UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

Prediction of transmittance and reflectance of a dielectric mirror using neural networks

Camilo José Naranjo Cerón

Física

Trabajo de titulación presentado como requisito

para la obtención del titulo de

Licenciado en Física

May 19, 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Camilo José Naranjo Cerón

Nombre del profesor, Titulo académico: Melissa Infusino, PhD. Daniel Riofrío, PhD

May 19, 2024

© Derechos de Autor

Nombres y apellidos:	Camilo José Naranjo Cerón
Código:	00212632
Cédula de Identidad:	172691000-1
Lugar y fecha:	May 19, 2024

ACLARACIÓN PARA LA PUBLICACIÓN

UNPUBLISHED DOCUMENT

Agradecimientos

Este trabajo es el resultado de muchos esfuerzos en conjunto. En primer lugar, quisiera agradecer a mi familia: a mis padres, hermano y mascotas que siempre han sido un soporte fundamental para mí; todo lo que he logrado es gracias a ellos. Agradezco a mis profesores Melissa y Daniel quienes fueron promotores de este proyecto y me permitieron ser parte de él. Agradezco a Christian, por su guía y los consejos que me ha dado, sin ellos este proyecto habría sido aún más difícil. A mis amigos Pablo, Jorge, Enrique, Estefanía, Xavier y Andrea por la amistad que hemos formado, este es el mayor regalo que he podido tener. Finalmente, agradezco a todas las personas que me han acompañado durante toda la carrera.

Resumen

En los últimos años las áreas de ópitca y fotónica han hecho uso extensivo de las redes neuronales para estudiar la respuesta óptica de distintos dispositivos. En este trabajo se desarolló una red neuronal capaz de predecir la reflectancia y transmitancia de 3 espejos dieléctricos de 4, 8 y 12 capas compuestos por dioxido de silicio (SiO_2) y sulfuro de zinc (ZnS). El modelo fue capaz de predecir dichos parámetros con errores inferiores al 1%.

Palabras clave: *espejos dieléctricos, transmitancia, reflectancia, redes neuronales, red neuronal totalmente conectada, red neuronal recurrente*

Abstract

In recent years, the areas of optics and photonics have made extensive use of neural networks to study the optical response of different devices. In this work, a neural network was developed capable of predicting the reflectance and transmittance of 3 dielectric mirrors of 4, 8 and 12 layers composed of silicon dioxide (SiO_2) and zinc sulfide (ZnS). The model was able to predict these parameters with errors of less than 1%. Keywords: *dielectric mirrors, transmittance, reflectance, neural networks, fully connected neural network, recurrent neural network*

Contents

1 Introduction **12**

1.1 Dielectric Mirrors 12

1.2 Neural Networks 19

1.2.1 Fully Connected Neural Networks 25

1.2.2 Recurrent Neural Networks 26

2 Methods **30**

2.1 Simulation and Data Set 30

2.2 Neural Network 31

3 Results **33**

4 Conclusion **37**

5 Appendix **39**

5.1 Code 39

Bibliografía **39**

List of Tables

1.1 Definition of Step Function, Sigmoid, Tanh and ReLU 21

3.1 Model evaluation metrics at the last epoch 33

List of Figures

1.1 Diagram of the incident plane 13

1.2 Reflection and transmittance of the light in a single layer system . . 14

1.3 Light propagation trough a multilayer system 15

1.4 Diagram of how the perceptron works 20

1.5 Plots of the four most used activation functions 22

1.6 Diagram of a simple FCNN of one hidden layer of two neurons . . . 26

1.7 Schematic of a dummy RNN, layer **z** has recurrent connections . . . 27

1.8 Diagram of the project 29

2.1 Flow diagram of the simulation process, training and testing of the Neural Network 31

2.2 Diagram representation of the neural network used in the project, the last layer represent 4 dimensional prediction vector which consist of S and P transmittance and reflectance 32

3.1 Evolution of R^2 for predictions during all epochs 34

3.2 Comparison of the predicted and the simulated values 35

3.3 Predicted and simulated P reflection for all three geometries 36

3.4 Evolution of the mean square error (MSE) and the validation for all the three experiments 36

Chapter 1

Introduction

1.1 Dielectric Mirrors

Dielectric mirrors are a type of mirror composed by periodic layers of dielectric materials. Layers are stacked into pairs of high and low refractive indices. This characteristic allows dielectric mirror to have some importants applications. For example, the mirror could be set to reflect the light at a certain wavelength while other rays are simply trasmited [1]. Also, it could be used as a high reflective mirror with a reflectivity over 95% [2].In addition, it is possible to build up hot and cold mirrors using this type of [3] [4] [5]. Hot mirrors transmit all the visible spectrum while reflecting the infrared light. On the other hand, cold ones have the opposite effect. These properties are why dielectric mirrors are a hot-take research field.

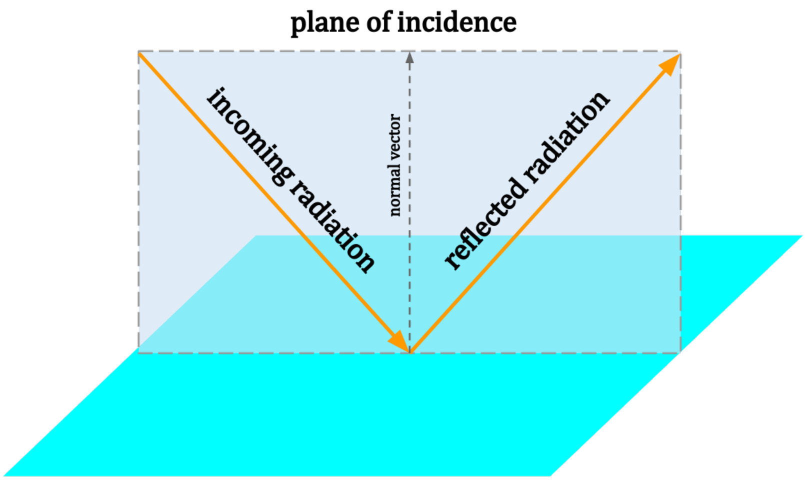


Figure 1.1: Diagram of the incident plane

Also, it is important to define the polarization of light since we will use this in this section to explain how it works dielectric mirrors. As we know light is composed by the electric field and magnetic field. Both fields are normal to each other. When light impacts an element we define an incident plane which is formed by the normal vector of the surface and the ray of incidence. We say that light is p polarized if the electric field is parallel to the incident plane. On the other hand, light is s polarized if the electric field is normal to the incident plane. If electric field is neither p-polarized nor s-polarized we say that the ray is not polarized.

With this we can understand how mirrors works. First, we first consider a single slab system [6] [7] . Here the reflection angle and the transmittance angle relates each other trough the Snell's Law.

$$n_0 \sin \theta_0 = n_1 \sin \theta_1 \tag{1.1}$$

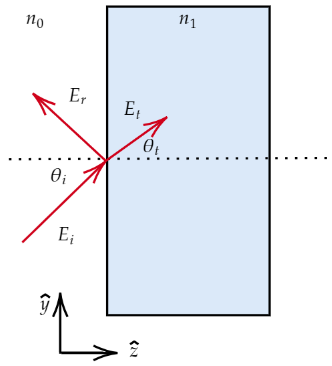


Figure 1.2: Reflection and transmittance of the light in a single layer system

Equation (1.1) still works if we consider a multilayer system since the refraction of the n-layer does not depend of its neighbourhoods. Mathematically, the follow relationship is followed.

$$n_0 \sin\theta_0 = n_1 \sin\theta_1 = ... = n_N \sin\theta_N \tag{1.2}$$

Due to Maxwell equations the parallel components of the electric field **E** and magnetic field **B** must be continuous in every side of the interface. It is important to remark that continuity equations will change if light is p-polarized or s-polarized. Firstly, we are going to dealing with the p case, then the process is similar for the s-case. According to Figure 1.3 the equations of continuity for p-polarized light are

$$\cos\theta_j \left(E^{(p)}_{j\rightarrow} e^{ik_j d_j \cos\theta_j} + E^{(p)}_{j\leftarrow} e^{ik_j d_j \cos\theta_j} \right) = \cos\theta_{j+1} \left(E^{(p)}_{j+1\rightarrow} + E^{(p)}_{j+1\leftarrow} \right) \tag{1.3}$$

$$n_j \left(E^{(p)}_{j\rightarrow} e^{ik_j d_j \cos\theta_j} - E^{(p)}_{j\leftarrow} e^{ik_j d_j \cos\theta_j} \right) = n_{j+1} \left(E^{(p)}_{j+1\rightarrow} + E^{(p)}_{j+1\leftarrow} \right) \tag{1.4}$$

We add the phase factor $k_j d_j \cos\theta_j$ because the ray acquires a phase every time

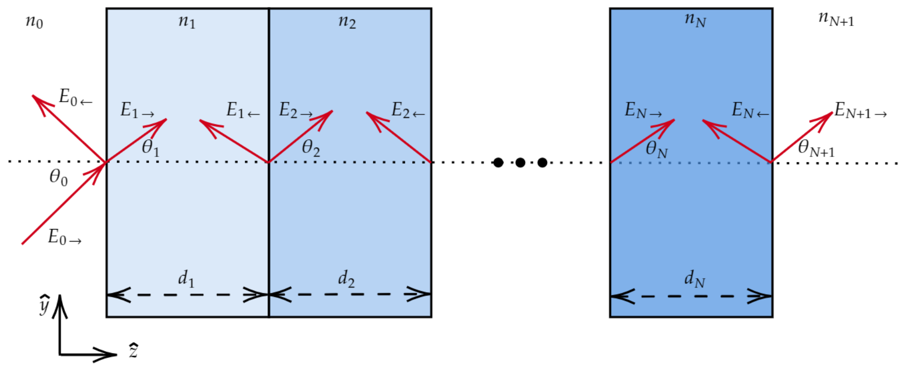


Figure 1.3: Light propagation trough a multilayer system

it traverse a layer of thickness d_j. The right-hand side does not have this factor since the phase it is analyzed in the $j+1$ equation.

In the final interface the set of equations is

$$\cos\theta_N\left(E^{(p)}_{N\rightarrow}e^{ik_N d_N\cos\theta_N}+E^{(p)}_{N\leftarrow}e^{ik_N d_N\cos\theta_N}\right)=\cos\theta_{N+1}E^{(p)}_{N+1\rightarrow} \tag{1.5}$$

$$n_N\left(E^{(p)}_{N\rightarrow}e^{ik_N d_N\cos\theta_j}-E^{(p)}_{N\leftarrow}e^{ik_N d_N\cos\theta_N}\right)=n_{N+1}E^{(p)}_{N+1\rightarrow} \tag{1.6}$$

We do not consider the term $E^{(p)}_{N+1\leftarrow}$ because there is not a ray reflected in the final medium.

We can write equations (1.3) (1.4) in a matrix form

$$\begin{pmatrix}\cos\theta_j e^{i\beta_j} & \cos\theta_j e^{i\beta_j}\\ n_j e^{i\beta_j} & -n_j e^{i\beta_j}\end{pmatrix}\begin{pmatrix}E^{(p)}_{j\rightarrow}\\ E^{(p)}_{j\leftarrow}\end{pmatrix}=\begin{pmatrix}\cos\theta_{j+1} & \cos\theta_{j+1}\\ n_{j+1} & -n_{j+1}\end{pmatrix}\begin{pmatrix}E^{(p)}_{j+1\rightarrow}\\ E^{(p)}_{j+1\leftarrow}\end{pmatrix} \tag{1.7}$$

Where

$$\beta_j=\begin{cases}0 & j=0\\ k_j d_j\cos\theta_j & 1\leq j\leq N\end{cases} \tag{1.8}$$

We can multiple by the inverse of the matrix of the left-hand side of the equation this arrives to

$$\begin{pmatrix} E^{(p)}_{j\rightarrow} \\ E^{(p)}_{j\leftarrow} \end{pmatrix} = \begin{pmatrix} \cos\theta_j e^{i\beta_j} & \cos\theta_j e^{i\beta_j} \\ n_j e^{i\beta_j} & -n_j e^{i\beta_j} \end{pmatrix}^{-1} \begin{pmatrix} \cos\theta_{j+1} & -\cos\theta_{j+1} \\ n_{j+1} & -n_{j+1} \end{pmatrix} \begin{pmatrix} E^{(p)}_{j+1\rightarrow} \\ E^{(p)}_{j+1\leftarrow} \end{pmatrix} \tag{1.9}$$

If we substitute $j = 0$ in equation(1.9) the right hand side turns on the equations for $j = 1$, but we can derive the equation for $j = 1$ using the same linear transformation. This process could be done recursively until the last layer i.e. $j = N + 1$. Hence the final equation for the entire system is

$$\begin{pmatrix} E^{(p)}_{0\rightarrow} \\ E^{(p)}_{0\leftarrow} \end{pmatrix} = \begin{pmatrix} \cos\theta_0 & \cos\theta_0 \\ n_0 & -n_0 \end{pmatrix}^{-1} \left(\prod_{j=1}^{N} M^{(p)}_j\right) \begin{pmatrix} \cos\theta_{N+1} & \cos\theta_{N+1} \\ n_{N+1} & -n_{N+1} \end{pmatrix} \begin{pmatrix} E^{(p)}_{N+1\rightarrow} \\ 0 \end{pmatrix} \tag{1.10}$$

Where

$$M^{(p)}_j = \begin{pmatrix} \cos\theta_j & \cos\theta_j \\ n_j & -n_j \end{pmatrix} \begin{pmatrix} \cos\theta_j e^{i\beta_j} & \cos\theta_j e^{i\beta_j} \\ n_j e^{i\beta_j} & -n_j e^{i\beta_j} \end{pmatrix}^{-1} \tag{1.11}$$

$$= \begin{pmatrix} \cos\beta_j & -i\sin\beta_j \cos\theta_j / n_j \\ -i n_j \sin\beta_j / \cos\theta_j & \cos\beta_j \end{pmatrix} \tag{1.12}$$

We can divide the entire equation by the incident field $E^{(p)}_{0\rightarrow}$ to obtain

$$\begin{pmatrix} 1 \\ E^{(p)}_{0\leftarrow}/E^{(p)}_{0\rightarrow} \end{pmatrix} = A^{(p)} \begin{pmatrix} E^{(p)}_{N+1\rightarrow}/E^{(p)}_{0\rightarrow} \\ 0 \end{pmatrix} \tag{1.13}$$

The term $A^{(p)}$ is

$$A^{(p)} = \frac{1}{2n_0 \cos\theta_0} \begin{pmatrix} n_0 & \cos_0 \\ n_0 & -\cos_0 \end{pmatrix} \left(\prod_{j=1}^{N} M_j^{(p)} \right) \begin{pmatrix} \cos_{N+1} & 0 \\ n_{N+1} & 0 \end{pmatrix} \tag{1.14}$$

$$= \begin{pmatrix} a_{11}^{(p)} & a_{12}^{(p)} \\ a_{21}^{(p)} & a_{22}^{(p)} \end{pmatrix} \tag{1.15}$$

Here we change the second column of the last matrix by a zero column since in the multiplication by the vector the bottom zero element cancels the entire column.

If we compute matrix $A^{(p)}$ it is simple to reach the transmittance and reflectance terms.

$$t_p = \frac{E_{N+1\rightarrow}^{(p)}}{E_{0\rightarrow}^{(p)}} = \frac{1}{a_{11}^{(p)}} \tag{1.16}$$

$$r_p = \frac{E_{N+1\leftarrow}^{(p)}}{E_{0\leftarrow}^{(p)}} = \frac{a_{21}^{(p)}}{a_{11}^{(p)}} \tag{1.17}$$

So the main goal on every multilayer system it is to compute equation (1.14) since the information of all the layers it is contained in a very simple 2×2 matrix.

The process to derive the matrices for s-polarized light is relatively similar. We start from the equations of continuity.

$$\begin{pmatrix} e^{i\beta_j} & e^{-i\beta_j} \\ \cos\theta_j e^{i\beta_j} & -\cos\theta_j e^{-i\beta_j} \end{pmatrix} \begin{pmatrix} E_{j\rightarrow}^{(s)} \\ E_{j\leftarrow}^{(s)} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \cos\theta_{j+1} n_{j+1} & -\cos\theta_{j+1} n_{j+1} \end{pmatrix} \begin{pmatrix} E_{j+1\rightarrow}^{(s)} \\ E_{j+1\leftarrow}^{(s)} \end{pmatrix} \tag{1.18}$$

We begin from the matrix form because it is not worthy to rewrite again all the previous steps. If we take the inverse matrix of the left-hand side of the equations we obtain

$$\begin{pmatrix} E^{(s)}_{j\rightarrow} \\ E^{(s)}_{j\leftarrow} \end{pmatrix} = \begin{pmatrix} e^{i\beta_j} & e^{-i\beta_j} \\ \cos\theta_j e^{i\beta_j} & -\cos\theta_j e^{-i\beta_j} \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ \cos\theta_{j+1} n_{j+1} & -\cos\theta_{j+1} n_{j+1} \end{pmatrix} \begin{pmatrix} E^{(s)}_{j+1\rightarrow} \\ E^{(s)}_{j+1\leftarrow} \end{pmatrix} \tag{1.19}$$

From here is easy to notice that the $M_j^{(s)}$ matrix is

$$M_j^{(s)} = \begin{pmatrix} 1 & 1 \\ \cos\theta_j n_j & -\cos\theta_j n_j \end{pmatrix} \begin{pmatrix} e^{i\beta_j} & e^{-i\beta_j} \\ \cos\theta_j e^{i\beta_j} & -\cos\theta_j e^{-i\beta_j} \end{pmatrix}^{-1} \tag{1.20}$$

$$= \begin{pmatrix} \cos\beta_j & -i\sin\beta_j / n_j \cos\theta_j \\ -in_j \sin\beta_j \cos\theta_j & \cos\beta_j \end{pmatrix} \tag{1.21}$$

Hence the equation of $A^{(s)}$ is

$$A^{(s)} = \frac{1}{2n_0\cos\theta_0} \begin{pmatrix} n_0\cos\theta_0 & 1 \\ n_0\cos\theta_0 & -1 \end{pmatrix} \left(\prod_{j=1}^{N} M_j^{(s)} \right) \begin{pmatrix} 1 & 0 \\ n_{N+1}\cos\theta_{N+1} & 0 \end{pmatrix} \tag{1.22}$$

$$= \begin{pmatrix} a^{(s)}_{11} & a^{(s)}_{12} \\ a^{(s)}_{21} & a^{(s)}_{22} \end{pmatrix} \tag{1.23}$$

Similarly to equations (1.16) (1.17) the coefficients of transmittance and reflectance are

$$t_s = \frac{E^{(s)}_{N+1\rightarrow}}{E^{(s)}_{0\rightarrow}} = \frac{1}{a^{(s)}_{11}} \tag{1.24}$$

$$r_s = \frac{E^{(s)}_{N+1\leftarrow}}{E^{(s)}_{0\leftarrow}} = \frac{a^{(s)}_{21}}{a^{(s)}_{11}} \tag{1.25}$$

The previous equations works for any kind of multilayer system. In the case of a dielectric mirror they consists of a periodic configuration of high and low refractive indices. So the product of equation (1.10) turns

$$\prod_{j=1}^{2k} M_j = (M_H M_L)^k \tag{1.26}$$

Here $2k = N$ since dielectric mirror are in pairs high and low layers.

1.2 Neural Networks

We as humans are constantly receiving a lot of inputs of our environment. These inputs are recollected by our 5 senses and then they are process in the brain. Finally, we make a decision. This decision could be based on many thing such as experience, analysis, improvisation, reflexes, instincts, and so on. The main point here is that the process of receiving and input and generate an output is well-structured, how efficient and precise are the responses depends of every person. Computer scientist note this and tries to simulate this behaviour trough a computer, specifically on what they call Neural Networks.

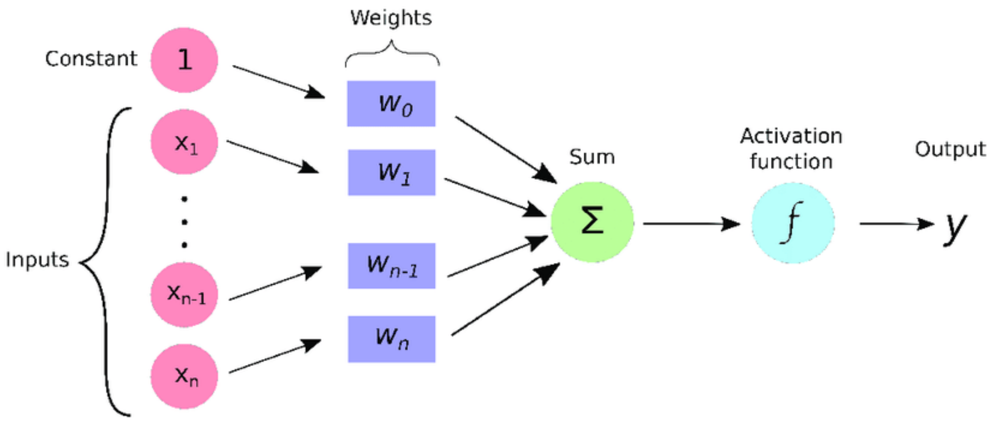


Figure 1.4: Diagram of how the perceptron works

There are large number of architecture that a developer could choose and write, but all of them have in common that they use perceptrons. Perceptrons try to copy what neurons do in our body, this means receive a series of signals, process them and generate a new signal which will be given to the following perceptrons. Mathematically, a perceptron is a linear classifier which receives an input vector $\mathbf{x}$, a weight vector $\mathbf{w}$ and bias constant u [8][9]. The weight vector represents the relative importance of every input channel, hence the vector $\mathbf{w}$ has the same dimension of the vector $\mathbf{x}$. On the other hand, the bias constant is used to adjust the output of the linear classifier. It is useful to displace the input vector so the perceptron could classifies data which is not centered at the origin of the plane.

$$z = \mathbf{w} \cdot \mathbf{x} - u \tag{1.27}$$

As we can see in Figure 1.4 after the linear combination the result of equation (1.27) pass trough an activation function f which is the final signal of the per-

Step	$f(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases}$
Sigmoid	$f(z) = \dfrac{1}{1+e^{-z}}$
Tanh	$f(z) = \dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
ReLu	$f(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$

Table 1.1: Definition of Step Function, Sigmoid, Tanh and ReLU

ceptron. If we have a single perceptron algorithm f is a step function, but if we have multiple perceptrons it is possible to use continuous functions like hyperbolic tangent, sigmoid or rectified linear unit (ReLU). All of those functions are defined in Table 1.1.

After the step of the activation function the value of $f(z)$ named as y is used to adjust the parameters of the weight vector to improve the predictions of the model. Matematically the learning process of the perceptron is ruled by the equation

$$w'_j = w_j + \alpha(y_0 - y)x_j \tag{1.28}$$

Where $w'j$ is the new of the j-value of the vector $\mathbf{w}$, w_j is the old value, y_0 is the desired output y the prediction of the perceptron and x_j the j-element of the input vector $\mathbf{x}$. The parameter α is known as the learning rate and set the speed of learning of the perceptron. This value is in the interval $0 < \alpha < 1$ where if α tends to 1 the rate of learning is fast and if it tends to 0 is slower. The performance of the network is determined by this parameter, so if we want a fast learning but no precise results we use α near to 1. On the other hand if we want to looking for patterns and an accurate prediction it is better to set a slow learning rate.

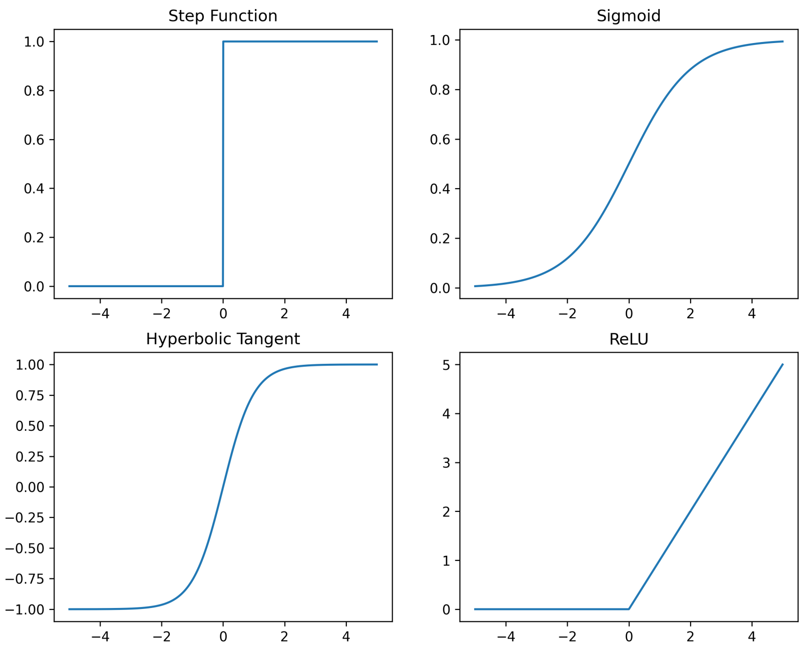


Figure 1.5: Plots of the four most used activation functions

The next step is to increase the number of neurons, so we can have a network of them. If we stack any number of nerons or cells we got a layer. Then we can put layers in front of others and connect them to have a neural network. The first layer is the input layer, here the number of neurons must be of the size of the input vector. The layers between the first and the last are called hidden layer, the reason of the name is because the user cannot see what is happening with the data in this stage. Lastly, the output layer is responsible of give to the user the prediction of the network, as the input layer, the output must be of the same size of the prediction vector.

The updating of the weight vector works similar as the case of a single perceptron but now we need to manage the performance of every layer to avoid random predictions. These is achieved by back-propagating the error of all the layers through the network. The first thing a neural network does is calculating the loss function. There are a large number of types of loss function but the most used is the mean square error.

$$L = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{1.29}$$

Then occurs the backward pass, which is the computing of the partial derivative of the loss function in the output layer respect the weight and bias of every neuron. Here is needed the chain rule to reach the input layer. For the last layer the partial derivative is

$$\frac{\partial L}{\partial W_j^l} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial W_j^l} \tag{1.30}$$

Then the weights are updated using an optimizer which is an algorithm to calculate the optimal weights in order to minimize the error in the current epoch of the network. There two most algorithm for optimizers are the Stochastic Gradient Descent (SDG) and Adaptive Moment Estimation (Adam). We are going to detail the Adam algorithm.

Firstly we need to compute the first and second moment of the network which is ruled by the following equations respectively

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \frac{\partial L}{\partial w_t} \tag{1.31}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \left(\frac{\partial L}{\partial w_t} \right)^2 \tag{1.32}$$

Where $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are parameters that represent the exponential decay rate of the moments. We use those equations to reach the global minimum the fast as possible and to avoid getting trapped into local minimums. Then we correct the first and second moment using the following equations

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{1.33}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{1.34}$$

We do this since at $t = 0$ the values of moment are biased towards zero, this correction solves the bias. Finally we update the weights

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{1.35}$$

Where ϵ is a small parameter to prevent the division by 0. In the next subsections we are going to introduce briefly some types of neural networks and how they work.

1.2.1 Fully Connected Neural Networks

As it was said, there are many types of neural networks that we can choose. The election of a certain model it will depend of the type of prediction that we want, how is structured the data or the limitation of the experiment. Fully Connected Neural Networks (FCNN) [8] are the most basic type of network. As the name suggest the main feature of this model is that all the neurons of the layer l are connected with the the entire neurons of the layer $l+1$. This characteristic converts the layer into a function of $\mathcal{R}^m \to \mathcal{R}^n$. Then the process of prediction is straightforward. Let denote a_j as the output of the neuron j, and $w_{i,j}$ the weight vector attached to link from neuron i to neuron j

$$a_j = f\left(\sum_i (w_{i,j} a_i)\right) \tag{1.36}$$

Where f is a nonlinear activation function (see Table 1.1). If we want to see the network as a whole, equation (1.36) change into

$$h_{\mathbf{w}} = \mathbf{f}^{(n)} \circ \mathbf{W}^{(n)} \mathbf{f}^{(n-1)} \circ ... \circ \mathbf{f}^{(2)} \circ \mathbf{W}^{(2)} \mathbf{f}^{(1)} (\mathbf{W}^{(1)} \mathbf{x}) \tag{1.37}$$

Here we used $\mathbf{W}$ as the weight matrix, $\mathbf{W}^{(j)}$ denotes the weight of all the neurons of the j layer and $h_{\mathbf{w}}$ is the prediction of the neural network. Then the back-

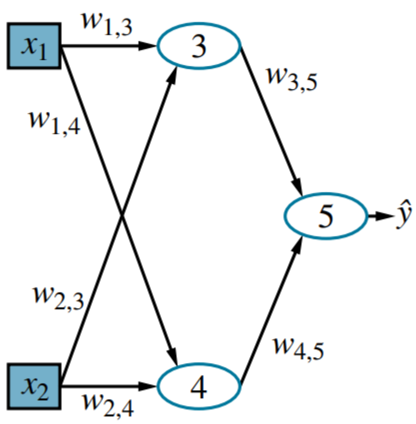


Figure 1.6: Diagram of a simple FCNN of one hidden layer of two neurons

propagation is the same that was describe in the previous section.

We can use the FCNNs in many problems of classification of images and text, prediction of time series such as traffic flow or demand of goods and voice recognition. The main reason of this is that FCNN are excellent in pattern detection so if data set follows a certain pattern FCNN could be used.

1.2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [8] are another kind of neural network. What makes them different from FCNN is how information flows though the net. In FCNNs the data is only allowed to move in forward direction. On the other hand, in RNNs there are cycles in their layers. This causes that Recurrent Networks to have an internal state or memory, so the previous inputs of the layer affects the layer response to the new ones.

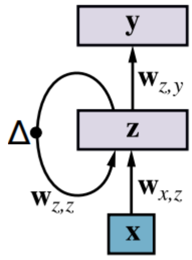


Figure 1.7: Schematic of a dummy RNN, layer $\mathbf{z}$ has recurrent connections

To treat the inputs vectors as events that happened in a certain time RNNs makes and an assumption: a hidden state $\mathbf{z}_t$ suffices to capture the information from all previous inputs [8]. Moreover, the process for updating the hidden state is ruled by the equation

$$\mathbf{z}_t = f_{\mathbf{w}}(\mathbf{z}_{t-1}, \mathbf{x}_t) \tag{1.38}$$

Where $f_{\mathbf{w}}$ is a parameterized function, typically the activation function.

In the training process we suppose a training data set $\vec{\mathbf{x}}, \mathbf{y}$. Thus, the equations that describe this stage are

$$\mathbf{z}_t = g_{\mathbf{z}}(\mathbf{W}_{z,z}\mathbf{z}_{t-1} + \mathbf{W}_{x,z}\mathbf{x}_t) \tag{1.39}$$

$$\vec{\hat{y}}_t = g_{\mathbf{y}}(\mathbf{W}_{z,y}\mathbf{z}_t) \tag{1.40}$$

Here, we have denoted $g_{\mathbf{z}}$ and $g_{\mathbf{y}}$ as the activation functions for the hidden and output layers. Then, the network computes and propagate the error. The main difference in the calculation of the error is that the algorithm needs to consider the recurrent cycles of the layer, this is called back-propagation trough time. and is usually managed automatically by deep learning software systems. Because of that we need to modify the equations of error and the backward pass.

$$L(\hat{y}, y) = \sum_{t=1}^{T} L(\hat{y}_t, y_t) \tag{1.41}$$

$$\frac{\partial L^T}{\partial W} = \sum_{t=1}^{T} \frac{\partial L}{\partial W}\bigg|_t \tag{1.42}$$

Here we have considered the error through every cycle of the neuron. Using this the network is able tho back-propagate the error though the time

In conclusion, this type of network is a great tool to analyze sequential data. For example, RNN could be used in models of Natural Language Processing because the language has an structure and how a sentence is organized its meaning might change. Also, RNN are used in meteorologic predictions since the past events affects the behaviour of the weather.

The goal of this work is to predict the spectrum of reflectance and transmittance of a dielectric mirror of SiO_2 and ZnO using a neural network. The results of this work are going to be used in a future project aimed at inverse design of dielectric mirrors using neural networks. The main reason we firstly solve the direct problem is because we want to ensure that a neural network is able to predict those parameters. After that, we are going to use those predictions to use them as input of other network whose aim is to retrieve the original geometry of the mirror. The aim of inverse design is to reduce the cost of production since in the past scientific built mirrors based on trial and error because in many situations is not obvious the best geometry that solves the problem that we are dealing.

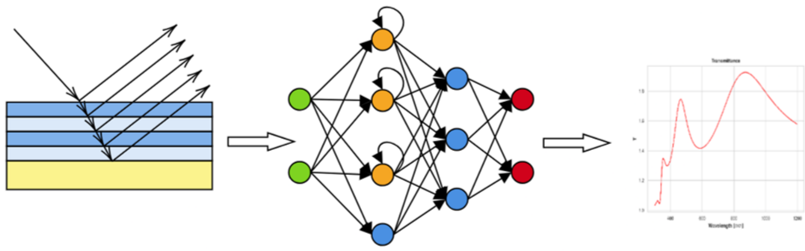


Figure 1.8: Diagram of the project

Chapter 2

Methods

2.1 Simulation and Data Set

The data set for training and testing were simulating from a code written in C++ that use the differential method [10]. The input vector were composed by the refractive indices of Silicon Dioxide SiO_2 and Zinc Sulfide ZnS in a wavelength range from $300[nm]$ to $1200[nm]$. These data were extracted from the database of RefractiveIndex Info. To increase the density between the points given by database a linear interpolation was done according to the user manual. Then, this information was used by the code to simulate the response of three different dielectric mirrors composed by 4, 8 and 12 layers respectively. All mirror layers have a width of 30 nanometers.

After the simulation the training data set was composed by an input vector

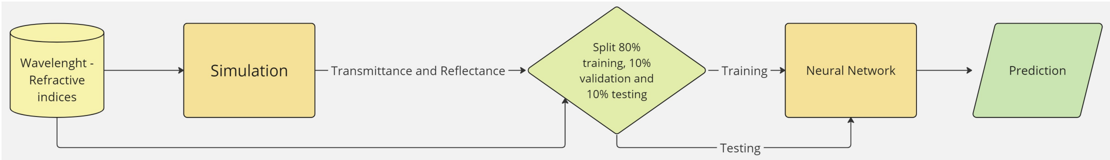


Figure 2.1: Flow diagram of the simulation process, training and testing of the Neural Network

of dimension 1200×5 where in the first column was the wavelength, in the second and thrid column were the real and imaginary refractive index of the SiO_2 and in the fourth and fifth column were the refractive index of the ZnS. On the other hand, the output vector have dimension 1200×4 where the first and the second column have the reflectance and transmittance corresponding to S-polarization and the third and fourth column have the same data of the P-polarization.

The dataset was split randomly into 2 subsets using a the random seed 42 of Python. The first subset were the 80% of the data and was used for training the neural network. The other 20% was split in two parts: 10% for testing and 10% for evaluation.

2.2 Neural Network

The architecture of the neural network was the following. The first layer consist of two neurons that received the data of the wavelength. Parallel, there was a recurrent layer of 66 neurons that received and processed both refractive indices as a matrix 2×2. Then the 2 output vectors were concatenated into one and then they pass through a FCNN layer of 266 neurons. Finally, the output layer

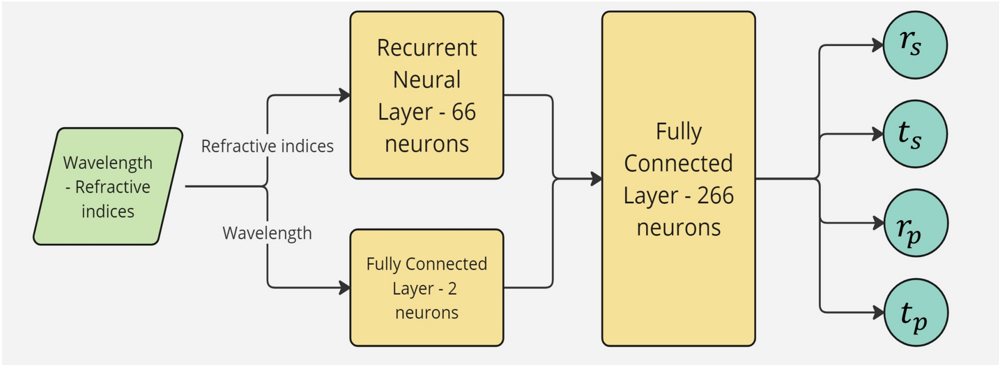


Figure 2.2: Diagram representation of the neural network used in the project, the last layer represent 4 dimensional prediction vector which consist of S and P transmittance and reflectance

consist of 4 neurons corresponding to the transmittance and reflectance of both polarization's. We used this model based on a grid search where we the parameters were the number of neurons. Our score was the R^2 of the network.

We followed the protocol of a previous work which used recurrent neural networks [11]. The activation for the neural network was the ReLU function. The optimazer was Adam and for loss function we used the mean square error (MSE). For the three simulations we train the neural network in 400, 800 and 1200 epochs respectively. Finally, the neural network was written in Python 3.10.4 using TensorFlow and Keras. The workstation use Windows 11 Pro and CPU Intel Core i7 2.80 GHz.

Chapter 3

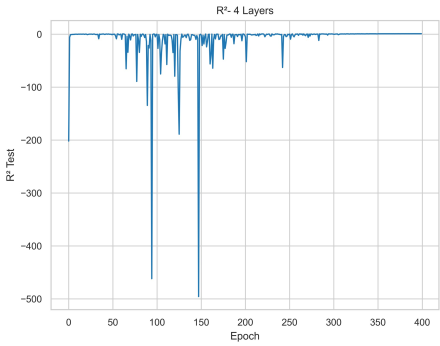
Results

Firstly, the model evaluation metrics are summarized in the Table 3.1. All the values correspond to the last epoch of training of each network indicated in section 2.2. It is important to note that the R^2, the loss and the validation loss converge into the results presented and did not show any random behaviour except in the first epochs. Also, the evolution of the R^2 for the three cases are shown in Figure 3.1

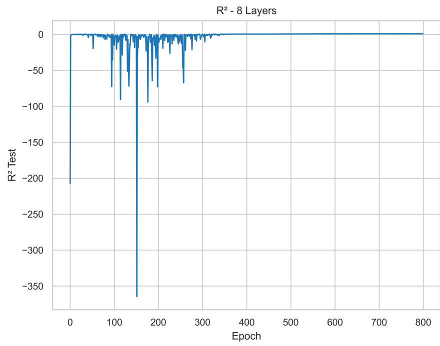
The results of the prediction are shown in the the Figure 3.2a and Figure 3.2b. The blue points represent the dispersion between the simulated values and

Table 3.1: Model evaluation metrics at the last epoch

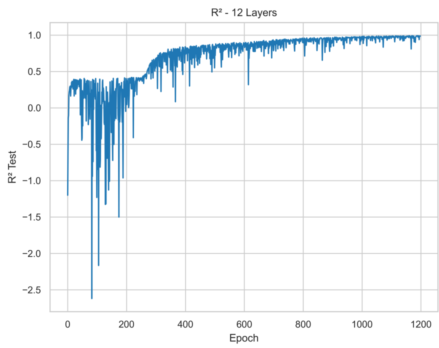
	R^2	**MSE**	**Validation MSE**
4 Layers	0.96695	5.59277×10^{-5}	4.18251×10^{-5}
8 Layers	0.98194	4.74679×10^{-4}	2.49354×10^{-4}
12 Layers	0.98655	1.58727×10^{-4}	1.38227×10^{-4}



(a)

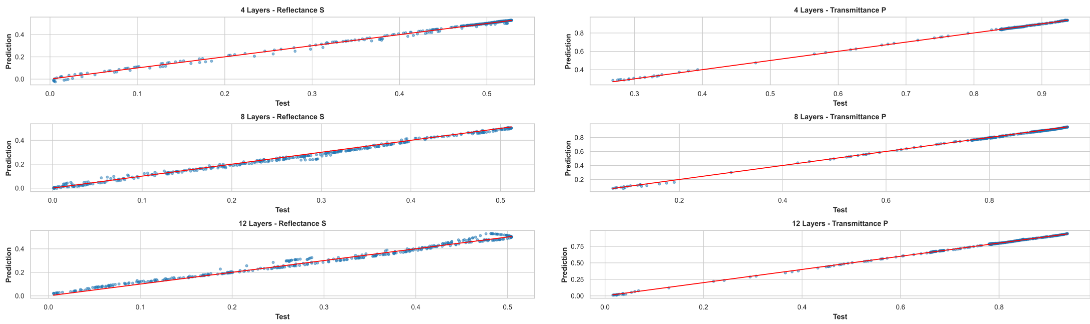


(b)



(c)

Figure 3.1: Evolution of R^2 for predictions during all epochs



(a) S reflection (b) P transmittance

Figure 3.2: Comparison of the predicted and the simulated values

the prediction of the neural network. Ideally, all points should be above the identity red line. However, in both graphs the majority of them are near the line. Despite of this good behaviour, there are some points that are considerably far away from the center. This is specially visible for the S reflection where there are two discordant points. The first is at the center of the interval for the 8 and 12 layer case. The second is at the end of the graph of the 12 layers S reflection.

We can also visualize the results by their response for each wavelength. In Figure 3.3 is plotted the simulated and predicted answer of the mirrors for the P reflected light. There was set an offset of 0.1 between both values of the corresponding layer and an offset of 0.2 among the layers. In the three mirrors there is a peak of reflectance around $400[nm]$, which is the ultraviolet range. After that region there is a second peak around the $500[nm]$. However, later the $600[nm]$ the responses of the three mirrors are not the same, since they are plotted as straight lines instead of a smooth curve.

Finally we present the evolution of mean square error (MSE) and it's validation for the three experiment. We only show the last epochs since we need to verify

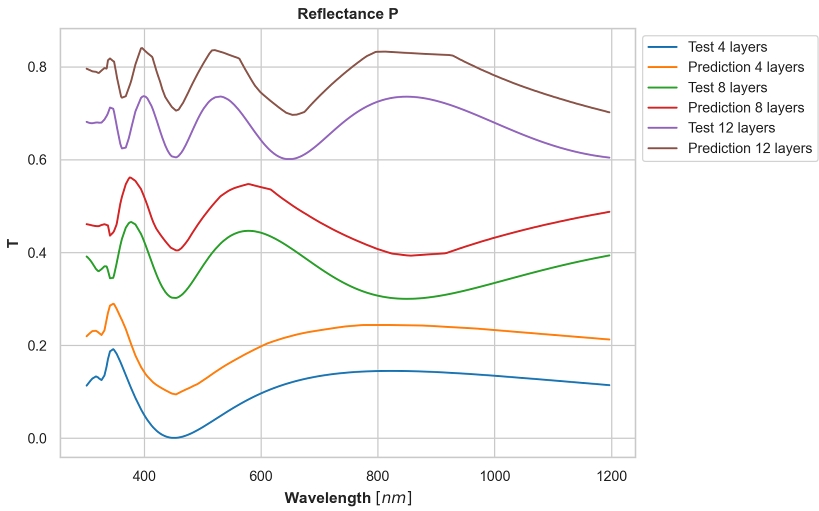
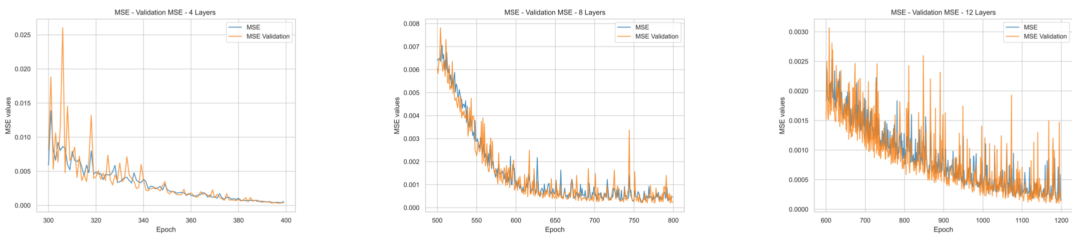


Figure 3.3: Predicted and simulated P reflection for all three geometries



(a) (b) (c)

Figure 3.4: Evolution of the mean square error (MSE) and the validation for all the three experiments

if the network is not memorizing the predictions instead of solving the Maxwell equations.

Chapter 4

Conclusion

On this work we build up a neural network to predict the reflectance and transmittance of a dielectric mirror composed by layers of Silicon dioxide and Zinc sulfide. Since reflection and transmittance are sequential process, that means that the answer of every layer is influenced by the response of the previous element, we had to use a recurrent layer which captures this feature.

In Figure 3.3 we can notice that the curves of the predicted values are not smooth as the simulated, specially for the 12 layers case. The reason is that neural networks are a good tool for interpolation [12], so if there is a low density of data in those points the network would try to approximate this as a straight line. Continuing with this idea, the prediction is not completely precise in the ultraviolet section since the patterns of reflection have more complexity due to the rapid change of the gradient in that zone. Additionally, in Figure 3.2a there are some points which are far form the identity line. That means that the prediction is not good for that reflectance. This behaviour is more notorious for the 12 layer case where for the maximum values of reflectivity the model fails giving a good result. Despite those imperfections the model still works properly, so it could be use for predicting for any visible incident light.

Table 3.1 summarize the model evaluation metrics at the last epoch for all the three geometries. We can observe that in each case we reach a good R^2, whose represent the quality of our prediction being 1 the most precise. Also, as the loss of the three cases are in the order of 10^{-5} we can infer that the error of the neural network is less than 1%. Finally, we can conclude that our model does not overfit its weights. In Figure 3.4 the curves of MSE and Validation MSE did not diverge respect the other, this behavior is expected in a network that does not overfit.

Considering all the previous remarks we can conclude that our model is capable to make precise predictions of transmittance and reflectance for both light polarization's. This is model is part of a more complex project which his aim is to design the mirror given an optical response. That means to predict the number and the width of the layers corresponding to a desired reflectance and/or transmittance.

Chapter 5

Appendix

5.1 Code

In the following url is uploaded the code implementation of the neural network used as well as the files used in this project `https://github.com/cjcamilo/Prediction-of-transmittance-and-reflectance-of-a-dielectric-mirror-using-neural-networks` .

Bibliography

[1] Mathias Kolle, Bo Zheng, Nicholas Gibbons, Jeremy J. Baumberg, and Ullrich Steiner. Stretch-tuneable dielectric mirrors and optical microcavities. *Opt. Express*, 18(5):4356–4364, Mar 2010.

[2] Wen-Fei Zhang, Wei-Jin Kong, Mao-Jin Yun, Jun-Hai Liu, and Xin Sun. Broadband and high efficiency metal multi-layer dielectric grating based on non-quarter-wave coatings as a reflective mirror. *Chinese Physics B*, 21(9):094218, 2012.

[3] VV Elyutin, MA Butt, and SN Khonina. Cold mirror based on high-low-high refractive index dielectric materials. In *CEUR Workshop Proc*, volume 1900, pages 5–9, 2017.

[4] Yusuf Bicer, Muhammad Usman Sajid, and Mohammed Al-Breiki. Optimal spectra management for self-power producing greenhouses for hot arid climates. *Renewable and Sustainable Energy Reviews*, 159:112194, 2022.

[5] G.R. Fowles. *Introduction to Modern Optics*. Dover Books on Physics Series. Dover Publications, 1989.

[6] Justin Peatross and Michael Ware. Physics of light and optics: A free online textbook. In *Frontiers in Optics 2010/Laser Science XXVI*, page JWA64. Optica Publishing Group, 2010.

[7] E. Hecht. *Optics*. Pearson, 2012.

[8] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: a modern approach*. Pearson, 3 edition, 2009.

[9] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley Publishing Company, USA, 6th edition, 2008.

[10] Jorge Luis Campos Pazmiño. Analysis of hyperbolic metamaterials using the differential method, 2018.

[11] Tomomasa Ohkubo, Ei ichi Matsunaga, Junji Kawanaka, Takahisa Jitsuno, Shinji Motokoshi, and Kunio Yoshida. Recurrent neural network for predicting dielectric mirror reflectivity. *Journal of advanced computational intelligence and intelligent informatics*, 23(6):1012–1018, 2019.

[12] Leonard Berrada, Andrew Zisserman, and M. Pawan Kumar. Training neural networks for and by interpolation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 799–809. PMLR, 13–18 Jul 2020.