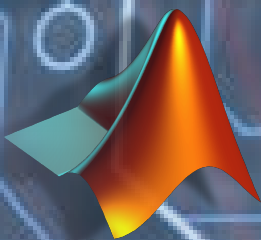


Juan Vorobioff
Santiago Cerrotta
Nicolas Eneas Morel
Ariel Amadio



Inteligencia Artificial y Redes Neuronales



**Fundamentos, Ejercicios
y
Aplicaciones con Python y Matlab**

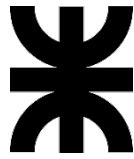


UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

CiN REUN

Red de Editoriales
de Universidades Nacionales
de la Argentina





Inteligencia Artificial y Redes Neuronales: Fundamentos, Ejercicios y Aplicaciones con Python y Matlab

Juan Vorobioff

Santiago Cerrotta

Nicolas Eneas Morel

Ariel Amadio

Universidad Tecnológica Nacional

Facultad Regional Buenos Aires

2022

Inteligencia Artificial y Redes Neuronales fundamentos, ejercicios y aplicaciones con Python y Matlab / Juan Vorobioff ... [et al.] ; editado por Fernando Cejas. - 1a ed. - Ciudad Autónoma de Buenos Aires : edUTecNe, 2022.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-4998-82-8

1. Inteligencia Artificial. I. Vorobioff, Juan. II. Cejas, Fernando, ed.

CDD 006.3



Universidad Tecnológica Nacional – República Argentina

Rector: Ing. Rubén Soro

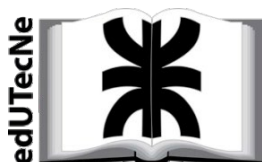
Vicerrector: Ing. Haroldo Avetta



Universidad Tecnológica Nacional – Facultad Regional Buenos Aires

Decano: Ing. Guillermo Oliveto

Vicedecano: Ing. Andrés Bursztyn



edUTecNe – Editorial de la Universidad Tecnológica Nacional

Coordinador General a cargo: Fernando H. Cejas

Queda hecho el depósito que marca la Ley N° 11.723

© edUTecNe, 2022

Sarmiento 440, Piso 6 (C1041AAJ)

Buenos Aires, República Argentina

Publicado Argentina – Published in Argentina



ISBN 978-987-4998-82-8



Reservados todos los derechos. No se permite la reproducción total o parcial de esta obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de los titulares del copyright. La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.

A nuestras familias.

Agradecimientos

Depto. de Micro y Nanotecnología - Comisión Nacional de Energía Atómica (CNEA)

Facultad Regional Buenos Aires - Universidad Tecnológica Nacional (FRBA – UTN)

Facultad Regional Delta - Universidad Tecnológica Nacional (FRD – UTN)

Facultad Regional General Pacheco - Universidad Tecnológica Nacional (FRGP – UTN)

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Introducción a Redes Neuronales

**Algoritmos de entrenamiento de redes: propagación hacia atrás
y otros algoritmos**

Redes de funciones de base radial (RBF)

**Reconocimiento estadístico de patrones, aprendizaje
automático y redes neuronales**

Redes Neuronales con Mapas Autoorganizados (SOM)

Redes Neuronales Dinámicas

Redes Neuronales Convolucionales

Inteligencia Artificial en Dispositivos Móviles y de IoT

Visión Artificial

Aplicaciones con métodos combinados de Inteligencia Artificial

**Aplicaciones. Procesado de señales interferométricas con redes
neuronales: Estimando frecuencias**

**Redes Neuronales aplicadas a la detección de picos de glucosa
después de las comidas en series temporales de pacientes
diabéticos tipo II**

Lista de ejercicios

Epílogo

Contenidos

Capítulo XI -Aplicaciones. Procesado de señales interferométricas con redes neuronales: Estimando frecuencias

A lo largo de este capítulo se muestra el uso de redes neuronales artificiales para el análisis de señales de interferometría de baja coherencia, donde es de particular interés la determinación de frecuencias. Se estudian diferentes arquitecturas de redes neuronales, analizando su comportamiento y características principales de entrenamiento, los tiempos de entrenamiento para diferentes cantidades de datos de entrenamiento, la dependencia de la resolución en función de los datos de entrenamientos y parámetros de la red. Por último, se comparan los resultados obtenidos utilizando redes con el algoritmo tradicional de la fft. Esta innovadora forma de procesar señales abre las puertas a nuevas alternativas la hora de calcular la frecuencia de cualquier tipo de señales unidimensionales.

Introducción

Las Redes Neuronales Artificiales se han desarrollado y crecido de una manera considerable en los últimos años captando la atención no solo de la academia, sino también de disciplinas que van desde la medicina hasta las finanzas, interviniendo en problemáticas teóricas y también experimentales, que involucran a instituciones públicas como privadas y regiones locales como globales.

Siempre es bueno recordar, y volver a disfrutar, la idea que dio inicio al surgimiento de las Redes Neuronales Artificiales. ¿Alguna vez pensaron en cómo un ser humano aprende a jugar al ping pong o identificaron los pasos que va dando un recién nacido hasta que consigue hablar fluidamente? Pocas son las personas que conocen las ecuaciones de movimiento de cuerpos esféricos inmersos en fluidos, que saben estimar la trayectoria de un movimiento paraboloidal en fracciones de segundo, calcular el ángulo de impacto y velocidad necesaria de la paleta para colocar la pelota justo en el ángulo opuesto de la mesa. Menos aún son los y las niñas que conocen la anatomía de las cuerdas vocales, la construcción semántica de una oración unimembre o que realizan la transformada de Fourier para identificar las componentes espectrales propias de sus seres queridos. O que memorizan las veintitantas letras del abecedario y las 2000 palabras más usadas del idioma español. Sin embargo, muchas personas juegan al ping pong y aun muchas más saben hablar. Dado que el ser humano es capaz resolver de una manera sencilla, rápida y cotidiana problemáticas que ni la tecnología más innovadora consigue y además lo consigue sin la necesidad de solucionar ninguna ecuación diferencial. ¿Por qué en vez de formular teorías ultra racionales y modelos matemáticos con tantas hipótesis y descripciones algebraicas simplemente intentamos copiarnos (o mejor aún inspirarnos) del motor central de aprendizaje humano o también llamado Cerebro? Neuronas interconectadas con otras neuronas que reciben estímulos, se activan y transmiten nuevas respuestas que serán estímulos de otras conexiones neuronales y que ante sucesivas experiencias de aprendizaje logran moderar los estímulos para conseguir el objetivo deseado. He aquí las bases fundacionales de la Inteligencia Artificial y el Machine Learning.

Estas no son técnicas nuevas, ya existen hace varias décadas, pero los avances progresivos en esta área sumado al aumento de la capacidad humana de generar, almacenar y transmitir grandes cantidades de información hicieron que estos últimos años se vuelva a realzar y valorizar su gran potencialidad. En los últimos 5 años la cantidad de publicaciones y relevancia en Inteligencia Artificial aumenta a pasos agigantados, según Google Scholar, dentro de las 10 publicaciones más citadas del año 2019 de todas las áreas más de un tercio son acerca de Redes Neuronales Artificiales. Dato que aumenta a más de un 40% para el año 2020. Además, en estos últimos años constantemente nuevas disciplinas están incorporando el uso de Inteligencia Artificial y Machine Learning. Algunas de sus finalidades son: mejorar su efectividad, automatizar procesos, detectar anomalías, reconocer comportamientos, manipular grandes cantidades de información, entre otros.

El uso de Redes Neuronales Artificiales se suma, sin lugar a duda, al paradigma ya instalado de la interdisciplinariedad. Vivimos en una sociedad con un gran nivel de desarrollo y complejidad y a un ritmo acelerado, cada área del conocimiento a calado bien profundo en sus temáticas. El desafío disruptivo actual no pasa tanto por descubrir algo nuevo si no en adaptar el conocimiento existente de una disciplina a solucionar una problemática de otra. A entrecruzar los saberes y las necesidades para desarrollar nuevos enfoques teóricos y técnicos.

Ya hace varias décadas que nos encontramos en la época de la Información, para tomar prácticamente cualquier decisión todas las áreas del conocimiento tienen que poder detectar (sea analógico, digital o de alguna otra manera), manipular, transformar y analizar datos para obtener información de valor. Tan importante es esta necesidad que, ya hace varios años, el procesamiento de señales se ha constituido en un área en sí y las redes neuronales en una de las herramientas de más desarrollo en la actualidad. Una característica muy interesante para analizar una señal es la velocidad con que se repite, es decir su frecuencia. Concepto central que nos ayuda a describir gran parte del universo que nos rodea. Nos permite explicar, por ejemplo: los movimientos oscilatorios que van desde resortes a modos vibracionales de moléculas, fenómenos ondulatorios de una ola en un estanque como de la luz viajando en el espacio, la transferencia de más de millones de símbolos por segundo a través de fibras ópticas, hasta incluso evidenciar la naturaleza cuántica del mundo de lo pequeño. Por este motivo estimar las frecuencias existentes en una señal es muy importante y hay mucho trabajo realizado sobre el tema.

La interferometría es un conjunto de técnicas que basan en el principio de interferencia para la medición de diferentes fenómenos físicos, básicamente se centra en estudiar las maneras en que se combinan diferentes ondas electromagnéticas. Son muy utilizadas en aplicaciones que van desde la astronomía hasta la física nuclear pasando por metrología y espectroscopia entre otras. El resultado de superponer dos o más ondas que interfieren entre ellas es un patrón oscilante de intensidades, donde generalmente es en la frecuencia que se codifica la información de relevancia.

Lo que se busca en este capítulo es justamente utilizar Redes Neuronales Artificiales para reemplazar herramientas convencionales del procesamiento de señales para aplicarlo en el área de interferometría de ondas electromagnéticas, puntualmente a la luz.

Interferometría de baja coherencia

La interferometría de baja coherencia (LCI por sus siglas en inglés, low coherence interferometry), es una técnica de metrología óptica que permite la medición de perfilometrías, tomografías, vibraciones, velocidades, entre otras, utilizando como elemento sensor a la luz (Huang et al., 1991) (Walecki et al., 2005). Para ello se utiliza típicamente un dispositivo conocido como interferómetro de Michelson, se puede destacar que existe una gran cantidad de interferómetros que se pueden utilizar, pero el de Michelson es el más característico. Una característica importante

es que la fuente de luz que se utiliza es de baja coherencia, esto implica que la fuente posee un ancho de banda grande (mayor a los 20 nm), si bien existen numerosas fuentes de luz que cumplen con este propósito la tecnología más usada es la de los leds super luminiscentes (SLD). En particular el sistema de detección que se utiliza es un espectrómetro que permite obtener la señal de interferencia.

En la Fig. 174 se muestra una configuración típica de un interferómetro de tipo Michelson, donde se puede ver esquemáticamente como un haz de luz proveniente de una fuente de baja coherencia se dirige al divisor de haz (beam splitter, BS), se asume que el BS divide la potencia óptica incidente de manera uniforme en brazos de muestra y de referencia, aunque muchos diseños prácticos de sistemas OCT aprovechan la división de potencia desequilibrada. A partir del BS un haz de luz se dirige a la muestra en estudio y el otro a un espejo de referencia R. El haz del brazo de muestra incide sobre un mecanismo de escaneo y óptica de enfoque que permite realizar un escaneo de la muestra, estos sistemas se controlan típicamente con una computadora. Se han desarrollado muchos sistemas de escaneo especializados para aplicaciones de imágenes de LCI en microscopía, oftalmología, endoscopía y en ensayos no destructivos como se indica en (M. R. Wang, n.d.)(Grulkowski et al., 2013) (Walecki et al., 2006).

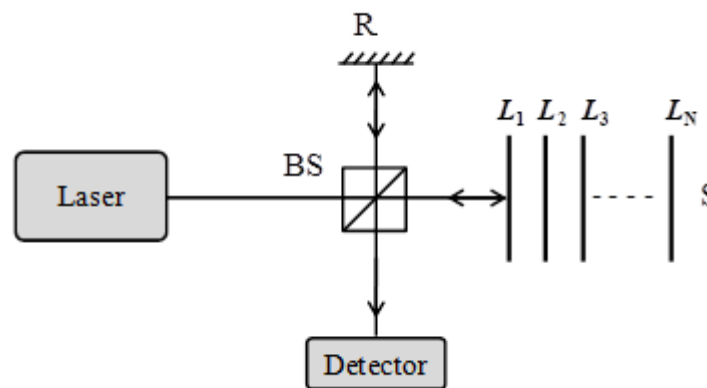


Fig. 174 – Esquema de una configuración experimental de interferencia de baja coherencia, basado en un interferómetro de Michelson

La luz reflejada de la muestra se redirige a través del mismo sistema óptico de la rama donde se encuentra la muestra, puede tener múltiples capas (L_1, \dots, L_N), esto implica la capacidad de la técnica para realizar topografías y tomografías. Las reflexiones provenientes de la muestra y del espejo de referencia vuelven por el mismo camino hacia el BS, que redirige a los haces hacia el detector. Los haces después del BS se combinan generando la interferencia que se detecta con el espectrómetro. Como la luz recorre el camino tanto a la ida como a la vuelta y que además puede ser en distintos materiales, se suele usar el término diferencia de camino óptico DCO para referirse a ella. Que no es ni más ni menos que dos veces la distancia de interés multiplicado por el índice de refracción n del material $DCO = 2n(z_r - z_{L_i})$.

Se debe destacar que la cámara o array lineal en el espectrómetro detecta la irradiancia de la señal de interferencia que es función de los campos.

$$I = \frac{c}{\mu_0} \langle \mathbf{B}^2 \rangle_T \quad (11.1)$$

$$I = \varepsilon_0 c \langle \mathbf{E}^2 \rangle_T \quad (11.2)$$

Típicamente la expresión general de las señales obtenidas en el detector tiene la siguiente representación

$$\begin{aligned} i_D(k) = & \frac{\rho}{4} [s(k)(r_r + r_{s1} + r_{s2} + \dots)] \\ & + \frac{\rho}{4} \left[s(k) \sum_{n=1}^N \sqrt{r_r r_{sn}} (e^{j2k(z_r - z_{sn})} + e^{-j2k(z_r - z_{sn})}) \right] \\ & + \frac{\rho}{4} \left[s(k) \sum_{n \neq m=1}^N \sqrt{r_r r_{sm}} (e^{j2k(z_{sn} - z_{sm})} + e^{-j2k(z_{sn} - z_{sm})}) \right] \end{aligned} \quad (11.3)$$

Donde $s(k)$, es la forma espectral de la fuente de luz. En general se utilizan espectros de fuente de luz de forma Gaussiana por su conveniencia en el modelado y porque se aproxima a la forma de las fuentes de luz reales y también tiene propiedades útiles de la transformada de Fourier. La función gaussiana normalizada $s(k)$ y su transformada de Fourier inversa $S(z)$ están dadas por:

$$s(k) = \frac{1}{\Delta k \sqrt{\pi}} e^{-\left[\frac{(k-k_0)}{\Delta k}\right]^2} \leftrightarrow S(z) = e^{-z^2 \Delta k^2} \quad (11.4)$$

Aquí, k_0 representa el número de onda central del espectro de la fuente de luz y Δk representa su ancho de banda espectral, correspondiente a la mitad del ancho del espectro a 1/e de su máximo.

Al realizar la transformada de Fourier de las señales $i_D(k)$, obtenemos la señal $I_D(z)$, donde se puede apreciar las distintas características de la señal de interferencia.

$$\begin{aligned} i_D(z) = & \frac{\rho}{8} [S(z) \otimes \delta(z)(r_r + r_{s1} + r_{s2} + \dots)] \\ & + \frac{\rho}{4} \left[S(z) \otimes \sum_{n=1}^N \sqrt{r_r r_{sn}} (\delta(z \pm 2(z_r - z_{sn}))) \right] \\ & + \frac{\rho}{8} \left[S(z) \otimes \sum_{n \neq m=1}^N \sqrt{r_r r_{sm}} (\delta(z \pm 2(z_{sn} - z_{sm}))) \right] \end{aligned} \quad (11.5)$$

El primer término de la ecuación (11.5) se denomina componente "constante" o "DC". Este es el componente más grande de la corriente del detector si la reflectividad de referencia domina la reflectividad de la muestra. El segundo término de la ecuación (11.5) es la componente de correlación cruzada para cada reflexión de muestra, que depende tanto del número de onda de la fuente de luz como de la diferencia de longitud de trayectoria entre el brazo de referencia y las interfaces de la muestra donde se generan reflexiones. Este es el componente deseado para la obtención de imágenes OCT. Dado que estos componentes son proporcionales a la raíz cuadrada de las reflectividades de la muestra, normalmente son más pequeños que la componente DC. Sin embargo, la dependencia de la raíz cuadrada representa un factor de ganancia logarítmica importante sobre la detección directa de reflexiones de la muestra. El tercer término de la ecuación (11.5) son los términos de autocorrelación que representan la interferencia que ocurre entre las diferentes interfaces de muestra. Dado que los términos de autocorrelación dependen linealmente de la reflectividad de las diferentes interfaces de la muestra, serán pequeños en comparación con los otros.

Si bien las señales de interferencia $i_D(k)$ e $i_D(z)$ ilustran el principio fundamental de la LCI, en implementaciones prácticas los dispositivos de detección presentan varios factores adicionales que deben ser tomados en cuenta, como se detalla en (Dorrer et al., 2000). La señal de interferencia se obtiene mediante instrumentación que tiene limitaciones del mundo real, y normalmente se adquieren mediante una operación de muestreo para el cálculo rápido de la señal digital de su transformada de Fourier inversa.

Lo primero para tener en cuenta son aspectos relacionados a artefactos introducidos por el sistema de detección. Uno de ellos es la resolución propia del espectrómetro aquí denotada por Δk , que es la resolución espectral del espectrómetro (incluido el espaciado finito de los píxeles del CCD), en número de onda.

El efecto de la resolución espectral finita se puede modelar convolucionando la señal de interferencia ideal con una función gaussiana donde la mitad de su ancho máximo es Δk_d que interpretamos como la resolución espectral con el criterio ancho a mitad de altura (FWHM), que representa la función de dispersión del espectrómetro (PSF) y una función seno cardinal (sinc) que aporta el efecto del ancho finito de cada pixel, se dimensiona el ancho de cada pixel como Δk_p . Estos efectos generan una atenuación de la visibilidad de la señal a medida que se incrementa la DCO, este efecto se conoce como Fall-Off. A través de la propiedad de convolución, $i(k)$ se relaciona con el Fall-Off, en el espacio transformado la señal $i_D(z)$ se multiplica por un factor de Fall-Off, que representa una pérdida de sensibilidad para valores de z grandes y cuya forma está dada por la transformada de Fourier inversa del factor de resolución.

$$I_D(z)e^{-\left(\frac{z}{\Delta z_d}\right)^2} \text{sinc} \left(\frac{z}{\Delta z_p} \right) \leftrightarrow i_D(k) \otimes e^{-\left(\frac{k^2}{Ra}\right)^2} \otimes \prod \left(\frac{k}{\Delta k_p} \right) \quad (11.6)$$

Donde a es el tamaño del spot de luz para cada k y R de dispersión lineal recíproca. R permite relacionar la variable k con la coordenada espacial sobre el array de pixels (eje y , distancia sobre el array CCD, tomando como origen el centro del primer pixel). En este trabajo se supone que esta relación es lineal y por lo tanto R es constante y se puede estimar como $R = \Delta k_p / \Delta y$, siendo Δy el ancho espacial del pixel. Para el caso en que el sistema de dispersión es una red de difracción el valor de R depende de la distancia focal f , la longitud focal del espectrómetro, θ_r el ángulo de difracción y m el orden de difracción de la red:

$$\left. \frac{\partial k}{\partial y} \right|_{k_0} = \frac{d \cos(\theta_r) k_0^2}{2\pi m f} \quad (11.7)$$

La segunda consideración importante en el procesamiento del mundo real de datos en la técnica de LCI en el Dominio de las Frecuencias es que la detección implica muestrear la señal de interferencia $i_D(k)$, como el muestreo se realiza por medio de una cámara o array lineal la trama muestreada es finita, esto se puede representar por medio de una señal rectangular la cual posee un ancho igual al ancho de la cámara o array lineal (Δk_a) en términos de k .

$$\begin{aligned} & \left[I_D(z) e^{-\left(\frac{z}{\Delta z_a}\right)^2} \operatorname{sinc}\left(\frac{z}{\Delta z_p}\right) \right] \otimes \operatorname{sinc}\left(\frac{z}{\Delta z_a}\right) \\ & \leftrightarrow \left[i_D(k) \otimes e^{-\left(\frac{k^2}{Ra}\right)^2} \otimes \prod\left(\frac{k}{\Delta k_p}\right) \right] \prod\left(\frac{k - k_0}{\Delta k_a}\right) \end{aligned} \quad (11.8)$$

Un tercer elemento para tener en cuenta es que, la señal de interferencia $i_D(k)$ se muestra de manera no lineal. Esto es debido al diseño del espectrómetro. Quedando entonces la señal con la siguiente forma:

$$i(k) = \left[i_D(k) \otimes e^{-\left(\frac{k^2}{Ra}\right)^2} \otimes \prod\left(\frac{k}{\Delta k_p}\right) \right] \prod\left(\frac{k - k_0}{\Delta k_a}\right) \quad (11.9)$$

$$I(z) = \left[I_D(z) e^{-\left(\frac{z}{\Delta z_a}\right)^2} \operatorname{sinc}\left(\frac{z}{\Delta z_p}\right) \right] \otimes \operatorname{sinc}\left(\frac{z}{\Delta z_a}\right) \quad (11.10)$$

En términos completos las expresiones anteriores quedan de las siguientes formas:

$$\begin{aligned}
 i(k) = & \left[\frac{\rho}{4} [s(k)(r_r + r_{s1} + r_{s2} + \dots)] \dots \right. \\
 & + \frac{\rho}{4} \left[s(k) \sum_{n=1}^N \sqrt{r_r r_{sn}} (e^{j2k(z_r - z_{sn})} + e^{-j2k(z_r - z_{sn})}) \right] \dots \\
 & + \frac{\rho}{4} \left[s(k) \sum_{n \neq m=1}^N \sqrt{r_r r_{sm}} (e^{j2k(z_{sn} - z_{sm})} + e^{-j2k(z_{sn} - z_{sm})}) \right] \dots \left. \right] \dots \\
 & \otimes \left[e^{-\left(\frac{k^2}{Ra}\right)^2} \prod \left(\frac{k}{\Delta k_p} \right) \right] \prod \left(\frac{k - k_0}{\Delta k_a} \right)
 \end{aligned} \tag{11.11}$$

$$\begin{aligned}
 I(z) = & \left[\frac{\rho}{8} [S(z) \otimes \delta(z)(r_r + r_{s1} + r_{s2} + \dots)] \dots \right. \\
 & + \frac{\rho}{4} \left[S(z) \otimes \sum_{n=1}^N \sqrt{r_r r_{sn}} (\delta(z \pm 2(z_r - z_{sn}))) \right] \dots \\
 & + \frac{\rho}{8} \left[S(z) \otimes \sum_{n \neq m=1}^N \sqrt{r_r r_{sm}} (\delta(z \pm 2(z_{sn} - z_{sm}))) \right] \dots \left. \right] \dots \\
 & \left[e^{-\left(\frac{z}{\Delta z_d}\right)^2} \text{sinc} \left(\frac{z}{\Delta z_p} \right) \right] \otimes \text{sinc} \left(\frac{z}{\Delta z_a} \right)
 \end{aligned} \tag{11.12}$$

De esta expresión se desprenden las características relevantes de la técnica como la resolución y el rango dinámico. La resolución dependerá claramente de la fuente de luz utilizada, que se puede expresar en términos de k o de λ .

$$\text{Resolucion} = \frac{2\sqrt{\ln(2)}}{\Delta k} = \frac{2\sqrt{\ln(2)}}{\pi} \frac{\lambda_0^2}{\Delta \lambda} \tag{11.13}$$

El rango dinámico es un poco más complejo ya que depende de la cantidad de muestras que se puedan adquirir (cantidad de píxeles), y de diferentes elementos que componen el detector (tamaños de los píxeles, red de difracción, entre lo más importantes), pero típicamente el rango de medición no supera los 3 a 5 mm.

Caso de estudio: Perfilometría/topografía

Una de las aplicaciones más interesantes son la perfilometría/topografía y la medición de distancias, la medición por LCI permite la detección de luz a alta sensibilidad, limitada solo por el ruido de disparo. Diversas aplicaciones industriales se analizan en (Dufour, 2006) (Cerrotta et al., 2015)(Morel et al., 2016). Las mediciones se pueden realizar sin contacto físico con la muestra bajo investigación. Se consigue una resolución de profundidad de unos pocos micrómetros incluso en muestras muy difusivas, mediante la supresión de la luz dispersa.

En los últimos años, la LCI se ha desarrollado vigorosamente hasta convertirse en una poderosa herramienta. La mayor complejidad y los altos volúmenes de producción en la fabricación moderna de estructuras exigen métodos de inspección, control de calidad y metrología adecuados. Las desviaciones geométricas de las estructuras y las inhomogeneidades de los materiales pueden provocar errores sistemáticos durante todos los pasos de la fabricación, lo que podría afectar la calidad y la funcionalidad de grandes volúmenes de un producto. Entre los campos de aplicación y solo por mencionar algunos de relevancia tecnológica se encuentran aplicaciones como semiconductores de potencia, energía fotovoltaica, detección de defectos en MEMS y electrónica impresa. La metrología que acompaña a la producción se vuelve vital.

En particular en el área metal mecánica, autopartista o siderúrgica, el estudio de las desviaciones no deseadas en parámetros como la rugosidad de la superficie y la topografía pueden causar problemas importantes. Esta técnica además es adecuada para ser aplicada en línea de producción.

Con la finalidad de obtener una mejor comprensión del comportamiento de las señales al realizar topografías mediante LCI, teniendo en cuenta que ahora se supone una única interfase de la muestra de interés, se reescribirán las ecuaciones (11.11) y (11.12) quedando:

$$i(k) = \frac{\rho}{4} \left[s(k) \left[(r_r + r_s) + [\sqrt{r_r r_s} (e^{j2k(z_r - z_s)} + e^{-j2k(z_r - z_s)})] \right] \prod \left(\frac{k - k_0}{\Delta k_a} \right) \right] \otimes \left[e^{-\left(\frac{k^2}{Ra}\right)^2} \prod \left(\frac{k}{\Delta k_p} \right) \right] \quad (11.14)$$

$$I(z) = \frac{\rho}{8} \left[S(z) \otimes \left[(r_r + r_s) \delta(z) + [\sqrt{r_r r_s} (\delta(z \pm 2(z_r - z_s)))] \right] \right] \otimes \text{sinc} \left(\frac{z}{\Delta z_a} \right) \left[e^{-\left(\frac{z}{\Delta z_d}\right)^2} \text{sinc} \left(\frac{z}{\Delta z_p} \right) \right] \quad (11.15)$$

Las cuales podemos simplificar si tomamos la parte real de la señal $i(k)$, quedándonos:

$$i(k) = \frac{\rho}{4} \left[s(k) \left[[(r_r + r_s)] + [\sqrt{r_r r_s} \cos(k2(z_r - z_s))] \right] \prod \left(\frac{k - k_0}{\Delta k_a} \right) \right] \otimes \left[e^{-\left(\frac{k^2}{Ra}\right)^2} \prod \left(\frac{k}{\Delta k_p} \right) \right] \quad (11.16)$$

En la Fig. 175 se puede observar una típica señal de LCI, al realizar topografías, donde se puede apreciar como la visibilidad o pico a pico de la señal no es máximo, esto es debido a dos factores, el primero es la relación entre la reflectividad r_r y r_s , el segundo son los artefactos introducidos por el sistema de detección. Ademas se ve como la forma espectral de la fuente de luz utilizada genera una envolvente sobre la señal, que en este caso se utilizó la forma espectral doble gaussiana, vale aclarar que los perfiles espectrales de las fuentes de luz pueden tener diferentes formas.

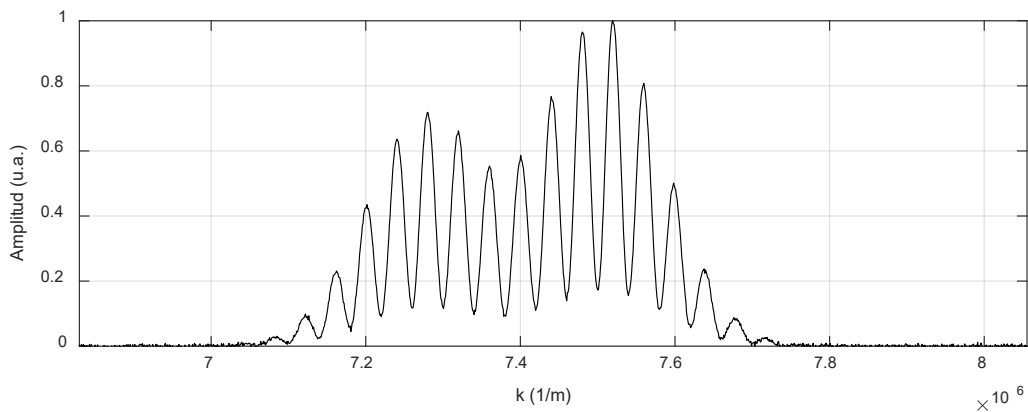


Fig. 175 – Señal típica de LCI, donde se puede percibir la oscilación de la amplitud.

Recordemos que el espectrómetro realiza un muestreo no lineal en función de k , por tal motivo después de la etapa de adquisición se realiza un remuestreo y conversión de λ a k , ver Fig. 176. La siguiente etapa realiza un filtrado de la componente de continua, luego se determina la envolvente que representa la forma espectral de la fuente de luz, luego se procede a realizar una normalización, un recorte de la señal en función de los puntos útiles, para poder realizar la transformada rápida de Fourier (fft) y finalmente la determinación de la frecuencia espacial o posición del pico, como se comenta en el artículo (Ali & Parlapalli, 2010) .

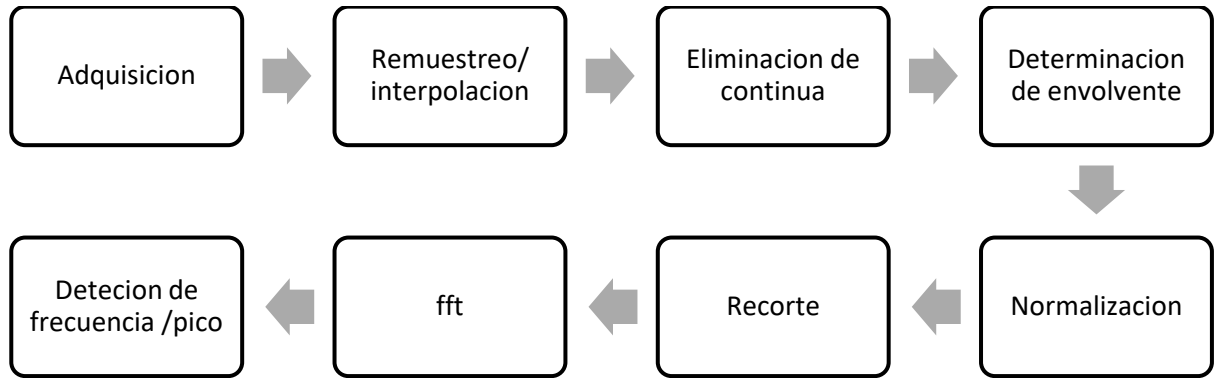


Fig. 176 – Esquema de las etapas de preprocesado convencional de una señal de LCI.

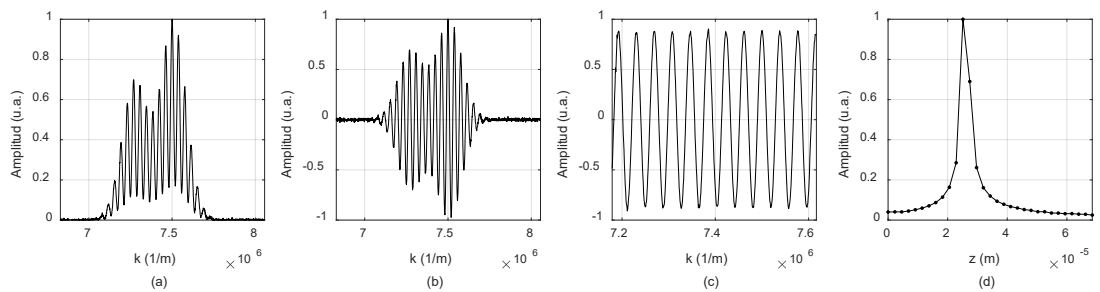


Fig. 177 – Preprocesado convencional, primero se detecta la señal de LCI, luego se filtra las frecuencias bajas, se identifica la envolvente y divide para finalmente realizar la transformada rápida de Fourier de la señal oscilante.

Al realizar la fft la resolución espectral queda determinada por el número de muestras de la trama de la señal original, como en el caso de la Fig. 177, esto puede generar una limitación en la detección del máximo del pico y por lo tanto un error en la determinación de la posición. Este efecto se puede corregir utilizando técnicas como zero padding. En particular la técnica de zero padding implica un agregado de datos a la trama original por lo tanto se trabaja con vectores más grandes incrementando la carga computacional. Una alternativa es utilizar la transformada chirp de Fourier, en este caso se permite seleccionar el rango de frecuencias donde se desea obtener la transformación y se indica la cantidad de puntos deseados entre ese par de frecuencias, esto mejora la resolución, pero imposibilita realizar la anti transformada.

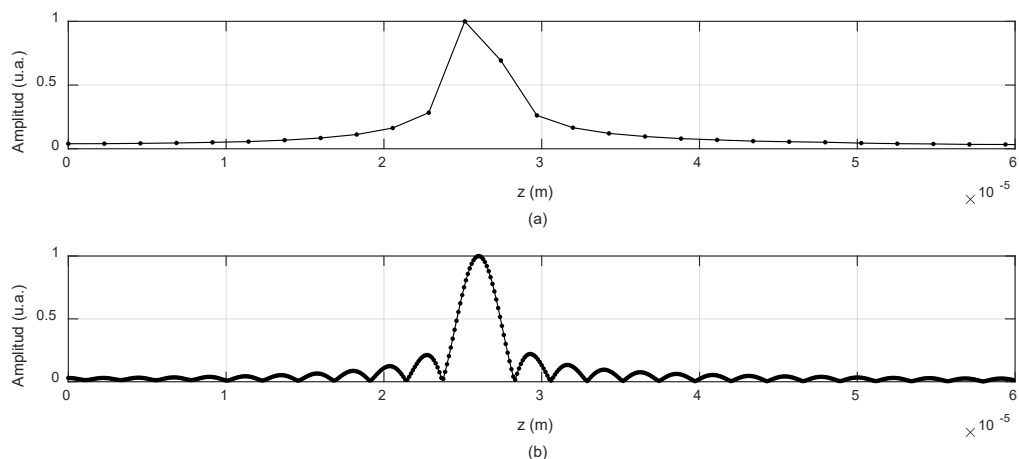


Fig. 178 – Comparación de la transformada de Fourier de una señal de LCI para menor (arriba) y mayor (abajo) número de muestras. Tomando mayor número de muestras se puede identificar la posición del máximo con mejor precisión

Al aplicar la transformada chirp se evidencian adicionalmente los efectos de la cantidad de muestras del frame, como se puede observar en la Fig. 178.

Estimación de frecuencia

Como se vio anteriormente, la determinación de la posición del pico de la transformada de Fourier indica la información deseada, claramente esa información también se puede obtener al medir los periodos de la señal en el espacio k , por lo cual el problema central que se desea resolver es la estimación de frecuencias.

La estimación de la frecuencia de una onda sinusoidal ruidosa con precisión ha sido uno de los principales problemas en el campo del procesamiento de señales y las comunicaciones, debido a sus amplias aplicaciones que incluyen sistemas de energía (Routray et al., 2002), comunicaciones (Classen & Meyr, 1994), radar (Orović et al., 2010)(Bamler, 1991)(Liu et al., 2012), vibraciones mecánicas, acústica, ultra sonido, medicina, entre los más destacados.

Se han propuesto muchas técnicas teóricas para resolver este problema; los ejemplos incluyen la transformada discreta de Fourier (Belega & Dallet, 2008), los métodos de mínimos cuadrados (Yardibi et al., 2010) y los lazos de fase bloqueada (Martínez-Montejano et al., 2014). Todos los métodos propuestos se centran en la velocidad y precisión de la estimación.

El uso de redes neuronales ya se ha introducido en el campo del procesamiento de señales (Deng, 2008) (Tang & Eliasmith, 2010) (LeCun et al., 2007) y en temáticas asociadas a frecuencias por ejemplo para encontrar la estimación de máscara de relación ideal para filtrar el ruido de un espectrograma, monitoreo de frecuencia en tiempo real (Lai et al., 1999), detección de canal en sistemas de multiplicación por división de frecuencia ortogonal (Gao et al., 2018), y en la obtención de retroalimentación de información de estado de canal de sistemas masivos de múltiples entradas y múltiples salidas (T. Wang et al., 2019). Pero hasta la actualidad, son pocos y recientes los trabajos que se centraron en estimar directamente la frecuencia de señales mediante redes neuronales. En (Sajedian & Rho, 2019) se encuentra que mediante una red neuronal sencilla densamente conectada se pueden estimar frecuencias de señales senoidales con presencia de ruido. Esta técnica se analiza en mayor profundidad en (Almayyali & Hussain, 2021) y se hace una comparación con los trabajos existentes tales como (Xu & Shimodaira, 2019) en el cual se estima la frecuencia fundamental de una

señal y en (Chen et al., 2019) donde se estiman señales linealmente moduladas en frecuencias mediante redes convolucionales.

Otra ventaja es que las técnicas que usan la transformada de Fourier discreta (DFT) requieren el uso de una aritmética de valor complejo, mientras que los sistemas de redes neuronales usan solo una aritmética de valor real. La implementación de una DFT para grandes cantidades de datos es una tarea muy complicada para los diseños de hardware y software a pesar de la eficiencia significativa proporcionada a través de las últimas versiones del algoritmo de transformada rápida de Fourier, que reduce el número de órdenes de multiplicaciones (complejas) de N^2 a $N \log(N)$ donde N es el número de muestras.

Es claro que se puede entrenar a una red neuronal para realizar la transformada de Fourier, como se destaca en (Rosemarie, 2008), pero no sería el método más eficiente, ya que como se mencionó los algoritmos existentes tienen gran eficiencia y flexibilidad. La idea es entonces hacer que la red neuronal aprenda a estimar la frecuencia, generando a partir de su aprendizaje su propio método de estimación.

Consideraciones generales

Se toma como elemento de análisis la señal descrita en la ecuación (11.16) que describe la intensidad de la señal de interferencia detectada por el espectrómetro para la condición de perfilometría.

Sin lugar a duda el éxito de la implementación de una Red Neuronal Artificial (de ahora en más la abreviaremos como RN sin la necesidad de explicitar que son artificiales) depende de la cantidad de datos que tengamos para entrenar. Los datos para entrenar la RN se pueden obtener, simplificando un poco, por dos caminos, el primero por la medición directa, en particular este es un proceso bastante costoso en términos de tiempo, pero con la ventaja de que los datos son reales. Un segundo camino es la simulación, que presenta la ventaja de ser un método más rápido para la obtención masiva de datos de entrenamiento, con una desventaja que es la utilización de datos no reales. Se debe tener en cuenta que las simulaciones permiten acelerar los procesos de diseño de la RN, y ser modificadas las veces necesarias para asemejarse a la señal real.

En este apartado se trabajó sobre señales simuladas, por este motivo es que se debe tener un buen entendimiento de los fenómenos físicos en estudio para garantizar que la programación de las señales genere simulaciones realistas. Se simuló señales con espesores z_s comprendidos entre $100 \mu\text{m}$ a $1000 \mu\text{m}$ y con z_s comprendidos entre $30 \mu\text{m}$ a $500 \mu\text{m}$ dando exactamente el doble para las DCO, considerando la forma espectral de la fuente de luz gaussiana centrada en 850 nm y con un ancho de 50 nm . Calculando mediante la ecuación (11.13) se obtiene una resolución de $8 \mu\text{m}$.

Un tema que debe estar en consideración son las capacidades del hardware disponible. Esto juega un rol crucial en los tiempos que lleva generar las simulaciones de las señales, los entrenamientos y ejecuciones de las RN. Las operaciones presentes en este capítulo fueron realizadas en una computadora de 2 núcleos y procesador de 2.20 GHz Intel(R) Core (TM) con 8 GB de RAM. Los diseños y entrenamiento de las RN que se proponen a lo largo del capítulo fueron pensados para que puedan ser realizados en una computadora promedio personal y que el proceso no demande más de 24 hs .

El diseño, entrenamiento y manipulación de las RNs que se presentan fueron realizados en Matlab. Fácilmente pueden ser extendidos a Python utilizando los paquetes de TensorFlow y Keras.

Para medir el desempeño de las RN implementadas se usó el error cuadrático medio RMSE del subconjunto de testeo entre los valores reales Y_{test} y los predcidos Y_{pred} .

$$RMSE = \frac{1}{T_{test}} \sqrt{\sum_{i=1}^{T_{test}} (Y_{test_i} - Y_{pred_i})^2} \quad (11.17)$$

Vale la pena aclarar que al repetir el entrenamiento de una misma red con exactamente los mismos parámetros el valor de RMSE puede tener una leve variación debido a que los valores elegidos de testeo Y_{test} varían de manera aleatoria dentro de todo el conjunto total de datos. Otro parámetro que se utilizó para caracterizar la efectividad de las redes fue el error con respecto al peor valor predicho RSE_{max} sobre todo el conjunto Y (entrenamiento, validación y testeo).

$$RSE_{max} = \max \left\{ \sqrt{(Y - Y_{pred})^2} \right\} \quad (11.18)$$

Para poder comparar los tiempos de ejecución de las distintas RN con la forma convencional de procesar la señal se calcularon los tiempos que demora cada etapa, ver Tabla XVII.

Tabla XVII – Tiempo que demora cada etapa de preprocesada para la estimación de frecuencia para la técnica de LCI. Se marcan los órdenes de magnitud ya que *la precisión de los tiempos depende fuertemente del procesador.

Etapa de procesado	Tiempo [ms]*	Orden de magnitud
Eliminación de continua	1.10	ms
Determinación de envolvente	2.31	ms
Normalización	0.06	decenas de μ s
Recorte	1.52	ms
FFT	0.08	decenas de μ s
Detección de pico	0.01	decenas de μ s

Estimación paramétrica de frecuencias con redes neuronales

La estimación paramétrica se caracteriza por utilizar como elementos de entrada a la RN un conjunto de parámetros característicos de la señal, de manera tal que la cantidad de entradas a la red es pequeña, la conformación de la red neuronal se dice que es sencilla en cuanto al diseño de la cantidad de neuronas por capa, cantidad de capas y tipos de las neuronas. Esto claramente no implica que necesariamente sea más simple en cuanto al procesado de la información.

Esta propuesta consiste en obtener parámetros relevantes de nuestra señal de interés mediante un preprocesamiento de la señal. Cuanto más característicos y/o peculiares sean los parámetros elegidos de la señal mayor va a ser la efectividad en el entrenamiento de la red consiguiendo una óptima especificidad y selectividad en las predicciones. Es por este motivo que vale la pena tomarse un tiempo para pensar:

- ¿Cuáles son los parámetros más importantes?
- ¿Qué parámetros constituyen la huella digital de mi señal?
- ¿Cuáles son las características más significativas y de mayor variabilidad?

Las señales temporales tienen características típicas bien identificables como:

- La intensidad máxima.
- La distancia entre picos.
- El ancho de cada pico.
- El valor de la señal pico a pico.
- Valor de fase.
- Energía y/o potencia.

Se utiliza para esta descripción y a modo de ejemplo ilustrativo del análisis de señales que se desea analizar con una RN, una señal de oscilante típica de interferometría de baja coherencia, Ecuación (11.16). Se utilizaron 1000 simulaciones de señales típicas de LCI para valores de espesores z_s equiespaciados entre 100 a 1000 μm y se extrajeron cuatro parámetros de interés:

- La intensidad máxima global (I_{max}).
- El valor pico a pico (VPP) en las cercanías del máximo global.
- El promedio de separación entre dos máximos locales consecutivos (PSM).
- El promedio del ancho de cada uno de los máximos (PAM).

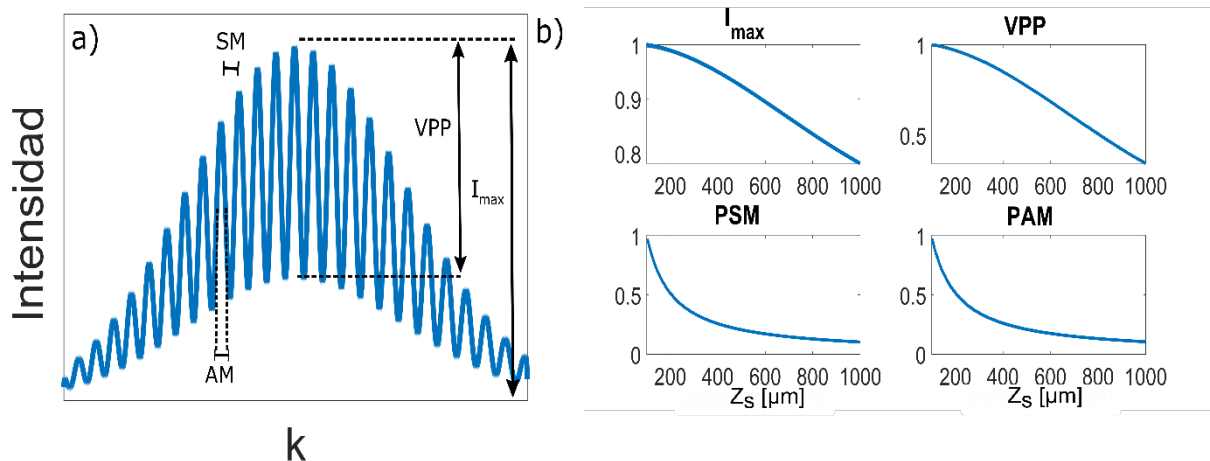


Fig. 179 – a) Esquema ilustrativo con los parámetros elegidos para caracterizar una señal típica de LCI. b) Valores de los parámetros para las 1000 simulaciones realizadas variando la distancia z_s .

Se diseñó una RN sencilla (o también llamada “shallow”) del tipo feedforward conformada por la capa de entrada que cuenta con tantas neuronas como parámetros de entrada (Input) deseados, seguido de una capa interna de 8 neuronas y una capa de salida de una única neurona para obtener una respuesta del tipo regresiva, Fig. 180.

Esto permite no solo usar a la RN para clasificar conjuntos de entradas (Inputs) dentro de ciertas categorías definidas, si no también obtener valores continuos como datos de salida (output) para cada conjunto de inputs. Notar que este trabajo lo realiza la función de activación de la neurona de la capa de salida. En general no se suele aclarar la conformación de la capa de entrada porque queda definida por el número de valores que conforman cada Input ni de la capa de salida ya que queda definida por el tipo y cantidad de salidas deseadas.

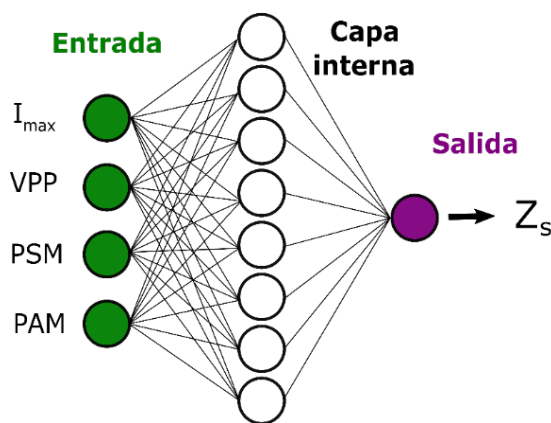


Fig. 180 – Estructura de la Red Neuronal de 8 neuronas densamente conectadas elegida para estimar frecuencia de señales parametrizadas. Se toman 4 valores de entrada y uno de salida.

Con un preprocesamiento de la señal se obtienen los 4 parámetros seleccionados para realizar el entrenamiento de la RN. Esta RN luego de ser entrenada puede obtener una estimación de la frecuencia de oscilación con una alta precisión.

Se utilizó el algoritmo de optimización para el entrenamiento de Levenberg-Marquardt. Para el diseño de la RN de 8 neuronas se usó la función *fitnet* que devuelve una función que se ajusta a la red neuronal con el tamaño de capa oculta seleccionado y se entrenó con la función *train(net, X, Y)* donde *net* es la red diseñada, *X* es una matriz con los datos de entrada de 7×1000 e *Y* es un vector de 1×1000 con los valores de frecuencia de la simulación requeridos para entrenar. El 72% de los datos se usó para el entrenamiento, el 18% para validación y el 10% para testeo. Se predijeron con la red entrenada los datos de testeo y se obtuvo un RMSE = $0.10 \mu\text{m}$.

Un análisis para este caso puede resultar poco redundante ya que parte del preprocesado es calcular la distancia entre máximos y la inversa de ese valor es la frecuencia en cuestión $z_s = 1/PSM$. En principio no sería necesario predecirlo con una RN. El objetivo era mostrar un ejemplo ilustrativo sencillo para comprender mejor la metodología de trabajo.

Por otro lado, puede resultar interesante usar RN en un caso similar, si se desea ponderar el valor de la frecuencia con algún (o algunos) parámetro asociado como el VPP o si se desea calibrar el detector a partir de mediciones conocidas. En general en el caso de LCI, y en otras aplicaciones también, la señal posee agregados y artefactos donde el uso de RN cobra una mayor utilidad y es uno de los temas que se trata en este capítulo.

En una señal real de LCI, como se comentó en la sección de Interferencia de Baja Coherencia, tanto los dispositivos de detección como la muestra en estudio pueden generar efectos no deseados o que modifican a la señal.

- a) La dispersión angular distinta para cada longitud de onda en la red de difracción sumada a la llegada de un frente de onda esférico (o gaussiano) a un detector con geometría plana, genera un muestreo no uniforme de la señal de LCI.
- b) Si el material en estudio tiene un índice de refracción $n(\lambda)$ que varía con la longitud de onda, genera un efecto dispersivo a la señal de LCI.

El efecto visible que estos efectos individuales generan en la señal de LCI es similar, en ambos casos, una deformación del pico en la transformada de Fourier, que genera errores en la detección de la frecuencia.

Para probar el método paramétrico se utilizó la señal de LCI con presencia de ruido. Teniendo en consideración que los métodos para corregir los efectos del muestreo no uniforme no son costosos y se encuentran bien establecidos, se utiliza entonces una señal de LCI considerando solo el efecto de índice de refracción $n(\lambda)$.

Se generó un conjunto de señales de LCI con el índice de refracción $n(\lambda)$ de un vidrio BK7 y se amplificó por 5 para evidenciar el efecto dispersivo, se puede observar en la Fig. 181.

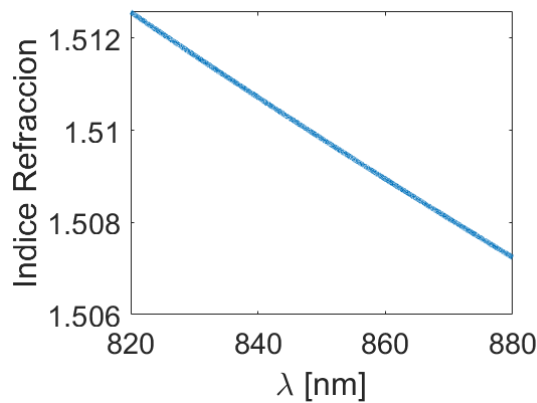


Fig. 181 – Índice de refracción en función de la longitud de onda utilizado en la simulación

Como la DCO aumenta al tomar valores de n mayores para iguales D , se disminuyó el efecto del Fall-off para que las señales simuladas tengan un VPP apreciable.

Al utilizar una señal más compleja, la obtención de los parámetros se vuelve más sofisticada al igual que el preprocesado de la señal de LCI para la obtención de los parámetros con precisión (filtrar espectralmente, calcular envolventes, agregar restricción para calcular mejor la posición de los máximos de las señales, etc.).

A diferencia del ejemplo anterior la señal de LCI presenta el efecto de dispersión por lo cual se agregaron entradas adicionales:

- La desviación estándar del ancho promedio de los picos.
- La desviación estándar de la separación entre máximos.

Como se observa en la Fig. 182 los valores I_{max} y V_{PP} tiene mayor variación debido a la incorporación de ruido.

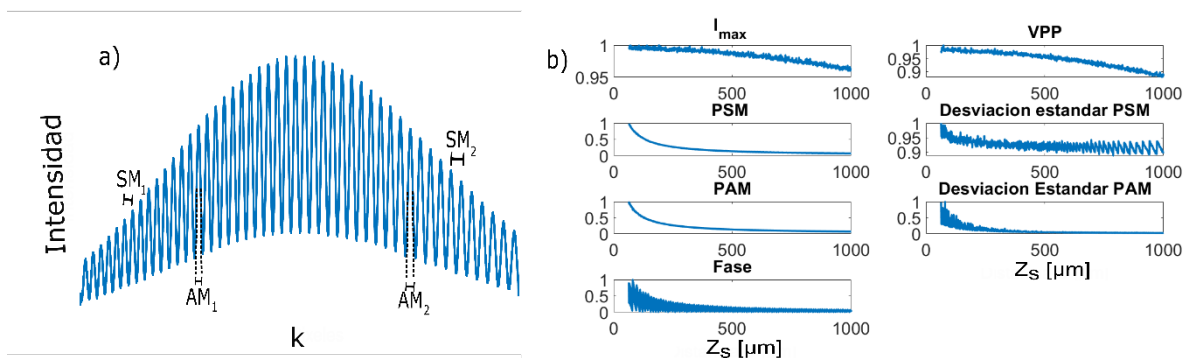


Fig. 182 – a) Esquema de una señal de LCI experimental con chirpeo, notar que $SM_1 < SM_2$ y $AM_1 < AM_2$ para una distancia z_s . b) Valores de los 7 parámetros para las 1000 simulaciones realizadas variando la Distancia z_s .

Se utilizó la misma configuración de RN que en la Fig. 180 y el mismo proceso de simulaciones, pero tomando ahora 7 valores de input. La función *train* tiene la ventaja que adecua

automáticamente la cantidad de entradas que requiere la RN a partir de las dimensiones de X y T prácticamente sin la necesidad de alterar el código.

Al entrenar la RN con las 1000 simulaciones se obtuvo un RMSE = 1.1 μm probando que sigue siendo una buena manera de estimar la frecuencia. Los tiempos de las simulaciones y preprocesado dependen fuertemente del tamaño del vector k que queda definido por Δk y km. Considerando un tamaño de k de 1×10.000 . Los tiempos de procesamiento son: 4.65 minutos para las simulaciones de la señal, 30 segundos para el preprocesado de todas las señales ($\frac{30\text{ s}}{1000} = 30\text{ ms}$ cada una), 16 segundos para el entrenamiento y 9 ms para la ejecución de un conjunto de entrada teniendo la red ya entrenada.

Es evidente que a medida que las señales se tornan más complejas y/o se intenta que la RN permita resolver un conjunto más grande de problemas el preprocesado necesario para la obtención de los parámetros se vuelve cada vez más importante. Adicionalmente, se debe destacar que cada material tiene un índice de refracción particular, por lo cual se deberá tener un conjunto de señales de entrenamiento y preprocesado adecuado para cada material. Se pudo probar que con una RN simple se puede obtener una muy buena estimación en frecuencia por debajo de la resolución esperada típica de LCI.

Estimación de frecuencias a partir de la señal completa

Las preguntas que se desean contestar son:

- ¿Se puede evitar la necesidad de seleccionar parámetros?
- ¿Se puede independizar del preprocesado
- ¿Cómo se puede lograr?
- ¿Qué tipo de RN se deben utilizar?

En esta sección se irán implementando diversas configuraciones de RNs para ir evitando en distintos grados las instancias de preprocesado sobre la señal original de LCI.

La primera diferencia con respecto al caso anterior es que ahora los datos de entrada a la RN serán los datos muestreados de la señal de LCI, de esta manera se evita la selección de parámetros que tienen que preestablecerse con antelación. En el caso de LCI la señal es detectada utilizando un espectrómetro como se describe en la sección anterior, el detector CCD de los espectrómetros pueden variar desde 512 a más de 12k pixeles, según los diferentes modelos presentes en el mercado.

Las características del espectrómetro determinan entre otros parámetros la frecuencia de muestreo y la cantidad de muestras (ventana espacial).

Se destaca que de acuerdo con el teorema de Nyquist-Shannon, la frecuencia máxima de la señal está determinada por:

$$f_m \geq 2f_{s_{max}} \quad (11.19)$$

Donde f_m es la frecuencia de muestreo y $f_{s,max}$ es la frecuencia máxima de señal para poder reconstruir sin distorsión. Este marco teórico es necesario en el proceso de análisis que se desea realizar, ya que al tener fija la f_m si las señales se encuentran a frecuencias superiores de $f_{s,max}$ aparecerían problemas de aliasing que ocasionan interpretaciones erróneas.

Otro aspecto para tener en cuenta es la resolución, que de acuerdo con la teoría del análisis de señales se puede definir como “al mínimo salto en frecuencia (Δf) que se puede detectar correctamente sin alteraciones”, matemáticamente se describe como:

$$\Delta f = \frac{f_m}{N} \tag{11.20}$$

Donde N es la cantidad de muestras adquiridas en la ventana de adquisición. Un ejemplo que puede resultar esclarecedor es: si tenemos una señal de audio típicamente muestreadas a 48 kHz y durante la ventana temporal se adquieren 128 muestras, la fft tendrá una frecuencia de resolución de 375Hz (48000/128). Por tanto, no se deberán considerar ninguna señal por debajo 375Hz.

Por otro lado, la señal de LCI tiene una envolvente, en general de tipo gaussiana, por lo cual la frecuencia mínima detectable estará condicionada al ancho de dicha envolvente

En una primera aproximación y teniendo en cuenta a la Fig. 176, se considera que la señal de entrada a la RN se encuentra preprocesada como indica la Fig. 183.

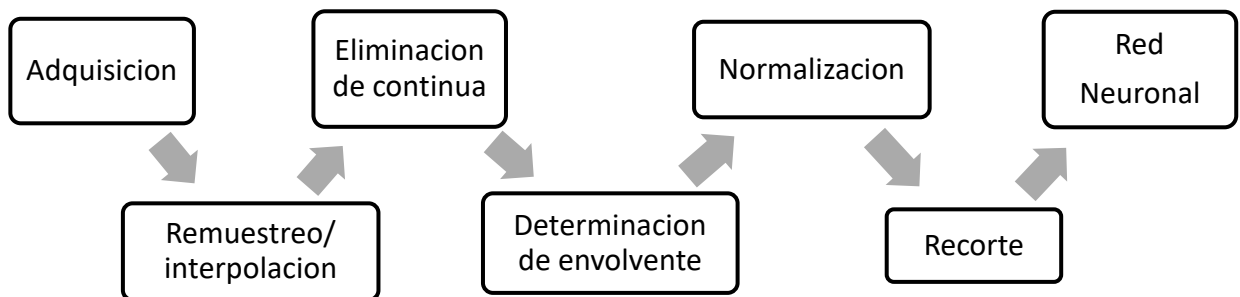


Fig. 183 – Esquema inicial de las etapas de preprocesado de una señal LCI antes de ser introducida a la red neuronal.

Por lo tanto, la señal de LCI a la entrada de la red neuronal queda definida como:

$$i(k) = \cos(kDCO) = \cos(k2z_s) \tag{11.21}$$

Donde z_s es el valor que se desea estimar, como n=1 la DCO es el doble del valor de z_s . Una situación muy similar se encuentra estudiada en mayor profundidad en (Sajedian & Rho, 2019) pero considerando señales temporales. Los autores luego de probar diferentes diseños de redes proponen uno asombrosamente sencillo capaz de estimar frecuencias, dentro de ciertos rangos. Una de las características más interesantes que se propone en (Sajedian & Rho, 2019) es analizar un rango de frecuencias que estén comprendidas dentro de un orden de magnitud, toma como ejemplo señales

entre 1 kHz a 10 kHz tomando 2000 muestras con una $f_m = 1$ MHz. Esta selección permite, por un lado, visualizar por lo menos dos oscilaciones completas para la frecuencia más lenta y, por otro lado, muestrear 100 veces la oscilación para la frecuencia más rápida. También amplía el rango de señales para la cual RN ya entrenada puede utilizarse ya que en caso de tener señales en otros rangos de frecuencias hay que cambiar la escala y la relación de la figura de entrada a la red sea la misma. Sin embargo, también es posible entrenar la RN propuesta para estimar rangos de frecuencia mayores. En el trabajo de (Sajedian & Rho, 2019) se utilizó una RN con 2000 elementos en la capa de entrada, 3 capas densamente conectadas o también llamadas “fully-connected” de 2, 2 y 3 neuronas cada una (también denominadas “Hidden layers”) y una neurona como capa de salida para obtener un único resultado regresivo como valor de frecuencia.

A lo largo de esta sección se trabajó con simulaciones de señales con z_s comprendidas entre 30 a 500 μm (es decir DCO de 60 a 1000 μm) tomando 30.000 o 60.000 o 100.000 valores equiespaciados en ese rango. Se utilizó una ventana espacial de 2000 muestras y una $f_m = \frac{1}{k_m} = 9.3$ mm. Esto permite que para la frecuencia mínima (DCO = 60 μm) se observen dos ciclos completos de la señal y que la señal de mayor frecuencia tenga 60 muestras por ciclo ya que $f_{max} \ll f_m$. Esto permite generar datos de entrada significativos para el aprendizaje de la RN.

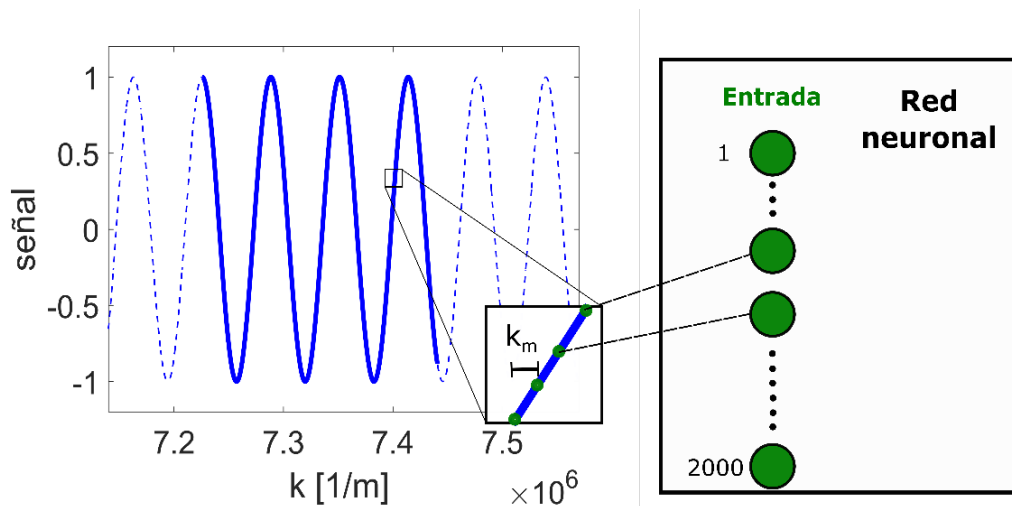


Fig. 184 – Muestreo de la señal a procesar para las diferentes redes neuronales. Los puntos 2000 se encuentran separados una distancia k_m y son los valores de entrada a la red neuronal. El ejemplo en este caso es una señal cosenoidal con un z_s de 50 μm .

Para el entrenamiento de todas las RN presentes en esta sección, a cada conjunto de señales se las separo de modo aleatorio de la siguiente manera: el 72% para entrenamiento, el 18% para validación y el 10% restante para testeo. Se utilizó en su mayoría el algoritmo de optimización Adam con un coeficiente de aprendizaje de 0.01 y el gradiente estocástico descendente con momento (sgdm) se utilizó en dos casos, que se lo detalla explícitamente.

De acuerdo a los buenos resultados obtenidos en (Sajedian & Rho, 2019) se decidió diseñar una RN igual, ver Fig. 183, y probar su funcionamiento para la ecuación (11.21). Se utilizó un conjunto de 100.000 señales, se entrenó para 60 iteraciones (epochs) con la función de entrenamiento Adam con coeficiente de entrenamiento inicial de 0.01. Como se puede observar en la Fig. 187, se obtuvo un $RMSE = 2.3$ μm y un $RSE_{max} = 14.8$ μm , el tiempo del entrenamiento demoro 7 minutos y la RN una vez ya entrenada tiene un tiempo de ejecución 2.0 ms para estimar el resultado. La simulación de las 100.000 señales demoro 40 minutos aproximadamente. Este es un muy buen

resultado ya que se encuentra para la mayoría de los valores de z_s salvo los que se encuentran en los extremos por debajo de la resolución típica de la técnica de LCI. Este comportamiento suele suceder en los extremos debido a que no existen valores con los que se puedan seguir entrenando a la RN. Se podría solucionar agrandando el rango de valores de z_s para el entrenamiento y luego recortando los extremos.

Un detalle que siempre ayuda al entrenamiento de las redes es normalizar los valores para que los outputs estén entre 0 y 1. Simplemente es dividir el valor de los outputs por su máximo y también recordar multiplicar por este mismo a los valores predichos.

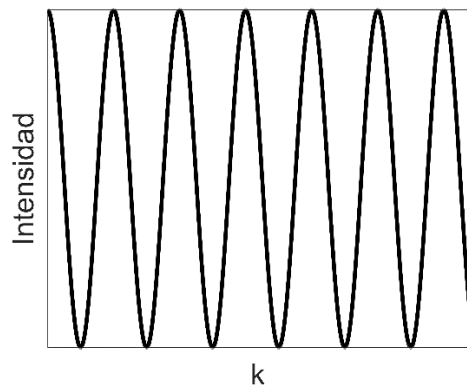


Fig. 185 – Intensidad de una señal cosenoidal en función de k . La frecuencia de la oscilación está dada por la distancia z_s

Entre las funciones utilizadas para el diseño y entrenamiento de las RNs se destacan: *imageInputLayer* que permite manejar de manera sencilla el gran número de inputs. Si bien sirve para manipular imágenes también se puede adaptar para señales unidimensionales, *layers* para el diseño de las sucesivas capas que componen a la RN y *trainNetwork* para el entrenamiento de redes neuronales de aprendizaje profundo.

El código utilizado para el armado de las capas queda entonces de la siguiente manera:

```
layers = [ ...
    imageInputLayer([2000 1 1])
    fullyConnectedLayer(2)
    fullyConnectedLayer(2)
    fullyConnectedLayer(3)
    fullyConnectedLayer(1)
    regressionLayer];
```

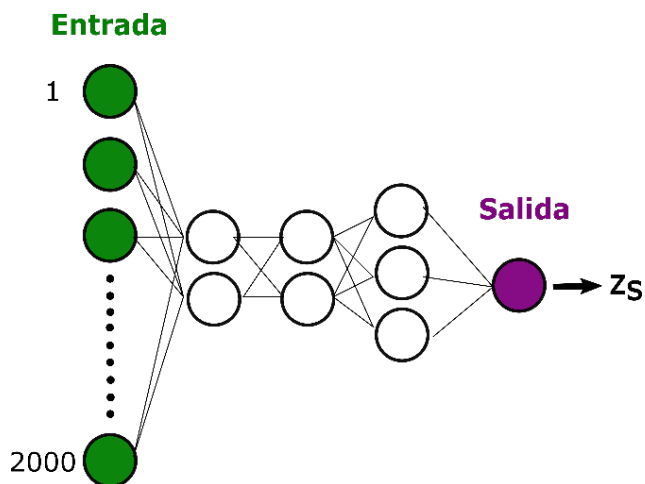


Fig. 186 – Esquema de la una red neuronal de tres capas densamente conectadas de 2,2 y 3 neuronas con salida regresiva.

Notar que no sirve usar la RN ya entrenada para estimar la frecuencia de otro tipo de señal. Por ejemplo, si se utiliza esta misma RN para predecir los valores de testeo, pero multiplicados por una amplitud de $A=2$ o $A=5$ se obtienen RMSE mayores a $100 \mu\text{m}$. Es por este motivo que cada conjunto de señales requiere su entrenamiento apropiado. Evidenciando además que la normalización juega un rol importante.

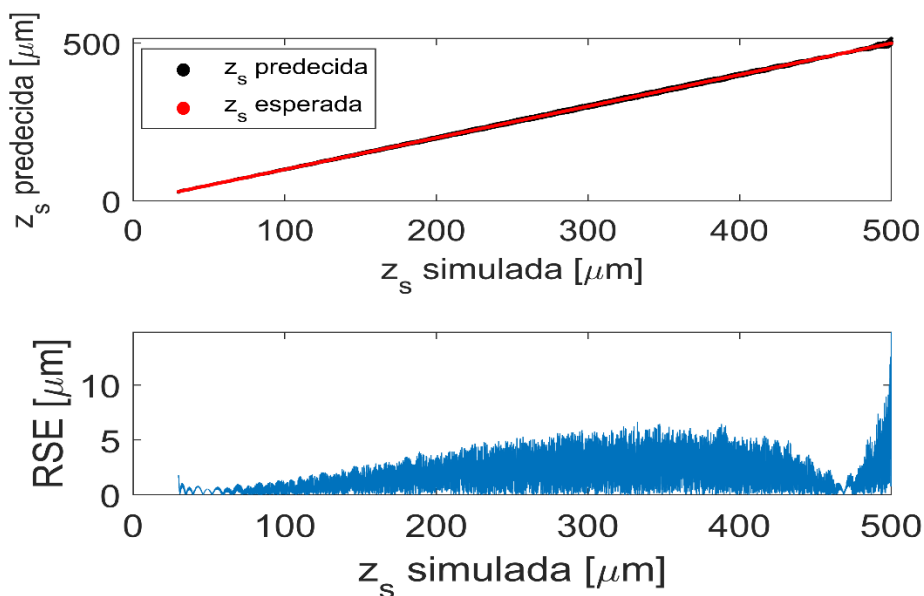


Fig. 187 – predicción para 100.000 señales coskOPD para red 223 para 60 epochs. $RMSE=2.3 \mu\text{m}$ y $RSE_{max}=14.8 \mu\text{m}$

Un problema que se logró identificar es que esta RN no funciona adecuadamente para el caso en que la señal de LCI presenta fase, en la Fig. 188 se puede observar el grafico de esta señal y en la ecuación (11.22) su expresión matemática.

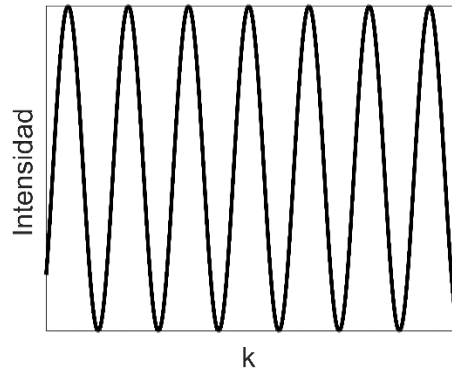


Fig. 188 – Esquema de una señal cosenoidal con la presencia de un desplazamiento debido a la fase extra.

$$i(k) = \cos(kDCO + \varphi) \quad (11.22)$$

Si las señales están desfasadas por φ tanto aleatoriamente como con cierta relación constante entre sí la RN no puede estimar la frecuencia con buena precisión con este tipo de red. La aparición de fase φ en las señales de LCI se pueden deber al sistema de medición o a la muestra en estudio. En la Fig. 189 se puede apreciar el valor estimado por la RN y el valor teórico, claramente la estimación de este modelo de RN es de mala calidad.

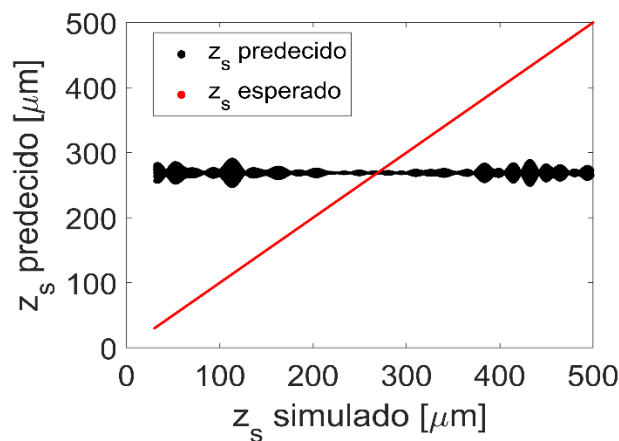


Fig. 189 – Se puede observar la mala predicción (negro) de la FCN 2-2-3 para señales que presentan fase distinta

El problema de la fase se puede solucionar con un preprocesamiento de la señal. En cuyo caso se utiliza una ventana deslizante con 2000 puntos que se ajusta su posición para que la señal de LCI siempre inicie desde fase cero.

Lo que se desea evidenciar con esta situación es que si bien la RN fully-connected o densamente conectada de capas de 2 neuronas, 2 neuronas y 3 neuronas (FCN 223) tiene gran capacidad de aprendizaje nunca va a poder aprender en situación donde las señales tengan un desfase entre ellas. Es por esto por lo que hay que recurrir a nuevos diseños de RN si se desean solucionar esta problemática.

Se pueden utilizar configuración de RN más elaboradas agregando más capas y más neuronas por capa del tipo fully-Connected (Almayyali & Hussain, 2021), agregando capas de activación. Si bien esto incrementa el tiempo requerido para entrenamiento permite mejorar la precisión de la predicción de la red.

Por lo tanto, se diseñó una RN de 7 capas con numero decreciente de neuronas por capa 1000, 500, 250,125, 50, 25 y 10. Como se puede observar en la Fig. 190.

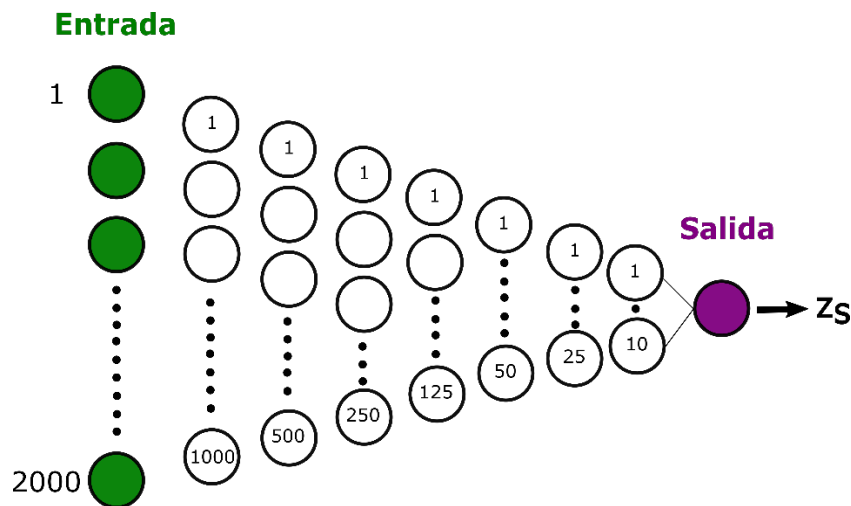


Fig. 190 – Esquema de la red densamente conectada FCN 100-500-250-125-50-25-10

Entrenando a este tipo de red con mayor número de neuronas, capas y funciones de activación del tipo tangente hiperbólica, como la que se describe a continuación:

```
layers = [
    imageInputLayer([2000 1 1])
    fullyConnectedLayer(1000)
    tanhLayer
    fullyConnectedLayer(500)
    tanhLayer
    fullyConnectedLayer(250)
    tanhLayer
```

```

fullyConnectedLayer(125)
tanhLayer
fullyConnectedLayer(50)
tanhLayer
fullyConnectedLayer(25)
tanhLayer
fullyConnectedLayer(10)
tanhLayer
fullyConnectedLayer(1)
tanhLayer
regressionLayer];
    
```

Donde después de cada etapa fullyConnected se utiliza una TanhLayer que actúa como una capa de activación de tangente hiperbólica. Para el entrenamiento de esta RN se usó el conjunto de 100.000 señales, 90 epochs, con función de entrenamiento se utilizó el sgd, una tasa de aprendizaje de 0.1 y una frecuencia de validación de 100. Se obtuvo un RMSE=2.2 μm y RSE_{MAX} = 16.5 μm . El entrenamiento demoró 27 minutos y el tiempo de 4.8 ms En la Fig. 191, se pueden observar los resultados predichos y los valores esperados, claramente hay un gran nivel de coincidencia, además se puede observar que las variaciones más grandes se encuentran en los extremos, es decir para los valores de frecuencia más pequeños y valores de frecuencia más grandes. También se observa la variación del RSE para cada valor proporcionado por la red.

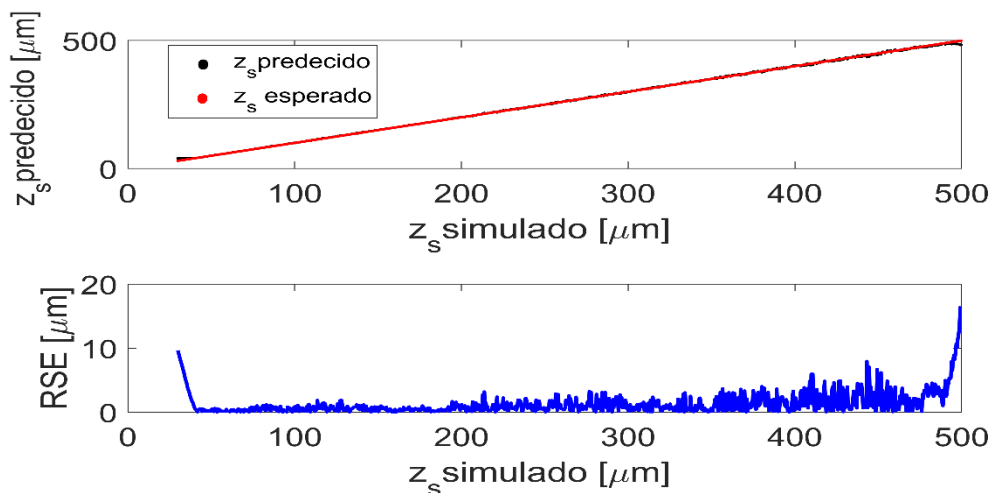


Fig. 191 – predicción y error para 100.000 señales $\text{Cos}(kD\text{CO} + \varphi)$ para una red densamente conectada de 1000-500-250-125-50-25-10 neuronas.

Los resultados anteriores son alentadores pensando en el objetivo final buscado que es el de eliminar todo el sistema de preprocesamiento y que sea la RN la única encargada de realizar el análisis sobre la señal de LCI.

En este punto, la RN ha demostrado poder determinar las frecuencias de las señales de LCI con fase, en el sistema de preprocesado ya no resulta necesaria la etapa de recorte quedando el esquema de la Fig. 192:

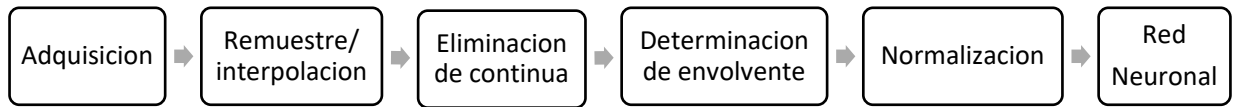


Fig. 192 – Esquema de preprocesado de la señal. Modificando la cualidad de la red neuronal se puede evitar la etapa de recorte.

Una etapa interesante de eliminar es la etapa de eliminación de envolvente, la señal antes de ingresar en esta etapa esta descrita por la Ecuación (11.23) y se puede observar su grafico en la Fig. 193.

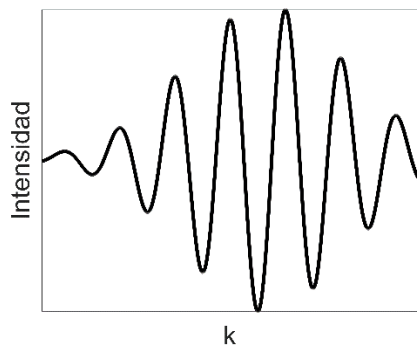


Fig. 193 – Esquema de una señal de LCI sin la componente de baja frecuencia. La cual posee una oscilación cosenoidal con una envolvente.

$$i(k) = s(k) \cos(kDCO + \varphi) \tag{11.23}$$

Por lo cual se entrenó la RN anterior con este tipo de señales para determinar la efectividad de su utilización. Se plantean condiciones similares a las que se utilizaron con anterioridad para el entrenamiento de la RN, se toma un conjunto 30.000 de señales de entrenamiento, los resultados se pueden ver en la Fig. 194 donde se puede observar que las frecuencias esperadas y las frecuencias estimadas difieren notoriamente, se obtuvo un RMSE=13.1 μm y RSE_{MAX} mayor a 100 μm este modelo de RN, no es apropiada para señales con una componente que le varié la amplitud para distintos valores de k, como lo indica la ecuación (11.23), si se toman más cantidades de señales se sigue observando un comportamiento similar.

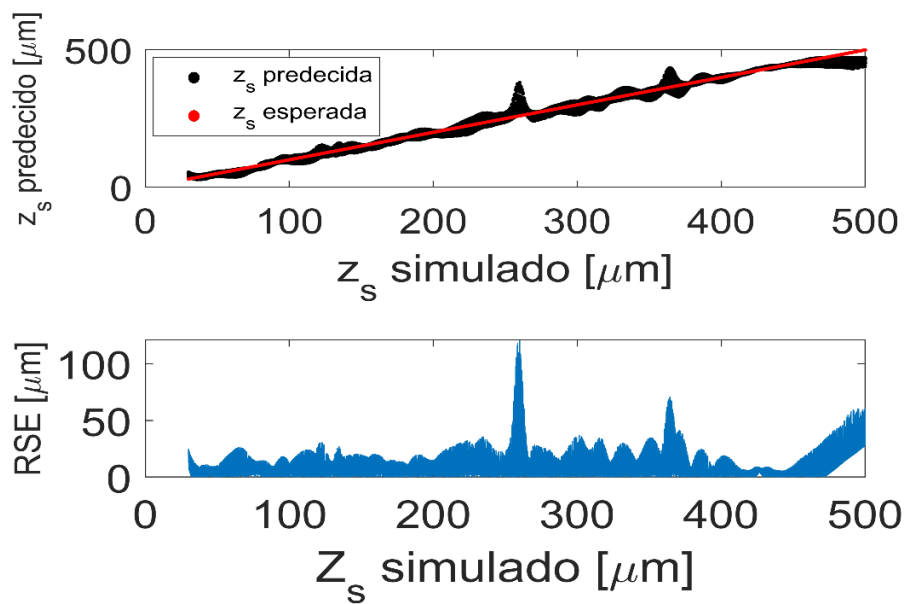


Fig. 194 - Predicciones para señales del tipo $s(k) \cos(kDCO + \varphi)$ para la red densamente conectada de 1000-500-250-125-50-25-10 neuronas.

Seguir agregando capas y neuronas densamente conectadas no consigue solucionar este problema. Si se desea solucionar esta problemática resulta interesante empezar a considerar RNs que tengan capacidades de aprendizaje distintas. Que su diseño y manera de aprender les permita reconocer patrones de mayor complejidad. Es necesario ir más profundo o como se comenta en una de las publicaciones más citadas sobre Redes Neuronales “We need to go deeper” (Szegedy et al., 2015). Si bien esa frase se usa en un contexto un poco diferente a este, aplica para la situación en que estamos. La propuesta para ir más profundo es usar Redes Convolucionales.

No está de más remarcar, que las RNs propuestas a lo largo de todo este capítulo tienen la capacidad de estimar frecuencias de señales oscilantes sea cual sea su naturaleza. Se estuvo analizando particularmente señales de LCI, pero con las mismas RNs se pueden analizar cualquier otra problemática donde haya una única oscilación y se desea estimar su frecuencia. Por ejemplo, podrían servir para estudiar el movimiento de un péndulo pasando por la tensión eléctrica de entrada a las casas hasta llegar a una comunicación satelital.

Ganando profundidad para estimar frecuencias de señales completas

Las redes neuronales convolucionales son un tipo de RN diseñado originalmente para procesar datos de imágenes, pero las mismas propiedades que hacen propicias para problemas de visión las hacen muy relevantes para procesar señales. Se puede extrapolar entendiendo al tiempo como una dimensión espacial, esta línea ha nacido recientemente y es denominada redes convolucionales en 1D, trabajos actuales han demostrado que para ciertas aplicaciones se puede obtener resultados similares que usando convoluciones en 2D (Kiranyaz et al., 2021) (Kiranyaz et al., 2019). La gran ventaja es que la configuración simple y compacta, debido a sus operaciones unidimensionales permite implementaciones en tiempo real y con menores requerimientos del hardware.

Las capas de convolución en una dimensión obtienen nuevas secuencias a través de filtros que interpretan ciertas características de las secuencias originales que permiten reconocer patrones locales en la misma. Son altamente utilizadas en aplicaciones como clasificación de datos biomédicos y diagnóstico temprano, detección e identificación de anomalías se señales electrónicas, detección de fallas en motores eléctricos, etc.

En este sentido la capa convolucional es, tal vez, la capa más significativa de este tipo de RNs. La diferencia fundamental entre una capa densamente conectada y una capa convolucional es que las capas densas aprenden patrones globales de sus entradas, mientras que las convolucionales aprenden filtros (o Kernels) que modifican la señal original, generando descriptores o mapas de características.

Esta arquitectura le permite a la red concentrarse en características más simples en las primeras capas y agruparlas luego en características más complejas en las capas siguientes. Las capas convolucionales se componen de estructuras, filtros y mapas de características. Los filtros son esencialmente los distintos grupos de pesos sinápticos de las neuronas de la capa. Durante el entrenamiento, la red neuronal convolucional encuentra los filtros más útiles para su tarea. La salida de la RN convolucional es el mapa de características (Albawi et al., 2017). Es en este punto donde la RN determina las características más relevantes, se puede decir que los parámetros antes seleccionados por el usuario ahora los determina la RN convolucional (Khan et al., 2018).

La capa de pooling tiene como objetivo reducir la muestra anteriormente procesada, lo que disminuye la carga computacional, utilización de memoria y número de parámetros, extrayendo subsecuencias de una entrada y devolviendo el valor máximo o promedio.

Otra capa interesante que puede tener una RN convolucional es la de batchnormalization. Esta capa normaliza el valor medio y la varianza de la salida de activación. Hay grandes discusiones en torno a las consecuencias estrictas de esta capa (Ioffe & Szegedy, 2015) pero es innegable su capacidad de mejora de la velocidad de convergencia y estabilidad frente a la elección de hiperparámetros o a la inicialización de los pesos.

Utilizando una red convolucional es posible generar análisis de señales de mayor complejidad. El aprendizaje en los coeficientes del kernel sumado a las sucesivas capas como de activación (por ejemplo, de rectificación lineal (ReLU), extracción de promedios o máximos (MaxPooling), las mejoras aportadas por la capa de batchnormalization y las ya conocidas Fully-connected permiten una identificación de patrones más profunda.

Se propuso una red conformada por una capa Convolucional, ver Fig. 195 y se fueron modificando los hiperparámetros para su funcionamiento óptimo. Se eligieron 10 filtros con una dimensión del kernel de 5x1 ya que es una señal unidimensional. Se configuro dependiendo el caso, una cantidad máxima de epochs de 30, 60 o 90. A continuación, se colocó una capa ReLU seguida de una capa MaxPool que extrae el máximo valor dentro 5 lugares y se desplaza un stride de 5, finalmente dos capas fully-Connected de 10 neuronas para obtener el valor regresivo deseado. El código del armado de las capas queda en Matlab de la siguiente manera:

```
layers = [ ...  
    imageInputLayer([2000 1 1])  
    convolution2dLayer([5 1],10)  
    reluLayer  
    maxPooling2dLayer([5 1],'Stride',5)  
    fullyConnectedLayer(10)
```

```
fullyConnectedLayer(10)
fullyConnectedLayer(1)
regressionLayer];
```

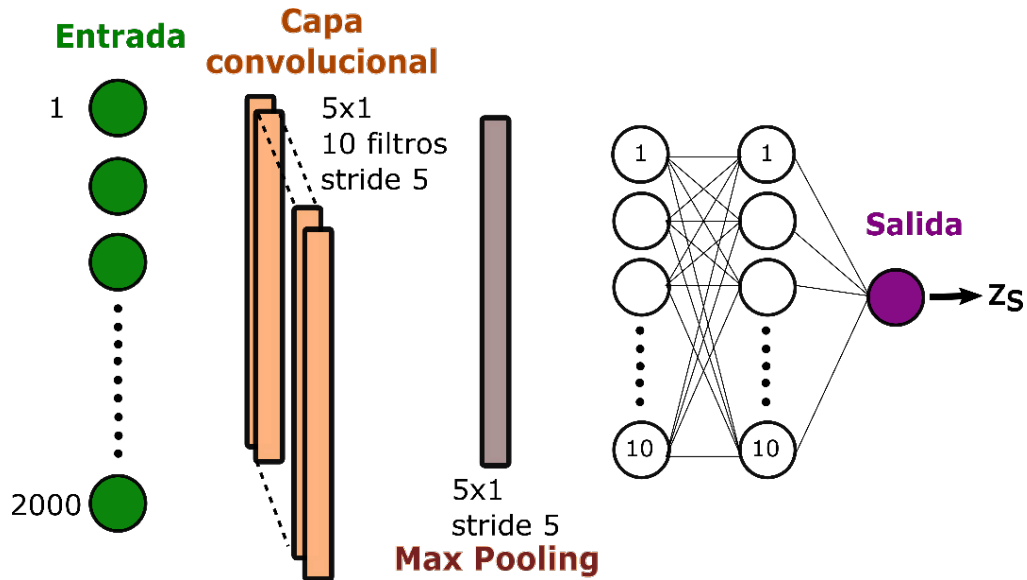


Fig. 195 -Esquema de la red neuronal que contiene una única capa convolucional, seguido de una capa max Pooling y dos capas densamente conectadas de 10 neuronas.

A esta red se la entreno con la señal de la ecuación (11.22) , para poder comparar los resultados con las RNs anteriores, donde se obtuvo un $RMSE = 1.5 \mu m$ y $RSE_{MAX} = 6.9 \mu m$, se usó el conjunto de 30000 señales, 60 epochs para el entrenamiento. En la Fig. 196 se puede observar la curva de las frecuencias estimadas y de las frecuencias utilizadas, la precisión obtenida es mejor que para la red anterior propuesta y necesita de menos cantidad de señales para su entrenamiento evidenciando su mayor capacidad de aprendizaje. El error está por debajo de los valores típicos y tolerables de LCI.

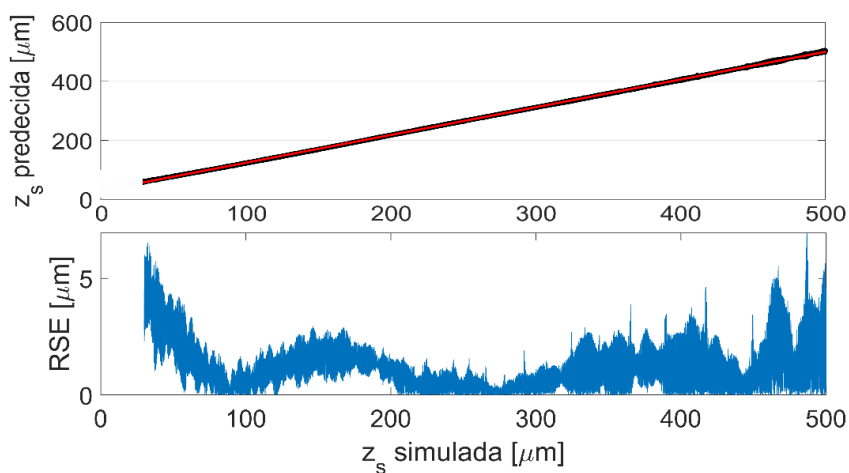


Fig. 196 - Predicción y error absoluto de la red convolucional para todos los valores de z_s para señales cosenoidales con presencia de fases distintas.

Claramente presenta una mejoría con respecto a los casos utilizados anteriormente, donde además el tiempo utilizado para su entrenamiento fue de 18 minutos, cuestión de interés ya que permite entrenar a la red rápidamente si se cambian parámetros del sistema de medición, como por ejemplo el propio espectrómetro que se utiliza en la adquisición. El tiempo de ejecución de la red es de 2.3 ms.

Si bajo las mismas condiciones anteriores se utilizan señales no normalizadas, es decir con amplitudes variadas emulando por ejemplo las distintas reflectividades de la muestra a analizar:

$$i(k) = A \cos(kDCO + \varphi) \tag{11.24}$$

Donde para cada DCO simulada se tomó un valor de A aleatorio comprendido entre 0 y 1, se obtienen las siguientes predicciones:

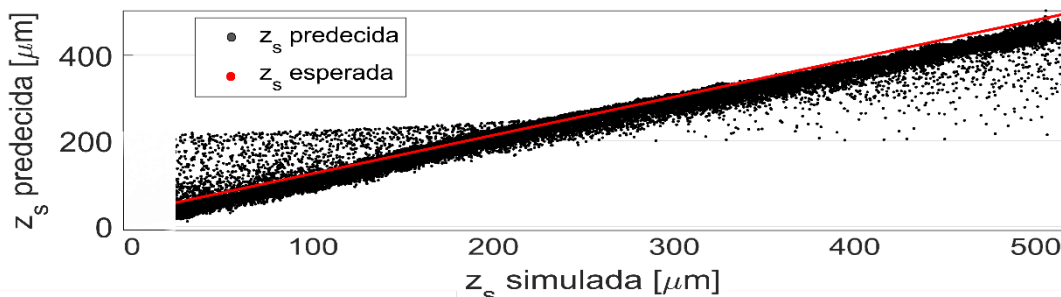


Fig. 197 - Predicción para señales cosenoidales con presencia de fase y amplitudes distintas con red convolucional.

Donde el valor de $RMSE = 37 \mu m$, lo que indica un incremento casi de un orden del error como se puede observar en la Fig. 197. Este comportamiento se evidenció independientemente de que señal era, siempre que las señales no estaban normalizadas la precisión de la estimación empeoraba. Esto nos permite de manera práctica, postular que es mejor que las señales que se utilicen se encuentren normalizadas. La normalización permite obtener una mejor aproximación sobre el mismo modelo de RN.

Volvamos a la situación donde se pretende analizar la señal de LCI con envolvente, ecuación (11.23), en este caso se utilizaron 30000 señales normalizadas, y la misma RN convolucional se la entrenó utilizando los mismos parámetros. Para el caso en que se tomaron 60 epochs se obtuvo $RMSE = 2.5 \mu m$ y un $RMS_{max} = 7.5 \mu m$, un tiempo de entrenamiento de 18 minutos y un tiempo de ejecución de 2.4 ms. En la Fig. 198 se muestran los valores de frecuencia esperados y los valores de frecuencias simulados y la curva de error estimados para esta RN. Los resultados obtenidos son altamente positivos ya que se encuentran por debajo de la resolución de la técnica de FD-LCI.,

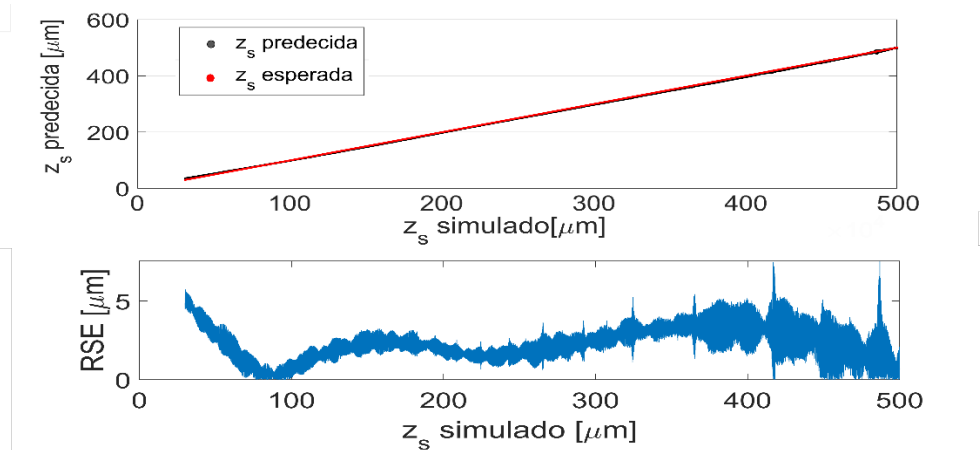


Fig. 198 - Predicciones y error absoluto de la red convolucional para señales.

Bajo estas condiciones se puede acortar el preprocesamiento nuevamente, Fig. 199, dejando solo tres bloques de procesamiento que son el remuestreo, la eliminación de continua y la normalización necesaria.

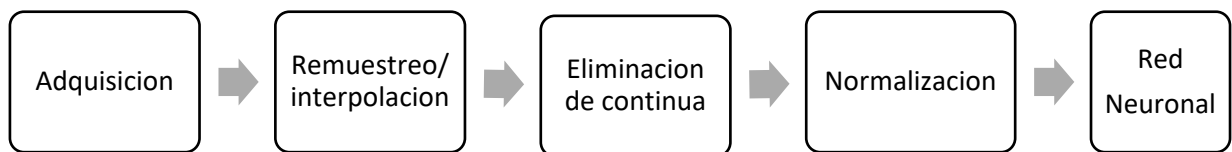


Fig. 199 - Esquema de las etapas de preprocesado, Notar que empleando una red neuronal de mayor capacidad de aprendizaje se puede evitar también la etapa de determinación de la envolvente.

Esta vez se utilizaron las señales simuladas a partir de la ecuación (11.25), donde se contemplan todas las características típicas de una señal de perfilometría sin procesamiento previo, es decir con la presencia de una componente continua A_{DC} y con la envolvente espectral $s(k)$. Recordamos que para cada valor de DCO se tomaron valores aleatorios para A_{DC} , A y φ :

$$i(k) = s(k)[A_{DC} + A \cos(kDCO + \varphi)] \quad (11.25)$$

No se obtuvieron buenos resultados al entrenar la red con el diseño mostrado en la Fig. 195 para este tipo de señales, ya que no es simplemente agregar un valor continuo A_{DC} (notar que la capa ImageInputLayer suele extraer el valor medio de la señal) si no que ese valor esta modulado para cada valor de k por $s(k)$. Por lo tanto, se probaron nuevos diseños de RN. Se fue variando la cantidad de capas convolucionales, la cantidad de filtros por capa como su tamaño. Se buscaron configuración donde el número de filtros aumentaba por cada capa convolucional agregada y otras donde el número de filtros decrecía. Análogamente, se probó con tamaños de los filtros grandes (50x1) en la/s primera/s capas y también con filtros pequeños para todas las capas (5x1, 3x1, etc.). También se fue variando el paso del stride de los filtros y agrego "padding" ya que estos repercuten fuertemente en la dimensión de la salida de cada capa. Además, se agregaron capas de Bathnormalization y

diferentes capas de activación después de cada capa convolucional, se fue modificando la cantidad de capas densamente conectadas y su número de neuronas en función de la dimensión de la salida de la última capa convolucional y/o de extracción (max pooling). Vale la pena notar el aumento significativo del tiempo de entrenamiento al tomar más cantidad de capas convolucionales con gran cantidad de filtros y pasos de stride muy pequeños.

El diseño de red que mejor precisión logró fue para una RN de más de 40 capas. Esta configuración posee 7 capas convolucionales donde la cantidad de filtros fue duplicándose, con un tamaño de 2x1 para la primera capa, seguido de 4x1 para las intermedias y de 3x1 para las últimas dos, el paso de stride fue de 2x1 para la primera capa convolucional y de 3x1 para las restantes. Entre capa y capa se tomó la función de activación de tangente hiperbólica, se usó también etapas de batchnormalization y de dropout. Finalmente se colocaron capas densamente conectadas de 5000-5000-1000-1000-500-500-250-64-32-16-8-1. El código de la red quedó de la siguiente manera:

```
layers = [ ...
  imageInputLayer([2000 1 1])
  convolution2dLayer([2 1],8, 'Stride',[2 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([4 1],16, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([4 1],32, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([4 1],64, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([4 1],128, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([3 1],256, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  convolution2dLayer([3 1],512, 'Stride',[3 1], 'padding', 'same')
  batchNormalizationLayer
  tanhLayer
  dropout(0.5)
  fullyConnectedLayer(5000)
```

```

    tanhLayer
    fullyConnectedLayer(5000)
    tanhLayer
    fullyConnectedLayer(1000)
    tanhLayer
    fullyConnectedLayer(1000)
    tanhLayer
    fullyConnectedLayer(500)
    tanhLayer
    fullyConnectedLayer(500)
    tanhLayer
    fullyConnectedLayer(250)
    tanhLayer
    fullyConnectedLayer(64)
    tanhLayer
    fullyConnectedLayer(32)
    tanhLayer
    fullyConnectedLayer(16)
    tanhLayer
    fullyConnectedLayer(8)
    tanhLayer
    fullyConnectedLayer(1)
    tanhLayer
    regressionLayer];

```

Se uso para el entrenamiento el conjunto de señales de 100.000 señales, la función de optimización del sgd y se obtuvo un RMSE = 3.3 μm y un $\text{RSE}_{\text{max}} = 104.4 \mu\text{m}$, un tiempo de ejecución de 18.3 ms y un tiempo de entrenamiento de 1218 minutos. Como se puede observar en la figura 27, se obtienen en general valores muy próximos a los esperados, salvo por 3 valores que se obtienen predicciones con un alto error y nuevamente se observa un incremento continuo del error para el extremo de mayores z_s pero que se encuentran dentro del error aceptable. Si bien la RN tiene un óptimo desempeño para la mayoría de los valores de DCO existe una muy baja probabilidad de predicción del z_s con gran error.

En general, cuando se realizan una perfilometría con la técnica de LCI, una vez tomadas las mediciones se observa si hay puntos que se encuentran ampliamente fuera de la tendencia de la muestra y de manera aislada, de ser así pueden ser descartados ya que no representan información fiel de la superficie de la muestra. Esto sucedería para el caso de esta RN al predecir uno de los puntos de mayor error. Al tener una gran variación respecto de los valores de su alrededor, pueden no ser tenidos en cuenta para el análisis.

Por otro lado, se observa como la red incrementa considerablemente su estructura, como así también los tiempos tanto para entrenamiento como de ejecución. En especial el tiempo de ejecución aumenta un orden de magnitud haciendo que pierda sentido reemplazar la etapa de preprocesado de eliminación de continua por esta RN.

Se buscó aumentar la información (data augmentation) espejando las señales y/o se tomaron hasta 500.000 señales con diferentes DCO, pero esto no sirvió para corregir esas pequeñas estimaciones con alto grado de error.

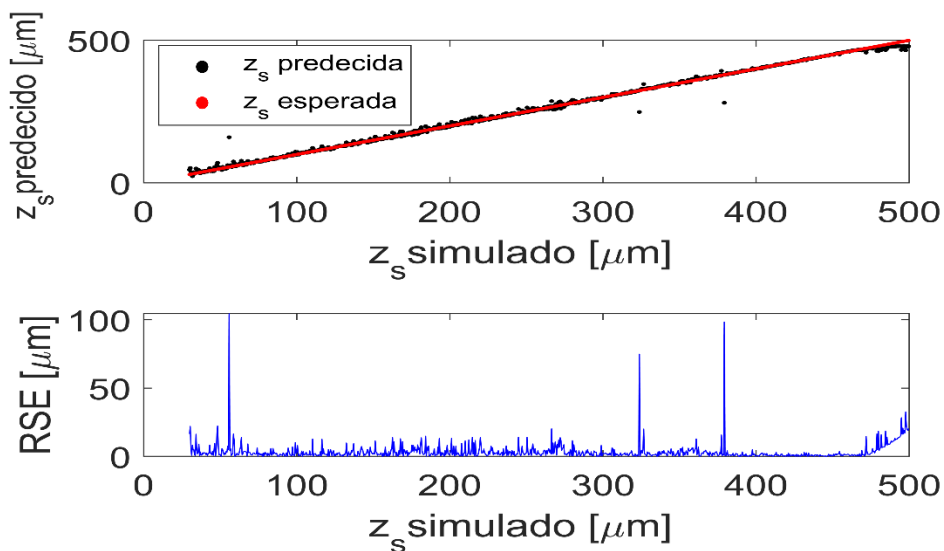


Fig. 200 - Predicciones y error absoluto de la red con 7 capas convolucional para señales típicas de LCI

En la Tabla II se recopilaron los datos de las distintas señales analizadas a lo largo de capítulo, considerando la cantidad de señales utilizadas para el entrenamiento, el diseño de las diversas redes, la precisión y los tiempos tanto de entrenamiento como de ejecución para poder evaluar de una manera rápida el desempeño de las redes.

En general dada un diseño de RN establecido una manera de aumentar la precisión es incrementando el número de señales con las que se entrena y también aumentando el número de epochs. Esto obviamente posee un límite que esta dado por la capacidad de aprendizaje de la red como se mostró a lo largo del capítulo. Si se duplica la cantidad de epochs el tiempo de entrenamiento también se duplica mientras que si se duplica la cantidad de señales el tiempo de entrenamiento crece menos que el doble.

Tabla XVIII – Resumen de los resultados de las redes y señales de mayor precisión. Se evidencian los tiempos tanto de entrenamiento como de ejecución.

Señal	Cantidad de señales	epoch	Red Neuronal	RMSE [μm]	RSE _{max} [μm]	Tiempo entrenamiento [minutos]	Tiempo ejecución [ms]
$\cos(kDCO)$	100.000	60	FCN 2-2-3	2.3	14.8	7	2.0
$\cos(kDCO + \phi)$	100.000	90	FCN 100-500-250-125-	2.2	16.5	27	4.8
$\cos(kDCO + \phi)$	30.000	60	Conv-Relu-MaxPool-FCN 10-10	1.5	6.9	18	2.3
$s(k) \cos(kDCO + \phi)$	30.000	60	Conv-Relu-MaxPool-FCN 10-10	2.5	7.5	18	2.4
$s(k)[A_{DC} + A \cos(kDCO + \phi)]$	100.000	60	7 Conv	6.3	104.3	1218	18.4

Para estimar la frecuencia de señales oscilante simples la mejor opción es usar una RN FC 2-2-3 ya que se puede obtener una muy buena precisión con pocos minutos de entrenamiento y el tiempo de ejecución es el más bajo respecto a las otras redes. Si la señal a analizar tiene mayor grado de complejidad (presencia de desfase, señal continua, variación en la amplitud) conviene utilizar redes convolucionales ya que debido a su gran capacidad de aprendizaje requiere de menor cantidad de señales, epochs y tiempo para su entrenamiento, logrando una alta precisión y con bajo tiempo de ejecución. En particular para analizar las señales de LCI lo mejor es implementar la opción híbrida entre redes neuronales y preprocesado que se observa en la Figura 26 ya que se logran precisión por debajo del error característico de LCI y se reduce el tiempo de cómputo.

Conclusiones

En este capítulo se estudió el uso de diferentes diseños de RN para estimar frecuencias de distintos tipos de señales oscilantes analizando su precisión y tiempos característicos, tanto de entrenamiento como de ejecución. Se mostró que el uso de redes neuronales del tipo densamente conectadas como convolucionales son una alternativa interesante para reducir el preprocesado necesario y reemplazar el uso de la transformada de Fourier para calcular la información de interés en la técnica de FD-LCI.

Se concluyó que la mejor opción para estimar la DCO de una señal de LCI es mantener las etapas de preprocesado de eliminación de continua y normalización, Fig. 199 , y usar una red neuronal con una capa convolucional. Comparando este esquema con el de la Fig. 176 y teniendo en cuenta los tiempos de ejecución de las Tabla XVII y la Tabla XVIII, se observa que teniendo ya entrenada la red neuronal se puede estimar la frecuencia de una señal de aproximadamente 2 ms más rápido. En caso de otras aplicaciones donde, la señal oscilatoria sea más sencilla (como senos o cosenos) usar redes densamente conectadas es una muy buena opción.

Se continuó profundizando en el estudio sobre diferentes redes neuronales para el análisis de frecuencias y se aplicó para señales ópticas de interferencia. Si bien existen algunos trabajos recientes que usan redes densamente conectadas para estimar frecuencias de señales senoidales con presencia de ruido (Sajedian & Rho, 2019) (Almayyali & Hussain, 2021) y otros que usan redes neuronales convolucionales para estimar Modulaciones lineales de frecuencia (Chen et al., 2019). Es un tema que aún falta mucho por estudiar y más aun buscando nuevos ámbitos donde aplicarlas.

Los resultados obtenidos en esta sección se basan entorno a señales de interferencia de baja coherencia, pero fácilmente se pueden implementar para otro ámbito donde se desea analizar la frecuencia de señales similares o inclusive tomar la estructura de las redes neuronales propuestas y modificarlas para optimizar su rendimiento en la aplicación deseada.

Algunas de las ideas a seguir trabajando en el futuro son: continuar trabajando en el diseño de RN para que permitan estimar señales que posean más de una frecuencia presente. Esto permitirá extender el uso del aprendizaje profundo (deep learning) para un gran número de nuevas aplicaciones, puntualmente dentro de la interferencia de baja coherencia permitirá realizar tomografías ya que se podrá identificar información de las interfases dentro de material. Además, gracias al entrenamiento que requiere la red es altamente probable que se pueda extraer la etapa de remuestreo/interpolación, que es muy importante en caso de realizar la fft, pero no si se desea usar una red neuronal, permitiendo reducir aún más los tiempos de estimación de frecuencia. Por último, nos gustaría explotar la característica de personalización de las redes neuronales. Dado que es necesario su entrenamiento, personalizar su aprendizaje para que identifique más fácilmente características puntuales de la muestra de interés como así también que permite minimizar algún artefacto asociado al sistema de detección.

Referencias

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a Convolutional Neural Network. *2017 International Conference on Engineering and Technology (ICET) IEEE*. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Ali, M., & Parlapalli, R. (2010). Signal processing overview of optical coherence tomography systems for medical imaging. *Texas Instruments, June, June, 1–22*. <http://www.tij.co.jp/jp/lit/wp/sprabb9/sprabb9.pdf>
- Almayyali, H. R., & Hussain, Z. M. (2021). Deep learning versus spectral techniques for frequency estimation of single tones: Reduced complexity for software-defined radio and iot sensor communications. *Sensors, 21(8)*. <https://doi.org/10.3390/s21082729>
- Bamler, R. (1991). Doppler frequency estimation and the Cramer-Rao bound. *IEEE Transactions on Geoscience and Remote Sensing, 29(3)*, 385–390. <https://doi.org/10.1109/36.79429>
- Belega, D., & Dallet, D. (2008). Frequency estimation via weighted multipoint interpolated DFT. *IET Science, Measurement and Technology, 2(1)*, 1–8. <https://doi.org/10.1049/iet-smt:20070022>
- Cerrotta, S., Morel, E. N., & Torga, J. R. (2015). Scanning Optical Coherence Tomography Applied to the Characterization of Surfaces and Coatings. *Procedia Materials Science, 9*, 142–149. <https://doi.org/10.1016/j.mspro.2015.04.018>
- Chen, X., Jiang, Q., Su, N., Chen, B., & Guan, J. (2019). LFM signal detection and estimation based on deep convolutional neural network. *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019, November, 753–758*. <https://doi.org/10.1109/APSIPAASC47483.2019.9023016>
- Classen, F., & Meyr, H. (1994). Frequency synchronization algorithms for OFDM systems suitable for communication over frequency selective fading channels. *IEEE Vehicular Technology Conference, 3*, 1655–1659. <https://doi.org/10.1109/vetec.1994.345377>
- Deng, L. (2008). Expanding the scope of signal processing. *IEEE Signal Processing Magazine, 25(3)*. <https://doi.org/10.1109/MSP.2008.920380>
- Dorrer, C., Belabas, N., Likforman, J.-P., & Joffre, M. (2000). Spectral resolution and sampling issues in Fourier-transform spectral interferometry. *Journal of the Optical Society of America B, 17(10)*, 1795. <https://doi.org/10.1364/JOSAB.17.001795>
- Dufour, M. (2006). Inspection of hard-to-reach industrial parts using small-diameter probes. *SPIE Newsroom, January, 2–5*. <https://doi.org/10.1117/2.1200610.0467>
- Gao, X., Jin, S., Wen, C. K., & Li, G. Y. (2018). ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers. *IEEE Communications Letters, 22(12)*, 2627–2630. <https://doi.org/10.1109/LCOMM.2018.2877965>
- Grulkowski, I., Liu, J. J., Potsaid, B., Jayaraman, V., Jiang, J., Fujimoto, J. G., & Cable, A. E. (2013). High-precision, high-accuracy ultralong-range swept-source optical coherence tomography using vertical cavity surface emitting laser light source. *Optics Letters, 38(5)*, 673–675.
- Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., Hee, M. R., Flotte, T., Gregory, K., Puliafito, C. A., & Fujimoto, J. G. (1991). Optical Coherence Tomography. *Science, 254(5035)*, 1178–1181. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4638169/>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015, 1*, 448–456.

- Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. In G. Medioni & S. Dickinson (Eds.), *Synthesis Lectures on Computer Vision* (Vol. 8, Issue 1). Morgan & Claypool. <https://doi.org/10.2200/s00822ed1v01y201712cov015>
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398. <https://doi.org/10.1016/j.ymssp.2020.107398>
- Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., & Gabbouj, M. (2019). 1-D Convolutional Neural Networks for Signal Processing Applications. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2019-May*, 8360–8364. <https://doi.org/10.1109/ICASSP.2019.8682194>
- Lai, L. L., Tse, C. T., Chan, W. L., & So, A. T. P. (1999). Real-time frequency and harmonic evaluation using artificial neural networks. *IEEE Transactions on Power Delivery*, *14*(1), 52–57. <https://doi.org/10.1109/61.736681>
- LeCun, Y., Chopra, S., Ranzato, M. A., & Huang, F. J. (2007). Energy-based models in document recognition and computer vision. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 1(Icdar)*, 337–341. <https://doi.org/10.1109/ICDAR.2007.4378728>
- Liu, L., McLernon, D., Ghogho, M., Hu, W., & Huang, J. (2012). Ballistic missile detection via micro-Doppler frequency estimation from radar return. *Digital Signal Processing: A Review Journal*, *22*(1), 87–95. <https://doi.org/10.1016/j.dsp.2011.10.009>
- Martínez-Montejano, R. C., Castillo Meraz, R., Lozoya-Ponce, R. E., Campos-Cantón, I., Martínez-Montejano, M. F., & Lozoya Ponce, R. O. (2014). Phase locked loop based on adaptive observer. *International Review of Electrical Engineering*, *9*(1), 244–248. <https://doi.org/10.15866/iree.v9i1.221>
- Morel, E. N., Russo, N. A., Torga, J. R., & Duchowicz, R. (2016). Interferometric system based on swept source-optical coherence tomography scheme applied to the measurement of distances of industrial interest. *Optical Engineering*, *55*(1), 014105-1/7. <https://doi.org/10.1117/1.OE.55.1.014105>
- Orović, I., Stanković, S., Thayaparan, T., & Stanković, L. J. (2010). Multiwindow S-method for instantaneous frequency estimation and its application in radar signal analysis. *IET Signal Processing*, *4*(4), 363–370. <https://doi.org/10.1049/iet-spr.2009.0059>
- Rosemarie, V. (2008). Discrete Fourier Transform computation using neural networks. *Proceedings - 2008 International Conference on Computational Intelligence and Security, CIS 2008, 1*, 120–123. <https://doi.org/10.1109/CIS.2008.36>
- Routray, A., Pradhan, A. K., & Rao, K. P. (2002). A novel Kalman filter for frequency estimation of distorted signals in power systems. *IEEE Transactions on Instrumentation and Measurement*, *51*(3), 469–479. <https://doi.org/10.1109/TIM.2002.1017717>
- Sajedian, I., & Rho, J. (2019). Accurate and instant frequency estimation from noisy sinusoidal waves by deep learning. *Nano Convergence*, *6*(1), 2–6. <https://doi.org/10.1186/s40580-019-0197-y>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tang, Y., & Elasmith, C. (2010). Deep networks for robust visual recognition. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 1055–1062.

- Walecki, W. J., Lai, K., Pravdivtsev, A., Souchkov, V., Van, P., Azfar, T., Wong, T., Lau, S. H., & Koo, A. (2005). Low-coherence interferometric absolute distance gauge for study of MEMS structures. *Reliability, Packaging, Testing, and Characterization of MEMS/MOEMS IV*, 5716, 182. <https://doi.org/10.1117/12.590013>
- Walecki, W. J., Pravdivtsev, A., Santos II, M., & Koo, A. (2006). High-speed high-accuracy fiber optic low-coherence interferometry for in situ grinding and etching process monitoring. *Interferometry XIII: Applications*, 6293, 62930D. <https://doi.org/10.1117/12.675592>
- Wang, M. R. (n.d.). *Optical Coherence Tomography and Its Non- medical Applications*.
- Wang, T., Wen, C. K., Jin, S., & Li, G. Y. (2019). Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels. *IEEE Wireless Communications Letters*, 8(2), 416–419. <https://doi.org/10.1109/LWC.2018.2874264>
- Xu, S., & Shimodaira, H. (2019). Direct F0 estimation with neural-network-based regression. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2019-Septe*, 1995–1999. <https://doi.org/10.21437/Interspeech.2019-3267>
- Yardibi, T., Li, J., Stoica, P., Xue, M., & Baggeroer, A. B. (2010). Source localization and sensing: a nonparametric iterative adaptive approach based on weighted least squares. *IEEE Transactions on Aerospace and Electronic Systems*, 46(1), 425–443. <https://doi.org/10.1109/TAES.2010.5417172>