

# UNIVERSIDAD MAYOR, REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA

## Facultad de Ciencias de la Tecnología



## Redes Neuronales Cuánticas

**ESTUDIANTE:** Polo Orellana Brayan Simón

**CARRERA:** Ingeniería en Ciencias de la computación

**MATERIA:** Inteligencia Artificial 2

**SIGLA:** SIS-421

## **QNNs**

Introducción.....	3
¿Dónde se usan las redes Neuronales Cuánticas? .....	3
¿Qué son los Qubit? .....	3
¿Qué es un bit? .....	3
¿Qué son los Qubits? .....	4
Estructura de las Redes Neuronales Cuánticas .....	5
Modelo de Neurona Qubit.....	6
Beneficios de usar redes neuronales multicapa .....	8
Ejercicio explicativo .....	9
Bibliografías .....	9

## Introducción

Una Red **Neuronal Cuántica** es un modelo computacional sofisticado que amalgama principios de la mecánica cuántica con redes neuronales, esto con el objetivo de aprovechar las propiedades únicas de los sistemas cuánticos para mejorar capacidades computacionales.

Las QNN son parte del dominio más amplio del aprendizaje automático cuántico, que tiene la finalidad de explotar la computación cuántica para realizar tareas que son inviables o ineficientes para las computadoras clásicas.

## ¿Dónde se usan las redes Neuronales Cuánticas?

Las QNN son usadas en áreas como videojuegos, la aproximación de funciones y el procesamiento de Big data. Además, el algoritmo de Redes Neuronales cuánticas es útil en el modelado de redes neuronales cuánticas y es crucial, ya que influye directamente en el rendimiento y estabilidad de la red.

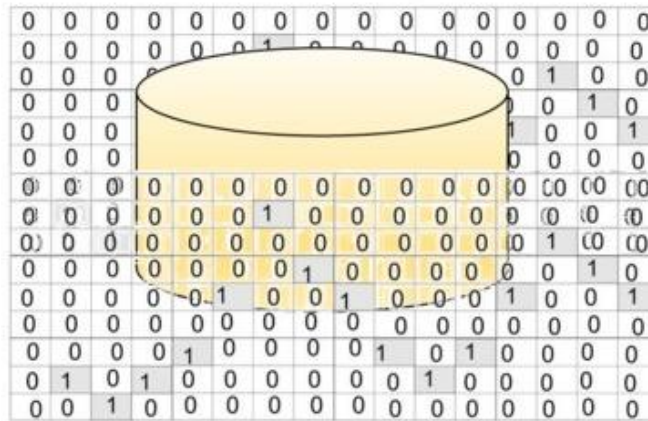
## ¿Qué son los Qubit?

La idea de las Redes Neuronales Cuánticas viene de reemplazar las neuronas binarias por **qubits** (también llamadas quron), esto genera el resultado en unidades neuronales que pueden estar en un estado de superposición (activa y en reposo a la vez).

Un qubit es la unidad básica de información utilizada para codificar datos de computación cuántica, dado que el término es atribuido al físico teórico estadounidense **Benjamín Schumacher**.

## ¿Qué es un bit?

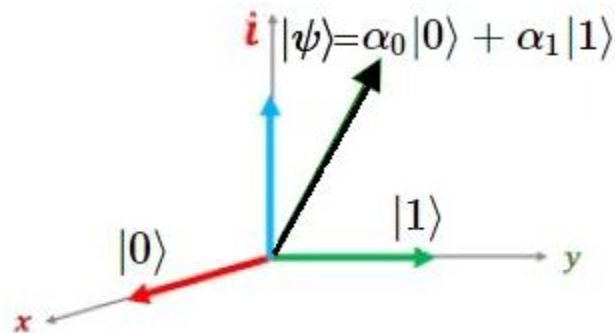
Un bit es físicamente observable y puede guardarse como tal en un dispositivo de almacenamiento digital, se trabaja con la dirección de magnetización u otras marcas binarias de esos núcleos. Estas asignaciones son apropiadas para construir un sistema de numeración binaria del almacenamiento porque permite utilizar dígitos binarios 0 y 1 para representar los datos, así llamado computación básica:



## ¿Qué son los Qubits?

Un qubit corresponde a un sistema físico que tiene dos **estados ortogonales**, el cual es denotado de acuerdo con la **Notación de Dirac o bra-ket** (usado para describir estados cuánticos) como:  $|1\rangle$  y  $|0\rangle$ .

Los estados del sistema, que denotamos como  $|\psi\rangle$  se denominan **kets**, es la **forma compacta de representar un estado cuántico** en un espacio vectorial.



Estos estados no vacíos se tratan como vectores de base. Es decir, se aplica una correspondencia hacia una estructura algebraica de Espacio Vectorial.

Su estructura está dotada con una operación interna **Adición**, y una operación externa **Producto por un escalar**, definida sobre un cuerpo **K** (como los números Reales o los números Complejos).

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

Un qubit puede encontrarse en una superposición de los estados  $|1\rangle$  y  $|0\rangle$ . Es decir, como una combinación lineal de los vectores de base, de la forma  $\alpha_0|\psi\rangle + \alpha_1|\psi\rangle$ .

Donde  $\alpha_i$  es denominado amplitud de probabilidad, los cuales son números complejos que se utilizan para describir el comportamiento del sistema.

La suma del cuadrado de cada uno de esos módulos:

$$|\alpha_0|^2 + |\alpha_1|^2 = 1$$

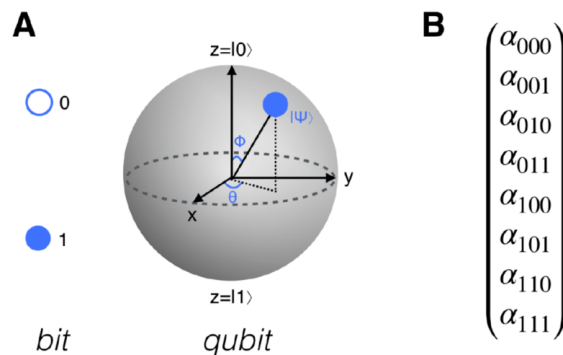
Representa una probabilidad o densidad de probabilidad. Es decir que estos coeficientes  $\alpha_i$  son de la forma:

$$z = x + iy$$

Donde:  $i = \sqrt{-1}$

Esta propiedad permite que los qubits codifiquen más información que los bits clásicos.

**Ejemplo:**



Representación de 3 qubits con 8 amplitudes de probabilidad, donde la probabilidad total debe ser 100%:

$$|\psi\rangle = \alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \dots + \alpha_7|111\rangle = 1$$

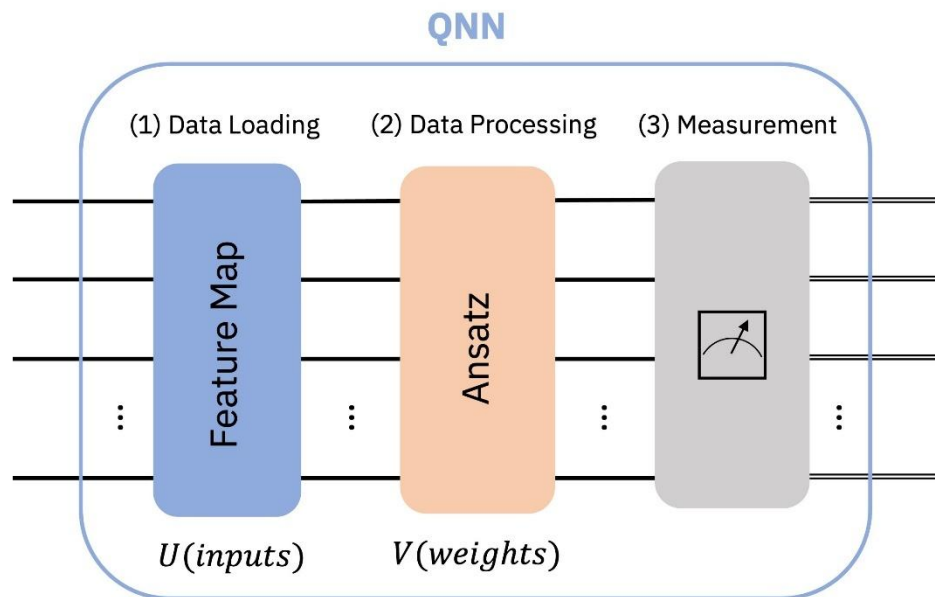
## Estructura de las Redes Neuronales Cuánticas

Desde una perspectiva de machine learning las QNNs son, modelos algorítmicos que se pueden entrenar para encontrar patrones ocultos de manera similar que sus contrapartes clásicas.

- Cargar datos clásicos (features) en un estado cuántico
- Procesarlos con compuertas cuánticas parametrizadas por pesos entrenables

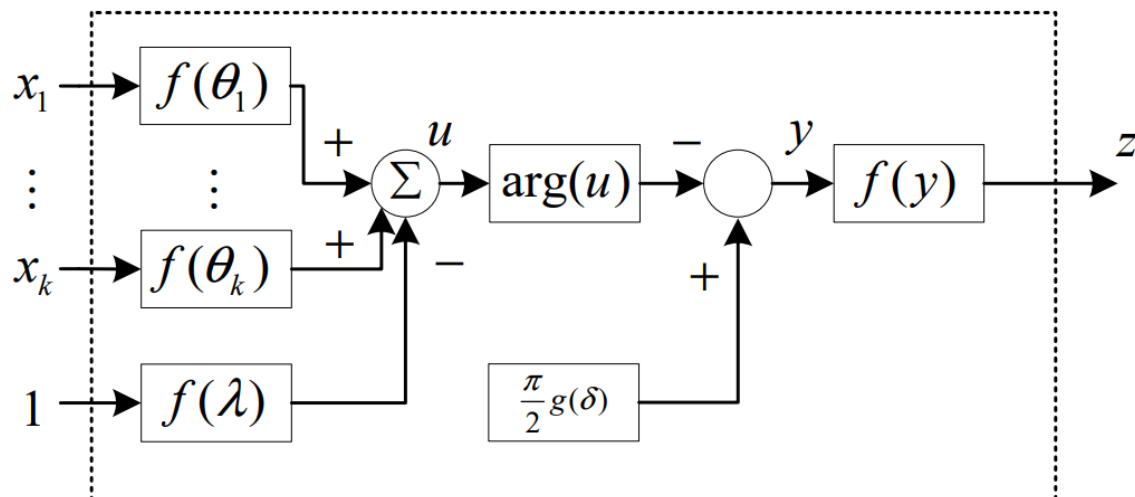
Desde la **perspectiva de la computación cuántica**, las QNNs son algoritmos cuánticos basados en circuitos cuánticos parametrizados que se pueden entrenar de forma variacional utilizando optimizadores clásicos.

Estos circuitos contienen un **mapa de características** (con parámetros de entrada) y un **ansatz** (con pesos entrenables):



### Modelo de Neurona Qubit

El modelo de neurona Qubit se representa de la siguiente manera:



Teniendo el estado de un qubit como:

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

Donde mencionado antes  $\alpha$  y  $\beta$  son números complejos llamados amplitud de probabilidades, que satisface la siguiente ecuación:

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

Reescribiendo el estado de la neurona como:

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

$f(\varphi)$  es otra forma de expresar el estado cuántico,  $\varphi$  es la fase que describe el estado cuántico.

La neurona Z representa el total de inputs de diferentes neuronas k de la siguiente manera:

$$1) \quad u = \sum_k^K f(\theta_k) x_k - f(\lambda) = \sum_k^K f(\theta_k) f(y_k) - f(\lambda)$$

$$2) \quad y = \frac{\pi}{2} g(\delta) - \arg(u)$$

$$3) \quad z = f(y)$$

**Para la ecuación 1):**

Aquí el modelo neuronal tiene los siguientes dos parámetros:

- El parámetro de fase de forma de conexiones ponderadas  $\theta_k$
- Los umbrales  $\lambda$

$f(\theta_k)$  representa los pesos correspondientes al estado de la neurona  $x_k$

$f(\varphi)$  convierte el valor del input  $x_k$  en un estado cuántico cuyo valor de fase es  $\theta_k$

$x_k$  es el estado de la neurona k-esima.

Después de la multiplicación de pesos  $f(\theta_k)$  con el input  $x_k$ , el estado de la neurona experimenta rotación según las puertas de rotación.

Esta ecuación 1) expresa el estado  $u$  de una neurona de manera habitual, es decir, como la suma ponderada de los estados de las entradas menos un umbral.

**Para la ecuación 2):**

$\arg(u)$  representa el argumento del número complejo  $u$ , y este es implementado por  $\arctan(\text{Im}(u)/\text{Re}(u))$ .

La función de activación  $g(\delta)$  es usada con el objetivo de obtener una representación inversa generalizada de una puerta lógica cuántica que funciona como una compuerta NOT en computación cuántica.

Además, puede compensar el efecto de pequeños cambios en los pesos en la salida de la red. Aquí,  $g$  se define como la función sigmoidea en los siguientes casos:

$$g(\delta) = \frac{1}{1 + e^{-\delta}}$$

**Para la ecuación 3):**

El estado de la neurona  $Z$  es el estado de salida del modelo, y se obtiene mediante un cálculo ponderado de  $k$  estados neuronales diferentes.

### **Beneficios de usar redes neuronales multicapa**

Basándose en los principios de la computación cuántica **Noriaki Kouda, Nobuyuki Matsui y Haruhiko Nishimura**, propusieron una red neuronal de cúbits multicapa con neuronas de cúbits.

Exploraron su rendimiento en dos escenarios de aplicación adicionales:

- compresión de imágenes.
- reconocimiento de patrones.

Las redes neuronales de cúbits presentan una potencia de cálculo y capacidades de procesamiento paralelo superiores a las redes neuronales tradicionales, lo que permite un manejo más eficaz de datos de alta dimensión y modelos complejos.

Además, las redes neuronales de cúbits ofrecen mayor protección de la privacidad y mayor seguridad, lo que las hace adecuadas para aplicaciones en comunicaciones seguras y cifrado de datos.



## Ejercicio explicativo

Para este ejercicio se realizará un simulador con **PyTorch + GPU** logrando hacer **simulación cuántica**, lo cual es útil para aprender, investigar y probar.

Enlace al repositorio explicativo sobre el ejercicio:

<https://github.com/bspoloo/SIS421-022025>

## Bibliografías

- <https://www.mdpi.com/2079-9292/13/8/1514>
- <https://www.ibm.com/es-es/topics/qubit>
- [https://docirs.cl/bit\\_qubit\\_superposicion\\_simuladores\\_reales\\_y\\_complejos.asp](https://docirs.cl/bit_qubit_superposicion_simuladores_reales_y_complejos.asp)
- <https://link.springer.com/article/10.1023/A:1019708909383>
- [https://qiskit-community.github.io/qiskit-machine-learning/locale/es\\_UN/tutorials/01\\_neural\\_networks.html](https://qiskit-community.github.io/qiskit-machine-learning/locale/es_UN/tutorials/01_neural_networks.html)
- <https://es.eitca.org/artificial-intelligence/eitc-ai-tfgml-tensorflow-quantum-machine-learning/overview-of-tensorflow-quantum/layer-wise-learning-for-quantum-neural-networks/examination-review-layer-wise-learning-for-quantum-neural-networks/what-is-a-quantum-neural-network-qnn-and-how-does-it-process-data-using-qubits/>