Smart inverse design of graphene-based photonic metamaterials by an adaptive artificial neural network

Yingshi Chen[1], Jinfeng Zhu[1], Yinong Xie[1], Naixing Feng[2] and Qing Huo Liu[3]

[1]Institute of Electromagnetics and Acoustics, and Department of Electronic Science, Xiamen University, Xiamen 361005, China
[2]College of Electronic Science and Technology, Shenzhen University, Shenzhen 518060, China
[3]Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA

E-mail: nanoantenna@hotmail.com (J Zhu) and qhliu@duke.edu (Q H Liu)

The Python programming package is introduced as below:

§ 1 Introduction of the Package

MetaLab is an open-source Python programming package for the deep-learning study and inverse design of nanoscale optics based on graphene and other 2D materials. By using powerful machine learning algorithms (deep neural network, gradient boosting on the regression tree ...), MetaLab supports quick design of optical structures with high accuracy. The latest version could be accessed and downloaded from https://github.com/closest-git/MetaLab.

§ 2 Package Files

MetaLab V0.1 contains the following directory structure and files:

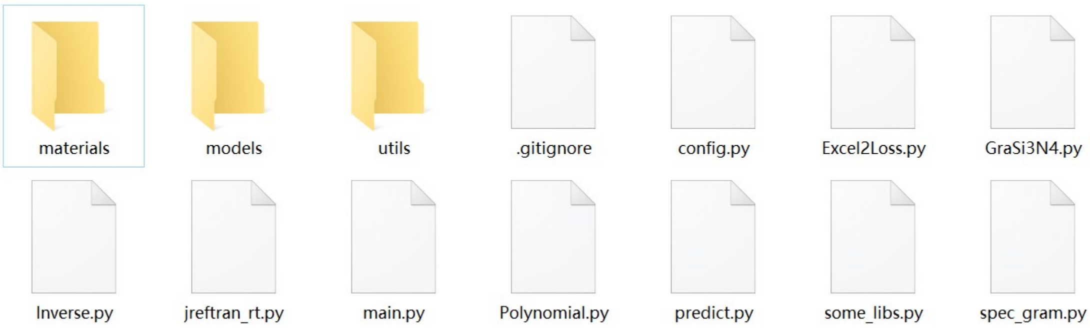

Figure 1 Directory of the files

1 materials

This directory contains files listing all material properties

For example, *Graphene index new.txt* lists the optical parameters of graphene.

Note: Users could add their own files of material optical properties to this directory.

2 models

This directory contains machine learning models. Users could add new models to this directory.

MetaLab V0.1 has two models:

1) **Embed_NN.py** A model of deep embedding neural network on the architecture of PyTorch (https://pytorch.org/).

2) **gbrt_lgb.py** A gradient boosting model based on the regression tree. On

the base of LiteMORT (https://github.com/closest-git/LiteMORT). This part is still under development.

3 utils

Some auxiliary functions.

4 config.py

All configuration parameters for the programming package. For example, the incident angle, the number of layers, the type of optical polarization, et al.

5 Excel2Loss.py

Read the prediction results (saved in Microsoft Excel files) and compute the loss function.

6 GraSi3N4.py

Python class to represent the specific multilayer nanophotonic structure, whose materials are alternating graphene and Si3N4.

7 Inverse.py

Prepare the data for the inverse design from samples.

8 jreftran_rt.py

Python implementation file - A layered thin film transmission and reflection coefficient calculator based on the method of characteristic matrices.

9 main.py

Main function. Implement generating samples, training, predicting and comparing results.

10 Polynomial.py

Generate Chebyshev points and Chebyshev polynomials.

11 predict.py

Read the prediction results by different methods. Compare the results and plot figures.

The derivation process for Equation (3) and Equation (5) is demonstrated as below, (Reference: H. A. Macleod, Thin-Film Optical Filters, CRC Press, Florida, 2010)

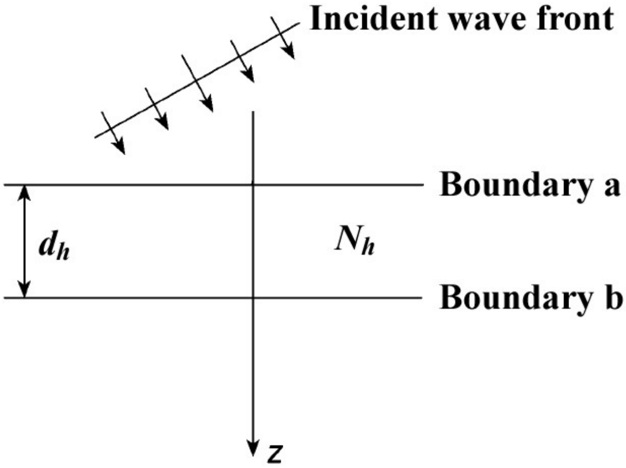

Figure 2 Plane wave incident on the multilayer structure

As shown in Figure 2, we denote waves in the direction of incidence by the symbol + (positive-going) and waves in the opposite direction by − (negative-going). At Boundary b, the tangential components of *E* and *H* for the hth layer with the complex refractive index N_h are shown as below,

$$\begin{aligned} E_b &= E_{hb}^{+} + E_{hb}^{-} \\ H_b &= \eta_h E_{hb}^{+} - \eta_h E_{hb}^{-} \end{aligned} \quad (1)$$

where η_h is the tilted admittance as below,

$$\eta_1 = \begin{cases} Y_h \cos\theta_h & (s-polarized \text{ Wave}) \\ Y_h / \cos\theta_h & (p-polarized \text{ Wave}) \end{cases} \quad (2)$$

In above equation, Y_h represents the admittance in the *h*th layer, and θ_h represents the light angle in the *h*th layer. Based on Equation (1), we have

$$\begin{aligned} E_{hb}^{+} &= \frac{1}{2}\left(H_b/\eta_h + E_b\right) \\ E_{hb}^{-} &= \frac{1}{2}\left(-H_b/\eta_h + E_b\right) \\ H_{hb}^{+} &= \eta_h E_{hb}^{+} = \frac{1}{2}\left(H_b + \eta_h E_b\right) \\ H_{hb}^{+} &= -\eta_h E_{hb}^{-} = \frac{1}{2}\left(H_b - \eta_h E_b\right) \end{aligned} \quad (3)$$

For the *h*th layer with a thickness d_h, the phase factor of the positive-going light wave is $\delta_h = 2\pi N_h d_h \cos\theta_h / \lambda$. Thus, we have the following equations for *E* and *H* on Boundary A in the *h*th layer.

$$
\begin{aligned}
E_{ha}^{+} &= \frac{1}{2}\left(H_b/\eta_h + E_b\right)e^{j\delta_h} \\
E_{hb}^{-} &= \frac{1}{2}\left(-H_b/\eta_h + E_b\right)e^{-j\delta_h} \\
H_{hb}^{+} &= \eta_h E_{hb}^{+} = \frac{1}{2}\left(H_b + \eta_h E_b\right)e^{j\delta_h} \\
H_{hb}^{+} &= -\eta_h E_{hb}^{-} = \frac{1}{2}\left(H_b - \eta_h E_b\right)e^{-j\delta_h}
\end{aligned} \tag{4}
$$

The electric and magnetic fields on Boundary A are expressed as below,

$$
\begin{aligned}
E_a &= E_{ha}^{+} + E_{ha}^{-} \\
&= E_b\left(\frac{e^{j\delta_h} + e^{-j\delta_h}}{2}\right) + H_b\left(\frac{e^{j\delta_h} - e^{-j\delta_h}}{2\eta_h}\right) \\
&= E_b\cos\delta_h + H_b\frac{j\sin\delta_h}{\eta_h} \\
H_a &= H_{ha}^{+} + H_{ha}^{-} \\
&= E_b\eta_h\left(\frac{e^{j\delta_h} - e^{-j\delta_h}}{2}\right) + H_b\left(\frac{e^{j\delta_h} - e^{-j\delta_h}}{2\eta_h}\right) \\
&= E_b j\eta_h\sin\delta_h + H_b\cos\delta_h
\end{aligned} \tag{5}
$$

This can be written in matrix notation as below,

$$
\begin{bmatrix} E_a \\ H_a \end{bmatrix} = \begin{bmatrix} \cos\delta_h & (j/\eta_h)\sin\delta_h \\ j\eta_h\sin\delta_h & \cos\delta_h \end{bmatrix}\begin{bmatrix} E_b \\ H_b \end{bmatrix} \tag{6}
$$

Therefore, the characteristic matrix for the *h*th layer is:

$$
M_h = \begin{bmatrix} \cos\delta_h & (j/\eta_h)\sin\delta_h \\ j\eta_h\sin\delta_h & \cos\delta_h \end{bmatrix} \tag{7}
$$

Based on the transmission line theory, the reflectance of the interface between an incident medium (the hemispheric prism) of admittance η_0 and the metamaterial structure with an admittance Y ($Y=C/B$) can be expressed as below,

$$
\rho = \frac{1/Y - 1/\eta_0}{1/Y + 1/\eta_0} = \frac{\eta_0 - Y}{\eta_0 + Y} = \frac{\eta_0 - C/B}{\eta_0 + C/B} = \frac{\eta_0 B - C}{\eta_0 B + C} \tag{8}
$$

$$
R = \rho\rho^{*} = \left(\frac{\eta_0 B - C}{\eta_0 B + C}\right)\left(\frac{\eta_0 B - C}{\eta_0 B + C}\right)^{*} \tag{9}
$$