



INSTITUTO POLITÉCNICO  
NACIONAL



CENTRO DE INVESTIGACIÓN EN  
COMPUTACIÓN

TESIS

Aprendizaje de máquina cuántico  
para la clasificación de imágenes

PARA OBTENER EL GRADO DE:  
**MAESTRÍA EN CIENCIAS DE LA  
COMPUTACIÓN**

PRESENTA:

**Lic. Héctor Iván García Hernández**

DIRECTORES DE TESIS:

Dra. GuoHua Sun

Dr. Shi-Hai Dong

## **Resumen**

La Computación Cuántica y en especial el Aprendizaje de Máquina Cuántica, en un corto periodo de tiempo ha ganado mucho interés entre los diversos grupos de investigación a través del mundo. Lo cual se pone en evidencia por el considerable número de modelos propuestos que aplican en mayor o menor medida las propiedades de la mecánica cuántica. A pesar del incremento en el número de modelos propuestos, hay un déficit en la prueba de ellos, siendo esta no existente o simplemente una aplicación sobre bancos de datos sintéticos.

El presente trabajo contribuye a la investigación al validar y probar un modelo cuántico de clasificación binaria bajo distintos escenarios, tanto en clases balanceadas como altamente desbalanceadas. Este último con una posible aplicación directa sobre bancos de datos médicos. Más aún, se propone, prueba y valida una extensión del modelo y método de entrenamiento para procesar y clasificar imágenes de múltiples canales el cual puede ser explotado para un procesamiento, entrenamiento y clasificación por lotes. Los modelos propuestos son robustos ante el desbalance de clases y pueden converger en tan poco como en una sola época de entrenamiento.

## **Abstract**

Quantum Computing and especially Quantum Machine Learning, in a short time, has gained much interest among the various research groups throughout the world. This is evident by the considerable number of proposed models that apply, to a greater or lesser extent, the properties of quantum mechanics. Despite the increase in the number of proposed models, there is a deficit in their verification, being non-existent or an application on synthetic data sets.

This work contributes to research by validating and testing a quantum model for binary classification under different scenarios, both in balanced and highly unbalanced classes. The latter with a possible direct application on medical datasets. Moreover, an extension of the model, as well as the training procedure is proposed, tested, and validated. The model aims to process and classify multi-channel images and could exploit processing, training, and classification in batches. The proposed models are robust against class imbalance and can converge in as little as one epoch of training.

## **Agradecimientos**

*A mis padres, por su dedicación y compromiso.*

*A mi hermana, por sus invaluable consejos.*

*A mi familia, por siempre estar conmigo.*

*A Agar, por su apoyo, comprensión y guía. Sin ella, esto no sería posible.*

*Y al CONACyT, por su invaluable apoyo y por permitirme emprender este increíble viaje.*

# Índice general

<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de cuadros</b>	<b>VIII</b>
<b>Nomenclatura</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Justificación . . . . .	3
1.3. Objetivos . . . . .	3
1.3.1. Objetivo General . . . . .	3
1.3.2. Objetivos Específicos . . . . .	3
1.4. Estructura . . . . .	4
<b>2. Estado del arte</b>	<b>5</b>
<b>3. Materiales y Métodos</b>	<b>8</b>
3.1. Reconocimiento de Patrones . . . . .	8
3.1.1. Redes Neuronales . . . . .	9
3.2. Cómputo Cuántico . . . . .	9
3.2.1. Qubit . . . . .	10
3.2.2. Esfera de Bloch . . . . .	10
3.2.3. Producto Tensorial o Producto de Kronecker . . . . .	11
3.2.4. Fenómenos . . . . .	13
3.2.5. Circuitos Cuánticos . . . . .	14
3.3. Bancos de datos . . . . .	16

---

<b>4. Propuesta de Investigación</b>	<b>19</b>
4.1. Perceptrón Cuántico . . . . .	19
4.1.1. Los operadores unitarios $U_w$ y $U_i$ . . . . .	21
4.1.2. Método de entrenamiento . . . . .	24
4.2. Generalización a Tres Canales . . . . .	28
<b>5. Resultados y discusión</b>	<b>32</b>
5.1. Métodos de validación . . . . .	32
5.2. Medidas de desempeño . . . . .	33
5.3. Resultados . . . . .	34
<b>6. Conclusiones y trabajo futuro</b>	<b>48</b>
<b>Bibliografía</b>	<b>50</b>

# Índice de figuras

3.1. Ángulos de Euler . . . . .	11
3.2. Esfera de Bloch . . . . .	11
3.3. Circuito de Entrelazamiento . . . . .	16
3.4. Instancia del banco de datos Digits . . . . .	17
3.5. Instancia del banco de datos Semeion . . . . .	17
3.6. Instancia del banco de datos Fruits . . . . .	18
4.1. Diagrama de flujo para HSGS . . . . .	22
4.2. Generación de $U_w y U_i$ con barreras . . . . .	23
4.3. Generación de $U_w y U_i$ sin barreras . . . . .	24
4.4. Distribución de <i>readouts</i> en relación al umbral . . . . .	25
4.5. Flujo de Entrenamiento . . . . .	26
4.6. Flujo de Validación . . . . .	27
4.7. Flujo de Entrenamiento Multicanal . . . . .	29
4.8. Flujo de Validación Multicanal . . . . .	30
4.9. Circuito Cuántico Extendido . . . . .	31
5.1. Instancia Original, Binarizada y Vector de Pesos . . . . .	35
5.2. Conjunto Completo de Vectores de Pesos para Digits . . . . .	37
5.3. Conjunto Completo de Vectores de Pesos para Semeion . . . . .	38
5.4. Vector de Pesos para Imagen a Color . . . . .	43
5.5. Accuracy para Imagen a Color . . . . .	45
5.6. Medidas de desempeño para Imagen a Color . . . . .	46

# Índice de cuadros

3.1. Resumen de los bancos de datos. . . . .	18
5.1. Matriz de confusión para un banco de datos con dos clases. . . . .	33
5.2. Resubstitution Error en clases balanceadas. . . . .	39
5.3. Resubstitution Error en clases altamente desbalanceadas. . . . .	40
5.4. Accuracy en clases balanceadas. . . . .	40
5.5. Accuracy en clases altamente desbalanceadas. . . . .	41
5.6. Resubstitution Error en clases balanceadas. . . . .	42
5.7. Resubstitution Error en clases altamente desbalanceadas. . . . .	42
5.8. Promedio y desviación estándar de las medidas de desempeño obtenidas después de la validación por el método <i>RkFCV</i> con 5 repeticiones y 5 folds cada repetición.	44



# Nomenclatura

## Acronimos y Abreviaturas

*ANN* Artificial Neural Networks

*AUC* Area Under the Curve

*CNN* Convolutional Neural Networks

*HSGS* Hypergraph States Generation Subroutine

*IR* Imbalance Ratio

*KNN* K-Nearest Neighbors

*ML* Machine Learning

*MLP* Multi Layer Perceptron

*PCA* Principal Component Analysis

*PPV* Positive Predict Value

*QML* Quantum Machine Learning

*RBM* Restricted Boltzmann Machine

*TPR* True Positive Rate

# 1 Capítulo 1

## 2 Introducción

3 El aprendizaje de máquina (por sus siglas en inglés Machine Learning ML) ha demos-  
4 trado ser de gran utilidad en diversas tareas y en especial en la clasificación supervisada de  
5 patrones, pudiendo alcanzar desempeños comparables con el del humano y en ciertos casos  
6 incluso superarlo [1, 2]. Una de estas tareas es la clasificación de imágenes y para esto se han  
7 empleado diversos modelos computacionales, como: clasificador lineal [3], decision streams [4],  
8 los K vecinos más cercanos [1], perceptrón multicapa [3, 5] hasta las redes convolucionales [3, 5]

9  
10 Las aplicaciones de la clasificación de imágenes son diversas, por ejemplo la identificación  
11 de defectos en chapas de madera [6], predicción de producción de energía eólica y solar [7],  
12 procesamiento de imágenes médicas [8], procesamiento de imágenes radiológicas [9] e incluso  
13 como auxiliar en tratamientos médicos personalizados [10]; desde el surgimiento de las redes  
14 neuronales convolucionales (CNN por sus siglas en inglés) se han convertido en el modelo  
15 preferido al resolver problemas de este tipo por su desempeño sobresaliente [2].

16  
17 Los esfuerzos de investigación se han dividido principalmente en generar nuevos modelos  
18 computacionales y en mejorar los ya existentes, con la tendencia empírica de que un mode-  
19 lo con mejor desempeño implica casi siempre un modelo computacionalmente más complejo [3].

20  
21 Con el surgimiento de la computación cuántica y el avance reciente en hardware por diversas  
22 compañías, se han propuesto modelos cuánticos que intentan simular a su contraparte clásica,  
23 con el objetivo de explotar la gran potencia que muestran la superposición y el paralelismo  
24 cuántico.

25

En un esfuerzo para contribuir en el avance del aprendizaje de máquina cuántico (por sus siglas en inglés Quantum Machine Learning QML) en este trabajo se han implementado y propuesto algunos modelos con la intención, no de competir con los modelos del estado del arte, sino de ser un peldaño para que el QML tenga mayor exposición y sea tomado en cuenta para un desarrollo futuro.

## 1.1. Antecedentes

Una de las primeras tareas a resolver en la visión por computadora fue la del reconocimiento y clasificación de objetos y símbolos escritos a mano [11]. El Modified National Institute of Standards and Technology (MNIST) es uno de los primeros bancos de datos en esta área y se ha convertido en uno de los más famosos y además ha ganado un estatus como punto de referencia cuando se desea medir el desempeño de un nuevo modelo propuesto [3]. Por ejemplo, históricamente se ha buscado clasificarlo con los siguientes modelos:

LeCun et al. [3] utilizan como punto de partida un clasificador lineal con el cual alcanzan un error del 7,6 %.

Ignatov et al. [4] logran un error del 2,66 % utilizando Decision Streams, el cual es un modelo similar a los árboles de decisión en el que las ramas de decisión son separadas vagamente y pueden volver a unirse.

Keysers et al. [1] alcanzan un error del 0,52 % empleando un modelo de deformaciones no lineales precediendo un clasificador de los K vecinos más cercanos (por sus siglas en inglés K Nearest Neighbors KNN).

LeCun et al. [3] mediante un Análisis de Componentes Principales (Principal Componente Analysis, PCA) como preprocesamiento extraen 40 componentes que se emplean como el patrón de entrada en un clasificador cuadrático. De esta forma alcanzan un error de 3,3 %.

LeCun et al. [3] con un modelo Perceptrón multicapa (por sus siglas en inglés Multi Layer Perceptron MLP) con una sola capa oculta con 300 unidades ocultas alcanza un error de 4,7 % y 4,5 % con 1000 unidades ocultas.

Simard et al. [5] con un MLP de 800 unidades ocultas se logra un error de 1,6 % y de 0,7 % con distorsiones elásticas.

Simard et al. [5] con un CNN sencilla logra un error de 0,4 % con distorsiones elásticas.

## 1.2. Justificación

La visión por computadora se ha convertido en un campo de alta relevancia en los sistemas inteligentes actuales. Esta se encuentra en sistemas de verificación de rostro para un acceso automatizado o incluso como auxiliar en un diagnóstico médico y hasta en la detección múltiple de objetos como parte de un sistema más complejo que permita la navegación automática de un automóvil. La batalla por el mejor desempeño al trabajar con imágenes parece que ha sido ganada por las CNN; sin embargo, el cómputo cuántico puede ofrecer ciertas opciones que favorezcan el procesamiento y clasificación de imágenes. Por ejemplo, un circuito cuántico trabaja intrínsecamente de forma paralela; y debido a la superposición es posible trabajar con imágenes de dimensiones grandes utilizando pocos qubits.

Al ofrecer un nuevo modelo para la clasificación de imágenes, la visión por computadora y las áreas donde esté involucrada, se verán beneficiadas.

## 1.3. Objetivos

El problema de investigación que se busca resolver es el siguiente: ¿Es posible que un modelo cuántico del perceptrón sea capaz de clasificar un banco de datos de imágenes en un problema biclase, tanto en clases balanceadas como altamente desbalanceadas?

Para resolver esta problemática se plantearon los siguientes objetivos:

### 1.3.1. Objetivo General

Mostrar que el aprendizaje de máquina cuántico tiene un gran potencial, proponiendo e implementando modelos para la clasificación de imágenes, y de patrones en general, que permitan explotar las ventajas intrínsecas de los fenómenos cuánticos.

### 1.3.2. Objetivos Específicos

Como objetivos específicos se tienen:

- Implementar un clasificador cuántico existente.
- Clasificar imágenes binarias en un problema biclase.
- Proponer un modelo cuántico para la clasificación de imágenes multicanal.

- Implementar el modelo cuántico propuesto para la clasificación de imágenes multicanal. 1
- Clasificar imágenes de tres canales. 2

## 1.4. Estructura 3

Este trabajo está constituido por seis capítulos en los que se realiza un estudio sobre algunos modelos cuánticos para la clasificación de imágenes y patrones en general. 4 5

En el Capítulo 1 se presentó una breve introducción en la cual se incluye de forma no exhaustiva algunos antecedentes, justificación y los objetivos a cubrir. El resto del trabajo consta de la siguiente estructura: 7 8 9 10

El Capítulo 2 contiene el estado del arte, incluyendo los trabajos e investigaciones más recientes y relevantes para los objetivos que nos atañen. Los materiales utilizados y los métodos empleados son descritos en el Capítulo 3; además, en ese capítulo se incluyen conceptos necesarios para la fundamentación y desarrollo del trabajo. En el Capítulo 4 se presentan los modelos propuestos e implementados que buscan cubrir el objetivo general. En el Capítulo 5 se reportan los resultados obtenidos derivados de los procesos experimentales y se procura una interpretación de estos. El Capítulo 6 contiene las conclusiones relacionadas a los objetivos originales y se presentan propuestas de trabajos futuros derivados de este. Posteriormente se incluyen todas las referencias relevantes del presente trabajo. 11 12 13 14 15 16 17 18 19

# 1 Capítulo 2

## 2 Estado del arte

3 En este capítulo se reportan algunos de los trabajos e investigaciones más relevantes relacio-  
4 nados con QML y en especial con la clasificación de imágenes.

5  
6 La visión por computadora es un campo científico multidisciplinario en el que se busca  
7 que las computadoras obtengan información a partir de imágenes o vídeo. Algunas de las áreas  
8 científicas con las que hay intersección es el reconocimiento de patrones e inteligencia artificial.  
9 El conjunto de tareas es amplio, desde el procesamiento de imágenes hasta la segmentación,  
10 clasificación y extracción de información a partir de una imagen o una secuencia de imágenes.  
11 El rango de aplicaciones que puede tener el reconocimiento de patrones dentro de la visión por  
12 computadora es bastante amplio, pudiendo ir desde la clasificación de objetos en un ámbito  
13 industrial [6, 7], hasta ayudar en la detección y diagnóstico médico de enfermedades [8–10].  
14 Con el poder ya conocido que tienen los modelos clásicos para resolver el problema de la  
15 clasificación de imágenes [3], y en conjunto con el avance de los últimos años en hardware  
16 para el diseño de computadoras cuánticas programables, así como en la liberación de software  
17 que nos permite interactuar con ellas [12], se ha observado un surgimiento considerable de  
18 propuestas de modelos cuánticos para la clasificación de patrones [13].

19  
20 Maria Schuld et al. [14] proponen una implementación de un perceptrón cuántico basándose  
21 en un algoritmo cuántico de estimación de fase. También proponen un algoritmo de entrena-  
22 miento compatible con la teoría cuántica. Sin embargo, el algoritmo necesita  $2n$  qubits para  
23 funcionar, con  $n$  siendo el número de bits en la representación binaria del valor de entrada.

24

Sebastien Piat et al. [15] proponen un modelo híbrido completo para la clasificación de imágenes. Se compone de tres etapas, dos de ellas clásicas y una cuántica. La primera etapa es clásica, en la que se lleva a cabo el entrenamiento de un autoencoder para la reducción de dimensiones en las imágenes originales. La segunda etapa es cuántica, la cual consiste en entrenar una máquina de Boltzmann restringida (por sus siglas en inglés restricted Boltzmann machine RBM) utilizando una máquina de *quantum annealing* y tomando las imágenes dimensionalmente reducidas como los datos de entrenamiento. Finalmente, la última etapa es clásica y se compone del entrenamiento de una red neuronal clásica que tome como entradas las señales de salida de la RBM y arroje la clase asignada.

Yamamoto et al. [16] extienden, con modificaciones, el trabajo de Maria Schuld et al. [14] empleando una modificación en los operadores unitarios necesarios en el algoritmo de estimación de fase. El entrenamiento que proponen es básicamente el del descenso de gradiente para el perceptrón clásico. Con estos ajustes argumentan que es posible clasificar un banco de datos de dos clases no linealmente separables. Sin embargo, en este trabajo no se muestran resultados sobre esta aseveración sino solamente sobre la evolución de los pesos sinápticos durante el entrenamiento.

Maxwell Henderson et al. [17] proponen una capa cuántica de convolución en esencia similar a las capas de convolución clásicas. Sin embargo, los circuitos que calculan la *convolución* son generados de forma aleatoria, permanecen fijos y no son entrenados de ninguna forma. Ya que la capa de convolución cuántica es el primer procesamiento y es seguida por una CNN clásica, entonces es muy probable que los resultados favorables presentados sean consecuencia del buen desempeño de esta CNN clásica al aprender un mapeo de la salida de los circuitos a la clase de pertenencia. En este caso la capa de convolución cuántica no sería un extractor de características, sino que actuaría como una función de transformación.

Francesco Tacchino et al. [18] proponen un modelo de perceptrón cuántico que utiliza una subrutina llamada *hypergraph states generation subroutine (HSGS)* [19] que permite codificar un patrón binario en una representación de un estado cuántico por medio de una secuencia de compuertas cuánticas. Al representar de esta forma tanto al patrón de entrada como al vector de pesos permite emular al perceptrón clásico. También se dan a la tarea de plantear un modelo de entrenamiento sobre el vector de pesos.

1       A pesar de tantos modelos propuestos, como puede observarse en los anteriores trabajos, hay  
2 una falta de evaluación de dichos modelos sobre bancos de datos reales para poner a prueba la  
3 eficacia de las propuestas. Si algunos lo hacen es con bancos artificiales creados específicamente  
4 para ese modelo, lo cual sesga el resultado, o son modelos híbridos lo cual pone en duda la  
5 participación de la etapa cuántica.



## Capítulo 3

1

## Materiales y Métodos

2

En este capítulo se abordan los conceptos necesarios para la correcta fundamentación de este trabajo. Finalmente se describirán los bancos de datos utilizados para evaluar los modelos propuestos e implementados, descritos en el Capítulo 4.

3

4

5

Los conceptos y temas para analizar en este capítulo son: conceptos esenciales en la clasificación de patrones; algoritmos clásicos para la clasificación de patrones, prestándole principal atención a las redes neuronales artificiales (por sus siglas en inglés artificial neural networks ANN); presentación del cómputo cuántico y sus componentes principales.

6

7

8

9

### 3.1. Reconocimiento de Patrones

10

El campo de reconocimiento de patrones se ocupa del descubrimiento automático de regularidades en los datos mediante el uso de algoritmos computacionales y del uso de estas regularidades para tomar acciones tales como la clasificación de los datos en distintas categorías [20].

11

12

13

14

Estos algoritmos pueden ser agrupados en tres conjuntos:

15

Aprendizaje Supervisado [21]. En este conjunto, un experto proporciona previamente una categoría o costo por cada patrón en un conjunto de entrenamiento. El objetivo es diseñar un modelo que minimice la suma de los costos para ese conjunto de patrones.

16

17

18

Aprendizaje No Supervisado [21]. En este tipo de algoritmos no existe un experto que proporcione información explícita. El sistema busca formar agrupamientos *naturales* para los patrones de entrada.

19

20

21

Aprendizaje por Reforzamiento [21]. En esta categoría no existe un sistema experto sino un crítico, el cual clasificaría de forma binaria, es decir, correcto o incorrecto, la clasificación generada por el sistema de clasificación. Sin embargo, este crítico no proporciona información sobre el porqué de su veredicto.

### 3.1.1. Redes Neuronales

Es considerable el número de modelos con cierta aceptación para la clasificación de patrones [20, 21], lo cual podría generar un trabajo extenso analizar cada uno de ellos, por lo que sólo hablaremos un poco del modelo más pertinente para este trabajo, el cual es el perceptrón y su generalización a las MLP y ANN.

El algoritmo perceptrón fue inventado en 1957 por Frank Rosenblatt [22]. Se le diseñó originalmente para ser una máquina para la clasificación de imágenes, fue implementado primero como un programa de computadora [20].

En 1969, Minsky y Papert fueron famosamente malinterpretados cuando demostraron que un único perceptrón solo puede aprender reglas en problemas linealmente separables, por ejemplo, es incapaz de producir la función XOR [23]. Sin embargo, ya era sabido por ellos que un ensamble de estos perceptrones, es decir una red neuronal artificial, era capaz de resolver problemas no linealmente separables.

En 1986, Rumelhart, Hinton y Williams [24] comenzaron la popularización del término *backpropagation* para el algoritmo de entrenamiento de las redes neuronales artificiales. Sin embargo, este algoritmo puede ser rastreado hasta la década de los sesentas [25].

El ensamble de perceptrones, llamado MLP es el inicio de las ANN más complejas que se pueden observar en el estado del arte, como lo son las CNN y sus variantes, y los resultados que estas otorgan son notables; sin embargo, también es considerable la enormidad de parámetros que deben ser optimizados [26–29].

## 3.2. Cómputo Cuántico

El Cómputo Cuántico es el uso de los fenómenos cuánticos tales como la superposición y el entrelazamiento para procesar información con base en los principios de la mecánica cuántica [30]. Fue propuesta por primera vez a principios de 1980 como una forma de simular y modelar sistemas cuánticos que por medio de las computadoras clásicas sería imposible [31].

Explotando los fen  menos de la mec  nica cu  ntica y aplic  ndolos en procesos de c  mputo se puede acelerar y optimizar el c  mputo cl  sico e incluso resolver problemas inviables por mecanismos cl  sicos; al menos, de forma te  rica [32].

### 3.2.1. Qubit

En la computaci  n e informaci  n cu  ntica, el qubit es la unidad b  sica de informaci  n, en este sentido es el an  logo al bit cl  sico. F  sicamente el qubit es un sistema cu  ntico con dos estados posibles, pudiendo realizarse por medio de la polarizaci  n de un fot  n, el esp  n de un electr  n, puntos cu  nticos o por las uniones de Josephson.

Matem  ticamente cada estado se representa por medio de uno de dos vectores que forman una base ortonormal, conocida como la base computacional. Dichos vectores son usualmente denotados por medio de la notaci  n de Dirac como  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  y  $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  y son llamados *ket 0* y *ket 1*, respectivamente. Tambi  n existe los vectores transpuestos conjugados, los cuales se conocen como *bra 0* y *bra 1* y son representados como  $\langle 0| = (1 \ 0)$  y  $\langle 1| = (0 \ 1)$ .

### 3.2.2. Esfera de Bloch

En computaci  n cu  ntica la esfera de Bloch es una representaci  n geom  trica de un sistema cu  ntico de dos estados. La esfera de Bloch es una esfera en un espacio de tres dimensiones donde los pares de puntos antipodales corresponden a vectores mutuamente ortogonales. Por convenci  n, el *polo norte* corresponde al estado  $|0\rangle$  mientras que el *polo sur* corresponde al estado  $|1\rangle$ .

Dado que los vectores base son ortonormales, la esfera de Bloch es unitaria y la superficie corresponde a los estados puros, mientras que el interior de ella corresponde a los estados mixtos. En la Figura 3.2 se pueden observar algunos estados representados en la esfera de Bloch.

Cualquier estado representado en la esfera de Bloch puede ser inequ  vocamente representada por su magnitud y los tres   ngulos de Euler, como pueden verse en la Figura 3.1, los cuales efect  an las siguientes rotaciones:

- El   ngulo  $\theta$  describe una rotaci  n alrededor del eje N, anteriormente x.
- El   ngulo  $\phi$  describe una rotaci  n alrededor del eje z.

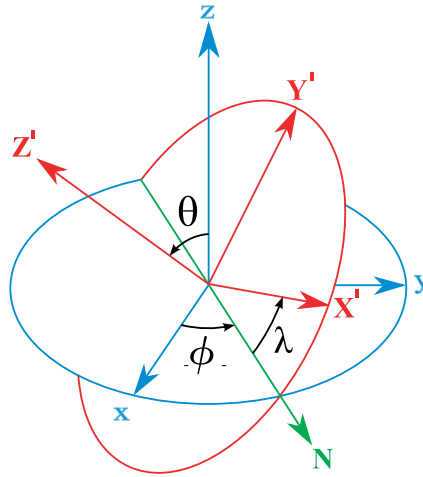


Figura 3.1 Rotaciones efectuadas por los   ngulos de Euler.

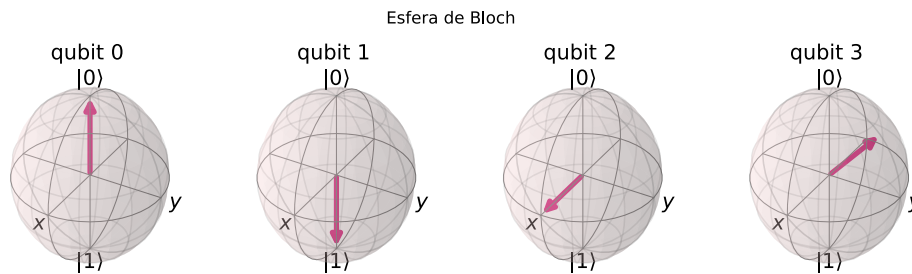


Figura 3.2 Esfera de Bloch representando diversos estados cu  nticos. La primera esfera contiene ilustra el estado  $|0\rangle$ . El qubit 1 se encuentra en el estado  $|1\rangle$ , mientras que el qubit 2 se encuentra en una superposici  n de estados. El   ltimo qubit ha sido rotado con los   ngulos  $(\frac{\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4})$ .

- 1 ■ El   ngulo  $\lambda$  describe una rotaci  n alrededor del eje  $Z'$ , anteriormente  $z$ .

### 2 3.2.3. Producto Tensorial o Producto de Kronecker

3 Matem  ticamente hablando, el producto de Kronecker,  $\otimes$ , es una operaci  n en dos matrices  
4 que resulta en una matriz a bloques. Es una generalizaci  n del producto externo vectorial al  
5 producto externo matricial.

6 En forma general, si  $\mathbf{A}$  es una matriz de dimensiones  $m \times n$  y  $\mathbf{B}$  es una matriz de dimensiones  
7  $k \times l$ , el producto de Kronecker  $\mathbf{A} \otimes \mathbf{B}$  es entonces de dimensiones  $km \times ln$  y tiene la forma

compacta:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}, \quad (3.1)$$

Mientras que explícitamente puede escribirse como:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}. \quad (3.2)$$

Es usado para describir sistemas cuánticos que consisten en múltiples subsistemas. Cada subsistema esta descrito por un vector que habita su propio espacio vectorial. Si tomamos como ejemplo la base computacional  $|0\rangle, |1\rangle$  y generamos el producto de Kronecker entre ellos se obtiene:

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 0 \\ 1 \cdot 1 \\ 0 \cdot 0 \\ 0 \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \quad (3.3)$$

Y de forma similar se pueden obtener las otras tres combinaciones de productos y llegar al conjunto  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$  que son la base ortonormal para un sistema cuántico de cuatro estados.

En palabras simples, el producto tensorial o de Kronecker nos permite formar sistemas cuánticos complejos a partir de subsistemas más simples.

### 3.2.4. Fenómenos

El Cómputo Cuántico hace uso principalmente de dos fenómenos: superposición y entrelazamiento. Es necesario conocer estos conceptos como base para el desarrollo y entendimiento de este trabajo.

#### Superposición

Paul Dirac describió formalmente los principios básicos de la mecánica cuántica [33], donde establece que la superposición cuántica implica que la adición de dos estados cuánticos genera un nuevo estado cuántico. Matemáticamente emerge de la linealidad de la ecuación de Schrödinger, ya que si dos estados cuánticos son solución a la ecuación para un sistema dado, también lo será su combinación lineal. Por ejemplo, considerando la base computacional  $\{|0\rangle, |1\rangle\}$  se puede expresar la superposición:

$$c_1|0\rangle + c_2|1\rangle, \quad c_1, c_2 \in \mathbb{C}, \quad |c_1|^2 + |c_2|^2 = 1. \quad (3.4)$$

Esta propiedad es explotable al momento de hacer cómputo ya que permite considerar varios estados simultáneamente, a diferencia de los modelos clásicos, en los que solo se puede analizar un único estado en una operación.

Un qubit, que es el análogo a un bit clásico, explota esta propiedad al poder estar en dos estados de forma simultánea, es decir, en una superposición de estados base.

#### Entrelazamiento

El entrelazamiento cuántico es el fenómeno que ocurre cuando dos o más sistemas cuánticos no pueden ser descritos de forma independiente, incluso si estos sistemas son separados espacialmente de forma arbitraria.

Gracias a esta propiedad, si se realizara alguna medición de un observable físico a alguno de los sistemas que componen el entrelazamiento, se encontraría que el mismo observable en los otros sistemas se encuentra correlacionado con el que fue medido. Sin embargo, esto no implica de ninguna forma transmisión de información entre dichos sistemas, como nos asegura el teorema de *No-Comunicación* [34, 35].

Por ejemplo, si se considera la base  $\{|0\rangle_A, |1\rangle_A\}$  de un sistema  $A$  y la base  $\{|0\rangle_B, |1\rangle_B\}$  de un sistema  $B$ , el estado

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} (|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B) \quad (3.5) \quad 1$$

es un estado cuántico entrelazado. Este es uno de los muy conocidos estados de Bell. 2

Si se realizara una medición sobre la observable  $\sigma_z$  en la base de  $A$ , encontraríamos que el sistema colapsaría a  $|0\rangle_A|1\rangle_B$  o  $|1\rangle_A|0\rangle_B$  con la misma probabilidad. Sin importar qué medición se realice, el resultado obtenido de esta en el sistema  $B$  estará correlacionado con el obtenido en el sistema  $A$ . Por ejemplo, si el resultado de la medición de la partícula  $A$  es  $|0\rangle_A$ , es decir, el sistema colapsa a  $|0\rangle_A|1\rangle_B$ , así que sin realizar medición en la partícula  $B$  seguramente se encuentra en el estado  $|1\rangle_B$ . Si por el contrario, el resultado de la medición en  $A$  fuera  $|1\rangle_A$ , en  $B$  se encontraría  $|0\rangle_B$ . 3 4 5 6 7 8 9

El entrelazamiento cuántico es un fenómeno muy poderoso al realizar cómputo cuántico, ya que, al realizar una operación sobre una parte del sistema, afecta de forma simultánea e inmediata a todos los sistemas con los que se tenga el entrelazamiento. Este fenómeno en conjunto con la superposición nos otorgan el paralelismo de forma natural al realizar cómputo de forma cuántica. 10 11 12 13 14

### 3.2.5. Circuitos Cuánticos 15

El método más común y discutiblemente más sencillo de realizar cómputo cuántico es a través de los circuitos cuánticos. Estos circuitos utilizan como componentes de construcción las compuertas cuánticas, construyendo los circuitos cuánticos de una forma similar en que las compuertas lógicas clásicas construyen los circuitos digitales convencionales. 16 17 18 19

Las compuertas cuánticas son modeladas por medio de matrices unitarias. Estas compuertas pueden actuar en uno o más qubits y cumplen que el número de qubits de entrada coincide con el número de qubits de salida. Aparte de la condición de que la matriz sea unitaria, no hay, en principio otra restricción sobre ellas, y en consecuencia sobre la compuerta cuántica que representa. Al ser matrices unitarias siempre existe su inversa, la cual es la transpuesta conjugada de la misma matriz; esta propiedad permite que las operaciones en el cómputo cuántico sean reversibles, a diferencia del cómputo clásico. 20 21 22 23 24 25 26

La forma general para las compuertas que actúan sobre un solo qubit en su representación matricial tiene la forma: 27 28

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)} \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \quad (3.6) \quad 29$$

1 donde  $\theta, \phi, \lambda$  son los ángulos de Euler.

2 A partir de esta, se pueden generar todas las compuertas cuánticas actuantes sobre un qubit.

3 Por ejemplo, las compuertas más famosas son los operadores de Pauli y la compuerta Hadamard.

4 Los operadores de Pauli tienen la forma:

$$\begin{aligned}
 \sigma_0 &= I = U(0, \phi, \lambda) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
 \sigma_1 &= \sigma_x = U(\pi, 0, \pi) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 \sigma_2 &= \sigma_y = U(\pi, \frac{\pi}{2}, \frac{\pi}{2}) = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\
 \sigma_3 &= \sigma_z = U(0, 0, \pi) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},
 \end{aligned}
 \tag{3.7}$$

6 mientras que la compuerta Hadamard tiene la forma:

$$H = U\left(\frac{\pi}{2}, 2\pi, \pi\right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},
 \tag{3.8}$$

8 la cual, al actuar sobre un qubit lo pone en superposición cuántica de sus estados base. Es  
 9 decir, mapea del estado  $|0\rangle$  a  $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$  y del  $|1\rangle$  a  $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ . En conjunto con la compuerta CNOT, la  
 10 compuerta Hadamard puede generar entrelazamiento. Si se toma un estado de entrada  $|x\rangle|y\rangle$  con  
 11  $x, y \in 0, 1$  los entrelaza a uno de los estados de Bell, que tienen la forma general  $\frac{|0,y\rangle+(-1)^x|1,\bar{y}\rangle}{\sqrt{2}}$ ,  
 12 donde  $\bar{y}$  es la negación de  $y$ . La representación matricial de la compuerta cuántica CNOT es la  
 13 siguiente:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.
 \tag{3.9}$$

15 La CNOT es un caso especial de la compuerta controlada CU. Esta solo actuará sobre un  
 16 segundo qubit si y solo si el qubit de control se encuentra en el estado  $|1\rangle$ , donde U es una



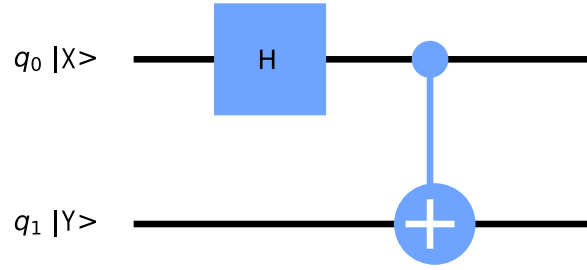


Figura 3.3 Este circuito, empleando la compuerta Hadamard seguida por una CNOT genera entrelazamiento entre los qubits de entrada  $q_0$  y  $q_1$ .

transformación unitaria.

$$CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}. \quad (3.10)$$

Utilizando la compuerta  $U$  general, la  $CNOT$  y con la ayuda del producto tensorial se puede construir cualquier compuerta cuántica que actúe sobre un número arbitrario de qubits que sea potencia de dos [36, 37]. Lo que es más, el teorema *Solovay–Kitaev* nos asegura que podremos aproximar cualquier compuerta que actúe sobre un solo qubit de manera eficiente [38, 39].

### 3.3. Bancos de datos

En esta sección se describen los bancos de datos usados para este trabajo. Los primeros dos están compuestos por varias imágenes de dígitos. El tercero está compuesto por imágenes de frutas.

El primero de ellos es el *Optical Recognition Handwritten Digits Data Set* (Digits Dataset) el cual contiene 5620 instancias en total, cada una de ellas es una imagen en escala de grises de dimensión  $8 \times 8$  para un total de 64 atributos. Cada atributo tiene un valor entero en el rango  $[0, 16]$ . Este banco de datos es uno balanceado y consta de 10 clases, una por cada dígito [40].

El segundo banco de datos es el *Semeion Handwritten Digit Dataset* (Semeion Dataset) el cual contiene 1593 instancias totales. Cada instancia es una matriz de  $16 \times 16$  para un total de 256 atributos binarios, es decir, solo toman un valor del conjunto  $\{0, 1\}$ . También es un banco balanceado y consta de 10 clases, una por cada dígito [41, 42].

En la Tabla 3.1 se presenta de forma condensada la descripción de ambos bancos de datos, incluyendo la proporción de desbalance (por sus siglas en inglés Imabalance Ratio IR), el cual

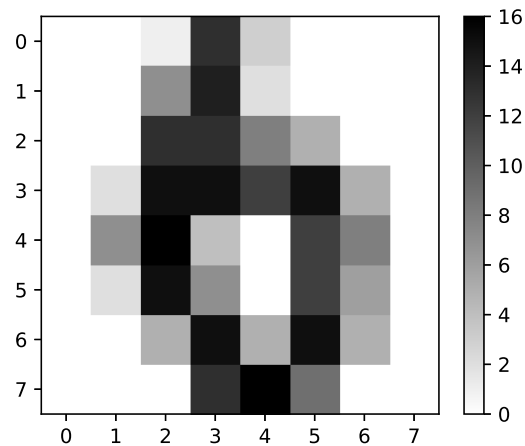


Figura 3.4 Instancia de la clase 6 en el banco de datos *Digits*. Cada atributo, o píxel, tiene un valor entero perteneciente al intervalo  $[0, 16]$ . Como puede verse, la baja resolución implica una complejidad extra.

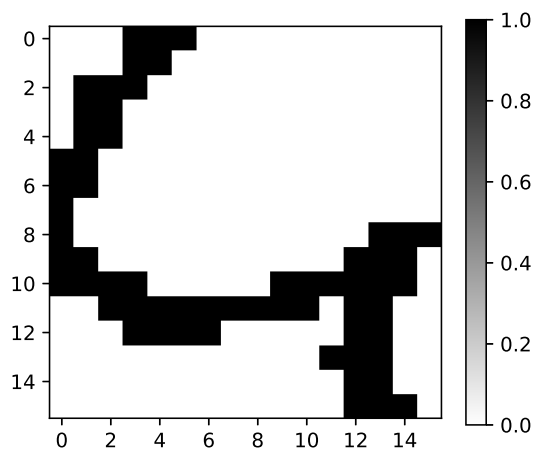


Figura 3.5 Instancia de la clase 4 en el banco de datos *Semeion*. Cada atributo, o píxel, es un valor binario. Estos patrones tienen una dimensionalidad mayor, por lo que la complejidad del procesamiento incrementa.

Cuadro 3.1 Resumen de los bancos de *Digits* y *Semeion*.

Banco de Datos	Clases	Desbalance	Instancias Totales
Digits	10	1.032	5620
Semeion	10	1.045	1593

Figura 3.6 Instancias del subconjunto usado del banco de datos *Fruits*.

se define como la proporción entre la clase mayoritaria y la clase minoritaria [43]. Mientras que en la Figura 3.4 y Figura 3.5 se muestra un ejemplo del banco de datos *Digits* y *Semeion*, respectivamente.

En este momento es necesario señalar que cuando se hable de problemas de clases altamente desbalanceadas en las secciones posteriores, se estará refiriendo a la siguiente modificación lógica en el banco de datos: se seleccionará una de las clases y se etiquetará como clase positiva, dejando al resto del banco como la clase negativa. Esta modificación genera un desbalance de clases bastante considerable, teniendo un  $IR \approx 9$ .

El último banco de datos es el *Fruit Recognition*, el cual contiene un total de 22,495 instancias totales repartidas en 33 clases. Como puede verse en la Figura 3.6, cada instancia es una imagen RGB de un tamaño de  $100 \times 100$  píxeles. De acuerdo con su publicación original [44], el banco se encuentra separado en un conjunto de entrenamiento con 16,854 instancias y un conjunto de prueba de 5,641 imágenes. El conjunto de entrenamiento se encuentra separado por clases, sin embargo, el conjunto de prueba no. Debido a esto y a que el modelo a emplear es en esencia un clasificador binario, solo se empleará un subconjunto de este banco de datos. En especial se tomarán las instancias de las clases *Banana* y *Onion White*, con 490 y 438 instancias totales respectivamente para un  $IR \approx 1,11$ .

# Capítulo 4

## Propuesta de Investigación

En este capítulo se describen los métodos y modelos utilizados para el desarrollo de este trabajo, tomando primordial atención en la implementación teórica.

### 4.1. Perceptrón Cuántico

En esta sección se describe la parte teórica desarrollada en [18], en el cual se propone un perceptrón cuántico, el cual ha sido implementado en este trabajo para realizar la clasificación de los bancos de datos descritos en la Sección 3.3.

Para este método, es necesario que el patrón de entrada sea de atributos binarios. Comenzando con el patrón de entrada  $\vec{x}^T = (x_0 \dots x_{m-1})$  y un vector de pesos  $\vec{\Omega}^T = (\Omega_0 \dots \Omega_{m-1})$ , donde  $x_j, \Omega_j \in \{0, 1\}$  se mapean a

$$\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{pmatrix}, \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix} \quad (4.1)$$

donde  $i_j, w_j \in \{-1, 1\}$  y con esto se pueden definir dos estados cuánticos

$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle \text{ y } |\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle. \quad (4.2)$$

El mapeo puede ser definido arbitrariamente, y en este trabajo, para efectos de similitud con [18], se usó el mapeo  $i_j = (-1)^{x_j}$  y similarmente  $w_j = (-1)^{\omega_j}$ .

Los estados  $|j\rangle \in \{|00\dots 00\rangle, |00\dots 01\rangle, \dots, |11\dots 11\rangle\}$  constituyen la base computacional de un procesador cuántico. Si  $N$  qubits son usados en el registro del procesador, entonces hay  $m = 2^N$  estados base etiquetados  $|j\rangle$ , y se pueden usar factores  $\pm 1$  para codificar patrones y vectores de peso clásicos,  $m$ -dimensionales, en una superposición uniformemente pesada usando la base computacional.

El primer paso consiste en preparar el estado  $|\psi_i\rangle$  al codificar los valores de entrada de  $\vec{i}$ . Con los qubits inicializados en el estado cero  $|00\dots 00\rangle \equiv |0\rangle^{\otimes N}$ , se le aplica una transformación unitaria  $U_i$

$$U_i|0\rangle^{\otimes N} = |\psi_i\rangle . \quad (4.3)$$

El segundo paso consiste en calcular el producto interno entre  $\vec{w}$  y  $\vec{i}$  usando los registros cuánticos. Esto puede obtenerse al definir una transformación unitaria,  $U_w$ , tal que

$$U_w|\psi_w\rangle = |1\rangle^{\otimes N} = |m-1\rangle . \quad (4.4)$$

Si se aplica  $U_w$  después de  $U_i$ , el estado cuántico se transforma en

$$U_w|\psi_i\rangle = \sum_{j=0}^{m-1} c_j |j\rangle \equiv |\phi_{i,w}\rangle . \quad (4.5)$$

Usando la Ec. (4.4), el producto interno entre dos estados cuánticos es

$$\begin{aligned} \langle \psi_w | \psi_i \rangle &= \langle \psi_w | U_w^\dagger U_w | \psi_i \rangle \\ &= \langle m-1 | \phi_{i,w} \rangle = c_{m-1} , \end{aligned} \quad (4.6)$$

y por las definiciones en Ec. (4.2) se puede observar que el producto interno del patrón de entrada y el vector de pesos es  $\vec{w} \cdot \vec{i} = m \langle \psi_w | \psi_i \rangle$ . Por tanto, el valor deseado se encuentra codificado, considerando un factor de normalización, en el coeficiente  $c_{m-1}$  del estado final  $|\phi_{i,w}\rangle$ .

Para poder extraer tal información, un qubit auxiliar ( $a$ ) inicializado en el estado  $|0\rangle$  es actualizado a través de una CNOT multicontrolada a través de los  $N$  qubits utilizados para la codificación, esto nos lleva a

$$I^{\otimes N} C^N NOT |\phi_{i,w}\rangle |0\rangle_a = \sum_{j=0}^{m-2} c_j |j\rangle |0\rangle_a + c_{m-1} |m-1\rangle |1\rangle_a . \quad (4.7)$$

La *no linealidad* requerida por la función de activación en la salida del perceptrón es satisfecha de forma inmediata al realizar una medición cuántica. Efectuando la medición en el qubit auxiliar, se obtiene una lectura de  $|1\rangle_a$  con probabilidad  $|c_{m-1}|^2$ .

#### 4.1.1. Los operadores unitarios $U_w$ y $U_i$

Los operadores unitarios  $U_i$  y  $U_w$  son la base fundamental para el funcionamiento del algoritmo.

En principio, el operador  $U_i$  puede ser construido por una matriz unitaria de dimensiones  $m \times m$  que contenga el vector  $\vec{i}$  como su primera columna. En un modelo ideal, el estado  $|\psi_i\rangle$  sería extraído directamente de una memoria cuántica donde fue previamente almacenado.

De forma similar, el operador  $U_w$  puede ser construido por una matriz unitaria de dimensiones  $m \times m$  que contenga el vector  $\vec{w}$  en su última fila.

Sin embargo, los operadores también pueden ser generados por un algoritmo basado en la *hypergraph state generation subroutine (HSGS)* [19], el cual es descrito a continuación y cuyo diagrama de flujo puede verse en la Figura 4.1.

Comenzando con el vector de entrada  $\vec{i}$  se puede obtener el operador unitario iniciando desde una superposición cuántica al aplicar el producto tensorial de compuertas Hadamard  $H^{\otimes N}$  al estado inicial  $|0\rangle^{\otimes N}$ . Después se itera con los siguientes pasos. Primero se checan los estados con un solo qubit en el estado  $|1\rangle$ , en otras palabras, los estados que tienen la forma  $|0\dots 010\dots 0\rangle$  que requieran un factor de  $-1$  en la representación del estado  $|\psi_i\rangle$  en la base computacional. Si este es el caso se aplica una compuerta Z al qubit que esté en el estado  $|1\rangle$ . Ya que esta compuerta puede introducir una fase de  $-1$  en algunos estados que tengan más de un qubit en el estado  $|1\rangle$ .

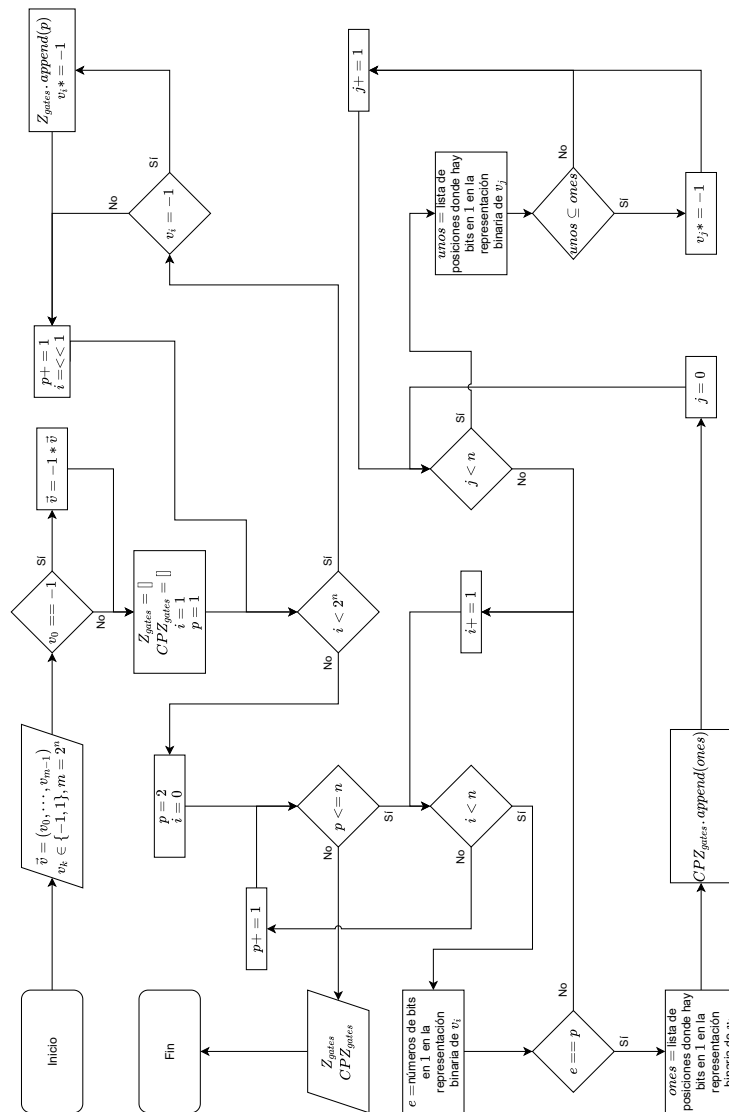


Figura 4.1 Diagrama de flujo que describe el algoritmo *HSGS* que permite obtener la secuencia de compuertas  $Z$  y  $C^pZ$  para la implementación de los operadores unitarios  $U_i$  y  $U_w$ .

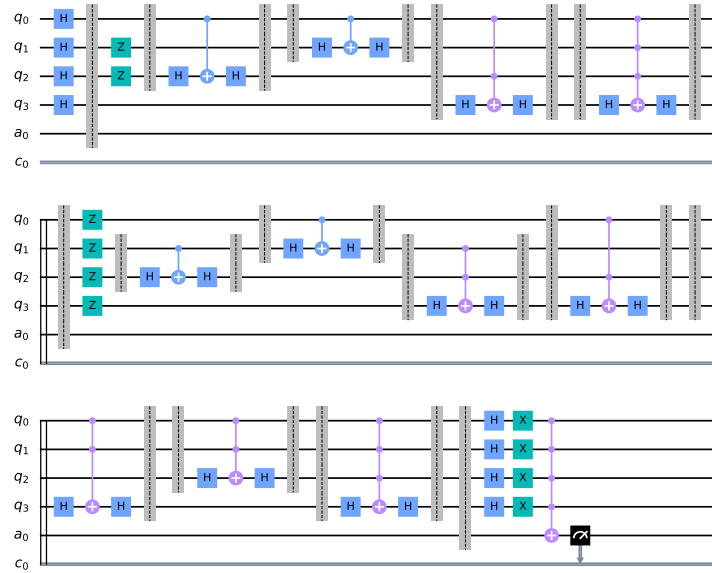


Figura 4.2 Circuito cuántico que genera  $U_i$  y  $U_w$ . Las barreras separan cada bloque relevante, tanto la generación de las compuertas Z multicontroladas empleando la compuerta X, así como la secciones que componen las compuertas que generan cada uno de los operadores unitarios.

El siguiente paso consiste en iterar  $p$  en un rango de  $2, \dots, N$  considerando las componentes que contengan exactamente  $p$  qubits en el estado  $|1\rangle$ . Por cada una de estas componentes se genera una fase de  $-1$ , en caso de ser necesario, por medio de una compuerta Z múltiplemente controlada,  $C^p Z$ , por los  $p$  qubits en el estado  $|1\rangle$ . Análogamente, si ya existe una fase no deseada de  $-1$ , esta puede ser removida aplicando la misma compuerta  $C^p Z$ , ya que esta compuerta actúa sobre todos los estados con  $p$  o más qubits en el estado  $|1\rangle$ , mientras que las componentes con un número menor a  $p$  qubits en el estado  $|1\rangle$  permanecen invariantes. Continuando con este procedimiento, al llegar a  $p = N$  se obtendrán todos los signos deseados y se habrá transformado  $|0\rangle^{\otimes N}$  en  $|\psi_i\rangle$ .

El operador unitario  $U_w$  puede ser generado de forma similar. Se generan los dos pasos análogos en la obtención de  $U_i$ , y como todas las compuertas utilizadas en HSGS son las inversas de si mismas y conmutan consigo mismas, la generación de  $U_w$  puede verse como la transformación unitaria que evoluciona  $|\psi_i\rangle$  en  $|0\rangle^{\otimes N}$ ; es decir, regresa el estado  $|\psi_i\rangle$  a una superposición homogénea en la base computacional  $|\psi_0\rangle$ . Para finalizar la transformación unitaria, basta con aplicar una compuerta Hadamard multiqubit  $H^{\otimes N}$  y aplicar una compuerta  $NOT$  multicontrolada por los  $N$  qubits,  $CNOT^{\otimes N}$ , y que controlan el estado del qubit ancilla  $|a\rangle$ .



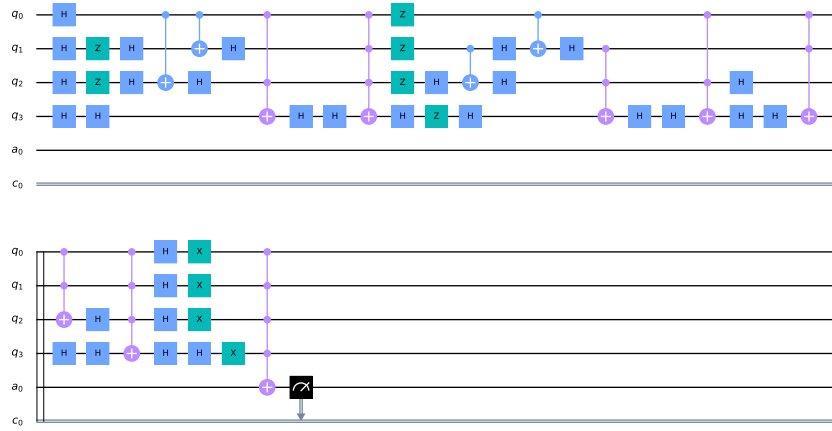


Figura 4.3 Circuito equivalente al de la Figura 4.2 pero sin las barreras, para ejemplificar el poder de cómputo cuántico en paralelo.

Es trivial mostrar la siguiente igualdad en compuertas cuánticas  $HXH = Z$  por lo que implementar las  $C^pZ$  es equivalente a  $I \otimes \dots \otimes I \otimes H \otimes I \otimes \dots \otimes IC^pXI \otimes \dots \otimes I \otimes H \otimes I \otimes \dots \otimes I = C^pZ$ , donde las compuertas Hadamard se aplican sobre el qubit objetivo de la compuerta multicontrolada. Esta modificación es necesaria ya que, hasta el momento de la implementación, Qiskit [12] no tiene implementada nativamente las compuertas  $Z$  multicontroladas.

En la Figura 4.2 se muestra el circuito cuántico con barreras activas, las cuales delimitan tanto las compuertas Hadamard iniciales, como los operadores  $U_i$  y  $U_w$  y las dos últimas secciones de Hadamard y la  $NOT$  multicontrolada hacia el qubit ancilla.

En la Figura 4.3 se muestra el mismo circuito que en Figura 4.2 con la diferencia de que las barreras han sido removidas, lo que permite ilustrar que en un circuito cuántico las operaciones pueden realizarse en paralelo de forma natural y tan pronto como sea posible. Por ejemplo, para aplicar una  $C^pZ$  es necesario aplicar una compuerta  $X$ , sin embargo, si el qubit al que se le va a aplicar esta compuerta no está involucrado en operaciones con el resto de qubits, se le puede aplicar inmediatamente. De esta forma se puede evolucionar un sistema cuántico aplicando varias compuertas de forma paralela a distintos qubits.

### 4.1.2. Método de entrenamiento

Antes de describir el método de entrenamiento, descrito en la Figura 4.5 y el método de clasificación descrito en la Figura 4.6, es necesario definir ciertos aspectos importantes. El primer aspecto a definir es la lectura del circuito cuántico. Debido a la forma en la que fue

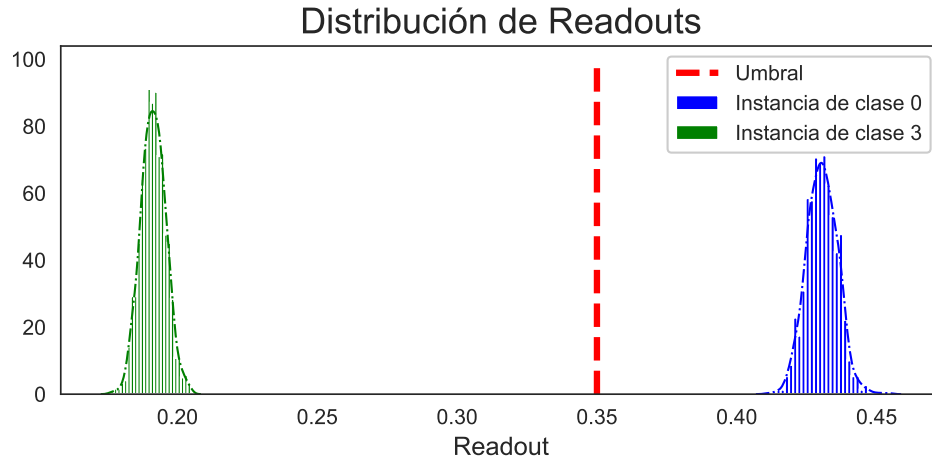


Figura 4.4 Distribución de los *readouts* después de mil iteraciones del mismo circuito cuántico empleando el vector de pesos final ya entrenado para el clasificador binario 3/0. La distribución a la izquierda es para una instancia de la clase 3 y se encuentra muy por debajo del umbral, mientras que la distribución a la derecha es para una instancia de la clase 0 y se encuentra por encima del umbral.

1 diseñado el circuito, la lectura almacenará en un bit clásico el valor de 0 si el qubit ancilla se  
 2 encontraba en el estado  $|0\rangle$  o un 1 si es que se encontraba en el estado  $|1\rangle$ . Esta operación de  
 3 medición, empleando el mismo circuito, se repite un número arbitrario de veces, y se calcula la  
 4 proporción de mediciones en el estado  $|1\rangle$  con respecto al número total de mediciones realizadas,  
 5 a este valor se le llamará *readout*. Entre más mediciones se realizan más cercano estará esta  
 6 proporción a la probabilidad encontrada al realizar la medición en la Ec. (4.7). El segundo  
 7 aspecto por considerar son las tasas de aprendizaje. Este es el análogo a la tasa de aprendizaje  
 8 común en los algoritmos clásicos de aprendizaje de máquina. En este caso hay dos tasas de  
 9 aprendizaje, una para la clase positiva y otra para la clase negativa,  $I_p$  y  $I_n$  respectivamente, con  
 10  $0 \leq I_p, I_n \leq 1$ .

11 Como es usual, la fase de entrenamiento es seccionada en épocas, en la que cada época  
 12 equivale a que el perceptrón cuántico procese el banco de datos por completo. Iniciando con un  
 13 vector de pesos  $\vec{w}_0$ , por cada patrón de entrada en cada época se realiza lo siguiente: se genera el  
 14 circuito cuántico y se calcula el readout. Si este valor es mayor que un umbral, definido previa y  
 15 arbitrariamente, a este patrón se le asignará la clase positiva, en caso contrario la clase asignada  
 16 será la negativa. Como se está trabajando en modelo de clasificación supervisada, se conoce de  
 17 antemano la clase real a la que pertenece el patrón en cuestión. Si el patrón fue correctamente

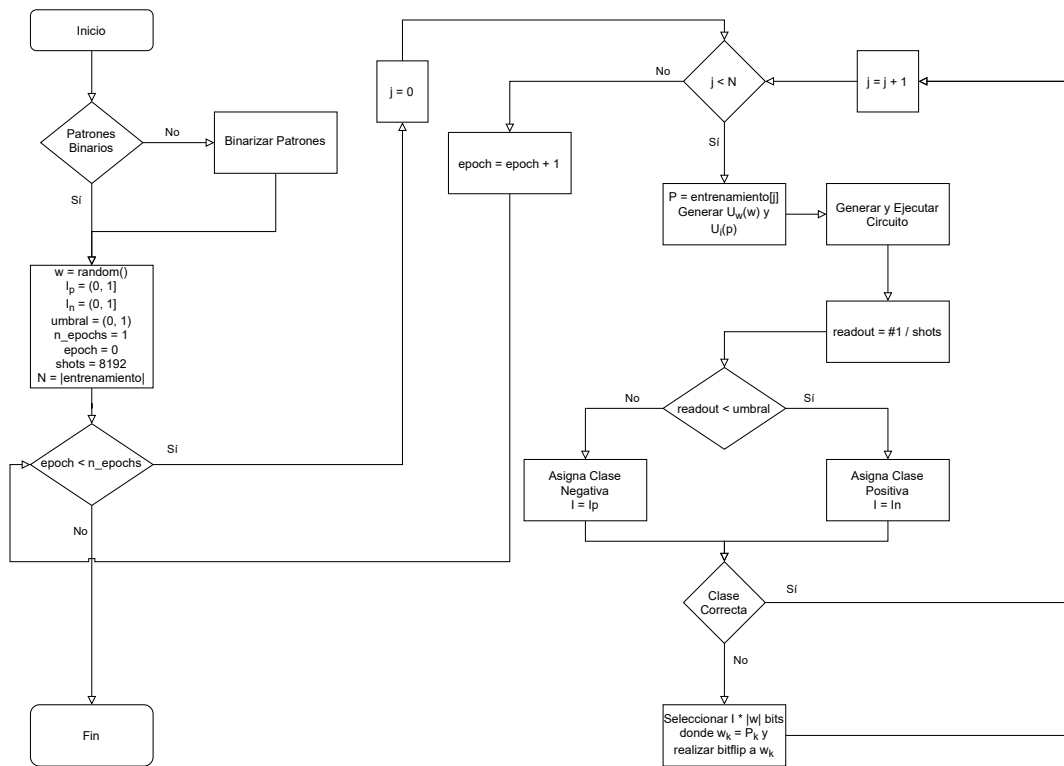


Figura 4.5 Diagrama de flujo para el proceso de entrenamiento para imágenes binarias.

clasificado simplemente se procesa el siguiente patrón. En caso contrario se modifica el vector de pesos  $\vec{w}$  realizando un cambio de signo  $\pm 1$  sobre cierta proporción del total de componentes donde coincidan los vectores  $\vec{i}$  y  $\vec{w}$ . Esta proporción es determinada por  $I_p$  si el patrón de entrada fue erróneamente clasificado como negativo y por  $I_n$  si fue clasificado como positivo, pero en realidad pertenece a la clase negativa.

Los valores del umbral con el que se compara el readout para la asignación de clase, como los valores de  $I_p$  y  $I_n$  son hiperparámetros que pueden ser ajustados para obtener un mejor desempeño.

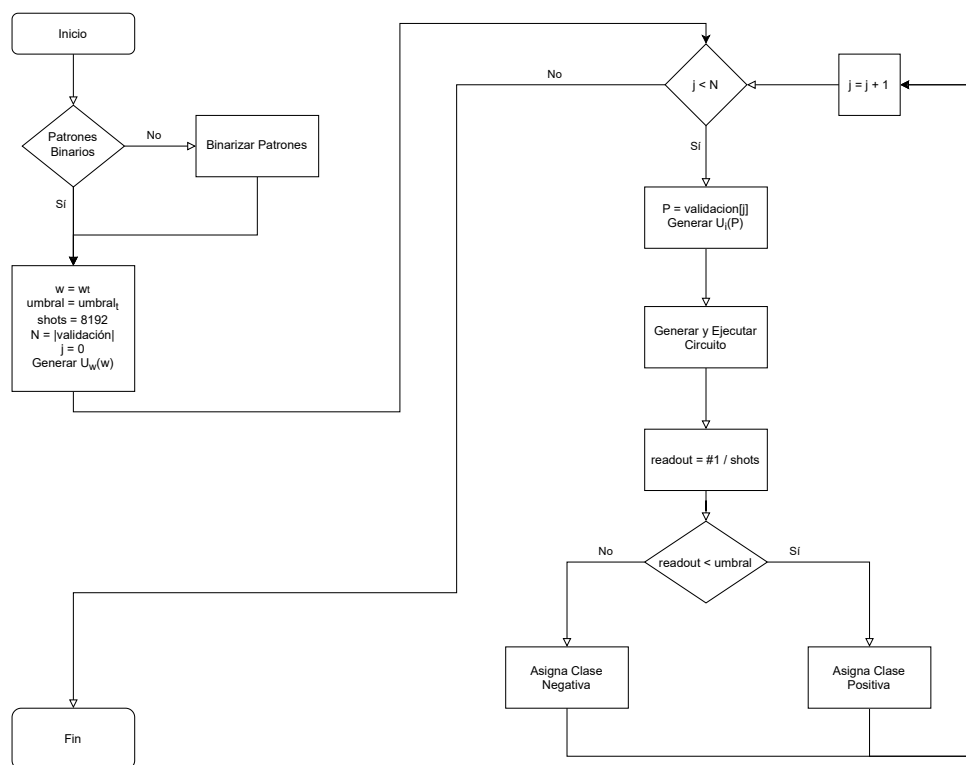


Figura 4.6 Diagrama de flujo para el proceso de validación para imágenes binarias.

## 4.2. Generalización a Tres Canales

Las imágenes a color usualmente se componen de tres o más canales. Por esto es interesante tratar de generalizar el modelo descrito previamente, del cual la generalización que se propone se extiende directamente.

Para esto consideramos una imagen de  $k$  canales binarios, es decir, todos los valores de cada canal toman valores 0 o 1. Cada canal se procesará independiente pero paralelamente en un circuito cuántico extendido.

El circuito cuántico extendido (CCE) se compone de  $k$  circuitos cuánticos que implementan el modelo descrito en 4.1, uno por cada canal, uno después del otro, expandiendo el CCE de forma vertical, como puede verse en la Figura 4.9. Al finalizar la ejecución paralela de los  $k$  circuitos se obtendrán las lecturas correspondientes, arrojando en esta ocasión una cadena de bits de orden  $k$ , por ejemplo  $1_1 0_2 1_3$  para  $k = 3$ , el cual es el número de canales más común correspondiente al modelo *RGB*. Los subíndices indican el canal al que pertenece la lectura de ese bit.

Similarmente al caso de un solo canal, el CCE se repite un número arbitrario de veces, con la intención de aproximarnos al resultado teórico ideal. De este conteo obtenemos los *readouts*, uno por cada canal. Cada *readout* es comparado con el umbral y se aplica la regla de clasificación, es decir, si el  $readout_i > umbral$  al canal  $i$  se le asigna la clase positiva, en caso contrario se le asigna la clase negativa.

$$clase_i = \begin{cases} positiva, & readout_i < umbral \\ negativa, & \text{otro caso} \end{cases} \quad (4.8)$$

La clasificación final de la imagen a color se realiza por mayoría de votos. La clase que más se haya asignado durante el análisis de los  $k$  *readouts* será la clase asignada a la imagen global. Si asignamos a la clase negativa el índice 0 y a la clase positiva el índice 1 podemos expresar formalmente la regla global de clasificación de la siguiente manera:

$$C = \arg \max_i \sum_{j=1}^k \delta_{C_r P_i}, \quad \delta_{xy} = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \quad (4.9)$$

donde  $C_r$  es la clase real y  $P_i$  es la clase asignada por el canal  $i$ .

La fase de entrenamiento es esencialmente la misma, con la diferencia que se actualizará los pesos por cada canal de forma individual. Si la clasificación global es errónea, por cada

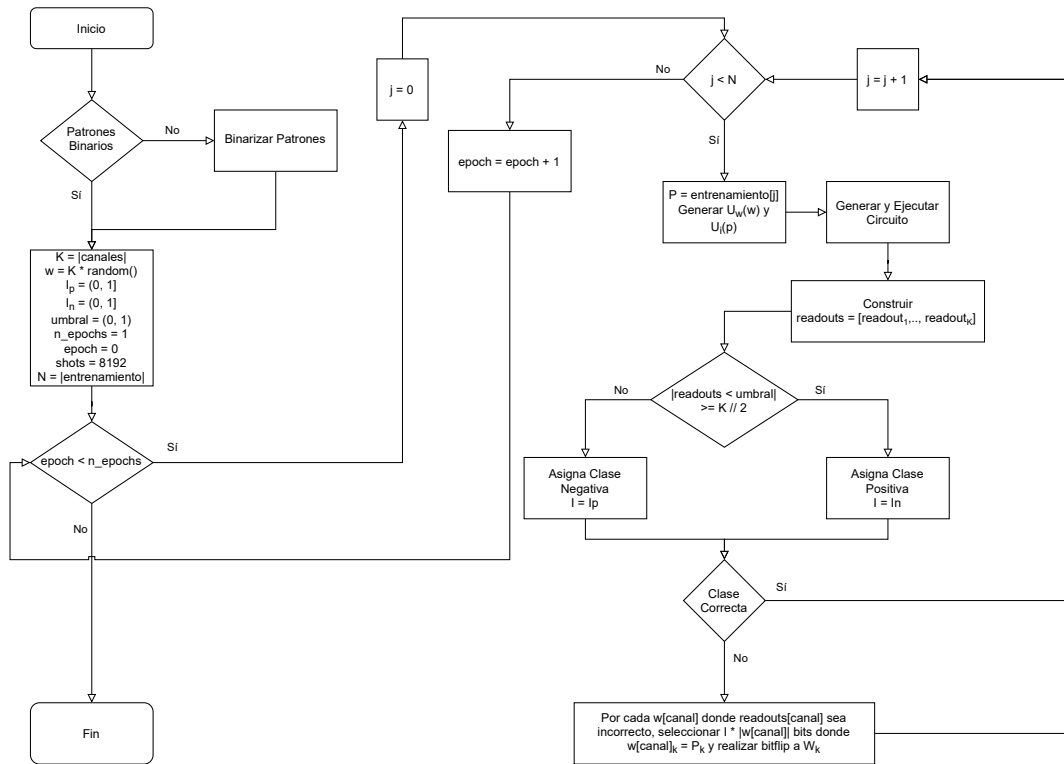


Figura 4.7 Diagrama de flujo para el proceso de entrenamiento para imágenes multicanal.

1 vector de pesos  $i$  que haya clasificado erróneamente su canal de acuerdo con la clase global  
 2 se actualizará dicho vector de acuerdo con el método descrito en la Sección 4.1.2. Diagramas  
 3 de flujo tanto para el entrenamiento como para la validación se pueden ver en la Figura 4.7 y  
 4 Figura 4.8, respectivamente.

5 En general cada canal puede tener su propio conjunto de hiperparámetros, tales como el  
 6 umbral de clasificación, las tasas de aprendizaje  $I_p$  e  $I_n$ .

7 Si bien esta extensión se propone para analizar imágenes a color, es evidente que puede ser  
 8 usado para procesar varias imágenes binarias en paralelo. Si se usara de esta forma, se podría  
 9 entrenar por lotes, por ejemplo, realizando XOR a todas las imágenes mal clasificadas y utilizar  
 10 la imagen resultante para realizar la actualización del vector de pesos.

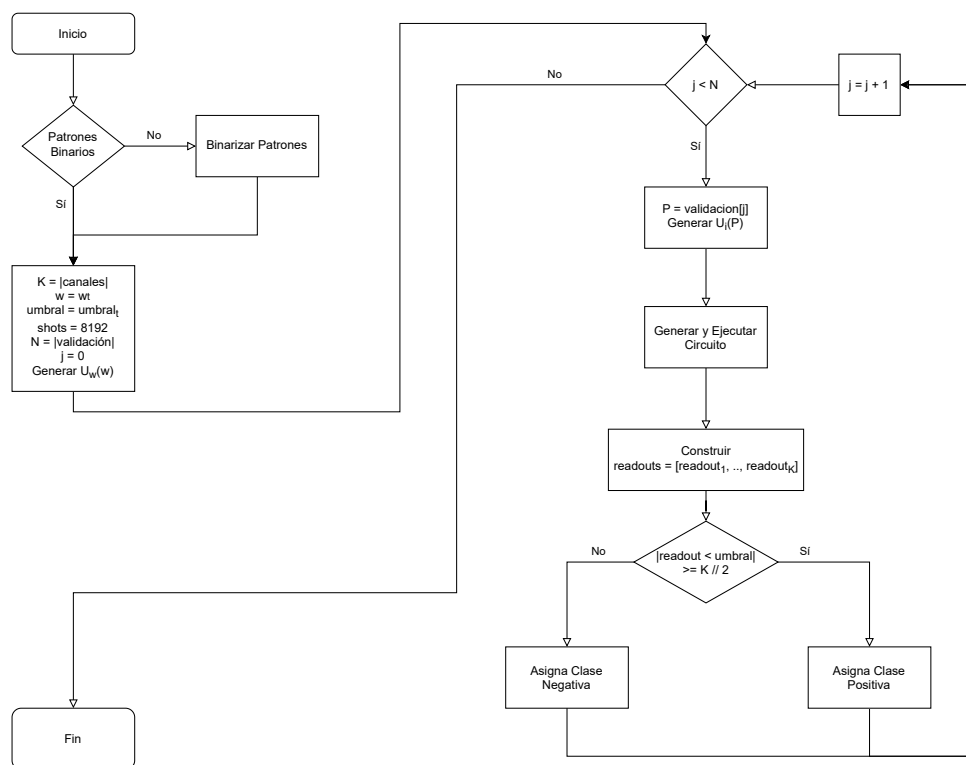


Figura 4.8 Diagrama de flujo para el proceso de validación para imágenes multicanal.

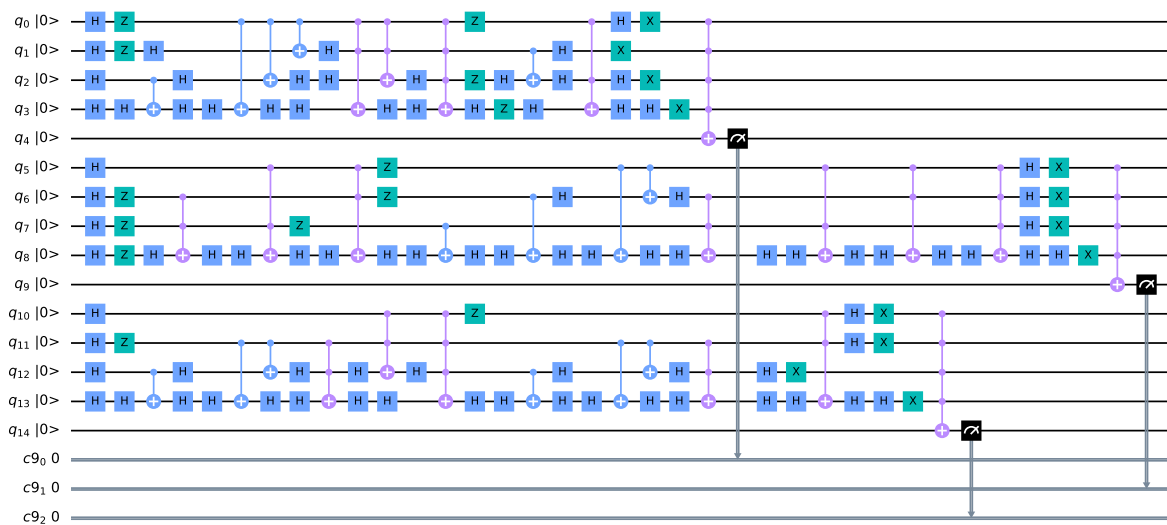


Figura 4.9 Circuito Cuántico Extendido. Se forma colocando un circuito por cada canal, uno *encima* del otro, apilados verticalmente.



## Capítulo 5

1

## Resultados y discusión

2

En este capítulo se presentan los resultados obtenidos al realizar la clasificación de los bancos de datos descritos en la Sección 3.3 usando el modelo descrito en el Capítulo 4. Comenzamos con una introducción sobre los métodos de validación empleados en este trabajo, y continuamos con una sección dedicada a las medidas de desempeño utilizadas para obtener una valoración objetiva del modelo. Se finaliza mostrando los resultados obtenidos por el modelo al enfrentarse a distintos problemas de clasificación.

3

4

5

6

7

8

### 5.1. Métodos de validación

9

Una parte importante de los modelos en el aprendizaje de máquina, y por extensión en su forma cuántica, es la validación de estos modelos. Esto nos permite tener una apreciación sin sesgo de lo bien, o mal, que se comporta un modelo sobre cierto banco de datos.

10

11

12

En este trabajo se empleó el método de validación conocido como *Holdout* y también se utilizó *Resubstitution Error* como métrica de evaluación. En el caso de *Holdout* el banco de datos es particionado en dos conjuntos con intersección nula para evitar filtrado de información. En el caso de *Resubstitution Error* se emplea el mismo conjunto de datos tanto para la fase de entrenamiento como para la fase de prueba.

13

14

15

16

17

Si bien estos dos métodos no son la mejor opción para validar un modelo con el objetivo de ser aplicado en la industria, fueron seleccionados porque son los menos costosos computacionalmente y porque se busca una *metavalidación*, es decir, se busca evaluar la veracidad del modelo y no la comparación con el estado del arte.

18

19

20

21

Cuadro 5.1 Matriz de confusión para un banco de datos con dos clases.

Predicted Class	Real Class	
	Positive	Negative
	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Una vez lograda la *metavalidación* con los primeros dos bancos de datos, el tercero fue validado con el método conocido como *Repeated k-Fold Cross Validation (RkFCV)*. En el modelo *k-Fold Cross Validation (kFCV)*, el banco de datos completo se particiona en  $k$  partes iguales, una de ellas se utiliza como conjunto de prueba y el  $k - 1$  restante es empleado como conjunto de entrenamiento, realizando este proceso  $k$  veces de tal manera que cada una de las particiones actúe como conjunto de prueba. En el caso del *RkFCV*, el método *kFCV* es repetido un número arbitrario de veces, y en cada una de ellas se genera una partición nueva.

## 5.2. Medidas de desempeño

Si en un banco de datos cada clase contiene aproximadamente el mismo número de instancias por clase se le conoce como un banco de datos balanceado. Cuando esto sucede, la mayoría de las medidas de desempeño son adecuadas, siempre que no haya ningún sesgo hacia ninguna clase. Sin embargo, dependiendo de la aplicación que se le vaya a dar al clasificador, o de la relevancia de alguna de las clases, alguna medida de desempeño puede ser preferida.

La medida más usual es la de *accuracy*, la cual mide la proporción de instancias correctamente clasificadas con respecto al número total de instancias. Cuando la métrica de evaluación empleada es Resubstitution Error al *accuracy* se le conocerá también por este nombre, es decir, *Resubstitution Error*.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5.1)$$

Un banco de datos es desbalanceado cuando una o más clases está escasamente representada. La mayoría de las medidas de desempeño producen un sesgo hacia la clase mayoritaria cuando se evalúa un problema de clasificación con clases desbalanceadas. En este caso, el *True Positive Rate (TPR)*

$$TPR = \frac{TP}{TP + FN} . \quad (5.2) \quad 1$$

a la cual también se le conoce como *Recall* o *Sensibilidad* (*Sensitivity*) puede ser usada para medir la razón entre el número de instancias positivas correctamente clasificadas y el número total de instancias positivas. 2 3 4

La *Precisión* (*Precision*) o también conocida como *Positive Predictive Value* (*PPV*) 5

$$PPV = \frac{TP}{TP + FP} . \quad (5.3) \quad 6$$

mide la razón entre el numero de instancias positivas correctamente clasificadas y el número total de instancias clasificadas como instancias positivas. 7 8

Con el *TPR* y *PPV* se puede obtener la medida *F1*, la cual es su media armónica: 9

$$F1 = 2 * \frac{PPV * TPR}{PPV + TPR} = \frac{2 * TP}{2 * TP + FP + FN} . \quad (5.4) \quad 10$$

### 5.3. Resultados 11

En esta sección se presentan los resultados experimentales al tratar de clasificar ambos bancos de datos descritos en la Sección 3.3, *Digits* y *Semeion*, empleando el modelo descrito previamente en el Capítulo 4. 12 13 14

Para el banco de datos *Digits* se utilizaron *Resubstitution Error* y *Holdout* como métodos de evaluación y validación. En el caso del *Holdout* se utilizó la partición propuesta por los autores en [40]. Para el caso de *Resubstitution Error* solo el conjunto de entrenamiento fue utilizado, tanto en la fase de entrenamiento como en la de prueba. 15 16 17 18

El banco de datos *Digits* requiere un procesamiento previo para poder ser alimentado al modelo cuántico, ya que los atributos  $x_i$  de cada patrón cumplen  $x_i \in [0, 16]$  y el modelo solo trabaja con valores binarios, por lo que un umbralado es necesario. Dicho umbralado fue aplicado de la siguiente forma: 19 20 21 22

$$\text{píxel binarizado} = \begin{cases} 0, & \text{si el valor original} < 10 \\ 1, & \text{otro caso} \end{cases} \quad (5.5) \quad 23$$

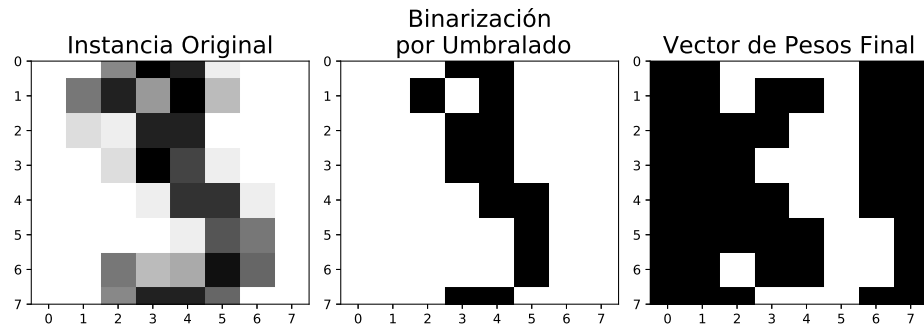


Figura 5.1 A la izquierda se muestra una instancia original perteneciente al Digits Dataset. La imagen del centro muestra el resultado después de la binarización por umbralado. A la derecha se muestra el vector de pesos ya entrenado para el clasificador binario 0/3. El vector de pesos tiende a tomar la forma de una especie de *máscara* que asemeja a una de las clases.

1 El umbral en la Ec. (5.5) es un nuevo hiperparámetro que puede ser optimizado. Como  
 2 se puede observar en la Figura 5.1, este umbral determina cuánta información original es  
 3 preservada. De forma visual las imágenes claramente se ven deformadas.

4 Para el banco *Semeion*, se utilizó *Resubstitution Error* debido al relativamente pequeño  
 5 número de instancias. Este banco de datos, al ser de atributos binarios, no requiere de procesa-  
 6 miento previo.

7 Para cada banco de datos, se entrenaron dos tipos de clasificadores binarios: *clase vs clase*,  
 8 que en este caso representa un problema de clases balanceadas, y *clase vs complemento* el cual  
 9 representa un problema de clases altamente desbalanceadas. Los resultados son condensados en  
 10 varias tablas.

11 En la Tabla 5.2 la diagonal principal representa la clasificación trivial de una sola clase. Es  
 12 interesante notar que, aunque se estaría tentado a utilizar solo la mitad superior, o inferior, para  
 13 clasificar el problema inverso, es decir, usar el clasificador ya entrenado sobre el problema de  
 14 clases *positiva/negativa* para tratar de resolver el problema *negativa/positiva*, no se obtuvieron  
 15 buenos resultados con esta estrategia, por lo que se entrenó el conjunto completo de clasifica-  
 16 dores. La imposibilidad de utilizar el mismo clasificador para resolver el problema reflexivo  
 17 se vuelve evidente cuando se estudia el vector de pesos que arroja cada clasificador, ya que  
 18 este tiende a tomar la forma de una especie de máscara que se asemeja a una de las clases,  
 19 en este caso a la clase negativa. Lo anterior puede verse más claramente en la Figura 5.2 y

Figura 5.3 donde los vectores de pesos toman una estructura similar a una de las clases. Esto es 1  
más notorio en el caso del banco de datos Semeion debido a que su dimensionalidad es mayor. 2

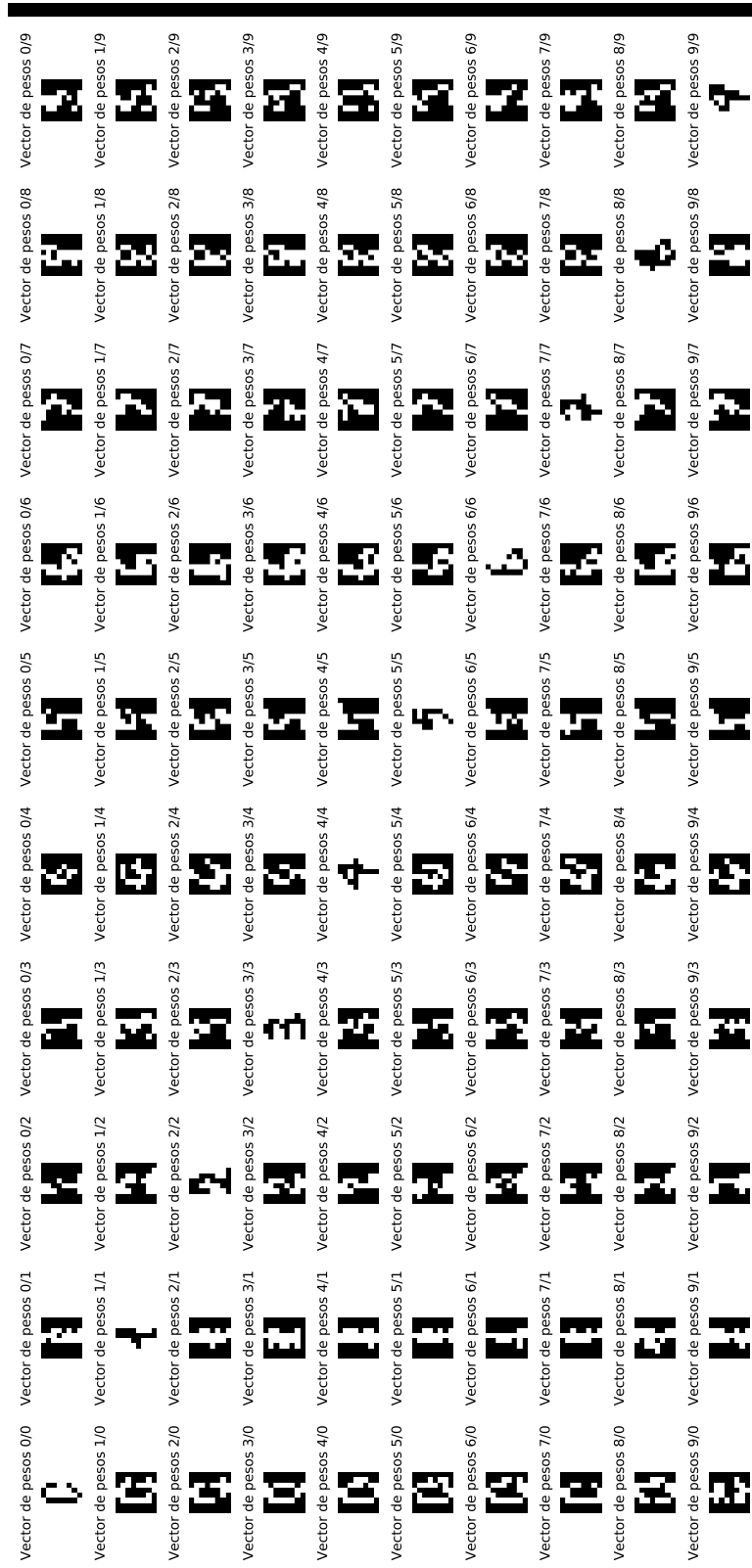


Figura 5.2 Conjunto completo de vectores de pesos para el banco de datos Digits. Cada uno de ellos es el resultado del entrenamiento del modelo cuántico como resultado de una clasificación binaria entre cada par de clases.

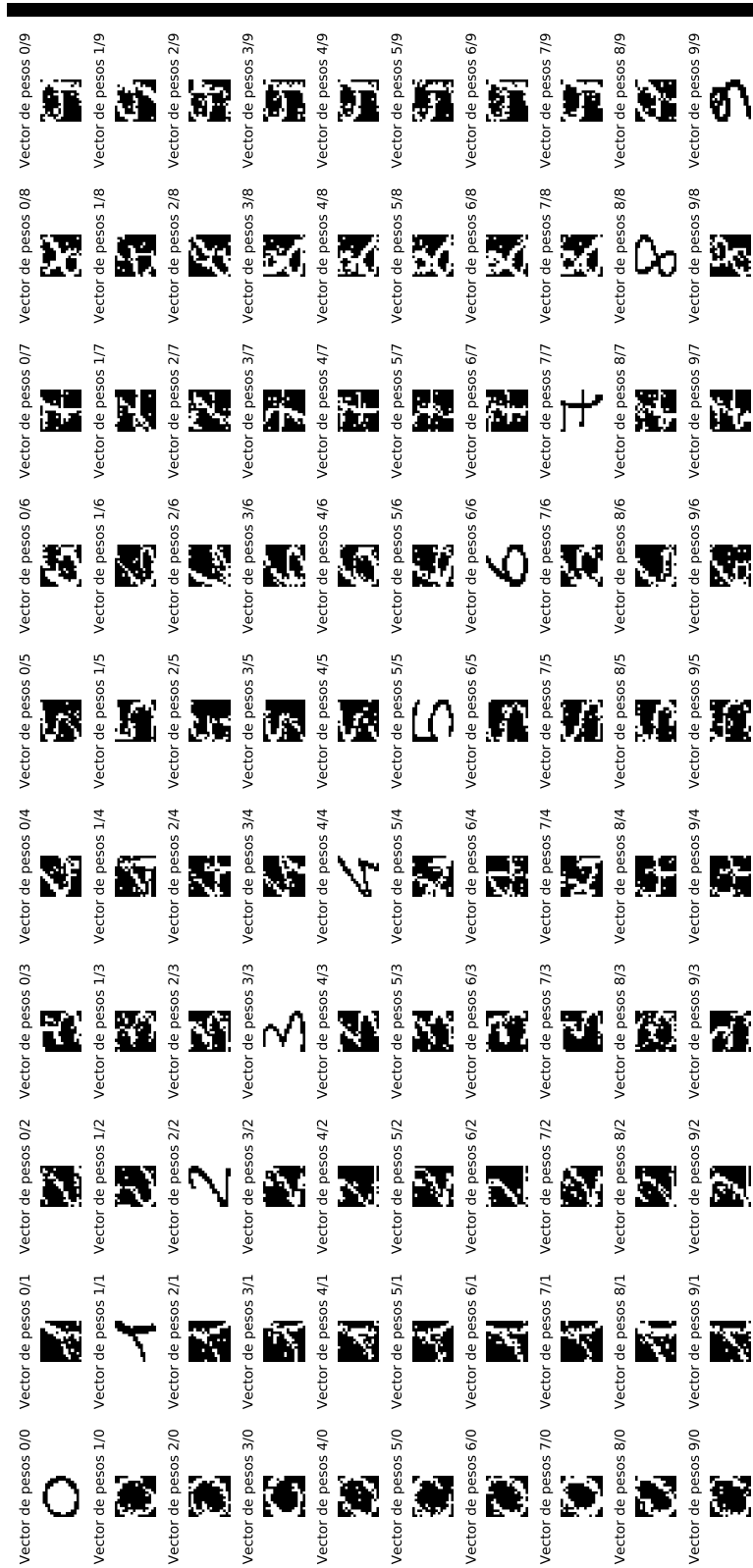


Figura 5.3 Conjunto completo de vectores de pesos para el banco de datos Semeion. Cada uno de ellos es el resultado del entrenamiento del modelo cuántico como resultado de una clasificación binaria entre cada par de clases.

Cuadro 5.2 Resubstitution Error en clases balanceadas.

Resubstitution Error en Digits										
Clase Negativa										
	0	1	2	3	4	5	6	7	8	9
Clase Positiva										
0	1.0	0.725	0.794	0.714	0.671	0.769	0.353	0.834	0.678	0.589
1	0.377	1.0	0.660	0.849	0.749	0.780	0.898	0.833	0.609	0.759
2	0.833	0.710	1.0	0.730	0.804	0.894	0.631	0.764	0.564	0.820
3	0.933	0.753	0.833	1.0	0.780	0.887	0.807	0.676	0.624	0.652
4	0.974	0.567	0.837	0.758	1.0	0.785	0.792	0.780	0.749	0.764
5	0.886	0.785	0.830	0.715	0.779	1.0	0.953	0.695	0.800	0.640
6	0.635	0.807	0.843	0.807	0.682	0.785	1.0	0.883	0.540	0.828
7	0.983	0.653	0.803	0.776	0.875	0.797	0.905	1.0	0.728	0.832
8	0.752	0.480	0.706	0.792	0.783	0.651	0.830	0.597	1.0	0.666
9	0.564	0.812	0.815	0.696	0.883	0.527	0.678	0.813	0.788	1.0

En la Tabla 5.3 se mantiene un registro de varias medidas de desempeño, incluyendo el Área Bajo la Curva (por sus siglas en inglés Area Under the Curve AUC). Los resultados del *Recall* indican que el modelo cuántico puede distinguir la clase positiva del resto de instancias de forma adecuada. La proporción entre la clase minoritaria y el resto de las instancias es de aproximadamente 1:100. Este resultado es prometedor en las áreas médicas, donde los bancos de datos son generalmente altamente desbalanceados.

En la Tabla 5.4 se muestra el incremento en el accuracy al cambiar el método de validación. Es notable el poder latente de generalización, ya que en este caso la evaluación fue realizada sobre un conjunto de instancias que el modelo no había visto previamente durante la etapa de entrenamiento.

En la Tabla 5.5 se muestran las métricas usuales, donde es notorio el beneficio obtenido al utilizar *Holdout* en este modelo sobre este banco de datos en especial. Se registró una mejora en siete de las diez clases totales. Este método de validación otorga mayor seguridad sobre el poder de generalización que puede tener este modelo de clasificación.

En la Tabla 5.6 y en la Tabla 5.7 se muestran las ya conocidas métricas de desempeño. Se observa un desempeño aceptable en casi todas las tareas de clasificación, pero es notable la disminución al compararse con el banco de datos *Digits*. Esta caída en el desempeño podría ser explicada por el hecho de que estas instancias yacen en un espacio de dimensionalidad mayor, y por tanto el circuito cuántico necesario para procesar cada patrón es también más grande



Cuadro 5.3 Resubstitution Error en clases altamente desbalanceadas.

Resubstitution Error en Digits				
Clase Positiva	Recall	Precision	F1	AUC
0	0.983	0.122	0.218	0.605
1	0.857	0.129	0.224	0.602
2	0.966	0.133	0.234	0.639
3	0.814	0.112	0.198	0.543
4	0.955	0.143	0.249	0.657
5	0.873	0.123	0.216	0.587
6	0.845	0.121	0.212	0.580
7	0.837	0.128	0.222	0.604
8	0.724	0.107	0.186	0.538
9	0.861	0.132	0.229	0.615

Cuadro 5.4 Accuracy en clases balanceadas.

Digits validados por Holdout										
Clase Negativa										
	0	1	2	3	4	5	6	7	8	9
Clase Positiva										
0	1.0	0.844	0.833	0.900	0.896	0.833	0.924	0.831	0.732	0.731
1	0.969	1.0	0.852	0.863	0.818	0.826	0.914	0.853	0.775	0.779
2	0.895	0.824	1.0	0.841	0.810	0.860	0.877	0.828	0.740	0.843
3	0.883	0.819	0.730	1.0	0.920	0.854	0.964	0.823	0.851	0.719
4	0.933	0.815	0.899	0.840	1.0	0.876	0.790	0.819	0.819	0.811
5	0.933	0.892	0.793	0.802	0.856	1.0	0.964	0.872	0.811	0.765
6	0.827	0.809	0.843	0.925	0.825	0.953	1.0	0.902	0.839	0.739
7	0.974	0.878	0.876	0.859	0.794	0.869	0.977	1.0	0.813	0.871
8	0.889	0.682	0.860	0.826	0.814	0.823	0.842	0.804	1.0	0.824
9	0.849	0.850	0.907	0.746	0.822	0.850	0.955	0.835	0.836	1.0

Cuadro 5.5 Accuracy en clases altamente desbalanceadas.

Digits validados por Holdout				
Positive Class	Recall	Precision	F1	AUC
0	0.955	0.107	0.192	0.539
1	0.972	0.112	0.201	0.552
2	0.858	0.098	0.176	0.500
3	0.803	0.093	0.167	0.460
4	0.994	0.108	0.195	0.539
5	0.950	0.114	0.204	0.560
6	1.0	0.106	0.192	0.529
7	0.843	0.116	0.204	0.567
8	0.965	0.105	0.190	0.544
9	0.966	0.107	0.193	0.536

1 y complejo. Sin embargo, no se debe olvidar que, aunque ambos bancos de datos parezcan  
 2 similares, son en realidad distintos y no se debería esperar un desempeño similar.

3 Para el banco de datos *Fruits* es necesario aplicar un procesamiento previo para que cada  
 4 canal sea binario. En este caso se aplicó un umbralado similar al mostrado en la Ec. (5.5).  
 5 En general, este umbralado podría aplicarse independientemente a cada canal, con un umbral  
 6 diferente para cada uno de los canales. Para este trabajo el umbral fue el mismo para los tres  
 7 canales, cuyo valor se fijó en 120. Aparte del umbralado, fue necesario aplicar un escalamiento  
 8 a cada imagen, para reducir las dimensiones originales de  $100 \times 100$  a  $16 \times 16$ . Debido a  
 9 que las técnicas de escalamiento están fuera del alcance de este trabajo, se utilizó el modo  
 10 *INTER\_AREA* en la función *resize* ofrecida por *OpenCV*.

11 El modelo descrito en la Sección 4.2 fue aplicado al subconjunto de este banco de datos,  
 12 validándolo a través del método *RkFCV*, realizando 5 repeticiones con 5 fold cada una, para un  
 13 total de 25 folds.

14 Al finalizar cada fold se obtiene un conjunto de vectores de pesos, uno por cada canal. En la  
 15 Figura 5.4 se ve a detalle la imagen global y cada uno de los tres canales por separado.

Cuadro 5.6 Resubstitution Error en clases balanceadas.

Resubstitution Error en Semeion										
Clase Negativa										
	0	1	2	3	4	5	6	7	8	9
Clase Positiva										
0	1.0	0.851	0.806	0.890	0.788	0.840	0.593	0.843	0.835	0.680
1	0.947	1.0	0.728	0.869	0.352	0.763	0.832	0.700	0.608	0.650
2	0.937	0.707	1.0	0.767	0.793	0.707	0.706	0.716	0.601	0.637
3	0.912	0.797	0.672	1.0	0.759	0.757	0.853	0.712	0.525	0.624
4	0.981	0.801	0.793	0.837	1.0	0.793	0.751	0.749	0.515	0.664
5	0.890	0.794	0.672	0.657	0.693	1.0	0.721	0.703	0.528	0.570
6	0.757	0.842	0.800	0.881	0.636	0.809	1.0	0.805	0.506	0.692
7	0.946	0.734	0.624	0.779	0.702	0.722	0.724	1.0	0.530	0.667
8	0.854	0.728	0.786	0.866	0.655	0.671	0.781	0.683	1.0	0.610
9	0.905	0.809	0.763	0.769	0.689	0.611	0.780	0.737	0.607	1.0

Cuadro 5.7 Resubstitution Error en clases altamente desbalanceadas.

Resubstitution Error en Semeion				
Clase Positiva	Recall	Precision	F1	AUC
0	0.770	0.090	0.161	0.449
1	0.962	0.113	0.202	0.555
2	0.849	0.1	0.178	0.500
3	0.937	0.106	0.191	0.534
4	0.807	0.092	0.166	0.460
5	0.937	0.105	0.189	0.528
6	0.925	0.105	0.189	0.520
7	0.873	0.102	0.183	0.516
8	0.864	0.097	0.175	0.502
9	0.873	0.100	0.180	0.506

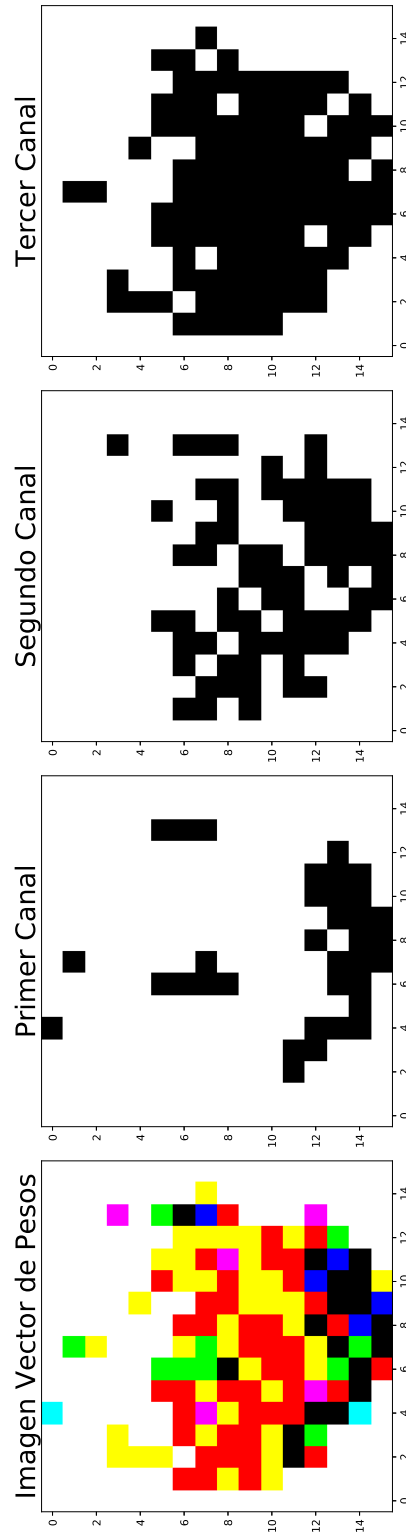


Figura 5.4 A la izquierda se muestra la imagen global conformada por los vectores de pesos de cada canal. También se muestra de forma independiente el vector de pesos correspondiente a cada canal.

Cuadro 5.8 Promedio y desviación estándar de las medidas de desempeño obtenidas después de la validación por el método *RkFCV* con 5 repeticiones y 5 folds cada repetición.

Método de Validación RkFCV					
	Accuracy	Accuracy Validación	Precision	Recall	F1
Promedio	0.856	0.924	0.996	0.859	0.922
Desviación Estándar	0.026	0.015	0.009	0.028	0.016

En la Tabla 5.8 se muestran los promedios y las desviaciones estándar de las medidas de desempeño y en la Figura 5.5 se muestran los valores de *accuracy* por cada fold y el promedio, tanto en el conjunto de entrenamiento como en el de validación o prueba. En la Figura 5.6 se muestran las medidas *precision*, *recall* y *F1*, evaluadas en el conjunto de validación.

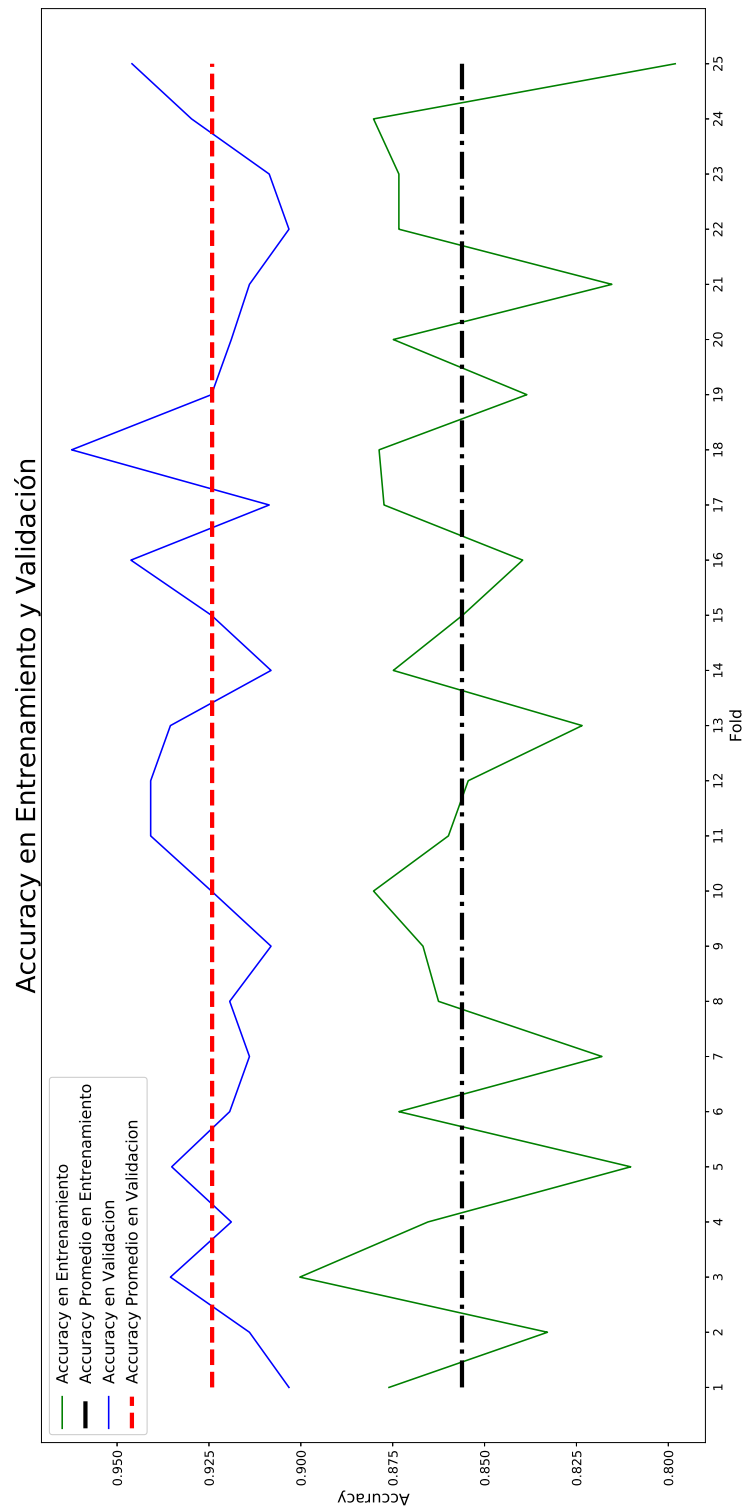


Figura 5.5 Valores de *accuracy* en cada fold, para el conjunto de entrenamiento y validación. Se muestra también el promedio obtenido a través del método de validación.

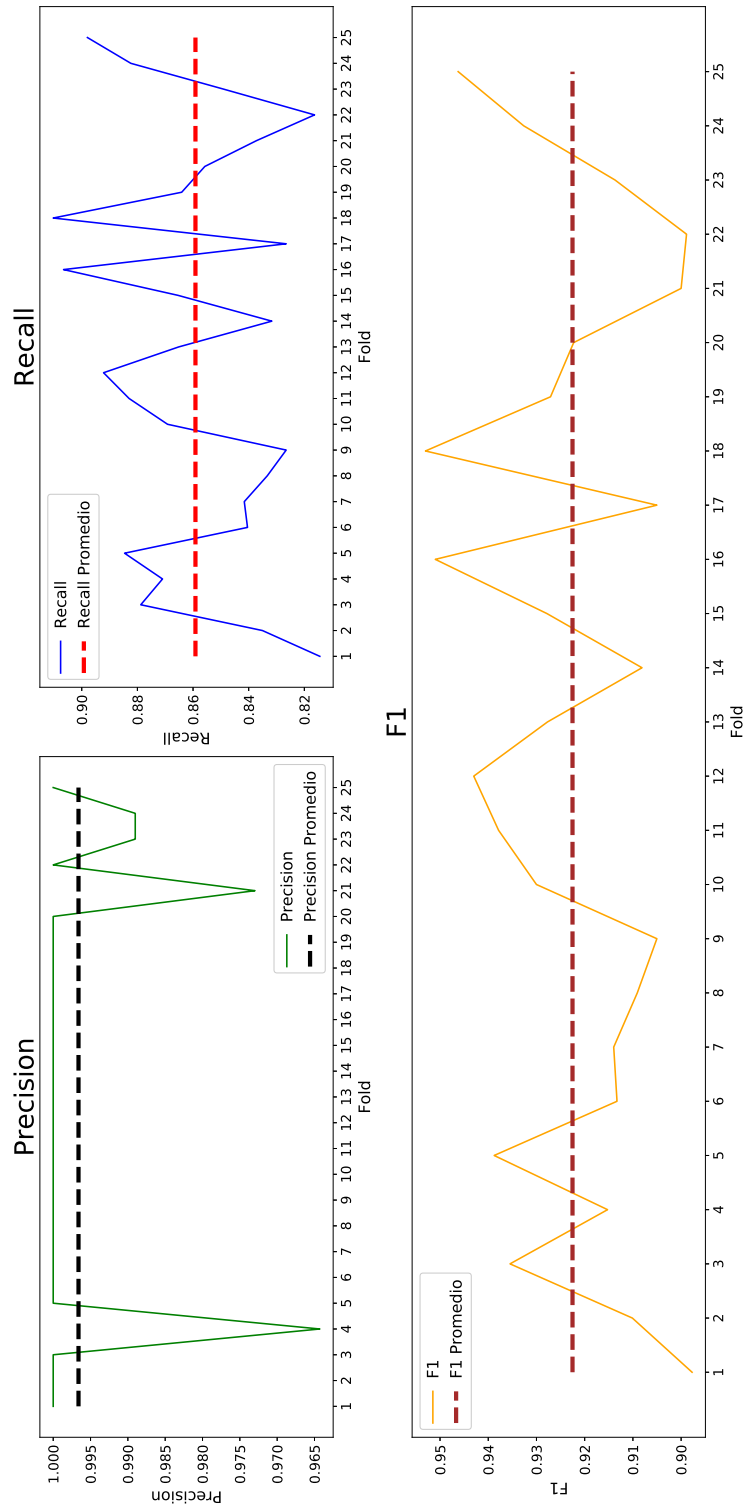


Figura 5.6 Valores de *precision*, *recall* y *F1* en cada fold para el conjunto de validación. Se muestra también el promedio obtenido a través del método de validación.

1 Es relevante señalar que tanto en los bancos de datos con imágenes binarias como en banco  
2 *Fruits*, la mayoría de los modelos convergieron, a las métricas señaladas, en la primera iteración  
3 o *epoch*, con algunos pocos requiriendo un par de iteraciones extra.

4 Es más interesante aún, notar que la métrica *Recall* es muy buena en la mayoría de las  
5 evaluaciones. Desde un punto de vista médico, esto nos indica que el modelo identifica con  
6 alta veracidad a los patrones pertenecientes a la clase minoritaria, la cual comúnmente coincide  
7 con la de la enfermedad a identificar. Sin embargo, al estudiar la métrica *Precision*, se nota  
8 que esta es baja, por lo que en conjunto con el *Recall* indican que el modelo trata de clasificar  
9 la mayoría de las instancias de la clase minoritaria, o la enfermedad a identificar, a expensas  
10 de sobreestimar el número de enfermos. En otras palabras, el modelo es propenso a dar falsos  
11 positivos que falsos negativos, lo cual es preferible en el caso de una enfermedad mortal o muy  
12 contagiosa.



## Capítulo 6

1

## Conclusiones y trabajo futuro

2

En este capítulo se presentan las conclusiones del presente trabajo de investigación y se mencionan algunos posibles trabajos que pueden emerger como consecuencia o extensión de este.

3

4

5

En la presente tesis se presentó la implementación de un modelo cuántico para la clasificación binaria de imágenes binarias y se propuso la extensión de este para poder procesar imágenes de un número arbitrario de canales, en este caso se trabajó con tres canales.

6

7

8

Los beneficios de los modelos cuánticos sobre su contraparte clásica es que los primeros permiten codificar y procesar grandes cantidades de información con ayuda del entrelazamiento y superposición de los estados cuánticos, la cantidad de qubits necesarios para el procesamiento es exponencialmente menor que los bits clásicos y que la ejecución es de forma paralela y optimizada por el modelo de circuito cuántico.

9

10

11

12

13

Una desventaja es que los modelos propuestos solo trabajan con patrones con atributos binarios.

14

15

Los resultados experimentales muestran que los modelos cuánticos propuestos tienen un buen rendimiento en problemas de clasificación binaria, tanto en problemas de clases balanceadas como en problemas de clases altamente desbalanceadas. Son alentadores estos resultados, y en especial en los problemas de clases altamente desbalanceadas, ya que esto podría significar una aplicación en bancos de datos médicos donde esta situación es predominante. Esto último debido al buen resultado en la métrica *Recall* ya que esto indica que el modelo es bueno en reconocer a la clase minoritaria con alta precisión.

16

17

18

19

20

21

22

Otra ventaja, discutible, es que la mayoría de los clasificadores entrenados en este trabajo convergieron a los resultados mostrados en una sola observación completa del banco de datos.

23

24

Durante el desarrollo de esta tesis se cumplieron los objetivos esperados:

25

- 1 ■ Se mostró que el Aprendizaje de Máquina Cuántico puede ser aplicado para la clasifica-  
2 ción de patrones, y en especial, para la clasificación de imágenes.
- 3 ■ Se implementó un clasificador cuántico que procesa patrones binarios, y en especial  
4 imágenes binarias.
- 5 ■ Se clasificaron problemas biclase en bancos de datos de imágenes binarias, tanto proble-  
6 mas de clases balanceadas como altamente desbalanceadas.
- 7 ■ Se propuso un modelo cuántico para procesar imágenes de un número arbitrario de  
8 canales, y en especial el caso de tres canales.
- 9 ■ Se implementó el modelo propuesto para clasificar un banco de datos de imágenes de tres  
10 canales en un problema biclase.
- 11 ■ Se demostró la validez del modelo multicanal clasificando un banco de datos biclase,  
12 balanceado y con imágenes de tres canales.

13 Como trabajo futuro se propone la extensión los modelos para poder realizar la clasifica-  
14 ción multiclase. Se propone explotar el modelo multicanal para implementar un proceso de  
15 entrenamiento por lotes, lo cual propiciaría un incremento aún mayor en la velocidad de proce-  
16 samiento. También se propone aplicar los modelos en un banco de datos médicos. Finalmente  
17 se propone desarrollar e implementar un modelo de convolución cuántico, lo cual acerque más  
18 el aprendizaje de máquina cuántico a los modelos clásicos del estado del arte.

# Bibliografía

- [1] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, “Deformation models for image recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1422–1435, 2007.
- [2] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [4] D. Ignatov and A. Ignatov, “Decision stream: Cultivating deep decision trees,” *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, vol. 2017-Novem, no. September, pp. 905–912, 2018.
- [5] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2003-Janua, pp. 958–963, 2003.
- [6] J. Shi, Z. Li, T. Zhu, D. Wang, and C. Ni, “Defect detection of industry wood veneer based on NAS and multi-channel mask R-CNN,” *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–17, 2020.
- [7] M. Mishra, J. Nayak, B. Naik, and A. Abraham, “Deep learning in electrical utility industry: A comprehensive review of a decade of research,” *Engineering Applications of Artificial Intelligence*, vol. 96, no. December 2019, p. 104000, 2020.
- [8] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, no. December 2012, pp. 60–88, 2017.
- [9] D. Ueda, A. Shimazaki, and Y. Miki, “Technical and clinical overview of deep learning in radiology,” *Japanese Journal of Radiology*, vol. 37, no. 1, pp. 15–33, 2019.
- [10] S. Zhang, S. M. H. Bamakan, Q. Qu, and S. Li, “Learning for Personalized Medicine: A Comprehensive Review from a Deep Learning Perspective,” *IEEE Reviews in Biomedical Engineering*, vol. 12, pp. 194–208, 2018.

- 
- 1 [11] P. W. Frey and D. J. Slate, “Letter Recognition Using Holland-Style Adaptive Classifiers,”  
2 *Machine Learning*, vol. 6, no. 2, pp. 161–182, 1991.
- 3 [12] H. Abraham, AduOffei, R. Agarwal, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander,  
4 M. Amy, E. Arbel, Arijit02, A. Asfaw, *et al.*, “Qiskit: An open-source framework for  
5 quantum computing,” 2019.
- 6 [13] S. K. Jeswal and S. Chakraverty, “Recent Developments and Applications in Quantum  
7 Neural Network : A Review,” *Archives of Computational Methods in Engineering*, vol. 26,  
8 no. 4, pp. 793–807, 2019.
- 9 [14] M. Schuld, I. Sinayskiy, and F. Petruccione, “Simulating a perceptron on a quantum  
10 computer,” *Physics Letters A*, vol. 379, no. 7, pp. 660–663, 2015.
- 11 [15] S. Piat, N. Usher, S. Severini, M. Herbster, T. Mansi, and P. Mountney, “Image classifica-  
12 tion with quantum pre-training and auto-encoders,” vol. 16, no. 8, pp. 1–14, 2018.
- 13 [16] A. Y. Yamamoto, K. M. Sundqvist, P. Li, and H. R. Harris, “Perceptron,” *Quantum*  
14 *Information Processing*, vol. 17, no. 6, pp. 1–12, 2018.
- 15 [17] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quantvolutional Neural Networks:  
16 Powering Image Recognition with Quantum Circuits,”
- 17 [18] F. Tacchino, “An artificial neuron implemented on an actual quantum processor,” *npj*  
18 *Quantum Information*, no. March, pp. 1–8, 2019.
- 19 [19] M. Rossi, M. Huber, D. Bruß, and C. Macchiavello, “Quantum hypergraph states Quantum  
20 hypergraph states,” no. 15, 2013.
- 21 [20] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- 22 [21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. USA:  
23 Wiley-Interscience.
- 24 [22] F. Rosenblatt, “A Perceiving and Recognizing Automaton,” Tech. Rep. 85-460-1, Cornell  
25 Aeronautical Laboratory, 01 1957.
- 26 [23] M. Minsky and S. Papert, *Perceptrons*. M.I.T Press., 1969.
- 27 [24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-  
28 propagating errors,” *Nature*, vol. 323, pp. 533–536, Oct 1986.
- 29 [25] H. J. Kelly, “Gradient Theory of Optimal Flight Paths,” *ARS Journal*, vol. 30, no. 10,  
30 pp. 947–954, 1960.
- 31 [26] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *2017*  
32 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807,  
33 2017.

- [27] K. H., X. Z., S. R., and J. S., “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [28] S. K. and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [29] S. C., V. V., I. S., S. J., and W. Z., “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun 2016.
- [30] E. Grumbling and M. Horowitz, *Quantum Computing: Progress and Prospects*. Washington, DC: The National Academies Press, 2019.
- [31] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, Jun 1982.
- [32] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing*. OUP Oxford, 2007.
- [33] P. A. M. Dirac, *The Principles of Quantum Mechanics*. OUP Oxford, 1947.
- [34] P. H. Eberhard and R. R. Ross, “Quantum field theory cannot provide faster-than-light communication,” *Foundations of Physics Letters*, vol. 2, May 1988.
- [35] G. C. Ghirardi, R. Grassi, A. Rimini, and T. Weber, “Experiments of the EPR type involving CP -violation do not allow faster-than-light communication between distant observers,” *Europhysics Letters (EPL)*, vol. 6, pp. 95–100, may 1988.
- [36] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, “Elementary gates for quantum computation,” *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov 1995.
- [37] M. Mottonen and J. J. Vartiainen, “Decompositions of general quantum gates,” *arXiv preprint quant-ph/0504100*, 2005.
- [38] A. Y. Kiatev, “Quantum computations: algorithms and error correction,” *Russian Mathematical Surveys*, vol. 52, pp. 1191–1249, 1997.
- [39] C. M. Dawson and M. A. Nielsen, “The solovay-kitaev algorithm,” *arXiv preprint quant-ph/0505030v2*, 2005.
- [40] E. Alpaydin and C. Kaynak, “Cascading classifiers,” *Kybernetika*, vol. 34, no. 4, pp. 369–374, 1998.
- [41] M. Buscema, “Metanet\*: The theory of independent judges,” *Substance Use & Misuse*, vol. 33, no. 2, pp. 439–461, 1998.
- [42] “Semeion handwritten digit data set.” <https://archive.ics.uci.edu/ml/datasets/semeion+handwritten+digit>. [En línea; visitado 02-Noviembre-2020].

- 
- <sup>1</sup> [43] V. García, J. Sánchez, and R. Mollineda, “On the effectiveness of preprocessing methods  
<sup>2</sup> when dealing with different levels of class imbalance,” *Knowledge-Based Systems*, vol. 25,  
<sup>3</sup> no. 1, pp. 13 – 21, 2012. Special Issue on New Trends in Data Mining.
- <sup>4</sup> [44] E. Zhang, “Fruit recognition.” [https://www.kaggle.com/sshikamaru/fruit-recognition/](https://www.kaggle.com/sshikamaru/fruit-recognition/metadata)  
<sup>5</sup> metadata, 2020. [En línea; visitado 19-Noviembre-2020].