



UNIVERSIDAD DE GUANAJUATO
DIVISIÓN DE CIENCIAS E INGENIERÍAS, CAMPUS LEÓN

**Determinación de la carga topológica de
vórtices ópticos perturbados utilizando redes
neuronales convolucionales**

T E S I S

Que para obtener el reconocimiento de trabajo laureado en
Licenciatura en Ingeniería Física

P R E S E N T A

Jesús Adrián Mejía Contreras

Directores de tesis

Dr. Fernando Arce Vega

Dr. Carmelo Rosales Guzmán

Junio 2025

*Para mi yo joven,
por siempre creer en la frase “intentar, fallar e iterar”*



Agradecimientos

A mi madre, por su amor infinito, su cuidado constante, su paciencia y su confianza en mí. A mi padre, por su incondicional apoyo, cariño, y por guiarme a ser la persona que soy hoy en día. A mis hermanas, cuya compañía, consejos y risas han sido un refugio y una fuente de plenitud.

A mis amigos y compañeros, quienes, con sus pláticas, consejos y experiencias, me han enseñado valiosas lecciones, permitiéndome conocerme mejor y entender un poco más de este viaje llamado *vida*.

A mis profesores de la División de Ciencias e Ingenierías, quienes, a través de sus enseñanzas, han formado en mí una pasión por el conocimiento y la ciencia.

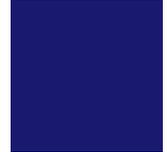
Quiero expresar un especial agradecimiento al Dr. Fernando Arce Vega, por brindarme la oportunidad de trabajar a su lado, por el aprendizaje invaluable que he adquirido a lo largo del proyecto, por su constante apoyo, asesoría y paciencia, recibíendome siempre con una sonrisa y una enorme disposición y amabilidad al resolver mis dudas.

Asimismo, agradezco profundamente al Dr. Carmelo Rosales Guzmán por la confianza depositada en mí y en el proyecto. Su dedicación, amabilidad y pasión por el área de la luz estructurada no solo me ayudaron a resolver mis dudas, sino que despertaron en mí un profundo interés en el campo. De igual forma, quiero agradecer a su grupo de estudiantes, los cuales me recibieron con los brazos abiertos en el laboratorio de luz estructurada, en especial a Edgar Medina, el cual me resolvió mis dudas relacionadas al área.

Agradezco también al Centro de Investigaciones en Óptica (CIO) por abrirme sus puertas y permitirme trabajar en sus laboratorios, donde pude aprender de sus investigadores y alumnos que hicieron de esta experiencia algo invaluable.

Extiendo mi gratitud a todas aquellas personas que, con sus aportes, han dejado huella en mi camino, contribuyendo a mi formación como persona, hijo y estudiante.

A todos los mencionados, muchísimas gracias.



Resumen

Este trabajo presenta la aplicación de redes neuronales convolucionales (CNN) para el análisis de haces con momento angular orbital con fase azimutal $\exp(il\varphi)$, característica de los haces con vórtices ópticos de carga topológica l . El objetivo principal es abordar dos tareas simultáneas: la clasificación de la carga topológica y la predicción del nivel de turbulencia atmosférica que ha afectado al haz, cuantificado mediante el valor de Strehl.

Se introduce la teoría relevante sobre óptica de haces estructurados y el funcionamiento de las redes neuronales artificiales. A continuación, se detalla el proceso de generación de dos conjuntos de datos: uno experimental, compuesto por 1,250 imágenes adquiridas en laboratorio, y otro simulado, compuesto por 50,000 imágenes generadas computacionalmente. Ambos conjuntos están organizados en 10 clases, correspondientes a valores de carga topológica que van del -18 al 18 en incrementos de 4 unidades, y con niveles de turbulencia representados por valores de Strehl entre 0.3 y 0.7 con pasos de 0.1.

Se describen las consideraciones y criterios seguidos para la generación de ambos conjuntos de datos, así como la arquitectura de la red neuronal utilizada. Posteriormente, se presentan los resultados obtenidos de los modelos entrenados con cada conjunto. El modelo entrenado con imágenes experimentales alcanzó una precisión del 69.09% en la clasificación de la carga topológica y un error absoluto medio de 0.0892 en la predicción del valor de turbulencia. Por su parte, el modelo entrenado con imágenes simuladas logró una precisión del 98.54% y un error absoluto medio de 0.0536 en las mismas tareas.

Finalmente, se discuten posibles mejoras al enfoque propuesto, así como futuras líneas de investigación para fortalecer la generalización y aplicabilidad de los modelos desarrollados.



Índice general

| | |
|--|-----------|
| 1. Introducción | 10 |
| 1.1. Introducción general | 10 |
| 1.2. Planteamiento del problema | 11 |
| 1.3. Hipótesis de investigación | 12 |
| 1.4. Justificación | 12 |
| 1.5. Objetivos | 13 |
| 1.5.1. Objetivo general | 13 |
| 1.5.2. Objetivos específicos | 13 |
| 1.6. Aporte científico | 13 |
| 1.7. Estructura del documento | 14 |
| 2. Generación y perturbación de haces con vórtices ópticos | 15 |
| 2.1. Introducción | 15 |
| 2.2. Haces estructurados | 16 |
| 2.2.1. Ecuación de onda paraxial | 16 |
| 2.2.2. Haz Gaussiano | 17 |
| 2.2.3. Propiedades del haz Gaussiano | 20 |
| 2.2.4. Haces Hermite - Gauss | 24 |
| 2.2.5. Haces Laguerre - Gauss | 28 |
| 2.3. Generación de haces estructurados con moduladores espaciales de luz | 34 |
| 3. Redes neuronales artificiales | 37 |
| 3.1. Introducción | 37 |
| 3.2. Redes neuronales densas | 39 |
| 3.2.1. Neurona artificial | 39 |
| 3.2.2. Perceptrón | 41 |
| 3.2.3. Perceptrón multicapa | 42 |
| 3.2.4. Funciones de activación | 45 |
| 3.2.5. Funciones costo | 48 |

| | | |
|-----------|--|-----------|
| 3.2.6. | Técnicas de regularización | 50 |
| 3.2.7. | Métricas de evaluación | 53 |
| 3.3. | Redes neuronales convolucionales | 56 |
| 3.3.1. | Convolución | 56 |
| 3.3.2. | Padding | 58 |
| 3.3.3. | Stride | 59 |
| 3.3.4. | Pooling | 59 |
| 3.3.5. | Capas | 60 |
| 4. | Metodología | 63 |
| 4.1. | Generación del conjunto de datos experimental | 63 |
| 4.2. | Generación del conjunto de datos simulado | 65 |
| 4.3. | Preprocesamiento de las imágenes | 66 |
| 4.4. | Entrenamiento de la red neuronal convolucional | 67 |
| 5. | Resultados | 70 |
| 5.1. | Desempeño de la CNN entrenada con datos experimentales | 70 |
| 5.2. | Desempeño de la CNN entrenada con datos simulados | 73 |
| 5.3. | Evaluación cruzada simulado-experimental | 75 |
| 6. | Conclusiones | 79 |
| 6.1. | Conclusiones generales | 79 |

Introducción

1.1 Introducción general

Una de las áreas más modernas de la óptica, cuyo impacto tecnológico ha aumentado considerablemente en los últimos años, es el de la **luz estructurada** [10]. Los avances en esta área han permitido el desarrollo de diversas técnicas cada vez más sofisticadas para su estudio y uso, lo que a su vez ha dado lugar a nuevas tecnologías con aplicaciones en diversos campos de la ciencia y la tecnología, como la microbiología, la metrología y la interacción luz-materia. No obstante, uno de los ámbitos donde su desarrollo ha destacado en mayor medida a lo largo de las décadas es el de las **comunicaciones ópticas** [10].

La demanda de transmisión de datos aumenta cada año, lo que ha generado la necesidad de desarrollar métodos de comunicación que operen a mayor escala y velocidad [24]. En este contexto, se han creado diversas técnicas de comunicación óptica que han mostrado tener resultados sobresalientes [24]. Para ello, se han implementado técnicas de multiplexado de flujos de información codificadas en propiedades como la polarización, la amplitud, la forma del pulso y la longitud de onda de la luz [28]. Adicional a estas propiedades, la luz con **momento angular orbital (OAM)** ofrece un grado de libertad adicional para la transmisión de información [28].

El sistema de comunicación óptica que emplea luz con OAM, conocido como multiplexado por división de modos (MDM), véase figura 1.1, se basa en el uso de un conjunto de haces con OAM, donde cada uno presenta una estructura de fase transversal helicoidal caracterizada por un número entero l , denominado **carga topológica**. Este tipo de haces, conocidos como **vórtices ópticos**, presentan una singularidad de fase en el eje de propagación, lo que da lugar a una distribución de intensidad con forma de anillo y a una fase que varía azimutalmente, la cual depende del valor

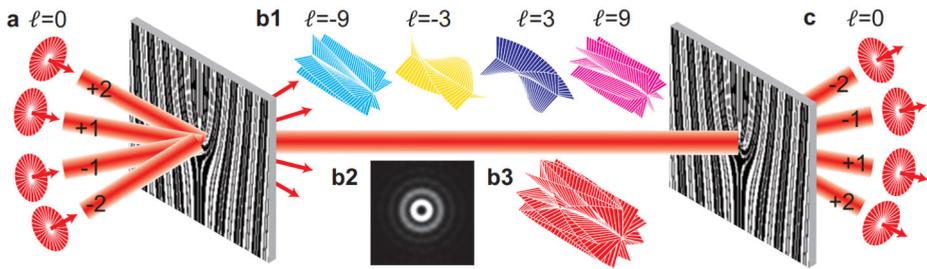


Figura 1.1: Esquema de multiplexado y demultiplexado de haces con OAM. Imagen recuperada de [17].

de la carga topológica. Los modos con diferente valor de l son ortonormales entre sí, lo que permite multiplexarlos en un mismo canal de transmisión con diafonía reducida bajo condiciones ideales [24]. En consecuencia, el uso de múltiples modos en un canal se traduciría en un aumento de velocidad y capacidad de transmisión de la información [10].

Se ha demostrado que, bajo condiciones controladas, este sistema de comunicación puede superar velocidades del orden de gigabits por segundo en espacio libre, e incluso alcanzar velocidades de terabits por segundo en fibras especialmente diseñadas para la transmisión de haces con OAM, superando la tasa de transmisión de gigabits por segundo de las fibras ópticas convencionales. Estos resultados representan un avance significativo con respecto a otros sistemas de transmisión de información, ampliando considerablemente el panorama tecnológico [24] [28]. En consecuencia, el interés por estudiar la propagación de haces con OAM en espacio libre ha ido en aumento, impulsando mejoras graduales en los sistemas de comunicación en espacio libre basados en estos haces [28].

1.2 Planteamiento del problema

Si bien el uso de MDM ha demostrado un gran desempeño en distintas condiciones, la transmisión de datos en espacio libre presenta limitaciones claras. Los principales retos que enfrenta este tipo de sistema se deben a la divergencia del haz y a la turbulencia atmosférica [23].

La divergencia hace que los haces se expandan transversalmente, lo que provoca una pérdida de potencia al ser detectados por aperturas de tamaño fijo. Por su parte, la turbulencia, causada por inhomogeneidades en la temperatura, presión y composición de la atmósfera, genera variaciones en el índice de refracción del medio, lo que degrada los canales de información y aumenta la diafonía entre modos [24]. Los efectos de estos fenómenos se intensifican con la distancia que recorre el haz a través del medio.

Para recuperar la información contenida en haces multiplexados, es necesario obtener simultáneamente tanto su fase como su potencia [24]. La fase permite realizar un correcto demultiplexado, mientras que la potencia del haz asegura una relación

señal-ruido adecuada para la recuperación de la información de manera confiable [24].

Por ello, resulta crucial desarrollar estrategias capaces de recuperar la información transportada por haces con OAM, tanto su fase como su potencia, incluso bajo perturbaciones provocadas por la divergencia y la turbulencia atmosférica, a fin de asegurar la fiabilidad de los sistemas de multiplexado en espacio libre.

1.3 Hipótesis de investigación

Se propone el uso de inteligencia artificial como una herramienta prometedora para el análisis y recuperación de la información codificada en los modos de haces OAM. En particular, se ha demostrado que los modelos basados en redes neuronales convolucionales son eficaces para abordar problemas complejos de clasificación a partir de imágenes [15]. Con base en lo anterior, se plantea la siguiente hipótesis:

Las redes neuronales convolucionales pueden ser entrenadas para identificar correctamente la carga topológica de haces con vórtices ópticos, incluso en presencia de perturbaciones como la divergencia y la turbulencia atmosférica.

1.4 Justificación

Es necesario desarrollar técnicas que mejoren la eficiencia y confiabilidad de los sistemas de comunicación óptica que emplean MDM. Actualmente, resulta natural considerar soluciones basadas en modelos de inteligencia artificial o *deep learning*, dada la notable evolución y el impacto que estas áreas han tenido en los últimos años.

El desempeño de la inteligencia artificial ha demostrado ser altamente efectivo en la resolución de problemas complejos, especialmente en tareas donde los patrones son difíciles de identificar mediante métodos tradicionales. Esta capacidad la convierte en una herramienta prometedora y adecuada para abordar el problema de la identificación de la carga topológica en haces con OAM, cuya morfología depende directamente de dicha carga.

Aunque existen métodos ópticos para determinar la carga topológica de estos haces [28], su rendimiento puede verse considerablemente afectado por perturbaciones atmosféricas, como la turbulencia o la divergencia del haz, las cuales introducen ruido e imprecisiones en los patrones observados. Estas condiciones dificultan la identificación precisa y en tiempo real de los diversos modos OAM, especialmente en sistemas de transmisión en espacio libre, donde las condiciones atmosféricas no son controlables.

Considerando estas limitaciones, la inteligencia artificial surge como una herramienta prometedora para el análisis y recuperación de la información codificada en múltiples modos. En particular, el uso de modelos de inteligencia artificial entrenados para identificar los modos presentes en haces perturbados por turbulencia, a partir de patrones de intensidad, ofrece un enfoque efectivo para superar las dificultades asociadas a las distorsiones provocadas por el medio.

Las redes neuronales convolucionales, al estar diseñadas específicamente para el análisis y reconocimiento de patrones en imágenes, se presentan como una solución eficaz para este tipo de problemas [15]. Su capacidad para extraer características generales incluso en presencia de distorsiones o traslaciones las convierte en candidatas ideales para desarrollar sistemas de clasificación de haces con OAM, capaces de operar en entornos reales con un alto grado de precisión y eficiencia computacional.

Buscando expandir el alcance de la inteligencia artificial en el desarrollo de soluciones tecnológicas aplicadas a la óptica moderna, este trabajo tiene como objetivo explorar el uso de redes neuronales convolucionales para la clasificación e identificación de modos con OAM en condiciones no ideales.

1.5 Objetivos

1.5.1 Objetivo general

El objetivo general de este trabajo es desarrollar y entrenar una red neuronal convolucional capaz de clasificar haces con vórtices ópticos perturbados según su carga topológica, utilizando imágenes de intensidad del haz.

1.5.2 Objetivos específicos

Los objetivos específicos del presente trabajo son:

1. Diseñar un montaje experimental que permita capturar automáticamente imágenes de haces con vórtices ópticos bajo condiciones de medición controladas, con parámetros como distancia de medición, carga topológica y nivel de perturbación de manera ajustable.
2. Desarrollar un código para la simulación de imágenes de haces con vórtices ópticos, que permita variar características como la distancia de medición, la longitud de onda, la cintura del haz, el nivel de perturbación y la carga topológica, para así generar un conjunto de datos para el entrenamiento de la red neuronal.
3. Desarrollar una red neuronal convolucional con salida múltiple que permita predecir el nivel de perturbación presente en los haces, utilizando el cociente de Strehl como referencia para la evaluación del nivel de turbulencia.
4. Evaluar el desempeño de la red neuronal convolucional entrenada con datos simulados utilizando el conjuntos de datos experimentales, con el fin de garantizar la capacidad de generalización del modelo en condiciones reales.

1.6 Aporte científico

Este trabajo propone el uso de redes neuronales convolucionales como una herramienta efectiva para clasificar haces con momento angular orbital en condiciones

no ideales, como turbulencia atmosférica o divergencia. A diferencia de los métodos ópticos tradicionales, permite identificar la carga topológica directamente a partir de imágenes de intensidad, incluso cuando el haz está perturbado. Además, se plantea un modelo con salida múltiple que también estima el nivel de perturbación presente en el haz. Con este enfoque, se busca ampliar el entendimiento sobre la interacción entre turbulencia y vórtices ópticos, y ofrecer herramientas basadas en inteligencia artificial que puedan integrarse en sistemas de comunicación óptica de nueva generación, con aplicaciones potenciales en tecnologías de intercambio de información y metrología.

1.7 Estructura del documento

El resto del documento se encuentra estructurado de la siguiente forma:

- El capítulo 2 introduce la teoría física y matemática en la que los haces estructurados se fundamentan. A partir de la ecuación de onda, se obtienen matemáticamente dos soluciones de familias de haces estructurados de gran importancia: los haces Hermite-Gauss y los Laguerre-Gauss. Se describen propiedades y comportamientos de dichos haces. Posteriormente se explica el concepto de vórtices ópticos, fenómeno que presentan algunos haces estructurados. Por último, se describe como generar haces estructurados usando un regulador de luz espacial.
- El capítulo 3 presenta los fundamentos de las redes neuronales artificiales, comenzando con los primeros modelos bioinspirados. Se explica el algoritmo de retropropagación, el perceptrón multicapa y se detallan los hiperparámetros clave en las capas densas. Se abordan las técnicas de evaluación y regularización que se usaran durante el entrenamiento. El capítulo finaliza con las redes neuronales convolucionales, se explica la teoría detrás de estos modelos, así como los hiperparámetros específicos para este tipo de redes. Por último, se presenta un resumen de las capas que componen este tipo de modelo.
- El capítulo 4 presenta el protocolo experimental y la metodología utilizada para alcanzar los objetivos del trabajo. Se describe el diseño del montaje experimental, la captura automatizada de imágenes, la generación de datos simulados con los criterios considerados y el diseño de la red neuronal convolucional, junto con las estrategias de entrenamiento empleadas para optimizar su rendimiento y evitar el sobreajuste.
- En el capítulo 5 se muestran los resultados obtenidos con los experimentos realizados. Se muestran gráficas de entrenamiento de la red neuronal convolucional, tanto para el conjunto de imágenes experimentales como simuladas. Además, se presentan las métricas de desempeño alcanzadas por el modelo en los conjuntos de pruebas, así como predicciones realizados por la red en dichos conjuntos.
- En el capítulo 6 se discuten los resultados obtenidos, así como posibles opciones de mejora y futuras líneas de investigación.

Generación y perturbación de haces con vórtices ópticos

2.1 Introducción

En los últimos siglos, el interés por la manipulación de la intensidad, fase y polarización de la luz ha crecido considerablemente. En años recientes, este interés se ha intensificado debido a sus potenciales aplicaciones en áreas como microscopía, metrología, comunicación, imagenología e interacción luz-materia, entre otras [10] [24]. En este contexto, el estudio de la **luz estructurada** se ha consolidado como un campo de la óptica dedicado a la generación y control de haces con propiedades espaciales bien definidas [24].

Un punto de inflexión en este ámbito fue la invención del láser, que proporcionó una fuente de luz altamente monocromática, coherente y direccional [30]. Estas propiedades han permitido el desarrollo de diversas técnicas de modulación, ya sea mediante elementos ópticos dentro de la cavidad láser o a través de dispositivos externos, tales como **moduladores espaciales de luz (SLM)** y elementos geométricos de fase (*q-plates*) [10] [24]. Gracias a los avances tecnológicos, estas técnicas se han vuelto cada vez más robustas, destacándose métodos como la holografía y la litografía para la estructuración de la luz [24] [9]. Estas estrategias han permitido la generación de estados de luz con polarización espacialmente variante y perfiles de intensidad modificables [9].

Entre las diversas familias de haces estructurados, algunas presentan propiedades de especial interés, como el **momento angular orbital (OAM)**. Los haces con OAM son capaces de transferir momento angular a partículas atrapadas en su campo, permitiendo su manipulación óptica.

Dado que estos haces siguen siendo ondas electromagnéticas, su función de onda satisface la ecuación de onda y, en particular su aproximación paraxial. Matemáticamente, pueden describirse como el producto de tres factores: una amplitud compleja dependiente de la variable de propagación, una envolvente Gaussiana y una función asociada a la solución no difractante de interés [9].

En este capítulo, se deduce matemáticamente la ecuación de onda paraxial bajo la cual estos haces se rigen, así como dos familias bien conocidas de haces estructurados: los **Hermite-Gauss** y los **Laguerre-Gauss**. Este último es de particular interés debido a la presencia de vórtices ópticos, originados por su momento angular orbital [9]. Finalmente, se describe la generación de estos haces mediante un modulador espacial de luz.

2.2 Haces estructurados

La luz se propaga como ondas a lo largo del espacio, las cuales pueden ser descritas mediante una función escalar ψ . Estas ondas están dadas por una ecuación diferencial de segundo orden denominada la **ecuación de onda**, que se define como

$$\nabla^2 \psi - \frac{1}{c^2} \frac{\partial^2 \psi}{\partial t^2} = 0, \quad (2.1)$$

donde ∇^2 es el operador laplaciano y c es la velocidad con la que se propaga la onda. De lo anterior se observa que $\psi(r, t)$, es una onda con dependencia espacial y temporal. Resulta conveniente trabajar con ondas armónicas en el tiempo, cuyo período se denota como τ . En este caso, una función ψ que varía armónicamente en el tiempo se puede escribir como

$$\psi(r, t) = A(r) \exp(-i\omega t),$$

donde $A(r)$ representa la parte espacial de la onda y ω es su frecuencia angular, con $\omega = 2\pi/\tau$. En esta expresión, se ha realizado una separación explícita entre las dependencias espacial y temporal. Si se sustituye esta solución en la ecuación de onda y se simplifica se obtiene la expresión

$$\nabla^2 A + k^2 A = 0, \quad (2.2)$$

que se denomina como la **ecuación de Helmholtz**, donde k es el número de onda y viene dado como $k = \omega/c$. Esta ecuación permite obtener diferentes soluciones para la parte espacial de la onda en función de las condiciones iniciales dadas.

2.2.1 Ecuación de onda paraxial

Los haces ópticos, normalmente provenientes de láseres, se propagan a lo largo de un eje z . Si bien la intensidad y fase cambian de manera transversal, lo hacen en menor medida en comparación al cambio que sufren a lo largo del eje de propagación z . Por esta razón, conviene modelar la parte espacial $A(r)$ considerando tanto su evolución

en la dirección de propagación como en el plano transversal. Para ello, se establece que

$$A(r) = u(r) \exp(ikz)$$

donde $\exp(ikz)$ denota el cambio a lo largo del eje de propagación z y $u(r)$ determina la variación en la parte transversal del haz. La dependencia de z en la componente transversal u se debe principalmente a la difracción que experimenta el haz debido al medio. Sustituyendo esta función en la ecuación de Helmholtz se obtiene la siguiente expresión

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + 2ik \frac{\partial u}{\partial z} = 0.$$

Dado que $u(r)$ varía lentamente con respecto a z , se establece la condición

$$\frac{\partial^2 u}{\partial z^2} \ll 2k \frac{\partial u}{\partial z},$$

la cual se conoce como *aproximación de envolvente de variación lenta* [26]. Al aplicar esta condición se elimina el término con la segunda derivada con respecto a z , por lo que la ecuación se reduce a

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 2ik \frac{\partial u}{\partial z} = 0,$$

la cual también se suele escribir como

$$\nabla_{\perp}^2 u + 2ik \frac{\partial u}{\partial z} = 0, \tag{2.3}$$

donde ∇_{\perp}^2 se denomina el operador laplaciano transversal. Esta ecuación se conoce como **ecuación de onda paraxial** o **aproximación paraxial de la onda** [26].

Así pues, la función de onda ψ de un haz estructurado tiene la forma

$$\psi(r, t) = u(r) \exp(i(kz - \omega t)) \tag{2.4}$$

donde $u(r)$ es la función que modula transversalmente la onda y sigue la ecuación (2.3). A continuación se deducirán algunas soluciones para la ecuación de onda paraxial.

2.2.2 Haz Gaussiano

La ecuación de onda paraxial permite describir haces de luz en términos de un plano transversal (x, y) y un eje de propagación z [9]. Estos haces se caracterizan por estar confinados, presentar baja dispersión angular y mantener su forma durante la propagación [30] [26].

El caso más representativo son los haces láser, que exhiben un ángulo de dispersión mínimo. Entre ellos, el **haz Gaussiano** es el más simple y el más deseado en fuentes láser [29]. Esta solución fundamental de la ecuación de onda paraxial destaca por sus propiedades bien caracterizadas y su evolución en propagación, la cual es suave y predecible [26] [29].

La expresión del haz Gaussiano puede obtenerse mediante una transformación de la conocida onda esférica, una solución en coordenadas cilíndricas de la ecuación de Helmholtz [26]. Esta onda, descrita por

$$A(r) = \frac{A_0}{r} \exp(ikr), \quad (2.5)$$

corresponde a frentes de onda que se propagan radialmente desde una fuente puntual en todas direcciones. Dichos frentes tienen forma esférica, son concéntricos a la fuente, y su radio r aumenta conforme la onda se propaga.

Considere ahora una onda esférica en el régimen lejano del eje de propagación z , de modo que $z^2 \gg x^2 + y^2$. Bajo esta condición, el radio r del frente de onda esférico puede aproximarse mediante una expansión en series de Taylor a un polinomio de segundo grado, tal que

$$r = \sqrt{x^2 + y^2 + z^2} \approx z + \frac{x^2 + y^2}{2z}.$$

Sustituyendo esta aproximación en la expresión de la onda esférica, se obtiene la función

$$A(r) = \frac{A_0}{z} \exp(ikz) \exp\left(ik \frac{x^2 + y^2}{2z}\right),$$

donde se ha omitido el término $(x^2 + y^2)/z$ en el denominador debido a la condición previa. Esta expresión es conocida como **aproximación de Fresnel de la onda esférica** [26]. Bajo esta condición la onda se considera como onda plana cuando z es relativamente grande, tal y como se puede observar en la figura 2.1. Note que, debido a la condición impuesta sobre el eje de propagación z , la aproximación de Fresnel contiene el término $\exp(ikz)$, el cual está directamente relacionado con la ecuación de onda paraxial. De hecho, la función

$$u(r) = \frac{A_0}{z} \exp\left(ik \frac{x^2 + y^2}{2z}\right) \quad (2.6)$$

es solución a dicha ecuación. A partir de esta expresión, se propone una función de prueba más general, en la que la dependencia en z se sustituye por una función $q(z)$. Por lo que se propone a la función

$$u(r) = \frac{A_0}{q(z)} \exp\left(ik \frac{x^2 + y^2}{2q(z)}\right) \quad (2.7)$$

como solución a la ecuación de onda paraxial. Derivando esta solución, se obtienen las siguientes relaciones

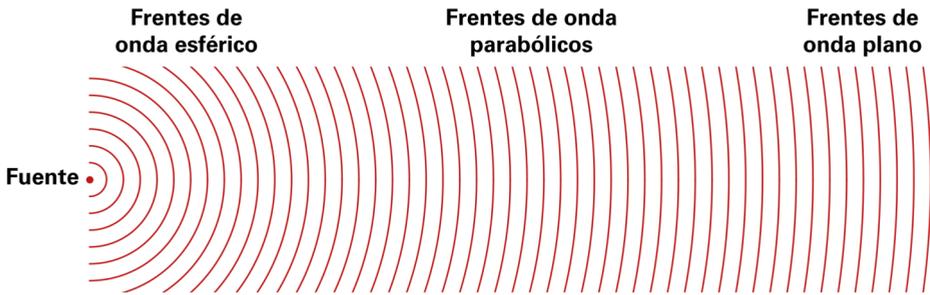


Figura 2.1: Onda esférica bajo la aproximación de Fresnel. A distancias lejanas sobre el eje de propagación, las ondas esféricas se consideran ondas planas.

$$\begin{aligned} \frac{d^2u}{dx^2} &= u \left[i \frac{k}{q} - \left(\frac{kx}{q} \right)^2 \right] \\ \frac{d^2u}{dy^2} &= u \left[i \frac{k}{q} - \left(\frac{ky}{q} \right)^2 \right] \\ \frac{du}{dz} &= u \left[\frac{1}{q} + i \frac{k}{2} \frac{x^2 + y^2}{q^2} \right] \frac{dq}{dz} \end{aligned}$$

donde $u = u(r)$ y $q = q(z)$. Al sustituir estas expresiones en la ecuación de onda paraxial y simplificar se obtiene la siguiente relación

$$\left(2ik - \left(\frac{k}{q} \right)^2 (x^2 + y^2) \right) \left(\frac{dq}{dz} - 1 \right) = 0.$$

Para que lo anterior se cumpla, alguno de los términos debe ser cero, por lo que se establece la condición

$$\frac{dq}{dz} - 1 = 0. \tag{2.8}$$

Al resolver esta condición, se obtiene como solución una función lineal $q(z)$ con una constante z_0 . Si $z_0 = 0$, se obtiene la ecuación (2.6). Sin embargo, al asumir que la constante z_0 es un número puramente complejo y sustituir $q(z)$ en la ecuación (2.7), la función de onda adopta una forma que permite describir adecuadamente su propagación, capturando características físicas relevantes del haz. Así pues, se propone que la solución a la ecuación (2.8) es

$$q(z) = z - iz_0. \tag{2.9}$$

Dado que en la ecuación (2.7) la función compleja $q(z)$ aparece en el denominador, es posible manipularla algebraicamente para separar su parte real y su parte imaginaria, obteniendo que

$$\frac{1}{q(z)} = \frac{1}{R(z)} + i \frac{2}{kw^2(z)}, \tag{2.10}$$

donde $R(z)$ y $w(z)$ están asociadas al radio de curvatura y a la cintura del haz, respectivamente. Estos conceptos serán definidos con mayor detalle en la siguiente sección. Por otro lado, el parámetro $q(z)$ también puede expresarse como un número complejo en forma polar, es decir, mediante su módulo y fase de la siguiente forma

$$\frac{i}{q(z)} = \left| \frac{i}{q(z)} \right| \arg \left(\frac{i}{q(z)} \right) = \frac{w_0}{z_0 w(z)} \exp(-i\zeta(z)), \quad (2.11)$$

donde w_0 representa el radio de cintura del haz y $\zeta(z)$ es la llamada *fase de Gouy*. Ambos términos serán abordados en profundidad más adelante.

De este modo, al sustituir las expresiones (2.10) y (2.11) en la ecuación (2.7), se obtiene la siguiente expresión

$$u(r) = u_0 \frac{w_0}{w(z)} \exp\left(-\frac{x^2 + y^2}{w^2(z)}\right) \exp\left(ik \frac{x^2 + y^2}{2R(z)}\right) \exp(-i\zeta(z)), \quad (2.12)$$

con $u_0 = A_0/iz_0$. Esta expresión es la **función del haz Gaussiano**, la cual permite describir la evolución transversal del campo en un sistema óptico libre.

2.2.3 Propiedades del haz Gaussiano

Cada uno de los términos en la expresión del haz Gaussiano varía con la coordenada z y describe una propiedad física específica del haz. A continuación, se presentan dichas funciones y su interpretación física.

Envolvente gaussiana

El haz Gaussiano, así como las demás familias de haces que se mencionarán a lo largo del capítulo, presenta una envolvente gaussiana que determina la distribución de su magnitud en la dirección transversal al eje de propagación. La función gaussiana, dada por los términos

$$\frac{w_0}{w(z)} \exp\left(-\frac{x^2 + y^2}{w^2(z)}\right),$$

hace que el máximo del campo $u(r)$ se localice en el eje del haz. Si bien, teóricamente el campo ocupa todo el espacio transversal, este decrece de forma continua, provocando que la contribución del campo a grandes distancias transversales, es decir, cuando $\rho = \sqrt{x^2 + y^2}$ es suficientemente grande, se vuelve despreciable en la práctica. De esta manera, el campo se encuentra confinado a una región próxima al eje de propagación. El valor de $w(z)$ dentro de la exponencial modifica el ancho de la curva gaussiana. Si la distancia trasversal ρ es exactamente $w(z)$, la magnitud del campo se reduce hasta e^{-1} de su valor en el centro, el cual es alrededor del 36.79% de la magnitud máxima en el plano. En la figura 2.2 se puede apreciar como la intensidad del campo decae conforme se aleja transversalmente del eje de propagación.

Por su parte, el denominador de $w(z)$ fuera de la exponencial modifica la intensidad del haz en función de la distancia de propagación z . Este valor, junto con la exponencial, permiten que la curva gaussiana se encuentre normalizada. En la figura

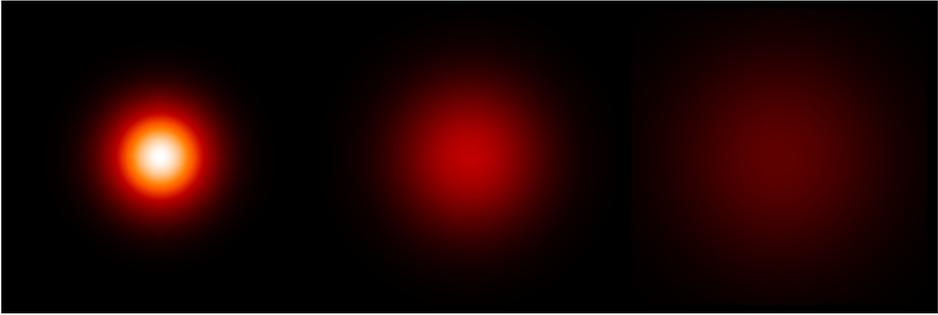


Figura 2.2: Simulación de los perfiles de intensidad de un haz Gaussiano de 669nm a distintas distancias de propagación. De izquierda a derecha, las distancias de propagación son 0.1, z_0 y $\sqrt{3}z_0$ metros. Es posible apreciar cómo la cintura del haz se ensancha y pierde intensidad conforme el haz se propaga.

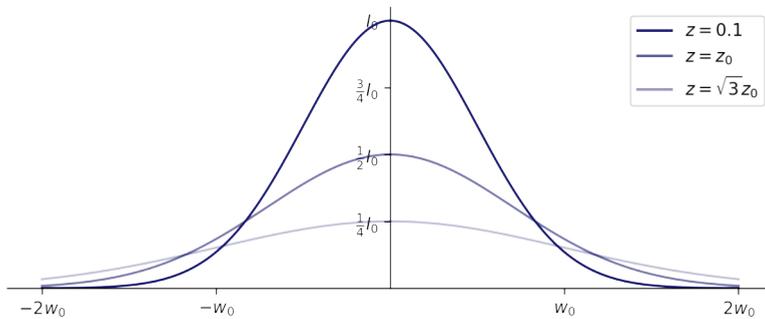


Figura 2.3: Intensidad a lo largo del eje x de un haz a distintas distancias.

2.2 se observa como a diversas distancias de propagación z la intensidad del haz cambia transversalmente. En la figura 2.3 se observa la intensidad en el centro del haz a lo largo del eje x para un haz a distintas distancias de propagación z .

Cintura del haz

Como se mencionó, la función $w(z)$ determina la distancia transversal desde el centro del haz en el que el campo decae a e^{-1} , o a e^{-2} de su intensidad, del valor máximo en el haz. Este radio, cambia a lo largo del eje de propagación z , tal y como se puede apreciar en la figura 2.2. La expresión de $w(z)$, conocido como cintura del haz, viene dado por la siguiente expresión

$$w(z) = w_0 \sqrt{1 + \left(\frac{z}{z_0}\right)^2}, \quad (2.13)$$

donde w_0 es el radio de cintura mínimo del haz y z_0 es la constante de la ecuación (2.9). En la figura 2.4 se puede apreciar la gráfica de esta función. Para distancias

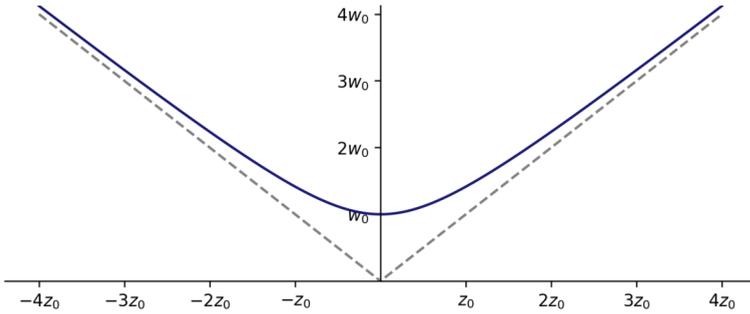


Figura 2.4: Radio del haz en función de la distancia de propagación. La curva sólida azul representa la función $w(z)$ mientras que la curva punteada gris muestra su aproximación.

grandes, el radio del haz puede ser aproximado mediante las rectas

$$w(z) \approx \pm \frac{w_0}{z_0} z$$

Así pues, el haz tendrá un radio de cintura mínima w_0 , el cual irá creciendo conforme el haz se propague. A una distancia z_0 el radio habrá aumentado a $\sqrt{2}w_0$, mientras que a una distancia de $\sqrt{3}z_0$ este será el doble de w_0 .

Radio de curvatura

El radio de curvatura, denotado como $R(z)$, viene dado por la expresión

$$R(z) = z \left(1 + \left(\frac{z_0}{z} \right)^2 \right). \quad (2.14)$$

Esta función modifica la forma del frente de onda del haz Gaussiano a lo largo del eje de propagación z . Como se puede observar en la figura 2.5, para el valor de z en el origen el radio de curvatura es infinito, por lo cual el frente de onda en ese punto es plano. El frente de onda se curva a lo largo de z , teniendo un valor de curvatura mínimo en z_0 para posteriormente aumentar de manera lineal. En la figura 2.6 se puede observar el cambio en la curvatura para distintos frentes de onda a lo largo del eje de propagación.

Fase de Gouy

La función $\zeta(z)$, definida como

$$\zeta(z) = \arctan \left(\frac{z}{z_0} \right), \quad (2.15)$$

describe un retraso en la fase del haz en comparación con una onda plana [26]. Este fenómeno, conocido como **fase de Gouy**, se origina debido al confinamiento transversal del haz [29]. Físicamente, esto implica que la constante de propagación efectiva

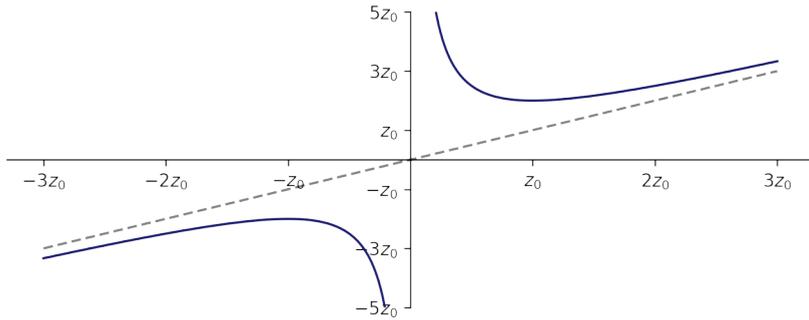


Figura 2.5: Función del radio de curvatura $R(z)$ en azul y la aproximación lineal en gris.

k en la región cercana a la cintura del haz es ligeramente menor, lo que genera una acumulación adicional de fase a lo largo del eje z . Desde el punto de vista matemático, este comportamiento también se deduce al resolver la ecuación de onda en la aproximación paraxial, la cual permite un pequeño término adicional asociado a la variación longitudinal de la fase [29]. En la figura 2.7 se observa el cambio en la fase que sufre el haz a lo largo del eje de propagación.

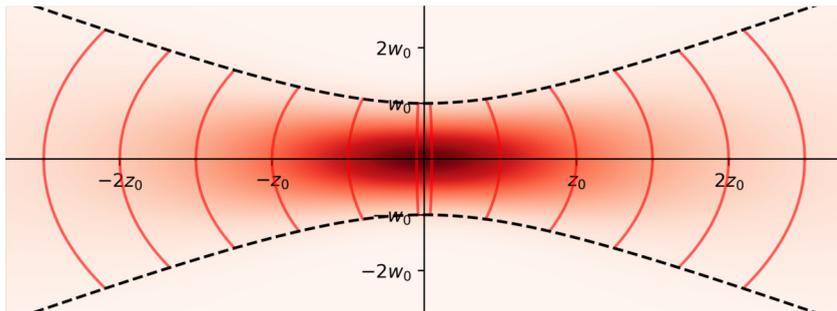


Figura 2.6: Intensidad del haz Gaussiano en propagación. La intensidad en el origen es máxima decayendo conforme el haz se propaga transversal y longitudinalmente. Las curvas negras punteadas marcan la cintura del haz en el que la intensidad decae a e^{-2} del valor máximo en el plano, las líneas sólidas rojas muestran el frente de onda del haz a lo largo del eje de propagación.

Parámetro de haz complejo

La ecuación (2.9) es fundamental para la caracterización de haces Gaussianos, ya que permite describir todas sus propiedades relevantes mediante un único parámetro. Como se observa en la ecuación (2.10), a partir del valor de $q(z)$ es posible obtener tanto la curvatura del frente de onda como el radio del haz. El parámetro z_0 conocido

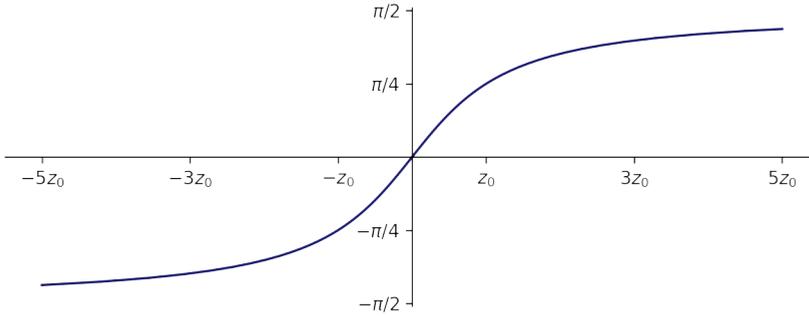


Figura 2.7: Retraso en fase de un haz Gaussiano dado por la función $\zeta(z)$. Este retraso en la fase es conocido como Fase de Gouy.

como *distancia de Rayleigh*, y denotado en algunos textos como z_R [26], se define de la siguiente forma

$$z_0 = \frac{\pi w_0^2}{\lambda}, \quad (2.16)$$

donde w_0 es el radio de cintura mínimo y λ es la longitud de onda del haz. Esta constante establece una escala longitudinal característica para la propagación del haz, y se utiliza comúnmente como parámetro de normalización. Además de su capacidad para condensar la información del haz, el parámetro $q(z)$ es especialmente útil en el análisis de sistemas ópticos mediante matrices ABCD [29].

2.2.4 Haces Hermite - Gauss

Si bien el haz Gaussiano es una solución de la ecuación de onda paraxial, corresponde al orden más bajo dentro de una familia más amplia de soluciones. La ecuación de onda paraxial permite múltiples expresiones, dependiendo del sistema de coordenadas utilizado para su resolución. Entre las familias de soluciones se encuentran los haces Hermite-Gauss en coordenadas cartesianas, los Laguerre-Gauss en coordenadas cilíndricas, y los Ince-Gauss en coordenadas elípticas. En esta sección, nos enfocaremos en los haces Hermite-Gauss.

Para obtener la expresión de los haces **Hermite-Gauss (HG)**, se propone una solución a la ecuación de onda paraxial, de tal forma que se encuentren separadas las coordenadas transversales x y y en distintas funciones, tal que

$$u(r) = u_x(x, z) u_y(y, z). \quad (2.17)$$

Al sustituir esta expresión en la ecuación (2.3), se obtiene que

$$u_y \left(\frac{d^2 u_x}{dx^2} + 2ik \frac{du_x}{dz} \right) + u_x \left(\frac{d^2 u_y}{dy^2} + 2ik \frac{du_y}{dz} \right) = 0.$$

Para que esta igualdad se satisfaga, es necesario que cada término entre paréntesis sea igual a 0. Note que cada uno de estos términos tiene la forma de la ecuación paraxial

pero unidimensional. Se propondrán soluciones para u_x y u_y con la misma estructura funcional, pero variando su dependencia espacial en x y y , respectivamente.

Resolviendo para u_x , se propone una solución similar a la función de prueba presentada en la ecuación (2.7), de tal forma que

$$u_x(x, z) = A_0 h(\theta(x, z)) \exp\left(ik \frac{x^2}{2q(z)}\right) \exp(ip(z)), \quad (2.18)$$

donde la función $h(\theta)$ permite generar la familia de haces Hermite-Gauss, modificando la estructura transversal del haz, y $\exp(ip(z))$ introduce un grado de libertad adicional para la resolución de la ecuación de onda paraxial. Con respecto a la función $\theta(x, z)$, se propone que tenga la siguiente forma

$$\theta(x, z) = \frac{x}{a(z)}, \quad (2.19)$$

donde $a(z)$ es, en principio, una función desconocida. Derivando la ecuación (2.18) se obtienen las siguientes relaciones:

$$\begin{aligned} \frac{d^2 u_x}{dx^2} &= u_x \left[\frac{1}{a^2 h} \frac{d^2 h}{d\theta^2} + 2ik \frac{x}{ahq} \frac{dh}{d\theta} + i \frac{k}{q} - \left(k \frac{x}{q}\right)^2 \right] \\ \frac{du_x}{dz} &= -u_x \left[\frac{x}{a^2 h} \frac{da}{dz} \frac{dh}{d\theta} + i \frac{k}{2} \frac{x^2}{q^2} \frac{dq}{dz} - i \frac{dp}{dz} \right] \end{aligned}$$

Al sustituir en la ecuación de onda paraxial unidimensional y simplificar se obtiene la siguiente expresión

$$\frac{d^2 h}{d\theta^2} + 2ikx \left(\frac{a}{q} - \frac{da}{dz} \right) \frac{dh}{d\theta} + \left[\left(k \frac{x}{q}\right)^2 \left(\frac{dq}{dz} - 1 \right) + k \left(\frac{i}{q} - 2 \frac{dp}{dz} \right) \right] a^2 h = 0.$$

Es necesario resolver esta ecuación diferencial de segundo orden para obtener la función $h(\theta)$. Debido a su estructura, se propone que tenga la forma de la ecuación diferencial de Hermite, la cual viene dada como

$$\frac{d^2 h}{d\theta^2} - 2\theta \frac{dh}{d\theta} + 2mh = 0. \quad (2.20)$$

Para que esto se cumpla, es necesario igualar términos de ambas ecuaciones, estableciendo las siguientes condiciones

$$\frac{dq}{dz} - 1 = 0 \quad (2.21)$$

$$2ikx \left(\frac{a}{q} - \frac{da}{dz} \right) = -2 \frac{x}{a} \quad (2.22)$$

$$k \left(\frac{i}{q} - 2 \frac{dp}{dz} \right) a^2 = 2m \quad (2.23)$$

La condición (2.21) es la misma que aparece en (2.8), por lo tanto su solución $q(z)$ es la función (2.9). Por su parte, la condición (2.22) es una ecuación diferencial reducible a lineal la cual puede ser resuelta por un cambio de variable. Resolviendo esta ecuación se obtiene la función

$$a(z) = \sqrt{a_0 q^2(z) + i \frac{2}{k} q(z)}.$$

Si se asume que $a(z)$ toma unicamente valores reales, se obtiene que $a_0 = 1/(z_0 k)$, por lo cual la función se reduce a

$$a(z) = \frac{w(z)}{\sqrt{2}}, \tag{2.24}$$

donde $w(z)$ es la función de la cintura del haz, presentado en la ecuación (2.13). La condición (2.23) es una ecuación diferencial lineal, cuya solución es

$$p(z) = \frac{1}{2} \ln(q(z)) - m \arctan\left(\frac{z}{z_0}\right) + p_0. \tag{2.25}$$

Por lo tanto, bajo estas restricciones, la ecuación de onda paraxial adopta la forma de la ecuación diferencial de Hermite, cuya solución está dada por los polinomios de Hermite H_m . Estos constituyen una familia de polinomios ortogonales que dependen del parámetro entero m , conocido como orden del polinomio. Existen distintas formas de calcular o definir los polinomios de Hermite. A continuación se muestran dos de las más comunes.

1. Mediante la fórmula de Rodrigues:

$$H_m(\theta) = (-1)^m \exp(\theta^2) \frac{d^m}{d\theta^m} (\exp(-\theta^2))$$

2. Mediante una relación de recurrencia:

$$H_m(\theta) = 2\theta H_{m-1}(\theta) - 2(m-1)H_{m-2}(\theta), \quad m \geq 2$$

con $H_0(\theta) = 1$ y $H_1(\theta) = 2\theta$.

En la figura 2.8 se muestra el comportamiento de varios polinomios de Hermite de distinto orden m .

Al sustituir en la función (2.18) las funciones encontradas por las condiciones impuestas y simplificar se obtiene que

$$u_x(x, z) = A_{0x} H_m \left(\frac{\sqrt{2}x}{w(z)} \right) \frac{\exp(ip_{0x})}{\sqrt{q(z)}} \exp \left(-\frac{x^2}{w^2(z)} \right) \exp \left(ik \frac{x^2}{2R(z)} \right) \exp(-im\zeta(z)),$$

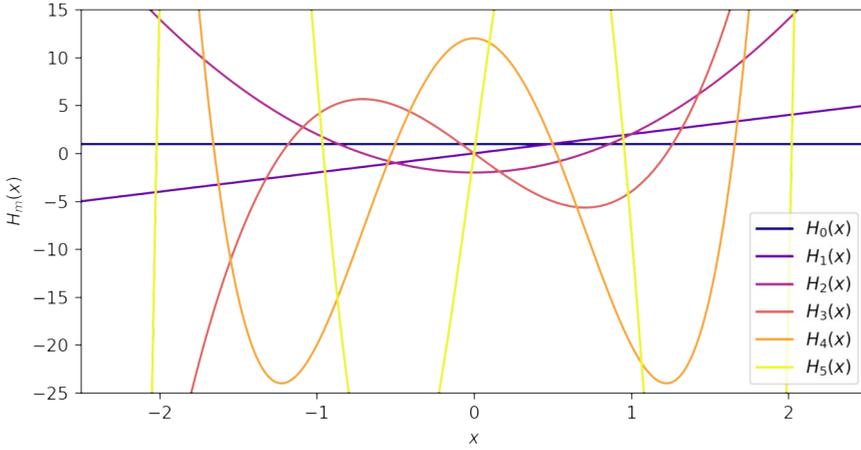


Figura 2.8: Algunos polinomios de Hermite.

donde $\zeta(z)$ es la expresión de la fase de Gouy, presentada en la ecuación (2.15). La función u_y puede ser obtenida de manera análoga, teniendo la misma forma que u_x , de tal forma que

$$u_y(y, z) = A_{0y} H_l \left(\frac{\sqrt{2}y}{w(z)} \right) \frac{\exp(ip_{0y})}{\sqrt{q(z)}} \exp \left(-\frac{y^2}{w^2(z)} \right) \exp \left(ik \frac{y^2}{2R(z)} \right) \exp(-i\zeta(z)),$$

donde se ha sustituido el orden m por l . Con estas dos funciones es posible encontrar $u(r)$, obteniendo la expresión

$$u(r) = A_0 H_m \left(\frac{\sqrt{2}x}{w(z)} \right) H_l \left(\frac{\sqrt{2}y}{w(z)} \right) \frac{\exp(ip_0)}{q(z)} \exp \left(-\frac{x^2 + y^2}{w^2(z)} \right) \exp \left(ik \frac{x^2 + y^2}{2R(z)} \right) \exp(-i(m+l)\zeta(z))$$

donde A_0 ha absorbido el producto de las constantes y $p_0 = p_{0x} + p_{0y}$. Nótese que la función $q(z)$ aparece en el denominador de $u(r)$, por lo cual, es posible expresar esta función en su forma polar, de manera análoga a como se hizo en la expresión del haz Gaussiano (véase la ecuación (2.11)). Por lo tanto, al aplicar dicha transformación se obtiene que

$$u_y(y, z) = u_0 \frac{w_0}{w(z)} H_m \left(\frac{\sqrt{2}x}{w(z)} \right) H_l \left(\frac{\sqrt{2}y}{w(z)} \right) \exp \left(-\frac{x^2 + y^2}{w^2(z)} \right) \exp \left(ik \frac{x^2 + y^2}{2R(z)} \right) \exp(-i(N+1)\zeta(z)), \quad (2.26)$$

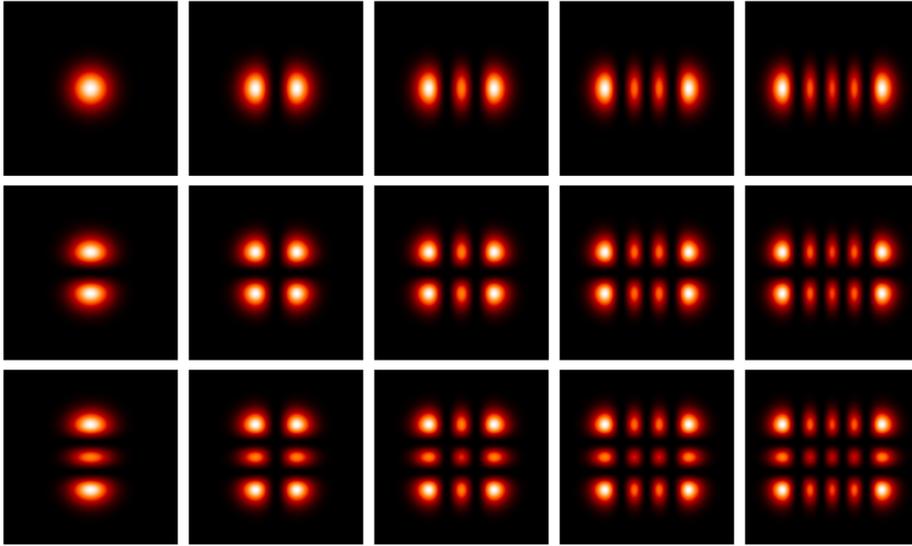


Figura 2.9: Perfiles de intensidad de haces Hermite-Gauss. De izquierda a derecha, el orden m toma valores de 0 a 4, y de arriba a abajo, el orden l varía de 0 a 2.

donde $u_0 = -iA_0$, con $p_0 = 0$ y $N = m + l$. Esta expresión corresponde a la **función de los haces Hermite-Gauss**. Nótese que si $m = l = 0$ se obtiene la expresión del haz Gaussiano, lo cual indica que este caso es el menor orden dentro de la familia de haces HG. En la figura 2.9 se muestran distintos haces Hermite-Gauss con diferentes ordenes m y l .

2.2.5 Haces Laguerre - Gauss

Así como los haces Hermite-Gauss presentan una simetría en coordenadas rectangulares, los haces **Laguerre-Gauss (LG)** poseen una simetría en coordenadas cilíndricas. Como se verá a continuación, la derivación de la función de onda de los haces Laguerre-Gauss comparte similitudes con la de los haces Hermite-Gauss.

Debido a la simetría en coordenadas cilíndricas que poseen los haces Laguerre-Gauss, es necesario representar la ecuación de onda paraxial en este conjunto de coordenadas. El operador laplaciano en coordenadas cilíndricas está dado como

$$\nabla^2 = \frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2}.$$

Dado que el operador laplaciano transversal omite el término con la segunda derivada con respecto a z , la ecuación paraxial en coordenadas cilíndricas queda definida de la siguiente forma

$$\frac{\partial^2 u}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial u}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2 u}{\partial \varphi^2} + 2ik \frac{\partial u}{\partial z} = 0 \quad (2.27)$$

De manera análoga a los haces HG, se propone una función prueba que cumpla con la ecuación de onda paraxial. La función $u(r)$ que se propone es

$$u(r) = A_0 h(\theta(\rho, z)) \exp\left(ik \frac{\rho^2}{2q(z)}\right) \exp(ip(z)) \exp(is(\varphi)). \quad (2.28)$$

Nótese que esta expresión es idéntica a la función prueba utilizada para los haces HG (véase ecuación (2.18)), con la diferencia de que incluye el factor $\exp(is(\varphi))$, el cual introduce la dependencia azimutal a través de la función $s(\varphi)$. Por su parte, la función $h(\theta)$ ahora toma la forma

$$h(\theta) = \theta^m g(\theta), \quad (2.29)$$

donde $g(\theta)$ es una función desconocida. Asimismo, la variable $\theta(\rho, z)$ se redefine como

$$\theta(\rho, z) = \left(\frac{\rho}{a(z)}\right)^2. \quad (2.30)$$

La razón por la cual θ se define como una función cuadrática, a diferencia del caso de los haces HG, es que, debido a la simetría cilíndrica de los haces LG, este factor puede interpretarse como una medida de distancia radial, y no solo como la distancia con respecto a un eje de propagación transversal, como ocurría en el caso de los haces HG.

A continuación, se presentan relaciones que serán de importancia para obtener la ecuación de los haces Laguerre-Gauss. Dada la definición de $h(\theta)$, se obtienen las siguientes derivadas

$$\begin{aligned} \frac{dh}{d\theta} &= h \left[\frac{1}{g} \frac{dg}{d\theta} + \frac{m}{\theta} \right] \\ \frac{d^2h}{d\theta^2} &= h \left[\frac{1}{g} \frac{d^2g}{d\theta^2} + 2 \frac{m}{g\theta} + \frac{m^2 - m}{\theta^2} \right] \end{aligned}$$

Por su parte, las derivadas relacionadas a $\theta(\rho, z)$ son

$$\begin{aligned} \frac{d\theta}{d\rho} &= 2 \frac{\rho}{a^2} \\ \frac{d\theta}{dz} &= -2 \frac{\rho^2}{a^3} \frac{da}{dz} \end{aligned}$$

Con esto en cuenta, es posible obtener las derivadas de $u(r)$ necesarias para resolver

la ecuación de onda paraxial. Estas son

$$\begin{aligned}\frac{\partial u}{\partial \rho} &= u \left[\frac{2}{a^2 g} \frac{dg}{d\theta} + 2 \frac{m}{a^2 \theta} + i \frac{k}{q} \right] \rho \\ \frac{\partial^2 u}{\partial \rho^2} &= u \left[4 \frac{\theta}{a^2 g} \frac{d^2 g}{d\theta^2} + \frac{2}{a^2 g} \left(1 + 4m + 2ik \frac{\rho^2}{q} \right) \frac{dg}{d\theta} \right. \\ &\quad \left. + \frac{2}{a^2 \theta} \left(2m^2 - m + 2ikm \frac{\rho^2}{q} + i \frac{k}{q} - \left(k \frac{\rho}{q} \right)^2 \right) \right] \\ \frac{\partial^2 u}{\partial \varphi^2} &= u \left[i \frac{d^2 s}{s \varphi^2} - \left(\frac{ds}{d\varphi} \right)^2 \right] \\ \frac{\partial u}{\partial z} &= u \left[-2 \frac{\theta}{ag} \frac{da}{dz} \frac{dg}{d\theta} - 2 \frac{m}{a} \frac{da}{dz} - i \frac{k}{2} \frac{\rho^2}{q^2} \frac{dq}{dz} + i \frac{dp}{dz} \right]\end{aligned}$$

Al sustituir estas derivadas en la ecuación de onda paraxial, agrupando y reduciendo términos, se obtiene la siguiente relación

$$\begin{aligned}\theta \frac{d^2 g}{d\theta^2} + \left[1 + 2m + \theta \left(ika \left(\frac{a}{q} - \frac{da}{dz} \right) \right) \right] \frac{dg}{d\theta} + \left[\frac{1}{\theta} \left(m^2 + \frac{1}{4} \left(i \frac{d^2 s}{d\varphi^2} - \left(\frac{ds}{d\varphi} \right)^2 \right) \right) + \right. \\ \left. m \left(ika \left(\frac{a}{q} - \frac{da}{dz} \right) \right) + a^2 \frac{k}{2} \left(\frac{i}{q} - \frac{dp}{dz} \right) + \frac{a^2}{4} \left(k \frac{\rho}{q} \right)^2 \left(\frac{dq}{dz} - 1 \right) \right] g = 0.\end{aligned}$$

De igual forma que con los haces HG, es necesario resolver la ecuación diferencial para encontrar la función $g(\theta)$, por lo cual, se propone que esta tenga la estructura de la ecuación diferencial de Laguerre, la cual está dada como

$$\theta \frac{d^2 g}{d\theta^2} + (1 + l - \theta) \frac{dg}{d\theta} + ng = 0. \quad (2.31)$$

Para que esto se cumpla, se establece el siguiente conjunto de condiciones

$$\frac{dq}{dz} - 1 = 0 \quad (2.32)$$

$$ika \left(\frac{a}{q} - \frac{da}{dz} \right) = -1 \quad (2.33)$$

$$a^2 \frac{k}{2} \left(\frac{i}{q} - \frac{dp}{dz} \right) - m = n \quad (2.34)$$

$$m^2 + \frac{1}{4} \left(i \frac{d^2 s}{d\varphi^2} - \left(\frac{ds}{d\varphi} \right)^2 \right) = 0 \quad (2.35)$$

Nótese que las 3 primeras condiciones impuestas para los haces LG son condiciones análogas a las condiciones de los haces HG, por lo que sus soluciones son equivalentes. La condición (2.32) proporciona la solución al término $q(z)$ (véase ecuación (2.9)). La

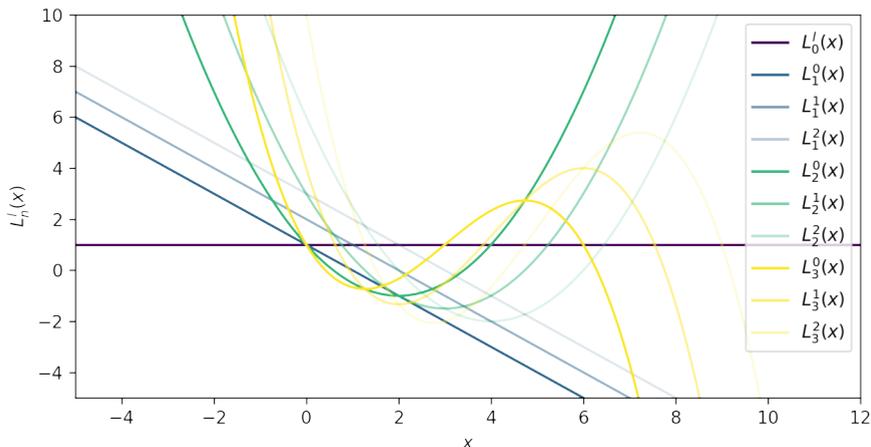


Figura 2.10: Algunos polinomios asociados de Laguerre.

condición (2.33) es la misma que la condición (2.22), cuya solución está dado por la ecuación (2.24) al tomar el mismo valor inicial a_0 . Por su parte, la solución de la ecuación (2.34) es equivalente a la función obtenida para los haces HG (véase ecuación (2.25)), de tal forma que para los haces LG se obtiene que

$$p(z) = i \ln(q(z)) - (2n + 2m) \arctan\left(\frac{z}{z_0}\right) + p_0. \quad (2.36)$$

Por otro lado, la condición (2.35) se debe exclusivamente a la dependencia azimutal de los haces LG. Esta condición es una ecuación diferencial de segundo orden reducible a primer orden, cuya solución está dada por función

$$s(\varphi) = 2m\varphi - i \ln(1 - s_0 \exp(-4im\varphi)) + s_1. \quad (2.37)$$

Con esto en cuenta, la ecuación diferencial paraxial toma la forma de la ecuación diferencial asociada de Laguerre, cuya solución son los polinomios asociados de Laguerre L_n^l , por lo que $g(\theta) = L_n^l(\theta)$ con $l = 2m$. Estos polinomios pueden ser definidos de manera similar a los haces HG. A continuación se presentan estas formas.

1. Mediante la fórmula de Rodrigues:

$$L_n^l(\theta) = \frac{\theta^l e^\theta}{n!} \frac{d^n}{d\theta^n} (e^{-\theta} \theta^{n+l})$$

2. Mediante una relación de recurrencia:

$$L_n^l(\theta) = \frac{1}{n} [(2n - 1 + l - \theta)L_{n-1}^l(\theta) - (n - 1 + l)L_{n-2}^l(\theta)], \quad n \geq 2$$

con $L_0^l(\theta) = 1$ y $L_1^l(\theta) = 1 + l - \theta$.

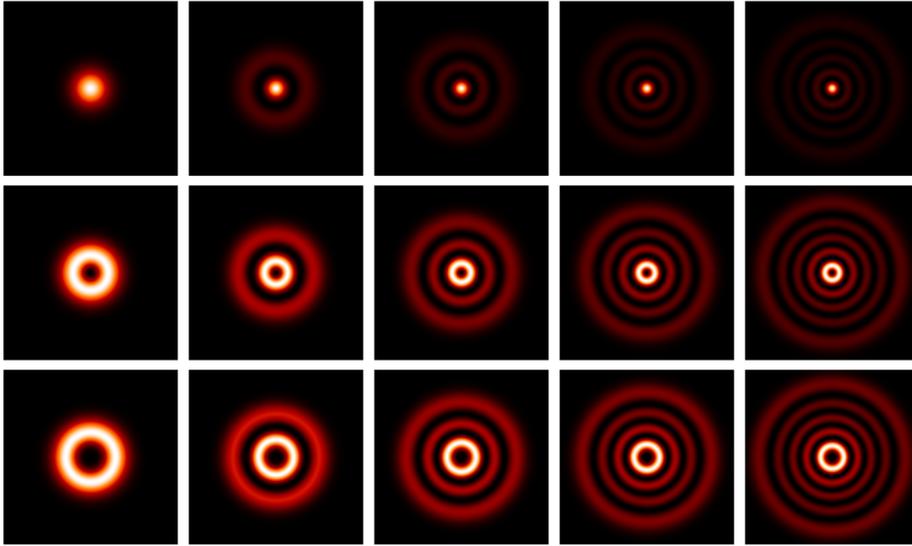


Figura 2.11: Perfiles de intensidad de haces Laguerre-Gauss. De izquierda a derecha, el índice radial n toma valores de 0 a 4, y de arriba a abajo, el índice azimutal l varía de 0 a 2.

En la figura 2.10 se puede apreciar algunos de los polinomios de Laguerre asociados, para distintos valores de n y l .

Al reemplazar las funciones encontradas en la función (2.28), y simplificar se obtiene que

$$u(r) = A_0 \left(\frac{\sqrt{2}\rho}{w(z)} \right)^l L_n^l \left(2 \frac{\rho^2}{w^2(z)} \right) \frac{\exp(ip_0)}{q(z)} \exp \left(- \frac{\rho^2}{w^2(z)} \right) \exp \left(ik \frac{\rho^2}{2R(z)} \right) \exp(-i(2n+l)\zeta(z)) \exp(il\varphi) (1 - s_0 \exp(-4im\varphi)) \exp(is_1)$$

donde $N = 2n + l$. Nótese que en esta función vuelve a aparecer el término $q(z)$ en el denominar, por lo que es posible transformar la función en su forma polar (véase ecuación (2.11)). Así pues al aplicar esta transformación y hacer que las condiciones iniciales sobre $p(z)$ y $s(\varphi)$ sean 0, se obtiene la función

$$u(r) = u_0 \frac{w_0}{w(z)} \left(\frac{\sqrt{2}\rho}{w(z)} \right)^l L_n^l \left(2 \frac{\rho^2}{w^2(z)} \right) \exp \left(- \frac{\rho^2}{w^2(z)} \right) \exp \left(ik \frac{\rho^2}{2R(z)} \right) \exp(-i(N+1)\zeta(z)) \exp(il\varphi), \quad (2.38)$$

la cual es la **función de los haces Laguerre-Gauss**, donde el valor de n es el

índice radial y l es el índice azimutal. En la figura 2.11 se muestran varios perfiles de intensidad de haces Laguerre-Gauss variando el índice radial y azimutal. Note que si $n = l = 0$ se obtiene nuevamente la expresión del haz gaussiano.

Vórtices ópticos

Se define como **vórtice óptico** al lugar donde la fase de un campo escalar complejo en espacio libre se encuentra indefinida [24]. Se les denomina de esta forma debido a que la fase gira alrededor del eje de propagación, análogo al flujo de un fluido en un vórtice o torbellino, tal y como se observa en la figura 2.12. En la *singularidad de fase*, el punto donde esta se indefine, se genera un punto de intensidad cero, dando lugar a una mancha oscura en el haz [24] [28]. El fenómeno de vórtices ópticos se debe exclusivamente a la estructura en el espacio de fase del haz.

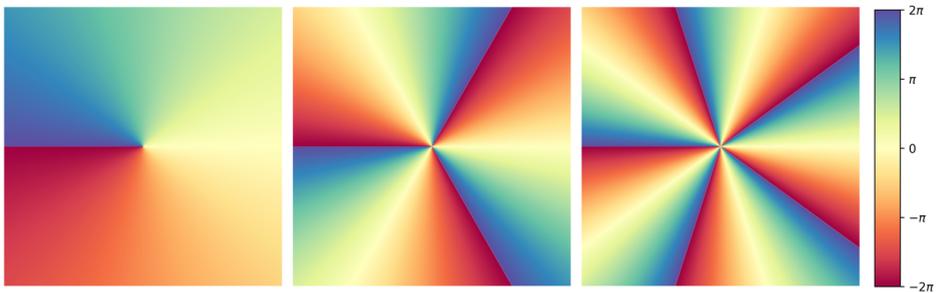


Figura 2.12: Fases de vórtices ópticos. De izquierda a derecha las cargas topológicas son 1, -3 y 5, respectivamente.

En 1992, Allen descubrió que los haces con vórtices ópticos poseen momento angular orbital, lo cual estableció una conexión entre fenómenos de óptica macroscópica y efectos cuánticos [28]. Debido a la facilidad con la que pueden generarse estos haces en el laboratorio, se convirtieron en una herramienta sumamente útil para el estudio de las propiedades macroscópicas y cuánticas de los vórtices ópticos [28]. Uno de los primeros haces reportados con OAM fueron los haces Laguerre-Gauss. Estos descubrimientos dieron lugar a una nueva área de investigación en la óptica moderna [24].

Si bien, el fenómeno de OAM y vórtices ópticos son distintos, estos tienen un traslape en sus conceptos en haces estructurados, tales como una intensidad rotacionalmente simétrica y un incremento de fase que se propaga helicoidalmente alrededor de un eje de propagación [24], tal y como se observa en la figura 2.13. Los haces con vórtices ópticos se caracterizan por contar con una estructura de fase con dependencia azimutal, la cual se expresa mediante el término $\exp(il\varphi)$, donde l es la **carga topológica** del haz, cuyo valor es un número entero el cual establece el número de frentes de ondas helicoidales y cuyo signo determina la quiralidad del haz. El momento angular orbital L de los fotones que forman haces con fases helicoidales y poseen una carga topológica l es de $|l|\hbar$ [28]. Si bien los haces con vórtices ópticos pueden presentar cargas topológicas de distinto signo, lo que altera la dirección del giro de su fase, su

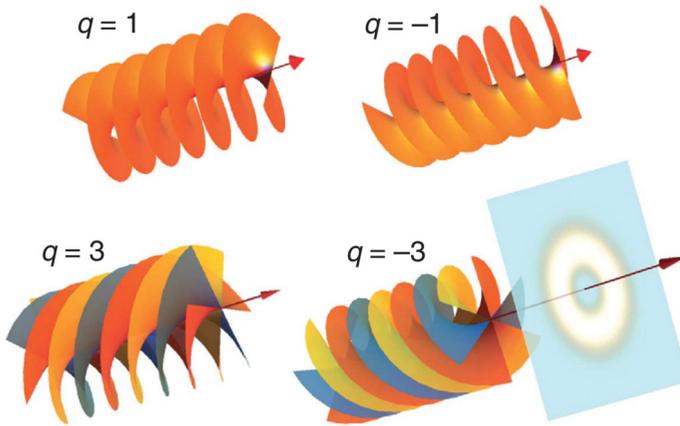


Figura 2.13: Fases helicoidales. Nótese como varía el número de fases helicoidales y su quiralidad en función de la carga topológica. Imagen recuperada de [4].

patrón de intensidad permanece igual para cargas con el mismo valor absoluto.

La capacidad de manipulación de haces con OAM impulsó el desarrollo de múltiples aplicaciones, como el entrelazamiento de fotones mediante la carga topológica, pinzas ópticas para la manipulación y retención de partículas, y tecnología de comunicación óptica [24] [28].

Para medir la carga topológica de los haces con vórtices ópticos, y con ello su OAM, se utilizan técnicas de interferogramas para generar patrones de difracción e interferencia. El número de líneas y rejillas generadas permite determinar la carga topológica del haz. Además, el uso de moduladores espacial de luz permite realizar transformaciones de fase, lo que facilita la reconstrucción de la fase objetivo y la detección precisa de la carga topológica y su OAM [28].

2.3 Generación de haces estructurados con moduladores espaciales de luz

Una de las técnicas más utilizadas para la generación de haces es la modulación de luz con hologramas generados por computadoras, utilizando **moduladores espaciales de luz**, comúnmente conocidos como dispositivos **SLM** [22].

Los SLM utilizan cristal líquido sobre silicio, lo que les permite proyectar hologramas computacionales capaces de modular ópticamente la luz que incide sobre ellos. Los cristales líquidos (LC) son materiales compuestos por moléculas alargadas que presentan un estado intermedio entre las fases líquida y sólida, combinando fluidez con cierto orden estructural, lo que les otorga propiedades únicas. Gracias a su forma, estas moléculas son capaces de formar dipolos los cuales responden a estímulos eléctricos [22].

En esencia, un SLM es una pantalla compuesta por millones de celdas rellenas de

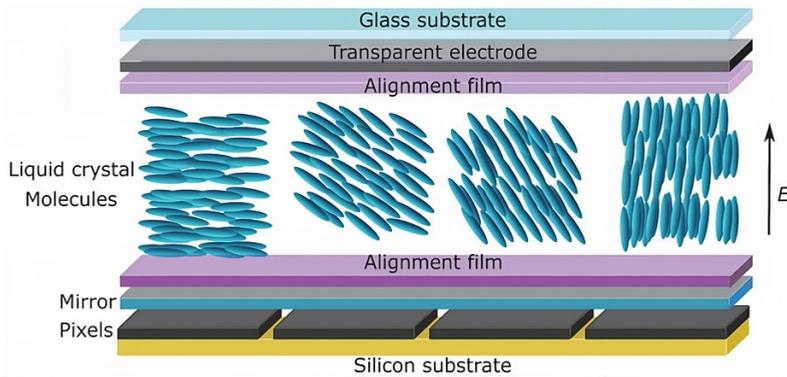


Figura 2.14: Estructura de un SLM. El valor de E denota el campo eléctrico aplicado entre electrodos. Imagen recuperada de [22].

LC, organizadas en un arreglo de píxeles, los cuales pueden ser controlados uno a uno, tal y como se presenta en la figura 2.14. El índice de refracción de los píxeles varía en función de la orientación de las moléculas de LC. Debido a esto, es posible modificar la luz que incide sobre la pantalla.

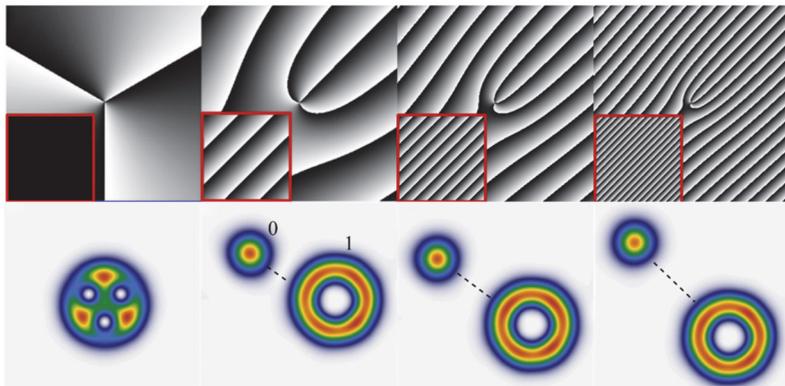


Figura 2.15: Haz generado mediante hologramas. En la parte superior se muestran los hologramas utilizados, mientras que en la parte inferior se observa el haz resultante para cada uno de ellos. Nótese cómo, al agregar y modificar la rejilla blazed, los distintos órdenes de difracción del haz se separan. Imagen recuperada de [22].

Estos dispositivos son fáciles de manejar, debido a que pueden ser utilizados como una segunda pantalla de monitor. La rejilla de cristal líquido se encuentra entre dos capas de electrodos transparentes (véase figura 2.14), los cuales son capaces de alinear las moléculas de LC a un ángulo proporcional al voltaje aplicado entre estos [22]. Para modificar el voltaje entre los electrodos, se despliega una imagen en escala de grises

en codificación 8-bit sobre la pantalla. El voltaje aplicado se encuentra directamente relacionado a la intensidad de la escala de grises de la imagen. Así pues, la modulación de la luz recae únicamente en encontrar la imagen apropiada (el holograma digital), para obtener el resultado deseado. En la figura 2.15 se muestran ejemplos de hologramas desplegados en un SLM, así como los haces modulados [22]. El modulador espacial de luz funciona como un retardador lineal de fase y un material birrefringente, con el que es posible modificar la amplitud, fase y polarización de cualquier haz que incide sobre este.

Sin embargo, los SLM no son 100 % eficientes debido a la pixelación de su rejilla, la cual difracta la luz en múltiples direcciones, generando varios órdenes de difracción. Para obtener únicamente el haz modulado, es necesario aislar el orden de difracción que contenga la información deseada. Para ello, se añade una *rejilla blazed* al holograma proyectado en el SLM, de modo que los distintos órdenes queden separados y puedan filtrarse físicamente [22]. En la figura 2.15 se observa como al modificar la rejilla blazed en el holograma los órdenes de difracción se separan.

Uno de los montajes que se sigue para generar y filtrar los haces modulados consiste en un sistema tipo telescopio, conocido como configuración $4f$, que permite observar el haz en el plano de Fourier. A esta configuración se le coloca una apertura con la cual se limita el paso de la luz, filtrando el orden deseado. En la figura 2.16 se presenta este montaje.

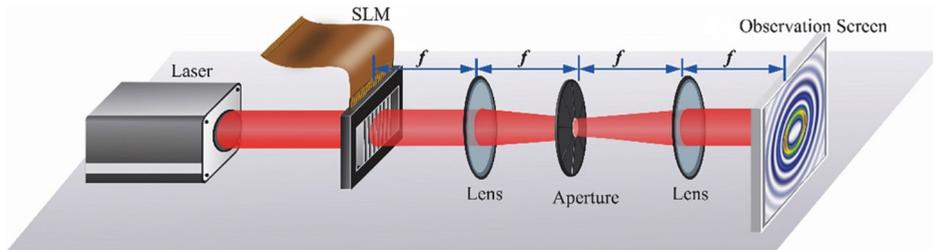


Figura 2.16: Montaje $4f$. Imagen recuperada de [22].

Redes neuronales artificiales

3.1 Introducción

Las **Redes Neuronales Artificiales** (ANN por sus siglas en inglés), son modelos computacionales que toman inspiración en la estructura de las redes neuronales biológicas, al estar compuestas por diversos nodos, también llamados neuronas, los cuales se encuentran interconectados entre sí, similar a como el sistema nervioso se encuentra formado [32]. Gracias a este modelo, las ANN desarrollan conexiones organizadas en diversos niveles [13], lo que les permite adquirir un aprendizaje jerarquizado, capaz de captar patrones de diversa complejidad [12]. Este tipo de arquitectura jerárquica impulsó el desarrollo de un área clave de la **inteligencia artificial (IA)** conocida como **Deep Learning (DL)** [13].

La inteligencia artificial es un campo de estudio perteneciente a las ciencias computacionales. El término “inteligencia artificial” fue acuñado por John McCarthy, quien, en el verano de 1956, se reunió en el Dartmouth College junto a otros científicos para debatir lo siguiente:

The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. [5] [2]

En esta conferencia se sentaron las bases de lo que hoy conocemos como IA. Actualmente, existen diversas definiciones de inteligencia artificial, siendo algunas de estas como:

- La capacidad de las máquinas de automatizar tareas intelectuales que normalmente realizan los humanos [5].

- El desarrollo de máquinas de manera que se comporten como si fuesen inteligentes [8].
- El estudio de cómo lograr que las computadoras realicen tareas en las que, por el momento, los seres humanos son superiores [8].

Aunque estas definiciones presentan similitudes, cada una resalta matices distintos. En esencia, todas coinciden en que el objetivo principal de la inteligencia artificial es crear sistemas o máquinas que sean o aparenten ser inteligentes, permitiéndoles realizar una amplia variedad de tareas. Hoy en día, la inteligencia artificial ha ganado un gran impacto, teniendo relevancia y utilidad en actividades que realizamos en el día a día.

Una rama importante del campo de la inteligencia artificial es el **aprendizaje de máquina** o **machine learning (ML)**, el cual consiste en la extracción de características de un conjunto de datos sin procesar (**dataset**) utilizando algoritmos computacionales, los cuales permiten adquirir información relevante sobre este conjunto de datos. Los algoritmos de ML requieren de instancias caracterizables, es decir, que cada una de las instancias que conforman al dataset puedan ser descritas por características (**features**) bien definidas [12]. Usualmente, el conjunto de características que definen a las instancias se organizan como un vector de características y son estos los que permiten que el algoritmo adquiera información sobre los datos. La correcta elección de características en un conjunto de datos permitirá que los algoritmos de ML puedan aprender la mayor cantidad de información sobre estos, reconociendo patrones y pudiendo utilizar dicha información para resolver diversas tareas como clasificación, regresión, predicción, etc.

Existen distintos paradigmas bajo los cuales los algoritmos de ML pueden aprender, de los cuales, los principales son el **aprendizaje supervisado**, el **aprendizaje no supervisado** y el **aprendizaje por refuerzo**. A continuación, se explican estos paradigmas.

Aprendizaje supervisado: Cuando un conjunto de datos tiene variables objetivo (*targets*) o etiquetas (*labels*), se dice que sigue el paradigma de aprendizaje supervisado. El objetivo de estos modelos es encontrar una función de mapeo entre el conjunto de datos a estas variables [6], donde estas servirían como guías para obtener un resultado esperado, estableciendo relaciones entre el conjunto de características de los datos [12].

Aprendizaje no supervisado: Cuando un conjunto de datos no posee variables objetivo ni etiquetas, se dice que sigue el paradigma de aprendizaje no supervisado. Los modelos entrenados con este enfoque buscan identificar relaciones, patrones o distribuciones de probabilidad que caracterizan al dataset [12].

Aprendizaje por refuerzo: Este paradigma se centra en una estrategia de prueba y error [12], donde el modelo adquiere información del entorno. Si la información adquirida por el modelo es satisfactoria se premia y se refuerza dicha interacción (*reward*), en caso contrario se penaliza y se corrigen futuras interacciones (*penalty*) con el entorno [27] [20].

Las ANN son modelos que pertenecen al deep learning, un subcampo del machine learning [12]. Como se mencionó, los modelos de ML buscan extraer información de un conjunto de datos al identificar patrones, distribuciones o relaciones, *representando* la información de tal forma que se favorezca el aprendizaje. Por su parte, el deep learning se basa en el concepto de *aprendizaje por representaciones jerárquicas*, en el que la información se organiza en múltiples capas, donde las capas superficiales capturan las características generales y en las capas más profundas se extrae información más compleja o abstracta. El nombre de deep learning hace referencia a la profundidad de representaciones en la que se organiza la información [5].

Los modelos de ANN, en su mayoría, son modelos de aprendizaje supervisado y que requieren de una gran cantidad de datos para obtener buenos resultados [27]. El objetivo de estos modelos, al igual que de otros modelos de ML, es aprender de la información presente en los datos, reconociendo patrones y estableciendo relaciones entre estos. Para lograrlo, las conexiones entre los nodos, denominadas **pesos** (*weights*) y *bias*, se ajustan de manera que la diferencia entre el resultado obtenido y el resultado esperado, denominado *error*, disminuya progresivamente. A través de la retroalimentación del error, el modelo ajusta los pesos en sus nodos, mejorando su rendimiento en posteriores iteraciones [13].

A continuación, se presentan los conceptos y la teoría necesaria para entender las redes neuronales artificiales utilizadas en este trabajo.

3.2 Redes neuronales densas

3.2.1 Neurona artificial

Las ANN son modelos computacionales inspirados en el sistema nervioso del ser humano [3]. Se denomina *neurona artificial* a la unidad fundamental con la que estos modelos se encuentran formados. El funcionamiento de las neuronas artificiales se basa en como operan las neuronas biológicas, con las cuales el sistema nervioso se encuentra formado. La neurona biológica es capaz de recibir y enviar señales eléctricas a otras neuronas [20].

En la figura 3.1 se pueden observar las principales estructuras de una neurona biológica. Cada parte de esta cumple con una función en específico, las cuales permiten el correcto funcionamiento del sistema nervioso. Las *dendritas* se encargan de recibir los estímulos eléctricos de otras neuronas, el *soma* es el encargado de recibir los impulsos de las dendritas para posteriormente producir un potencial de acción (generar una señal eléctrica) y mandarla a lo largo del *axón*, para luego ser recibida por otras dendritas o por algún tejido nervioso [20].

El primer modelo matemático de una neurona se remonta al año de 1943, con los investigadores Warren McCulloch y Walter Pitts [12]. En su artículo propusieron un modelo simple pero que cumplía con las características mencionadas de una neurona, pudiendo establecer interconexiones entre estas y teniendo un umbral de activación definido [18]. Este modelo sería conocido como **la neurona de McCulloch-Pitts** y serviría como base para la formulación de otros modelos de neuronas artificiales [20].

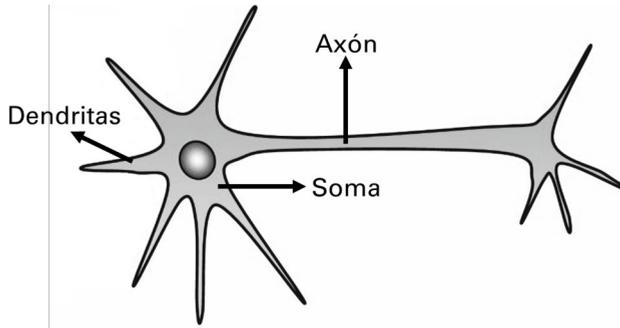


Figura 3.1: Representación de una neurona biológica. Imagen recuperada y modificada de [20].

El modelo de McCulloch-Pitts consiste en la suma de un conjunto de valores x_i ponderados por coeficientes w_i . El resultado de esta suma se evalúa en la función de activación escalón, la cual permite obtener una salida \hat{y} en función de los valores de entrada recibidos.

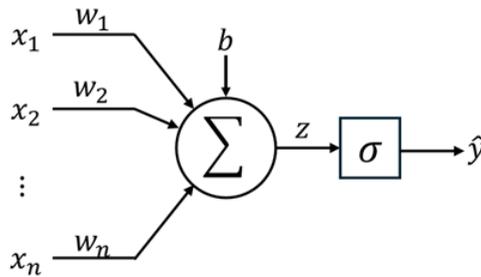


Figura 3.2: Modelo de neurona artificial.

El ponderado de \mathbf{X} , el vector que contiene los valores x_i , con los pesos \mathbf{W} , el vector que contiene los pesos w_i , simulan el proceso de adquisición de señales eléctricas realizado por las dendritas en una neurona biológica. Mediante la suma ponderada de estos valores se obtiene un valor de salida con una función de activación, de manera similar a como el soma de la neurona biológica genera una señal eléctrica en función de los impulsos recibidos por las dendritas, para posteriormente ser enviada a lo largo del axón a otra neurona [20].

Matemáticamente, el valor obtenido \hat{y} por la neurona de McCulloch-Pitts se obtiene de la siguiente forma

$$z = \sum_{i=1}^n w_i x_i \tag{3.1}$$

$$\hat{y} = H(z) \tag{3.2}$$

donde x_i son las señales de entrada (*inputs*), w_i son los pesos sinápticos (*weights*) y n es el número de entradas. Posteriormente el potencial de acción z es evaluado en una función de activación H que proporcionará el valor de salida \hat{y} (*output*) [20]. Para la neurona de McCulloch-Pitts la función de activación utilizada era la función *Heaviside*, la cual se define de la siguiente manera

$$H(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (3.3)$$

Este modelo lineal era capaz de separar dos categorías a partir de los valores de entrada [12]. Para ello era necesario establecer los pesos \mathbf{W} y b manualmente. Esto cambiaría en modelos de neuronas artificiales posteriores.

3.2.2 Perceptrón

Posterior a la salida de la neurona de McCulloch-Pitts, en 1958 Rosenblatt propuso un nuevo modelo de neurona artificial al que llamaría **perceptrón** [23]. Rosenblatt, interesado en conocer como el sistema nervioso era capaz de guardar información, utilizando trabajo realizado anteriormente por matemáticos y neurocientíficos, formuló el perceptrón en términos de probabilidad más que en lógica simbólica como se venía haciendo [23]. Uno de los primeros objetivos del perceptrón de Rosenblatt fue el de identificar patrones a través de un modelo basado en el funcionamiento de la retina [20] [23].

La formulación matemática del perceptrón tenía una estructura parecida a la de la neurona de McCulloch-Pitts, sin embargo, eliminó algunas de las limitaciones que esta última tenía. Uno de los grandes avances que tuvo el perceptrón, y que lo destacó sobre otros modelos, fue que los pesos sinápticos podían ser aprendidos por el mismo modelo [20]. Para ello seguía una estrategia de aprendizaje supervisado, lo que implicaba que debía existir una salida deseada para cada valor de entrada con la que se entrenaba el modelo [20]. El algoritmo con la que se modificaban los pesos se basaba en la *regla de aprendizaje de Hebb* o *aprendizaje Hebbiano* [3], el cual suele resumirse con la siguiente frase nemotécnica:

Cells that fire together, wire together.

Sin embargo, este suele ser más complicado que eso, tal y como se analiza en Keyser y Gazola (2014) [14]. Hebb [19], en su libro menciona lo siguiente:

Cuando el axón de la célula A está lo suficientemente cerca para excitar a la célula B y participa repetida o persistentemente en su activación, se produce algún proceso de crecimiento o cambio metabólico en una o ambas células, de modo que aumenta la eficiencia de la célula A, como una de las células que activa a la célula B.

En la figura 3.3 se ilustra esta regla. Matemáticamente, si la salida producida por el perceptrón es la misma que la salida deseada, los pesos y el umbral de activación

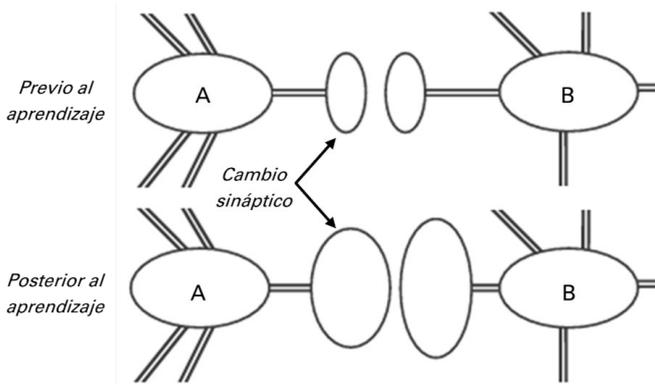


Figura 3.3: Aprendizaje Hebbiano. Imagen recuperada y modificada de [3].

se mantienen igual, si no, estos son ajustados proporcionalmente a los valores de entrada [20]. Matemáticamente, siguen la siguiente relación

$$W^{\text{actual}} = W^{\text{previo}} + \alpha (y^{(k)} - \hat{y}^{(k)}) \cdot X \quad (3.4)$$

donde $y^{(k)}$ y $\hat{y}^{(k)}$ son las salidas esperadas y obtenidas de la k -ésima muestra respectivamente, y α es una constante denominada *tasa de aprendizaje*, la cual regula el cambio entre los pesos al establecer una magnitud en el refuerzo de estos. Si bien existen varias estrategias para inicializar los pesos, para el perceptrón de Rosenblatt los pesos se inicializan de manera aleatoria con valores pequeños [20].

Sin embargo, la opinión sobre este modelo pronto se vería afectada negativamente. En 1969, Minsky y Papert, criticaron duramente al perceptrón en su libro *“Perceptrons: An introduction to Computation Geometry”* [20] en el cuál demostraron que el perceptrón era incapaz de resolver el problema *XOR* y, en general, cualquier tipo de problema no lineal. Esto provocaría una recesión en la investigación de este tipo de modelos computacionales.

3.2.3 Perceptrón multicapa

No sería hasta 1986 en el que Rumelhart, Hinton y Williams, propondrían un nuevo modelo computacional inspirado en las neuronas biológicas y en el concepto de *conexionismo* (*connectionism*), también llamado *procesamiento distribuido en paralelo* [12], en su libro *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* [25]. En este modelo, las señales de entrada se propagarían a lo largo de capas constituidas por unidades de neuronas artificiales, basadas en el perceptrón, hasta llegar a una capa de salida, tal y como se muestra en la figura 3.4.

Esta arquitectura, llamada **perceptrón multicapa (MLP)**, cuenta con al menos una capa intermedia entre la capa de entrada y la capa de salida, llamada capa oculta. Este tipo de modelo se basa en el concepto de *representación distribuida*, el

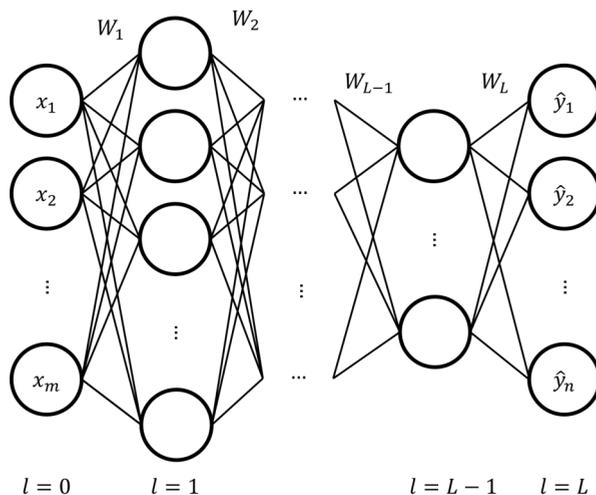


Figura 3.4: Diagrama de perceptrón multicapa.

cual establece que es posible definir a un sistema como un conjunto de características jerárquicamente relacionadas, las cuales pueden ser obtenidas mediante variables relacionadas de manera sencilla [12]. En este tipo de arquitectura, las capas ocultas serían las encargadas de extraer la mayoría de información relacionada al comportamiento del sistema para posteriormente ser codificado con los pesos de las capas posteriores.

No solo fue el tipo de arquitectura que destacó el trabajo de Rumerlhart y colaboradores, sino también el algoritmo de aprendizaje propuesto, es decir, la forma en como los pesos de la red se actualizan, favoreciendo el aprendizaje de los datos. Esta arquitectura junto con el algoritmo de aprendizaje, llamado **retropropagación (backpropagation)**, se convertirían en una herramienta increíblemente poderosa [25].

Retropropagación (Backpropagation)

El algoritmo de retropropagación se puede separar en dos etapas. La primera etapa, llamada **propagación hacia delante (forward propagation)** consiste en la evaluación de las variables x_m de una instancia de un conjunto de datos en una red MLP, la cual irá modificando estos datos a lo largo de sus capas con los pesos de cada nodo, hasta obtener un resultado en su capa final. Esta etapa se centra en la obtención de la respuesta del modelo ante los datos que se están estudiando. A continuación, se describe de manera más detallada el algoritmo de *forward propagation*.

Sea una red neuronal multicapa, o red MLP, de L capas que recibe como entrada un vector de m variables, (véase figura 3.4), donde la capa de entrada es la capa $l=0$, y la capa de salida es $l=L$. El vector resultante de la capa l , denotado como $\mathbf{A}^{[l]}$, donde el superíndice indica la pertenencia a la capa l , tiene tamaño $(j^{[l]}, 1)$, donde $j^{[l]}$ indica el número de neuronas (nodos) que constituyen la capa l -ésima. De lo anterior

se sigue que el vector de entrada puede ser denotado como $\mathbf{A}^{[0]}$ y el vector de salida como $\mathbf{A}^{[L]}$.

Sea la matriz de pesos de la capa l $\mathbf{W}^{[l]}$, tal que $\mathbf{W}^{[l]} = (w_{ji})$, donde w_{ji} representa el peso entre el j -ésimo nodo de la capa l y el i -ésimo nodo de la capa $l - 1$. La matriz $\mathbf{W}^{[l]}$ tiene dimensiones $(j^{[l]}, j^{[l-1]})$. Dado que no existen pesos para la capa de entrada, para $\mathbf{W}^{[l]}$ el valor de l comienza en 1. La salida obtenida $\mathbf{A}^{[l]}$ estará dado por las siguientes operaciones

$$\mathbf{Z}^{[l]} = \mathbf{W}^{[l]} \mathbf{A}^{[l-1]} + \mathbf{b}^{[l]} \quad (3.5)$$

$$\mathbf{A}^{[l]} = \sigma^{[l]} \left(\mathbf{Z}^{[l]} \right) \quad (3.6)$$

donde $\mathbf{b}^{[l]}$ es el *bias* o *threshold*, el cual se define como el valor del bias $b^{[l]}$ multiplicado por un vector compuesto de unos de tamaño $(j^{[l]}, 1)$, y $\sigma^{[l]}(\cdot)$ es la función de activación de la capa l , la cual se aplica elemento a elemento (*element-wise*) al vector $\mathbf{Z}^{[l]}$ para así obtener el vector $\mathbf{A}^{[l]}$ [20].

Estas operaciones modifican los valores de $\mathbf{A}^{[l-1]}$ de manera geométrica, conocida como *transformación afín* [5]. Las transformaciones afines son combinaciones de transformaciones lineales (traslaciones, rotaciones y escalamientos), tal y como se puede observar en la figura 3.5. Así pues, $\mathbf{Z}^{[l]}$ no es más que una transformación afín en un espacio dimensional de $\mathbf{A}^{[l-1]}$. Luego, la función de activación $\sigma^{[l]}$ permite implementar transformaciones geométricas no lineales más complejas, resultando en espacios de hipótesis muy ricos para la interpretación de los datos [5].

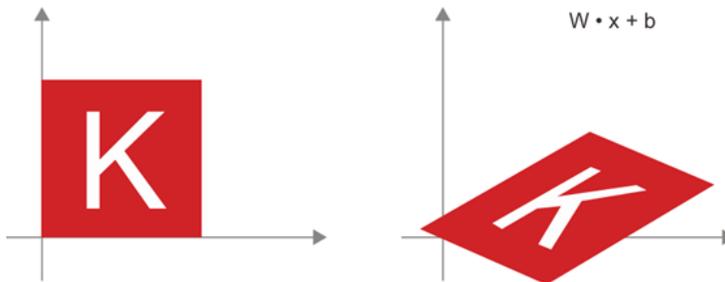


Figura 3.5: Transformación afín sobre un conjunto de datos K . Imagen recuperada de [5].

Con la evaluación de los datos en la capa final, se obtiene la respuesta de la red a los datos iniciales, la cual se ve expresada en $\mathbf{A}^{[L]}$. Posteriormente, se compara el resultado obtenido $\hat{\mathbf{Y}}$, donde $\hat{\mathbf{Y}} = \mathbf{A}^{[L]}$, con la respuesta esperada \mathbf{Y} mediante una **función costo** o **función error** [12]. Este resultado es el error E de la red, el cual será utilizado para realizar el ajuste de los pesos en futuras iteraciones.

La segunda etapa del algoritmo de retropropagación, llamada **propagación hacia atrás** (**backward propagation**), modifica los pesos de la red mediante el algoritmo

de **gradiente descendiente**. Para ello, se utiliza el error E calculado al final de la etapa de forward propagation. Los valores $\mathbf{W}^{[l]}$ y los valores umbral $b^{[l]}$ son modificados proporcionalmente a la tasa de cambio del error E con respecto a $\mathbf{W}^{[l]}$ o $b^{[l]}$, respectivamente, mediante las siguientes ecuaciones iterativas

$$\mathbf{W}_{i+1}^{[l]} = \mathbf{W}_i^{[l]} - \alpha \frac{\partial E}{\partial \mathbf{W}^{[l]}} \quad (3.7)$$

$$b_{i+1}^{[l]} = b_i^{[l]} - \alpha \frac{\partial E}{\partial b^{[l]}} \quad (3.8)$$

donde el subíndice i indica la iteración y α es la tasa de aprendizaje o *learning rate*.

Para obtener los valores de los gradientes de E , es necesario utilizar el siguiente conjunto de ecuaciones, las cuales se obtienen de derivar la función costo y aplicar regla de la cadena sobre las diversas variables. Estas ecuaciones son

$$\frac{\partial E}{\partial \mathbf{W}^{[l]}} = \frac{\partial E}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{W}^{[l]}} \quad (3.9)$$

$$\frac{\partial E}{\partial b^{[l]}} = \frac{\partial E}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial b^{[l]}} \quad (3.10)$$

$$\frac{\partial E}{\partial \mathbf{Z}^{[l]}} = \frac{\partial E}{\partial \mathbf{A}^{[l]}} \frac{\partial \mathbf{A}^{[l]}}{\partial \mathbf{Z}^{[l]}} \quad (3.11)$$

$$\frac{\partial E}{\partial \mathbf{A}^{[l-1]}} = \frac{\partial E}{\partial \mathbf{Z}^{[l]}} \frac{\partial \mathbf{Z}^{[l]}}{\partial \mathbf{A}^{[l-1]}} \quad (3.12)$$

El algoritmo de propagación hacia atrás ajusta cada variable en proporción a la tasa de cambio del error, con el objetivo de reducir E . En otras palabras, las variables se actualizan siguiendo la dirección opuesta al gradiente de E , lo que garantiza la minimización del error. En la figura 3.6 se observa un ejemplo del descenso del error.

Un ejemplo útil para entender cómo actúa una red neuronal multicapa sobre un conjunto de datos es imaginar dos hojas de papel de distinto color que se arrugan juntas para tener una sola bola de papel. Esta bola deformada representaría los datos de entrada. Las operaciones de la red neuronal se interpretarían como una serie de transformaciones geométricas que se realizan para desdoblar y alisar la bola de papel, con el fin de restaurarlas a su estado original: dos hojas separadas por su color [5].

3.2.4 Funciones de activación

Una de las principales ventajas de la arquitectura MLP frente a otras arquitecturas de redes neuronales más antiguas es su capacidad para realizar múltiples tareas, como el reconocimiento de patrones, la clasificación de categorías, la predicción de valores, el análisis de series temporales y, en términos generales, actuar como una función aproximadora universal [20].

El amplio repertorio de tareas que las redes MLP pueden realizar no solo depende de la cantidad de salidas que el sistema tenga, sino también de la elección adecuada de la función de activación en las capas intermedias y, sobre todo, de la correcta elección de la función de activación en la capa de salida

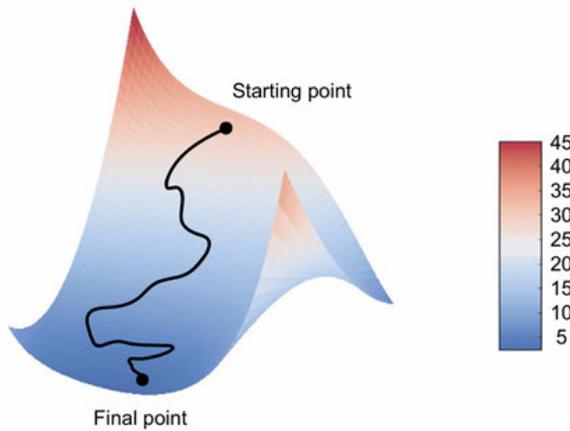


Figura 3.6: Descenso del error E . El entrenamiento en la red permite que el error disminuya en cada iteración.

La elección de las funciones de activación de las capas intermedias permitirá la reducción de iteraciones para la convergencia del error en la red, así como su optimización, mientras que las funciones de activación de la última capa estarán determinadas por el tipo de problema que se desea resolver [31].

A continuación, se presentan algunas de las funciones de activación más comunes.

1. **Sigmoide:** La función de activación sigmoide es una de las primeras funciones de activación utilizadas. Esta función, dada por la expresión

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.13)$$

posee un codominio acotado en el intervalo $(0, 1)$, siendo asintótica en los extremos de este, teniendo una forma de S alargada y una parte casi lineal para los valores de x cercanos a 0. Esta función es usualmente usada en la capa de salida en problemas de clasificación binaria, dando como resultado una probabilidad de que una determinada instancia pertenezca a una clase 0 o a una clase 1 [21]. Una de las desventajas que presenta la función de activación sigmoide es que puede presentar una saturación para valores extremos, lo cual produce un problema conocido como *desvanecimiento del gradiente* [7], el cual provoca que el gradiente se vuelve relativamente pequeño, dificultando el cambio en los pesos de la red y por lo tanto su convergencia. En la figura 3.7 se observa la gráfica de esta función.

2. **Tanh:** La función tangente hiperbólica, denotada como \tanh , es similar a la función sigmoide, teniendo la misma forma, pero variando el codominio de esta, siendo una función con codominio en el intervalo de $(-1, 1)$. Dicha función, se

encuentra dada por la siguiente expresión

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.14)$$

Esta función también presenta el problema del desvanecimiento del gradiente. De igual forma que la sigmoide, esta función se utiliza en la capa de salida para clasificación binaria, siendo ahora el intervalo de separación entre las clases de -1 y 1 [7]. En la figura 3.7 se puede observar la gráfica de esta función.

3. **ReLU**: La función de *unidad lineal rectificadora*, o simplemente *ReLU*, es la función de activación predilecta en el estado del arte para las capas ocultas, debido a su simplicidad y su alto rendimiento en las redes neuronales [7]. La función ReLU, se define de la siguiente forma

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (3.15)$$

Debido a su simplicidad, permite realizar cálculos rápidos a comparación de otras funciones que presentan mayor complejidad o número de operaciones en su expresión. Con un codominio acotado entre $[0, \infty)$, la función ReLU presenta el problema del desvanecimiento de gradiente únicamente para valores negativos, sin embargo, empíricamente se ha observado el gran desempeño de esta función como función de activación para capas intermedias de numerosos tipos de problemas [7] [15]. En la figura 3.7 se observa la gráfica de esta función.

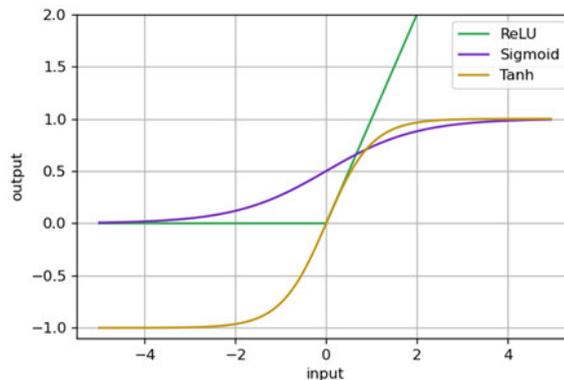


Figura 3.7: Gráficas de funciones de activación.

4. **Softmax**: La función *softmax* es una extensión de la función sigmoide, utilizada para la clasificación multiclase [11]. Su propósito es mapear un conjunto de valores no normalizados x_j a un vector de probabilidades, donde cada valor

representa la probabilidad de pertenecer a una clase específica [11]. La función se define por la siguiente expresión:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \tag{3.16}$$

donde x_i es uno de los n valores a normalizar. El denominador funciona como una constante que permite normalizar el vector de valores x_j . Así pues, $\text{softmax}(x_i)$ representa la probabilidad de x_i de pertenecer a la clase i . Por su definición matemática, la función softmax es una función no lineal que mapea los valores numéricos más altos a probabilidades más altas. En la figura 3.8 se puede observar como la función softmax transforma un conjunto de valores numéricos a una probabilidad.

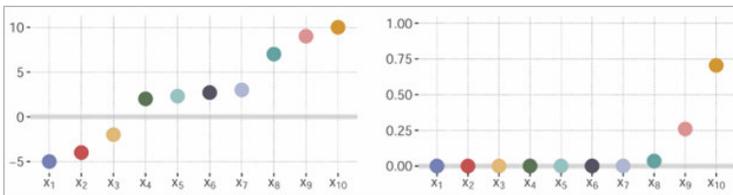


Figura 3.8: Mapeo de puntajes obtenidos de pertenecer a una determinada clase (izquierda) a un vector de probabilidades utilizando softmax (derecha). Imagen recuperada de [11].

3.2.5 Funciones costo

Las funciones costo, también llamadas funciones error, permiten evaluar el desempeño del modelo al calcular las diferencias entre los resultados obtenidos y los deseados. El valor resultante de la función costo permite observar el aprendizaje de la red a lo largo del entrenamiento [31].

Gracias al cambio de los pesos \mathbf{W} y b debido al entrenamiento se minimiza el error E , reflejándose como una reducción en la salida de la función costo. La elección de la función costo dependerá del tipo de problema específico a resolver, de manera análoga a la función de activación [31], su correcta elección penalizará de mejor manera los errores obtenidos por la red, permitiendo un mejor desempeño. A continuación se mencionan las funciones costo más usadas.

1. **Error Cuadrático Medio:** El error cuadrático medio, abreviado como **MSE** por sus siglas en inglés de *Mean Squared Error*, mide el promedio de la diferencia cuadrada entre el valor $\hat{\mathbf{Y}}$ y el valor \mathbf{Y} . Matemáticamente, el MSE se define como

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \tag{3.17}$$

Un valor de 0 indicaría que el modelo ajustó a la perfección los n valores correspondientes para cada instancia i , por el contrario, valores elevados en la métrica indicaría una gran diferencia entre las predicciones y los valores reales. Al ser un promedio, el MSE es sensible cuando existen errores grandes en el modelo. La métrica de MSE, también conocida como *L2 Loss* [31], no solo es utilizada como una función costo, sino también como métrica de evaluación del desempeño del modelo.

2. **Error Absoluto Promedio:** El error absoluto promedio, **MAE** por sus siglas en inglés de *Mean Absolute Error*, es una métrica que mide el promedio de las diferencias absolutas entre el valor predicho $\hat{\mathbf{Y}}$ y el valor real \mathbf{Y} . Matemáticamente esta métrica se define como

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (3.18)$$

El MAE, también llamada como *L1 Loss* [31], posee propiedades similares a la MSE, siendo menos susceptible a errores grandes, pero perdiendo sus propiedades de suavidad y diferenciación.

3. **Pérdida de entropía cruzada binaria:** La pérdida de entropía cruzada binaria o **BCE**, por sus siglas en inglés de *Binary Cross-Entropy Loss* es una métrica utilizada para problemas de clasificación binaria [31]. Esta métrica mide la disimilitud entre dos clases, usualmente etiquetadas como 0 y 1. Matemáticamente esta función se define como

$$\text{BCE} = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (3.19)$$

donde \hat{y}_i es el valor predicho para la instancia i , sin embargo, dado que en BCE el valor predicho puede ser entendido como una probabilidad de pertenecer a una determinada clase (0 o 1), también puede denotarse como p_i . De igual forma, el error es minimizado cuando el valor predicho \hat{y}_i y el valor real y_i son iguales. El uso de la BCE para problemas de clasificación binaria permite un mejor desempeño del modelo al penalizar correctamente las diferencias entre las predicciones y las etiquetas reales.

4. **Pérdida de entropía cruzada categórica:** La pérdida de entropía cruzada categórica, abreviada como **CCE** por sus siglas del inglés *Categorical Cross-Entropy Loss*, es una métrica de medición para problemas de categorización de k clases [31]. Esta métrica mide la disimilitud entre las probabilidades predichas de pertenecer a las distintas clases con respecto a los valores reales. Esta métrica es la generalización de la BCE. Matemáticamente, se define como

$$\text{CCE} = - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k y_{i,j} \log(p_{i,j}) \quad (3.20)$$

donde $y_{i,j}$ y $\hat{y}_{i,j}$ están en codificación *one-hot*. Un modelo de clasificación categórica predice, para cada instancia i , la probabilidad de pertenencia a las distintas k clases, generando un vector de probabilidades de tamaño $1 \times k$. Por otro lado, la etiqueta real de la instancia i también será un vector de tamaño $1 \times k$, donde todos los elementos serán 0 excepto en la posición correspondiente a la clase a la que pertenece, donde será 1. El producto de $y_{i,j} \log(\hat{y}_{i,j})$ devuelve el logaritmo de la probabilidad de que la instancia i pertenezca a la clase j . Si el modelo predice a la perfección esa instancia se tiene que $\hat{y}_{i,j} = 1$, luego $\log(\hat{y}_{i,j}) = 0$ por lo que no contribuye en el error.

3.2.6 Técnicas de regularización

Hasta el momento, se ha visto el proceso que las redes MLP siguen para adquirir información a partir de un conjunto de datos de entrenamiento, aprendiendo patrones y relaciones que los caracterizan. El algoritmo de backpropagation provoca que el error disminuya en cada iteración, viéndose reflejado en la disminución de la salida de la función costo. Sin embargo, uno de los principales retos que se tiene con este tipo de modelos, y en general con modelos de ML, es que tengan un buen desempeño en datos no vistos y sean capaces de **generalizar** y extender la información aprendida a otros datos del mismo tipo [12].

Al entrenar una red MLP se observará que después de ciertas iteraciones las predicciones sobre los datos de entrenamiento empiezan a mejorar. Sin embargo, al realizar predicciones sobre datos no vistos, con las mismas características que los datos de entrenamiento, se observará que el desempeño sobre estos no son tan buenos como con los de entrenamiento. Esta situación es conocida como **sobreajuste (overfitting)**.

Se dice que un modelo tiene sobreajuste cuando existe una brecha (*gap*) entre el error de los datos de entrenamiento y el error de los datos no vistos [12]. El sobreajuste ocurre debido a que en el proceso de entrenamiento la red memoriza la respuesta a ciertos estímulos de entrada [20], sobreoptimizando patrones que poseen los datos de entrenamiento, pero perdiendo la generalización sobre otros datos del mismo tipo [5].

De manera opuesta, cuando un modelo no logra obtener resultados óptimos sobre los datos de entrenamiento, y tampoco en los datos no vistos, se dice que el modelo se encuentra **subajustado (underfitting)**.

Por lo tanto, en un modelo de MLP se busca un punto medio entre estos dos escenarios. Se desea que un modelo aprenda de los datos de entrenamiento lo suficiente para poder realizar predicciones sobre estos pero sin memorizarlos, conservando la generalidad. En la figura 3.9 se muestran estas tres situaciones.

Existen técnicas para evitar el subajuste y el sobreajuste. Para evitar el subajuste es necesario aumentar la profundidad de la red y el número de iteraciones antes de detener el entrenamiento. El subajuste es fácil de corregir en comparación con el sobreajuste. Las técnicas que se centran en la corrección del sobreajuste se conocen como técnicas de *regularización* [12].

Dado que el error en el **conjunto de entrenamiento** no siempre refleja la capacidad del modelo para generalizar, es necesario evaluarlo con un conjunto de datos que no haya sido utilizado en el entrenamiento. Este conjunto, llamado **conjunto de**

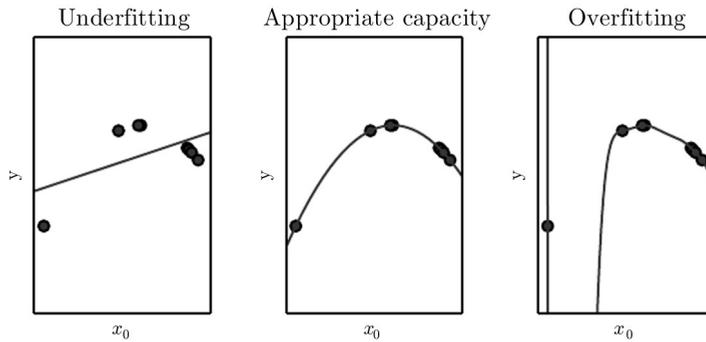


Figura 3.9: Ajustes de modelo a datos de entrenamiento. En la figura se pueden apreciar las 3 situaciones que se pueden dar en el entrenamiento de una red MLP: un subajuste (izquierda), un ajuste óptimo (centro) y un sobreajuste (derecha). Imagen recuperada de [12].

validación, permite estimar el desempeño real del modelo.

A lo largo del entrenamiento se evaluará el error del modelo, tanto en el conjunto de entrenamiento como en el conjunto de validación. Tal y como se puede observar en la figura 3.10, el error en el conjunto de entrenamiento disminuye a lo largo de este, mientras que el error de validación desciende hasta un mínimo, para luego aumentar. Durante el periodo donde el error de validación disminuye, la red aprende las generalidades de los datos hasta llegar a un punto mínimo. A partir de aquí, la red empieza a memorizar patrones particulares de los datos de entrenamiento, aumentando el error en los datos de validación.

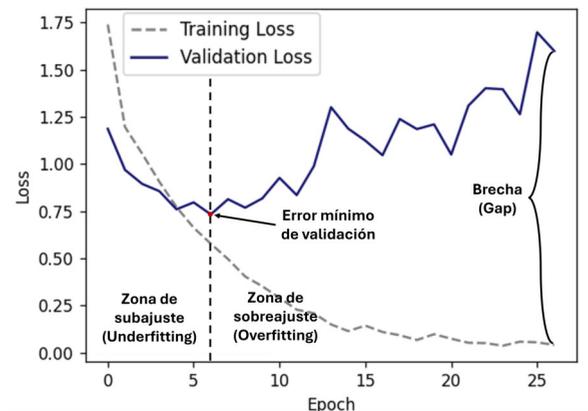


Figura 3.10: Errores de entrenamiento y validación durante el entrenamiento.

Sabiendo que el mejor desempeño del modelo en datos no vistos será en ese punto mínimo, se desea detener el entrenamiento, ya que a partir de aquí el modelo se

sobreajustará al conjunto de entrenamiento.

Existe una técnica de regularización que se centra en lo anterior, detener el entrenamiento en el punto de menor error en el conjunto de validación, llamada **early stopping**. La técnica de early stopping monitorea los valores de error, o cualquier otra métrica que ayude a determinar el desempeño del modelo (revisar sección 3.2.7) a lo largo del entrenamiento, guardando los pesos del modelo que tenga el mejor desempeño. La técnica de early stopping proporciona un margen de épocas o iteraciones antes de detener el proceso de entrenamiento, sin embargo, si durante estas épocas no se encuentra un mejor desempeño, se restauran los pesos del mejor modelo encontrado.

Una vez que se ha encontrado el mejor modelo, y se desea reportar el desempeño de este, no se utilizan los resultados obtenidos con el conjunto de validación. Si bien el modelo no se entrenó directamente con el conjunto de validación, los resultados en validación pueden influir en la elección de hiperparámetros, introduciendo un sesgo en la evaluación del modelo. En su lugar, se utiliza un tercer conjunto de datos conocido como **conjunto de prueba**. El conjunto de prueba, al igual que el de validación, se extrae de la misma distribución de datos que el conjunto de entrenamiento, asegurando que las muestras sean estadísticamente representativas. Realizar la evaluación del desempeño del modelo utilizando un conjunto de datos que no se ha utilizado anteriormente evita un segundo sobreajuste sobre los datos.

Con lo anterior se ha visto que es necesario particionar el dataset en tres conjuntos: entrenamiento, validación y prueba. La literatura sugiere que entre el 50 % y el 90 % de los datos que conforman el dataset sean utilizados para el entrenamiento [20], mientras que el restante se divida para el conjunto de validación y prueba. Existe una partición típica conocida como 70-15-15, en la cual el 70 % de los datos se utiliza para entrenamiento, un 15 % para validación y otro 15 % para prueba o testing. Sin embargo, la partición depende del tamaño del dataset. En conjuntos de datos muy grandes, se suele asignar un mayor porcentaje de datos al conjunto de entrenamiento.

Otra técnica de regularización que corrige la sobreoptimización de valores es la técnica de **dropout**. En el sobreajuste, algunos nodos empiezan a aprender patrones de ruido o relaciones particulares del conjunto de entrenamiento, los cuales reducen la generalización del aprendizaje [5]. Para evitar esto, el dropout implementa a las salidas de las capas de entrada o ocultas *máscaras* que retiran durante una iteración o época un porcentaje de neuronas aleatorias, provocando que no participen en el aprendizaje [12]. Al *apagar* neuronas en una determinada capa se evita una saturación en la salida de esta, lo que permite a la red mejorar la generalización del aprendizaje. El porcentaje de neuronas que se apagan al agregar la capa de dropout es definido por el usuario mediante un parámetro llamado *dropout rate*. En la figura 3.11 se muestra una representación del efecto del dropout en las capas de la red.

La magnitud de $\mathbf{A}^{[l]}$ es menor cuando se aplica dropout en la capa l en comparación con la red original, ya que un porcentaje de nodos produce una salida de cero. Para compensar esta reducción, la salida resultante de la capa con dropout debe escalarse por el inverso del factor de $1 - \text{dropout rate}$ [5]. Por su parte, para la actualización de pesos los nodos que no contribuyeron durante esa iteración conservan sus valores.

El dropout solo se aplica durante el entrenamiento, durante la parte de prueba todas las neuronas permanecen encendidas. En la figura 3.12 se compara la función

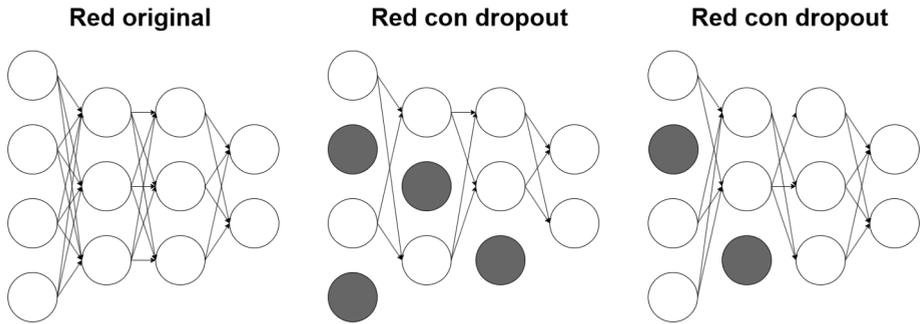


Figura 3.11: Efecto de dropout en las capas de la red. Los nodos grises han sido apagados aleatoriamente por la máscara de dropout.

costo de un modelo entrenado con y sin dropout. Podemos observar como esta técnica mejora el desempeño, pero sobre todo dificulta el sobreajuste sobre los datos de entrenamiento.

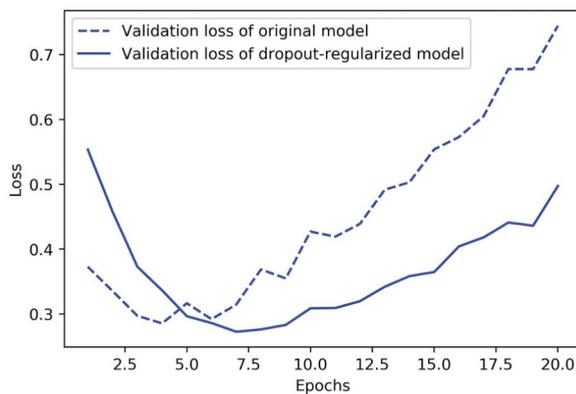


Figura 3.12: Efecto de dropout en el error de validación. Imagen recuperada de [5].

3.2.7 Métricas de evaluación

Una vez que las redes se han entrenado con los datos de entrenamiento, es necesario utilizar métricas para evaluar el desempeño de estas sobre datos no vistos y evitar el sobreajuste sobre los datos. Las métricas de evaluación ayudan a determinar el aprendizaje que la red ha adquirido a través del entrenamiento al realizar predicciones sobre datos no vistos. Además, estas métricas permiten comparar diferentes modelos e identificar aquel que tenga mejor desempeño en la tarea asignada [31]. En general, la elección de la métrica de evaluación dependerá del tipo de problema que se esté

resolviendo, por ejemplo, clasificación o regresión.

En secciones anteriores se mencionaron algunas funciones costo que también son usadas como métricas de evaluación, por ejemplo, para problemas de regresión las funciones **MAE** y **MSE** son ampliamente utilizadas para evaluar el desempeño del modelo en estas tareas. Por su parte, para tareas de clasificación categórica existen otras métricas de evaluación aparte de las funciones costo mencionadas. Algunas de estas métricas de evaluación se presentan a continuación.

1. **Matrices de confusión:** Las matrices de confusión permiten visualizar el desempeño de los modelos en la tarea de clasificación. Para problemas de clasificación multiclase, la matriz de confusión tendrá un tamaño $k \times k$, donde k es el número de clases. Los renglones de la matriz corresponderán a las etiquetas reales, mientras que las columnas corresponderán a las predicciones realizadas por el modelo [31]. Así pues, el elemento m_{ij} pertenecerá a la clase i y será clasificado en la clase j .

Cuando se tienen 2 clases, es posible definir 4 grupos en los que el modelo podrá clasificar dichas clases. En la figura 3.13 se muestran estos grupos. A continuación se explican cada uno estos.

Verdadero positivo (TP): *True positive* en inglés, son instancias que pertenecen a la clase 0 y que son clasificadas correctamente por el modelo en la clase 0.

Verdadero Negativo (TN): *True negative* en inglés, son instancias que pertenecen a la clase 1 y que son clasificadas correctamente por el modelo en la clase 1.

Falso positivo (FP): *False positive* en inglés, son instancias que pertenecen a la clase 1 y que el modelo ha clasificado erróneamente a la clase 0.

Falso Negativo (FN): *False negative* en inglés, son instancias que pertenecen a la clase 0 y que el modelo ha clasificado erróneamente a la clase 1.

| | | Predicción | |
|------|---------|------------|---------|
| | | Clase 0 | Clase 1 |
| Real | Clase 0 | TP | FN |
| | Clase 1 | FP | TN |

Figura 3.13: Matriz de confusión con dos clases. Imagen propia.

Para matrices de confusión de mayor tamaño, es común representarlas como *mapas de calor* para visualizar de manera rápida el desempeño de clasificación del modelo. En la figura 3.14 se muestra un ejemplo de matriz de confusión con su mapa de calor. En estas matrices de confusión existe un mayor número de agrupaciones, por lo que se redefinen los grupos antes mencionados, reduciéndolos a tres:

Verdadero positivo (TP): Son las instancias clasificadas correctamente por el modelo, las cuales se encuentran en la diagonal de la matriz de confusión. En general, las instancias m_{ij} , que cumplan que $i = j$ se encuentran correctamente clasificadas.

Falso positivo (FP): Son las instancias ubicadas en la columna j , pero fuera de la diagonal. Estas instancias son clasificadas por el modelo en la clase j cuando no pertenecen a dicha clase.

Falso negativo (FN): Son las instancias que realmente pertenecen a la clase i , pero que el modelo clasificó erróneamente en una clase j , tal que $i \neq j$.

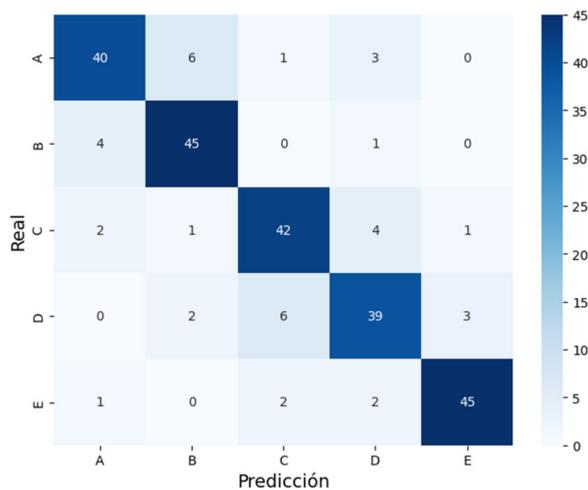


Figura 3.14: Matriz de confusión para problemas multiclase. Imagen propia.

2. **Precisión:** La precisión mide la exactitud de la clasificación a una determinada clase. Este valor establece una proporción de las instancias correctamente clasificadas a una clase con respecto a todas las clasificadas en esa clase. Matemáticamente, se define de la siguiente manera

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.21)$$

Como se puede observar a partir de la expresión, se obtendrá un alto valor de precisión cuando existan valores bajos de falsos positivos FP , es decir, cuando haya cometido pocas predicciones incorrectas a una determinada clase.

Para clasificación multiclase, existen dos maneras de calcular la precisión, cada una con su respectiva interpretación.

3. **Precisión media macro:** *Macro-average Precision* en inglés, es una métrica que mide la precisión obtenida para cada clase para luego obtener el promedio de estas. Matemáticamente se define como

$$\text{Macro-average Precision} = \frac{1}{k} \sum_{i=1}^k \frac{TP_i}{TP_i + FP_i} \quad (3.22)$$

4. **Precisión media micro:** *Micro-average Precision* en inglés, es una métrica que calcula todos los valores de TP obtenidos por el modelo, así como los FP predichos, para luego aplicar la definición de precisión [31]. Matemáticamente esto se expresa de la siguiente forma

$$\text{Micro-average Precision} = \frac{\sum_{i=1}^k TP_i}{\sum_{i=1}^k (TP_i + FP_i)} \quad (3.23)$$

Para problemas de clasificación multiclase, la definición de Micro-average Precision, coincide con el concepto de exactitud (*accuracy*) y sensibilidad (*recall*).

Usualmente, la elección de la métrica depende de la distribución de las clases, siendo la precisión media macro la mejor opción cuando se desea tratar todas las clases con la misma importancia. Sin embargo, es recomendable proporcionar ambas métricas, pues permiten obtener información relevante del modelo [31].

3.3 Redes neuronales convolucionales

Las **redes neuronales convolucionales** (CNN) son un tipo de red neuronal especializadas en el análisis de datos estructurados de manera matricial [12]. En especial, las CNN se utilizan para la extracción de características de imágenes 2D. El nombre de redes neuronales convolucionales se debe a la operación en las que se encuentran basados: la convolución [12].

3.3.1 Convolución

La convolución es una operación matemática que describe cómo una función $g(t)$ afecta o modifica a otra función $f(t)$ al ponderarla. Matemáticamente, se define como

$$s(t) = (f * g)(t) = \int f(\tau)g(t - \tau)d\tau \quad (3.24)$$

donde $*$ se denomina el operador convolución y τ es la variable sobre la que se desplaza g a lo largo de t [12].

Si bien el nombre de este tipo de redes se debe a esta operación, estrictamente hablando, las redes neuronales convolucionales no aplican esta transformación matemática. Realizando la analogía con las redes neuronales multicapa, la función f denotaría la entrada de la red, mientras que la función g sería un filtro, también llamado *kernel*, el cual afecta a f . De manera discreta, la operación de convolución sobre un espacio bidimensional se define de la siguiente forma

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n), \quad (3.25)$$

donde se ha representado a la función f como I , dado que usualmente se convoluciona sobre una imagen, y a g como K , que es el kernel con el que se realiza la convolución. La función de convolución se define de tal forma que sea conmutativa, lo que implica invertir el kernel sobre la imagen I .

Sin embargo, esto no es necesario para el algoritmo utilizado en CNN. En su lugar, se utiliza una implementación parecida a la expresión anterior, conocida como *correlación cruzada* [12], la cual se define de la siguiente forma

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i + m, j + n). \quad (3.26)$$

donde m y n son las dimensiones del kernel.

Fundamentar las redes neuronales en esta operación trae consigo ciertas ventajas con respecto a las redes neuronales densas o MLP. Las CNN son capaces de extraer características locales de un espacio específico, mientras que las MLP lo realizan de manera global. Este hecho permite que la red logre identificar patrones específicos pero invariantes ante traslaciones [5]. Al igual que el ser humano es capaz de reconocer rostros sin importar su posición, ángulo o forma, las CNN son capaces de caracterizar e identificar patrones en las imágenes, tal y como se observa en la figura 3.15. Otra ventaja de las CNN es que siguen el concepto de representación distribuida, al igual que las redes MLP. De igual forma que las redes neuronales densas, las redes neuronales convolucionales son capaces de extraer la mayor cantidad de características generales en sus primeras capas para luego profundizar en los detalles.

La extracción de características con CNN se debe, en su mayoría, a la manera en que estas procesan la información en forma tensorial.

Una imagen puede representarse como un tensor de tres dimensiones: ancho, alto (que determinan sus dimensiones espaciales) y un tercer elemento denominado *canal*. Las imágenes RGB, por ejemplo, poseen tres canales, cada uno correspondiente a un color primario (rojo, verde y azul), lo que permite la representación de imágenes a color al combinar las intensidades captadas por estos filtros. Por su parte, las imágenes en escala de grises o monocromáticas cuentan únicamente con un canal [5].

El resultado de una operación de convolución también es un tensor tridimensional. Sin embargo, en este caso, los canales no están asociados a los colores de la imagen, sino a los filtros aplicados. Estos filtros son los encargados de extraer características

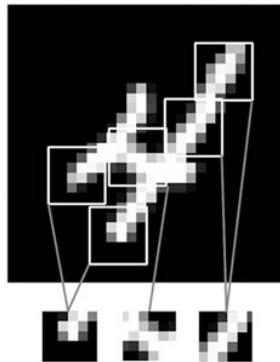


Figura 3.15: Extracción de características mediante CNN. La red logra reconocer patrones, formas y bordes. Imagen recuperada de [5].

específicas o resaltar patrones y detalles relevantes dentro de la imagen. El aprendizaje de la red, análogo a los pesos de las capas ocultas de las MLP, se encontrará en los filtros, los cuales permitirán la optimización de resultados en la red.

Las dimensiones de los kernel de una CNN es menor que el de la matriz de entrada, provocando que la matriz resultante de la operación de convolución sea de menor dimensión, lo que origina una pérdida de información en los datos originales. De manera empírica se ha observado que aumentar el número de filtros reduce la pérdida de información.

En la figura 3.16 se observa un tensor de tamaño $3 \times 4 \times 1$ convolucionando con un kernel de tamaño $2 \times 2 \times 1$. El mapa de características resultante ve su tamaño reducido a un tensor de tamaño $2 \times 3 \times 1$.

Después de obtener el tensor resultante de la operación de convolución, se aplica una función de activación, al igual que en las redes neuronales multicapa. Esta función introduce no linealidad en los datos, permitiendo realizar transformaciones más complejas y enriqueciendo el tensor obtenido, llamado *mapa de características*, con información relevante.

3.3.2 Padding

Si bien la matriz resultante se va reduciendo con cada operación de convolución, existen técnicas para evitar su reducción de tamaño. Una técnica sencilla para evitar la reducción es el *padding*. La técnica de padding consiste en agregar a los extremos de las matrices de I , columnas y filas adicionales, para que así, al realizar la operación de convolución, el tensor resultante sea más grande que el original [5]. Las columnas y filas agregadas suelen contener ceros para así no introducir información adicional. Usualmente suelen usarse dos técnicas de padding, conocidas como *valid* y *same*. El valid padding no agrega nada alrededor de la entrada, o agrega un padding de cero, mientras que el same padding agrega tantas columnas y filas necesarias tal que la matriz después de la convolución conserve sus dimensiones [5]. Pueden agregarse

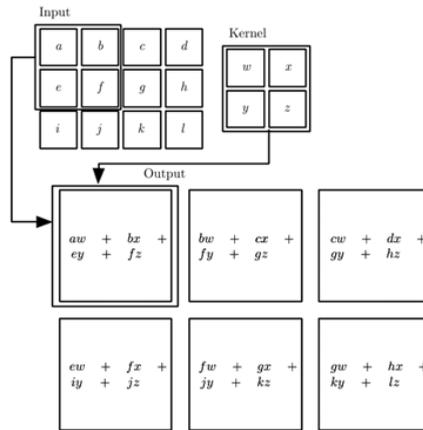


Figura 3.16: Proceso de convolución. Imagen recuperada de [12].

tantas columnas y filas como el usuario desee.

3.3.3 Stride

Como se ha mencionado, el tamaño de los tensores, y también del mapa de características, dependerán tanto del tamaño del kernel como del padding. Sin embargo, se ha considerado que el kernel recorre continuamente el tensor de entrada. El *stride* define el número de posiciones que recorre el filtro antes de realizar la operación de convolución. Hasta el momento, se ha definido un stride de 1, el decir, el kernel recorre una posición cada vez que realiza la convolución, sin embargo, este número puede variar como el usuario desee [5]. En la figura 3.17 se puede apreciar el proceso de convolución con padding y strides definidos.

Es posible determinar las dimensiones H_{out} del mapa de característica resultante de la convolución de una matriz de lado H_{in} con un kernel de lado K al utilizar un padding de P y un stride de tamaño S siguiendo la siguiente relación

$$H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1 \quad (3.27)$$

donde $\lfloor \cdot \rfloor$ es la función piso.

3.3.4 Pooling

Después de realizar la operación de convolución, el mapa de características resultante se introduce en una capa que realiza una operación distinta al de la convolución, conocida como *pooling*. La capa de pooling, al igual que la capa de convolución, cuenta con kernels y strides definidos por el usuario, la cual realiza una operación específica sobre el mapa de características, generando uno nuevo.

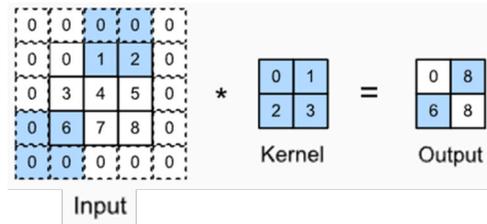


Figura 3.17: Proceso de convolución con padding de 1 y stride horizontal de 2 y vertical de 3. Imagen recuperada y modificada de [1].

Las operaciones más conocidas en las capas de pooling son el *pooling promedio* (*average pooling*) y el *pooling máximo* (*max pooling*). El average pooling promedia los valores del mapa de características sobre el área en el que actúa el kernel, mientras que el max pooling extrae el valor máximo dentro de la ventana sobre la que el kernel actúa.

La operación de pooling suele reducir drásticamente las dimensiones del mapa de características, sin embargo, la reducción en esta etapa trae consigo ciertas ventajas, como aumentar la velocidad de procesamiento (al contar con un número menor de datos) pero sobre todo la generación de un nuevo mapa de características con información más relevantes.

Empíricamente se ha mostrado un mayor desempeño al utilizar max pooling que average pooling o al no usar capas de pooling. La razón podría deberse a que al extraer el valor máximo dentro de un área se captura mayor información significativa en esta, encontrando patrones que caractericen mejor regiones del mapa que al extraer el promedio, el cual podría encontrarse difuso por características no relevantes, generando ruido en la identificación de patrones. Usualmente, las capas de pooling suelen contener kernels de tamaño 2×2 con un stride de 2, lo que reduce las dimensiones del mapa de características resultante a la mitad [5].

3.3.5 Capas

Las redes neuronales convolucionales, también llamadas *ConvNets*, se construyen utilizando diferentes tipos de capas, las cuales realizan distintas operaciones sobre los valores de entrada en cada una de estas. A continuación, se describen las principales capas que conforman las arquitecturas clásicas de este tipo de modelo.

1. **Capa de convolución:** Esta capa aprende mediante la operación de convolución características relevantes sobre los tensores de entrada, con dimensiones $H \times W \times C$. Para ello, en cada capa de convolución se asigna un número de n kernels, normalmente cuadrados, de tamaño $k \times k$ y con el mismo número de canales C que el tensor de entrada. Una convención para denotar las dimensiones y número de filtros en una capa de convolución es utilizar la notación $k \times k @ n$. Además, es necesario establecer el valor de padding P y stride S , así como su función de activación. Cada canal del tensor de entrada se convolucio-

na con el respectivo canal del filtro, obteniendo C matrices las cuales se suman formando una sola matriz, la cual se evaluará elemento a elemento en la función de activación. Este proceso se realiza para los n filtros obteniendo el mapa de característica resultante, el cual tendrá dimensiones $H_{out} \times W_{out} \times n$, donde H_{out} y W_{out} están dados por la ecuación (3.27).

2. **Capa de pooling:** Esta capa aplica alguna de las operaciones de pooling, siendo las más comunes average pooling y max pooling. Para ello, utiliza ventanas de tamaño $k \times k$ las cuales usan un stride de tamaño S definido por el usuario. En estas capas es común utilizar strides S con la misma longitud de k .
3. **Capa de aplanamiento:** En las ConvNet se suelen anidar en las capas finales capas MLP, también llamadas capas densas. Dado que las capas densas reciben únicamente vectores como valores de entrada es necesario transformar los tensores de salida de las capas de convolución en un vector. Una de las formas para lograr esto, es utilizar la capa de aplanamiento o *flatten layer*, la cual transforma el tensor de la última capa de convolución de tamaño $H \times W \times n$ en un vector de tamaño $H \cdot W \cdot n \times 1$.
4. **Capa densa:** Las capas densas o MLP se encargan de aprender relaciones o patrones de un vector de características. Cada capa densa tiene un número m de neuronas y una función de activación asignada.
5. **Capa de salida:** Si bien la capa de salida sigue siendo una capa densa, la elección de la función de activación y del número de neuronas en esta capa depende del problema a resolver (revisar sección 3.2.4). En esta capa se obtienen los resultados obtenidos por la CNN.

La elección de distintas capas permiten la creación de diversas arquitecturas de redes neuronales convolucionales, como si de piezas de Lego se tratasen [5]. En la figura 3.18 se puede apreciar las capas que constituyen a una ConvNet clásica.

A lo largo del tiempo, las ConvNets han sido ampliamente utilizadas para el reconocimiento de patrones en imágenes. El modelo *LeNet-5* marcó un hito en la implementación de estos modelos, consolidando su aplicación en tareas de visión por computadora [16]. Este modelo, basado en el algoritmo de gradiente descendiente, se entrenó utilizando un dataset de imágenes de dígitos escritos a mano, llamado *MNIST*, un conocido *benchmark* constituido de 70,000 imágenes binarias (monocromáticas) de tamaño 28×28 , de las cuales 60,000 imágenes fueron utilizadas para entrenamiento y 10,000 para testing. La precisión alcanzada en el conjunto de testing por el modelo de LeNet-5 fue de 99.05%. Si bien, este modelo estaba construido con las capas descritas anteriormente, presentaba distinciones, como conexiones específicas entre algunas capas o funciones de activación distintas, como en su capa de salida, en donde utilizaba la *función de base radial* (RBF), la cual media la distancia euclidiana entre el vector de salida esperado y obtenido [16].

En 2012, Krizhevsky, Sutskever y Hinton publicarían detalles acerca de una arquitectura de red neuronal convolucional que entrenaron utilizando parte de la base de datos utilizada en la competición anual *ImageNet Large-Scale Visual Recognition*

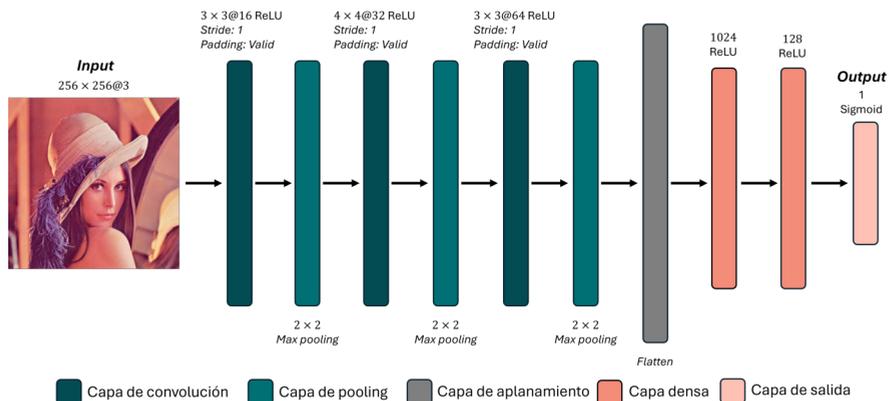


Figura 3.18: Arquitectura clásica de una red neuronal convolucional. En cada capa se aprecia las características de cada capa. Imagen propia.

Challenge (ILSVRC) del año 2010 [15]. En el año de la publicación de este artículo, ganarían la competición ILSVRC-2012 utilizando una variante de la red entrenada con los datos del 2010. Esta arquitectura, conocida como *AlexNet*, establecería precedentes para la construcción de redes neuronales convolucionales, destacando algunos aspectos como el uso de funciones de activación ReLU en sus capas ocultas. En la tabla 3.1 se detalla la estructura implementada en la arquitectura de AlexNet.

| # | Capa | Dimensiones | Stride | Padding | Función |
|----|-------------|-------------|--------|---------|---------|
| 0 | Entrada | 227 × 227 | NA | NA | NA |
| 1 | Convolución | 11 × 11@96 | 4 | 0 | ReLU |
| 2 | Pooling | 3 × 3 | 2 | 0 | Max |
| 3 | Convolución | 5 × 5@256 | 1 | 2 | ReLU |
| 4 | Pooling | 3 × 3 | 2 | 0 | Max |
| 5 | Convolución | 3 × 3@384 | 1 | 1 | ReLU |
| 6 | Convolución | 3 × 3@384 | 1 | 1 | ReLU |
| 7 | Convolución | 3 × 3@256 | 1 | 1 | ReLU |
| 8 | Pooling | 3 × 3 | 2 | 0 | Max |
| 9 | Densa | 4096 | NA | NA | ReLU |
| 10 | Densa | 4096 | NA | NA | ReLU |
| 11 | Densa | 1000 | NA | NA | Softmax |

Tabla 3.1: Arquitectura de AlexNet [15]. En la columna de dimensiones se encuentra el número de kernels o nodos, según el tipo de capa utilizada, mientras que en la columna de función se colocan las funciones de activación en capas que aprenden o el tipo de operación realizada por la capa de pooling.

Para cumplir el objetivo de desarrollar una red neuronal convolucional capaz de determinar las cargas topológicas de haces con vórtices ópticos, se generó un conjunto de datos el cual se utilizó para el entrenamiento, validación y prueba del modelo. Estos datos consistieron en un conjunto de imágenes de haces cuya fase presentaba un término azimutal con la forma $\exp(il\varphi)$, característico de los vórtices ópticos con carga topológica l . Además, el conjunto de datos utilizado se dividió en imágenes de haces experimentales y haces simulados. A continuación, se explica la metodología utilizada para la generación de cada conjunto de datos.

4.1 Generación del conjunto de datos experimental

Para generar el conjunto de imágenes experimentales, se utilizó un montaje $4f$, tal y como se muestra en la figura 4.1.

En el montaje se empleó un haz láser altamente monocromático con longitud

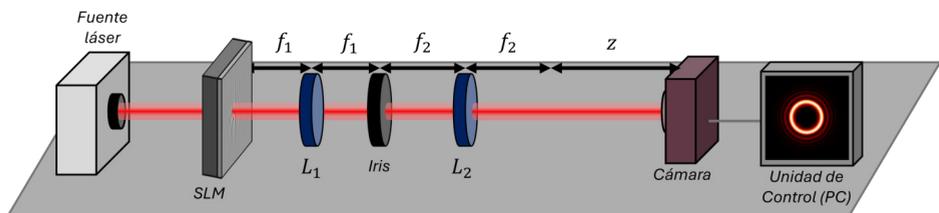


Figura 4.1: Montaje experimental utilizado para la generación de imágenes experimentales de vórtices ópticos.

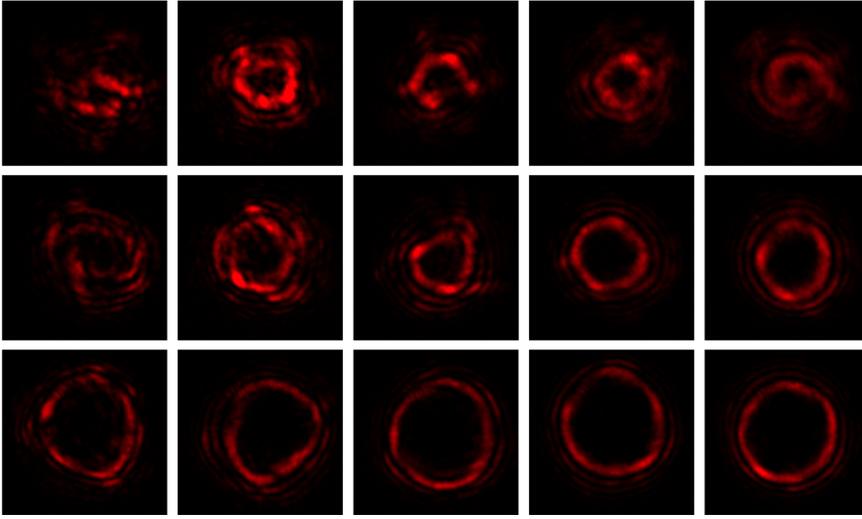


Figura 4.2: Vórtices ópticos experimentales. De izquierda a derecha el nivel Strehl aumenta de 0.3 hasta 0.7 y de arriba a abajo las cargas topológicas de los haces son 2, 10 y 18. Nótese como los haces con menor nivel de Strehl presentan una mayor turbulencia.

de onda de 633 nm , previamente colimado. Este haz fue dirigido hacia la pantalla del modulador espacial de luz por fase, específicamente un dispositivo HOLOEYE PLUTO-NIR-011 con resolución de 1920×1080 píxeles. El SLM, controlado por un computador, proyectaba hologramas digitales que modulaban la fase del haz incidente, generando la estructura deseada. El haz modulado se reflejaba hacia un sistema óptico $4f$, compuesto por dos lentes positivas L_1 y L_2 , con distancias focales de 250 mm y 125 mm , respectivamente. En el plano de Fourier entre ambas lentes, se colocó una apertura o iris que permitió filtrar los órdenes de difracción no deseados, dejando pasar únicamente el orden cero. A una distancia 30 cm del plano de Fourier generado por la segunda lente, se ubicó una cámara Thorlabs con resolución de 1280×1024 píxeles, la cual se encargó de capturar la imagen de la intensidad del haz. Las imágenes obtenidas fueron almacenadas automáticamente en el computador.

Con el montaje experimental realizado, se hicieron mediciones que permitieron identificar el rango de haces que podían generarse y registrarse adecuadamente en la cámara, delimitando así el espacio de búsqueda en el conjunto de datos. Se concluyó que los haces con cargas topológicas mayores a $|l| = 18$ excedían el tamaño del campo de visión de la cámara, lo que impedía capturar la estructura completa del haz. Por lo tanto, el rango de cargas topológicas generadas abarcaría desde -18 hasta 18 , con un incremento de 4 , generando así un total de 10 clases distintas de haces. Asimismo, se simuló diferentes niveles de turbulencia atmosférica en los haces al variar el parámetro de Strehl, parámetro que mide la turbulencia del haz, modificando así el holograma digital. La turbulencia generada tomó valores entre 0.3 y 0.7

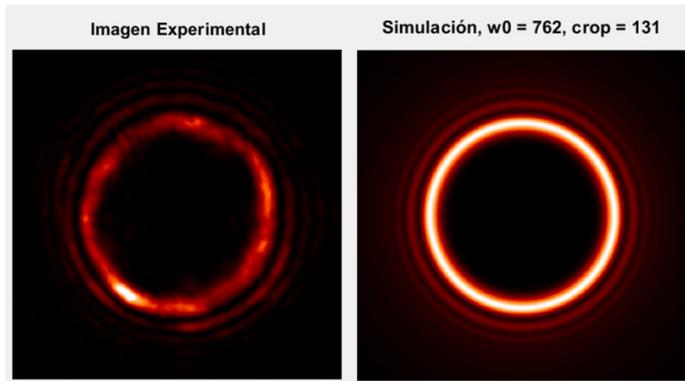


Figura 4.3: Proceso de comparación entre imagen experimental e imagen simulada.

con incrementos de 0.1, dando lugar a cinco niveles distintos de turbulencia. Combinando estas variaciones con las distintas cargas topológicas, se generaron imágenes correspondientes a un total de 50 clases distintas de haces.

Mediante un código elaborado en MATLAB se controló tanto el SLM, modificando el holograma que se proyectaba en este, como la cámara, capturando la imagen de la intensidad del haz modulado y almacenándola en el computador. Este proceso se realizó de manera automática generando un conjunto de 1,250 imágenes experimentales, 25 imágenes para cada una de las clases.

En la figura 4.2 se presentan imágenes tomadas experimentalmente de haces con vórtices ópticos con distintas cargas topológicas y diferentes niveles de turbulencia.

4.2 Generación del conjunto de datos simulado

Se utilizó el software MATLAB para la generación del conjunto de datos simulados. Para ello, se desarrolló un código que permitiera simular haces con fase azimutal de la forma $\exp(il\varphi)$. Además, se agregaron perturbaciones a sus frentes de onda mediante la simulación de turbulencia atmosférica, añadiendo una máscara de fase aleatoria a esta.

Para mejorar la similitud entre las imágenes simuladas con las experimentales, se modificaron dos parámetros clave: el radio de cintura del haz w_0 y un parámetro denominado *crop size*, el cual ajustaba el tamaño del recorte aplicado a la imagen. Para determinar la combinación óptima de estos parámetros, se utilizó una imagen experimental de referencia correspondiente a un haz con carga topológica $l = 18$. Con esto fue posible realizar una comparación visual entre la imagen experimental y las imágenes simuladas, tal y como se muestra en la figura 4.3. Sin embargo, para cuantificar la similitud entre ambas, se redimensionó la imagen simulada al tamaño de la imagen experimental utilizando la función `imresize` con el método de interpolación *bilinear*. Posteriormente se empleó la función `corr2` para calcular el coeficiente de

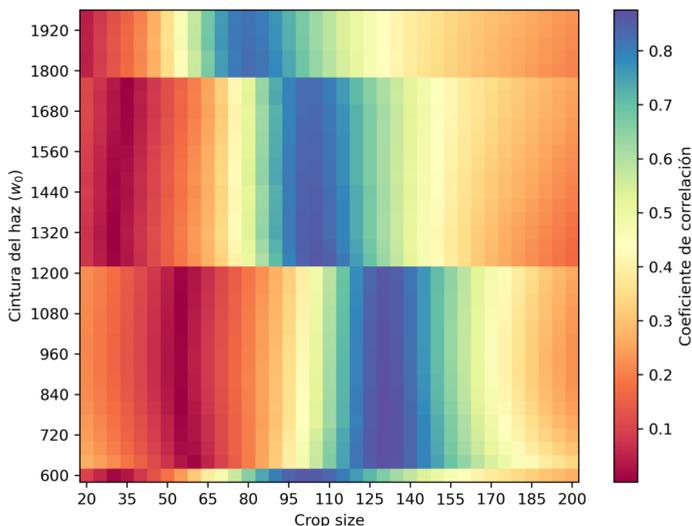


Figura 4.4: Mapa de calor de coeficientes de correlación obtenidos para distintos valores de w_0 y $crop\ size$.

correlación entre las dos imágenes. Se exploró un amplio rango de valores para w_0 y $crop\ size$ y se seleccionó la combinación que maximizara la correlación con la imagen experimental. En la figura 4.4 se muestran los resultados obtenidos para los distintos valores utilizados.

Una vez definidos los parámetros óptimos, se procedió a la generación del conjunto de datos simulados. Se consideraron las mismas clases que las que se utilizaron para el conjunto experimental: haces con cargas topológicas desde $l = -18$ hasta $l = 18$, con incrementos de 4, y niveles de turbulencia caracterizados por el índice de Strehl, variando entre 0.3 y 0.7 en incrementos de 0.1. En total, se generaron 50,000 imágenes, 1,000 por cada clase.

En la Figura 4.5 se ilustran algunos ejemplos de haces simulados con distintas cargas topológicas y niveles de turbulencia.

4.3 Preprocesamiento de las imágenes

Una vez generados ambos conjuntos de imágenes, fue necesario unificarlos para que fueran lo más homogéneos posible entre sí. Para ello, se utilizaron como referencia imágenes de haces con carga topológica $l = 18$, ya que presentan el mayor diámetro.

El primer paso consistió en binarizar las imágenes para eliminar el ruido y la información irrelevante. En el caso del conjunto experimental, se analizó el histograma de intensidades, observándose una alta concentración de píxeles con valores inferiores a 6 (en una escala de 0 a 255), lo cual se interpretó como ruido o zonas sin intensidad. Por esta razón, se estableció dicho valor como umbral. En cambio, para las imágenes

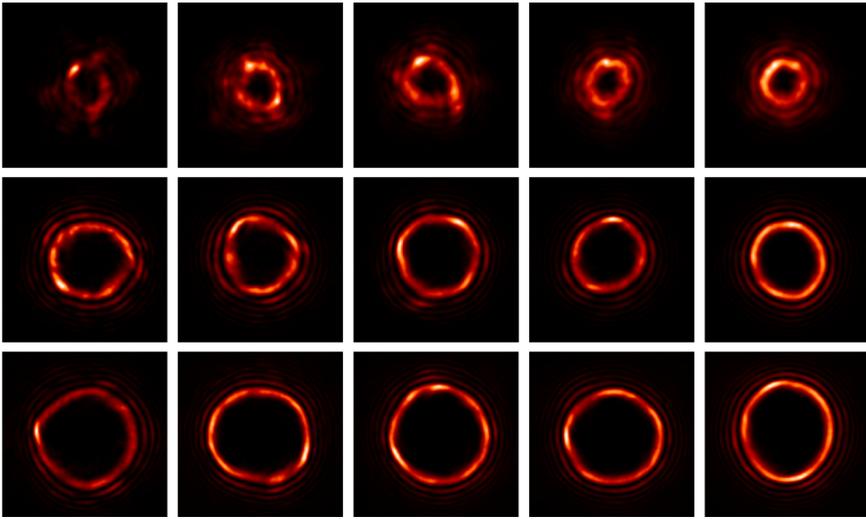


Figura 4.5: Vórtices ópticos simulados. De izquierda a derecha el nivel de turbulencia aumenta de 0.3 hasta 0.7 y de arriba a abajo las cargas topológicas de los haces son 2, 10 y 18.

simuladas se empleó la función `imbinarize` con el método *global*, el cual proporcionó buenos resultados.

Posteriormente, se aplicó la función `imfill` con la opción *holes*, lo que permitió rellenar huecos internos y obtener contornos más completos. A continuación, se utilizó la función `bwareaopen` para eliminar regiones de ruido residual, considerando una conectividad de 8 píxeles y un umbral de área de 400 píxeles conectados.

Este procesamiento permitió determinar las dimensiones en píxeles de los vórtices ópticos, a partir de las cuales se calcularon su centro y diámetro mediante la función `regionprops`.

Con el centro y el diámetro definidos, se recortaron las imágenes y se normalizaron, escalando los valores de intensidad al rango $[0, 1]$. Finalmente, las imágenes se redimensionaron a un tamaño fijo de 256×256 píxeles utilizando la función `imresize` con el método *nearest*. Como resultado, se obtuvieron imágenes monocromáticas y normalizadas de vórtices ópticos, tanto experimentales como simuladas, todas con el mismo tamaño y formato.

4.4 Entrenamiento de la red neuronal convolucional

Una vez conformado el conjunto de datos, se procedió con el entrenamiento de las redes neuronales convolucionales. Para ello, se empleó la biblioteca de TensorFlow (versión 2.17) de Python, ampliamente utilizada en tareas de aprendizaje profundo por su flexibilidad y eficiencia computacional. El entrenamiento se realizó en un

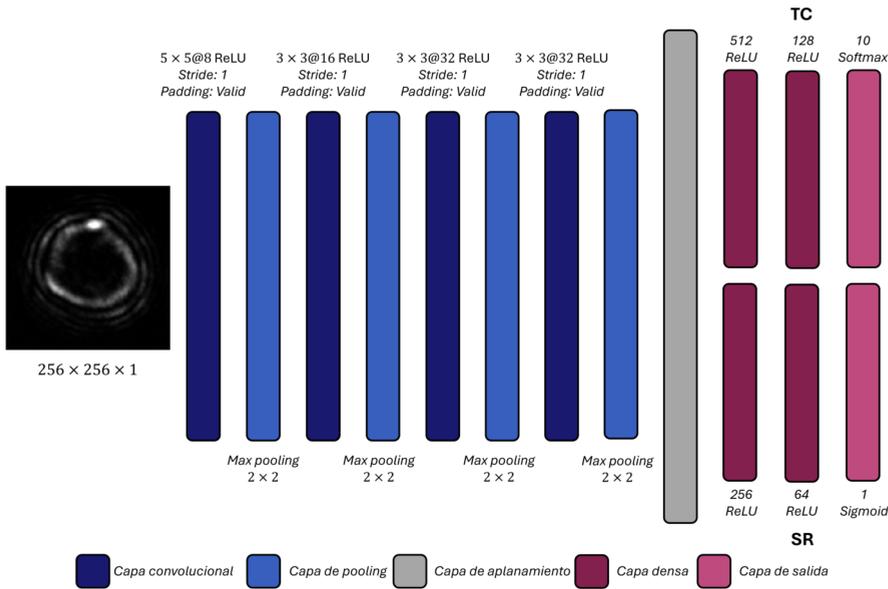


Figura 4.6: Arquitectura de las CNN implementadas. En la figura se muestran los tamaños de los kernels, el número de filtros y el número de nodos en cada capa. TC hace referencia a las capas utilizadas para la predicción de las cargas topológicas, mientras que SR corresponde a las capas destinadas al problema de regresión del nivel de turbulencia.

computador con GPU, específicamente la tarjeta gráfica NVIDIA GeForce RTX 3090.

Se diseñaron dos CNN con múltiples salidas (CNN multi-output), con el objetivo de predecir simultáneamente la carga topológica y el nivel de turbulencia de los vórtices ópticos. Cada modelo fue entrenado utilizando un conjunto de datos específico: uno experimental y otro simulado.

Cada conjunto de datos, tanto el experimental como el simulado, se dividió en tres subconjuntos: entrenamiento, validación y prueba. Para el conjunto experimental se utilizó una proporción de 76-12-12 con la cual se obtuvieron 950, 150 y 150 imágenes para el entrenamiento, validación y prueba. Dado que se tenía una mayor cantidad de imágenes en el conjunto simulado se utilizó la clásica partición de 70-15-15. El conjunto experimental estuvo compuesto por 1,250 imágenes y el conjunto simulado por 50,000 imágenes.

Se optó por entrenar dos CNN con diferentes conjuntos de datos, ya que se deseaba realizar una validación cruzada entre los modelos. En particular, se buscaba evaluar la capacidad de generalización del modelo entrenado con los datos simulados, verificando su desempeño al realizar predicciones sobre el conjunto de datos experimentales.

Aunque los modelos fueron entrenados con diferentes conjuntos de datos, ambos compartieron la misma arquitectura, la cual se muestra en la figura 4.6. Esta fue diseñada tomando como referencia la reconocida red AlexNet. Se evitó utilizar una

arquitectura más profunda para minimizar el riesgo de sobreajuste debido a la estructura del conjunto de datos. A continuación, se describe en detalle la arquitectura utilizada.

La red constó de cuatro bloques compuestos por capas de convolución y max pooling. Cada capa convolucional utilizó la función de activación ReLU y kernels cuyo tamaño se indica en la figura 4.6. El stride y el padding fueron fijados en 1 y 0 (*valid*), en todas las capas convolucionales. Por su parte, las capas de pooling emplearon kernels y stride de 2, lo cual reducía progresivamente las dimensiones de los mapas de características resultantes.

Tras los bloques de convolución y pooling, se añadió una capa de aplanamiento para preparar las características extraídas de las imágenes para las capas densas. A partir de ahí, la red se dividió en dos ramas, con el objetivo de abordar los dos problemas a resolver: la clasificación de la carga topológica y la predicción del nivel de turbulencia. El primero fue tratado como un problema de clasificación multiclase, mientras que el segundo se planteó como un problema de regresión. Cada rama incluyó dos capas densas con función de activación ReLU, utilizando un dropout con una tasa del 50 %, valor comúnmente utilizado para prevenir el sobreajuste en redes profundas. El número de nodos de cada capa densa se muestra en la figura 4.6.

Posteriormente, las capas de salida utilizaron funciones de activación asociadas al problema que resolverían, siendo *softmax* para la clasificación multiclase y *sigmoid* para la regresión. Esta última fue elegida debido a que los valores de turbulencia se encontraban acotados en el intervalo $[0, 1]$, coincidiendo con el codominio de la función sigmoide. En cuanto a las funciones de error utilizadas durante el entrenamiento, se empleó la entropía cruzada categórica (CCE) para la clasificación multiclase y el error cuadrático medio (MSE) para la regresión. A estos valores se les denominó *TC loss* y *SR loss*.

El entrenamiento de la red se llevó a cabo mediante el método de mini-batch gradient descent, utilizando un tamaño de lote de 32 imágenes, pues se observó un mejor desempeño con este tamaño de mini-batch que con respecto a otros. Se empleó el optimizador *Adam*, ampliamente utilizado en el estado del arte para redes neuronales convolucionales, con una tasa de aprendizaje de 0.001, valor por defecto en dicho optimizador. Durante el entrenamiento se monitorearon las métricas de *TC Loss*, *SR Loss*, así como el accuracy en la clasificación y MAE en la regresión, tanto en los conjuntos de entrenamiento como en los de validación. El proceso de entrenamiento fue configurado durante un tiempo inicial de 100 épocas.

Para prevenir el sobreajuste, se aplicó la técnica de early stopping, que monitoreó específicamente la métrica de *TC Loss* obtenida en el conjunto de validación, favoreciendo así la generalización del modelo a nuevos datos. Esta decisión se fundamentó en que la tarea principal de la red era la clasificación de los haces ópticos, siendo la predicción de la turbulencia un objetivo secundario. Se configuró con un umbral mínimo de mejora de 10^{-4} en el error y un tiempo de espera de 15 épocas para el modelo entrenado con imágenes simuladas y 25 épocas para el modelo entrenado con imágenes experimentales. Esto permitió detener el entrenamiento antes de completar las 100 épocas inicialmente establecidas, conservando el modelo con mejor desempeño sobre el conjunto de validación.

5.1 Desempeño de la CNN entrenada con datos experimentales

En esta sección se presentan los resultados obtenidos tras el entrenamiento, evaluación y prueba de la red neuronal convolucional utilizando el conjunto de datos experimental. A lo largo del capítulo a este modelo se le llamará *modelo experimental*. Se incluyen las gráficas del entrenamiento del modelo, así como las métricas de evaluación correspondientes a los dos problemas abordados.

La figura 5.1 muestra la evolución de las funciones de pérdida durante el entrenamiento para los conjuntos de entrenamiento y validación. En estas gráficas se puede observar como descienden ambas métricas para el conjunto de entrenamiento, lo cual es usual en este tipo de modelos pues la red empieza a aprender patrones específicos de los datos de entrenamiento, perdiendo generalidad. Sin embargo, a partir de la época 9 en el TC loss empieza a notarse el sobreajuste, provocando que la curva de error en el conjunto de validación aumente a lo largo de las épocas. Por otro lado, para el error de la predicción, SR loss, se observa que disminuye a lo largo de cada época en ambos conjuntos, teniendo una menor diferencia en el sobreajuste de los datos.

Por otra parte, en las figuras 5.2 y 5.3 se presentan las curvas de desempeño correspondientes al accuracy y al MAE, respectivamente. En la gráfica del accuracy se observa como la predicción de los datos de entrenamiento mejora a lo largo de las épocas, estancándose para el conjunto de validación. Por su parte, el MAE es equivalente a lo observado en el SR loss (MSE), disminuyendo a lo largo del entrenamiento para ambos conjuntos, teniendo un pequeño sobreajuste sobre los datos de entrenamiento.

El mejor desempeño del modelo experimental, bajo la monitorización del TC loss,

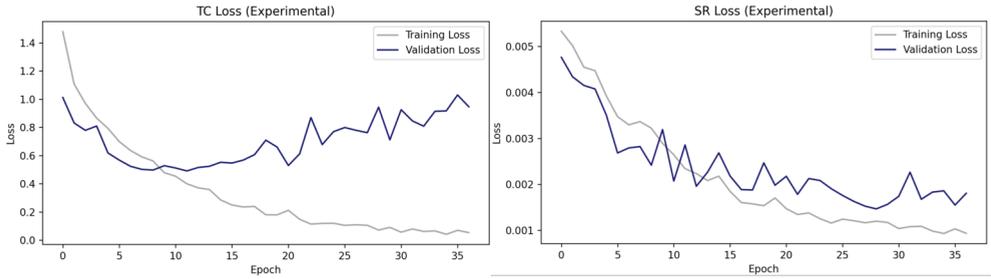


Figura 5.1: Evolución de las funciones de error durante el entrenamiento en el modelo experimental.

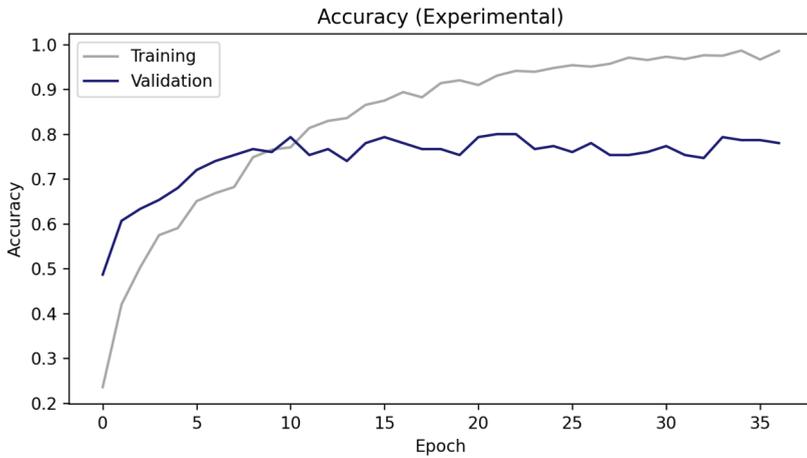


Figura 5.2: Gráfica de accuracy durante el entrenamiento en el modelo experimental.

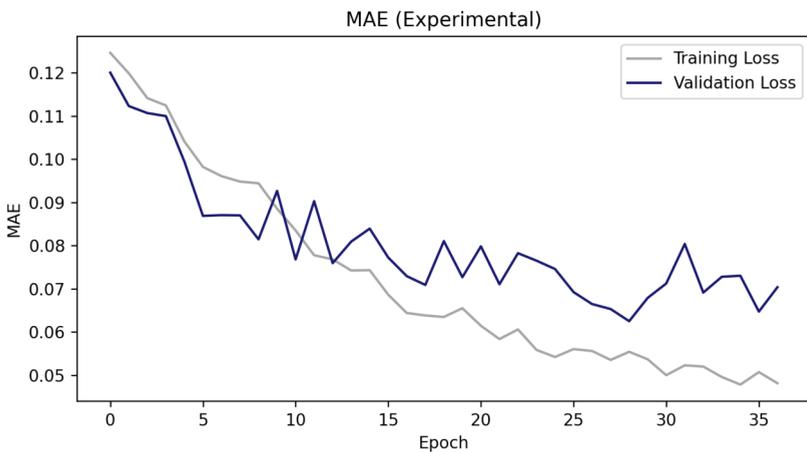


Figura 5.3: Gráfica de MAE durante el entrenamiento en el modelo experimental.

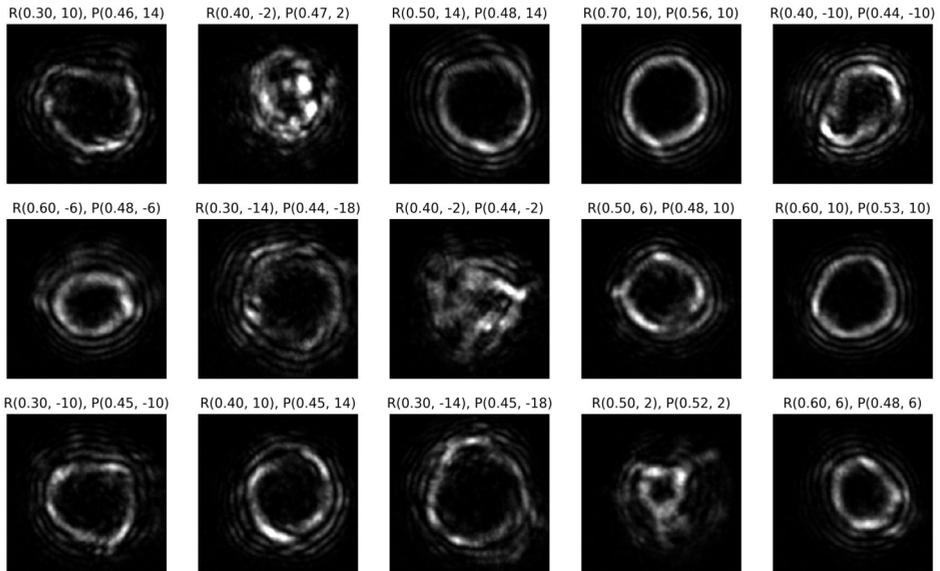


Figura 5.5: Algunas predicciones del modelo experimental sobre el conjunto experimental de prueba. Se indican los valores reales R y los valores predichos P con la notación (SR, TC) .

prueba. En cada imagen, se muestran los valores reales, indicados con la etiqueta R , y los valores predichos, indicados con la etiqueta P , utilizando la notación (SR, TC) , donde SR representa el valor de turbulencia y TC la carga topológica.

5.2 Desempeño de la CNN entrenada con datos simulados

En esta sección se presentan los resultados obtenidos tras el entrenamiento, evaluación y prueba de la red neuronal convolucional utilizando el conjunto de datos simulados. A lo largo del capítulo a este modelo se le llamará *modelo simulado*. De igual forma que en la sección anterior se incluyen las gráficas de error durante el entrenamiento, así como las métricas de evaluación correspondientes a los dos problemas abordados.

La figura 5.6 muestra la evolución de las funciones de pérdida durante el entrenamiento para los conjuntos de entrenamiento y validación en el modelo simulado. La función de error en clasificación, dada por el TC loss, disminuye drásticamente en las primeras épocas, lo que sugiere que el modelo ha logrado aprender características específicas para discernir entre las distintas clases de vórtices ópticos de manera precisa. De hecho, existe muy poco sobreajuste sobre los datos, lo que habla de la homogeneidad del conjunto de datos simulados. Por su parte, el error en el problema

5.2 Desempeño de la CNN entrenada con datos simulados

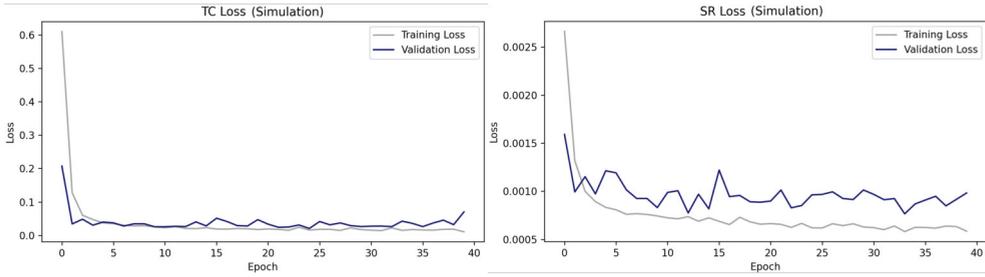


Figura 5.6: Evolución de las funciones de error durante el entrenamiento en el modelo simulado.

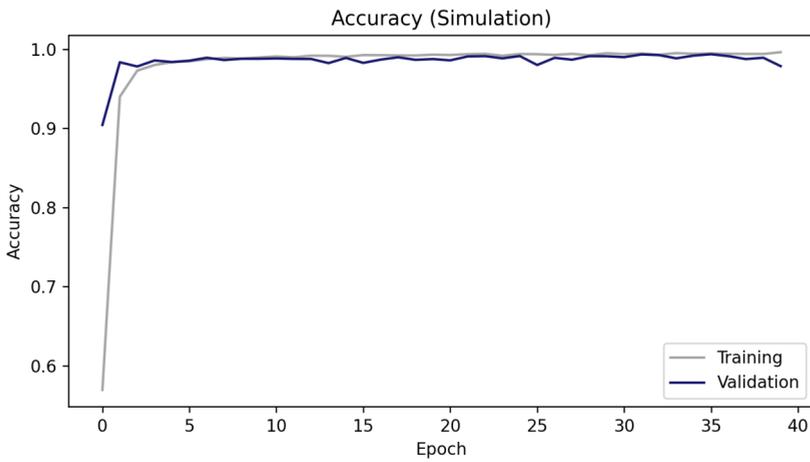


Figura 5.7: Gráfica de accuracy durante el entrenamiento en el modelo simulado.

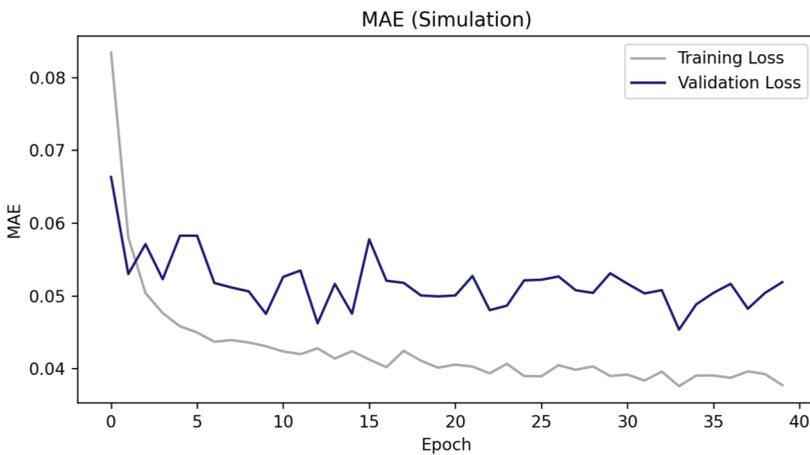


Figura 5.8: Gráfica de MAE durante el entrenamiento en el modelo simulado.

de predicción de la turbulencia, dada por SR loss, disminuye en ambos conjuntos, pero presenta un mayor sobreajuste.

Por otro lado, en las figuras 5.7 y 5.8 se presentan las curvas de desempeño correspondientes al accuracy y al MAE del modelo, respectivamente. Se observa un excelente valor de accuracy llegando aproximadamente al 100 % en los conjuntos de entrenamiento y validación. De igual forma que con el SR loss, el MAE del modelo desciende con las épocas pero se estanca para el conjunto de validación, lo que indica pérdida de generalidad.

El mejor desempeño del modelo simulado se alcanzó en la época 24, con los siguientes resultados en el conjunto de validación: un TC loss de 0.0207, un SR loss de 9.0627×10^{-4} , un accuracy del 99.13 % y un MAE de 0.0521.

Una vez finalizado el entrenamiento, se evaluó el modelo simulado utilizando el conjunto de prueba de los datos simulados. De igual forma que para el modelo experimental, para la tarea de clasificación se construyó una matriz de confusión con el fin de visualizar las clasificaciones realizadas por el modelo. En la figura 5.9 se muestra esta matriz. El modelo obtuvo una precisión media macro del 98.54 % y una precisión media micro del 98.49 %. Las predicciones realizadas por el modelo se encuentran sobre la diagonal principal, lo que habla de una excelente predicción sobre los datos, existiendo errores principalmente sobre los vórtices ópticos con menor carga topológica.

Finalmente, en la figura 5.10 se presentan ejemplos de las predicciones realizadas por el modelo simulado sobre distintas imágenes del conjunto simulado de prueba, utilizando la misma representación de las etiquetas R y P que con el modelo experimental.

5.3 Evaluación cruzada simulado-experimental

En esta sección se evalúa el desempeño del modelo simulado en el conjunto de imágenes experimentales. El objetivo es determinar la capacidad de generalización del modelo simulado cuando se enfrenta a datos reales. Para ello, se presentan los valores de pérdida correspondientes a cada una de las tareas (TC loss y SR loss), así como las métricas previamente utilizadas para evaluar los modelos.

Al evaluar el modelo simulado sobre el conjunto experimental de prueba, se obtuvieron los siguientes resultados: un TC loss de 2.8265, un SR loss de 0.0350, un MAE de 0.1633, una precisión media macro del 65.38 % y una precisión media micro del 63.33 %. Estos resultados muestran que, aunque el modelo presenta un excelente desempeño con imágenes simuladas, su capacidad de clasificación disminuye significativamente al aplicarse a datos reales. El elevado valor del TC loss muestra que el modelo presenta dificultades para discernir entre las distintas clases de vórtices ópticos. Esto sugiere que existen estructuras y patrones en el conjunto de datos simulados que no caracterizan correctamente los datos experimentales o que estos últimos resultan ser más complejos.

La Figura 5.11 muestra la matriz de confusión correspondiente a la clasificación de la carga topológica en las imágenes experimentales. Aunque se observa la presencia de

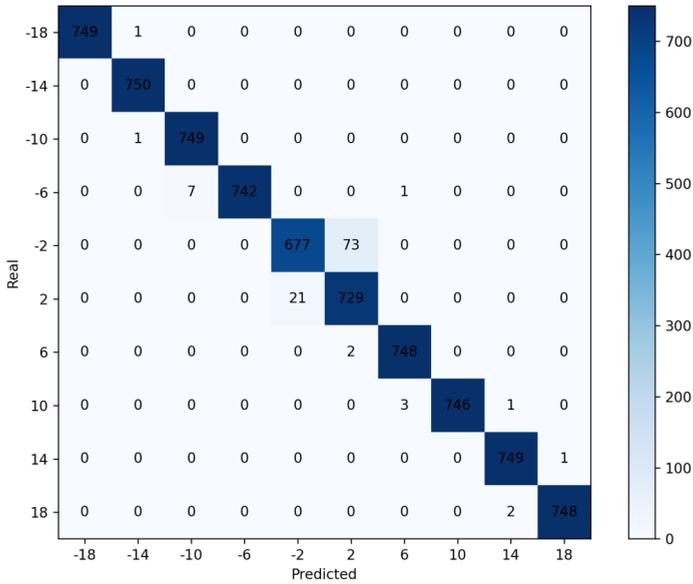


Figura 5.9: Matriz de confusión para la clasificación de la carga topológica sobre el conjunto simulado de prueba.

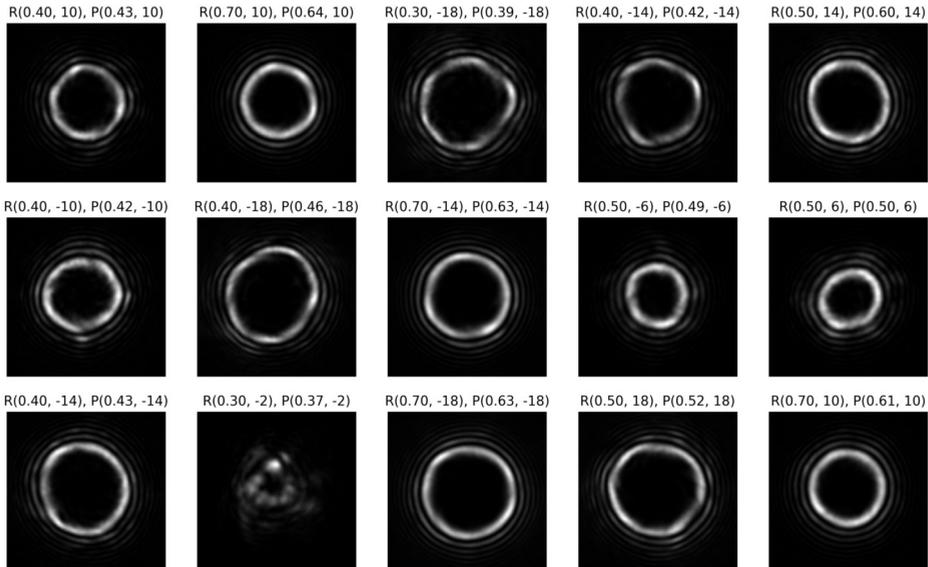


Figura 5.10: Algunas predicciones del modelo simulado sobre el conjunto simulado de prueba. Se indican los valores reales R y los valores predichos P con la notación (SR, TC) .

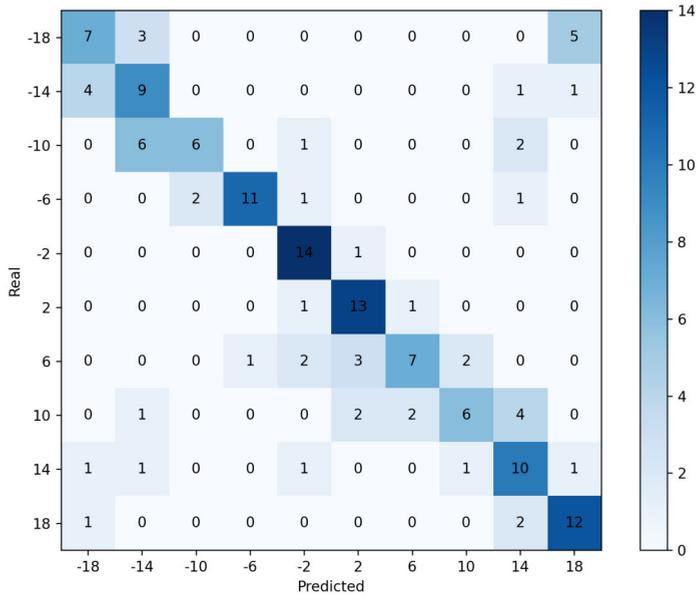


Figura 5.11: Matriz de confusión para la clasificación de la carga topológica sobre el conjunto experimental de prueba utilizando el modelo simulado.

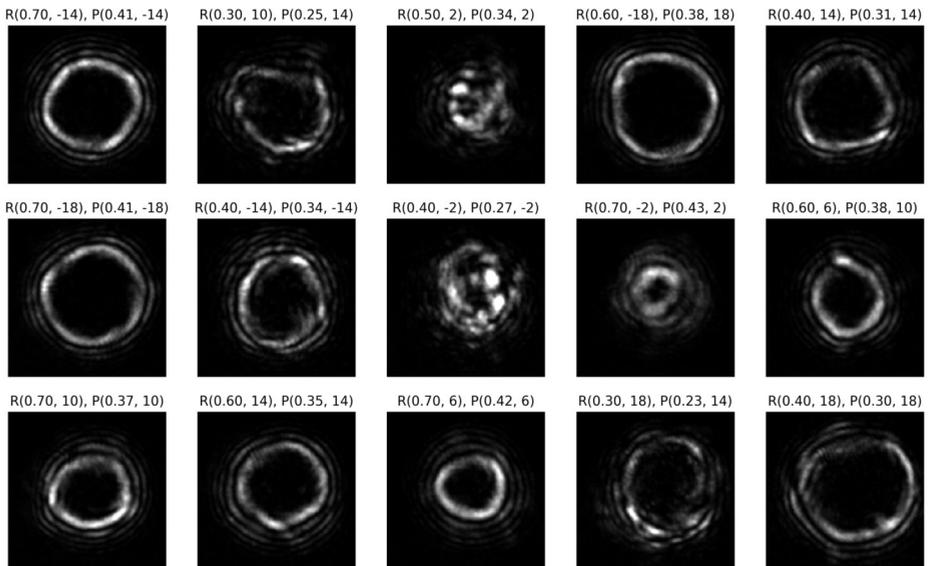


Figura 5.12: Algunas predicciones del modelo simulado sobre el conjunto experimental de prueba.

| Modelo | Conjunto | <i>TC Loss</i> | <i>SR Loss</i> | MAE | Macro AP (%) | Micro AP(%) |
|--------|----------|----------------|----------------|--------|--------------|-------------|
| Exp | Exp | 0.7044 | 0.0109 | 0.0892 | 69.09 | 69.33 |
| Sim | Sim | 0.0251 | 0.0041 | 0.0536 | 98.54 | 98.49 |
| Sim | Exp | 2.8265 | 0.0350 | 0.1633 | 65.38 | 63.33 |

Tabla 5.1: Resumen de resultados. La columna *Modelo* indica el tipo de modelo utilizado para el entrenamiento (experimental o simulado), mientras que la columna *Conjunto* corresponde al conjunto de datos de prueba empleado en la evaluación.

la diagonal principal, que representa las predicciones correctas, también se presentan errores al predecir clases adyacentes. Además, destaca la aparición de un esbozo de la diagonal inversa, lo que significa que el modelo tiene dificultades para identificar correctamente el signo de la carga topológica.

La Figura 5.12 muestra algunas de las predicciones realizadas por el modelo simulado sobre el conjunto experimental de prueba. Se observa que, si bien muchas predicciones coinciden con la clase correcta, el error más frecuente es la clasificación en clases adyacentes a la correcta. En cuanto al valor de turbulencia, las predicciones presentan un desfase considerable respecto al valor real. Este comportamiento era esperable, ya que las imágenes experimentales exhiben una mayor deformación, producto de una turbulencia más intensa, en comparación con las imágenes simuladas, a pesar de compartir el mismo valor de SR.

En la tabla 5.1 se muestra un resumen de los resultados obtenidos al evaluar los modelos en los distintos conjuntos de prueba. A partir de estos resultados se pueden establecer varias conclusiones. Se observa que el mejor desempeño es el obtenido por el modelo simulado sobre las mismas imágenes simuladas. Esto confirma que ambos problemas pueden resolverse eficazmente mediante el uso de modelos basados en CNN, en entornos controlados y con una gran cantidad de datos. Por otro lado, el modelo experimental evaluado sobre las imágenes experimentales no logra un desempeño tan elevado, pero sus resultados resultan aceptables dada la cantidad de datos. Los resultados sugieren una mayor complejidad en el conjunto de datos experimentales que los en el conjunto de datos simulados. Sin embargo, la diferencia más significativa se presenta sobre el modelo simulado evaluado sobre imágenes experimentales (Sim-Exp), teniendo los valores menos favorables en los 3 escenarios analizados. Esta caída en el rendimiento puede atribuirse a la diferencia entre ambos conjuntos de datos. Aunque se procuró que los conjuntos simulado y experimental fueran lo más similares posible, persisten diferencias estructurales que impiden una generalización efectiva del modelo simulado hacia datos reales. No obstante, este resultado no debe interpretarse como un fallo del modelo, sino como una manifestación de que los datos provienen de conjuntos distintos, lo cual justifica la necesidad de crear modelos de simulación mejorados o técnicas de adaptación al conjunto experimental más certeros.

Conclusiones

6.1 Conclusiones generales

Las redes neuronales convolucionales desarrolladas en este trabajo demostraron ser capaces de identificar correctamente las cargas topológicas de los vórtices ópticos. Aunque teóricamente los patrones de intensidad de haces con cargas topológicas de igual valor absoluto (pero distinto signo) son equivalentes, el modelo logró discernir entre signos opuestos en la mayoría de los casos. Este resultado sugiere que podrían existir características sutiles en las imágenes, posiblemente inducidas por asimetrías ópticas, efectos de propagación o ruido, que la red logra aprovechar para su clasificación, aunque aún es necesario investigar con mayor profundidad las razones físicas y computacionales que permiten esta diferenciación.

Si bien se presentaron algunas confusiones en la clasificación de cargas topológicas cercanas, especialmente en el conjunto de datos experimental, como se evidenció en las matrices de confusión, la red logró identificar correctamente la mayoría de estas clases. Sin embargo, una de las limitaciones fue la cantidad de clases utilizadas, restringidas a 10 valores de cargas topológicas con pasos de 4 unidades entre ellas. Dado que este problema es de naturaleza morfológica, las clases presentan un alto grado de similitud entre sí, lo que dificulta la clasificación, la cual se ve agravada por la turbulencia presente en el medio de propagación. Se sugiere la implementación de redes neuronales más profundas o arquitecturas avanzadas que permitan identificar patrones más específicos y robustos, así como técnicas de aprendizaje más avanzadas como el uso de *transfer learning*, con el fin de mejorar la capacidad de clasificación entre clases similares y reducir los solapes inherentes de la naturaleza morfológica del problema y las perturbaciones del medio.

Si bien el código utilizado para simular los vórtices ópticos permite generar re-

presentaciones realistas de estos haces, la transferencia de información a datos reales no logra un buen desempeño. Se identificaron diferencias entre los conjuntos de datos simulados y experimentales, asociadas a artefactos visuales y variaciones en los parámetros de simulación. En particular, las perturbaciones provocadas por la turbulencia del medio no están representadas de forma idéntica a los datos experimentales, por lo cual es necesario elaborar técnicas de simulación adicionales y mejoradas para reducir la brecha entre ambos conjuntos de datos. Con esto se podría generar un conjunto de datos con una correspondencia más fiel a los datos reales que permitan mejorar el desempeño de estos modelos al realizar entrenamientos con datos híbridos.

A pesar de estas limitaciones, este trabajo representa un avance significativo al aplicar tecnologías modernas de aprendizaje profundo a un área emergente como lo es la luz estructurada. La implementación de redes neuronales convolucionales en este contexto no solo abre nuevas posibilidades para la clasificación automática de haces ópticos, sino que también impulsa la integración de herramientas de inteligencia artificial en la investigación en óptica moderna.

Como trabajo futuro, es fundamental ampliar la base de datos experimental, tanto en cantidad de imágenes como en variedad de clases topológicas o tipos de haces, para entrenar redes más robustas. También se propone la implementación de modelos capaces de operar en tiempo real para la identificación de haces multiplexados o demultiplexados, aplicando técnicas de segmentación y clasificación simultáneas que permitan abordar escenarios más complejos y cercanos a aplicaciones prácticas.

Bibliografía

- [1] 7.3. padding and stride — dive into deep learning 1.0.3 documentation. https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html. Accessed: 2025-5-15.
- [2] Artificial intelligence (AI) coined at dartmouth. <https://home.dartmouth.edu/about/artificial-intelligence-ai-coined-dartmouth>. Accessed: 2025-5-15.
- [3] Artificial neural network and supervised learning. In *Studies in Computational Intelligence*, pages 23–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [4] Antonio Ambrosio, Lorenzo Marrucci, Fabio Borbone, Antonio Roviello, and Pasqualino Maddalena. Light-induced spiral mass transport in azo-polymer films under vortex-beam illumination. *Nat. Commun.*, 3(1):989, 2012.
- [5] François Chollet. *Deep learning with python*. Manning Publications, New York, NY, 2022.
- [6] Matthieu Cord and Pádraig Cunningham, editors. *Machine learning techniques for multimedia: Case studies on organization and retrieval*. Springer, Berlin, Germany, 2008 edition, 2008.
- [7] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.
- [8] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer, Dordrecht, Netherlands, 2025.
- [9] Andrew Forbes. Structured light from lasers. *Laser Photon. Rev.*, 13(11):1900140, 2019.

- [10] Andrew Forbes, Michael de Oliveira, and Mark R Dennis. Structured light. *Nat. Photonics*, 15(4):253–262, 2021.
- [11] Michael Franke and Judith Degen. The softmax function: Properties, motivation, and interpretation. 2023.
- [12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Alanna Maldonado, 2023.
- [13] J Matthew Helm, Andrew M Swiergosz, Heather S Haeberle, Jaret M Karnuta, Jonathan L Schaffer, Viktor E Krebs, Andrew I Spitzer, and Prem N Ramkumar. Machine learning and artificial intelligence: Definitions, applications, and future directions. *Curr. Rev. Musculoskelet. Med.*, 13(1):69–76, 2020.
- [14] Christian Keysers and Valeria Gazzola. Hebbian learning and predictive mirror neurons for actions, sensations and emotions. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 369(1644):20130175, 2014.
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
- [16] Y Lecun, L Bottou, Y Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE Inst. Electr. Electron. Eng.*, 86(11):2278–2324, 1998.
- [17] Ting Lei, Meng Zhang, Yuru Li, Ping Jia, Gordon Ning Liu, Xiaogeng Xu, Zhaohui Li, Changjun Min, Jiao Lin, Changyuan Yu, Hanben Niu, and Xiaocong Yuan. Massive individual orbital angular momentum channels for multiplexing enabled by dammann gratings. *Light Sci. Appl.*, 4(3):e257–e257, 2015.
- [18] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity (1943). In *Ideas That Created the Future*, pages 79–88. The MIT Press, 2021.
- [19] R G Morris. D.O. hebb: The organization of behavior, wiley: New york; 1949. *Brain Res. Bull.*, 50(5-6):437, 1999.
- [20] Ivan Nunes Da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco Dos Reis Alves. *Artificial neural networks: A practical course*. Springer International Publishing, Cham, Switzerland, 2018.
- [21] Purwono Purwono, Alfian Ma’arif, Wahyu Rahmaniari, Haris Imam Karim Fathurrahman, Aufaclav Zatu Kusuma Frisky, and Qazi Mazhar ul Haq. Understanding of convolutional neural network (CNN): A review. *International Journal of Robotics and Control Systems*, 2(4):739–748, 2023.
- [22] Carmelo Rosales-Guzmán and Andrew Forbes. How to shape light withspatial light modulators, May 2024.

- [23] F Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958.
- [24] Halina Rubinsztein-Dunlop, Andrew Forbes, M V Berry, M R Dennis, David L Andrews, Masud Mansuripur, Cornelia Denz, Christina Alpmann, Peter Banzer, Thomas Bauer, Ebrahim Karimi, Lorenzo Marrucci, Miles Padgett, Monika Ritsch-Marte, Natalia M Litchinitser, Nicholas P Bigelow, C Rosales-Guzmán, A Belmonte, J P Torres, Tyler W Neely, Mark Baker, Reuven Gordon, Alexander B Stilgoe, Jacqueline Romero, Andrew G White, Robert Fickler, Alan E Willner, Guodong Xie, Benjamin McMorran, and Andrew M Weiner. Roadmap on structured light. *J. Opt.*, 19(1):013001, 2017.
- [25] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [26] Bahaa E A Saleh and Malvin Carl Teich. *Fundamentals of photonics, 2 volume set*. John Wiley & Sons, Nashville, TN, 3 edition, 2020.
- [27] R Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *Int. J. Adv. Res. Artif. Intell.*, 2(2), 2013.
- [28] Yijie Shen, Xuejiao Wang, Zhenwei Xie, Changjun Min, Xing Fu, Qiang Liu, Mali Gong, and Xiaocong Yuan. Optical vortices 30 years on: OAM manipulation from topological charge to multiple singularities. *Light Sci. Appl.*, 8(1):90, 2019.
- [29] Anthony E Siegman. *Lasers*. University Science Books, Sausalito, CA, 1986.
- [30] Orazio Svelto. *Principles of Lasers*. Springer, New York, NY, 5 edition, 2010.
- [31] Juan Terven, Diana-Margarita Cordova-Esparza, Julio-Alejandro Romero-González, Alfonso Ramírez-Pedraza, and E A Chávez-Urbiola. A comprehensive survey of loss functions and metrics in deep learning. *Artif. Intell. Rev.*, 58(7), 2025.
- [32] Guoping Xu, Xiaxia Wang, Xinglong Wu, Xuesong Leng, and Yongchao Xu. Development of residual learning in deep neural networks for computer vision: A survey. *Eng. Appl. Artif. Intell.*, 142(109890):109890, 2025.



Asunto: Revisión de Tesis
León, Gto., a 26 de mayo del 2025

DR. MODESTO ANTONIO SOSA AQUINO
DIRECTOR
DIVISIÓN DE CIENCIAS E INGENIERIAS
CL -UNIVERSIDAD DE GUANAJUATO

A través de la presente constato que he revisado la tesis del C. **Jesús Adrián Mejía Contreras** con el fin de obtener el mérito Laureado en su grado de Licenciatura en Ingeniería Física. El trabajo de tesis se titula **“Determinación de la carga topológica de vórtices ópticos perturbados utilizando redes neuronales convolucionales”**. En este proyecto de investigación realizado por Adrián realizó se propone el uso de redes neuronales convolucionales como herramienta para la clasificación de la carga topológica de vórtices ópticos en medios no ideales, donde también se hace la predicción del nivel de turbulencia atmosférica que ha afectado a dichos haces, que son del tipo paraxial de la familia de Laguerre-Gauss. El trabajo de titulación satisface con la completez y solidez de un proyecto de titulación a nivel licenciatura. También se han hecho las correcciones pertinentes al documento de la tesis. Además, he cuestionado a Adrián sobre los temas relacionados a su trabajo de tesis, demostrando su dominio en los temas abordados en su trabajo de tesis. Por lo que considero que ya se puede proceder con la disertación de tesis.

Sin más por el momento le envío saludos cordiales.

Atentamente

Dr. Carlos Herman Wiechers Medina
Profesor Titular A
Departamento de Física
División de Ciencias e Ingenierías
Campus León
Universidad de Guanajuato
Tel. +52 (477) 7885100 Ext. 3867
Cel. +52 (477) 1080605
e-mail 1: carherwm@fisica.ugto.mx
e-mail 2: ch.wiechers@ugto.mx



León, Gto, a 02 de junio de 2025

Dr. Modesto Antonio Sosa Aquino
Director de la División de Ciencias e Ingenierías
Campus León, Universidad de Guanajuato

Estimado Director:

Por medio de la presente informo a usted que he leído el manuscrito de tesis del estudiante de Ingeniería Física de la DCI, C. Jesús Adrián Mejía Contreras, la cual lleva por título "**Determinación de la carga topológica de vórtices ópticos perturbados utilizando redes neuronales convolucionales**".

Después de haber realizado mis comentarios, el estudiante realizó las correcciones pertinentes. Por lo anterior doy mi consentimiento para que el **C. Jesús Adrián Mejía Contreras** defienda de manera oral su tesis de licenciatura en la fecha que sus directores de tesis juzguen conveniente.

Sin más por el momento, y agradeciendo la atención a la presente, me despido enviando un cordial saludo

Atentamente

Una firma manuscrita en azul que parece decir "Luis Carlos Padierna García".

Dr. Luis Carlos Padierna García
Profesor-Investigador



UNIVERSIDAD
DE GUANAJUATO

Campus León
División de Ciencias e Ingenierías

Oficio número: AGV-2025-05-001

Asunto: Carta conformidad

Tesis de Licenciatura en Ingeniería Física de Jesús Adrián Mejía Contreras,
León Gto., 27 de mayo de 2025.

Dr. Modesto Antonio Sosa Aquino
Director.
División de Ciencias e Ingenierías Campus León

Estimado Dr. Sosa:

Sirva la presente para hacer de su conocimiento que he revisado el trabajo titulado **“Determinación de la carga topológica de vórtices ópticos perturbados usando redes neuronales convolucionales”**. que para obtener el grado de Licenciado en Ingeniería Física pone a consideración el **C. Jesús Adrián Mejía Contreras**. Trabajo de tesis donde funjo como sinodal.

Le comunico que en mi opinión el trabajo reúne las características de nivel y calidad necesarias para una tesis de licenciatura. Asimismo, he discutido con el C. Mejía algunos aspectos de su trabajo y le he indicado las correcciones que considero pertinentes, las cuales ha incluido en la versión final del trabajo. De esta manera no tengo objeciones para el desarrollo de su defensa de acuerdo con la reglamentación respectiva.

Agradeciendo la atención prestada a la presente, me despido

A T E N T A M E N T E.
“LA VERDAD OS HARÁ LIBRES”

Dr. Arturo González Vega
Profesor Titular “A”
Correo: gonzart@ugto.mx
Ext. Telefónica 8431

C.c.p. Jesús Adrián Mejía Contreras
C.c.p. Archivo AGV.

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA

Lomas del Bosque #103,
Lomas de Campestre, León Gto.

C.P. 37150

(477) 788 5100 Ext. 8435, Fax. Ext. 8410

www.diqeb.ugto.mx



León Gto., a 02 de junio de 2025

Dr. Modesto Antonio Sosa Aquino

Director de la División de Ciencias e Ingenierías

Campus León, Universidad de Guanajuato

Por medio de la presente, me permito informarle que he revisado el trabajo de tesis del estudiante de la Licenciatura en Ingeniería Física, Jesús Adrián Mejía Contreras, titulado:

“Determinación de la carga topológica de vórtices ópticos perturbados utilizando redes neuronales convolucionales”.

Una vez emitidos mis comentarios, el estudiante realizó las correcciones correspondientes de forma satisfactoria. En virtud de ello, otorgo mi aprobación para que el estudiante Jesús Adrián Mejía Contreras proceda a la defensa oral de su tesis de licenciatura, en la fecha que decida pertinente.

Sin otro particular, agradezco de antemano su atención y le envío un cordial saludo.

Atentamente,

Fernando Arce Vega

Dr. Fernando Arce Vega

farce@cio.mx

Profesor-Investigador.





León, Gto., a 02 de junio de 2025

Dr. Modesto Antonio Sosa Aquino
Director de la División de Ciencias e Ingenierías
Campus León, Universidad de Guanajuato

Estimado Dr. Sosa:

Por medio de la presente, me permito informarle que he revisado el trabajo de tesis del **C. Jesús Adrián Mejía Contreras**, estudiante de la Licenciatura en Ingeniería Física, titulado: **“Determinación de la carga topológica de vórtices ópticos perturbados utilizando redes neuronales convolucionales”**.

Después de emitir mis observaciones, el estudiante realizó las correcciones pertinentes de forma satisfactoria. Asimismo, he sostenido conversaciones con el tesista sobre diversos aspectos de su trabajo, constatando su dominio del tema y la solidez de su propuesta.

En virtud de lo anterior, doy mi aprobación para que el C. Jesús Adrián Mejía Contreras proceda a la defensa oral de su tesis de licenciatura, en la fecha que se considere conveniente.

Sin más por el momento, agradezco de antemano su atención y le envío un cordial saludo.

Atentamente,

Dr. Carmelo Rosales Guzmán
carmelorosalesg@cio.mx
Profesor - Investigador

