Machine learning approach for computing optical properties of a photonic crystal fiber

SUNNY CHUGH,[*] AAMIR GULISTAN, SOUVIK GHOSH, AND B. M. A. RAHMAN

School of Mathematics, Computer Science & Engineering, City, University of London, London, EC1V 0HB, U.K
[*]*sunny.chugh@city.ac.uk*

Abstract: Photonic crystal fibers (PCFs) are the specialized optical waveguides that led to many interesting applications ranging from nonlinear optical signal processing to high-power fiber amplifiers. In this paper, machine learning techniques are used to compute various optical properties including effective index, effective mode area, dispersion and confinement loss for a solid-core PCF. These machine learning algorithms based on artificial neural networks are able to make accurate predictions of above mentioned optical properties for usual parameter space of wavelength ranging from 0.5-1.8 μm, pitch from 0.8-2.0 μm, diameter by pitch from 0.6-0.9 and number of rings as 4 or 5 in a silica solid-core PCF. We demonstrate the use of simple and fast-training feed-forward artificial neural networks that predicts the output for unknown device parameters faster than conventional numerical simulation techniques. Computation runtimes required with neural networks (for training and testing) and Lumerical Mode Solutions are also compared.

1. Introduction

Photonic crystal fiber (PCF) was first proposed by Knight *et al.* [1] in 1996, which consists of a core with the periodic arrangement of air holes running along the length of the fiber. The core of the PCF can be solid or hollow. For a solid-core PCF, there is a positive refractive index difference between the core and equivalent index of the cladding, and light is guided using the modified total internal reflection (TIR) phenomenon. On the other hand, hollow core PCF has a negative refractive index difference between the core and cladding, and light guidance is based on photonic band gap (PBG) mechanism [2]. Such structures exhibit the novel properties of being low loss and endlessly single mode propagation. Other specific fiber properties, including effective index (n_{eff}), propagation constant, effective mode area (A_{eff}), dispersion (D), non-linearity, birefringence and confinement loss (α_c) can be easily controlled by changing the holes size, spacing between them and number of air-hole rings. The unique properties of PCF over standard optical fibers has motivated researchers to use PCF for supercontinuum generation [3, 4], Raman scattering [5, 6], fiber laser [7], optical sensors [8, 9], spectroscopy [10], among other applications.

Accurate modeling and optimization of photonic crystal structures generally relies upon the numerical methods such as finite difference method [11], finite element method (FEM) [12], block-iterative frequency-domain method [13], and plane wave expansion method [14, 15]. However, these methods require significant computer resources when dealing with complex photonic crystal structures which needs to be simulated multiple times to obtain an optimized design. Such iterative analyses also depend upon the number of input design parameters need to be optimized.

Recently, machine and deep learning have risen to the forefront in many fields such as computer vision, robotics, chatbots, natural language processing, among others. Over the past few years, researchers have also explored the application of machine learning in the field of photonics including multimode fibers [16], plasmonics [17], metamaterials [18], biosensing [19], metasurface design [20, 21], optical communications [22] and networking [23]. Kiarashinejad *et al.* [24] proposed a deep learning-based algorithm using the dimensionality reduction technique to understand the properties of electromagnetic wave-matter interaction in nanostructures. A geometric deep learning approach has also been reported to study nanophotonics structures [25]. In 2018, extreme learning machine and deep learning were used for computing dispersion relations [26] and optimization of Q-factors [27] for photonic crystals. Here, we propose to use machine learning (ML) techniques for computing various optical properties of the PCF. We combine the finite element simulations and artificial neural networks (ANN) for the quick and accurate computation. The focus of this paper is to design a simple feed forward multilayer perceptron (MLP) model which can be trained quickly to estimate the n_{eff}, A_{eff}, D, and α_c for a PCF structure.

This work is organized as follows. Section 2 describes the ANN/MLP concepts and modeling parameters. Section 3 presents the assessment of the modeled ANN on testing PCF by comparing their estimations with actual values and computing runtime, and finally the paper is concluded in Section 4.

2. PCF modeling with ANN

The ANN/MLP architecture parameters are introduced in this section along with the PCF type that is used for generating the dataset. The first step of the training procedure of an ANN model is to have a finite and appropriate labeled dataset. This initially generated dataset plays a crucial role for any ANN model. The accuracy of the model depends upon how well the dataset is aligned to the problem to be solved. Variant PCFs were simulated by changing some geometric property values and their optical properties were calculated. In our case, set of PCF geometric data including diameter of holes (d), separation between center of two adjacent holes (pitch, Λ), refractive index of core (n_c), wavelength (λ), and number of rings (N_r) of a solid-core PCF (as shown in Fig. 1) were taken as input variables of the labeled dataset.

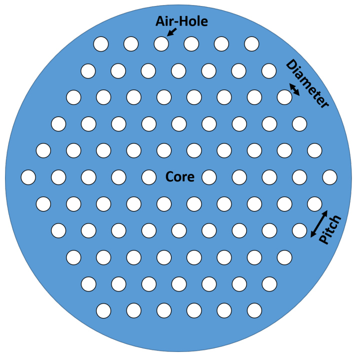


Fig. 1: Cross-section of a solid-core hexagonal PCF with five rings of air holes.

Subsequently, n_{eff}, A_{eff}, D, and α_c values were calculated using Lumerical Mode Solutions for more than 1000 samples, which were considered as the output variables of the labeled training dataset. A_{eff}, D, and α_c values can be defined by using the following equations [28]:

$$A_{\text{eff}} = \frac{\left(\iint_{\Omega} |\mathbf{H_t}|^2 \, \text{dxdy}\right)^2}{\iint_{\Omega} |\mathbf{H_t}|^4 \, \text{dxdy}} \tag{1}$$

$$D = -\frac{\lambda}{c} \frac{d^2 \, \text{Re}(n_{\text{eff}})}{d\,\lambda^2} \tag{2}$$

$$\alpha_c = 8.686 \times 10^6 \, k_0 \, \mathrm{Im}(n_{eff}) \quad \mathrm{dB/m} \tag{3}$$

where $\mathbf{H_t}$ is the transverse magnetic field vector, Ω is the area enclosed within the computational domain. Re and Im stand for the real and imaginary parts, respectively. c and k_0 are the free-space speed of light and wavenumber, respectively.

An ANN/MLP model with 3 hidden layers and 50 nodes/neurons in each layer was used throughout this paper, as shown in Fig. 2. These hidden layers were fully interconnected, which means each node/neuron of a layer is connected to each node/neuron in the following layer. 10% of data samples were randomly selected from the training dataset and allocated as the validation dataset to provide unbiased evaluation while tuning the ANN model parameters (weights and biases). Rectified linear unit (ReLU) [29] activation function and Adam [30] optimizer were used for approximating the non-linear function and to optimize the weights during the training process, respectively. ANN model predicts some outputs after each iteration/epoch. The mean squared error (MSE) value between this predicted and actual output was then calculated and back-propagation phenomenon [31] was used repeatedly to update the weights of the hidden layers for each epoch. Choosing the above-mentioned parameters are based on our prior experience of similar problems and a thorough study has been carried out in our previous work [32]. It has been observed that 3 hidden layers with 50 nodes in each layer were sufficient to quickly obtain a stable MSE for the considered PCF design. We avoided further increase in the number of layers and nodes to reduce the computational loading. Similarly, ReLU activation was chosen as it was performing better than other non-linear activation functions such as Sigmoid and Tanh (hyperbolic tangent). Adam optimizer was chosen over LBFGS and stochastic gradient descent (SGD) optimizers to optimize the weights values during the ML training process as it also performs well for a relatively large dataset. The number of epochs to be taken was decided by the user when MSE converges to an acceptable value. After optimizing the model having stable MSE value, we generate suitable outputs for the new input data which was not provided during the training process.

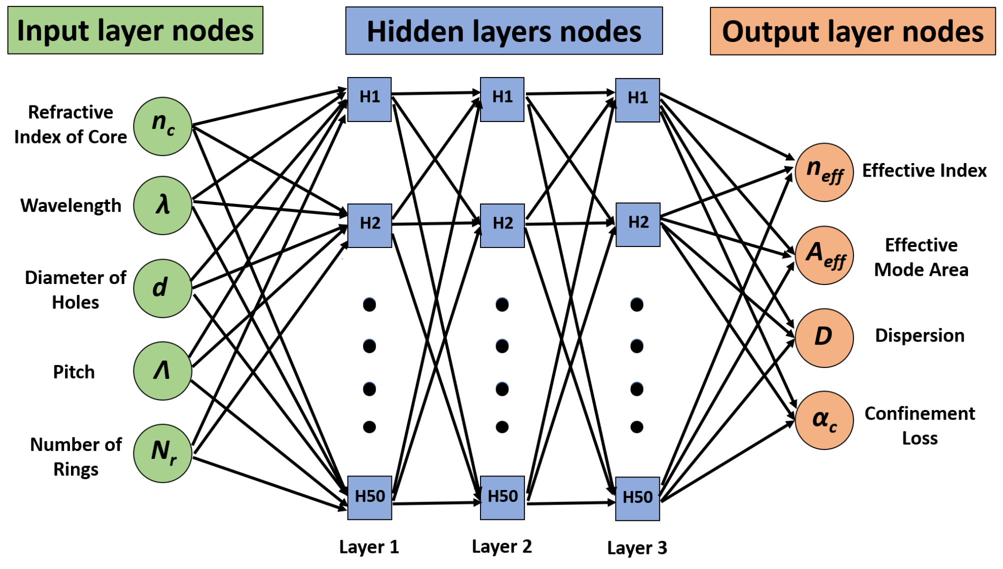


Fig. 2: Artificial neural network (ANN) representation with one input layer (5 input nodes), three hidden layers (50 nodes in each layer), and one output layer (4 output nodes).

3. Numerical Results and Computation Runtimes

In this section, we validate the trained ANN model by evaluating their outputs for a solid-core PCF at unknown design parameters, and finally the computational runtimes of ANN model are compared with the numerical simulations.

3.1. Effective Index (n_{eff})

Figure 3 shows a scatter plot of the n_{eff} values of dataset used for training of the ANN model. Predicted values of n_{eff} obtained from the ANN model are plotted against their actual values from FEM simulations. Each circle represents a single datapoint. For a well trained model, these values should be aligned closer to the y=x line (shown by a black solid line).

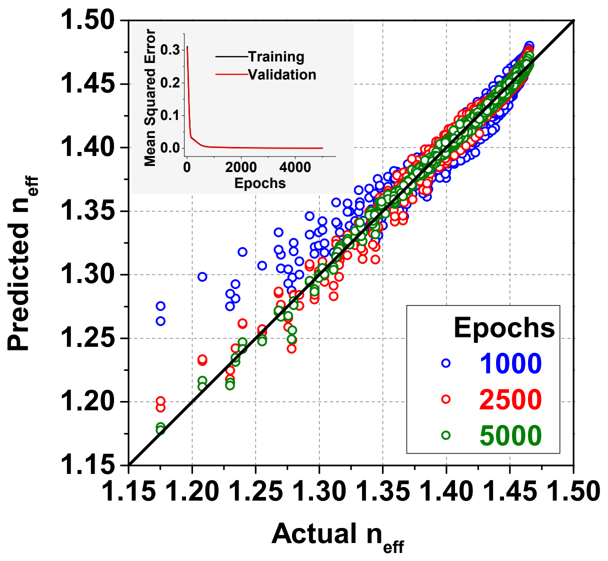


Fig. 3: The scatter plot of training dataset produced by ANN for different epochs, comparing n_{eff} values from the simulation (x-axis) and the ANN predictions (y-axis) along with the ideal linear model (y = x). Inset shows the mean squared error (MSE) obtained with epochs when training the ANN model.

It can be stated that for epochs = 1000, model is not yet well trained as the n_{eff} values (shown by blue circles) in the parameter space of 1.15-1.30 are not close to the y=x line. This can also be explained from the MSE curves for both training and validation datasets shown in the inset of Fig. 3. Here, MSE gives the average squared difference between the estimated and true values. In this case, with the particular set of optimized weights and biases of the ANN model, we noticed that there is a little difference between the MSE of training and validation datasets. Predictions are closer to the original values when MSE values are smaller. It can be observed that MSE for training dataset decreases with epochs from 0.31113 for epoch = 1 to 0.00367 for epochs = 1000. For epochs = 2500, this MSE reduces further to 0.00134. This implies that n_{eff} scatter plot should be more closer to y=x line for epochs = 2500 or larger, as can be seen by red circles. We run the ANN model till 5000 epochs at which this MSE reaches a stable value of 0.00065. For epochs = 5000, the trained ANN model for predicting n_{eff} agreed reasonably well with the actual n_{eff} values, being closest to the y=x line, as shown by green circles.

Next, in Fig. 4 we have compared actual and predicted n_{eff} for different epochs using the trained ANN model at an unknown PCF parameters, $\Lambda = 1.5$ µm, $d/\Lambda = 0.7$, and $N_r = 4$. It should

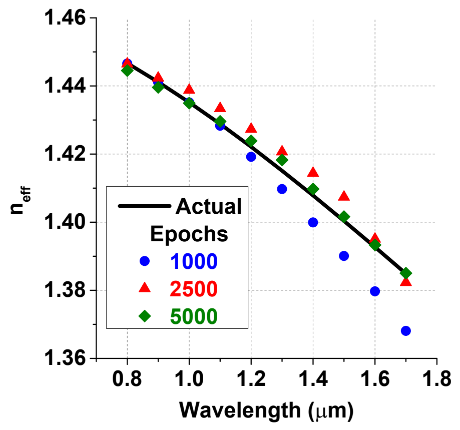
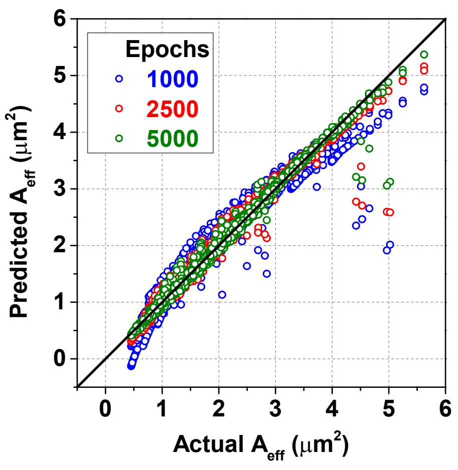


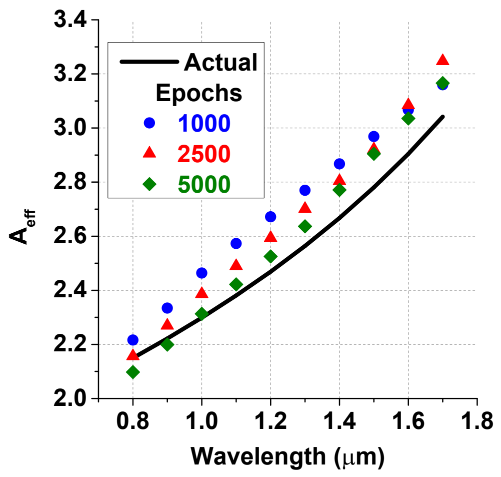
Fig. 4: Comparing actual (simulation) and predicted (ANN model) n_{eff} for different epochs at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4.

be noted that the generated input training dataset did not have any value corresponding to Λ = 1.5 μm. Generally, the n_{eff} of the fundamental mode of PCF decreases with the increase in λ. For epochs = 1000, ANN model was not able to predict the actual pattern (shown by black solid line) well, and the predicted values (shown by blue filled circles) steered away from the actual values as λ increases from 1.4 μm to 1.8 μm. When epochs were increased to 2500 or 5000, MSE for training dataset reached to a more stable value of 0.00134 and 0.00065, respectively. Hence, predicted and actual values were closer for both cases having a lower error. This also shows that 5000 epochs were sufficient for the training.

3.2. Effective Mode Area (A_{eff})



(a)



(b)

Fig. 5: (a) The scatter plot of training dataset produced by ANN for different epochs, comparing A_{eff} values from the simulation (x-axis) and the ANN predictions (y-axis) along with the ideal linear model (y = x), (b) Comparing actual (simulation) and predicted (ANN model) A_{eff} for different epochs at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4.

A_{eff} plays an important role in context of waveguiding properties in PCF, and can be calculated by using Eq. 1. A smaller A_{eff} is useful in applications with enhanced fiber nonlinearity, while a large A_{eff} can be useful in high power transmission applications.

Here, we use the previously trained ANN model to predict the A_{eff} values of the solid-core PCF. Figure 5a shows that how well the dataset was trained for A_{eff} values for epochs equal to 1000, 2500, and 5000, shown by blue, red and green circles, respectively. It can be seen that some datapoints for epochs equal to 1000 and 2500 were not well trained, especially when A_{eff} values were greater than 4 μm^2. When the epochs were increased to 5000, these datapoints became closer to y=x line. We could increase the epochs to a higher value but this would increase the simulation time, and may also lead to the overfitting problem. This trained ANN model was then used to predict the A_{eff} values at an unknown PCF parameters, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4, as shown in Fig. 5b. A_{eff} data corresponding to these parameters was never recorded or provided during the training of the model. However, our ANN model was still able to predict them. Figure 5b shows the curve of actual and predicted A_{eff} values for epochs = 1000, 2500, and 5000. When epochs were increased from 1000 to 5000, predicted A_{eff} values became closer to actual values, which is also justified by the fact that MSE decreased with increase in epochs, as shown in inset of Fig. 3. However, it was noted that even for epochs = 5000, solutions were not very accurate. Next, effect of data length or its distribution is studied.

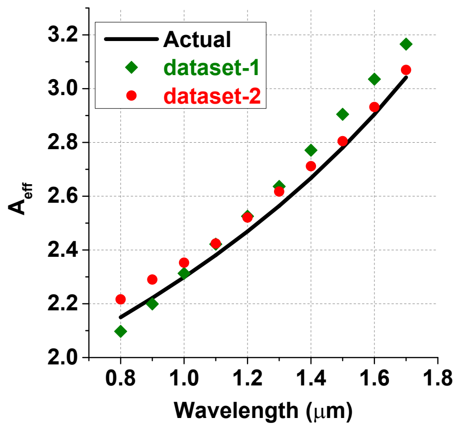


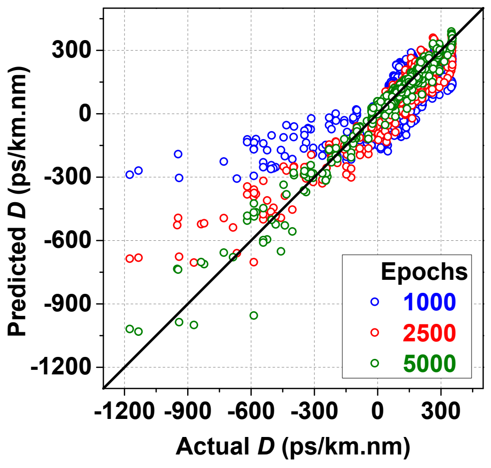
Fig. 6: Comparing actual (simulation) and predicted (ANN model) A_{eff} for different datasets at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4.

Figure 6 compared the actual and predicted A_{eff} values for two different datasets with epochs = 5000. Dataset-2 composed of all the values of dataset-1 and had some additional datapoints especially in the wavelength range from 1.3-1.7 μm, where previous predictions were poor. It can be seen that the error between the actual and predicted values was further reduced when ANN model was trained using dataset-2 (shown by red circles) in comparison to initial dataset-1 (shown by green diamonds), which clearly shows improved predictions in the higher wavelength range. Figures 5b and 6 show that both number of epochs and dataset play an important role during the training of the ANN model.

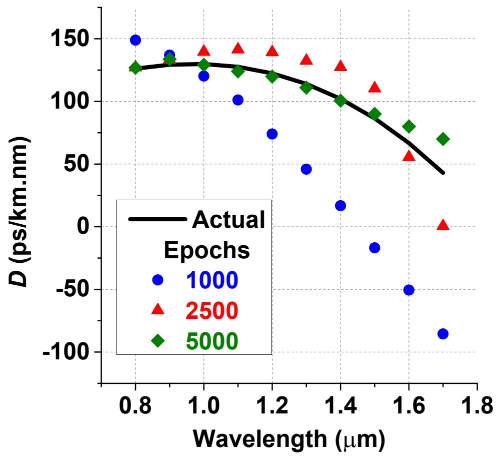
3.3. Dispersion (D)

The chromatic dispersion (D) of a PCF is an important parameter for many applications, such as supercontinuum generation, and considered here next which may be calculated using Eq. 2. The D depends on the second order derivative of n_{eff} with respect to λ.

The scatter plot showing training of the ANN model for D is shown in Fig. 7a. It can



(a)



(b)

Fig. 7: (a) The scatter plot of training dataset produced by ANN for different epochs, comparing D values from the simulation (x-axis) and the ANN predictions (y-axis) along with the ideal linear model (y = x), (b) Comparing actual (simulation) and predicted (ANN model) D for different epochs at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4.

be observed that at epochs = 1000, shown by blue circles, predicted D values significantly deviates from actual D values, especially when D is less than -400 ps/km.nm. This error was reduced if the model was trained till epochs = 2500 (red circles) or 5000 (green circles). But, at epochs = 5000, we could still observe some datapoints were not close to y=x line when D was less than -600 ps/km.nm. This error comes from the neural network modeling as there were insufficient datapoints in the parameter space of D between -1200 and -600 ps/km.nm. This leads to underfitting of the trained model for D values. Model was more biased to be trained towards D values greater than -300 ps/km.nm, as there were more datapoints. Increasing the datapoints in the lower D range further improves the training and accuracy of the ANN model for D calculations.

Actual and predicted D values at an unknown Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4 are shown in Fig. 7b. At epochs = 1000, error or gap between the simulated and ANN values was more because ANN model was not able to learn the curve shape well, and predicted it almost like a straight line. Increasing the epochs to higher values of 2500 (red triangles) or 5000 (green diamonds) reduced this error gap. It can be noted that epochs = 5000 were sufficient to predict the D pattern in this case.

3.4. Confinement Loss (α_c)

All PCF suffers from confinement loss, (α_c) which depends on the imaginary part of the complex effective index, n_{eff}, and can be computed by using Eq. 3.

Figure 8 displays the predictions of α_c of the fundamental mode with actual values when using the trained ANN model. It can be seen that for different epochs values of 1000, 2500 and 5000, the scatter plot looks significantly different. Majority of the scatter plot values lie in α_c ranging from 0-10 dB/cm for epochs equal to 1000, 2500 and 5000 as shown by blue, red and green circles, respectively. Only a few values were present for α_c greater than 20 dB/cm for different epochs mentioned, and these values were not close to y=x line which means data corresponding to these higher α_c values were not well trained. To understand the reason behind this, we need to see the general trend of α_c with λ for a particular set of PCF parameters, as shown by a blue

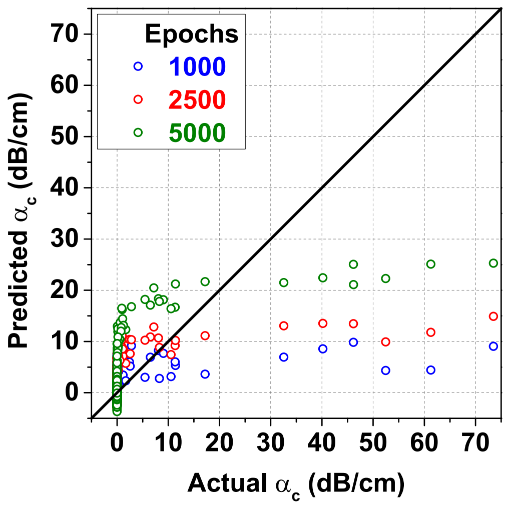


Fig. 8: The scatter plot of training dataset produced by ANN for different epochs, comparing α_c values from the simulation (x-axis) and the ANN predictions (y-axis) along with the ideal linear model (y = x).

line in Fig. 9. We have shown corresponding α_c values at a regular interval of 0.1 μm. It can be clearly observed that for λ in the range 0.8-1.4 μm, α_c values were quite small and similar, while it suddenly shoots up when λ was greater than 1.5 μm. This implies that when dataset was recorded for different Λ, d/Λ, N_r, and λ, majority of the output α_c values lied in one zone, with only a small number of values in the another zone. When majority of the values lie in one zone, the ANN model become biased towards these values. Similar thing happened in our case when training the ANN model for α_c values of the PCF.

The solution proposed to avoid this problem in our case was to take the logarithm of the initially collected α_c values when dataset was collected for different variations in Λ, d/Λ, N_r, and λ. Taking the logarithm converts the values to new set with more regular intervals, which can be trained efficiently by the ANN model. This can be observed by a red line in Fig. 9, which is the logarithm of the values of the blue line. A red line represents the logarithm of the general trend of α_c with λ for the particular set of PCF parameters. Now, this logarithmic values of α_c was

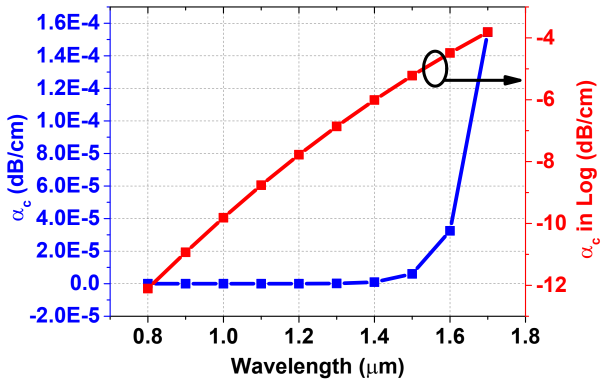
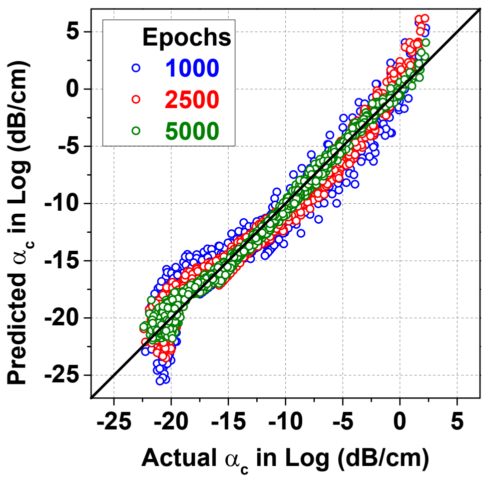
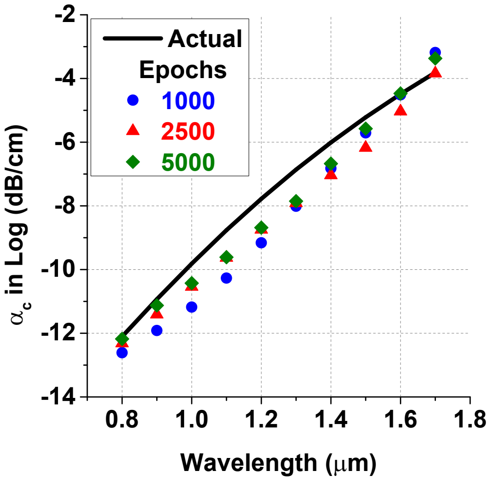


Fig. 9: Actual values of α_c from the simulation and in logarithm with wavelength for a general case.



(a)



(b)

Fig. 10: (a) The scatter plot of training dataset produced by ANN for different epochs, comparing α_c values in logarithm from the simulation (x-axis) and the ANN predictions (y-axis) along with the ideal linear model (y = x), (b) Comparing actual (simulation) and predicted (ANN model) α_c in logarithm for different epochs at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4.

trained by the ANN model like we were training the n_{eff}, A_{eff} and D, and α_c. During testing of the PCF at unknown parameters, we first used the ANN model trained using logarithm of α_c values, and in the final step we took the anti-logarithm to obtain the predicted α_c values and compared its accuracy with the actual α_c values.

Comparison of the predicted and actual α_c values in logarithm during training of the ANN model is shown in the Fig. 10a. For different mentioned epochs, it can be observed that the scatter plot values are now closer to the y=x line, showing a well trained model. Without taking the logarithm of α_c values, the same ANN model was not trained well as shown earlier in Fig. 8. The α_c values in logarithm from the simulation and ANN models for different epochs at an unknown Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4 are shown in Fig. 10b. As the epochs were increased from 1000 to 2500 to 5000, error decreased for α_c in logarithm values.

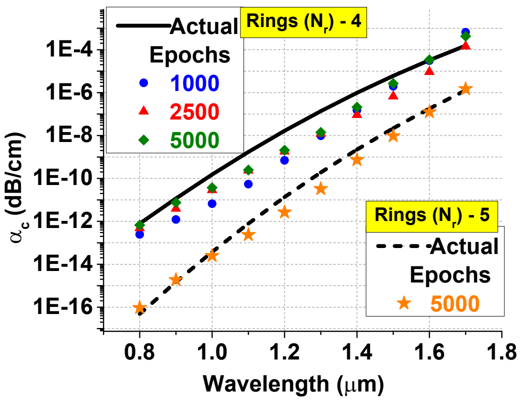


Fig. 11: Comparing actual values from the simulation and ANN model for different epochs at an unknown pitch, Λ = 1.5 μm, d/Λ = 0.7, and N_r = 4 or 5.

Next, we used the trained ANN model to predict and compare the α_c values for different epochs with the actual α_c values. The same trained model can be used to efficiently predict α_c for different N_r = 4 or 5 at an unknown Λ = 1.5 µm, d/Λ = 0.7, as shown in Fig. 11. But before comparing, we took anti-logarithm as we trained the ANN model with α_c in logarithm values. For N_r = 4 and epochs = 1000, it can be seen that predicted α_c values (shown by blue filled circles) were still far from actual α_c values (shown by solid black line). As epochs were increased to 5000 (shown by green diamond symbols), the estimations became closer to the actual α_c values. Similarly, we have shown the predicted and actual α_c values when N_r = 5 was taken for epochs = 5000. Predicted and actual values of α_c closely matched as shown by orange star symbols and dotted black line, respectively. This implies that ANN model performs better when logarithm of α_c values were taken to train the model, and 5000 epochs were sufficient to closely predict the actual values for a PCF structure.

3.5. Computing Performance

The computational platform used in this study was a laptop with Intel Core i7 CPU @ 2.80 GHz, 16 GB RAM having windows 10 operating system. The runtime to train the artificial neural network model depends on the training parameters including dataset size, number of hidden layers, number of neurons in each hidden layer and number of epochs, among others. For our particular model using 3 hidden layers with 50 nodes in each layer running for 5000 epochs, it took around 20 seconds to train the model with the generated dataset. Once the training was done, model weights and parameters were saved in the computer. Next step was to predict the output for unknown inputs, which takes only a few milliseconds to compute. This prediction was carried out using the already saved weights rather than first training the ANN again. On the other hand, the numerical computation using Lumerical Mode Solutions requires a few minutes for each point to be calculated, and can take even longer if a denser mesh is considered. The quantitative analyses comparing simulation times between Lumerical Mode Solutions for different number of mesh elements with training and testing times for the ANN model has been shown in Table 1. Multiple outputs can also be predicted simultaneously within milliseconds with the ANN model when sets of input parameters were given, while it can take even longer in numerical simulations where sweeps might be required for it.

Table 1: Simulation times with Lumerical Mode Solutions and ANN model.

	Lumerical Mode Solutions (Number of Mesh Elements)				**ANN model**	
	120 X 120	**250 X 250**	**500 X 500**	**1000 X 1000**	**Training**	**Testing**
Time taken	12 seconds	44 seconds	2 minutes 50 seconds	8 minutes 30 seconds	20 seconds	5 milliseconds

4. Conclusion

In summary, machine learning techniques have been employed to accurately predict the important properties for a silica core photonic crystal fiber design. This paper has shown that how we can predict the effective index, effective mode area, dispersion and confinement loss of a photonic crystal fiber within milliseconds, in contrast of needing few minutes with numerical simulations. Three hidden layers with 50 neurons in each layer were used throughout the code, which offer rapid convergence and sufficient accuracy in predicting outputs for unknown geometric dimensions.

The comparison between actual and predicted values in computing various optical properties were shown, and the errors between these values decreased when number of epochs were increased, as the mean squared error value reduces with number of epochs. The authors believe that these machine learning models are an efficient alternative and have potential to support computation solvers for both forward and inverse problems. In the future, the code can be easily extended to different core material as well as the hollow core photonic crystal fiber.

Code and Data Availability

We have used the open source machine learning framework - PyTorch [33] to build and test our artificial neural networks. The datasets and complete Python code used to generate the presented results during the current study are available via `https://github.com/sunnychugh/ML_PCF`. An additional dataset for testing purposes is also provided at the mentioned link.

Acknowledgments

This work was supported in part by the City, University of London, UK and in part by the Erasmus Mundus LEADERS PhD fellowship program.

Disclosures

The authors declare no conflicts of interest.

References

1. J. C. Knight, T. A. Birks, P. St. J. Russell, and D. M. Atkin, "All-silica single-mode optical fiber with photonic crystal cladding," Opt. Lett. **21**, 1547–1549 (1996).
2. P. St. J. Russell, "Photonic crystal fibers: Basics and applications," in *Optical Fiber Telecommunications VA,* (Elsevier, 2008), pp. 485–522.
3. W. J. Wadsworth, A. Ortigosa-Blanch, J. C. Knight, T. A. Birks, T.-P. M. Man, and P. St. J. Russell, "Supercontinuum generation in photonic crystal fibers and optical fiber tapers: a novel light source," JOSA B **19**, 2148–2155 (2002).
4. M. R. Karim, H. Ahmad, and B. M. A. Rahman, "All-normal dispersion chalcogenide PCF for ultraflat mid-infrared supercontinuum generation," IEEE Photonics Technol. Lett. **29**, 1792–1795 (2017).
5. F. Benabid, J. C. Knight, G. Antonopoulos, and P. St. J. Russell, "Stimulated Raman scattering in hydrogen-filled hollow-core photonic crystal fiber," Science **298**, 399–402 (2002).
6. F. Benabid, G. Bouwmans, J. C. Knight, P. St. J. Russell, and F. Couny, "Ultrahigh efficiency laser wavelength conversion in a gas-filled hollow core photonic crystal fiber by pure stimulated rotational Raman scattering in molecular hydrogen," Phys. Rev. Lett. **93**, 123903 (2004).
7. P. K. Cheo, A. Liu, and G. G. King, "A high-brightness laser beam from a phase-locked multicore Yb-doped fiber laser array," IEEE Photonics Technol. Lett. **13**, 439–441 (2001).
8. M. De, T. K. Gangopadhyay, and V. K. Singh, "Prospects of photonic crystal fiber as physical sensor: An overview," Sensors **19**, 464 (2019).
9. T. M. Monro, W. Belardi, K. Furusawa, J. C. Baggett, N. G. R. Broderick, and D. J. Richardson, "Sensing with microstructured optical fibres," Meas. Sci. Technol. **12**, 854 (2001).
10. R. Holzwarth, Th. Udem, T. W. Hänsch, J. C. Knight, W. J. Wadsworth, and P. St. J. Russell, "Optical frequency synthesizer for precision spectroscopy," Phys. Rev. Lett. **85**, 2264 (2000).
11. C.-P. Yu and H.-C. Chang, "Applications of the finite difference mode solution method to photonic crystal structures," Opt. Quantum Electron. **36**, 145–163 (2004).
12. A. Cucinotta, S. Selleri, L. Vincetti, and M. Zoboli, "Holey fiber analysis through the finite-element method," IEEE Photonics Technol. Lett. **14**, 1530–1532 (2002).
13. S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis," Opt. Express **8**, 173–190 (2001).
14. S. Shi, C. Chen, and D. W. Prather, "Plane-wave expansion method for calculating band structure of photonic crystal slabs with perfectly matched layers," JOSA A **21**, 1769–1775 (2004).
15. R. A. Norton and R. Scheichl, "Planewave expansion methods for photonic crystal fibres," Appl. Numer. Math. **63**, 88–104 (2013).
16. N. Borhani, E. Kakkava, C. Moser, and D. Psaltis, "Learning to see through multimode fibers," Optica **5**, 960–966 (2018).
17. J. Baxter, A. C. Lesina, J.-M. Guay, A. Weck, P. Berini, and L. Ramunno, "Plasmonic colours predicted by deep learning," Sci. Reports **9**, 8074 (2019).

18. W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," ACS Nano **12**, 6326–6334 (2018).
19. A. Tittl, A. John-Herpin, A. Leitis, E. R. Arvelo, and H. Altug, "Metasurface-based molecular biosensing aided by artificial intelligence," Angewandte Chemie Int. Ed. (2019).
20. C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, "Deep learning for accelerated all-dielectric metasurface design," Opt. Express **27**, 27523–27535 (2019).
21. J. Jiang, D. Sell, S. Hoyer, J. Hickey, J. Yang, and J. A. Fan, "Free-form diffractive metagrating design based on generative adversarial networks," ACS nano **13**, 8872–8878 (2019).
22. B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," J. Light. Technol. **36**, 4843–4855 (2018).
23. F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," IEEE Commun. Surv. Tutorials **21**, 1383–1408 (2018).
24. Y. Kiarashinejad, S. Abdollahramezani, M. Zandehshahvar, O. Hemmatyar, and A. Adibi, "Deep learning reveals underlying physics of light-matter interactions in nanophotonic devices," arXiv preprint arXiv:1905.06889 (2019).
25. Y. Kiarashinejad, M. Zandehshahvar, S. Abdollahramezani, O. Hemmatyar, R. Pourabolghasem, and A. Adibi, "Knowledge discovery in nanophotonics using geometric deep learning," arXiv preprint arXiv:1909.07330 (2019).
26. A. da Silva Ferreira, G. N. Malheiros-Silveira, and H. E. Hernández-Figueroa, "Computing optical properties of photonic crystals by using multilayer perceptron and extreme learning machine," J. Light. Technol. **36**, 4066–4073 (2018).
27. T. Asano and S. Noda, "Optimization of photonic crystal nanocavities based on deep learning," Opt. Express **26**, 32704–32717 (2018).
28. S. S. A. Obayya, B. M. A. Rahman, and K. T. V. Grattan, "Accurate finite element modal solution of photonic crystal fibres," IEE Proceedings-Optoelectronics **152**, 241–246 (2005).
29. F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications," J. Light. Technol. **37**, 493–516 (2019).
30. D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).
31. Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 Connectionist Models Summer School,* vol. 1 (CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988), pp. 21–28.
32. S. Chugh, S. Ghosh, A. Gulistan, and B. M. A. Rahman, "Machine learning regression approach to the nanophotonic waveguide analyses," http://dx.doi.org/10.1109/JLT.2019.2946572 (2019).
33. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop,* (2017).