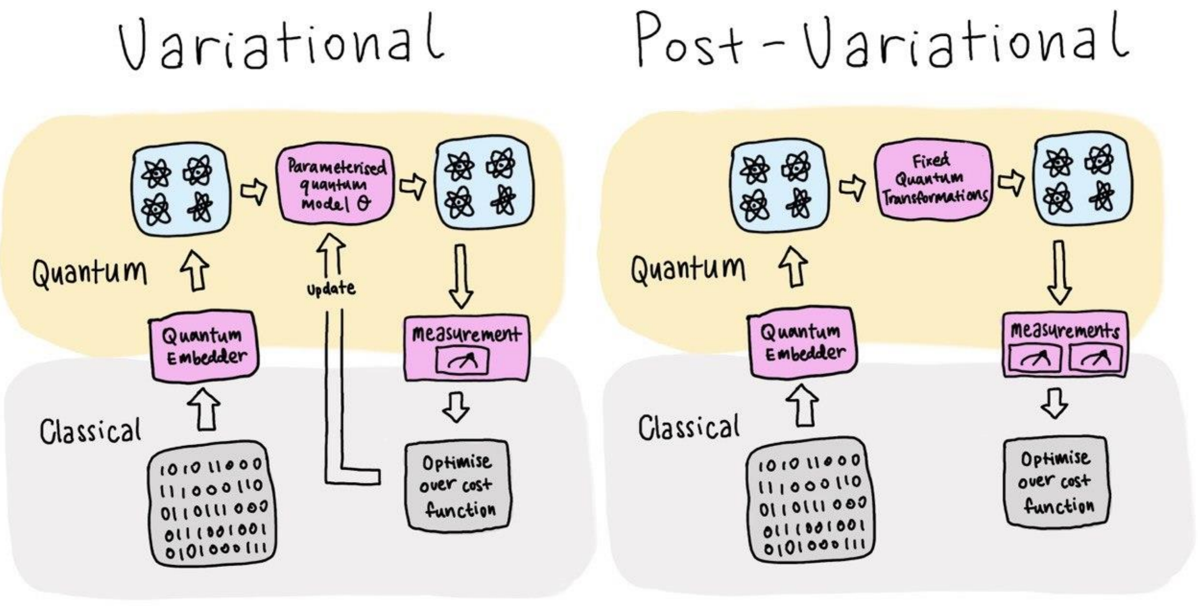
[3] E. Torrontegui and J. J. García-Ripoll, EPL 125 30004 (2019)

[4] M. Pechal F. Roy, S. A. Wilkinson, et.al., Phys. Rev. Research 4, 033190 (2022)

[5] G.R. Steinbrecher, J.P. Olson, D. Englund, et al., npj Quantum Inf 5, 60 (2019).

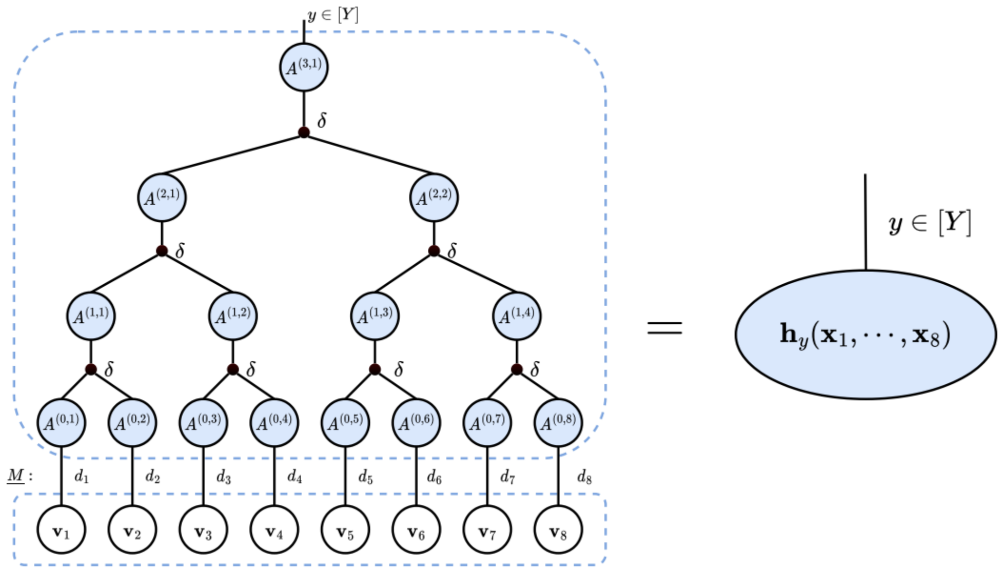
Otras propuestas para redes neuronales cuánticas

Post-Variacionales



[Post-variational quantum neural networks | PennyLane Demos]

Inspiración cuántica



M. Wang, Y. Pan, Z. Xu, X. Yang, G. Li, A. Cichocki, arXiv:2302.09019 (2023)

Software para redes neuronales cuánticas

Software para redes neuronales cuánticas

Qiskit Machine Learning

Qiskit | Ecosystem

Qiskit Machine Learning 0.8.2

Quantum neural networks (`qiskit_machine_learning.neural_networks`)

A neural network is a parametrized network which may be defined as a artificial neural network - classical neural network - or as parametrized quantum circuits - quantum neural network. Furthermore, neural networks can be defined with respect to a discriminative or generative task.

Neural networks may be used, for example, with the `VQC` algorithm.

See also the `TorchConnector` that allows the use of these neural networks in code written to PyTorch.

Neural Network Base Classes

`NeuralNetwork`	Abstract Neural Network class providing forward and backward pass and handling batched inputs.

Neural networks

`EstimatorQNN`	A neural network implementation based on the Estimator primitive.
`SamplerQNN`	A neural network implementation based on the Sampler primitive.

Software para redes neuronales cuánticas

Qiskit Machine Learning

Qiskit | Ecosystem

Qiskit Machine Learning 0.8.2

Quantum neural networks (qiskit_machine_learning.neural_networks)

A neural network is a parametrized network which may be defined as a artificial neural network - classical neural network - or as parametrized quantum circuits - quantum neural network. Furthermore, neural networks can be defined with respect to a discriminative or generative task.

Neural networks may be used, for example, with the `VQC` algorithm.

See also the `TorchConnector` that allows the use of these neural networks in code written to PyTorch.

Neural Network Base Classes

`NeuralNetwork`	Abstract Neural Network class providing forward and backward pass and handling batched inputs.

Neural networks

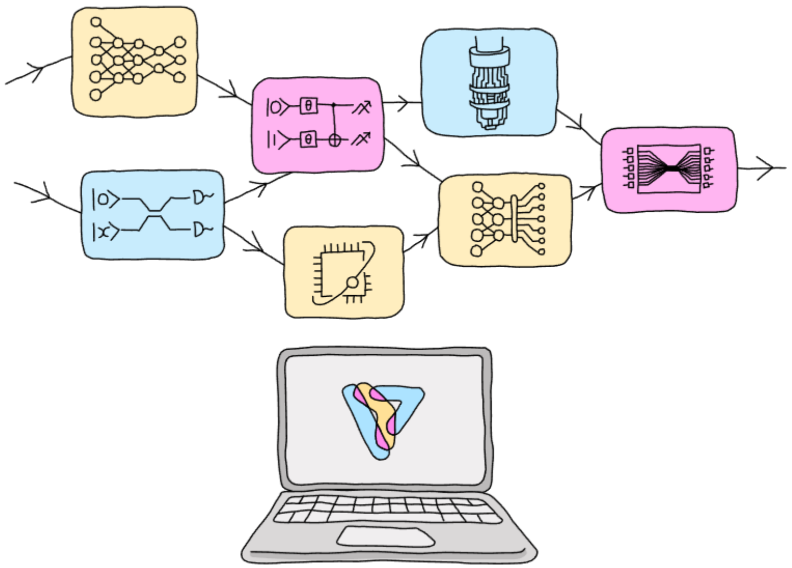
`EstimatorQNN`	A neural network implementation based on the Estimator primitive.
`SamplerQNN`	A neural network implementation based on the Sampler primitive.

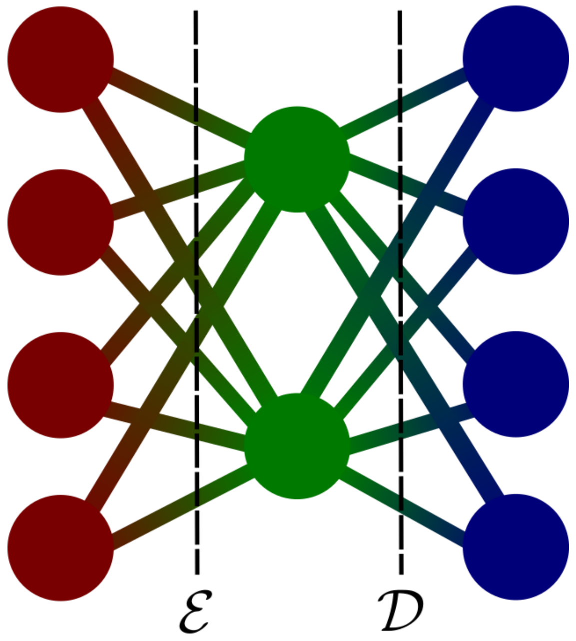
Pennylane

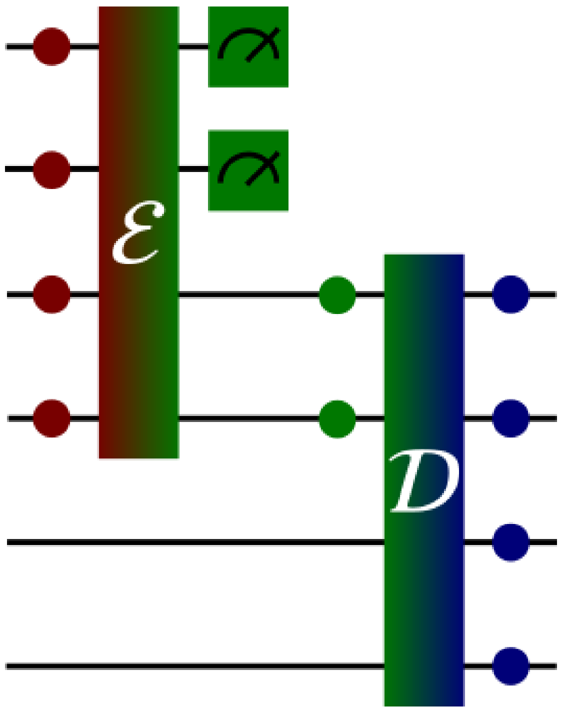
PENNYLANE

PennyLane is an **open-source software framework built around the concept of quantum differentiable programming**. It seamlessly integrates classical machine learning libraries with quantum simulators and hardware, giving users the power to train quantum circuits.

To find out more, visit the PennyLane Documentation, or check out the gallery of hands-on quantum machine learning demonstrations.







ICFO

@Luplaciano

Luciano Iván
Pereira Valenzuela

Luciano.pereira@icfo.eu

Luplaciano#4389

LucianoPereiraValenzuela

Luciano Pereira Valenzuela