

Quantum Simplicial Neural Networks

Simone Piperno, Claudio Battiloro, Andrea Ceschini, Francesca Dominici, Paolo Di Lorenzo, Massimo Panella

Eduardo Velázquez

12 de marzo de 2025

Resumen

En este trabajo se presenta el primer modelo Topológico de Aprendizaje Profundo *cuántico*, que consiste en *Redes Neuronales Cuánticas*¹ operando en complejos simpliciales.

¹Modelo de aprendizaje que combina una red neuronal con cómputo cuántico, es decir, parte de un *estado* de entrada y lo transforma en un *estado* de salida mediante compuertas y circuitos cuánticos.[5]

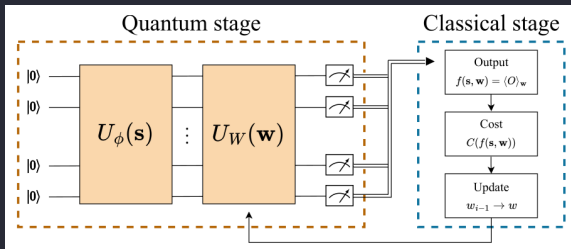
- ▶ Se espera que propiedades como la *superposición* de estados y el *entrelazamiento* conduzcan a un entrenamiento más rápido, un incremento en el poder de cómputo y eficiencia respecto a las redes tradicionales.
- ▶ Debido al desarrollo actual de las computadoras cuánticas, gran parte del trabajo se desarrolla en *circuitos cuánticos variacionales* (VQC), es decir, en circuitos cuánticos con parámetros ajustables que se ajustan en computadoras clásicas²

²Existen marcos de trabajo que permiten simular estos circuitos (Qiskit, Rigetti SDK, PennyLane).

2.2 *Variational Quantum Circuits (VQC)*

Son algoritmos cuánticos con parámetros ajustables que constan de tres partes [4]:

- ▶ Encaje. (Traducir el input a un estado de entrada)
- ▶ Parametrización del circuito. (Compuertas cuánticas con parámetros ajustables)
- ▶ Medición. (Obtención del valor esperado de observables)



Piperno, et .al. *Quantum Simplicial Neural Networks*

$$\{0, 1, 0, \dots\} \xrightarrow{\phi} |\psi^{IN}\rangle$$

U_W Compuertas cuánticas (operadores unitarios)

U_ϕ Mapeo de propiedades

$|\psi^{IN}\rangle > n$ -qubit

2.3 Quantum Graph Neural Networks (QGNN)

Son un tipo de Redes Neuronales Cuánticas diseñadas para trabajar con estructura de gráficas, entre sus ventajas permiten abordar problemas relacionados con gráficas a gran escala e isomorfismos de gráficas [2],[3].

Dada una gráfica G , para este tipo de redes, se busca asociar un estado $|G\rangle$ y a partir de ciertos axiomas y propiedades [2] se codifican QNNs.

Ejemplo A1. Si las gráficas $G = (V, E)$ y $G' = (V, E')$ difieren en una arista xy , donde $x, y \in V$. Entonces, $|G\rangle = U(xy) |G'\rangle$ para algún operador U que depende sólo de xy

(ver [1, 2])

Redes Neuronales Cuánticas (QNNs)

1. Dado un vector de datos $\mathbf{s} \in \mathcal{R}^n$ y un estado inicial $|0\rangle^{\otimes N}$ de N -qubits, se busca un operador unitario U_ϕ que codifique \mathbf{s} a un VQC de N -qubits.

$$U_\phi = |0\rangle^{\otimes N} \mapsto |\psi\rangle$$

2. A partir de compuertas cuánticas representadas por operadores unitarios y parámetros, las capas de la QNN transforman a $|\psi\rangle$:

$$U_W = U_{w_1} \cdots U_{w_k} |\psi\rangle = |\psi'\rangle$$

3. Sea \hat{O} el operador que representa la observable que queremos medir, por ser un sistema cuántico necesitamos calcular el valor esperado $f := \langle \hat{O} \rangle$.
4. Los parámetros de la red se actualizan optimizando f con algoritmos clásicos.

Redes Neuronales Simpliciales Cuánticas (QSNN)

Sea χ_2 un complejo simplicial de orden 2 compuesto por N vértices, E aristas y T triángulos, así como una señal de entrada $\mathbf{s}_{chi} \in \mathbb{R}^{N+E+T}$, es decir, $\mathbf{s}_{chi} = [s_0, s_1, s_2]$ tal que

$$s_0 : V \rightarrow \mathbb{R}$$

$$s_1 : E \rightarrow \mathbb{R}$$

$$s_2 : T \rightarrow \mathbb{R}.$$

1. Inicializamos un conjunto de $(N + E + T)$ -qubits,

$$|\psi\rangle^{IN} = |0\rangle^{\otimes N+E+T}$$

2. Codificamos \mathbf{s} como $|\psi\rangle^{ENC} = U_\phi |\psi\rangle^{IN}$

U_ϕ es la composición de compuertas cuánticas, en [3] se proponen *compuertas de rotación* R_k

$$\left(\bigotimes_{i=1}^N R_{k_1}(\pi s_i) \right) \otimes \left(\bigotimes_{i=N+1}^{N+E} R_{k_2}(\pi s_i) \right) \otimes \left(\bigotimes_{i=N+E+1}^{N+E+T} R_{k_3}(\pi s_i) \right)$$

donde πs_i es el ángulo proporcional al input s_i y está entre en $-\pi$ y π

- 3 *Learnable Embedding*. Se actualiza el estado global mediante compuertas de rotación en un ángulo aprendido $W_{i,i}$:

$$|\Psi\rangle^{\text{emb}} = \left(\bigotimes_{i=1}^N R_{k_4}(\mathbf{W}_{i,i}) \right) \otimes \left(\bigotimes_{i=N+1}^{N+E} R_{k_5}(\mathbf{W}_{i,i}) \right) \otimes \left(\bigotimes_{i=N+E+1}^{N+E+T} R_{k_6}(\mathbf{W}_{i,i}) \right) |\Psi\rangle^{\text{enc}}$$

- 4 *Learnable Interactions*.. Similar al paso anterior, pero ahora se aprende un ángulo que favorezca la interacción (entrelazamiento) $W_{i,i}$ entre qubits de cualquier tipo (V, E, T) , resultando en un nuevo estado $|\psi\rangle^{LI}$ a partir de $|\psi\rangle^{\text{emb}}$

$$LI_{k,p}(\mathbf{W}_{i,j}) = CX \cdot (\mathbb{I} \otimes R_{k,p}(\mathbf{W}_{i,j})) \cdot CX$$

donde CX es un tipo de compuerta cuántica usada para representar si hay o no entrelazamiento.

$$\begin{aligned}
|\Psi\rangle^{\text{LI}} = & \prod_{\substack{1 \leq i < j \leq T \\ [\mathbf{L}_2^{(d)}]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(N+E+i, N+E+j)} \right) \prod_{\substack{1 \leq i \leq E \\ 1 \leq j \leq T \\ [\mathbf{B}_2]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(N+i, N+E+j)} \right) \\
& \prod_{\substack{1 \leq i < j \leq E \\ [\mathbf{L}_1^{(u)}]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(N+i, N+j)} \right) \prod_{\substack{1 \leq i < j \leq E \\ [\mathbf{L}_1^{(d)}]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(N+i, N+j)} \right) \\
& \prod_{\substack{1 \leq i \leq N \\ 1 \leq j \leq E \\ [\mathbf{B}_1]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(i, N+j)} \right) \prod_{\substack{1 \leq i < j \leq N \\ [\mathbf{L}_0^{(u)}]_{i,j} \neq 0}} \left(\bigotimes_{n=1}^{N+E+T} o_n^{(i,j)} \right) |\Psi\rangle^{\text{emb}},
\end{aligned}$$

- 5 Se calcula el valor esperado a partir de los valores esperados de cada qubit, obteniendo un vector.

$$\hat{y}_i = \langle \Psi^{\text{LI}} | \sigma_z^{(i)} | \Psi^{\text{LI}} \rangle, \quad i = 1, \dots, N + E + T,$$

$$\hat{\mathbf{y}} = \left(\langle \Psi^{\text{LI}} | \sigma_z^{(1)} | \Psi^{\text{LI}} \rangle, \langle \Psi^{\text{LI}} | \sigma_z^{(2)} | \Psi^{\text{LI}} \rangle, \dots, \langle \Psi^{\text{LI}} | \sigma_z^{(N+E+T)} | \Psi^{\text{LI}} \rangle \right),$$

- 6 Este vector se optimiza con técnicas clásicas.

Base Quantum Simplicial Neural Network

Inspirado en el modelo de Ising, donde se modelan interacciones a diferente orden entre muchas partículas. En este caso, se hace una analogía entre qubits y las partículas en el modelo.

Debido a que los qubits codifican vértices, aristas y triángulos, las “interacciones” entre qubits se pueden representar mediante correlaciones vértice-vértice, arista-arista, triángulo-triángulos y sus combinaciones: vértice-arista, vértice-triángulo, etc.

En este modelo, las correlaciones que se propone buscar consisten en compuertas cuánticas de rotación en ejes específicos:

Encoding: R_x

Embedding: R_z

Interactions: R_{zz} .

Schematic Quantum Simplicial Network

De forma análoga, se proponen otras correlaciones en ejes específicos:

	BQSNN	SQSN
Encoding	R_x	R_x, R_y, R_z
Embedding	R_z	R_x, R_y, R_z
Interactions	R_{zz}	$R_{i,j}$

Resultados Numéricos

Detección de componentes solenoidales

Se generó un complejo simplicial aleatorio de orden 2 con un número mínimo de símlices igual a 12 y un número máximo igual a 16, de la siguiente manera: Se generaron los nodos considerando un promedio de 10, posteriormente se generaron las aristas con una probabilidad de 0.3 y finalmente se rellenaron los triángulos con una probabilidad de 0.8. Como señal sobre el complejo se

generaron dos vectores aleatoriamente y una variable aleatoria que seguía una distribución de Bernoulli con probabilidad de éxito fija ($p=0.5$). La base de señales se separó en un conjunto de

entrenamiento y en uno de validación, y la tarea de clasificación consistió en determinar si la variable aleatoria en la distribución de Bernoulli asociada había resultado 0 ó 1.

Table 3: Mean test accuracy per layers for the architectures applied to Task 1.

Architecture	Number of layers				
	1	2	3	4	5
SQSN	0.949	0.956	0.957	0.950	0.951
BQSN	0.833	0.805	0.874	0.868	0.891
MLP	0.991	0.903	0.897	0.932	0.939
GSCN	0.939	0.872	0.798	0.736	0.598
GSAN	0.929	0.886	0.885	0.877	0.879

Resultados Numéricos

Source Localization Problem

Se generaron gráficas aleatorias siguiendo el modelo de bloque estocástico. Cada gráfica consistía 6 nodos distribuidos uniformemente entre dos comunidades. Las aristas se generaron aleatoriamente fijando una probabilidad para aristas entre ambas comunidades y otra internamente entre cada comunidad.




Adicionalmente se generó ruido en la señales para algunas aristas elegidas aleatoriamente. A cada gráfica se asignó el complejo simplicial que resultó de rellenar todos los triángulos.

La tarea es identificar a la comunidad responsable de los picos.

Table 5: Mean test accuracy per layers for the architectures applied to Task 2.

Architecture	Number of layers				
	1	2	3	4	5
SQSN	0.674	0.665	0.686	0.650	0.681
BQSN	0.681	0.618	0.702	0.678	0.701
QGNN	0.608	0.578	0.659	0.593	0.665
MLP	0.394	0.531	0.630	0.583	0.606
GSCN	0.402	0.482	0.455	0.418	0.396
GSAN	0.535	0.602	0.579	0.587	0.513

Referencias I

-  G. Casati, D.L. Shepelyansky, P. Zoller, G. Benenti, and Società italiana di fisica.
Quantum Computers, Algorithms, and Chaos.
Proceedings int. school physics E. Fermi. IOS Press, 2006.
-  Radu Ionicioiu and Tim P. Spiller.
Encoding graphs into quantum states: An axiomatic approach.
Phys. Rev. A, 85:062313, Jun 2012.
-  Simone Piperno, Claudio Battiloro, Andrea Ceschini, Francesca Dominici, Paolo Di Lorenzo, and Massimo Panella.
Quantum simplicial neural networks, 2025.

Referencias II



Mayra Alejandra Rivera-Ruiz, Andres Mendez-Vazquez, and José Mauricio López-Romero.

Time series forecasting with quantum machine learning architectures.

In Obdulia Pichardo Lagunas, Juan Martínez-Miranda, and Bella Martínez Seis, editors, *Advances in Computational Intelligence*, pages 66–82, Cham, 2022. Springer Nature Switzerland.



N.B. Singh.

Fundamentals: Schrödinger's Equation to Deep Learning.

N.B. Singh.