

TFG Pablo Díaz Lobo

Francisco Javier Gómez Pulido

27 enero 2025

Resumen

Este trabajo aborda la optimización del diseño de convertidores analógico-digitales Sigma-Delta mediante redes neuronales, automatizando un proceso tradicionalmente manual. Se emplean modelos neuronales para predecir variables de diseño a partir de especificaciones, mejorando eficiencia y precisión. Los resultados muestran configuraciones óptimas para minimizar errores y evitar sobreajustes, validando la aplicabilidad de estas técnicas en diseños complejos. Aunque el entrenamiento requiere más tiempo que otros algoritmos, las redes son ventajosas para múltiples iteraciones de diseño.

Índice

1	Introducción	3
2	Convertidores analógico-digitales	3
3	Redes neuronales	5
3.1	Fundamentos	5
3.2	Conexiones entre neuronas	5
3.3	Topologías de red	6
3.4	Entrenamiento	6
3.5	Aplicación al diseño de circuitos	6
4	Planteamiento del problema	6
5	Primer caso de estudio: Modulador $\Sigma\Delta$ SC de segundo orden	7
5.1	Dependencia con el número de épocas	7
5.2	Dependencia con el tamaño del <i>batch</i>	7
5.3	Dependencia con el factor de aumento	8
5.4	Dependencia con el número de neuronas	9
5.4.1	¿Por qué se obtienen peores resultados?	9
5.5	Dependencia con el número de capas	10
5.6	Dependencia con el tamaño del dataset para validación	10
5.7	Resumen de los resultados	11
5.8	Validación de las predicciones de las redes	11
6	Segundo caso de estudio: Modulador $\Sigma\Delta$ Gm-C de segundo orden	12
6.1	Resumen de resultados	13
7	Conclusiones	13

1. Introducción

Los circuitos integrados (CI) son esenciales en dispositivos electrónicos y su complejidad aumenta cada año debido a la alta demanda. Aunque los circuitos digitales dominan, los analógicos siguen siendo indispensables, especialmente en aplicaciones que requieren alta precisión. Sin embargo, el diseño de estos circuitos aún es manual y largo, lo que limita su eficiencia.

Este trabajo se centra en los **convertidores analógico-digitales Sigma-Delta** ($\Sigma\Delta$), conocidos por su versatilidad y eficiencia en digitalización de señales de alta resolución y bajo ancho de banda, como en audio y dispositivos biomédicos. Se propone un enfoque basado en redes neuronales para automatizar y optimizar su diseño. Estas redes predicen variables clave directamente desde las especificaciones, reduciendo iteraciones y acelerando el proceso. Este modelo inverso aprovecha técnicas avanzadas de inteligencia artificial, mejorando la productividad y calidad en el diseño de CI.

2. Convertidores analógico-digitales

Convertidor analógico-digital (ADC): Dispositivo electrónico que convierte una señal analógica (continua) en una señal digital (discreta).

Las señales analógicas son continuas en el tiempo y en amplitud, como las ondas de sonido o las variaciones de temperatura. Sin embargo, las computadoras y muchos sistemas digitales solo pueden procesar información en forma de números, es decir, datos digitales.

El proceso de conversión se realiza en dos pasos principales:

1. **Muestreo:** La señal analógica se toma a intervalos regulares de tiempo, conocidos como frecuencia de muestreo.
2. **Cuantificación:** Cada muestra se redondea al valor digital más cercano dentro de un rango discreto.

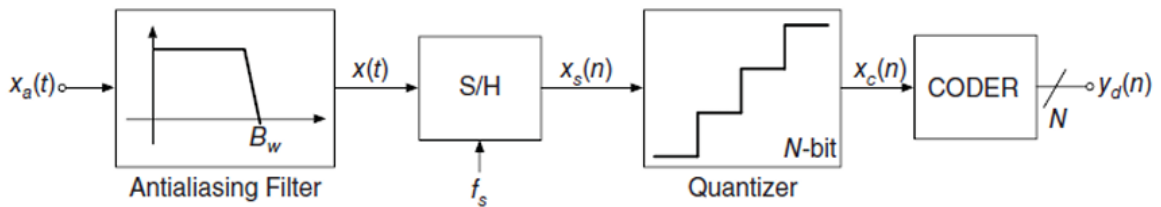


Figura 1: Diagrama de bloques conceptual

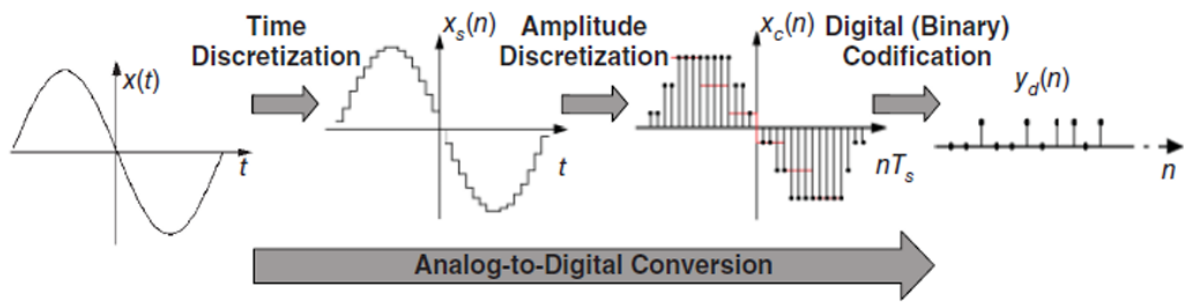


Figura 2: Pasos de la conversión

Componentes clave de los ADC:

1. **Filtro anti-aliasing** (AAF): Es un filtro de paso bajo que elimina las componentes de frecuencia superiores a la mitad de la frecuencia de muestreo para evitar errores (aliasing) según el teorema de Nyquist.
2. **Muestreador**: Convierte la señal analógica en una señal discreta en el tiempo.
3. **Cuantizador**: Redondea los valores de la señal muestreada a niveles discretos.
4. **Codificador**: Asigna un código binario a cada nivel cuantizado, produciendo la señal digital final.

Características importantes:

- **Frecuencia de muestreo** (f_s): Determina la cantidad de muestras tomadas por segundo. Según el teorema de Nyquist, f_s debe ser al menos el doble del ancho de banda de la señal ($2B_w$) para evitar pérdidas de información.
- **Resolución** (N): Indica el número de bits usados para representar cada muestra. A mayor N , mayor precisión.
- **Relación señal-ruido** (SNR): Mide la calidad de la conversión; una alta SNR implica menor ruido en la señal digital resultante.

Los **Sigma-Delta** ($\Sigma\Delta$) son un tipo particular de ADC que se usan cuando se requiere alta precisión y rango dinámico, como en audio digital, telecomunicaciones y sensores de precisión. El Sigma-Delta emplea una técnica especial de muestreo y cuantificación que se puede descomponer en:

1. **Sobremuestreo**: En lugar de tomar muestras a la frecuencia mínima requerida, el Sigma-Delta toma muchas más muestras por segundo. Esto desplaza el ruido a frecuencias altas.
2. **Modulación Sigma-Delta**: Este proceso convierte la señal analógica en una secuencia digital de alta frecuencia con solo unos pocos niveles posibles (0 y 1, por ejemplo).
3. **Filtrado digital**: Posteriormente, un filtro digital elimina el ruido introducido durante la modulación y reconstruye la señal digital con mayor precisión.

3. Redes neuronales

Las redes neuronales (RNN) son modelos computacionales inspirados en el funcionamiento del cerebro humano. Su desarrollo comenzó en 1943, con el trabajo de McCulloch y Pitts, quienes introdujeron el concepto de neurona como unidad computacional. Sin embargo, el interés inicial decayó tras la publicación del libro *Perceptrons* de Minsky y Papert en 1969, que señalaba las limitaciones de los modelos existentes. Décadas después, gracias a avances teóricos como el algoritmo de *backpropagation* y al aumento de la capacidad de procesamiento, las RNN resurgieron y se convirtieron en herramientas fundamentales para resolver problemas complejos en diversos campos, incluyendo el diseño de circuitos integrados.

3.1. Fundamentos

Las RNN están formadas por unidades de procesamiento, denominadas neuronas, conectadas entre sí mediante pesos que determinan la intensidad de las conexiones. Cada neurona tiene un estado de activación que se calcula en función de las entradas que recibe y una función de activación. Los elementos principales de una red neuronal incluyen:

- **Neuronas:** Unidades que procesan la información.
- **Pesos:** Coeficientes que ponderan la influencia de una neurona sobre otra.
- **Estado de activación:** Representa la salida de cada neurona.
- **Regla de propagación:** Calcula la entrada efectiva de una neurona como la suma ponderada de las entradas más un sesgo (*bias*).
- **Funciones de activación:** Determinan la salida de la neurona en función de su entrada efectiva. Entre las más comunes están la función sigmoide:

$$y_k = \frac{1}{1 + e^{-s_k}} \quad (1)$$

y la ReLu (Unidad Lineal Rectificada):

$$y_k = \max(0, s_k) \quad (2)$$

- **Aprendizaje:** Proceso mediante el cual los pesos de la red se ajustan para minimizar una función de pérdida.

3.2. Conexiones entre neuronas

Las conexiones entre las neuronas determinan cómo se transmite la información a través de la red. La entrada efectiva de una neurona se calcula como:

$$s_k(t) = \sum_j w_{jk} y_j(t) + \theta_k(t) \quad (3)$$

donde w_{jk} es el peso de la conexión, $y_j(t)$ es la salida de la neurona j y $\theta_k(t)$ es el sesgo de la neurona k .

3.3. Topologías de red

Existen diferentes configuraciones o topologías para las redes neuronales:

- **Redes *Feed-Forward*:** Las conexiones avanzan en una única dirección, desde las neuronas de entrada hacia las de salida, pasando por capas ocultas. Son ideales para problemas de regresión y clasificación.
- **Redes *Recurativas*:** Incluyen conexiones hacia atrás, permitiendo que las salidas previas influyan en los estados futuros. Son útiles para tareas que requieren memoria, como el procesamiento de secuencias.

En este trabajo se utilizan redes *Feed-Forward*, ya que se ajustan mejor a problemas de regresión donde se relacionan especificaciones con variables de diseño.

3.4. Entrenamiento

El entrenamiento consiste en ajustar los pesos de la red para minimizar el error entre las salidas predichas y las salidas reales, medida mediante una función de pérdida. Los pasos básicos son:

1. Propagación hacia adelante: Calcula las salidas de la red a partir de las entradas.
2. Cálculo de la pérdida: Evalúa el error utilizando métricas como el error cuadrático medio (MSE).
3. Retropropagación: Ajusta los pesos mediante el cálculo del gradiente de la pérdida.
4. Actualización de los pesos: Se realiza con algoritmos como Adam o SGD (*Stochastic Gradient Descent*).

El conjunto de datos se divide en dos partes: entrenamiento y validación. Durante el entrenamiento, los datos se procesan en lotes (*batches*) y, al completar una pasada completa por el conjunto, se dice que ha transcurrido una época.

3.5. Aplicación al diseño de circuitos

En este trabajo, las redes neuronales se utilizan para automatizar y optimizar el diseño de moduladores Sigma-Delta. En lugar del enfoque tradicional, donde se prueban múltiples diseños hasta alcanzar las especificaciones deseadas, se emplea un enfoque inverso: la red neuronal predice directamente las variables de diseño a partir de las especificaciones. Esto permite acelerar el proceso y mejorar la calidad del diseño final.

4. Planteamiento del problema

Dado un vector de especificaciones S_i (por ejemplo, SNR, OSR, consumo de potencia), se busca encontrar el vector de variables de diseño V_i que optimice dichas especificaciones. Para ello:

$$V = \arg \max P(V \mid S) \quad (4)$$

El modelo se entrena con un conjunto de datos que relaciona diseños válidos con sus especificaciones. Además, se aplican técnicas de extensión del conjunto de datos, escalado y normalización para mejorar el rendimiento de las redes. La función de pérdida utilizada incluye el MSE y un término de regularización L_2 para evitar sobreajustes.

Este enfoque permite que la red neuronal aprenda patrones óptimos de diseño y facilite la automatización del proceso, mejorando la eficiencia y reduciendo los tiempos de desarrollo.

5. Primer caso de estudio: Modulador $\Sigma\Delta$ SC de segundo orden

En este caso de estudio, se aborda el diseño de un modulador $\Sigma\Delta$ de segundo orden implementado mediante capacitores conmutados (*switched-capacitors*, SC). Este tipo de moduladores es ampliamente utilizado por su alta precisión y eficiencia energética, siendo ideales para aplicaciones de audio digital y dispositivos biomédicos.

El análisis se centra en optimizar las variables de diseño que influyen en el rendimiento del modulador, utilizando redes neuronales para automatizar este proceso y reducir el esfuerzo manual necesario.

5.1. Dependencia con el número de épocas

Para evaluar la influencia del número de épocas, se utilizó una red neuronal con 100 épocas de entrenamiento. La función de pérdida y el error absoluto medio (MAE) se estabilizan rápidamente, lo que sugiere que 100 épocas son suficientes para lograr una convergencia adecuada sin incurrir en un sobreajuste.

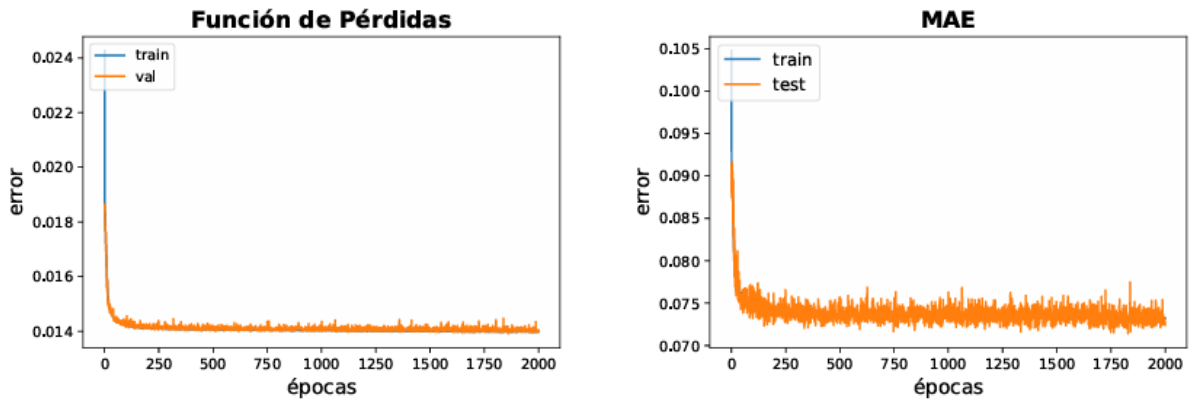


Figura 3: Dependencia con el número de épocas

5.2. Dependencia con el tamaño del *batch*

El tamaño del *batch* tiene un impacto significativo en el rendimiento de la red. Se evaluaron tamaños entre 64 y 8192, observándose que un *batch* de 256 logra un equilibrio óptimo entre precisión (MSE y MAE) y tiempo de entrenamiento. Aumentar el tamaño

del *batch* reduce el ruido en las actualizaciones, pero puede dificultar la adaptación precisa al conjunto de datos.

Tamaño Batch	8192	4096	2048	1024	512	256	128	64
MSE	0.0169	0.0152	0.0148	0.0144	0.0143	0.0143	0.0142	0.0143
MAE	0.0853	0.0784	0.0776	0.0758	0.0753	0.0749	0.0743	0.0747

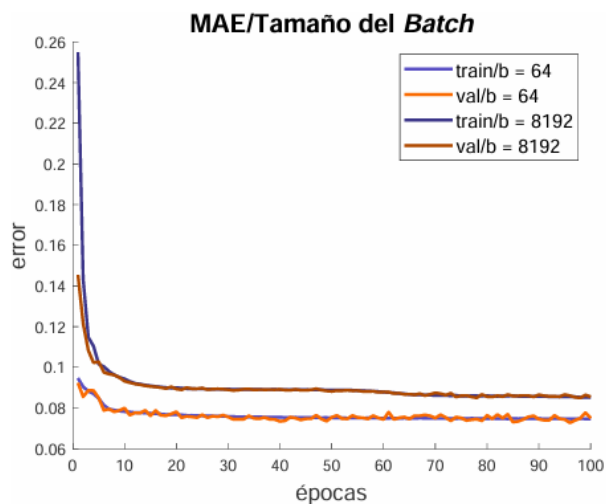


Figura 4: Dependencia con el tamaño del *batch*

5.3. Dependencia con el factor de aumento

El factor de aumento (K) determina la extensión del conjunto de datos de entrenamiento mediante copias adicionales con modificaciones en las especificaciones. Se evaluaron valores de $K = 1, 10, 40$ y 400 . Aunque el aumento del conjunto de datos mejora la capacidad de la red para generalizar fuera del rango original, no se observaron mejoras significativas en métricas como el error cuadrático medio (MSE) y el error absoluto medio (MAE). Se concluyó que valores bajos de K proporcionan mejores tiempos de entrenamiento sin comprometer la precisión.

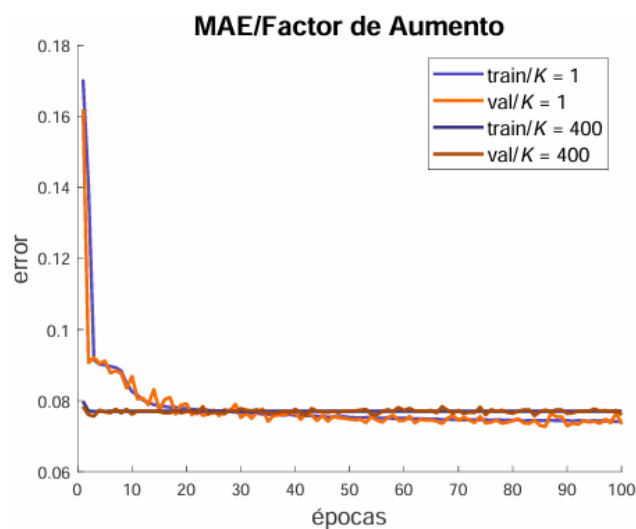


Figura 5: Dependencia con el factor de aumento

5.4. Dependencia con el número de neuronas

El número de neuronas en las capas ocultas afecta directamente la capacidad de la red para aprender patrones complejos. Se evaluaron redes con un factor de escala f aplicado al número de neuronas iniciales, donde f varió de 1 a 10. Los resultados mostraron que aumentar excesivamente el número de neuronas incrementa los errores debido a problemas como el sobreajuste y errores numéricos. Además, se identificó que la red base con un menor número de neuronas ya operaba de manera óptima, lo que indica que agregar más neuronas no aporta beneficios significativos y solo aumenta los recursos computacionales requeridos.

Neuronas	x1	x4	x7	x10
MSE	0.0142	0.0142	0.0580	0.1481
MAE	0.0743	0.0744	0.1607	0.3427

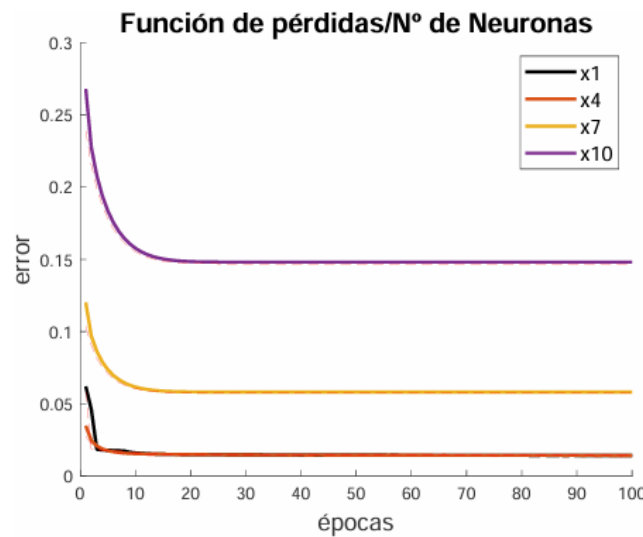


Figura 6: Dependencia con el número de neuronas

5.4.1. ¿Por qué se obtienen peores resultados?

Los resultados peores en ciertas configuraciones de la red se atribuyen a dos factores principales:

- **Sobreajuste:** Ocurre cuando la red neuronal es demasiado compleja para el tamaño del conjunto de datos, lo que provoca que la red memorice los datos de entrenamiento en lugar de generalizar correctamente.
- **Errores numéricos:** Aumentar excesivamente el número de neuronas y capas incrementa la complejidad computacional, lo que puede causar problemas de precisión numérica en los cálculos durante el entrenamiento.

Estos problemas resaltan la importancia de encontrar un balance entre la capacidad de la red y el tamaño del conjunto de datos.

5.5. Dependencia con el número de capas

El número de capas ocultas afecta la capacidad de la red neuronal para modelar relaciones complejas entre las variables. Se evaluaron configuraciones con 1, 4, 7 y 10 capas ocultas:

- Las redes con 4 capas mostraron un mejor desempeño, reduciendo el error cuadrático medio (MSE) y el error absoluto medio (MAE).
- Sin embargo, aumentar el número de capas más allá de 4 no proporcionó mejoras significativas y aumentó el riesgo de sobreajuste.

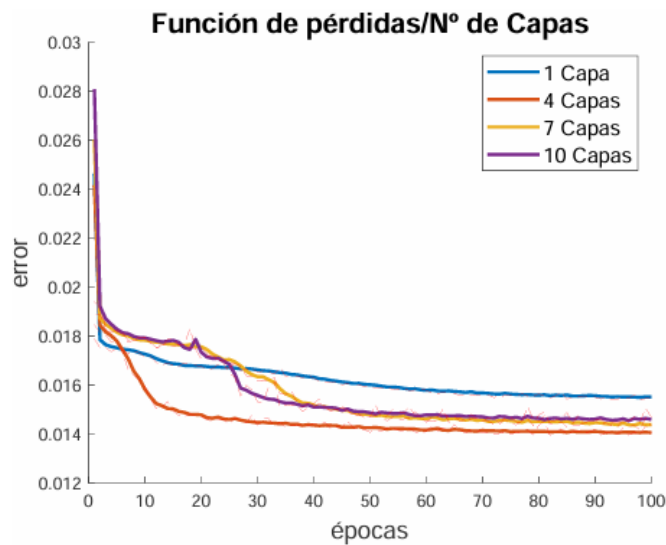


Figura 7: Dependencia con el número de capas

Se concluyó que una red con 4 capas ocultas ofrece un equilibrio óptimo entre precisión y eficiencia computacional.

5.6. Dependencia con el tamaño del dataset para validación

El tamaño del conjunto de datos de validación afecta la capacidad del modelo para evaluar su rendimiento de manera efectiva. Se analizaron diferentes tamaños de dataset de validación, observándose que:

- Un conjunto de validación muy pequeño puede no ser representativo, lo que conduce a una evaluación poco fiable del modelo.
- Por otro lado, un conjunto de validación excesivamente grande reduce la cantidad de datos disponibles para entrenamiento, afectando negativamente el rendimiento del modelo.

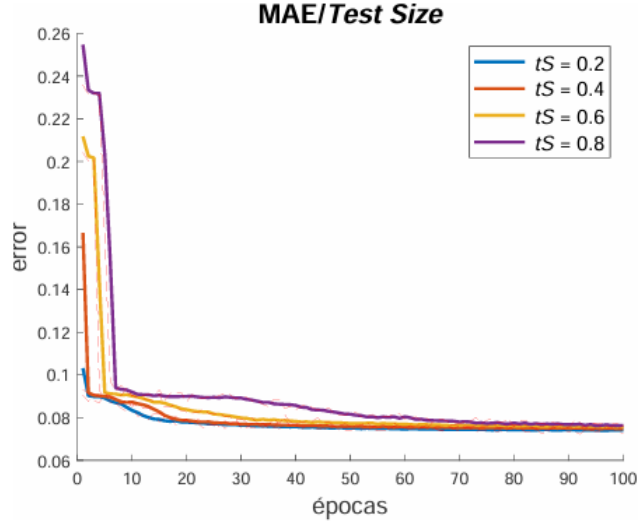


Figura 8: Dependencia con el tamaño del conjunto de validación

El mejor equilibrio se obtuvo con un conjunto de validación que representa el 20 % del total de datos, permitiendo una evaluación adecuada sin comprometer el entrenamiento.

5.7. Resumen de los resultados

El análisis del primer caso de estudio mostró que las redes neuronales pueden automatizar de manera efectiva el diseño de moduladores $\Sigma\Delta$ SC de segundo orden. Entre los resultados más relevantes destacan:

- La configuración óptima para la red incluye 4 capas ocultas, con un tamaño de *batch* de 256 y un conjunto de datos de validación equivalente al 20 % del total.
- Las métricas obtenidas validan el modelo, con un error cuadrático medio (MSE) de 0,0142 y un error absoluto medio (MAE) de 0,0743.
- Factores como el tamaño del conjunto de datos, el número de neuronas y capas, y el tamaño del *batch* tienen un impacto significativo en el rendimiento del modelo.

5.8. Validación de las predicciones de las redes

Para evaluar la precisión y confiabilidad de las redes neuronales entrenadas, se realizó una **validación cruzada** utilizando un conjunto de datos independiente. Resultados:

- Una alta correlación entre las predicciones de las redes y los valores reales, lo que confirma que el modelo generaliza correctamente a datos no vistos durante el entrenamiento.
- Los errores de predicción se mantienen bajos en todo el rango de especificaciones, evidenciando que el modelo es robusto y aplicable a diferentes configuraciones de diseño.
- La visualización de los resultados mostró una distribución uniforme de los errores, sin sesgos significativos hacia valores altos o bajos.

Estos resultados refuerzan la validez del enfoque propuesto y su capacidad para automatizar el diseño de moduladores $\Sigma\Delta$ de manera eficiente y precisa.

6. Segundo caso de estudio: Modulador $\Sigma\Delta$ Gm-C de segundo orden

En este segundo caso de estudio, se aborda el diseño y análisis de un modulador $\Sigma\Delta$ de segundo orden implementado mediante transconductancias (Gm -C, transconductance-capacitor). Este tipo de moduladores se utiliza ampliamente en aplicaciones de alta frecuencia, como telecomunicaciones y sistemas de conversión de datos, debido a su capacidad para operar a velocidades elevadas con un consumo energético relativamente bajo.

El modulador $\Sigma\Delta$ Gm-C de segundo orden consta de:

- **Transconductores:** Implementan las operaciones de integración de manera continua, a diferencia de los capacitores conmutados utilizados en el primer caso de estudio. Los transconductores convierten señales de voltaje en señales de corriente.
- **Capacitores:** Almacenan carga, proporcionando la capacidad necesaria para realizar las operaciones de integración de señales en el dominio analógico.
- **Cuantizador:** Como en el caso anterior, se utiliza un cuantizador de un bit que simplifica el diseño y permite un conformado eficiente del ruido.

La arquitectura del modulador incluye un lazo de retroalimentación que permite desplazar el ruido de cuantización fuera de la banda de interés, logrando una alta relación señal-ruido (SNR) dentro de la banda útil.

Se analizaron las siguientes variables de diseño:

- **Transconductancia (g_m):** Influye en la velocidad de operación y el rango dinámico del modulador.
- **Capacidad (C):** Afecta las propiedades de filtrado del sistema.
- **Frecuencia de corte (f_c):** Determinada por la relación entre g_m y C , define el rango de operación del modulador.

Las especificaciones utilizadas para evaluar el rendimiento del modulador fueron:

- Relación señal-ruido (SNR).
- Figura de mérito (FOM).
- Consumo de potencia (P).

Para entrenar las redes neuronales, se generó un conjunto de datos mediante simulaciones paramétricas del modulador:

- Se consideraron combinaciones amplias de g_m , C y f_c .
- Para cada configuración, se calcularon las especificaciones SNR, FOM y P mediante simulaciones en el dominio del tiempo y la frecuencia.

- Los datos se filtraron para excluir combinaciones no válidas o que no cumplieran con los objetivos de diseño.

Este proceso resultó en un conjunto de datos suficientemente representativo para capturar las relaciones entre las variables de diseño y las especificaciones.

Se utilizó una red neuronal con una arquitectura similar a la del primer caso de estudio, pero adaptada a las características específicas del modulador Gm-C. Los parámetros principales fueron:

- **Entradas:** Las variables de diseño normalizadas (g_m , C y f_c).
- **Capas ocultas:** Tres capas con 120, 240 y 60 neuronas, diseñadas para capturar las interacciones complejas entre las variables.
- **Salidas:** Predicciones de las especificaciones SNR, FOM y consumo de potencia.

El entrenamiento utilizó un tamaño de *batch* de 256 y una tasa de aprendizaje optimizada con el algoritmo Adam. Se realizaron múltiples pruebas para ajustar los hiperparámetros y evitar problemas como el sobreajuste.

6.1. Resumen de resultados

Los resultados obtenidos demuestran la efectividad del modelo:

- **Error cuadrático medio (MSE):** 0,0121, indicando una alta precisión en las predicciones.
- **Error absoluto medio (MAE):** 0,0625, reflejando una buena generalización del modelo.
- **Correlación:** Una alta correlación entre las predicciones de la red y los valores reales valida la capacidad del modelo para aprender patrones relevantes.

Además, las predicciones fueron consistentes en todo el rango de las variables de diseño, demostrando la robustez del modelo.

7. Conclusiones

- **Estructura de las redes:** Se comprobó que redes no excesivamente complejas, en términos de número de neuronas y capas, ofrecen un equilibrio óptimo entre rendimiento y recursos computacionales. Estas redes son capaces de entrenarse rápidamente en un ordenador personal y producen resultados comparables a los de redes más complejas. En los puntos validados del *dataset* de entrenamiento, las redes mostraron una tendencia a predecir un SNR mayor al solicitado, lo que resulta positivo.
- **Capacidad de generalización:** Aunque las redes presentan menor rendimiento al predecir para puntos generados aleatoriamente, se observó que generando múltiples predicciones se mejora la probabilidad de obtener resultados satisfactorios.

- **Comparación con otros algoritmos:** Las redes neuronales destacan por su capacidad de predecir directamente las variables de diseño necesarias para cumplir con ciertas especificaciones, reduciendo el tiempo de predicción una vez entrenadas. Sin embargo, la generación del *dataset* y el entrenamiento requieren un tiempo considerable, entre 5 y 20 horas dependiendo del caso de estudio, superando el tiempo de ejecución de otros algoritmos, como el genético. Por tanto, las RNN son más ventajosas cuando se requiere realizar múltiples diseños.