

UNIVERSIDAD NACIONAL DE CHIMBORAZO FACULTAD DE INGENIERÍA CARRERA INGENIERÍA EN TELECOMUNICACIONES

Evaluación del método de aprendizaje profundo para el diseño de un transmisor-receptor de microondas para redes en la banda media de 5G

Trabajo de Titulación para optar al título de Ingeniero en Telecomunicaciones

Autor: Huebla Huilca, Santiago Marcelo

Tutor: PhD. Pedro Fernando Escudero Villa

Riobamba, Ecuador. 2025

DECLARATORIA DE AUTORÍA

Yo, Santiago Marcelo Huebla Huilca, con cédula de ciudadanía 060417327-8, autor del trabajo de investigación titulado: **"Evaluación del método de aprendizaje profundo para el diseño de un transmisor-receptor de microondas para redes en la banda media de 5G**", certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mi exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la norma referida será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba. 22 de abril de 2025.



Santiago Marcelo Huebla Huilca. C.I: 0604173278

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

En la Ciudad de Riobamba, a los 22 días del mes de Abril de 2025, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante SANTIAGO MARCELO HUEBLA HUILCA con CC: 0604173278, de la carrera TELECOMUNICACIONES y dando cumplimiento a los criterios metodológicos exigidos, se emite el ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN titulado "EVALUACION DEL MÉTODO DE APRERNDIZAJE PROFUNDO PARA EL DISEÑO DE UN TRANSMISOR-RECEPTOR DE MICROONDAS PARA REDES EN LA BANDA MEDIA DE 5G", por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



PhD. Pedro Fernando Escudero Villa TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación "EVALUACIÓN DEL MÉTODO DE APRENDIZAJE PROFUNDO PARA EL DISEÑO DE UN TRANSMISOR RECEPTOR DE MICROONDAS PARA REDES EN LA BANDA MEDIA DE 5G", presentado por SANTIAGO MARCELO HUEBLA HUILCA, con cédula de identidad número 060417327-8, bajo la tutoría del PhD. Pedro Fernando Escudero Villa; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 27 de mayo de 2025.

PhD. Daniel Santillán PRESIDENTE DEL TRIBUNAL DE GRADO

PhD. Antonio Meneses MIEMBRO DEL TRIBUNAL DE GRADO

PhD. Carlos Peñafiel MIEMBRO DEL TRIBUNAL DE GRADO

ert Palli

CERTIFICADO ANTIPLAGIO



Dirección Académica VICERRECTORADO ACADÉMICO



CERTIFICACIÓN

Que, HUEBLA HUILCA SANTIAGO MARCELO con CC: 0604173278, estudiante de la Carrera TELECOMUNICACIONES, Facultad de INGENIERÍA; ha trabajado bajo mi tutoría el trabajo de investigación titulado "EVALUACIÓN DEL MÉTODO DE APRENDIZAJE PROFUNDO PARA EL DISEÑO DE UN TRANSMISOR- RECEPTOR DE MICROONDAS PARA REDES EN LA BANDA MEDIA DE 5G ", cumple con el 3 %, de acuerdo al reporte del sistema Anti plagio COMPILATIO, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente, autorizo continuar con el proceso.

Riobamba, 22 de abril de 2025



PhD. Pedro Fernando Escudero Villa TUTOR

DEDICATORIA

A Dios, por guiar cada paso de mi trayectoria y brindarme la fuerza y valentía para superar las dificultades.

A mis padres, Marcelo y Mirian que han sido mi apoyo durante este proceso y han depositado toda su confianza en mí.

A mis hermanitos, Julissa y Kevin que fueron mi inspiración para superarme y nunca rendirme a ellos dedico mi esfuerzo reflejado en este trabajo. Espero que mi logro sea motivo de inspiración para sus vidas.

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a mis familiares y amigos que estuvieron presentes de una u otra manera en mi vida universitaria.

A mis padres y hermanos por su incondicional apoyo en cada momento. Sus enseñanzas y sacrificio han sido parte fundamental para alcanzar este logro.

También quiero agradecer a mis amigos que me han ayudado con palabras de aliento en momentos difíciles.

A mi tutor, PhD. Fernando Escudero por brindarme su guía en cada etapa del presente trabajo. Su conocimiento y experiencia han sido de gran ayuda para alcanzar mis objetivos.

Finalmente, quiero agradecer a mi enamorada, Estefanía, por su cariño y por estar en los buenos y malos momentos.

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO ANTIPLAGIO

DEDICATORIA

AGRADECIMIENTO

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

ABSTRACT

CAPÍ	ÍTULO I	16
1.1	INTRODUCCIÓN	16
1.2	Planteamiento del problema y justificación	17
1.3	OBJETIVOS	18
1.3.1	Objetivo General	18
1.3.2	Objetivos Específicos	18
CAPÍ	ÍTULO II	19
2.1	MARCO TEÓRICO	19
2.1.1	Estado del Arte	19
2.1.2	Términos y Definiciones	21
2.1.3	Aprendizaje profundo	21
2.1.4	Estructura de una red neuronal profunda	22
2.1.5	Redes neuronales artificiales	23
2.1.6	Peso	23
2.1.7	Función de activación	23
2.1.8	Sesgo	24
2.1.9	Métodos de diseño de sistemas microondas	25
2.1.1	0 Aplicación del aprendizaje profundo en comunicaciones inalámbricas	27
2.1.1	1 Entrenamiento de una red neuronal	28
2.1.12	2 Proceso de aprendizaje	29

2.1.13	Métricas de rendimiento	. 29
CAPÍI	TULO III	. 31
3.1 N	METODOLOGÍA	. 31
3.1.1	Tipo de investigación	32
3.1.2	Población y muestra	32
3.1.3	Operación de variables	. 33
3.2 I	Diseño del sistema	33
3.2.1	Filtro de pasa banda de líneas acopladas	. 34
3.2.2	Amplificador microondas	37
3.2.3	Mezclador RF	40
3.2.4	Generación de datos	42
3.2.5	Aplicación de la red neuronal	. 43
CAPÍT	TULO IV	. 44
4.1 I	RESULTADOS Y DISCUSIÓN	44
4.1.1	Valores filtro de líneas acopladas base	44
4.1.2	Valores amplificador microondas base	45
4.1.3	Valores mezclador base	47
4.1.4	Integración transmisor base	48
4.1.5	Integración receptor base	. 48
4.2 H	Entrenamiento y evaluación	52
4.2.1	Red neuronal 1	. 52
4.2.2	Red neuronal 2	. 54
4.2.3	Red neuronal 3	. 55
4.2.4	Red neuronal 4	. 57
4.2.5	Análisis de errores	. 59
CAPÍT	TULO V	. 63
5.1 (CONCLUSIONES	63
5.2 I	RECOMENDACIONES	. 64
BIBLI	OGRAFÍA	. 65
ANEX	COS	. 69
7.1 A	Anexo 1: Códigos Matlab	69
7.1.1	Cálculo de dimensiones de filtro microondas	. 69

7.1.2	Análisis de estabilidad del transistor BFP420, cálculo de líneas de transistor BFP420, cálculo de transistor BFP420, cálculo de transistor BFP420, cálculo de transist	ransmisión
y stul	bs para el amplificador	70
7.1.3	Cálculo de las líneas del acoplador direccional	71
7.1.4	Cálculo de los componentes del filtro pasa bajo	71
7.2	Anexo 2: Códigos Python	72
7.2.1	Creación de red neuronal	72
7.2.2	Realizar predicciones	72
7.3	Anexo 3: Tablas de entrenamiento	73
7.3.1	Entrenamiento red neuronal 1	73
7.3.2	Entrenamiento red neuronal 2	74
7.3.3	Entrenamiento red neuronal 3	75
7.3.4	Entrenamiento red neuronal 4	76

INDICE DE FIGURAS

Figura 1: Clasificación de la inteligencia artificial	21
Figura 2: Estructura en capas de una red neuronal [16]	22
Figura 3: Unidad básica de una red funcional (perceptrón)	23
Figura 4: Ejemplos de funciones de activación más utilizadas	24
Figura 5: Variación del sesgo en una función Sigmoide	25
Figura 6: Estructura completa de perceptrón	25
Figura 7: Proceso de entrenamiento de la red neuronal [16]	28
Figura 8: Resultados de convergencia neuronal [16]	29
Figura 9: Fases del proyecto	31
Figura 10: Diagrama en bloques de los componentes del sistema transmisor-receptor	34
Figura 11: Selección del sustrato	37
Figura 12:Transistor BFP-420 [36].	38
Figura 13: Red de acoplo stub simple en paralelo [28]	40
Figura 14: Configuración de un mezclador: (a) Up conversión, (b) Down conversión	41
Figura 15: Componentes del mezclador.	41
Figura 16: Librerías y paquetes necesarios para la red neuronal	43
Figura 17: Filtro de líneas acopladas ADS	44
Figura 18: Parámetros S filtro de líneas acopladas ADS	45
Figura 19: Amplificado de microondas ADS	46
Figura 20: Máxima ganancia disponible vs máxima ganancia obtenida	46
Figura 21: Mezclador ADS	47
Figura 22: Análisis de armónicos up-conversion y down conversion	48
Figura 23: Transmisor base ADS.	48
Figura 24: Receptor base ADS	49
Figura 25: Señales de entrada y salida del transmisor en tiempo y frecuencia	49
Figura 26: Señales de entrada y salida del receptor en tiempo y frecuencia	50

Figura 27: Señales de entrada y salida de los componentes del sistema	51
Figura 28: Potencia de entrada y salida del receptor	51
Figura 29: Comparación error mse y mae, red neuronal 1	53
Figura 30: Parámetros S filtro de líneas acopladas, 5 entrenamientos realizados	53
Figura 31: Comparación error mse y mae, red neuronal 2	55
Figura 32: Ganancia máxima amplificador, 5 entrenamientos realizados	55
Figura 33: Comparación error mse y mae, red neuronal 3	56
Figura 34: Parámetros S11 del acoplador direccional, 5 entrenamientos realizados	57
Figura 35: Comparación error mse y mae, red neuronal 4	58
Figura 36: Parámetros S21 del filtro pasa bajo, 5 entrenamientos realizados	58
Figura 37: Porcentaje de reducción de error mse y mae	59
Figura 38: Promedio reducción de error mse y mae	59
Figura 39: Señales de entrada y salida del transmisor utilizando Deep Learning	60
Figura 40: Señales de entrada y salida del receptor utilizando Deep Learning	60
Figura 41: Comparativa de señales del transmisor	61
Figura 42: Comparativa de señales del receptor	61
Figura 43: Comparativa de potencia de entrada y potencia de salida	62

INDICE DE TABLAS

Tabla 1: Resumen de aplicaciones de aprendizaje profundo y redes neuronales orienta	das a
sistemas de microondas.	20
Tabla 2: Variables dependientes e independientes	33
Tabla 3: Características del componente	34
Tabla 4: Valores de polinomios para filtro Chevychev pasa bajo con rizado de 0.5 y 3. de rizado [28]	.0 dB 35
Tabla 5: Datos del fabricante transistor BFP420.	38
Tabla 6: Variables entrada y salida filtro de líneas acopladas	42
Tabla 7: Variables entrada y salida amplificador microondas	42
Tabla 8: Variables entrada y salida mezclador microondas.	43
Tabla 9: Secciones del filtro con sus valores	44
Tabla 10: Parámetros calculados del amplificador	45
Tabla 11: Parámetros calculados del mezclador.	47
Tabla 12: Resultado red neuronal 1	52
Tabla 13: Parámetros red neuronal 2	54
Tabla 14: Parámetros red neuronal 3	56
Tabla 15: Parámetros red neuronal 4	57
Tabla 16: Valores de potencia y ganancia diseño base vs Deep Learning	62
Tabla 17: Características de la red neuronal 1 (filtro)	73
Tabla 18: Valores Deep Learning vs valores calculado Red neuronal 1	73
Tabla 19: Características de la red neuronal 2 (amplificador)	74
Tabla 20: Valores Deep Learning vs valores calculados red neuronal 2	74
Tabla 21: Características de la red neuronal 3 (Acoplador direccional)	75
Tabla 22: Valores Deep Learning vs Valores calculados, red neuronal 3	75
Tabla 23: Características de la red neuronal 4, (Filtro pasa bajo)	76
Tabla 24: Valores Deep Learning vs Valores calculados, Red neuronal 4	76

RESUMEN

El desarrollo de nuevos sistemas y dispositivos de comunicación requiere métodos que permitan analizar patrones, optimizar su funcionamiento y ofrecer mejoras alineadas con las necesidades actuales. Con este enfoque, el presente trabajo evalúa el uso del aprendizaje profundo como alternativa para diseñar un sistema transmisor-receptor de microondas. Se inició con una búsqueda bibliográfica sobre modelos y arquitecturas de aprendizaje automático aplicados a dispositivos microondas. Luego, se desarrolló un diseño base del transmisor-receptor con cuatro etapas: generación de señal, amplificación, mezclado y filtrado, operando bajo el estándar 5G-NR FR1 y transmitiendo una señal en banda base de 100 MHz. Los componentes fueron diseñados individualmente según los requerimientos del sistema. Mediante simulaciones, se verificó su correcto funcionamiento y se generaron datos de entrenamiento. En un entorno Python, se programaron cuatro redes neuronales con arquitectura convolucional y modelo secuencial para organizar capas de entrada, ocultas y salida. Se usaron las métricas MSE y MAE para evaluar el rendimiento, obteniendo una reducción de error promedio superior al 90%. Además, se evaluó el sistema mediante ganancia máxima, ancho de banda, balance de armónicos y parámetros S, logrando recuperar la señal en banda base con una potencia de salida de -32.912 dBm y una ganancia de 0.738 dB.

Palabras clave: Redes neuronales, sistemas de microondas, estándar 5G -NR FRI, aprendizaje profundo, transmisor, receptor, frecuencia.

ABSTRACT

The development of new communication systems and devices requires methods that enable the analysis of patterns, the optimization of functionality, and the implementation of improvements aligned with current requirements. With this approach, the present study evaluates the use of deep learning as an alternative for designing a microwave transmitterreceiver system. The work began with a literature review on machine learning models and architectures applied to microwave devices. A baseline transmitter-receiver design was then developed with four stages: signal generation, amplification, mixing, and filtering, operating under the 5G-NR FR1 standard and transmitting a baseband signal at 100 MHz. The components were individually designed according to the system requirements. Simulations were used to verify their proper operation and to generate training data. In a Python environment, four neural networks with convolutional architecture and a sequential model were programmed to organize input, hidden, and output layers. MSE and MAE metrics were used to evaluate performance, achieving an average error reduction of over 90%. Additionally, the system was assessed using maximum gain, bandwidth, harmonic balance, and S-parameters, successfully recovering the baseband signal with an output power of -32.912 dBm and a gain of 0.738 dB.

Keywords: Neural networks, microwave systems, 5G-NR FRI standard, deep learning, transmitter, receiver, frequency.



Reviewed by: MsC. Edison Damian Escudero ENGLISH PROFESSOR C.C.0601890593

CAPÍTULO I

1.1 INTRODUCCIÓN

En la actualidad, los sistemas de comunicación, la tecnología de radiofrecuencia (RF) y los dispositivos de microondas han transformado la conectividad, al proporcionar velocidades de transmisión de datos altas y habilitar múltiples aplicaciones avanzadas [1]. En este contexto, la necesidad de mejorar la eficiencia y capacidad de adaptación de estos sistemas ha impulsado el desarrollo de nuevas herramientas que se complementen a los métodos de diseño tradicionales.

El aprendizaje profundo, una rama destacada de la inteligencia artificial [2], permite identificar patrones complejos y optimizar sistemas en diferentes dominios. Particularmente, las redes neuronales artificiales y redes neuronales profundas estructuras de retro alimentación con múltiples capas ocultas han demostrado ser herramientas eficaces para el modelado de circuitos de microondas. Diversos estudios han aplicado estas redes al diseño filtros de microondas [3], dispositivos microondas no lineales [4], componentes paramétricos y modelado multifísico [5].

Investigaciones recientes destacan el potencial del aprendizaje profundo en el diseño de transmisores y emisores de microondas. Es posible implementar métodos de aprendizaje automático para la simulación rápida en el dominio de la frecuencia de modelos paramétricos como lo propuesto por Mattuci [6]. Adicionalmente, el aprendizaje profundo puede ser empleado para predecir el comportamiento de estructuras de comunicaciones para bandas de frecuencia establecidas [7].

Con lo expuesto, el presente trabajo se centra en la evaluación del método de aprendizaje profundo como alternativa en el diseño de sistemas transmisor-receptor de microondas para redes en la banda media de 5G. El trabajo está estructurado por 5 capítulos. En el capítulo 1 se presenta la problemática y justificación, así como los objetivos. En el capítulo 2 se muestra el estado del arte y fundamento teórico. En el capítulo 3 se explica la metodología empleada y el procedimiento. En el capítulo 4 se presentan los resultados y la evaluación del método propuesto. Luego de realizar pruebas y simulaciones en el capítulo 5 se muestran las principales conclusiones y recomendaciones.

1.2 Planteamiento del problema y justificación

En los últimos años, el rápido desarrollo de los dispositivos y sistemas electrónicos y de las comunicaciones inalámbricas implica contar con dispositivos microondas cada vez más sofisticados. En consecuencia, el modelado y diseño de circuitos de este tipo se hace más complejo y desafiante en respuesta a la demanda de una mayor funcionalidad, una mayor fiabilidad y un ciclo de diseño más corto. Considerando lo anterior, es necesario disponer de herramientas adicionales que se complementen a los procesos de diseños ya conocidos.

Los métodos de aprendizaje automático basados en redes neuronales se perfilan como una alternativa por su capacidad para analizar patrones complejos en datos [8]. En este ámbito, investigaciones recientes implementan estos métodos para el modelado y diseño de dispositivos microondas como filtros, amplificadores, antenas, entre otros. Sin embargo, la mayoría de estas investigaciones enfocan la aplicación de modelos de aprendizaje automático al modelado u optimización de un componente en específico y no de un sistema de comunicación completo.

El trabajo propuesto muestra cómo el aprendizaje profundo es capaz de comprender las relaciones de entrada y salida de componentes microondas mediante la generación de datos y el entrenamiento de redes neuronales, en ese caso se demuestra que es posible integrar varios componentes a un sistema completo de comunicación en un entorno 5G para que su funcionamiento se asemeje a un sistema diseñado con métodos convencionales.

La viabilidad de esta investigación se sustenta en la capacidad de generar datos de entrenamiento e implementar redes neuronales sin la necesidad de contar con un ordenador sofisticado o de un servidor dedicado ya que el volumen de datos a trabajar no requiere de recursos computacionales excesivos.

1.3 OBJETIVOS

1.3.1 Objetivo General

Evaluar la pertinencia y contribución de la aplicación del aprendizaje profundo en el diseño de un transmisor-receptor de microondas para redes 5G en banda media.

1.3.2 Objetivos Específicos

- Realizar un estudio sobre los modelos de aprendizaje profundo aplicados a sistemas de microondas
- Aplicar una arquitectura de red neuronal profunda específicamente adaptada al diseño del sistema transmisor-receptor de microondas en el contexto de redes 5G.
- Diseñar un sistema de transmisor-receptor microondas utilizando modelos de aprendizaje profundo.
- Evaluar el rendimiento del sistema de comunicaciones desarrollado, considerando parámetros clave de eficiencia y calidad de servicio en entornos 5G mediante simulaciones.

CAPÍTULO II

2.1 MARCO TEÓRICO

2.1.1 Estado del Arte

Con el objetivo de comprender los aspectos más importantes del aprendizaje profundo y sistemas de microondas, se realiza una revisión bibliográfica donde se analizan investigaciones de relevancia. que guardan estrecha relación con el tema.

En el año 1998. Peik, menciona la introducción de las redes neuronales en el campo de la ingeniería de microondas, el autor propone un modelo basado en redes neuronales en el que compila los datos de entrenamiento mediante una simulación electromagnética y los emplea en el modelado de una antena parche microstrip y un filtro plano de 3 polos, con las redes neuronales entrenadas se logra optimizar el diseño de los circuitos. Sin embargo, resalta que el costo computacional para entrenar estos modelos es alto, por lo que para la época en la que fue realizada la investigación representó una limitante [9].

A partir del trabajo de Peik, y con el creciente desarrollo de la tecnología, años más tarde se introducen nuevas técnicas de entrenamiento de datos que facilitan el aprendizaje de las redes neuronales con un costo computacional razonable. Adicionalmente, existen varios estudios de interés que exploran diferentes tipos de redes neuronales para el aprendizaje profundo con diferentes aplicaciones. Dong en el año 2015, enfoca su investigación en las redes neuronales convolucionales (CNN) y destaca que son una parte importante del aprendizaje profundo ya que son capaces de detectar patrones locales en los datos [10]. Esto las hace muy eficientes para tareas como la detección de objetos, clasificación de imágenes y la detección de anomalías [11].

En el campo de las comunicaciones inalámbricas se ha explorado el uso de redes neuronales convolucionales (CNN). En el año 2019, Sun, introdujo un método llamado filtrado neuronal convolucional, que utiliza kernels de convolución unidimensionales para aprender un filtro coincidente o de paso de banda, con un mecanismo de atención basado en el kernel para enfatizar los filtros informativos [12]. Marseet por su parte, propuso un modelo de CNN de Valor Complejo para el proceso de detección en redes inalámbricas de próxima generación, logrando una reducción de la complejidad computacional [13].

Existe una investigación de alto interés como la propuesta por Jing en el año 2021, en la cual introduce el método de redes neuronales profundas (DNN) en el campo del modelado de sistemas de microondas de alta dimensión [14]. La técnica de red neuronal profunda propuesta puede resolver problemas de modelado en una dimensión más alta que métodos como el de red neuronal superficial. Esta técnica puede representar la relación fluida de entrada-salida que se requiere para el modelado de microondas. Se propone un algoritmo avanzado de aprendizaje profundo de tres etapas para entrenar el nuevo modelo de red neuronal profunda. Este estudio se destaca las aplicaciones del aprendizaje profundo para mejorar el rendimiento y las capacidades de los sistemas de comunicaciones.

Otra aplicación de las redes neuronales artificiales es discutida en la investigación realizada por Raghavan, en la cual propone un nuevo enfoque para desarrollar modelos precisos y

rápidos que puedan ser utilizados para simulaciones CAD [15]. El modelo propuesto emplea módulos computacionales de redes neuronales como una herramienta para el modelado y diseño RF y microondas. Las redes neuronales pueden ser entrenadas para aprender el comportamiento de componentes/circuitos pasivos/activos. En este contexto, el investigador presenta modelos de redes neuronales para filtros de microondas que juntamente con las redes entrenadas han optimizado su funcionamiento según ciertas especificaciones.

Con el pasar del tiempo el uso del aprendizaje profundo fue tomando más relevancia para el modelado y diseño de componentes microondas. En el año 2020 Corral, en su trabajo de fin de Máster estudia la nueva tendencia de usar algoritmos de Machine Learning para el modelado y linealización de amplificadores de potencia cuya aplicación permite obtener modelos de comportamientos más precisos, superando varias de las limitaciones existentes en modelos tradicionales [16].

En el trabajo de Corral también se abordan conceptos que resultan de gran interés ya que son parte fundamental en el desarrollo de una red neuronal. Entre los principales se destaca la arquitectura de una red neuronal, función de activación, sesgo, convolución y por último explica cómo funciona el proceso de aprendizaje de la red y los posibles resultados de convergencia para de esta manera evitar las patologías típicas de un mal entrenamiento.

En el ámbito local en el año 2023. Avila, en su trabajo de titulación de pregrado realizó un análisis de 2 algoritmos de aprendizaje máquina los cuales demostraron ser una herramienta eficaz para mejorar el desempeño de las redes 5G en situaciones de sobrecarga [17]. Los algoritmos propuestos son capaces de adaptarse a los cambios repentinos en la intensidad del tráfico, lo que reduce las colisiones de preámbulos y mejora la probabilidad exitosa de acceso a los recursos de la red.

Este trabajo tiene implicaciones importantes para el diseño y la operación de redes 5G. Los algoritmos propuestos se emplean para mejorar el rendimiento de dichas redes en entornos con alta densidad de usuarios y tráfico variable.

Luego de realizar una búsqueda bibliográfica En la Tabla 1 se muestra un resumen de aplicaciones de relevancia donde se presentan aplicaciones de redes neuronales que son de interés para el presente trabajo.

Aplicación	Tipo de red neuronal empleada	Función de activación	Número de capas	Eficacia	Referencia
Sintonización de Filtros resonadores acoplados	Red neuronal convolucional (CNN)	ReLU,Leaky ReLU, Flatten,Sigmoide	6 capas	Rápida convergencia y alta precisión	[18]
Extracción de parámetros de filtros de alta dimensión	Red neuronal profunda (DNN)	ReLU fluida y Sigmoide	14 capas	Rápida convergencia y extracción de parámetros en	[14]

Tabla 1: Resumen de aplicaciones de aprendizaje profundo y redes neuronales orientadas a sistemas de microondas.

				elementos de alta dimensión	
Antena parche microstrip, filtro plano de 3 polos	Red de perceptrón multicapa (MLP)	Función logística	5 capas	Buena concordancia en todo el rango de frecuencias de prueba	[9]
Modelado de amplificadores de potencia	Red neuronal artificial compleja basada en CR- Cálculo	Algoritmo de árbol de decisión	3 capas	Problemas de convergencia, modelo eficaz para selección de patrones	[19]
Modelado y predistorsión de amplificadores de potencia	Red convolucional con retardo de tiempo de valor real (RVTDCNN)	Tangente hiperbólica	4 capas	Buena precisión, bajo costo computacional, ajuste adecuado en señal de salida	[16]

2.1.2 Términos y Definiciones

2.1.3 Aprendizaje profundo

El Aprendizaje Profundo es un subconjunto del Machine Learning y de la inteligencia artificial, implica el aprendizaje en múltiples niveles de abstracción para extraer información útil de datos [20]. En la Figura 1 se muestra cómo se relacionan progresivamente estas disciplinas donde la inteligencia artificial es el campo general que desarrolla entornos y sistemas con capacidades que se asemejan a la inteligencia humana, el Machine Learning permite aprender patrones, crear algoritmos y ejecutar acciones a partir de datos.



Figura 1: Clasificación de la inteligencia artificial.

Por su parte el Deep Learning se caracteriza por el uso de múltiples capas ocultas en redes neuronales artificiales, mediante transformaciones no lineales y abstracciones de modelos de alto nivel en grandes bases de datos [21]. Este enfoque se ha aplicado en varios campos, como la inteligencia artificial, el procesamiento de imágenes, la robótica y la automatización [22]La capacidad del aprendizaje profundo para aprender de cantidades masivas de datos ha impulsado su uso en tareas cognitivas complejas como: detección de objetos, el reconocimiento visual de patrones, el reconocimiento de voz [23], incluso se ha reportado el uso de modelos de aprendizaje profundo el campo de las finanzas para realizar predicciones del comportamiento de divisas [24].

2.1.4 Estructura de una red neuronal profunda

Una red neuronal es un sistema de procesamiento de información que está construido por dos tipos de componentes principales, neuronas y enlaces. Las neuronas son los elementos de procesamiento y los enlaces son las interconexiones entre sí [25]. En una red neuronal cada neurona recibe estímulos o entradas de neuronas vecinas conectadas a ella, procesa la información y produce una salida. Las neuronas pueden procesar la información de distintas maneras, y las conexiones entre ellas pueden variar según el propósito. En la Ecuación 1, se muestra la relación entrada-salida de un modelo de red neuronal estándar.

$$y = y(x, w) \tag{1}$$

Siendo x un vector de tamaño N_X X_1 que contiene las **entradas** externas al modelo de red neuronal y siendo y un vector de tamaño N_X X_1 que contiene las **salidas**, w es un vector que contiene todos los parámetros de peso que representan diversas interconexiones en el modelo.

Una red neuronal puede estar conformada por varias capas según su aplicación, estas capas realizan transformaciones de manera lineal y no lineal a partir de información de las entradas de la red y entregan una salida esperada que se ubica en la última capa. En la Figura 2 se presenta una red neuronal conformada por 3 capas, la capa de entrada es la que recepta los datos y la capa de salida devuelve el resultado de la predicción deseada. La capa ubicada en la mitad se denomina capa oculta o capa profunda y pueden ser tantas según se requiera, el número de neuronas también puede variar en cada capa. Estas neuronas se interconectan entre cada una de las interfaces.



Figura 2: Estructura en capas de una red neuronal [16].

2.1.5 Redes neuronales artificiales

Las redes neuronales artificiales son modelos matemáticos que imitan el funcionamiento del cerebro humano, son capaces de aprender y almacenar conocimiento y gran cantidad de información. Estas redes están compuestas por neuronas artificiales, que están interconectadas entre sí. Las neuronas artificiales reciben señales de entrada y producen señales de salida. El aprendizaje en las redes neuronales artificiales se produce a través de un proceso de ajuste de pesos y sesgos de las conexiones entre las neuronas y pueden imitar el sistema nervioso biológico mediante la construcción de un gran número de neuronas simuladas [26].

Para lograr una mejor comprensión de las redes neuronales artificiales es necesario partir del modelo más simple, en este caso la unidad básica de una red funcional se la denomina perceptrón. A esta estructura se incorpora el peso, función de activación y finalmente el sesgo que son elementos fundamentales para el correcto funcionamiento de la red. A continuación, se analizará los conceptos antes mencionados.

2.1.6 Peso

En la Figura 3 se muestra la estructura de una unidad básica de red, la neurona recibe valores en su entrada y estas tienen un valor o ponderación asociada denotada por w. Este valor es lo que se conoce como peso y, en otras palabras, es la jerarquía que tiene cada componente de entrada.



Figura 3: Unidad básica de una red funcional (perceptrón).

2.1.7 Función de activación

En una red neuronal, la función de activación se encarga de propagar hacia adelante la salida de una neurona e introducen la no linealidad en el comportamiento de estructuras computacionales similares al comportamiento del cerebro humano. La función de activación también se conoce como función de aplazamiento ya que su trabajo es limitar el rango de amplitud permitido de la señal de salida a un valor infinito [27].

Existen varias funciones de activación que se pueden emplear según el campo de aplicación. En la Figura 4 se muestran ejemplos de funciones de activación con sus respectivas expresiones matemáticas, es posible realizar combinaciones de estas funciones para que entreguen el resultado esperado, esto se lo realiza directamente en las capas de la red neuronal.



Figura 4: Ejemplos de funciones de activación más utilizadas.

2.1.8 Sesgo

Es otro de los parámetros que se considera al momento de realizar un modelo de red neuronal. Su función es añadir la suma ponderada de los productos de cada neurona. El sesgo desplaza la función de activación hacia adelante o hacia atrás y de esta manera mejorar la capacidad de la red.

Según el tipo de función de activación el sesgo se aplica para que la red neuronal aprenda a representar datos y facilita la convergencia y ajuste del modelo de aprendizaje. En la Figura 5 se puede apreciar cómo se ingresa una variación de sesgo para una función de activación del tipo Sigmoide se evidencia que el sesgo desplaza la curva para de esta manera producir un valor esperado. En este caso se consideran 3 valores de sesgo, donde la función se desplaza hacia la izquierda (-3) y se activará con entradas bajas, donde el sesgo se centra en el origen o conocida como sigmoide estándar (0), en este escenario no existirá desplazamiento y se activará solo con una entrada positiva (0.5). Finalmente, la función desplazada hacia la derecha (3) se activará cuando existan valores. Esta variación en el sesgo permite a los modelos de aprendizaje activar las neuronas de acuerdo a los requerimientos de la red.



Figura 5: Variación del sesgo en una función Sigmoide.

En la Figura 6 se muestra como todos los elementos que componen el perceptrón se incorporan para formar su estructura básica.



Figura 6: Estructura completa de perceptrón.

2.1.9 Métodos de diseño de sistemas microondas

En el ámbito de las telecomunicaciones el diseño de sistemas de comunicaciones es fundamental para transmitir información a través de largas distancias. Estos sistemas requieren métodos de modelado que permitan emular su comportamiento y en muchos de los casos mejorar su rendimiento, esto se logra con la ayuda de modelos matemáticos, de análisis de circuitos y herramientas computacionales de simulación. A continuación, se describen los métodos de diseño más empleados para el diseño de sistemas de microondas.

Modelado de circuitos

Este método permite analizar de manera individual los componentes de un sistema como: amplificadores, filtros, mezcladores, osciladores. Existen herramientas de diseño asistido por computadora como SPICE y ADS que combinan análisis de teórico y análisis de red para modelar sistemas de radio frecuencia y microondas. Según Pozar [28], se debe considerar dos enfoques de análisis en esta técnica la primera es el análisis de circuitos de baja frecuencia, la cual se basa en la interconexión de componentes pasivos o activos concentrados, las dimensiones de este tipo de circuitos son de menor tamaño y se pueden utilizar leyes de Kirchhoff o conceptos de impedancia de teoría de circuitos. La segunda es el análisis de microondas en las que se analizan los campos eléctricos y magnéticos, requiere el uso de las leyes de Maxwell y su caracterización es más simple y eficiente.

Modelado de propagación

Un modelo de propagación es empleado para predecir como se propagan las ondas de microondas a través del espacio esto es posible gracias a expresiones matemáticas, diagramas y algoritmos [29], los modelos consideran parámetros como la atenuación, la refracción, difracción.

Los modelos de propagación se pueden clasificar según el entorno en el cual se los aplique. Como ejemplos se puede mencionar el modelo de Friis que estima la perdida que puede existir en un enlace punto a punto. Por otro lado, el modelo de Hokumura Hata es un modelo más complejo y considera factores adicionales como la altura del transmisor y el tipo de terreno. Para áreas más extensas se puede emplear el modelo de Longley-Rice que se utiliza para predecir y analizar la cobertura de un sistema de comunicaciones por microondas.

Modelado de sistemas

El proceso de modelado de sistemas se caracteriza por analizar el comportamiento de un sistema de manera general, en el diseño de sistemas de comunicaciones microondas es empleado para simular y evaluar el rendimiento de cada uno de sus componentes. Las herramientas más utilizadas para el modelado de sistemas son Matlab y Simulink.

Matlab es un entorno de cálculo numérico y de lenguajes de programación, se basa en la creación de scripts, brinda un entorno dinámico para desarrollar modelos, procesamiento de datos. Este lenguaje permite implementar vectores y matrices de manera eficaz y en corto tiempo.

Por su parte, Simulink es un entorno que se complementa con Matlab para la simulación de sistemas dinámicos lineales y no lineales [30]. Esta herramienta permite representar sistemas mediante diagramas de bloque o etapas donde cada elemento cumple un trabajo específico, para el diseño de sistemas microondas existen procesos integrados dentro de los bloques específicamente diseñados para el modelado de elementos como transmisores y receptores. Adicionalmente es posible modelar elementos de manera individual como filtros, antenas, mezcladores, osciladores, amplificadores, entre otros.

Modelado y simulación electromagnética

Mediante el modelado electromagnético se puede analizar la distribución de campos electromagnéticos en un sistema microondas, mediante las simulaciones se puede comprender como interactúan las ondas en estructuras físicas como antenas, filtros, acopladores, entre otros.

Existen varias herramientas CAD populares disponibles comercialmente como: Microwave Studio (CST), ADS (Aligent Technologies), HFSS. Estos programas permiten generar estructuras a partir de componentes específicos y un diseño inicial incluyendo pérdidas y desacoples para posteriormente generar un prototipo. Este tipo de software se puede emplear para optimizar diseños, visualizar efectos materiales o con fines educativos ya que permite comprender cómo se comporta un componente de microondas de manera dinámica.

El modelado mediate simuladores electromagnéticos puede reducir el tiempo y costo de un diseño real, evitando la construcción y medición de prototipos de laboratorio en cada iteración como en el caso de circuitos integrados monolíticos de microondas ya que estos circuitos no se pueden ajustar después de su fabricación [28].

2.1.10 Aplicación del aprendizaje profundo en comunicaciones inalámbricas

El uso de técnicas de aprendizaje automático en el campo de las comunicaciones inalámbricas se presenta como una herramienta para comprender patrones de datos y ser capaz de procesar la no linealidad propia de componentes y sistemas inalámbricos, presentando un nuevo enfoque que se puede aplicar a solucionar problemas y optimizar procesos. En este ámbito, se presentan varias aplicaciones de relevancia.

Uso de Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son arquitecturas de redes neuronales que se caracterizan por ser efectivas en la extracción de características espaciales y por su gran capacidad de aprender patrones de datos han sido aplicadas al análisis de señales de comunicación, modulación y demodulación de señales, así como en la detección de errores [31].

Uso de aprendizaje profundo en la optimización de recursos de redes.

El aprendizaje profundo también puede ser enfocado a optimizar la asignación de recursos en redes inalámbricas, considerando la capacidad, la latencia y la energía. Al aplicar algoritmos de DL es posible maximizar la eficiencia espectral y a mejorar la calidad de servicio (QoS) en entornos inalámbricos densos [32].

Seguridad y detección de anomalías

En redes inalámbricas es importante innovar y usar herramientas que mejoren protocolos de seguridad, el aprendizaje profundo se ha empleado en la detección de infiltraciones, ataques dns y detección de anomalías en la red que puedan afectar a los dispositivos o a los usuarios finales, al aplicar de estas técnicas en la seguridad se puede contar con procesos y protocolos de seguridad robustos [33]. Como ejemplo de modelos aplicados en esta área se puede mencionar: los autoencoders y redes adversarias generativas (GAN).

Diseño de dispositivos microondas asistido por aprendizaje profundo

Los algoritmos de aprendizaje profundo se aplican al diseño integral de dispositivos microondas mediante la interpretación de patrones de datos. Al implementar algoritmos de aprendizaje como el antes mencionado es necesario adaptar de la topología de la red neuronal, la configuración de parámetros de entrenamiento y la gestión de recursos

computacionales, al considerar estos aspectos es posible optimizar el funcionamiento de uno o varios componentes. [34].

2.1.11 Entrenamiento de una red neuronal

El entrenamiento de una red neuronal es el proceso en el que se ingresa un número determinado de datos se los procesa realizando ajustes de pesos y sesgos. Este proceso se repite n veces ya que recorre de inicio a fin las capas de las neuronas y se lo denomina *fordwarpropagation* (propagación hacia adelante) y *backpropagation* (propagación hacia atrás).

El entrenamiento de una red neuronal se puede describir en 3 fases y se lo puede visualizar su proceso en la Figura 7:



Figura 7: Proceso de entrenamiento de la red neuronal [16].

- **Forwardpropagation:** en esta fase se obtienen predicciones a partir de datos de entrenamiento, los datos recorren la estructura de neuronas enviando la información unas a otras, capa por capa hasta llegar a la capa final con la que se puede realizar predicciones.
- Pérdida o error: en este punto se hace uso de las métricas de rendimiento y de pérdida para conocer el error entre el resultado de la predicción y el resultado esperado, el objetivo principal es que el error se haga cero o que no existan divergencias significativas. Para lograr esto, se debe ajustar los pesos de las interconexiones de manera automática hasta obtener valores cercanos a cero.
- **Backpropagation:** una vez conocido el error de la etapa anterior la información se comienza a propagar desde la capa final o de salida hacia todas las neuronas de la capa oculta o hacia atrás. Cada una de estas neuronas recibe la señal total de pérdidas en función de la contribución relativa que se haya agregado a la salida. En este punto ya se pueden ajustar los pesos y sesgos entre neuronas. Para obtener un error de cero o cercano se puede hacer uso de optimizadores como Adam, SGD, Adagrad, entre otros. La función de los optimizadores es agregar cambios en los sesgos y pesos de

manera automática para ajustar el modelo, este proceso se lo realiza en grupos de datos con varias iteraciones.

2.1.12 Proceso de aprendizaje

El proceso de entrenamiento de una red neuronal puede derivar en distintos escenarios, dependiendo de cómo se realizó el análisis y procesamiento de datos, entre los principales escenarios están:

- **No convergencia:** el modelo es incapaz de encontrar el resultado y continúa analizando sin fin.
- **Convergencia:** el modelo encuentra el resultado esperado y muestra las predicciones de los datos.

Como resultado de la convergencia se muestran 3 posibles resultados:

- **Subajuste:** El algoritmo no es capaz de generalizar ningún patrón debido a que el conjunto de datos es suficiente para generar el aprendizaje y esto puede ocasionar un error en el modelo.
- **Sobreajuste:** Se produce cuando la capacidad de procesamiento de la red es poco eficaz incluso con buenos datos de entrenamiento, esto provoca que los nuevos datos no generen una clasificación acertada
- **Buen entrenamiento:** la red genera buenas predicciones, el procesamiento funciona de manera eficaz para recibir nuevos datos.

En la Figura 8 se presentan los posibles escenarios analizados anteriormente.



Figura 8: Resultados de convergencia neuronal [16].

2.1.13 Métricas de rendimiento

En el campo del aprendizaje profundo y redes neuronales es importante contar con herramientas que permitan dar una medida real del rendimiento de las redes neuronales y como se adaptan a un patrón de datos establecido. En este ámbito, las métricas de rendimiento o métricas de error se emplean para comparar de un modelo entrenado con datos reales u observados de un grupo de datos de prueba [35].

En este contexto, existen varias métricas de rendimiento empleadas en aprendizaje automático y pronóstico de fallas, entre las más conocidas se encuentran:

Error cuadrático medio (MSE): Realiza un promedio cuadrático de los errores considerando el valor estimado con el valor real o esperado, es enfocado a detectar errores grandes en datos con alta dispersión.

Error absoluto medio (MAE): Es la diferencia media absoluto promedio del valor estimado con el valor real o esperado, a diferencia de (MSE) detecta pequeñas fluctuaciones (errores) en un conjunto de datos.

Error porcentual absoluto (MAPE): Realiza un promedio absoluto del error porcentual. es empleado a grandes cantidades de datos.

CAPÍTULO III

3.1 METODOLOGÍA

En este capítulo se describe la metodología con la cual se llevará a cabo el presente proyecto, en la Figura 9 se muestra cómo se encuentra estructurada.



Figura 9: Fases del proyecto.

Fase 1

Se realiza una revisión bibliográfica exhaustiva de la literatura existente sobre el aprendizaje profundo aplicado al diseño de sistemas de microondas y los diferentes modelos de aprendizaje profundo. Esta revisión permite identificar los enfoques más prometedores y las limitaciones actuales de esta área de investigación. Para ello se recopilaron estudios relevantes de bases de datos de literatura científica, revistas, conferencias especializadas y páginas web de instituciones académicas y de investigación.

Fase 2

En esta fase se establecen los componentes que conforman el sistema de microondas considerando la frecuencia, ancho de banda, etapas del sistema. Posteriormente se realiza el diseño base de cada uno de sus componentes y se ejecuta simulaciones en el software ADS (Advances Desing System) para verificar el correcto funcionamiento.

Fase 3

En esta fase se procede a generar los datos necesarios para entrenar la red neuronal, para esto se utiliza el software Matlab para ingresar las ecuaciones de diseño para cada componente y guardar los datos obtenidos considerando las variables de entrada y salida. Posteriormente se aplica la red neuronal adecuada para procesar los datos de cada componente en un entorno Python.

Fase 4

En esta fase se procede con el entrenamiento de la red neuronal para cada componente del sistema, es importante recalcar que cada red neuronal se adapta de acuerdo con el número de variables y al número de datos que se van a utilizar.

Una vez procesados los datos de entrenamiento se ingresa al software para visualizar los parámetros de salida y respuesta en frecuencia según corresponda para cada componente.

Fase 5.

En esta fase se realiza la validación de los resultados, en este caso se consideran los resultados de los entrenamientos realizados y se los compara con valores ideales o de diseño base. De esta manera se puede seleccionar el conjunto de resultados que más se acercan o que se encuentran el rango deseado.

Fase 6.

Mediante el análisis estadístico y métricas de rendimiento se puede evaluar si el uso de aprendizaje profundo es aplicable al diseño de sistemas de microondas.

3.1.1 Tipo de investigación

El tipo de investigación empleado para el proyecto es como primera instancia investigación aplicada. Este tipo de investigación se centra en la aplicación de conocimientos científicos y tecnológicos para resolver problemas prácticos, implementar nuevas tecnologías o procesos.

En este caso, el aprendizaje profundo se plantea como un posible método de diseño de sistemas microondas, por lo que el proyecto de investigación se centrará en evaluar la su eficacia.

Posteriormente, se aplica la investigación experimental para realizar pruebas en los dispositivos que forman parte del sistema mediante observación y simulaciones.

3.1.2 Población y muestra

Población

Para este proyecto se considera como población a todos los datos y parámetros de diseño obtenidos al momento de desarrollar el sistema base y el sistema implementado con la técnica Deep learning, así como los datos de entrenamiento de las redes neuronales, modelos de aprendizaje, función de activación, optimizador, número de capas, épocas.

Muestra

Para la muestra se extraen todos los resultados de las pruebas realizadas mediante la técnica propuesta y con el análisis de errores se clasifican los mejores resultados. Es decir, los que tengan un porcentaje de error bajo y se los compara con los resultados del sistema diseñado con el método tradicional, con el fin de evaluar si es factible utilizar este nuevo método.

3.1.3 Operación de variables

VARIABLE DEPENDIENTE	CONCEPTO	INDICADORES
Rendimiento del sistema transmisor-receptor.	Capacidad del sistema de microondas para cumplir con los requisitos de rendimiento establecidos.	Ganancia, filtrado, pérdidas de potencia, métricas de rendimiento, eficiencia.
VARIABLE INDEPENDIENTE	СОМСЕРТО	INDICADORES
Ancho de banda	Rango de frecuencias utilizado por los componentes del sistema microondas	MHz, GHz
Potencia de transmisión- recepción	Potencia de la señal transmitida y recibida por el sistema microondas	dBm
Arquitectura de aprendizaje profundo empleado	Técnica de inteligencia artificial que utiliza redes neuronales artificiales para aprender de datos	Tipo de arquitectura de red neuronal utilizada.
Parámetros de configuración de la red neuronal	Ajustes necesarios para que la red se ajuste a las condiciones del sistema	Función de activación, optimizador, numero de capas, épocas.
Datos de entrenamiento	Cantidad de datos que se emplean para entrenar una red neuronal	Número

Tabla 2: Variables dependientes e independientes

3.2 Diseño del sistema

Para el diseño de un sistema de comunicación es primero establecer el rango de frecuencias que deseamos transmitir y recibir en este caso se establece la banda media de 5Ghz, hay que recordar que al ancho de banda en la que opera 5G es de 100 MHz que se lo denomina banda base, la cual es la señal original generada desde el transmisor y puede contener: datos de audio, voz, video, paquetes de datos 5G, señales de control. El tipo de transmisión antes mencionado se basa en el estándar 5G – NR FR1 el cual trabaja en el rango de 400 MHz hasta 7 GHz por lo que es aplicable en la presente investigación.

Con las consideraciones anteriores se establecen los componentes que conformar el sistema, en este caso se consta de 4 etapas: generación de señal, amplificación, mezclado y filtrado las cuales se encuentran detalladas en el diagrama de bloques de la Figura 10.



Figura 10: Diagrama en bloques de los componentes del sistema transmisor-receptor.

Como se puede apreciar a la entrada del sistema se genera la señal en banda base, posteriormente ingresa al mezclador para combinar las señales y se acopla al amplificador para finalmente filtrarla. En el receptor se recupera la señal original cambiando la configuración del mezclador y del filtro.

Una vez establecidos los componentes y las características del sistema se comienzan con el diseño base.

3.2.1 Filtro de pasa banda de líneas acopladas

Para el diseño de este componente se tomó en cuenta las características especificadas en la Tabla 3.

Componente	Filtro microondas pasa
Componente	i into interoondas pasa
	banda con líneas acopladas
Tecnología:	Microstrip
Тіро	Chevychev
N° orden	5
Diredo	0.05 JD
KIZAUO	0.05 dB
Ancho de banda	0.03 %
porcentual	
Impedancia de entrada	50 Ohms
Eroquancia da disaño	2.5 CHz
r recuencia de diseno	3.3 UTZ

Tabla 3: Características del componente

En la tabla anterior se establece las condiciones de diseño del filtro, es importante destacar que se considera el filtro de líneas acopladas ya que es relativamente más sencillo de diseñar y se adapta bien a la frecuencia de banda media establecida, el filtro es de orden 5 es decir que estará conformado por 6 secciones, se utilizó las tablas de polinomios para filtro Chevychev mostradas en la Tabla 3 estas tablas pueden ser empleadas para trabajar con rizado de 0.5 y 3.0 dB. Para calcular el ancho de banda porcentual se emplea la ecuación 2, donde se considera las frecuencias laterales.

$$\Delta = \frac{f_2 * f_1}{f_0} \tag{2}$$

 $\Delta = \frac{3.55 \ GHz * 3.45 \ GHz}{3.5 \ GHz} * 100$

 $\Delta = 3 \%$

Tabla 4: Valores de polinomios para filtro Chevychev pasa bajo con rizado de 0.5 y 3.0 dB de rizado [28].

	Rizado 3.0 dB										
N	g1	g2	g3	g4	g5	g6	g 7	g8	g9	g10	g11
1	1.9953	1.000									
2	3.1013	0.5339	5.8095								
3	3.3487	0.7117	3.3487	1.000							
4	3.3487	0.7883	4.3471	0.5920	5.8095						
5	3.4817	0,7618	4.5381	0,7618	3.4817	1.000					
6	3.5045	0.7685	4.6061	0.7929	4.4641	0.6033	5.8095				
7	3.5182	0.7723	4.6386	0.8089	4.6386	0.7723	3.5182	1.0000			
8	3.5227	0.7745	4.6575	0.8089	4.6990	0.8018	4.4990	0.6073	5.8095		
9	3.5340	0.7760	4.6692	0.8118	4.7272	0.8118	4.6692	0.7760	3.5340	1.0000	
10	3.5384	0.7771	4.6768	0,8136	4.7425	0.8164	4.7260	0.8051	4.5142	0.6091	5.8095
					Riza	do 0.5 dB	5				
N	g1	g2	g3	g4	g5	g6	g 7	g8	g9	g10	g11
1	0.6986	1.000									
2	1.4029	0.7071	1.9841								
3	1.5963	1.0967	1.5963	1.0000							
4	1.6703	1.1926	2.3661	0.8419	1.9841						
5	1.7058	1.2296	2.5408	1.2296	1.7058	1.0000					

6	1.7254	1.2479	2.6064	1.3137	2.4758	0.8696	1.9841				
7	1.7372	1.2583	2.6381	1.3444	2.6381	1.2583	1.7372	1.0000			
8	1.7451	1.2647	2.6564	1.3590	2.6964	1.3389	2.5093	0.8796	1.9841		
9	1.7504	1.2690	2.6678	1.3673	2.7239	1.3673	2.6678	1.2690	1.7504	1.0000	
10	1.7543	1.2721	2.6754	1.3725	2.7392	1.3806	2.7231	1,3485	2.5239	0.8842	1.9841

Según el rizado se obtienen los valores g para el filtro de orden 6, con estos valores se obtiene Z_0J_n que se calcula con la ecuación 2 y la impedancia par e impar (Z_{0e} , Z_{0o}) que se derivan de la ecuación 3.

$$\boldsymbol{Z}_{\boldsymbol{0}}\boldsymbol{J}_{\boldsymbol{1}} = \sqrt{\frac{\pi\Delta}{2\,g_1}} \tag{2.1}$$

$$\boldsymbol{Z}_{0}\boldsymbol{J}_{n} = \frac{\pi\Delta}{2\sqrt{g_{n} \times 1g_{n}}} \quad para \ n = 2, 3, \dots N.$$
(2.2)

$$\boldsymbol{Z}_{0}\boldsymbol{J}_{N+1} = \sqrt{\frac{\pi\Delta}{2 g_{N} \times g_{N} + 1}}$$
(2.3)

$$\mathbf{Z}_{0e} = Z_o \left[1 + Z_0 J_n + (Z_0 J_n)^2 \right]$$
(3.1)

$$\mathbf{Z}_{0o} = Z_o \left[1 - Z_0 J_n + (Z_0 J_n)^2 \right]$$
(3.2)

Una vez realizados los cálculos se procede con el diseño de las líneas acopladas, para esto se debe considerar las dimensiones de cada sección. Donde, W es el ancho de la línea el cual está relacionado con la impedancia, L es la longitud de las líneas y está relacionado con la frecuencia y S es la separación entre cada sección.

Adicionalmente se debe seleccionar un sustrato ya que la tecnología que se está empleando es Microstrip, para el componente en cuestión se empleará el paquete PCB que incluye el software de simulación sus características vienen descritas en la Figura 11.
PCB Layout Tech Choose mate	nology Setup - Materials rials used in PCB substrate	
Dielectric Mater	ials:	
Core material:	FR_4_Core ~ Thickness: 1.5	millimeter ~
Conductor Mate	rials:	
Outer metal mate	rial: Copper V Thickness: 0.0125	millimeter ~

Figura 11: Selección del sustrato.

Caracterizar un dispositivo directamente por sus dimensiones físicas puede resultar complejo, por este motivo se puede caracterizar las dimensiones mediante la impedancia par e impar, cuyos valores fueron calculados con las ecuaciones anteriores y los valores para cada sección se encuentran en la Tabla 3.

Mediante herramientas de simulación es posible caracterizar dispositivos microondas, en este caso el software empleado fue ADS (Advaced Desing Sistem) ya que permite el diseño de componentes de alta frecuencia, su plataforma cuenta con paquetes de: RF, microondas, simulación de circuitos, simulación electromagnética, entre otros. Su plataforma es bastante intuitiva y permite interactuar con los dispositivos e integrar varios componentes a la vez.

En este punto se cuenta con todos los parámetros necesarios para simular el filtro de líneas acopladas por lo que se ingresan todos los datos al software. Primero se selecciona el bloque del sustrato, luego se ingresa el valor de las líneas de transmisión para cada una de las secciones del filtro y se agregan las terminales. Finalmente se agrega el bloque que simulación de Parámetros S y se establece el rango de frecuencia que se va a analizar.

Mediante las herramientas de gráficas que incluye el programa es posible visualizar los parámetros de Scattering (Parámetros S) para visualizar el acoplamiento, la transmisión y la reflexión de la señal con respecto a sus puertos.

3.2.2 Amplificador microondas

En un sistema de microondas, el amplificador es un dispositivo muy importante ya que permite incrementar la potencia de las señales, esto es fundamental en un sistema donde existen componentes como el filtro microondas el cual elimina señales no deseadas, pero a su vez reduce la señal original, el amplificador ayuda a mitigar este problema, dependiendo de su ubicación también aumenta señales que desean ser transmitidas.

En el mercado existen componentes diseñados para trabajar como amplificadores, a distintas frecuencias. En este caso se optó por el transistor bipolar BFP 420 el cual es un dispositivo de bajo ruido y alta ganancia, su frecuencia de trabajo es de hasta 25GHz. En la Figura 12 se muestra una representación del encapsulado obtenido de la hoja de datos del fabricante.



Figura 12: Transistor BFP-420 [36].

Una vez conocidos los parámetros del amplificador, se aplican las ecuaciones de diseño. Una parte fundamental es calcular la estabilidad del transistor que se la puede calcular con los parámetros S proporcionados por el fabricante mostrados en la Tabla 5 representados en su forma fasorial y polar. De esta manera se puede conocer que tal confiable y predecible es el dispositivo y si pudiese existir ruido o interferencia a su salida.

Frecuencia	3.5 GHz
611	-0.4365 0.3522j
511	0.5617 ∠ 140°
\$12	0.894 0.0465j
512	0.895 ∠ 2.98°
821	2.4651 2.2431j
	3.328 ∠ 42.7°
S22	-0.1635 -0.0487j
	0.1706 ∠ 163.2°
Zona de activación	1.5 VCE 9.0 mA

Tabla 5: Datos del fabricante transistor BFP420.

El criterio que se considera para calcular la estabilidad es Δ (Delta) que es la distancia del punto de trabajo o de operación del transistor, otro criterio importante es el factor de Rollet (K) que es una medida de la estabilidad. Se conoce que si $\Delta < 0$ y K > 1 el amplificador será incondicionalmente estable lo que significa que mantendrá una operación adecuada y no generará ruido o interferencias en altas frecuencias. las expresiones para calcular estos parámetros se muestran en las ecuaciones 4 y 5.

$$\Delta = |S_{11} * S_{22} - S_{12} * S_{21}| \tag{4}$$

$$\Delta = \begin{cases} < 1, & \text{estable} \\ > 0, & \text{inestable} \end{cases}$$
$$K = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |\Delta|^2}{2 |S_{12} * S_{21}|}$$
(5)

$$\Delta = \begin{cases} < 1, & \text{inestable} \\ > 1, & \text{estable} \end{cases}$$

A partir de estos valores se calcula los parámetros B y C (ecuaciones 6-9) que están relacionados con la magnitud y fase de las reflexiones en la entrada y salida del transistor

$$B1 = 1 + |S_{11}|^2 - |S_{22}|^2 - |\Delta|^2$$
(6)

$$B2 = 1 + |S_{22}|^2 - |S_{11}|^2 - |\Delta|^2$$
(7)

$$C1 = S_{11} - \Delta(S_{22})^* \tag{8}$$

$$C2 = S_{22} - \Delta(S_{11})^* \tag{9}$$

Se conoce que debido a las desadaptaciones de impedancia se pueden generar ondas reflejadas en la fuente y en la carga de la red del transistor lo cual se denomina coeficientes de reflexión, en la Ecuación 10 se muestra la expresión para calcular el coeficiente de reflexión de la fuente (Γ_S) y en la ecuación 11 para la carga (Γ_L).

$$\Gamma_{S} = B1 \pm \frac{\sqrt{B1^{2} - 4|C1|^{2}}}{2 C1}$$
(10)

$$\Gamma_L = B2 \pm \frac{\sqrt{B2^2 - 4 |C2|^2}}{2 C2}$$
(11)

Para garantizar que un transistor funcione adecuadamente este debe alcanzar su ganancia máxima, esto lo hace apto para poder integrarlo a un sistema de microondas, conocido el valor de K y los parámetros S del transistor, se puede calcular la ganancia máxima del transistor mediante la Ecuación 12.

$$GT_{MAX} = \frac{|S_{21}|}{|S_{12}|} \left(K - \sqrt{K^2 - 1}\right)$$
(12)

Como se mencionó anteriormente en el transistor existen desadaptaciones de impedancia debido a la configuración del transistor por lo que es necesario crear secciones de acoplo mediante líneas de transmisión y stubs. Esto se lo puede realizar con la ayuda de la carta de Smith. Sin embargo, este proceso puede complicar la automatización de los cálculos que se lo realizará posteriormente por lo que se recurre a soluciones analíticas.

Para el diseño de la red de acoplo se utilizó el modelo de stub en paralelo que se muestra en la Figura 13. Donde Y_L es la carga a la cual se va a adaptar que en este caso es el transistor, la red cuenta con una línea de transmisión de longitud d y un sub de longitud l y Y corresponde a las impedancias en cada punto de la red.

< **A** \



Figura 13: Red de acoplo stub simple en paralelo [28].

Antes de calcular las dimensiones de la línea de transmisión y del stub, se debe calcular el valor de t con los valores del coeficiente de reflexión de fuente y de carga de acuerdo con la Ecuación 13.

$$t = XL \pm \frac{\sqrt{RL[(Z_0 - RL)^2 + XL^2]/Z_0}}{RL - Z_0}$$
(13)

Con los valores t y de acuerdo con la condición de la Ecuación 14 se puede calcular las dimensiones de la línea de transmisión tanto para fuente como para carga.

$$\frac{d}{\lambda} = \begin{cases} \frac{1}{2\pi} (\tan^{-1} t), & para \ge 0\\ \frac{1}{2\pi} (\pi + \tan^{-1} t), & para < 0 \end{cases}$$
(14)

Para la relación de admitancias de los stubs se emplea la Ecuación 15

$$B = \frac{RL^2 t - (Z_0 - XL t) (XL + Z_0 t)}{Z_0 [RL^2 + (XL + Z_0 t)^2]}$$
(15)

Finalmente, mediante la Ecuación 16 se calcula la dimensión de los stubs de fuente y carga.

$$\frac{\ell_1}{\lambda} = \frac{-1}{2\pi} \tan^{-1} \frac{B}{Y_0}$$
(16)

Luego de aplicar las ecuaciones anteriores se obtienen los parámetros del amplificador. Posteriormente, se ingresan los valores calculados al software para poder caracterizar el componente, en este caso podemos colocar directamente el transistor empleado ya que previamente se instaló las librerías correspondientes, de esta manera se puede elegir entre varias opciones de transistores comerciales.

Adicionalmente, se integra herramientas de simulación para poder visualizar la máxima ganancia del amplificador y de esta manera verificar el comportamiento del amplificador.

3.2.3 Mezclador RF

Pozar define al mezclador como un dispositivo con tres puertos utilizado para realizar conversiones de frecuencia tanto de manera ascendente (up conversion) como descendente (down conversion) entre señales de banda base y frecuencias portadoras RF. En la Figura 14 se muestran las 2 configuraciones básicas del mezclador.



Figura 14: Configuración de un mezclador: (a) Up conversión, (b) Down conversión.

Como se puede apreciar en la figura anterior, a la salida del mezclador se produce la suma y diferencia de las dos señales de entrada. Este comportamiento es de gran utilidad para la implementación de un sistema transmisor – receptor ya que se puede emplear la configuración de subida (up- conversion) para el transmisor y la de bajada (down-conversion) para el receptor.

Para el correcto funcionamiento de un mezclador se debe emplear componentes no lineales como son transistores y diodos ya que este tipo de elementos se encargan de crear ecos de la señal o también llamados armónicos. Sin embargo, también genera varios productos de la señal de entrada que pueden interferir en la señal que se desea, por lo que se debe emplear filtros para seleccionar los componentes de señal requeridos y despreciar los otros componentes.

En la práctica, se puede diseñar un mezclador con un acoplador direccional, un conjunto de diodos y un filtro. La estructura del mezclador antes mencionado se muestra en la Figura 15.



Figura 15: Componentes del mezclador.

El acoplador direccional es una red de 4 puertos adaptados que combina dos señales de entrada y en sus puertos de salida tendrá un desfase de 90° o 180°. Por otro lado, el conjunto de diodos empleados son del tipo Shottky los cuales se encargan de generar los armónicos de la señal, para el diseño y simulación se debe considerar diodos comerciales que se adapten a los requerimientos del sistema en este caso el diodo utilizado será el BAS140W que trabaja

con frecuencias de hasta 24 GHz. Finalmente, el filtro que se debe emplear es uno del tipo Chebychev pasa bajo con una frecuencia de corte de 300 MHz con el fin de que distinga solamente las señales generadas en banda base, esta configuración sirve como base tanto para el transmisor, como para el receptor.

3.2.4 Generación de datos

Es bien conocido que las redes neuronales necesitan una gran cantidad de datos para poder ser entrenadas, para el caso de los dispositivos microondas es necesario generar valores a partir de sus ecuaciones de diseño. En la práctica este proceso puede resultar bastante tedioso por lo que se requiere la asistencia de un software de procesamiento y análisis numérico como lo es MATLAB.

En el software antes mencionado se ingresan todas las ecuaciones y variables necesarias para el diseño de los dispositivos a través de un script, el programa realiza el cálculo numérico y devuelve como primera instancia los valores ideales para el diseño que son los mismos que se obtendrían al realizar el proceso de cálculo de manera manual. Para generar una cantidad significativa de datos es necesario establecer funciones que las contengan variables de diseño en un rango determinado, con esto se logra realizar varios análisis en función de las variables obteniendo así diferentes resultados.

Al final, el programa nos proporciona una matriz con todas las posibles combinaciones, dependiendo de la capacidad de la red neuronal se pueden generar más o menos datos.

En el Anexo 1 se muestra el código en Matlab de los componentes del sistema, para cada uno existe un script el cual se encarga de realizar los cálculos tal y como se lo explicó anteriormente.

Finalizando esta sección se establecen las variables de entrada y de salida para cada componente, en estas variables se almacenan los datos generados anteriormente es importante conocer el número de variables para estructurar las capas de la red neuronal.

Variables de entrada	Rango	Variables d	e salida
Ancho de banda	0.1 % - 10%	Impedancia par, impar	Impedancia par,
		sección 1	impar sección 4
Rizado	0.5 dB, 3.0 dB	Impedancia par, impar	Impedancia par,
		sección 2	impar sección 5
Impedancia	20 Ohms – 50 Ohms	Impedancia par, impar	Impedancia par,
		sección 3	impar sección 6

Tabla 6: Variables entrada y salida filtro de líneas acopladas.

	-		
Variables de entrada	Rango	Variables d	e salida
Frecuencia	0.1 GHz – 10 GHz	Acoplador	Fuente
		Dimensión línea de	dimensión stub
		transmisión	
		Acoplador	Carga
		Dimensión línea de	Dimensión stub

transmisión

Tabla 7: Variables entrada y salida amplificador microondas.

Variables de entrada	Rango	Variables d	e salida
Frecuencia de corte	1 GHz – 5GHz	Impedancia de las	Línea 1, 2, 3, 4
		líneas del acoplador	
Impedancia	20 Ohms – 100 Ohms	Componentes filtro	C1, L2, C3, L4,
		pasa bajo	C5

Tabla 8: Variables entrada y salida mezclador microondas.

3.2.5 Aplicación de la red neuronal

Se aplicará la arquitectura de red neuronal convolucional adaptada al sistema transmisorreceptor para la banda de frecuencia establecida en la red 5G. De acuerdo con la búsqueda bibliográfica de aplicaciones similares se evidenció que el modelo convolucional es el más empleado en dispositivos microondas ya que son capaces de aprender las relaciones entre las características de entrada y salida de componentes microondas.

El lenguaje de programación empleado para trabajar con redes neuronales fue Python ya que es muy versátil para crear aplicaciones con Deep Learning. Al trabajar con Python es necesario ejecutarlo en un entorno de desarrollo integrado (IDE) en donde se puedan crear programas. En este caso el IDE empleado fue PyCharm en su versión comunitaria ya que su entorno es bastante intuitivo y su interfaz es amigable en comparación con otros programas.

Es bien conocido que Python cuenta con varias versiones que a lo largo del tiempo se van actualizando, ofreciendo al usuario nuevas funcionalidades y corrigiendo posibles errores en versiones anteriores. Esto resulta contraproducente en ciertos casos ya que al trabajar con nuevas versiones estas pueden ser o no compatibles con librerías, comandos, extensiones, etc.

Para solucionar este inconveniente se utilizó el software Anaconda el cual permite crear entornos virtuales, dentro del programa se aloja gran de lenguajes de programación con las versiones más estables. De esta manera se creó un entorno con el lenguaje Python en su versión 3.10.14.

Una vez creado el entorno de programación se procede a crear la red neuronal, aquí se establecen varios parámetros como las librerías (Figura 16) que se va a emplear que en este caso será el paquete tensorflow.keras junto con el tipo de conexión y el framework y finalmente se importa el algoritmo para evitar el sobre entrenamiento.

```
import numpy as np #calculos matematicos
from scipy.io import loadmat #leer variables de matlab
from tensorflow.keras.models import Sequential #importar el framework
from tensorflow.keras.layers import Dense #tipo de conexion
from tensorflow.keras.callbacks import EarlyStopping #paquete para evitar el sobreentrenamiento
```

Figura 16: Librerías y paquetes necesarios para la red neuronal.

CAPÍTULO IV

4.1 RESULTADOS Y DISCUSIÓN

En esta sección se presenta el diseño base de los componentes del sistema, posteriormente el resultado de los entrenamientos de cada red neuronal y finalmente su integración al transmisor y receptor.

4.1.1 Valores filtro de líneas acopladas base

En la Tabla 9 se encuentran los valores prototipo para el filtro calculado mediante las ecuaciones de diseño, donde se dispone el número de secciones, los valores g correspondientes según el rizado.

Secciones	$\mathbf{g}_{\mathbf{n}}$	Z_0J_n	Z _{0e}	Z ₀₀
g1	1.7058	0.1662	59.6918	43.0708
g2	1.2296	0.0325	51.6799	48.4260
g3	2.5408	0.0266	51.3686	48.7025
g4	1.2296	0.0266	51.3686	48.7025
g5	1.7058	0.0325	51.6799	48.4260
g6	1	0.1662	59.6918	43.0708

Tabla 9: Secciones del filtro con sus valores

En la Figura 17. se muestra el esquema del filtro de líneas acopladas donde se ingresaron todos los parámetros de diseño calculados.



Figura 17: Filtro de líneas acopladas ADS.

Luego de establecer el esquema del filtro se analiza los parámetros de Scattering (Parámetros S) para visualizar el acoplamiento, la transmisión y la reflexión de la señal con respecto a sus puertos. En la Figura 18 se puede visualizar las curvas de los parámetros (S11, S21, S12, S22) los cuales indican que el filtro tiene una pérdida de inserción de -0.88 dB (S21), es decir que el filtro permite pasar eficientemente la señal a 3.5GHz y tiene un alto rechazo fuera de banda lo que implica que funciona correctamente como pasa banda



Figura 18: Parámetros S filtro de líneas acopladas ADS.

4.1.2 Valores amplificador microondas base

Mediante las ecuaciones de diseño se obtienen los valores del amplificador base, mostrados en la Tabla 10.

Valor Δ (Delta)	0.373
Valor K	1.1621
B1	1.1612
B2	0.5902
C1	-0.4581+0.2960i
C2	-0.0517-0.21181i
Ganancia máxima (dB)	12.76 dB
Coeficiente de reflexión de la fuente	-0.5878 -0.3799i
Coeficiente de reflexión de la carga	-0.1047+0.4288i
Valor t fuente	0.4883
Valor t carga	0.4746
Valor d fuente	5.65
Valor d carga	69.60
Valor B fuente	0.0204
Valor B carga	0.0975

Tabla 10: Parámetros calculados del amplificador.

Valor l fuente	60.02
Valor l carga	45.80

En base a lo realizado con componente anterior, se ingresan los valores calculados al software para poder caracterizar el componente, el esquema se muestra en la Figura 19.



Figura 19: Amplificado de microondas ADS.

En la Figura 20 se muestra la curva de la máxima ganancia disponible (m1) y la máxima ganancia analizada (m2). Se puede evidenciar mediante sus curvas que la máxima ganancia a la frecuencia de 3.5GHz es muy cercana, por lo que se comprueba que el amplificador funciona adecuadamente.



Figura 20: Máxima ganancia disponible vs máxima ganancia obtenida.

4.1.3 Valores mezclador base

En la Figura 15 se mostró los componentes necesarios para el mezclador, en esta parte se realiza el cálculo de estos componentes sus valores se muestran en la Tabla 11.

Lineas del acoplador	Valor
hibrido de 180°	(Calculado)
Linea 1	70.71°
Linea 2	70.71°
Linea 3	70.71°
Linea 4	270.00°
Componentes del filtro	Valor
pasabajo	(Calculado)
C1	6.557pF
L2	42.918nH
C3	21.220pF
L4	42.9180nH
C5	6 667pF

Tabla 11: Parámetros calculados del mezclador.

Para estructurar el mezclador en el software es necesario incorporar sus componentes mediante un encapsulado (MIXER_ENC) al interior de este bloque de diseño se encuentra el acoplador hibrido de 180° y el conjunto de diodos Schottky (BAS140W). Adicionalmente se agregan puertos tipo tono para seleccionar las frecuencias de entrada y sondas de prueba para poder visualizar la señal en cada punto del circuito. En la Figura 21 se puede visualizar el esquema completo.



Figura 21: Mezclador ADS.

Una vez diseñado el mezclador se procede con la simulación. En este caso se realizará un análisis de armónicos a la salida del mezclador y no de parámetros S ya que se ingresa una señal de entrada, con la ayuda del analizador de espectros se puede visualizar el comportamiento de los armónicos, en la Figura 22 se puede visualizar suma y diferencias de

frecuencias según la frecuencia (RF) y de oscilador local (LO), se evidencia que el mezclador realiza la conversión de valores y se encuentra listo para adaptarse al sistema.

MEZCLADOR



Figura 22: Análisis de armónicos up-conversion y down conversion.

4.1.4 Integración transmisor base

A continuación, se presenta la integración de los componentes diseñados anteriormente, para el caso del transmisor se integró el mezclador en la entrada para combinar la señal de entrada (100 Mhz) y la del oscilador local (3.5 Ghz) la cual es la frecuencia de transmisión, posteriormente el amplificador eleva las señales y se acopla al filtro pasa banda para depurar las señales no deseadas para finalmente obtener la señal de transmisión, el esquema se muestra en la Figura 23.



Figura 23: Transmisor base ADS.

4.1.5 Integración receptor base

Aunque el receptor emplea los mismos componentes fue necesario cambiar la configuración. Como se puede apreciar en la Figura 24, en la entrada del esquema se integra una etapa de amplificación y luego el mezclador para que se combine la señal amplificada con la del oscilador que en este caso será una frecuencia alta (3.6 Ghz) con el fin de que a la salida se recupere la señal generada en el transmisor (100 MHz). El filtro pasa bajo ubicado a la salida del sistema recupera únicamente la señal deseada y atenúa las señales que no se encuentren en ese rango.



Figura 24: Receptor base ADS.

Una vez realizado el esquema se procedió con la simulación del transmisor y receptor con el fin de visualizar las señales de entrada y salida del sistema. El software permite realizar varios análisis gráficos mediante herramientas de simulación, las que se emplearon en este caso fueron el balance de armónicos lo cual nos permite visualizar los espectros de señales generadas en puntos específicos. En la Figura 25 se muestran las señales del transmisor.



Figura 25: Señales de entrada y salida del transmisor en tiempo y frecuencia.

Paralelamente, se muestran los resultados de las señales del receptor, mostradas en la Figura 26 donde se puede apreciar que se recuperan la señal generada por el transmisor, debido al efecto de los componentes internos del sistema analizados anteriormente.



Figura 26: Señales de entrada y salida del receptor en tiempo y frecuencia.

Para tener una visualización completa de las señales en cada punto del sistema se puede colocar analizadores de armónicos en las entradas y salidas de los componentes tanto del transmisor como del receptor. En la Figura 27 se muestra el comportamiento de la señal a lo largo de todo el sistema.



Figura 27: Señales de entrada y salida de los componentes del sistema.

Para poder visualizar si la potencia del receptor no se ve afectada por los componentes internos del circuito, en la Figura 28 se muestra la potencia de entrada y la potencia de salida del receptor.



Figura 28: Potencia de entrada y salida del receptor.

Como se puede apreciar, a la salida del sistema se obtiene un valor de -32 dBm lo cual indica que no existe pérdidas de potencia en el circuito y a la salida se obtiene una ganancia de 1.116 dB.

4.2 Entrenamiento y evaluación.

A partir de los parámetros de la sección anterior, en este apartado se presentan los resultados de los entrenamientos realizados para cada componente, en cada entrenamiento se realizó un ajuste de la red neuronal para que se adapte al componente y a la naturaleza de los datos compilados y se agrega la evolución del error mse y mae después de los entrenamientos y adicionalmente se realiza la simulación en ADS con los valores obtenidos. En el Anexo 3 se muestran las características de las redes neuronales implementadas para cada componente, así como el número de datos utilizados.

4.2.1 Red neuronal 1

Filtro de líneas acopladas

En la Tabla 12 se presenta el resultado de los entrenamientos del componente en cuestión, en este caso es importante destacar que esta fue la red neuronal que más capas empleó ya que los datos de ingreso presentaron una dispersión alta. Por lo tanto, es necesario que la red genere más conexiones entre las capas de entrada y salida para poder aprender el patrón de datos.

Tipo de red neuronal	Red neuronal
	convolucional
N° total datos de	8000
entrenamiento	
N° total de capas	271
Tipo de conexión	Densa
Modelo	Secuencial
Función de activación	Relu, linear
Optimizador	Adam
N° total epocas	200

|--|

En la Figura 29 se muestra la evolución del error cuadrático medio (MSE) y el error absoluto medio (MAE) que son métricas para evaluar el rendimiento del aprendizaje automático, mediante este estos valores es posible conocer el nivel de error de la red durante los entrenamientos.



Figura 29: Comparación error mse y mae, red neuronal 1.

Una vez procesados los datos de entrenamiento se ingresa al software para visualizar los parámetros de Scattering, en este caso se considera el parámetro S21 como referencia para visualizar el rizado y la transmisión con respecto a sus puertos. En la Figura 30, se muestra gráficamente el resultado de las simulaciones.



Figura 30: Parámetros S filtro de líneas acopladas, 5 entrenamientos realizados.

Como se puede apreciar, a medida que se realizaron los entrenamientos las curvas se van ajustando a la respuesta esperada (S21).

4.2.2 Red neuronal 2

Amplificador microondas

Para la red neuronal del amplificador es necesario considerar que se está trabajando con un dispositivo físico que tiene valores específicos (parámetros S) en función de la frecuencia de trabajo y su zona de activación. Por tanto, la red analizará un número limitado de datos. Con esta limitante, se realizó un sobre ajuste en la red aumentando significativamente el número de épocas, esto implica que la red procesará varias veces hasta aprender el patrón de datos. Como se puede ver en los resultados de la Tabla 13, para este caso en específico el sobre ajuste obtenido proporcionó una respuesta adecuada.

Sin embargo, esta configuración no es útil cuando se desea analizar datos nuevos, cabe mencionar que al emplear esta configuración el tiempo de análisis es más lento.

Tipo de red neuronal	Red neuronal
•	convolucional
N° total datos de entrenamiento	232
N° total de capas	34
Tipo de conexión	Densa
Modelo	Secuencial
Función de activación	Relu, linear
Optimizador	Adam
N° total epocas	5000

Tabla 13: Parámetros red neuronal 2

Con la información extraída se grafica el nivel de error obtenido a lo largo de los 5 entrenamientos las gráficas de errores para el componente en cuestión se muestran en la Figura 31. Como se puede observar, el error es considerablemente alto en los primeros entrenamientos. Sin embargo, se reduce en gran medida en el último entrenamiento.



Figura 31: Comparación error mse y mae, red neuronal 2.

En la Figura 32 se muestran los resultados de las simulaciones realizadas, como se demostró en el análisis de errores, en los primeros entrenamientos las curvas no logran una convergencia adecuada, pero a medida que los datos se ajustan se acercan más al resultado esperado.



Figura 32: Ganancia máxima amplificador, 5 entrenamientos realizados.

4.2.3 Red neuronal 3

Acoplador direccional

Para el primer componente del mezclador se implementó una red neuronal con un numero de datos bastante razonable en comparación a las otras redes neuronales analizadas anteriormente. Esto se debe en gran medida al patrón de datos de este componente ya que las ecuaciones de diseño son de baja complejidad. En consecuencia, el procesamiento es rápido. En la Tabla 14 se presentan los parámetros de la red neuronal

Tipo de red neuronal	Red neuronal
	convolucional
N° total datos de entrenamiento	4000
N° total de capas	11
Tipo de conexión	Densa
Modelo	Secuencial
Función de activación	Relu, linear
Optimizador	Adam
N° total epocas	500

Tabla 14: Parámetros red neuronal 3

Como ya se lo realizó anteriormente, en la Figura 33 se muestra la evolución de los errores correspondientes a los datos del acoplador. Se observa que desde el tercer entrenamiento el nivel de error es bajo.



Figura 33: Comparación error mse y mae, red neuronal 3.



En la Figura 34 se muestra el resultado de los parámetros S correspondientes al acoplador. Aquí se puede mencionar que el ajuste de datos es similar a los componentes anteriores.

Figura 34: Parámetros S11 del acoplador direccional, 5 entrenamientos realizados.

4.2.4 Red neuronal 4

Filtro pasa bajo

El segundo componente del mezclador presentó una alta dispersión en los datos. En consecuencia, fue necesario entrenar un gran número de datos para poder reducir el error, aunque no fue necesario realizar un sobre ajuste el número de épocas es relativamente alto lo cual es de esperar tomando en cuenta el número de datos. En la Tabla 15 se presentan los parámetros para esta última red.

Tipo de red neuronal	Red neuronal
	convolucional
N° total datos de	12000
entrenamiento	
N° total de capas	22
Tipo de conexión	Densa
Modelo	Secuencial
Función de activación	Relu, linear
Optimizador	Adam
N° total epocas	1000
Error mse/mae	Por definir

Tabla 15: Parámetros red neuronal 4



En la Figura 35, se muestra la gráfica del error para la última red neuronal implementada.

Figura 35: Comparación error mse y mae, red neuronal 4.

Al momento de simular los resultados de los entrenamientos se puede visualizar que el ajuste es casi inmediato en comparación a los componentes anteriores. En la Figura 36 se aprecia el comportamiento de las curvas.



Figura 36: Parámetros S21 del filtro pasa bajo, 5 entrenamientos realizados.

4.2.5 Análisis de errores.

A partir de los entrenamientos realizados se realiza un análisis general de las métricas de error implementadas. La Figura 37 muestra en porcentajes la reducción de error MSE y MAE para cada una de las redes neuronales, mediante esta gráfica se tiene una perspectiva más clara del rendimiento del método propuesto y conocer hasta qué punto las redes han logrado ajustarse al patrón de datos.



Figura 37: Porcentaje de reducción de error mse y mae

Como se puede visualizar en la Figura 38, las métricas indican que las redes reducen el error mse y mae hasta en un 90%, lo que implica que los resultados no presentarán mayor diferencia a los que se consideraron como base.



Figura 38: Promedio reducción de error mse y mae

Luego de procesar todos los entrenamientos se realizó un análisis de los entrenamientos que presentaron un margen de error bajo (cercano a cero) es decir que se acercan al resultado esperado y se los emplea para integrar el sistema completo con datos obtenidos con el método Deep Learning.

Luego de la respectiva simulación de componentes se presentan los resultados del transmisor y receptor donde los parámetros de los componentes fueron obtenidos mediante Deep Learning. Posteriormente, se muestran las señales de entrada y salida para el transmisor (Figura 39).



TRANSMISOR DEEP LEARNING

Figura 39: Señales de entrada y salida del transmisor utilizando Deep Learning.

En el caso del receptor (Figura 40) las señales de entrada y salida se encuentran en los rangos de frecuencia establecidos, verificando su funcionamiento.



Figura 40: Señales de entrada y salida del receptor utilizando Deep Learning.

Posteriormente, se realiza una comparación entre las señales del sistema base y las señales obtenidas mediante Deep Learning. Para el caso del transmisor (Figura 41) se puede apreciar que la señal de entrada es la misma para ambos casos, mientras que la señal de salida presenta

una pequeña diferencia en la potencia. Sin embargo, ambas señales se encuentran en la frecuencia de transmisión requerida.



Figura 41: Comparativa de señales del transmisor.

En la parte del receptor (Figura 42) se puede visualizar que la señal de entrada como de salida es prácticamente igual para los 2 análisis por lo tanto se puede deducir que para el caso del receptor las redes neuronales aprendieron el comportamiento de los componentes a tal punto de presentar una respuesta igual a la del diseño base demostrando así que puede ser aplicable al diseño de sistemas microondas.



Figura 42: Comparativa de señales del receptor.

Como análisis adicional, en la Figura 43 se presenta una comparativa de potencia obtenida entre los 2 métodos



Figura 43: Comparativa de potencia de entrada y potencia de salida.

En base a las señales obtenidas mediante la simulación se puede hacer una relación de la potencia y ganancia entre el diseño base y el diseño con Deep Learning, estos valores se muestran en la Tabla 16.

Potencia diseño base	
Entrada	-33.753 dBm
Salida	-32.637 dBm
Ganancia	1.116 dB
Potencia diseño Deep Learning	
Entrada	-33.652 dBm
Salida	-33.912 dBm
Ganancia	0.738 dB

Tabla 16: Valores de potencia y ganancia diseño base vs Deep Learning

Como se puede apreciar, en ambos diseños existe una ganancia a la salida lo cual indica que la potencia no se ve afectada por los componentes del sistema y con esto se verifica el funcionamiento del sistema para ambos casos.

CAPÍTULO V.

5.1 CONCLUSIONES

Mediante el estudio bibliográfico se evidenció que existen modelos de aprendizaje profundo aplicados al diseño y optimización de dispositivos microondas. Sin embargo, la aplicación solo se limita a un dispositivo en específico. Por lo que, en contraste con investigaciones relacionadas, el presente trabajo integra varios dispositivos a un sistema de comunicaciones.

Se aplicó una arquitectura de red neuronal profunda basada en el modelo convolucional y el modelo secuencial. Esta combinación de modelos demostró ser eficaz al momento de comprender las relaciones de entrada y salida del sistema.

Para que las redes neuronales puedan aprender patrones de datos se diseñó un sistema base con 4 componentes y 4 redes neuronales. Se ejecutaron entrenamientos para cada red neuronal teniendo un total de 24.232 datos y 338 capas distribuidas entre capas de entrada, capas de salida y capas ocultas.

En base a los resultados de aprendizaje de las 4 redes neuronales se realizó un análisis del porcentaje de reducción de error considerando las métricas MAE y MSE, registrando más del 90% de reducción como resultado de los entrenamientos. De esta manera se valida que las redes neuronales se ajustan al patrón de datos.

Mediante la evaluación del sistema se concluye que la respuesta en frecuencia es de 3.5 GHz de transmisión y una señal recuperada de 100 MHz y que el sistema es eficiente ya que a la salida del receptor se obtiene una potencia de -32.912 dBm y una ganancia de 0.738 dB con respecto a su señal de entrada y contrastando con los valores del diseño la diferencia en ganancia es de 0.37 dB. Sin embargo, no existen pérdidas significativas en los componentes del transmisor y receptor. Por lo tanto, se puede transmitir en entornos 5G.

5.2 RECOMENDACIONES

- Ampliar la investigación del uso de la inteligencia artificial y en concreto el Deep Learning y su aplicabilidad en el campo de las telecomunicaciones.
- Se recomienda aplicar método de diseño presentado en este trabajo a más dispositivos microondas como: resonadores, atenuadores, multiplicadores y experimentar su integración a sistemas de comunicaciones más complejos.
- Para futuras investigaciones se puede aplicar máquinas de soporte vectorial (SVM) como modelo de aprendizaje automático enfocado a la predicción de errores en dispositivos microondas ya que esta arquitectura tiene mayor rango de clasificación de patrones lineales y no lineales.
- Se puede agregar métricas de error para tener una evaluación del aprendizaje de la red aún más exhaustiva y detectar posibles anomalías.
- En futuros trabajos se puede mejorar la estructura de las redes neuronales para optimizar el aprendizaje y conseguir un tiempo de ejecución más corto.

BIBLIOGRAFÍA

- Sagar, Md Saimul, Hassan and Omi, Asif I, Wisniewska, Anna y Jalajamony, «Application of machine learning in electromagnetics: Mini-review,» *Electronics*, vol. 10, p. 2752, 2021.
- Heaton y Jeff, «Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN,» *Genetic programming and evolvable machines*, vol. 19, pp. 305--307, 2018.
- [3] C. Zhang, J. Jin, W. Na, Q. J. Zhang y M. Yu, «Multivalued neural network inverse modeling and applications to microwave filters,» *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, nº 8, pp. 3781--3797, 2018.
- [4] W. Liu, W. Na, L. Zhu, J. Ma y Q. J. Zhang, «A Wiener-type dynamic neural network approach to the modeling of nonlinear microwave devices,» *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, nº 6, pp. 2043--2062, 2017.
- [5] W. Zhang, F. Feng, W. Lui, S. Yan, J. Zhang, J. Jin y Q. L. Zhang, «Advanced parallel space-mapping-based multiphysics optimization for high-power microwave filters,» *IEEE transactions on microwave theory and techniques*, vol. 69, nº 5, pp. 2470--2484, 2021.
- [6] E. Mattucci, L. Feng, P. Benner, D. Romano y G. Antonini, «Mattucci, E., Feng, L., Benner, P., Rom Fast Frequency-Domain Analysis for Parametric Electromagnetic Models Using Deep Learning.,» 2023 IEEE 32nd Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), pp. 1-3, 2023.
- [7] J. Ma, S. Dang, P. Li, G. T. Watkings, K. Morris y M. Beach, «Transfer Learning for the Behavior Prediction of Microwave Structures.,» *Ma, J., Dang, S., Li, P., Watkins, G.T., Morris, K., & Beach, M. (2023). TraIEEE Microwave and Wireless Technology Letters,*, vol. 33, pp. 126-129, 2023.
- [8] LeCun, Yann and Bengio, Yoshua and Hinton y Geoffrey, «Deep learning,» *nature*, vol. 521, pp. 436--444, 2015.
- [9] S. E. Peik, G. Coutts y R. R. Mansour, «"Application of neural networks in microwave circuit modelling,",» S. E. Peik, G. Coutts and R. R. Mansour, "Application of neural

networks in micrConference Proceedings. IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.98TH8341), vol. 2, nº IEEE, pp. S. E. Peik, G. Coutts and R. R. Mansour, "Application of neural networks in microwave circuit modelling," Conference Proceedings. I928-931, 1998.

- [10] S. Dong, P. Wang y K. Abbas, «A survey on deep learning and its applications,» *Computer Science Review*, vol. 40, p. 100379, 2021.
- [11] M. T. McCann, K. H. Jin y M. Unser, «Convolutional neural networks for inverse problems in imaging: A review,» *IEEE Signal Processing Magazine*, vol. 34, pp. 85-95, 2017.
- [12] Z. Sun, J. Li y J. Fan, «Convolutional Neural Filtering for Intelligent Communications Signal Processing in Harsh Environments.,» *IEEE Access*, pp. 9, 8212-8219., 2021.
- [13] A. Marseet y F. Sahin, «Application of complex-valued convolutional neural network for next generation wireless networks.,» *IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, pp. 1-5, 2017.
- [14] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma y Q. Zhang, «Jin, J., Zhang, C., Feng, F., Na, WDeep Neural Network Technique for High-Dimensional Microwave Modeling and Applications to Parameter Extraction of Microwave Filters,» Jin, J., Zhang, C., Feng, F., Na, W., Ma, J., & Zhang, Q. (2019). Deep Neural Network Technique for High-Dimensional Microwave MoIEEE Transactions on Microwave Theory and Techniques, vol. 67, pp. 4140-4155, 2019.
- [15] S. Raghavan y C. Ilango, «Artificial neural network modeling of microwave filters.,» In 2006 9th International Conference on Electromagnetic Interference and Compatibility (INCEMIC 2006)., nº IEEE, pp. 561-564.
- [16] A. Corral Sierra, «Evaluación de un modelo basado en aprendizaje profundo para el modelado y predistorsión de amplificadores de potencia,» 2020.
- [17] S. P. Avila, «Facultad de Ingeniería,» Doctoral dissertation, Universidad de Cuenca, 2023.
- [18] J. Sun, S. Sun, X. Yu, Y. P. Chen y J. Hu, «A deep neural network based tuning technique of lossy microwave coupled resonator filters.».

- [19] L. Alvarez-Lopez y J. A. Becerra, «Application of deep learning methods to the mitigation of nonlinear effects in communication systems.,» 2019.
- [20] C. L. Chen, «Deep learning for pattern learning and recognition,» 2015.
- [21] R. Vargas, A. Mosavi y R. Ruiz, «Deep Learning: A Review.,» 2018.
- [22] N. G. Polson y V. O. Solokov, «Deep Learnig,» ArXiv, p. abs/1807.07987, 2018.
- [23] V. V. Mohan, P. Prakash y K. R. Resmi, «A Survey on Deep Learning Concepts and Techniques.,» International Journal of Advanced Research in Science, Communication and Technology., 2022.
- [24] P. Escudero, W. Alcocer y J. Paredes, «Escudero, P., Alcocer, W., & Recurrent Neural Networks and ARIMA Models for Euro/Dollar Exchange Rate Forecasting.,» *Applied Sciences,*, vol. 5658, nº https://doi.org/10.3390/app11125658, p. 11(12), 2021.
- [25] Q. j. Zhang, K. C. Gulpa y V. K. Devabhaktuni, «Artificial neural networks for RF and microwave design-from theory to practice,» *IEEE transactions on microwave theory and techniques*, vol. 51, nº 4, pp. 1339--1350, 2003.
- [26] M. Ehteshamullah, «Artificial Neural Network.,» Journal of Mathematical Sciences & Computational Mathematics., 2019.
- [27] S. Haykin, Neural networks and learning machines, 3/E, Pearson Education India, 2009.
- [28] D. M. Pozar, Microwave engineering, John wiley & sons, 2011.
- [29] J. C. Garcia, O. A. Rodriguez y J. L. Castillo, «GarcDesempeño de Modelos de Propagación en Comunicación Móvil para la zona de Caldas Parte 1: Modelos para áreas urbanas.,» García, J. C., Rodríguez, O. A., & Castillo, J. L. (2002). Desempeño de Modelos de Propagación en Comunicación Móvil para la X Congreso Internacional de Telecomunicaciones. Valdivia, Chile, pp. 1-2, 2002.
- [30] J. W. Congo Pastrana, «Aplicaciones del software Matlab/Simulink en modelado de sistemas básicos de telecomunicaciones,» (Bachelor's thesis, Quito, 2015)., 2015.

- [31] T. O'shea, y J. Hoydis, « An introduction to deep learning for the physical layer.,» IEEE Transactions on Cognitive Communications and Networking,, pp. 3(4), 563-575., 2017.
- [32] C. Zhang, P. Patras y H. Haddadi, «Zhang, C., Patras, P., & Deep learning in mobile and wireless networking: A survey.,,» *IEEE Communications surveys & tutorials*, pp. 21(3), 2224-2287., 2019.
- [33] F. Chollet, Deep learning with Python. Simon and Schuster., 2021.
- [34] Y. Zhao, W. Zhai, J. Zhao, T. Zhang, S. Sun, D. Niyato y K. Y. Lam, « A comprehensive survey of 6G wireless communications.,» arXiv preprint arXiv, p. 2101.03889, 2020.
- [35] A. Botchkarev, «Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and pronostics: Properties and typology,» *preprint arXiv*, vol. 1809, nº 03006, 2018.
- [36] Infineon, "BFP420 Surface mount wideband silicon NPN RF bipolar transistor Product description.", [Online]: Available: www.infineon.com.

ANEXOS

7.1 Anexo 1: Códigos Matlab

7.1.1 Cálculo de dimensiones de filtro microondas.

```
[] function filtrovariable()
     %Tteraciones
     n= 5000;
     datosFiltrovar= zeros(n,15);
    for i = 1:n
          % Generación de datos
          Delta = (0.1 - 1e-5) * rand() + 1e-5;
          Z0 = (100 - 20) * rand() + 20;
          rizado = 0.5 + round(rand()) * 2.5;
          datosFiltrovar(i,1)=Delta;
          datosFiltrovar(i,2)=Z0;
          datosFiltrovar(i,3)=rizado;
          % valores g en función del rizado
          if rizado == 0.5
             g = [1.7058, 1.2296, 2.5408, 1.2296, 1.7058, 1]; % g para rizado de 0.05 dB
          elseif rizado == 3.0
             g = [3.4817, 0.7618, 4.5381, 0.7618, 3.4817, 1]; % g para rizado de 3.0 dB
          else
             error('inválido. Ingresar 0.5 o 3.0 dB.');
          end
          % cálculo de impedancias par e impar
          [Zeven, Zodd] = calcular_impedancias(g, Delta, Z0);
            datosFiltrovar(i,4:9)=Zeven;
            datosFiltrovar(i,10:15)=Zodd;
      end
            [] function [Zeven, Zodd] = calcular impedancias(g, Delta, Z0)
                   ZoJ1 = sqrt(pi * Delta / (2 * g(1)));
                   ZoJ2 = (pi * Delta) / (2 * sqrt(g(1) * g(2)));
                   ZoJ3 = (pi * Delta) / (2 * sqrt(g(2) * g(3)));
                  ZoJ4 = (pi * Delta) / (2 * sqrt(g(3) * g(4)));
ZoJ5 = (pi * Delta) / (2 * sqrt(g(4) * g(5)));
                   ZoJ6 = sqrt(pi * Delta / (2 * g(5) * g(6)));
                   % Secciones del filtro
                  Zeven(1) = ZO * (1 + ZoJ1 + ZoJ1^2);
Zodd(1) = ZO * (1 - ZoJ1 + ZoJ1^2);
                   Zeven(2) = ZO * (1 + ZoJ2 + ZoJ2^2);
                  Zodd(2) = ZO * (1 - ZoJ2 + ZoJ2^2);
                  Zeven(3) = ZO * (1 + ZoJ3 + ZoJ3^2);
                  Zodd(3) = ZO * (1 - ZoJ3 + ZoJ3^2);
                   Zeven(4) = ZO * (1 + ZOJ4 + ZOJ4^2);
                  Zodd(4) = ZO * (1 - ZoJ4 + ZoJ4^2);
                   Zeven(5) = ZO * (1 + ZoJ5 + ZoJ5^2);
                   Zodd(5) = ZO * (1 - ZoJ5 + ZoJ5^2);
                  Zeven(6) = Z0 * (1 + ZoJ6 + ZoJ6^2);
Zodd(6) = Z0 * (1 - ZoJ6 + ZoJ6^2);
              end
```

7.1.2 Análisis de estabilidad del transistor BFP420, cálculo de líneas de transmisión y stubs para el amplificador.

```
%% Cálculo de coeficientes de reflexión y ganancias
  deltacomplex = s11 * s22 - s12 * s21;
  deltaMod = abs(deltacomplex);
  k = (1 - abs(s11)<sup>2</sup> - abs(s22)<sup>2</sup> + abs(deltaMod)<sup>2</sup>) / (2 * abs(s12 * s21));
  B1 = 1 + abs(s11)^2 - abs(s22)^2 - deltaMod^2;
  B2 = 1 + abs(s22)^2 - abs(s11)^2 - deltaMod^2;
  C1 = s11 - deltacomplex * conj(s22);
  C2 = s22 - deltacomplex * conj(s11);
  % Coeficiente reflexión en la carga
  rs = (B1 + sqrt(B1^2 - 4 * abs(C1)^2)) / (2 * C1);
  if abs(rs) > 1
      rs = (B1 - sqrt(B1^2 - 4 * abs(C1)^2)) / (2 * C1);
  end
  % Coeficiente reflexión en la fuente
  rl = (B2 + sqrt(B2^2 - 4 * abs(C2)^2)) / (2 * C2);
  if abs(rl) > 1
      rl = (B2 - sqrt(B2^2 - 4 * abs(C2)^2)) / (2 * C2);
  end
         %mostrar resultados del coeficiente de reflexion en
         % Mostrar los resultados
         fprintf('\nResultados:\n');
         fprintf('Valor de delta: %.4f + %.4fi\n', real(delta), imag(delta));
         fprintf('Valor de K: %.4f\n', K);
         % condiciones de estabilidad
         if abs(delta) < 1 && K > 1
             fprintf('El amplificador es estable.\n');
%% Ganancias
gs = 10 * log10(1 / (1 - abs(rs)^2));
g0 = 10 * log10(abs(s21)^2);
gl = 10 * log10((1 - abs(rl)^2) / (abs(1 - s22 * rl)^2));
gtotal = gs + g0 + gl;
Gtotal = 10 * log10((abs(s21) / abs(s12)) * (k - sqrt(k^2 - 1)));
fprintf('Ganancia máxima: %d \n', gtotal);
%% Cálculo de longitudes de stub y línea
% Red de acoplo fuente
zs = z0 * ((1 + rs) / (1 - rs));
[dsarr, lsarr] = analitic_solution_single_stub(zs, z0, 1, 1); % Stub único
llines = 0.5 - dsarr(1);
lstubs = 0.5 - lsarr(1);
% Red de acoplo carga
zl = z0 * ((1 + rl) / (1 - rl));
[dlarr, llarr] = analitic_solution_single_stub(zl, z0, 1, 1);
llinel = 0.5 - dlarr(1);
lstubl = 0.5 - llarr(1);
fprintf('Acoplador fuente, linea: %d º, stub: %d º \n', real(llines) * 360, real(lstubs) * 360);
fprintf('Acoplador carga, linea: %d º, stub: %d º\n', real(llinel) * 360, real(lstubl) * 360);
```

7.1.3 Cálculo de las líneas del acoplador direccional

```
[] function Acoplador1()
 num = 3000;
   % MatriX para result
     coupler = zeros(num, 5);
for i = 1:num
         Z0 = (100 - 20) * rand() + 20; % Impedancia
 % longitud de onda
 lambda = 360;
 %longitudes líneas
 LA1 = sqrt(2) * Z0;
 LA2 = sqrt(2) * Z0;
 LA3 = sqrt(2) * Z0;
 LA4 = 3 * lambda / 4 ;
  % resultados en la matriX
         coupler (i, :) = [Z0, LA1, LA2, LA3, LA4];
  end
   %archivo mat
     save('datosAcoplador.mat', 'coupler');
 end
```

7.1.4 Cálculo de los componentes del filtro pasa bajo

```
[] function Pasabajo1()
 % parámetros
 g = [0.6180, 1.1680, 2.0000, 1.6180, 0.6180];
 num = 12000;
  % Matriz resultados
     lowpass = zeros(num, 7);
for i = 1:num
         Z0 = (100 - 20) * rand() + 20; % Impedancia entre 10 y 70 ohms
         f val = 10e6:10e6:500e6; % Valores de frecuencia de corte de 1 a 5 GHz
         fc = f_val(randi(length(f_val))); % Selección frecuencia
 % Cálculo frec angular
 wc = 2 * pi * fc;
 % Calculo componentes del filtro
 C1 = (g(1) / (Z0 * wc)) * 1e12; %pF
 L2 = ((Z0 * g(4)) / wc) * 1e9; %nH
 C3 = (g(3) / (Z0 * wc)) * 1e12; %pF
 L4 = ((Z0 * g(4)) / wc) * 1e9; %nH
 C5 = (g(5) / (Z0 * wc)) * 1e12; %pF
  % Guardar resultados matriX
         lowpass(i, :) = [Z0, fc/1e6,C1, L2, C3, L4, C5];
  end
   % archivo .mat
     save('datosPasabajo.mat', 'lowpass');
```

7.2 Anexo 2: Códigos Python

7.2.1 Creación de red neuronal

```
import numpy as np #calculos matematicos
                                                                                                                            A1
from scipy.io import loadmat #leer variables de matlab
from tensorflow.keras.models import Sequential #importar el framework
from tensorflow.keras.layers import Dense #tipo de conexion
from tensorflow.keras.callbacks import EarlyStopping #paquete para evitar el sobreentrenamiento
train_data = loadmat('C:/IET/TESIS/Archivos Matlab/datosAcoplador') #buscar el archivo
matrix_data = train_data['coupler'] #leer la matrix
X = matrix_data[:, 0:1] #datos de entrada de la red
y = matrix_data[:, 1:5] #salidas de la red
split = int(0.8 * len(X)) # entradas y salidas para entrenamiento y validación 80/20
X_train, X_val = X[:split], X[split:]
y_train, y_val = y[:split], y[split:]
#Definir el modelo de red neuronal
model = Sequential()
#Definir capas
#capas de entrada
model.add(Dense( units: 256, input_dim=1, activation='relu'))
#capas ocultas
model.add(Dense( units: 256 , activation='relu'))
model.add(Dense( units: 256 , activation='relu'))
model.add(Dense( units: 256 , activation='relu'))
#capa de salida
model.add(Dense( units: 4 , activation='linear'))
#compilar el modelo
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['mae']) #metricas de error
early_stopping = EarlyStopping(monitor='loss', patience=10, restore_best_weights=True)
#entrenar el modelo con early stop
history = model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=400,batch_size=256,callbacks=[early_stopping])
#resumen del modelo
model.summarv()
model.save('modelo4.keras')
```

7.2.2 Realizar predicciones

```
> import numpy as np
from tensorflow.keras.models import load_model
model = load_model('modelo3.keras')
nueva_entrada = np.array([[50,2.0],])
prediccion = model.predict(nueva_entrada)
print(prediccion)
```
7.3 Anexo 3: Tablas de entrenamiento

7.3.1 Entrenamiento red neuronal 1

Numero de entrenamiento	Datos de entrenamiento	Capas de entrada	Capas ocultas	Capas de salida	Épocas
	1000	3	64	12	100
1	Error mse/mae: 2	.1362/2.5085			
	3000	3	128	12	100
2	Error mse/mae: 2.1015/2.0195				
	5000	3	128	12	100
3	Error mse/mae: 2.0075/2.1022				
	7000	3	256	12	200
4	Error mse/mae: 0.6597/0.5893				
	8000	3	256	12	200
5	Error mse/mae: 0.7385/0.4672				

Tabla 17: Características de la red neuronal 1 (filtro)

Tabla 18: Valores Deep Learning vs valores calculado Red neuronal 1.

Número	Sección del	Z _{EVEN} Deep	Z _{EVEN}	Z _{ODD} Deep	ZODD
entrenamiento	filtro	Learning	Calculado	Learning	Calculado
	Sección 1	61.1343	59.6918	42.9974	43.0708
	Sección 2	53.1948	51.6799	47.7742	48.4260
	Sección 3	52.7114	51.3686	48.1680	48.7025
1	Sección 4	52.7011	51.3686	48.1478	48.7025
	Sección 5	53.7011	51.6799	47.7519	48.4260
	Sección 6	61.1376	59.6918	43.1330	43.0708
	Sección 1	60.3495	59.6918	43.3501	43.0708
	Sección 2	52.9399	51.6799	47.6728	48.4260
2	Sección 3	52.3359	51.3686	48.1515	48.7025
	Sección 4	52.5532	51.3686	48.1014	48.7025
	Sección 5	52.7652	51.6799	47.7628	48.4260
	Sección 6	60.4554	59.6918	43.2781	43.0708
3	Sección 1	60.5087	59.6918	43.2467	43.0708
	Sección 2	52.8754	51.6799	47.7915	48.4260
	Sección 3	52.4110	51.3686	48.1419	48.7025
	Sección 4	52.4571	51.3686	48.1379	48.7025
	Sección 5	52.9553	51.6799	47.8591	48.4260
	Sección 6	60.5335	59.6918	43.2446	43.0708
4	Sección 1	57.4822	59.6918	44.1157	43.0708
	Sección 2	51.5826	51.6799	47.9987	48.4260
	Sección 3	51.3282	51.3686	48.2443	48.7025
	Sección 4	51.3770	51.3686	48.3387	48.7025
	Sección 5	51.6828	51.6799	48.0352	48.4260
	Sección 6	57.5630	59.6918	44.1464	43.0708

5	Sección 1	58.2363	59.6918	43.822147	43.0708
	Sección 2	51.7741	51.6799	48.345592	48.4260
	Sección 3	51.3680	51.3686	48.454338	48.7025
	Sección 4	51.5112	51.3686	48.37619	48.7025
	Sección 5	51.8653	51.6799	48.13192	48.4260
	Sección 6	58.4753	59.6918	43.798462	43.0708

7.3.2 Entrenamiento red neuronal 2

Numero de entrenamiento	Datos de entrenamiento	Capas de entrada	Capas ocultas	Capas de salida	Épocas
	232	1	16	1	10000
1	Error mse/mae: 0.336/3.4403				
	232	1	128	1	7000
2	Error mse/mae: 0.6433/52.3755				
	232	1	128	1	4000
3	Error mse/mae: 0.6663/52.376				
	232	1	256	1	2000
4	Error mse/mae: 30.5179/57.5305				
	232	1	256	1	5000
5	Error mse/mae: 30.8196/62.5767				

Tabla 19: Características de la red neuronal 2 (amplificador)

Tabla 20: Valores Deep Learning vs valores calculados red neuronal 2.

Número entrenamiento	Acopladores	línea de transmisión (Deep Learning)	línea de transmisión (Calculado)	Stub (Deep Learning)	Stub (Calculado)
	Acoplador fuente	6.2749°	5.65°	63.5437°	60.02°
1	Acoplador carga	69.44°	69.60°	45.4440	45.80°
2	Acoplador fuente	5.7156°	5.65°	62.0877°	60.02°
	Acoplador carga	69.8943°	69.60°	44.5241°	45.80°
3	Acoplador fuente	6.45035°	5.65°	62.3648°	60.02°
	Acoplador carga	70.2683°	69.60°	44.9707°	45.80°
4	Acoplador fuente	76.4860	5.65°	53.0531°	60.02°

	Acoplador	51.5045°	69.60°	34.2549°	45.80°
	carga				
	Acoplador	84.8504°	5.65°	65.6277°	60.02°
5	fuente				
	Acoplador	48.1307°	69.60°	26.1839°	45.80°
	carga				

7.3.3 Entrenamiento red neuronal 3

|--|

Numero de	Datos de	Capas de	Capas	Capas de	Épocas	
entrenamiento	entrenamiento	entrada	ocultas	salida		
	800	1	4	3	100	
1	Error mse/mae: 1	Error mse/mae: 19.936/21.5889				
_	900	1	4	3	100	
2	Error mse/mae: 20.329/20.2037					
	1000	1	4	3	200	
3	Error mse/mae: 0.785/0.464					
_	2000	1	4	3	200	
4	Error mse/mae: 0.7499/0.2489					
	4000	1	7	3	500	
5	Error mse/mae: 0.1454/0.1515					

Tabla 22: Valores Deep Learning vs Valores calculados, red neuronal 3

Número de entrenamiento	Linea del acoplador	Valor (deep learnig)	Valor (calculado)
1	Línea 1	72.1533	70.71
	línea 2	72.1327	70.71
	línea 3	71.8341	70.71
	línea 4	199.101	270.00
2	Línea 1	72.6224	70.71
	línea 2	72.6087	70.71
	línea 3	72.6420	70.71
	línea 4	203.543	270.00
3	Línea 1	70.7211	70.71
	línea 2	70.5371	70.71
	línea 3	70.7141	70.71
	línea 4	270.420	270.00

4	Línea 1	70.5614	70.71
	línea 2	70.6226	70.71
	línea 3	70.6118	70.71
	línea 4	268.8304	270.00
	Línea 1	70.7266	70.71
	línea 2	70.7283	70.71
	línea 3	70.7698	70.71
	línea 4	269.7416	270.00

7.3.4 Entrenamiento red neuronal 4

Tabla 23: Características de la red neuronal 4, (Filtro pasa bajo)

Numero de entrenamiento	Datos de entrenamiento	Capas de entrada	Capas ocultas	Capas de salida	Épocas		
	2000	2	4	5	200		
1	Error mse/mae: 2	3.906/23.671					
	4000	2	6	5	300		
2	Error mse/mae: 1.452/1.121						
	6000	2	8	5	300		
3	Error mse/mae: 0.4761/0.4510						
	10000	2	15	5	7000		
4	Error mse/mae: 0.6431/0.4533						
_	12000	2	15	5	1000		
5 Error mse/mae: 0.5047/0.4373							

Tabla 24: Valores Deep Learning vs Valores calculados, Red neuronal 4

Número de	Componente del	Valor	Valor
entrenamiento	filtro	(deep learnig)	(calculado)
	C1	7.1972pf	6.557pf
	L2	39.489nh	42.918nh
1	С3	20.2363pf	21.220pf
	L4	38.7529nh	42.9180nh
	C5	6.3756pf	6.667pf
	C1	6.6619pf	6.557pf
	L2	42.7039nh	42.918nh
2	C3	20.8240pf	21.220pf

	L4	42.7759nh	42.9180nh
	C5	6.4156pf	6.667pf
3			
	C1	6.5535pf	6.557pf
	L2	43.6862nh	42.918nh
	C3	21.2556pf	21.220pf
	L4	43.7519nh	42.9180nh
	C5	6.5875pf	6.667pf
4			
	C1	6.5274pf	6.557pf
	L2	43.1312nh	42.918nh
	C3	21.0533pf	21.220pf
	L4	43.2073nh	42.9180nh
	C5	6.6057pf	6.667pf
5			
	C1	6.7450pf	6.557pf
	L2	42.5784nh	42.918nh
	C3	21.7143pf	21.220pf
	L4	42.7312nh	42.9180nh
	C5	6.8654pf	6.667pf