Universidad Zaragoza
1542

Trabajo Fin de Grado

Inteligencia artificial aplicada a problemas de interacción luz-materia

Autor

Javier Lagos García

Directores

Dr. Sergio Gutiérrez Rodrigo
Dr. Luis Martín Moreno

Facultad De Ciencias, Departamento de Física de la Materia Condensada

2021

ÍNDICE

1. INTRODUCCIÓN

La fotónica estudia la interacción entre la luz y la materia, que comprende fenómenos como la generación, absorción, emisión y procesado de la luz y sus posibles aplicaciones en dispositivos de distinto carácter [1]. La luz visible, capaz de ser detectada directamente por el ojo humano, se encuentra en el rango de longitudes de onda de ente 400 y 700 nanómetros aproximadamente. Normalmente, en el estudio de la fotónica se incluyen zonas cercanas al infrarrojo y al ultravioleta, ampliando así el rango de longitudes de onda que se estudian de unos 100 nanómetros hasta 2 micrómetros. La Nanofotónica, por otro lado, estudia la interacción de la luz con la materia a una escala inferior a la propia longitud de onda de la luz. En este caso, nuevas propiedades emergen del estudio de las propiedades electromagnéticas que surgen de la interacción entre la luz y la materia nanoestructurada.

El estudio de esta interacción a escala nanométrica puede ser interesante en muchos campos, no sólo en la física; por ejemplo, en terapias activadas mediante luz en nanomedicina o en diagnósticos ópticos [2]; procesos derivados de biomateriales, como el estudio de materiales inspirados en otros biológicos o el estudio de los procesos de producción de polímeros en biorreactores mediante fotones en las bacterias [2]; así como la implementación de guías de onda nano-ópticas en circuitos electrónicos integrados para reducir así los retrasos producidos en la conexión entre los transistores o aumentar el ancho de banda [3].

Los fenómenos investigados en Nanofotónica son muchos y se solapan con diversos campos, desde la biología hasta las tecnologías de la información. Todos tienen en común algo, la necesidad de una descripción lo más detallada posible; de ahí la importancia de los desarrollos teóricos que no solo sirven para explicar nuevos fenómenos descritos por la experimentación, sino que actúan como generadores de nueva física. El avance que la Nanofotónica ha vivido en las últimas décadas en parte se debe al auge de la teoría, y en particular de los métodos numéricos desarrollados. Entre los problemas más destacados nos encontramos con: i) el cálculo de la respuesta electromagnética (EM) de un sistema (coeficientes de dispersión, estructura de bandas, distribución modal del campo EM…) y ii) la obtención de parámetros geométricos y/o materiales a partir de medidas de la respuesta EM. En el primer caso hablaremos de obtención de la respuesta óptica del sistema y en el segundo de diseño inverso. Salvo en sistemas relativamente sencillos el cálculo de la respuesta óptica de un sistema en la escala nanométrica supone un reto computacional tanto desde el punto de vista de la memoria requerida como del tiempo de simulación. De ahí que muchos recursos se destinen precisamente a optimizar los métodos numéricos existentes (paralelización, discretizados adaptativos…) o a la compra de ordenadores para computación intensiva. Por otro lado, los problemas de diseño inverso requieren de una gran cantidad de datos, y lo que es peor, en muchos casos es necesario establecer una gran cantidad de hipótesis sobre el sistema, generando modelos matemáticos que dificultan la resolución.

Coste computacional, trabajo con grandes cantidades de datos y capacidad de generación de modelos altamente complejos son cuestiones muy relevantes en el desarrollo tecnológico. Un conjunto de herramientas en el campo de la Inteligencia Artificial (IA) ha venido en los últimos años a solucionar problemas de este tipo. En este trabajo se utilizarán redes neuronales y *Deep Learning*, una disciplina reciente del *Machine Learning* que cada vez recibe más atención debido a su gran éxito en múltiples campos. Su poder reside en ser capaz de obtener soluciones dentro de enormes espacios de parámetros aprendiendo a partir de bases de datos más o menos completas. Prueba

de esto es que las redes neuronales pueden aprender a jugar a juegos como el Go de manera muy eficaz [4], así como aplicaciones más directas en la ciencia como la reciente computación de la estructura secundaria de proteínas ayudada por un modelo de *Deep Learning* [5], mejorando los resultados obtenidos por métodos desarrollados anteriormente. Específicamente en el campo de la Nanofotónica, las redes neuronales son una herramienta poderosa en el cálculo de estructuras nanofotónicas para tareas específicas a través del diseño inverso mencionado previamente, un problemas que se suele tratar mediante fuerza "bruta" y que, como hemos dicho, requiere muchos recursos. Las redes neuronales ya se han utilizado con anterioridad en estudios físicos y específicamente en el campo de la Nanofotónica [6]. Por ejemplo, se ha comprobado que una red neuronal puede aprender a realizar cálculos de propiedades ópticas. Este resultado se ha utilizado para aproximar el problema de dispersión de la luz en una nanopartícula multicapa [7]. En este estudio, mediante una base de datos de 50000 ejemplos calculadas mediante una simulación de Monte Carlo se entrenó una red, que después fue capaz de reproducir la dispersión de partículas que no estaban en el conjunto de entrenamiento de manera muy precisa. La diferencia con los datos de entrenamiento más cercanos apunta a que la red no memoriza simplemente y ajusta los datos, sino que abstrae algún patrón estructural en relación a la entrada y la salida de manera que generalice la física del sistema.

Este trabajo se centrará en el estudio mediante IA de un fenómeno dentro de la Nanofotónica conocido como transmisión óptica extraordinaria (EOT, de sus siglas en inglés *Extraordinary Optical Transmission*). Más adelante se describirá con más detalle en que consiste el fenómeno, pero, a modo de introducción, la EOT es un conjunto de fenómenos ópticos resonantes en los que intervienen láminas metálicas perforadas y que tienen la particularidad de que los orificios tienen tamaños inferiores a la longitud de onda de la luz incidente [8].

1.1 OBJETIVOS Y METODOLOGÍA

A lo largo de este trabajo, se estudiará primero la regresión directa de espectros de transmisión en condiciones de EOT a partir de los parámetros del medio a través de dos modelos de redes neuronales. Se estudiará la viabilidad y precisión resultante de cada modelo en función del tamaño de la base de datos necesaria para entrenar la red para obtener resultados aceptables; así como el tamaño de los modelos en sí, y, por tanto, el tiempo de computación necesario. Además, se comparará este método con la contraparte numérica utilizada, llamada Expansión Modal (ME, del inglés *Modal Expansion)*, para evaluar el ahorro en tiempo de cómputo que puede suponer el utilizar *Deep learning* en cálculos semejantes.

A continuación, se tratará el diseño inverso de redes periódicas de agujeros a partir de espectros completos de transmisión mediante redes neuronales. Se evaluará el tamaño necesario de la base de datos para recuperar los parámetros geométricos de la red, así como la precisión de las predicciones a través de los modelos propuestos.

2. INTERACCIÓN LUZ-MATERIA. TRANSMISIÓN ÓPTICA EXTRAORDINARIA.

Como se ha mencionado anteriormente, la Nanofotónica estudia la interacción de la luz con la materia a escalas nanométricas, en escalas inferiores a la longitud de onda de la luz visible.

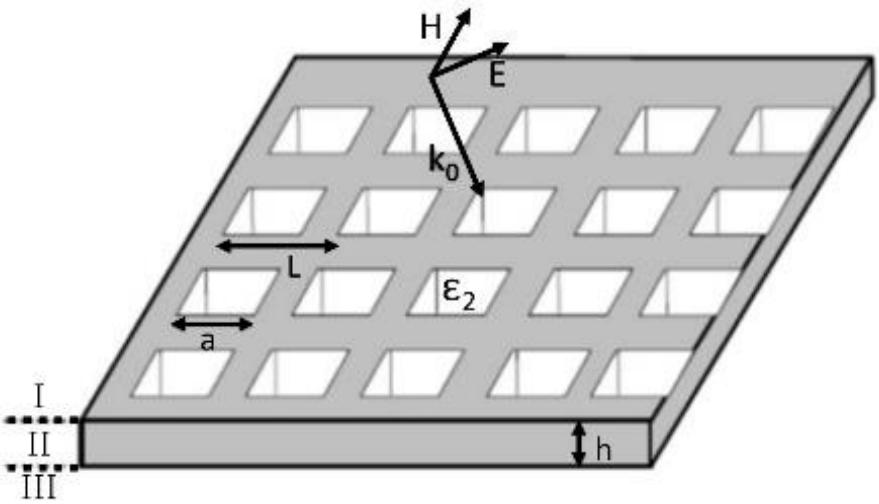

Figura 1: Esquema de la nano-estructura estudiada en el trabajo. Consiste en una red periódica (periodo, L) de agujeros cuadrados (lado del cuadrado, a) que perforan una lámina metálica delgada de espesor h. La respuesta material del metal se aproximará como si se tratara de un Conductor Perfecto (el campo EM no penetra en su interior). La lámina no está soportada ni cubierta por ningún material. Se podrá introducir un dieléctrico en su interior de constante dieléctrica ε_2. En el trabajo la iluminación será en incidencia normal.

Bethe predijo en 1944 que la eficiencia de transmisión de luz a través de una apertura circular de radio *r* en relación a la longitud de onda de la luz incidente es proporcional a $(\frac{r}{\lambda})^4$ [9]. Sin embargo, en 1998, un estudio de Ebbesen *et al*. reveló que, para ciertos materiales, configuraciones de agujeros periódicas (como la que se muestra en la Figura 1) permitían obtener valores de transmisión órdenes de magnitud superiores a los predichos teóricamente [10]. A este fenómeno se le conoce como EOT. La aparición de este tipo de transmisión muestra en el estudio de Ebbesen una fuerte dependencia con el material a través del cual se hace pasar la luz y con el ángulo de incidencia de la luz con la película de material. Estos hechos llevaron a teorizar que este fenómeno sucedía debido al acoplo de la luz con modos EM superficiales del material conocidos como plasmones de superficie (*Surface Plasmons, SP* en inglés).

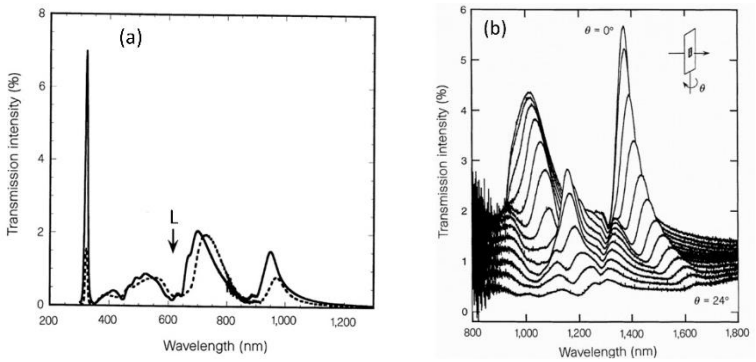

Figura 2: Figuras de la publicación original de Ebbesen en 1998 [10]*. (a) Espectro de dos configuraciones idénticas de Ag con diferente profundidad: para la línea continua, h=200 nm; para la línea discontinua, h=500 nm (este último espectro se ha multiplicado por 1.75 para una mejor comparación). Para ambas configuraciones, L= 0.6 μm y el diámetro, a = 150 nm. (b) Espectros tomados cada 2° hasta 24° para una configuración de Ag con agujeros cuadrados (L = 0.9 μm, a = 150 nm, h = 200 nm). Los espectros individuales se presentan con un incremento vertical de 0.1% los unos de los otros para mayor claridad.*

Una interfaz plana de metal-dieléctrico soporta plasmones de superficie. Sin embargo, estos no son accesibles a la iluminación externa ya que no están acoplados con modos radiativos, pues el proceso de interacción luz-materia no es capaz de conservar simultáneamente la energía y el momento. No obstante, la existencia de agujeros periódicos permite el acoplo indirecto de la luz incidente con los plasmones, gracias a la capacidad de la red de "añadir" o "quitar" momento a través del acoplamiento con los vectores de la red recíproca [11], [12].

Este fenómeno no está limitado a redes periódicas de agujeros [8]. El estudio de agujeros aislados y de estructuras con distribuciones aleatorias de agujeros sobre la superficie del metal llevó al descubrimiento de resonancias de transmisión relacionadas con la geometría del agujero, denominadas transmisión óptica extraordinaria localizada. También existen otros mecanismos para la EOT, como la transparencia inducida mediante absorbentes ópticos moleculares, que da lugar a nuevos picos de transmisión, localizados en este caso en las bandas de absorción de las moléculas.

La EOT es muy interesante para varias disciplinas científicas [8]. Siendo especialmente útil en el diseño de sensores y en aplicaciones de espectroscopia, como en la detección de compuestos o sustancias a partir de la respuesta óptica de un sistema. También se estudia, debido a la capacidad de absorber gran parte de la radiación electromagnética de estos modos, su uso en células solares, emisores térmicos o diodos orgánicos. Además, están presentes en el estudio y desarrollo de metamateriales, materiales diseñados por múltiples elementos con patrones que se repiten a escalas inferiores a la longitud de onda, presentando propiedades interesantes como dispersión parabólica. En el futuro, los fenómenos mencionados anteriormente pueden explotarse en la fabricación de polarizadores, filtros y sensores, desde el ultravioleta en adelante (energías menores).

Para el cálculo teórico de los espectros de transmisión se utilizará el método ME. Este método proporciona resultados rápidos y precisos en comparación con otros modelos numéricos ampliamente utilizados en EM computacional.

En este trabajo se ha utilizado la implementación explicada en la Ref. [13]. En este método tanto la transmisión como la reflexión se obtienen expandiendo los campos EM en las diferentes regiones del espacio, imponiendo en este caso las condiciones de contorno de metal perfecto, que es una aproximación razonable en el rango del visible e infrarrojo, y una muy buena aproximación en bandas de frecuencia menor, como el rango de terahercios, siempre y cuando los espesores del metal sean mayores que la distancia de penetración del campo EM en el metal ($\sim$25nm). Todo el espacio está dividido en tres regiones (ver Figura 1): (I) el sustrato, (II) agujeros y (III) el superestrato.

En la región (II), el campo EM se expande en términos de modos propios de guía de ondas TE y TM. Sin embargo, una buena convergencia se logra solo considerando el modo eléctrico transversal con menor decaimiento, $TE_{0,1}$ para agujeros rectangulares con tamaño sub-longitud de onda. En las regiones restantes el campo EM se expande en ondas planas, las propias del sustrato y el superestrato, que en este trabajo consideramos que son aire en ambos casos.

3. REDES NEURONALES Y *DEEP LEARNING*

La IA se puede definir como el esfuerzo por automatizar, mediante tecnología de computadores, tareas intelectuales normalmente realizadas por los seres humanos. Un tipo de inteligencia artificial lo constituyen las redes neuronales. El término red neuronal hace referencia a la neurobiología, ya que consisten en capas de lo que se llaman neuronas que se transmiten información entre ellas, como sucedería en un cerebro. Sin embargo, hay que tener claro que las redes neuronales no constituyen un modelo computacional del funcionamiento cerebral; lo que se realiza en una red neuronal no es lo que sucede en un cerebro durante el aprendizaje.

Las redes neuronales están formadas por capas de neuronas artificiales encadenadas entre sí, de manera que, a través de una serie de transformaciones matemáticas, se toma unos datos de entrada y se consiguen unos datos de salida.

Para este trabajo, se ha utilizado la biblioteca para Python *Tensorflow Keras*. Es una biblioteca de código abierto que actúa como interfaz sencilla para las redes neuronales artificiales. Permite configurar y entrenar una red en unos pocos comandos. La terminología usada a continuación hará referencia a los elementos presentes en dicha biblioteca.

Concretamente, una neurona artificial es una función matemática que recibe una serie de entradas y devuelve una única salida. Generalmente, estas entradas se multiplican por un peso w_i y se les suma un término constante *b*. Después, se suman todos estos resultados y se devuelve una salida en función del valor obtenido.

El caso más sencillo es la función escalón: se define un *bias* de activación para la neurona y se compara con el resultado de las operaciones con las entradas: si la suma definida previamente es mayor que el *bias,* la neurona devuelve un 1, y, en caso contrario, un 0. Una función de activación popular en las neuronas artificiales, que se usará a lo largo del trabajo, es la función sigmoide, que viene dada por $\sigma(z) \equiv \frac{1}{1+e^{-z}}$. También es relevante la función ReLU, del inglés *Rectified Linear Unit*, que viene dada por $f(z) \equiv \begin{cases} 0 \; si \; z < 0 \\ z \; si \; z > 0 \end{cases}$. En ambos casos anteriores, tenemos que $z = \sum w_i x_i + b$.

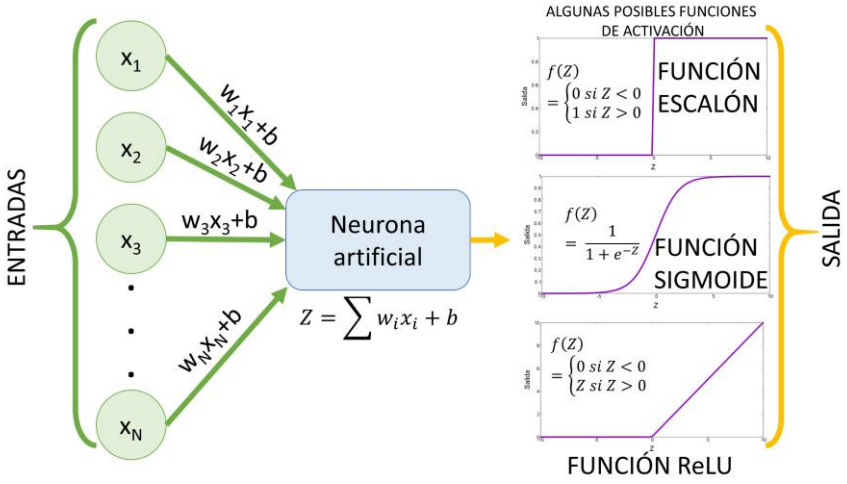

Ilustración 1: Esquema del funcionamiento de una neurona artificial. Se toman las entradas, se opera según los pesos y el desfase asignado y se introduce en una función de activación para obtener una salida. Se han representado algunas de las funciones de activación relevantes para el trabajo, de arriba abajo: la función escalón, la función sigmoidea y la función ReLU (Rectified Linear Unit).

Una capa es un conjunto de neuronas que reciben una serie de entradas y devuelven una serie de salidas. El término *Deep* en *Deep Learning* viene del hecho de que para el aprendizaje se utilizan capas consecutivas conectadas las unas con las otras: en un modelo completamente conectado (*fully connected*), también llamado denso (*dense*), las salidas cada una de las neuronas de una capa están conectadas a la entrada de cada una de las neuronas de la siguiente capa, permitiendo realizar operaciones más complejas.

Para entrenar estos modelos, se necesita una base de datos generada previamente. Esta base de datos se divide en varios subconjuntos para que el aprendizaje sea eficaz. Una gran parte se toma como conjunto de entrenamiento, que será los datos a partir de los que la red intente aprender. Una parte más pequeña se toma como conjunto de testeo, para evaluar cómo actúa la red sobre datos que no ha utilizado para entrenarse, pero de los cuales conocemos la clasificación correcta. Por último, otra parte se toma como conjunto de validación, que sirve para monitorizar cómo actúa la red sobre datos que no ha visto y modificar los hiperparámetros [14], que son elementos de la red ajenos a los pesos, como la velocidad a la que se intenta minimizar el error cometido en las predicciones. La división de la base de datos en subconjuntos previene el sobreajuste a los datos de entrenamiento; es decir, que la red aprenda a actuar sobre la base de datos, pero no sobre datos nuevos.

El entrenamiento se realiza a través de la minimización de una función de coste o *loss*, que denota el error que ha cometido el modelo en sus predicciones sobre el conjunto de entrenamiento con respecto de la clasificación correcta. Dos de las posibles funciones de coste, relevantes para el trabajo, son el Error Cuadrático Medio (ECM) y la Entropía Cruzada Binaria (*Binary Crossentropy*). El ECM viene dado por:

$$C(w,b) \equiv \frac{1}{2n}\sum_{x}\|y(x)-a\|^2$$

donde *y(x)* son los datos de entrenamiento y *a* es la salida cuando se introduce la entrada *x.*

La Entropía Cruzada Binaria viene dada por:

$$C(w,b) = -\frac{1}{n}\sum_{x}[y\cdot\ln(a)+(1-y)\cdot\ln(1-a)]$$

El aprendizaje actúa sobre los pesos w_i y las constantes b, de manera que se modifican en la dirección del gradiente de la función de coste para encontrar el mínimo. Este gradiente se calcula según el algoritmo de propagación hacia atrás (*backpropagation*), que permite calcular el gradiente de la función de coste de manera eficaz, empezando por la última capa y propagándolo hacia las primeras. La demostración del algoritmo no es excesivamente compleja, pero excede los límites de este trabajo. Se recomienda para una completa visión de este algoritmo el libro de Nielsen [15].

El método de búsqueda del mínimo se denomina optimizador. Un optimizador clásico es el Descenso por Gradiente Estocástico (SGD, *Stochastic Gradient Descent*), que calcula el gradiente para un sector aleatorio de los datos (denominado mini-remesa o, en inglés, *mini-batches*) y desplaza los parámetros entrenables (pesos y constantes) hacia el mínimo. El uso de estas pequeñas remesas de datos también ayuda a evitar mínimos locales. El tamaño de estos desplazamientos viene dado por la tasa de aprendizaje (*learning rate*). Además, se puede añadir un término de momento. Este momento mide la velocidad del cambio en las variables en lugar del valor neto. Así,

se modifican las velocidades de cambio en lugar de las variaciones en sí, y se permite llegar más eficazmente al mínimo deseado. Cada vez que se hace una pasada completa por todo el conjunto de entrenamiento, se denomina época (*epoch*).

Como medida adicional de la precisión de las predicciones, se define una función métrica que evalúa otro parámetro que sea relevante monitorizar pero que la red no intenta minimizar activamente. Por ejemplo, se puede utilizar como función de coste la Entropía Cruzada Binaria y, al mismo tiempo, evaluar el ECM, una medida directa de cuánto se aleja la predicción del espectro real en el conjunto de datos de entrenamiento.

4. CÁLCULO DE ESPECTROS DE TRANSMISIÓN ÓPTICA MEDIANTE REDES NEURONALES

Como se ha comentado en la introducción, uno de los problemas más destacados en Nanofotónica es el cálculo de la respuesta EM de un sistema (coeficientes de dispersión, estructura de campos, distribución modal del campo EM…). La dificultad que entraña resolver las ecuaciones de Maxwell en sistemas nano-estructurados se debe a las distintas escalas involucradas en el problema. Por ejemplo, los métodos numéricos que se basan en la discretización espacial y/o temporal de las nano-estructuras tienen que lidiar con el hecho de que el campo EM debe estar correctamente representado por un lado en escalas por debajo de los 25 nm (si se quiere representar el campo de un plasmón de superficie), y a su vez en escalas de varias micras, que se corresponde con el tamaño de los sistemas. Por lo tanto, los sistemas que pueden estudiarse están limitados por la memoria RAM de nuestros ordenadores. También existe un límite obvio en todos los métodos numéricos respecto del tiempo de simulación: volúmenes de simulación grandes en los que la interacción luz-materia genera fenómenos resonantes dan lugar a simulaciones que consumen mucho tiempo de computación.

Un objetivo razonable es pues utilizar las herramientas de IA con la idea de sustituir un determinado método numérico si el modelo de IA es mucho más rápido. Recientemente se estudió el diseño de sensores superconductores en el régimen óptico, que requería el cálculo de la absorción [16]. El diseño de las nanoestructuras ópticas hubiera requerido aproximadamente de 31 años para calcular los espectros necesarios con una herramienta numérica de uso habitual, mientras que con la red neuronal bastaban 15 minutos para obtener los mismos resultados. Es decir: la velocidad de la red neuronal es 6 órdenes de magnitud mayor en el cálculo de espectros que el método numérico.

Siguiendo ese trabajo, se plantea aquí un estudio sistemático de la resolución de este tipo de problemas. Para acelerar la obtención de datos hemos elegido el método ME (ver apartado 2. INTERACCIÓN LUZ-MATERIA. TRANSMISIÓN ÓPTICA EXTRAORDINARIA.). El método ME es capaz de calcular un espectro en unos 0,31 segundos de media. Veremos sin embargo que, si se trata de calcular cientos de miles de espectros, el uso de una red neuronal entrenada puede marcar la diferencia.

Estudiamos dos tipos de arquitectura de red neuronal para calcular los espectros de transmisión asociados a la EOT a partir de los parámetros geométricos del medio utilizando *Deep Learning* (ver Figura 3). Llamaremos a estos modelos: Modelo P (la entrada de la red son los parámetros geométricos – de ahí el nombre- h, L, a y ε_2; la salida el espectro al completo) y el Modelo P+λ que toma, además de los parámetros geométricos, la longitud de onda, y por lo tanto la salida solo puede ser la transmisión a esa longitud de onda. El Modelo P+λ fue desarrollado en el contexto de otro trabajo de fin de grado [16], pero aquí se estudiará en detalle. El Modelo P, más sencillo, veremos que, aun siendo peor en el cálculo de espectros, será clave en problemas de diseño inverso.

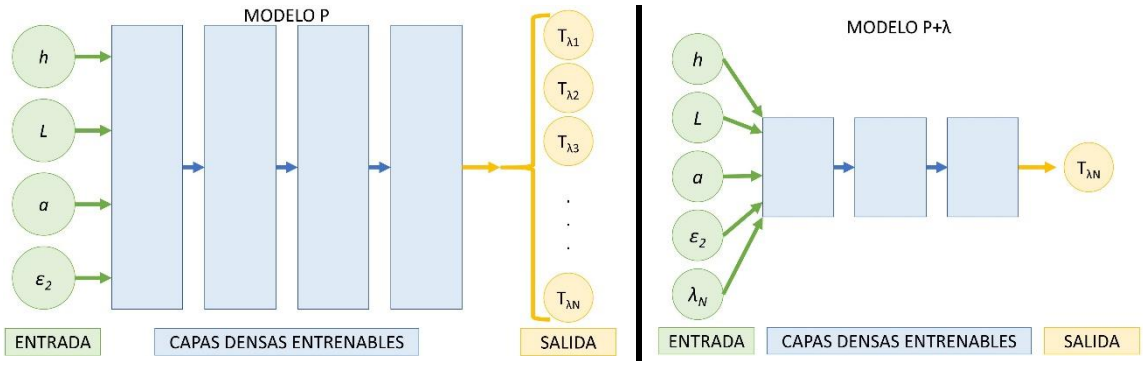

Figura 3: Esquemática del funcionamiento y topología de los dos modelos a discutir a continuación. El modelo P (izquierda) tiene 4 parámetros de entrada (h, L, a, ε_2) y como salida tiene los 1000 valores de transmisión de un espectro completo. El modelo P+ λ (derecha) tiene 5 parámetros de entrada (h, L, a, ε_2, λ), y como salida tiene el valor de transmisión para λ.

4.1 MODELO P

Como se ha dicho, el Modelo P toma cuatro parámetros (de ahí el nombre del modelo, P de parámetros) de la geometría de la red para deducir el espectro completo de transmisión: h, L, a y ε_2.

En la Figura 4 vemos la implementación en lenguaje Python del Modelo P, utilizando las librerías de Tensorflow-Keras. Como se puede ver, el modelo consta de 5 capas de neuronas, las dos primeras capas con neuronas activadas mediante la función ReLU y las dos últimas que utilizan la función sigmoide. Cada capa tiene 400 neuronas, y la salida se corresponde con la transmisión en el rango de longitudes de onda entre 700-1200 nm, en el que se han calculado utilizando el método ME 1000 valores. Esta elección de capas se traduce en 723800 parámetros a entrenar.

```
kernel_init = tf.keras.initializers.RandomNormal(mean=0.0, stddev=0.05,\
                                                 seed=None)
bias_init = tf.keras.initializers.Constant(0.5)
model=tf.keras.models.Sequential()
model.add(layers.Dense(400, activation='relu', input_shape=(neurons_in,)))
model.add(layers.Dense(400, activation='relu'))
model.add(layers.Dense(400, activation='sigmoid',\
                       kernel_initializer = kernel_init,\
                         bias_initializer = bias_init))
model.add(layers.Dense(neurons_out, activation='sigmoid',\
                       kernel_initializer = kernel_init,\
                         bias_initializer = bias_init))
opt = tf.keras.optimizers.Adam(0.0001)
model.compile(optimizer=opt, loss='mse')
```

Figura 4: Definición del Modelo P escrito en lenguaje Python, mediante las librerías de Tensorflow-Keras. Primero, se define una distribución normal para los pesos de dos de las capas. Además, se define también un tensor de constantes para el primer valor de los 'bias' de dichas capas. A continuación, se añade una capa que tiene por entrada el número de parámetros (neurons_in, en este caso 4), y tiene 400 neuronas (salida) activadas mediante la función ReLU. Se encadena una segunda capa de 400 neuronas activadas mediante la función ReLU. Las dos siguientes capas también tienen 400 neuronas, y los pesos y 'biases' se inicializan aleatoriamente según los parámetros definidos anteriormente. Se elige como optimizador Adam con una ratio de aprendizaje (velocidad a la que cambian los pesos) de 10^{-4}, y se toma como función de coste el ECM, que en la imagen aparece como 'mse' por sus siglas en inglés.

Como podemos ver, como función de coste se utilizará el ECM, que viene dado por la suma de los cuadrados de la diferencia entre la salida dada por la red y la salida correcta (ver apartado 3. REDES NEURONALES Y *DEEP LEARNING*).

Como optimizador, se usa Adam, un optimizador adaptativo (calcula una tasa de aprendizaje específica para cada parámetro) que utiliza el gradiente cuadrático para escalar la tasa de

aprendizaje, pero utiliza la media del movimiento del gradiente en lugar del gradiente en sí, utilizando así el momento a diferencia del SGD.

Los hiperparámetros seleccionados para esta arquitectura tras el estudio son: una tasa de aprendizaje de 10^{-4}, y un tamaño de mini-remesa (mini-batch) de 10 datos.

Este modelo de red neuronal se encontró siguiendo un proceso de prueba y error. Los resultados son sufrientemente satisfactorios para los objetivos previstos, pero, dicho esto, por supuesto deben existir otras topologías posibles.

Para comprobar el rendimiento del Modelo P estudiamos la relación entre el ECM, el parámetro que estamos utilizando para comprobar la validez de los *outputs* de la red, y el número de espectros con el que hemos entrenado la red. Probaremos con 1000, 3000, 5000, 7000, 9000 y 10000 espectros calculados con el método ME, y que tienen asociados tiempos de cálculo de aproximadamente 5, 15, 26, 36, 46 y 52 minutos respectivamente. En este caso, se realizarán 800 épocas, y se representa el valor del ECM como función del número de espectros dados a la red. Se puede observar que el valor del ECM tiene un mínimo cerca de 9000 espectros de entrenamiento (Figura 5).

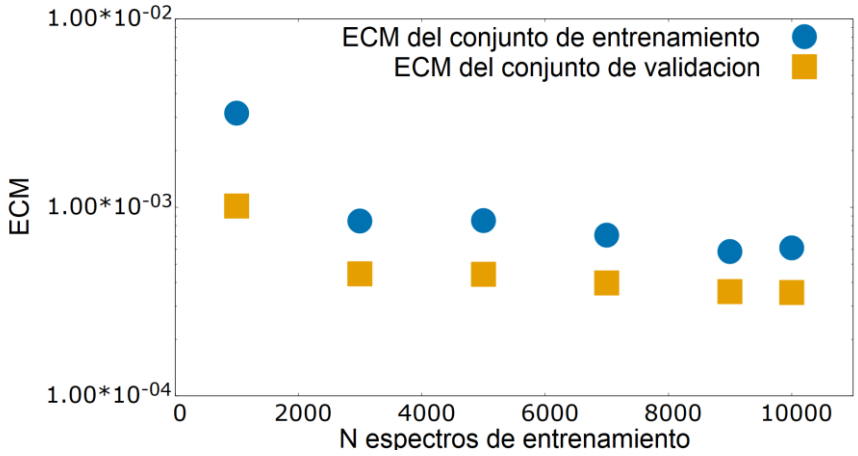

Figura 5: Estudio del ECM en función del número de espectros de entrenamiento en Modelo P. Los círculos azules corresponden al ECM del conjunto de datos de entrenamiento. Los cuadrados amarillos corresponden al ECM del conjunto de datos de validación.

Recordando que el método de cálculo de los espectros mediante el método ME tenía una cadencia de aproximadamente 0.31 segundos por espectro, encontramos que este modelo tarda unos $5.13 \cdot 10^{-4}$ segundos por espectro. Para el cálculo de estos tiempos de computación, se ha usado un entorno remoto, al cual se accede a través de la herramienta gratuita *Google Colab,* con una CPU de dos núcleos Intel Xeon con una RAM total de 12.69 GB a 2.20GHz y una GPU Nvidia Tesla K80 con una RAM de 12.8 GB a 2.50 GHz. Por lo tanto, la red calcula 600 espectros en lo que el método ME calcula un único espectro.

Podemos representar la transmitancia calculada mediante el método del método ME en función de la de la red (debería ser una recta de pendiente unidad) y un espectro completo calculado mediante el método ME frente al resultado de la red (Figura 6).

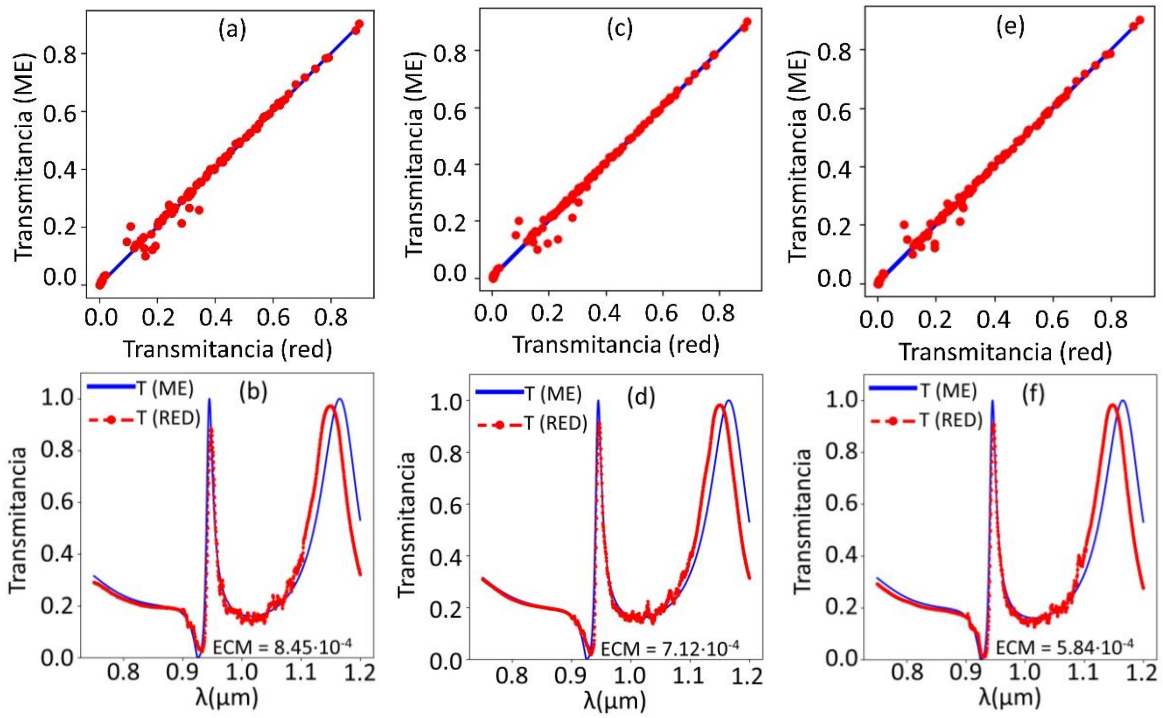

Figura 6: En la fila de arriba, comparación de la transmitancia obtenida mediante ME para 100 espectros del conjunto de datos de validación correspondientes a λ=1.14 μm con ella misma (línea continua azul) y con una red Modelo P (puntos rojos) entrenada con (a) 3000, (c) 7000 y (e) 9000 espectros completos. En la fila de abajo, comparación entre un espectro obtenido mediante ME (línea continua azul) y la red Modelo P (línea discontinua y puntos rojos) entrenada con (b) 3000 (d) 7000 y (f) 9000 espectros completos. El espectro calculado corresponde a los siguientes parámetros geométricos: h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, $\varepsilon_2 = 2.19$

Se puede observar cierto ruido en algunas zonas el desplazamiento de uno de los picos, a pesar del aparente valor pequeño del ECM. A continuación, se estudiará otro modelo para ver si es capaz de reducir el ECM y suavizar el espectro resultante.

4.2 MODELO P+λ

En el segundo modelo propuesto, la red, más reducida, se entrena para calcular un único coeficiente de transmisión a partir de 5 outputs: h, L, a, ε_2 y la longitud de onda λ a la que se está calculando la transmisión (de ahí el nombre P+λ). Debido a esto, por cada espectro de transmisión se tienen 1000 veces más datos de entrenamiento, pues cada espectro completo da un total de 1000 valores. Por lo tanto, la red necesitará menos espectros completos para alcanzar resultados similares.

Tras una experimentación previa con la topología de red, se llegó a la conclusión de que la red producía resultados satisfactorios para la siguiente configuración: 2 capas de neuronas (Figura 7), de activación mediante la función sigmoide, y el optimizador *SGD* con el método de Nesterov: el salto en el gradiente se realiza introduciendo una velocidad v en el cambio del gradiente que genera un momento, de manera que el movimiento se hace más rápido en las direcciones de mayor mejora y además ayuda a evitar los mínimos locales que puede haber en la función. En el método de Nesterov, el gradiente se calcula en una posición distinta a la actual en la simulación, aprovechando que el gradiente siempre apunta en la dirección correcta mientras que el momento puede no hacerlo, y así puede realizar correcciones antes de cometer errores al desplazarse.

```
model=tf.keras.models.Sequential()
model.add(layers.Dense(80, activation='sigmoid', input_shape=(neurons_in,)))
model.add(layers.Dense(80, activation='sigmoid'))
model.add(layers.Dense(neurons_out, activation='sigmoid'))
sgd = optimizers.SGD(lr=1.0, decay=1e-6, momentum=0.45, nesterov=True)
model.compile(optimizer=sgd,loss='binary_crossentropy',metrics=['mse'])
```

Figura 7: Definición del Modelo P+λ escrito en lenguaje Python, mediante las librerías de Tensorflow-Keras. La primera capa tiene 80 neuronas (salida) y de entrada toma el número de parámetros+λ ('neurons_in' entradas), y se activa mediante la función sigmoide. Después, se añade una segunda capa densa de 80 neuronas. La última capa tiene tantas neuronas como salidas ('neurons_out' salidas, en este caso 1, para la transmisión). Se elige un optimizador, en este caso el SGD con momento Nesterov. Por último, se compila el modelo, eligiendo como función de coste la entropía cruzada binaria y como métrica el ECM, que en la imagen aparece como 'mse' por sus siglas en inglés.

Cada capa consta de 80 neuronas, lo que da un total de 1122 parámetros. El proceso de entrenamiento es ligeramente más largo que con el primer modelo presentado, pero como veremos a continuación, es una red muy robusta y los resultados son muy satisfactorios. Un estudio más detallado de la topología de red se realizará a continuación.

En cuanto a los hiperparámetros, se ha elegido como tasa de aprendizaje 1.0 y como tamaño de las mini-remesas (*mini-batches*), se elige 64.

Vamos a estudiar primero el número de espectros que necesita esta red para obtener unos resultados que consideremos razonables. Entrenamos a la red con 100, 200, 400, 600, 800 y 1000 espectros, y observamos las diferencias entre los valores de ECM en el entrenamiento y en la validación. En todas las simulaciones se realizarán 100 épocas, 8 veces menos que en el primer modelo.

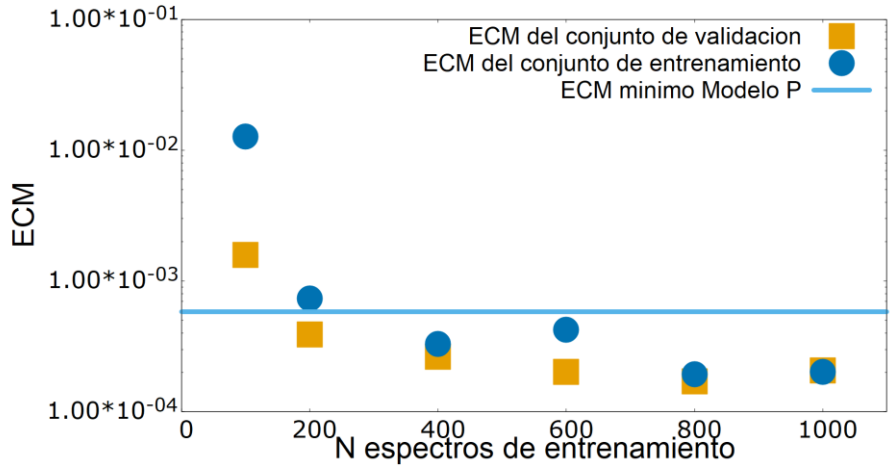

Figura 8: Estudio del ECM en función del número de espectros de entrenamiento en el Modelo P+λ, donde se puede observar una convergencia en los 800 espectros de entrenamiento. Los cuadrados amarillos corresponden al ECM calculado sobre el conjunto de validación. Los círculos azules corresponden al ECM calculado sobre el conjunto de entrenamiento. Se muestra con una línea azul el mínimo ECM obtenido en el caso del Modelo P.

Se puede apreciar cómo a partir de 600 el ECM llega a un valor con pocas variaciones, teniendo el mínimo en 800 espectros. Por lo tanto, se puede determinar que la base de datos tiene un tamaño óptimo de 800 espectros de entrenamiento.

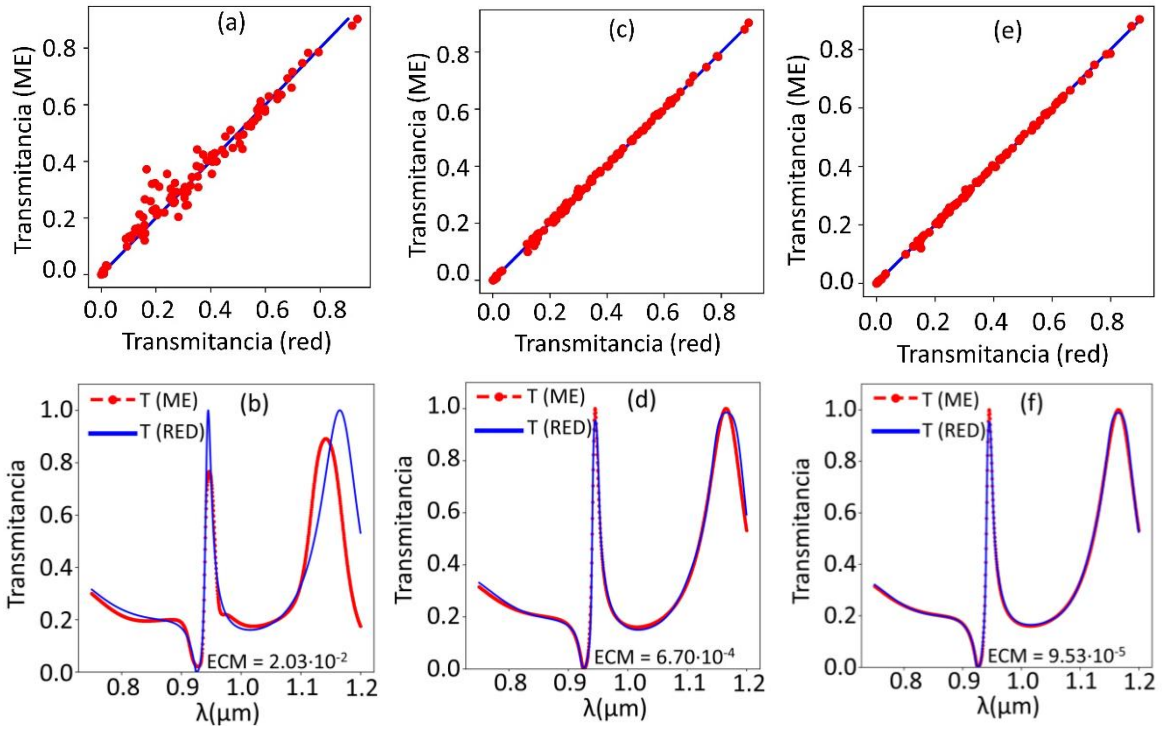

Figura 9: En la fila de arriba, comparación de la transmitancia obtenida mediante el método ME para 100 espectros del conjunto de datos de validación correspondientes a λ=1.14 μm con ella misma (línea continua azul) y con un Modelo P+λ (puntos rojos) entrenado con (a) 100 (c) 400 y (d) 800 espectros completos. En la fila de abajo, comparación entre un espectro obtenido mediante el método ME (línea continua azul) y un Modelo P+λ (línea discontinua y puntos rojos) entrenado con (b) 100 (d) 400 y (f) 800 espectros completos. El espectro calculado corresponde a los siguientes parámetros geométricos: h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, $\varepsilon_2 = 2.19$

Se puede observar (Figura 9) cómo, de manera acorde al estudio del ECM, para 100 espectros completos de entrenamiento la red no presenta un gran nivel de acierto, si bien se empieza a ver algo de aprendizaje. Para 400 espectros, el ECM ya es del orden de 10^{-4}, de manera que los espectros se ajustan bastante bien a los conseguidos mediante el método ME, y, por último, con 1000 espectros tenemos una pequeña mejora con respecto a la situación anterior en cuanto a la precisión.

Comparándolo con el método de cálculo numérico, recordamos que el método ME tardaba una media de 0.31 segundos en calcular un espectro completo. En el caso de un Modelo P+λ, se tarda una media de $5.63 \cdot 10^{-4}$ segundos. Por lo tanto, encontramos que la red calcula espectros 550 veces más rápido que el cálculo mediante el método ME, una cifra notable cuando se trata de calcular un gran número de predicciones y teniendo en cuenta que el método ME es de por sí un método de cálculo muy rápido comparado con otros simuladores.

A continuación, vamos a estudiar los resultados de esta red para distintas configuraciones y número de parámetros entrenables. Intentamos encontrar la red más sencilla que nos permitiría obtener un espectro aceptable.

Con la configuración actual de la red (2 capas de 80 neuronas) tenemos dos posibilidades: variar el número de neuronas en cada capa o variar el número de capas. Tomamos entonces varias configuraciones de 1 y 2 capas reduciendo el número de parámetros (excepto en un caso, Tabla 1) y tomamos el número de espectros de entrenamiento para el que anteriormente hemos visto que la red inicial convergía.

Número de capas ocultas	*Número de neuronas por capa*	*Número de parámetros entrenables*
1	80	642
1	100	802
1	200	1602
2	5	72
2	20	262
2	40	562
2	60	842
2	80	1122

Tabla 1: Número de parámetros entrenables en una red del Modelo P+λ según las configuraciones de 1 o 2 capas estudiadas en relación con el número de neuronas por capa.

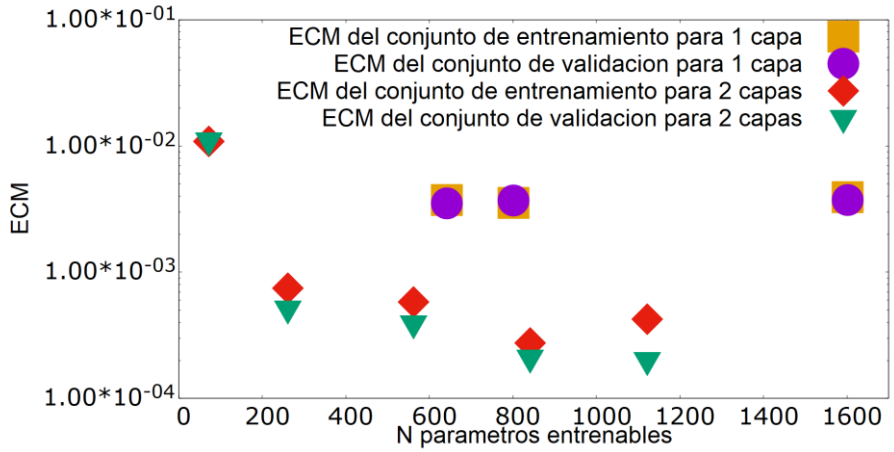

Figura 10: Estudio ECM en función del número de parámetros entrenables en la red para configuraciones con una y dos capas de neuronas según la Tabla 1. Se han utilizado 800 espectros de entrenamiento. Para el conjunto de entrenamiento, se representa en cuadrados amarillos las configuraciones de 1 capa oculta y en rombos rojos las configuraciones de 2 capas ocultas. Para el conjunto de validación, se representa en círculos morados las configuraciones de 1 capa y en triángulos verdes las configuraciones de dos capas. Se aprecia que la configuración con dos capas se vuelve rápidamente superior.

Podemos observar cómo las configuraciones con una única capa de neuronas tienen un desempeño notablemente inferior con respecto a las configuraciones con dos capas y número similar de parámetros (Figura 10). Además, se puede observar cómo, incluso para redes muy concisas, con tan solo 262 parámetros, ya se obtienen ECM del orden de 10^{-4}. Según la gráfica anterior, podemos establecer como red óptima de tipo Modelo P+λ de dos capas con 60 neuronas cada una, entrenándola 100 épocas con 800 espectros completos. No se encuentra una mejoría reseñable en la velocidad de predicción.

Retomando la representación visual que hemos realizado en estudios anteriores, se puede comprobar cómo el número de capas (la profundidad de la red neuronal), que es la piedra filosofal del *Deep Learning,* es más importante que el número de parámetros. La red no es capaz de obtener resultados satisfactorios si tiene muchos parámetros concentrados en una única capa (Figura 11), pero sin embargo, parece bastante robusta en cuanto a regresión de espectros de manera directa cuando consta de 2 neuronas y pocos parámetros (Figura 12). Es fácil observar cómo, a pesar de ser una red con 6 veces más parámetros, el espectro de la configuración 1 capa-200 neuronas se aleja más del espectro objetivo que la configuración 2 capas-20 neuronas. Esto es lógico, pues la "profundidad" de la red, el número de capas ocultas, aumenta la complejidad de las operaciones que se pueden realizar, y, por lo tanto, el nivel de abstracción que la red adquiere.

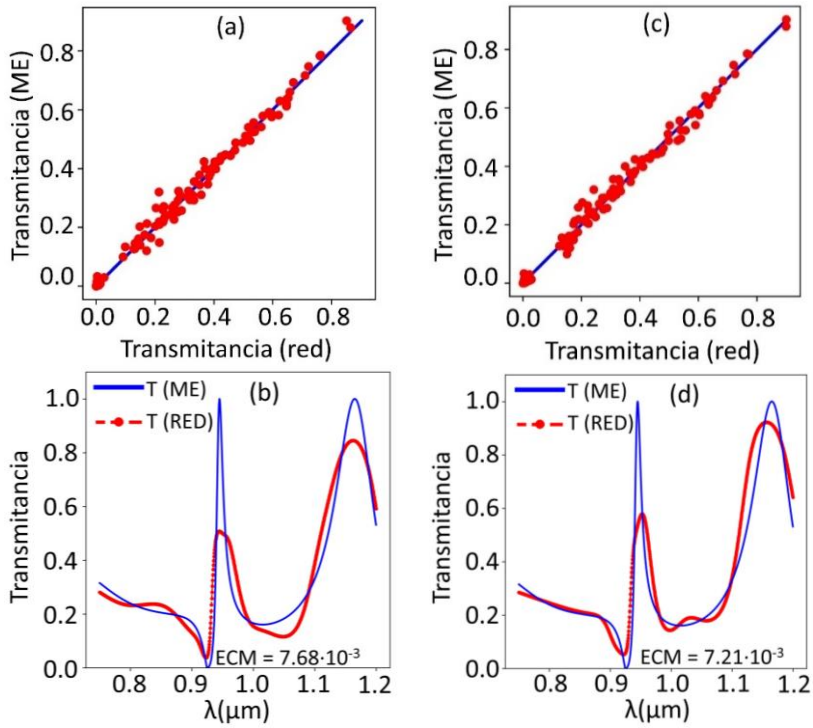

Figura 11: En la fila de arriba, comparación de la transmitancia obtenida mediante el método ME para 100 espectros del conjunto de datos de validación correspondientes a λ=1.14 μm con ella misma (línea continua azul) y con un Modelo P+λ (puntos rojos) constituido por 1 capa de (a) 100 neuronas y (c) 200 neuronas. En la fila de abajo, comparación entre el espectro obtenido mediante el método ME (línea continua azul) y un Modelo P+λ (línea discontinua y puntos rojos) constituida por 1 capa de (b) 100 neuronas y (d) 200 neuronas. El espectro calculado corresponde a los siguientes parámetros geométricos: h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, ε_2 = 2.19

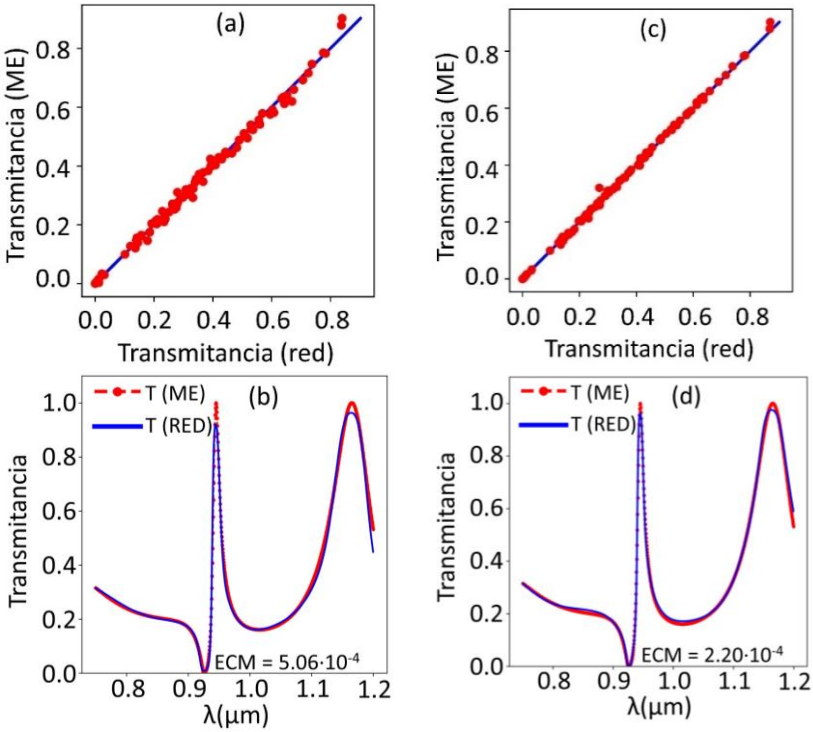

Figura 12: En la fila de arriba, comparación de la transmitancia obtenida mediante el método ME para 100 espectros del conjunto de datos de validación correspondientes a λ=1.14 μm con ella misma (línea continua azul) y el Modelo P+λ (puntos rojos) formado por (a) 2 capas de 20 neuronas y (c) 2 capas de 60 neuronas. En la fila de abajo, comparación de un espectro calculado mediante el método ME (línea continua azul) y un Modelo P+λ (puntos y línea discontinua rojos) formada por (b) 2 capas de 20 neuronas y (d) 2 capas de 60 neuronas. El espectro calculado corresponde a los siguientes parámetros geométricos: h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, ε_2 = 2.19

4.3 COMPARACIÓN ENTRE MODELOS

Una vez analizados por separado los dos modelos, podemos preguntarnos cuál es la mejor herramienta para la regresión directa de espectros de transmisión. Para ello, miraremos el número de espectros que necesita cada método para conseguir resultados satisfactorios y el número de parámetros que se necesitan.

El Modelo P consta de 723800 parámetros, mientras que la red óptima estudiada con anterioridad para el Modelo P+λ tiene 842 parámetros a entrenar. Esto supone unos 860 veces más parámetros en el primer modelo, casi 3 órdenes de magnitud de diferencia.

Por otro lado, el Modelo P converge para cerca de 9000 espectros, mientras que el Modelo P+λ, que optimiza el uso de cada espectro, converge para los 800 espectros; es decir, necesita una base de datos unas 10 veces más pequeña. Con un número comparable de espectros de entrenamiento, 1000 para ambos modelos, el ECM es 10 veces mayor para el Modelo P. Además, al representar ambos acercamientos podemos observar cómo el espectro correspondiente al Modelo P es mucho más ruidoso que el que corresponde al Modelo P+λ (Figura 13). Incluso al ampliar el tamaño de la base de datos utilizada para el Modelo P, se puede observar un mayor nivel de ruido en el espectro obtenido mediante este (Figura 14).

En cuanto a velocidad de cálculo, el Modelo P calcula un espectro en $5.13 \cdot 10^{-4}$ segundos, mientras que el Modelo P+λ calcula un espectro en $5.63 \cdot 10^{-4}$. Así, el Modelo P es un 10% más rápido, pero un 53% menos preciso en el mejor de los casos (ver diferencia en el ECM entre los dos modelos en Figura 13 y Figura 14).

Así pues, se pude concluir que el Modelo P+λ es el óptimo en relación a los recursos necesarios entre los dos estudiados para la regresión directa de espectros de transmisión a partir de los parámetros de la red, pues necesita una base de datos menor para entrenarse, tiene un valor de ECM menor y presenta menos ruido. Sin embargo, cómo veremos a continuación, el Modelo P es útil para estudiar el diseño inverso.

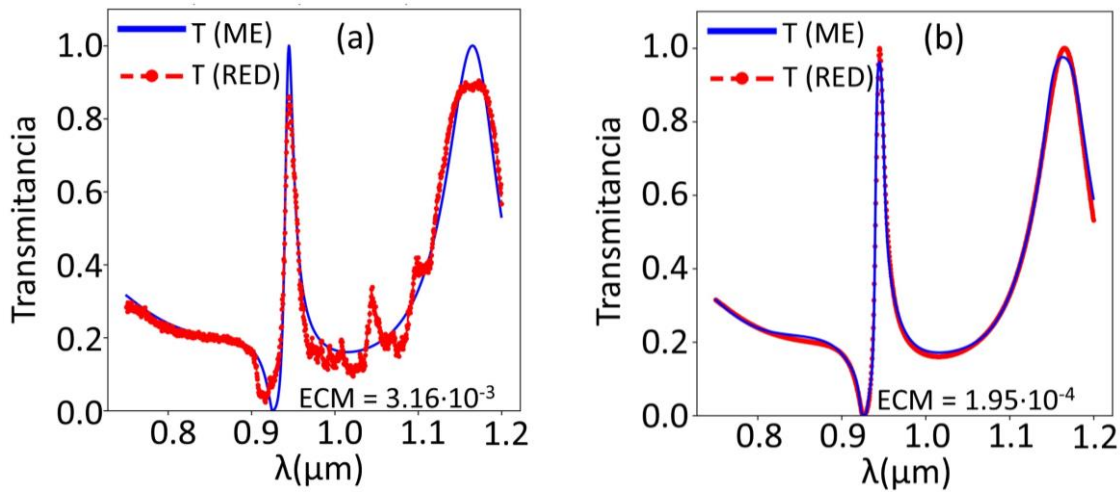

Figura 13: Comparación entre (línea continua azul) el espectro obtenido mediante el método ME y (línea discontinua y puntos rojos) (a) el Modelo P entrenado con 1000 espectros completos; (b) el Modelo P+λ entrenado con 1000 espectros completos. El espectro corresponde al calculado para los parámetros geométricos h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, $\varepsilon_2 = 2.19$

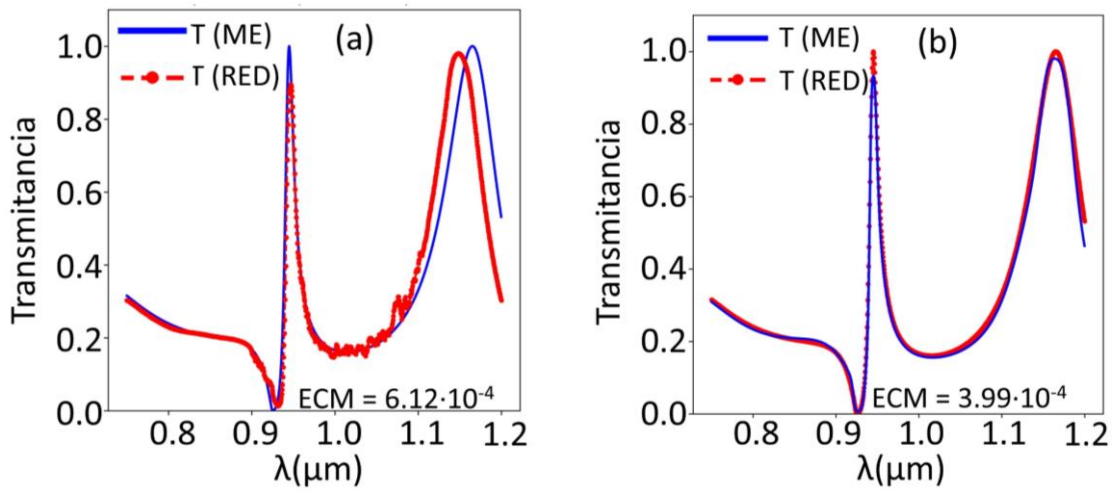

Figura 14: Comparación entre (línea continua azul) el espectro obtenido mediante el método ME y (línea discontinua y puntos rojos) (a) el Modelo P entrenado con 10000 espectros completos; (b) el Modelo P+λ entrenado con 600 espectros completos. El espectro corresponde al calculado para los parámetros geométricos h = 0.32 μm, L = 0.92 μm, a = 0.38 μm, $\varepsilon_2 = 2.19$

5. DISEÑO INVERSO EN NANOFOTÓNICA MEDIANTE REDES NEURONALES

En nuestro caso el diseño inverso consiste en, a partir de un espectro de transmisión, obtener qué parámetros geométricos conformarían una red que aproxime lo máximo posible su respuesta a la entrada.

Antes de empezar a usar redes neuronales, en el terreno del diseño inverso en Nanofotónica se usaba una mezcla de consideraciones intuitivas, métodos basados en gradientes o algoritmos como el *stimulated annealing* u optimización topológica. Sin embargo, esta clase de procesos acarrean la necesidad de disponer de un elevado poder de computación, y además, cada nuevo problema de optimización conlleva explorar el espacio de parámetros de cero [6]. El uso de redes neuronales pretende solventar este hecho, reduciendo el tiempo de computación y facilitando la búsqueda.

El diseño inverso en Nanofotónica mediante *Deep Learning* es un tema muy estudiado en los últimos años [7], [17]. En nuestro caso vamos a estudiar el aprendizaje supervisado para diseño inverso, en el que queremos que la red aprenda a asociar dos conjuntos de datos previamente clasificados (parámetros geométricos y espectros correspondientes). Sin embargo, también se ha trabajado el aprendizaje sin supervisión, en el que se intenta que las redes extraigan patrones de datos sin clasificar. Un ejemplo de esto son los sistemas GAN, formados por dos redes; una, llamada generador, toma parámetros aleatorios (ruido) y genera un espectro que debería tener los parámetros ópticos deseados; la otra, llamada discriminador, debe decidir si es un espectro real o falso. El objetivo es conseguir que el generador "engañe" al discriminador [17].

El primer acercamiento, que llamaremos *Transfered Learning* del Modelo P, se trata de un *autoencoder*; es decir, una red que se entrena para devolver los parámetros que hemos usado como entrada. En este caso, se deja una primera parte de la red entrenable, y se importa la red obtenida en el Modelo P estudiado anteriormente, que se congela. Congelar una sección implica que sus pesos y parámetros no son actualizados durante el aprendizaje (Figura 15).

En este caso, los parámetros geométricos se recuperan de las salidas de una de las capas ocultas. Al forzar que la salida sea el espectro de transmisión de entrada y que la última sección del modelo sea el Modelo P, debe haber una capa con el número de neuronas igual al número de parámetros geométricos que actúe como entrada para el Modelo P importado. Por lo tanto, la salida de esta capa oculta debería ser los parámetros que buscamos. Su funcionamiento viene limitado entonces por la capacidad del Modelo P de predecir los espectros ópticos de manera precisa.

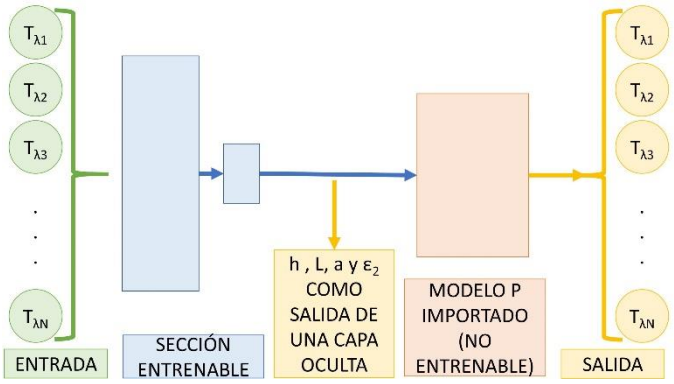

Figura 15: Esquemática del Transfered Learning del Modelo P. Como entrada y salida, tenemos un espectro completo. La primera sección en la red es entrenable, y la segunda sección es el Modelo P del apartado anterior congelada (no entrenable). La primera sección actúa como "cuello de botella", forzando una transformación de los espectros de entrada que lleve a los valores para los parámetros geométricos necesarios para que el Modelo P prediga el espectro de entrada.

El segundo modelo estudiado se trata de un *Autoencoder* Puro, donde ninguna sección está congelada. Debido a esto, al recuperar los parámetros de las salidas de una capa oculta como en el caso anterior, se debe entrenar una red auxiliar que realice una transformación de los parámetros a la geometría original (Figura 16).

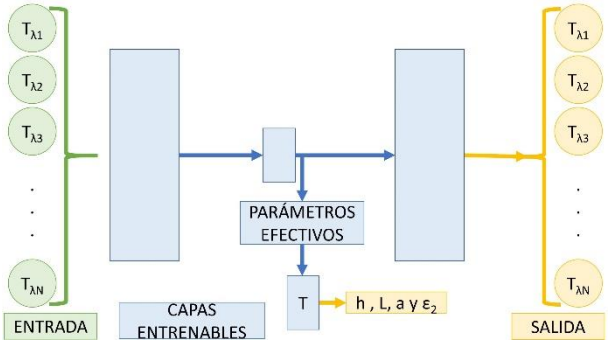

Figura 16:Esquemática del Autoencoder Puro. En este caso, toda la red es entrenable. Se obtienen unos parámetros efectivos de una de las capas ocultas de la red, que posteriormente deben someterse a una transformación mediante una red neuronal auxiliar (T) para obtener los parámetros geométricos reales.

Otro modelo estudiado consiste en una topología similar al Modelo P estudiado con anterioridad, pero intercambiando salida y entrada (Figura 17). Por lo tanto, lo denominaremos Modelo P *Flipped* (dado la vuelta). En este caso, la salida de la red es directamente los parámetros geométricos deseados.

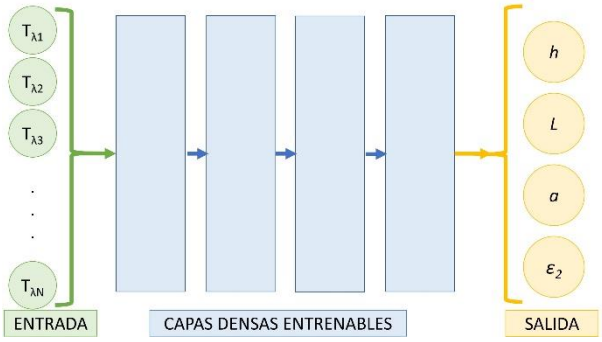

Figura 17: Esquemática del Modelo P 'Flipped'. Se utiliza una topología de red similar a la estudiada en la sección 4.1, pero con la entrada y la salida invertidas.

Para el entrenamiento se dispuso de una base de datos de 20000 espectros. Durante el entrenamiento se realizaron entre 100 y 1000 épocas. En este caso, para valorar el desempeño de los modelos, podemos representar el valor del ECM en cada uno de los parámetros para comparar los distintos acercamientos. Se puede observar cómo el Modelo P *Flipped* presenta el menor error a pesar de ser a priori el acercamiento más sencillo (Figura 18).

Además, se puede notar que el error en ε_2 y h es mayor, pues el fenómeno de transmisión óptica extraordinaria presenta una dependencia más compleja con estos parámetros que, por ejemplo, con L, que está relacionada con la posición de los mínimos de transmisión.

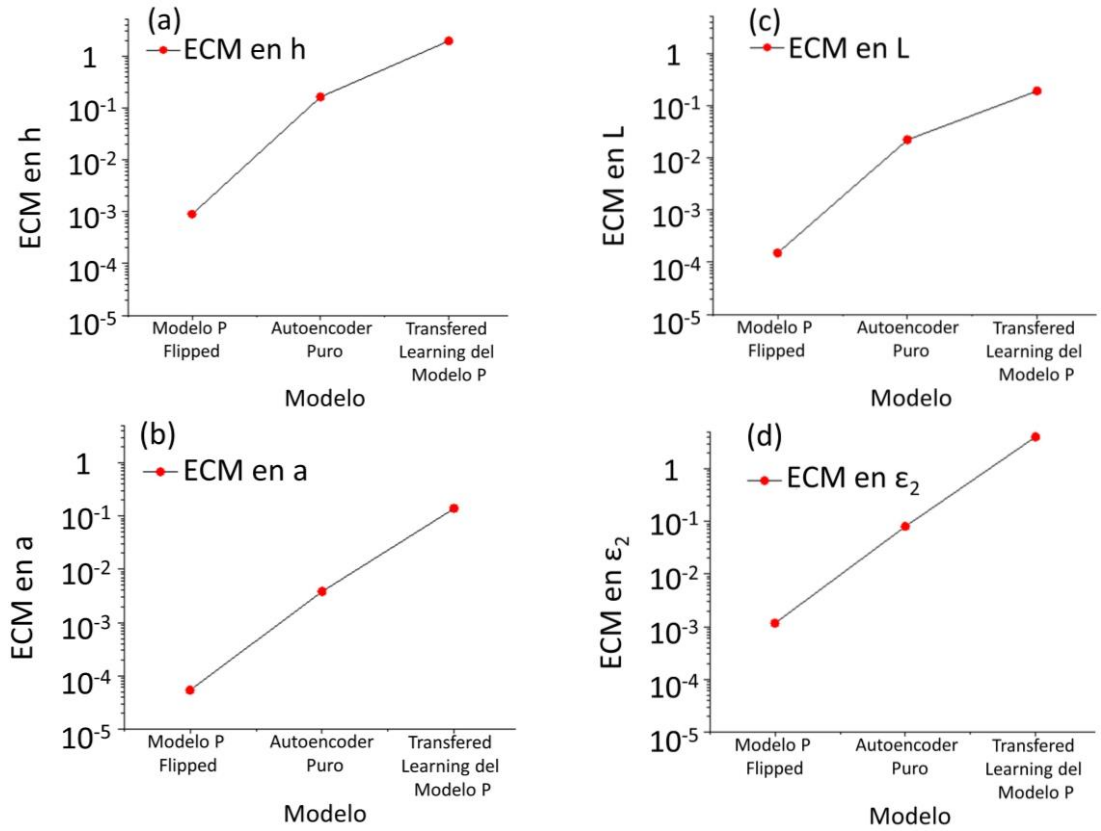

Figura 18: Representación de distintos valores del ECM hallados al predecir los parámetros geométricos usando los modelos estudiados (para cada cuadrícula, de izquierda a derecha: Modelo P 'Flipped', 'Autoencoder' Puro y 'Transfered Learning' del Modelo P). Los errores corresponden a la predicción de (a) h, (b) a, (c) L y (d) ε_2. Se puede apreciar cómo, en todos los casos, el Modelo P Flipped muestra el mínimo error. Además, se puede apreciar cómo el error en h y ε_2 es mayor que en el cálculo de a y L.

También se realizó un estudio con una base de datos menor. Si bien se nota una mejoría al aumentar el tamaño del conjunto de entrenamiento, hay que notar que el Modelo P *Flipped* funciona de manera satisfactoria para 1000 espectros, y por tanto necesita menos recursos que los otros acercamientos. A continuación, mostramos algunas predicciones de dos modelos, *Transfered Learning* del Modelo P, *Autoencoder* Puro y Modelo P *Flipped* (Figura 19) entrenados con la base de datos completa de 20000 espectros. Además, representamos un espectro de transmisión para una red *Transfered Learning* del Modelo P y el Modelo P *Flipped* (Figura 20) en estas condiciones.

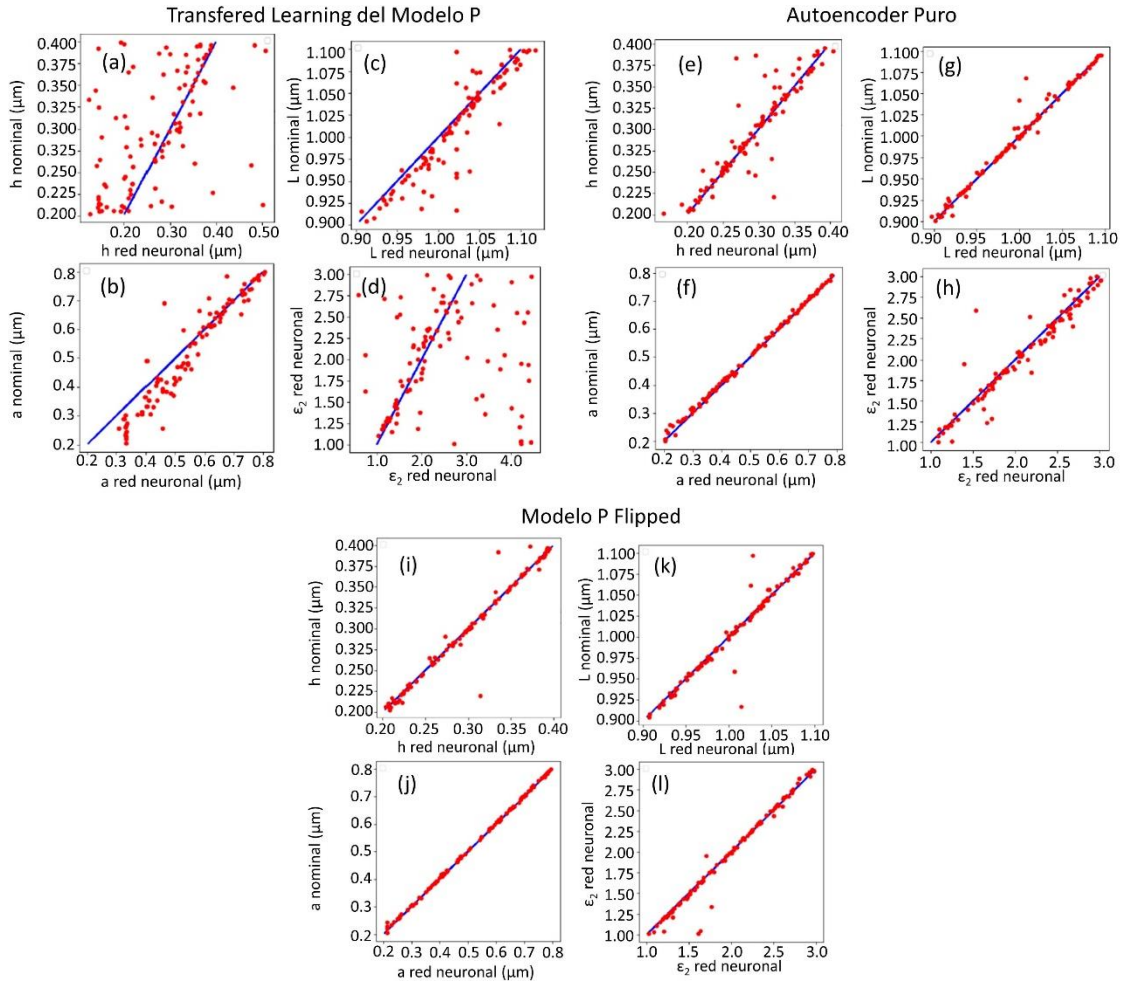

Figura 19: Representación de un mapa de dispersión para los 100 datos del conjunto de valoración con la representación nominal de los parámetros geométricos (a)(e)(i) h, (b)(f)(j) a, (c)(g)(j) L, (d)(h)(l) ε_2 frente a los predichos por la red. Se ha entrenado las distintas redes con una base de datos de 20000 espectros. (a)(b)(c)(d) corresponden al 'Transfered Learning' del Modelo P; (e)(f)(g)(h) corresponden al 'Autoencoder' Puro. (i)(j)(k)(l) corresponden al Modelo P 'Flipped'.

Aunque en lo expuesto aquí uno de los modelos parece prevalecer sobre los otros, no es necesariamente el mejor. Los otros dos modelos podrían ser ventajosos en ciertas circunstancias en las que se dispongan, por ejemplo, de más datos. La red *Transfered Learning* del Modelo P es similar a un modelo propuesto en la Ref. [18], por David Alonso González *et al.* En ese trabajo, que estudia un sistema parecido (agujeros con rugosidades), muestran que, con una base de datos mucho mayor a la utilizada en este trabajo (unas 600000 muestras), y realizando muchas más épocas (5000), la red similar a la presentada en este trabajo devuelve resultados muy precisos. Además, cabe decir que, a diferencia de los Modelos P y P+λ del apartado 4. CÁLCULO DE ESPECTROS DE TRANSMISIÓN ÓPTICA MEDIANTE REDES NEURONALES, en esta última parte de diseño inverso no se han optimizado las redes en lo referente a hiperparámetros. Hemos expuesto estos modelos con la intención de que puedan ser útiles en futuras investigaciones, en las que se debería llevar a cabo un proceso de optimización de dichos hiperparámetros.

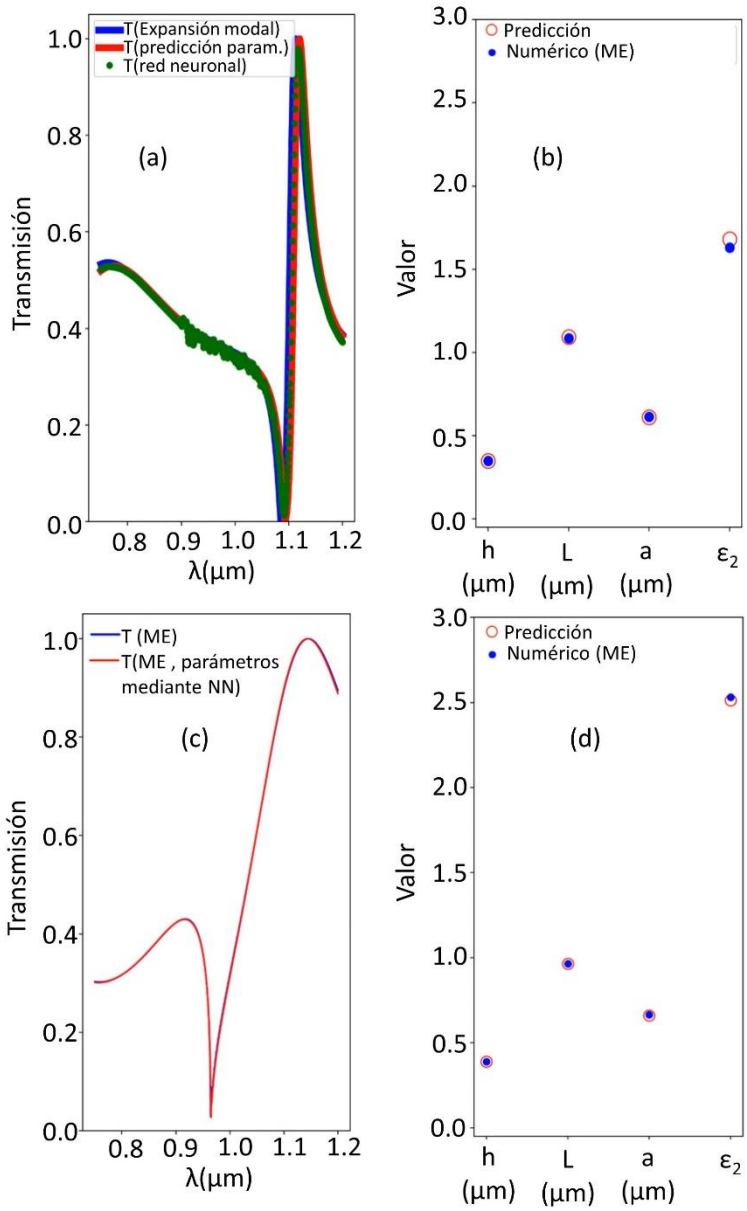

Figura 20: Estudio de las predicciones de los modelos Transfered Learning del Modelo P y Modelo P Flipped entrenados mediante una base de datos de 20000 espectros. Se presenta el espectro objetivo calculado mediante el método ME (línea azul), el espectro calculado mediante el método ME a partir de los parámetros predichos por la red (línea roja) y el espectro calculado por la red neuronal (puntos verdes) en el caso de (a) Transfered Learning del Modelo P; (c) Modelo P Flipped (no hay espectro de salida pues la salida es directamente los parámetros). El espectro en (a) corresponde a los parámetros geométricos h = 0.35 μm, L = 1.09 μm, a = 0.61 μm, ε_2 = 1.63. Los parámetros predichos por la red de Transfered Learning del Modelo P son h = 0.35 μm, L = 1.09 μm, a = 0.61 μm, ε_2 = 1.68. El espectro en (c) corresponde a los parámetros geométricos h = 0.39 μm, L = 0.96 μm, a = 0.66 μm, ε_2 = 2.53. Los parámetros predichos por la red P Flipped son h = 0.39 μm, L = 0.96 μm, a = 0.66 μm, ε_2 = 2.51. Se muestra además una comparativa muestra una comparativa entre el valor numérico de los parámetros objetivo (puntos azules) y el predicho mediante la red (círculos rojos) para los modelos (b) Transfered Learning del Modelo P y (d) Modelo P Flipped.

6. CONCLUSIONES

El objetivo del trabajo era estudiar la viabilidad de las redes neuronales para sustituir herramientas de simulación numéricas que consuman excesivo tiempo de computación y la posibilidad de estas redes de actuar como herramienta primaria en el desarrollo de software para diseño inverso en Nanofotónica.

En el estudio de simulación directa, en el caso del cálculo de espectros de transmisión a través de redes neuronales, se han estudiado dos modelos. A pesar de que el método numérico que se ha utilizado en las simulaciones teóricas previas es rápido con respecto a otros métodos de simulación, se ha encontrado que ambos modelos suponen una mejora de al menos 500 veces en la velocidad de cálculo. La precisión de estos modelos es aceptable, especialmente en el caso del Modelo P+λ, que, como hemos visto, se puede optimizar para a partir de una red sencilla y una base de datos no muy extensa recuperar resultados con un ECM del orden de 10^{-4} y sin presentar apenas ruido.

Por otro lado, se han estudiado tres posibles acercamientos al diseño inverso a través de redes neuronales. Los resultados son satisfactorios, apuntando a que efectivamente las redes neuronales son una herramienta útil a la hora de abordar problemas de diseño inverso. A pesar de que con la base de datos tratada el modelo que más precisión presenta es el más sencillo, otros estudios apuntan a que incrementar en unos órdenes de magnitud esta base de datos podría revelar mejoras en otros acercamientos. Se deberá, pues, hacer un estudio más extenso de las posibilidades que ofrece cada modelo, así como de la posible optimización de cada uno.

7. BIBLIOGRAFÍA

[1] S. V. Gaponenko, *Introduction to Nanophotonics*. Cambridge Universitty Press.

[2] P. N. Prasad, *Nanophotonics*. John Wiley & Sons.

[3] R. Kirchain and L. Kimerling, "A roadmap for nanophotonics," *Nat. Photonics*, vol. 1, no. 6, pp. 303–305, 2007, doi: 10.1038/nphoton.2007.84.

[4] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science (80-.).*, vol. 362, no. 6419, pp. 1140–1144, 2018, doi: 10.1126/science.aar6404.

[5] R. AlGhamdi, A. Aziz, M. Alshehri, K. R. Pardasani, and T. Aziz, "Deep learning model with ensemble techniques to compute the secondary structure of proteins," *J. Supercomput.*, vol. 77, no. 5, pp. 5104–5119, 2021, doi: 10.1007/s11227-020-03467-9.

[6] P. R. Wiecha, A. Arbouet, C. Girard, and O. L. Muskens, "Deep learning in nano-photonics: inverse design and beyond," *Photonics Res.*, vol. 9, no. 5, p. B182, 2021, doi: 10.1364/prj.415960.

[7] J. Peurifoy *et al.*, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, no. 6, pp. 1–8, 2018, doi: 10.1126/sciadv.aar4206.

[8] S. G. Rodrigo, F. De León-Pérez, and L. Martín-Moreno, "Extraordinary Optical Transmission: Fundamentals and Applications," *Proc. IEEE*, vol. 104, no. 12, pp. 2288–2306, 2016, doi: 10.1109/JPROC.2016.2580664.

[9] H. A. Bethe, "Theory of Diffraction by Small Holes," *Phys. Rev.*, vol. 66, no. 7–8, pp. 163–182, 1944, doi: 10.1103/PhysRev.66.163.

[10] T. W. Ebbesen, H. J. Lezec, H. F. Ghaemi, T. Thio, and P. A. Wolff, "Extraordinary optical transmission through sub-wavelength hole arrays," *Nature*, vol. 391, no. 6668, pp. 667–669, Feb. 1998, doi: 10.1038/35570.

[11] L. Martín-Moreno *et al.*, "Theory of extraordinary optical transmission through subwavelength hole arrays," *Phys. Rev. Lett.*, vol. 86, no. 6, pp. 1114–1117, 2001, doi: 10.1103/PhysRevLett.86.1114.

[12] C. Genet and T. W. Ebbesen, "Light in tiny holes," *Nature*, vol. 445, no. 7123, pp. 39–46, 2007, doi: 10.1038/nature05350.

[13] L. Martín-Moreno and F. J. García-Vidal, "Minimal model for optical transmission through holey metal films," *J. Phys. Condens. Matter*, vol. 20, no. 30, 2008, doi: 10.1088/0953-8984/20/30/304214.

[14] F. Chollet, *Deep Learning with Python*. 2021.

[15] M. Nielsen, *Neural Networks and Deep Learning*. 2019.

[16] C. P. A. Marta Sánchez Casi, Sergio Gutiérrez Rodrigo, Carlos Pobes "Diseño basado en redes neuronales de sensores superconductores para nanofotonica y aplicaciones cuanticas," TFG 2020 del Grado de Físicas de la Universidad de Zaragoza, https://zaguan.unizar.es/record/98148?ln=es.

[17] S. So, T. Badloe, J. Noh, J. Rho, and J. Bravo-Abad, "Deep learning enabled inverse design in nanophotonics," *Nanophotonics*, vol. 9, no. 5, pp. 1041–1057, 2020, doi: 10.1515/nanoph-2019-0474, arXiv: 2106.12898v1 [physics.optics].

[18] D. Alonso-González, M. Frising, F. Prins, and J. Bravo-Abad, “A deep learning approach to resonant light transmission through single subwavelength apertures,” pp. 1–4, 2021.