PAPER

Deep learning models for effective refractive indices in silicon nitride waveguides

IOP Publishing Journal of Optics
J. Opt. 21 (2019) 035801 (8pp) https://doi.org/10.1088/2040-8986/ab00d5

Deep learning models for effective refractive indices in silicon nitride waveguides

Gandhi Alagappan[1] and **Ching Eng Png**

Institute of High Performance Computing, Agency for Science, Technology, and Research (A-STAR), Fusionopolis, 1 Fusionopolis Way, #16-16 Connexis,138632, Singapore

E-mail: gandhi@ihpc.a-star.edu.sg and pngce@ihpc.a-star.edu.sg

Received 23 August 2018, revised 16 December 2018
Accepted for publication 22 January 2019
Published 18 February 2019

Abstract

This article displays the method of constructing deep learning models for optical mode solving, with a minimal number of exact numerical solutions to Maxwell's equations. We select a silicon nitride channel waveguide and show how the patterns in the effective refractive indices of the fundamental waveguide modes for both polarizations of light, can be uncovered with only 4–16 learning points for the entire parameter space that can be conveniently accessed using existing photo-lithographical and CMOS fabrication techniques. We also illustrate the effect of various transfer functions and neural network layouts to the overall performance of the deep learning model.

Keywords: integrated optics, deep learning, waveguide

(Some figures may appear in colour only in the online journal)

1. Introduction

Deep learning (DL) [1, 2] is a powerful methodology in pattern recognition [3, 4], classification [5], and decision-making problems [6]. Recently, its usage has been proven to be invaluable in many research fields such as material science [7], chemistry [8], particle physics [9], microscopy [10] and photonics [11]. In photonics, DL has been used to design multilayer films with targeted transmission spectra [12], and on-demand designs of metamaterials [13, 14]. DL also has been shown to accelerate photonic designs by completing the partial executions of finite difference time domain simulations with data driven predictions [15]. In this article, we illustrate how DL techniques can be applied to mode solving in nanophotonic waveguides with minimal number of numerical simulations.

Modes of an optical waveguide can be solved by seeking solutions to time-independent Maxwell's equations. Analytical solutions exist for simple one-dimensional slab waveguides [16]. For two-dimensional geometries such as channel, and strip waveguides, numerical simulations must be sought for accurate solutions [16, 17]. Usually, to tackle the numerical problem, one applies matrix-diagonalization based methods such as finite difference [18], and finite elements [19, 20]. Although, these methods are well established, they consume certain amount of computational resources, and such resource consumption especially matters when performing large number of geometrical sweeps, optimizations and group index calculations. The parameter space for the waveguide geometry in the photonic problems is usually well-defined. They are limited by the choice of available materials, and the dimensions that can be fabricated using existing fabrication capabilities. Researchers often explore the well-defined parameter space repeatedly with brute force numerical methods. Such repeated exploration is thus, an inefficient use of computational resources. In this article, we show how simple DL models can be applied to produce effective refractive index of optical waveguides almost quasi-instantaneously in the inference phase, with very minimal number of numerical simulations in the one-time, non-recurrent, training phase.

2. Models and results

For purpose of an illustration let us consider a silicon nitride (SiN) channel waveguide of width w, and height h embedded in silica host medium (figure 1(a)). Optical transparency of

[1] Author to whom any correspondence should be addressed.

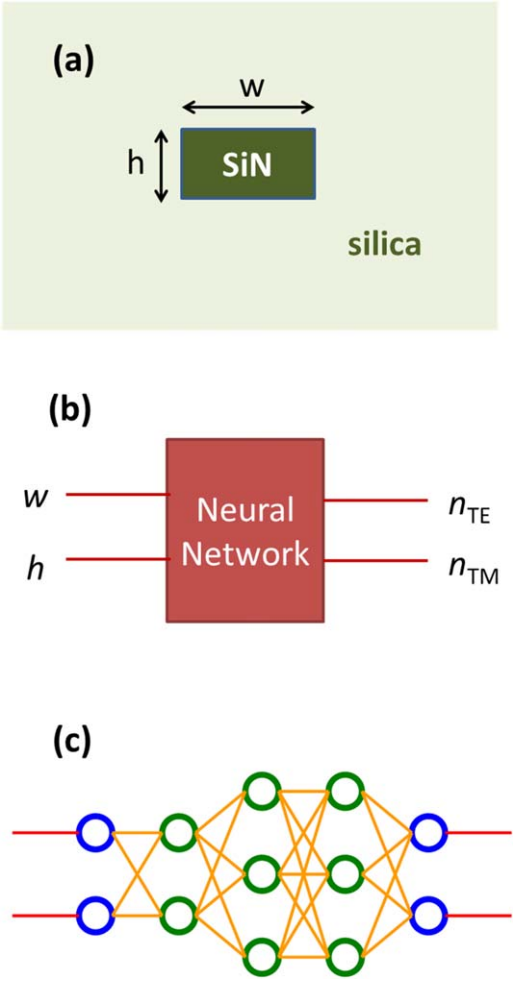

Figure 1. (a) The schematic of the waveguide cross section. (b) Deep learning model for solving effective refractive indices of the fundamental waveguide mode for both polarizations of light. (c) The specific NN layout in the closed red box of (b). The depicted layout has $N = 3$, and neurons distributions of $2 \times 3 \times 3$. The green circles represent neurons, and the blue circles represent input and output nodes.

SiN ranges from visible to infrared portion of the electromagnetic spectrum. This geometry as in figure 1(a) has been widely used many photonic devices covering the wider above-mentioned electromagnetic spectrum [21–26]. Some examples of photonic devices include directional couplers, beam splitters, ring resonators and arrayed waveguide gratings. One of the crucial information to design this list of devices is the effective refractive index of the optical waveguides. Many of the related optical quantities such as free spectral range, dispersion, and coupling length are all can be derived using the effective refractive indices of the constituent optical waveguide.

Figure 1(b) show the schematics the DL model for solving the effective refractive index of the fundamental waveguide mode. The DL model is an artificial neural network (NN) which takes w and h as input parameters. After an adequate training, it outputs the effective refractive indices of the fundamental waveguide modes for both polarizations of light, n_{TE} and n_{TM}. Here n_{TE} and n_{TM} are respectively stand for effective refractive indices of transverse electric (TE) and transverse magnetic ™ polarizations of the waveguide mode. The NN are trained using a set of N_L number of learning data points. A learning data point comprises an input (w, h) and output (n_{TE}, n_{TM}) pair. The output parameters n_{TE} and n_{TM} in the learning data points are exact solutions to Maxwell's equations, and they are numerically obtained using the method of finite difference [18]. In numerical simulations we used refractive indices of 2.0 and 1.45 for silicon nitride and silica, respectively. For the benefit of both visible and telecommunication wavelength photonics, we used a normalized unit of a. For visible light photonics at $\lambda = 0.738\ \mu$m, $a = 1\ \mu$m, and telecommunication wavelength photonics at $\lambda = 1.5\ \mu$m, $a = 2\ \mu$m. We considered an input parameter space that can be conveniently achieved using conventional photolithographic and CMOS fabrication techniques.

NN is basically a network of layers of neurons. Let's denote the number of layers as N, and the number of neurons in the ith layer, as n_i. Figure 1(c) shows an example of NN with $N = 3$ and n_i being 2, 3 and 3 for $i = 1$, 2 and 3, respectively. We say the corresponding NN has a neurons layout of $2 \times 3 \times 3$. In figure 1(c), the blue circles represent input and output nodes, and the green circles represent neurons in the internal (hidden) layer. The network of these green circles (neurons) is what is hidden in the closed red box of figure 1(b). The input for each neuron in the ith layer is the weighted output of all neurons in the $(i-1)$th layer with an added bias (i.e. a constant term). The input and output relationship of a neuron is called as the transfer function or activation function of the neuron. The unknown weights and the biases in the NN are found by training the NN using the learning data points. The training is performed by best matching the input–output pair from the learning data point using an error backpropagation algorithm.

As an illustration of training, and the use of trained NN as a meta-model for new sets of (w, h) let us revert to the NN in figure 1(c). We trained the NN in figure 1(c) using sets of uniformly spaced data points, by employing the Levenberg–Marquardt backpropagation algorithm [27–29]. This algorithm is a fast and a common choice of training algorithm for fully connected feedforward NNs of small sizes. For transfer functions, we used hyperbolic tangent sigmoid for all neurons in internal layer, and linear transfer function for neurons in output layer. The details of the transfer functions, and the effect of various transfer functions on NN training will be addressed later in the article.

One of the common design steps in integrated optics is to calculate the effective refractive indices of the waveguide mode as a function of waveguide width, or waveguide height. This is accomplished by performing numerical simulations for a series of waveguide widths or heights. For an example, to calculate the effective index curve of a waveguide when w is varied from 160 to 1000 nms (fixed h), one usually samples 10 widths between 160 and 1000 nms and perform numerical calculations. The results are then fitted to produce the required effective index curve as a function of w. If the waveguide h is changed, then it requires performing another set of 10 exact numerical simulations. Thus, if one has 6 cases of fixed h

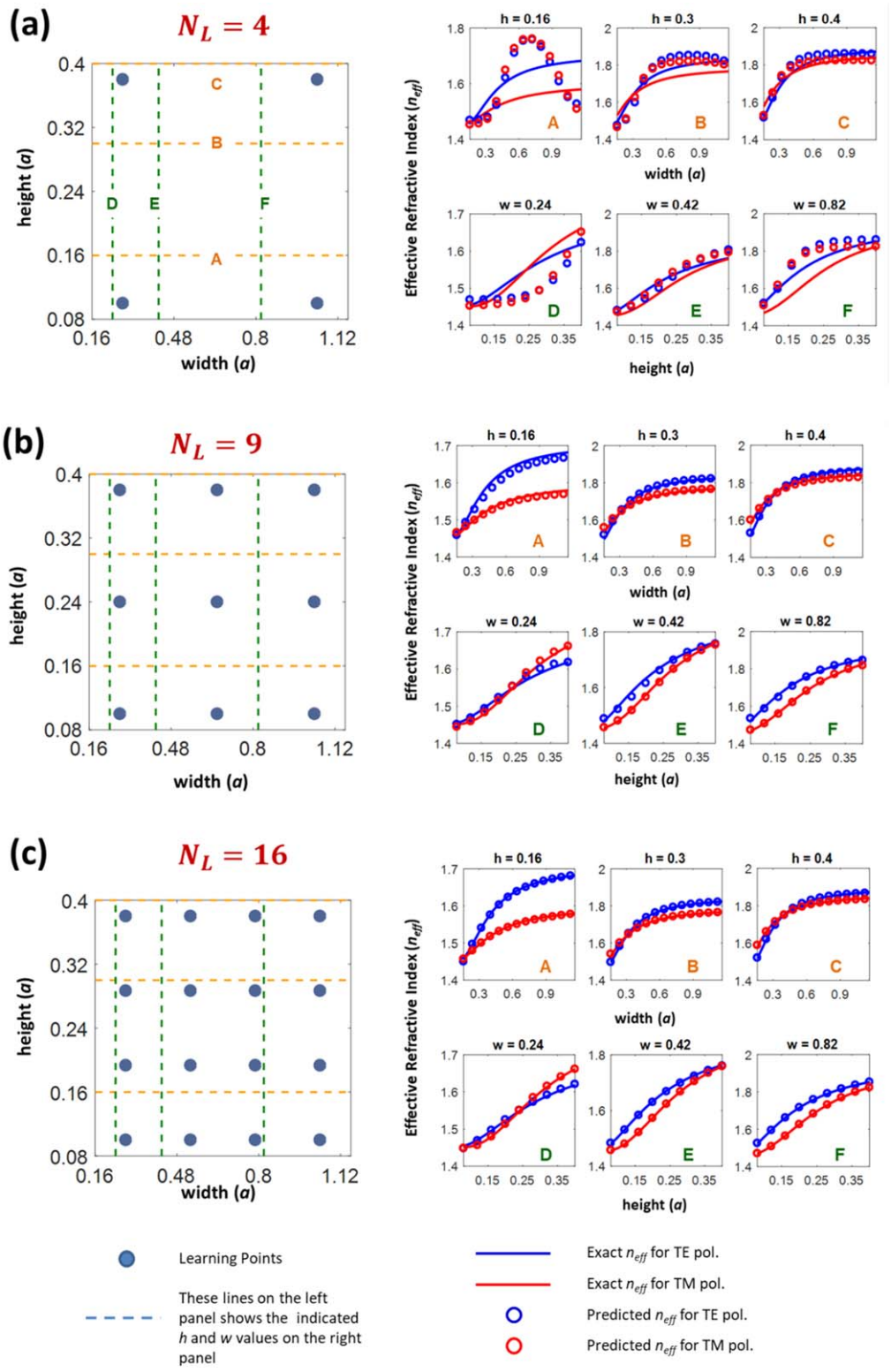

Figure 2. Performance of the NN in figure 1(c) as a function of number of learning points (N_L). (a) $N_L = 4$, (b) $N_L = 9$, and (c) $N_L = 16$. The diagram on the left in (a)–(c) represents the considered w–h parameter space with the learning points indicated in blue circles. The effective refractive indices (exact and predicted) along the dotted lines with specific h and w values (A, B, C, D, E and F—see the left panel) are shown in the right diagrams.

(or fixed w), roughly 60 simulations are needed to predict the effective index curves as a function of w (or h). In the following we shall show how one can use NN to reduce the number of exact simulations. We show only 4–16 points are required to predict the effective index curves for the entire input parameter space.

Figure 2(a) showcases the NN performance when the number of learning points is 4. The diagram on the left of

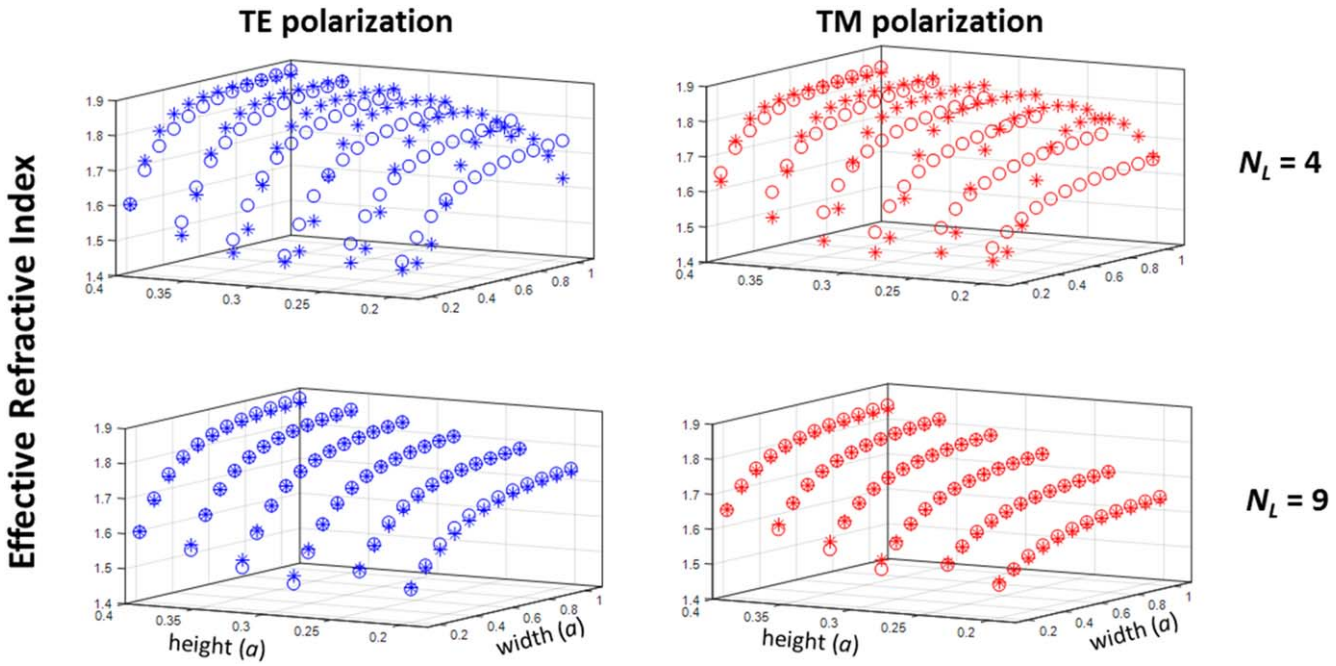

Figure 3. Three-dimensional plot showing NN predicted effective refractive indices (asterisks) and the exact values (circles) for different polarizations of light and number of learning points.

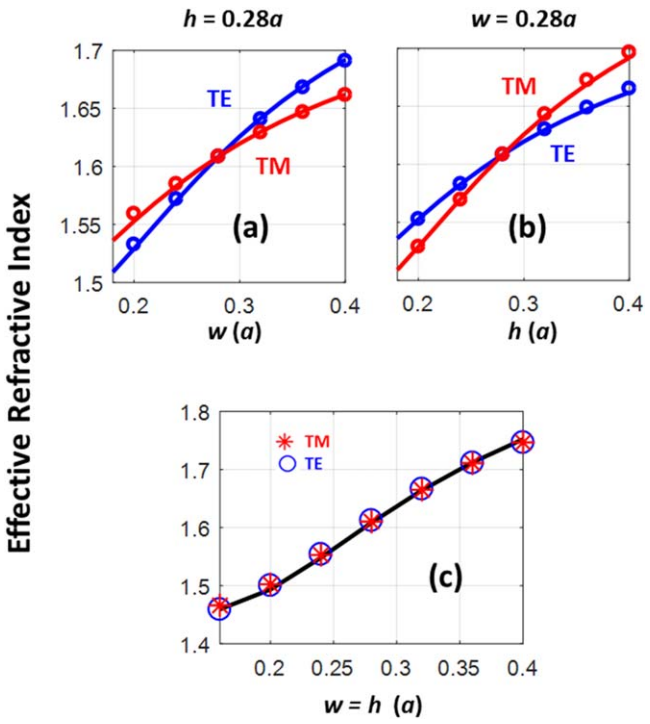

Figure 4. Ability of NN in predicting the symmetries between the effective refractive indices of different polarizations. (a) Effective refractive index as a function of waveguide width, and a fixed height of 0.28a. (b) Effective refractive index as a function of waveguide height, and a fixed width of 0.28a. (c) Effective refractive indices of square waveguides with width equals to height.

figure 2(a) explicitly shows the input parameter space and the locations of learning points (blue circles). The effective index predictions along the dashed lines (A, B, C, D, E and F) in this diagram are shown on the right diagrams. In the diagrams on the right, the blue and red solid lines represent the exact numerical solutions for n_{TE}, and n_{TM} respectively. On the other hand, the blue and red circles display the results of NN predictions. It is amazing to see with only 4 numerical simulations (learning points), the NN is able to predict the patterns in the wider parameter space. From figure 2(a), where we can see that the patterns predicted by NN has fairly a good agreement with the exact patterns (except for smaller dimensions of waveguide ($h = 0.16\ \mu$m, and $w = 0.24\ \mu$m cases)). The accuracy of NN prediction increases as a function of number of learning points. For examples, figures 2(b) and (c) reconstruct figure 2(a), but with the number of learning points increased to 9 and 16 respectively. When the number of learning points is 9 (figure 2(b)), there are very good agreements between the exact effective index curves and curves predicted by NN. When the number of learning points is increased further to 16, we can see that there is visually no discrepancy between the exact and NN calculations (figure 2(c)). It is worth noting that in all cases of $N_L = 4$, 9 and 16 the learning points are not in the paths where the tests are conducted. In figure 3, we show three-dimensional plots displaying NN predicted and exact effective indices for $N_L = 4$ and 9. Clearly the NN predictions are in good agreement with the exact values for the considered w–h parameter space (except for smaller dimensions of waveguides in the case of $N_L = 4$). The degree of agreement increases as N_L increases. In figure 4, we show the ability of NN in predicting the symmetries between the n_{TE} and n_{TM}. Figure 4(a) displays the effective indices as a function of w for $h = 0.28a$, and figure 4(b) shows the inverse scenario where h and w are interchanged. As we can see from the exact curves (solid lines) in figures 4(a) and (b), the effective index curve of TE in figure 4(a) is identical to the effective index curve of TM in figure 4(b) (similarly for TM in figure 4(a), and TE in

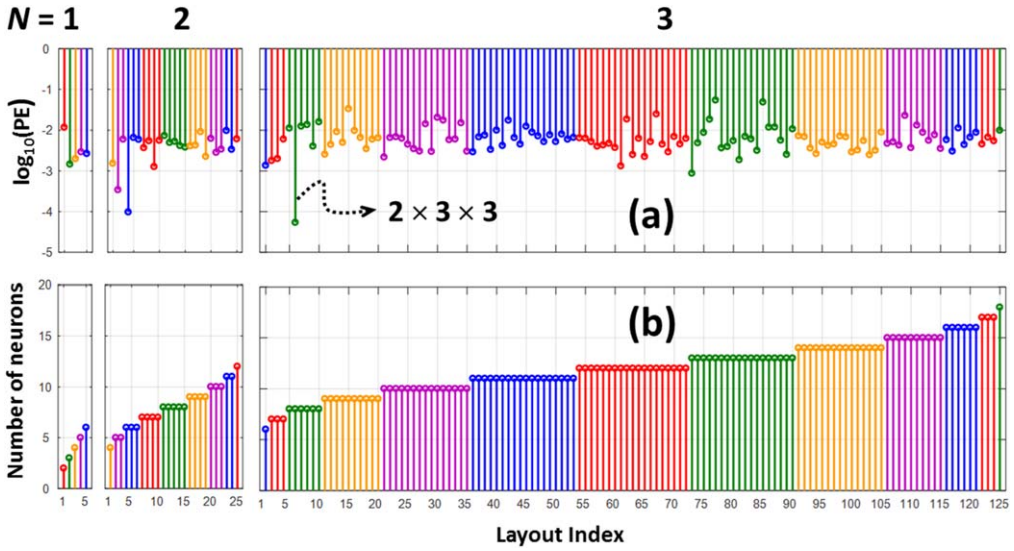

Figure 5. (a) Performance errors, and (b) number of neurons for all the layouts for $N = 1$ (5 possible layouts), 2 (25 possible layouts), and 3 (125 possible layouts).

figure 4(b)). These symmetries were also seen in the NN (trained with $N_L = 9$) predicted values (circles). Figure 4(c) shows the effective index curve for square waveguides ($w = h$) where both eigenmodes of TE and TM polarizations are degenerate with identical effective refractive indices [$n_{\mathrm{TE}} = n_{\mathrm{TM}}$]. As we can see from figure 4(c), this degeneracy is also vivid in the NN predictions.

3. Discussion

Firstly, let us justify the choice of NN layout in figure 1(c). There are no hard-written rules on finding the best NN layout. However, there are guidelines that helps the optimization. Here we follow the guidelines in [30] and take the number of neurons in any internal (hidden) layer [n_i] as a number that fall between the number of inputs and thrice the number of inputs. In our case, we have two inputs, and thus $2 \leqslant n_i \leqslant 6$ (i.e. five possible number of neurons). Therefore, the numbers of possible NN layouts are 5, 25 and 125 for single layer ($N = 1$), double layer ($N = 2$) and triple layer ($N = 3$), respectively. The corresponding performance errors, for all the layouts when trained with $N_L = 9$ are systematically shown in figure 5(a) for $N = 1$, 2 and 3. The performance error is defined as mean square error between the exact calculations and NN prediction for a dense and uniform grid in the w–h parameter space with a step size of $0.02a$. The horizontal axis in figure 5(a) represents an index that uniquely represents a specific layout. For each N, the index varies from 1 to number of possible layouts. Figure 5(b) shows the number of neurons (i.e. $\sum n_i$) for each layout in all the three cases of N. From figure 5(a), we can see that the layout $2 \times 3 \times 3$ neurons distribution yields the lowest performance error for $N_L = 9$. We extended the analysis for the cases of number of learning data points: $N_L = 25$, and number of layers: $N = 4$ and 5. For cases of $N = 4$ and 5, we have 625 and 3125 different NN layouts, and all of them were trained and analysed. The neurons distributions for the best layout as a function of N are shown in figure 6(a), and the corresponding lowest performance errors are graphed in figure 6(b).

Figure 7 investigates the effect of transfer functions on the performance of NN. We considered five different kinds of transfer functions [30, 31], namely hyperbolic tangent sigmoid (*tansig*), linear (*lin*), radial basis (*radbas*), log-sigmoid (*logsig*), and rectified linear unit (*relu*) transfer functions. The input (x), and output (y) relationship for *tansig*, *lin*, *radbas* and *logsig* transfer functions are $y = \frac{2}{1+e^{-2x}} - 1$, $y = e^{-x^2}$, $y = x$, and $y = \frac{1}{1+e^{-x}}$, respectively. For *relu* transfer function, we have $y = x$, for $x \geqslant 0$, and $y = 0$ for $x < 0$. In figure 7, we show the performance error for the trained NN (layout as in figure 1(c), and $N_L = 9$) with various neurons transfer functions (labelled in different colors). Figure 7 shows how the performance error changes when the transfer function in the internal layer changes. In all cases, the transfer functions for the neurons in the output layer is kept to be *lin*. A linear transfer function at the output layer allows a better fitting with continuous range of outputs (i.e. effective indices in this case). As is visually apparent from figure 7, the cases with transfer functions for the internal layers being either *tansig*, or *logsig* yield the lowest errors.

Figure 8 compares performances of different NN layouts, and the conventional fitting methods as a function of number of learning data points. Specifically, we presented the performances of two different NN layouts, $2 \times 3 \times 3$ and $2 \times 6 \times 3 \times 2$, linear interpolation and cubic-spline interpolation [32]. Firstly, consider the two different NN layouts. The layouts $2 \times 3 \times 3$ and $2 \times 6 \times 3 \times 2$ are the best performing layouts for the cases of $N_L = 9$ and 25, respectively (see figure 6). The best performing NN at one N_L, may not be the best for the other N_L. For example, when we have fewer data points ($N_L = 9$), $2 \times 3 \times 3$ layout outperforms $2 \times 6 \times 3 \times 2$, and we have a larger data set ($N_L = 25$),

N	Best Layout	
	$N_L = 9$	$N_L = 25$
1	3	5
2	2×4	4×2
3	$2 \times 3 \times 3$	$4 \times 5 \times 2$
4	$2 \times 3 \times 4 \times 4$	$2 \times 6 \times 3 \times 2$
5	$2 \times 3 \times 4 \times 4 \times 3$	$2 \times 4 \times 6 \times 6 \times 2$

(a)

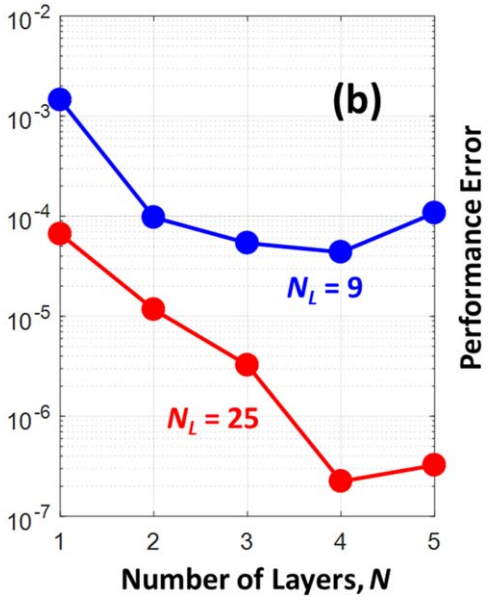

Figure 6. (a) Best NN layout (layout with lowest performance error) as a function of number of layers in the NN when trained with 9 and 25 learning points (N_L). (b) The lowest performance error corresponding to the best layout listed in (a).

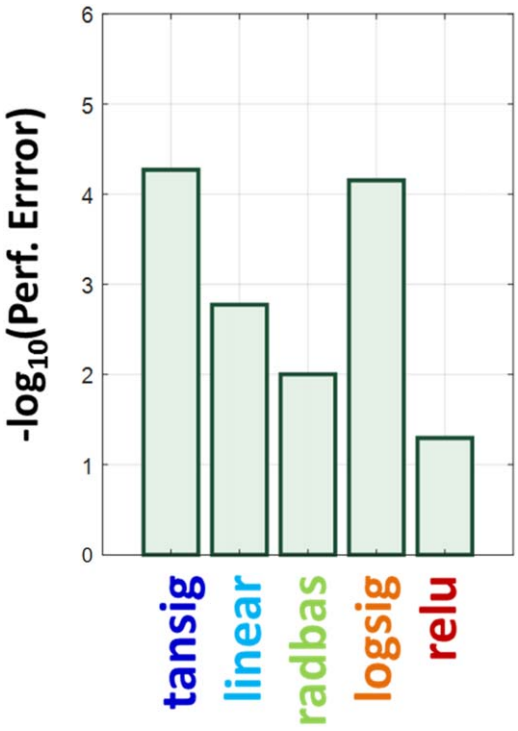

Figure 7. Log_{10} (performance error) for the layout in figure 1(c) when trained with various transfer functions.

$2 \times 6 \times 3 \times 2$ layout outperforms $2 \times 3 \times 3$ (see figure 8). When only fewer data are available for training, a small network is better than a larger network. Larger networks work better with a larger training set. Now consider, the interpolation methods. As we can see from figure 8, cubic-spline interpolation outperforms the linear interpolation. However, the performances of these conventional interpolation techniques are lower than the performances of NN. Additionally, NN techniques allow one to appropriately modify the layout with respect to number of available data points to yield the lowest performance error. As we have illustrated in figure 8, a smaller network for a smaller learning data set, and a larger network for a larger data set.

In linear interpolation method, the two adjacent data points are assumed to be connected with a straight line, and in cubic-spline interpolation method instead of a straight line, a third order polynomial is assumed to connect the two neighbouring data points. The goal in these techniques are then to find the corresponding polynomial coefficients that provide best-fits to the data points [32]. DL techniques, on the other hand, provides a general framework for regression problems. The main aims of DL are pattern recognition and generalization, rather than a curve fitting. Here, generalization refers to performance with respect to new and unseen data points (validation and test data points—these points are not used in NN training). The training in DL is stopped when the error in validation data points diverges. Furthermore, as illustrated in figure 8, DL techniques allow one to optimize a suitable network that generalizes well on unseen data points, depending on number of available learning data points.

4. Conclusion

In summary, we have demonstrated how a fully connected NN models can be employed to solve effective refractive indices of the fundamental waveguide mode for both polarizations of light, in a silicon nitride channel waveguide. With only four to sixteen data points obtained from exact numerical

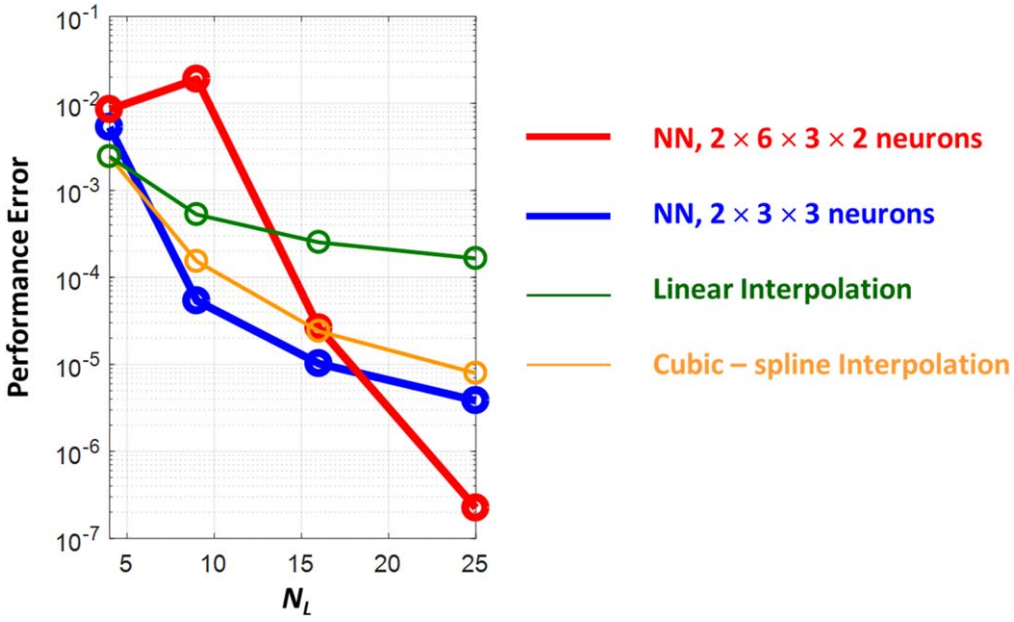

Figure 8. Performance comparison between NN predictions, and conventional interpolation techniques. N_L refers to number of learning points in the case of NN, and number of data points made available for interpolation in the case of linear and cubic-spline interpolations.

simulations, the NN can be trained to accurately predict the patterns in the effective refractive indices. We have also illustrated the effect of various transfer function and network layout to the overall performance.

Acknowledgments

This work is supported by NRF-CRP14-2014-04, 'Engineering of a Scalable Photonics Platform for Quantum Enabled Technologies'.

ORCID iDs

Gandhi Alagappan https://orcid.org/0000-0002-4730-2503

References

[1] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[2] Courville A, Goodfellow I and Bengio Y 2016 *Deep Learning* (Cambridge: MIT Press)
[3] Hinton G *et al* 2012 Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups *IEEE Signal Process. Mag.* **29** 82–97
[4] Krizhevsky A, Sutskever I and Hinton G E 2017 Imagenet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90
[5] Chen Y, Lin Z, Zhao X, Wang G and Gu Y 2014 Deep learning-based classification of hyperspectral data *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7** 2094–107
[6] Silver D *et al* 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
[7] Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A and Kim C 2017 Machine learning in materials informatics: recent applications and prospects *Comput. Mater* **3** 54
[8] Goh G B, Hodas N O and Vishnu A 2017 Deep learning for computational chemistry *Comput. Chem.* **38** 1291–307
[9] Baldi P, Sadowski P and Whiteson D 2014 Searching for exotic particles in high-energy physics with deep learning *Nat. Commun.* **5** 4308
[10] Rivenson Y, Göröcs Z, Günaydin H, Zhang Y, Wang H and Ozcan A 2017 Deep learning microscopy *Optica* **4** 1437–43
[11] Zibar D, Wymeersch H and Lyubomirsky I 2017 Machine learning under the spotlight *Nat. Photon.* **11** 749–51
[12] Liu D, Tan Y, Khoram E and Yu Z 2018 Training deep neural networks for the inverse design of nanophotonic structures *ACS Photonics* **5** 1365–9
[13] Malkiel I, Nagler A, Mrejen M, Arieli U, Wolf L and Suchowski H 2017 Deep learning for design and retrieval of nanophotonic structures arXiv:1702.07949
[14] Ma W, Cheng F and Liu Y 2018 Deep-learning-enabled on-demand design of chiral metamaterials *ACS Nano* **12** 6326–34
[15] Kojima K, Wang B, Kamilov U, Koike-Akino T and Parsons K 2017 Acceleration of FDTD-based inverse design using a neural network approach *Advanced Photonics 2017 (IPR, NOMA, Sensors, Networks, SPPCom, PS) OSA Technical Digest* (Optical Society of America) ITu1A.4
[16] Okamoto K 2016 *Fundamentals of Optical Waveguides* 2nd edn (London: Elsevier)
[17] Chuang S L 1995 *Physics of Optoelectronic Devices* (New York: Wiley)
[18] Yu C P and Chang H C 2004 Yee-mesh-based finite difference eigenmode solver with PML absorbing boundary conditions for optical waveguides and photonic crystal fibers *Opt. Express* **12** 6165–77
[19] Rahman B M A, Fernandez F A and Davies J B 1991 Review of finite element methods for microwave and optical waveguides *Proc. IEEE* **79** 1442–8

[20] Mabaya N, Lagasse P E and Vandenbulcke P 1981 Finite element analysis waveguides of optical *IEEE Trans. Microw. Theory Tech.* **29** 600–5
[21] Gorin A, Jaouad A, Grondin E, Aimez V and Charette P 2008 Fabrication of silicon nitride waveguides for visible-light using PECVD: a study of the effect of plasma frequency on optical properties *Opt. Express* **16** 13509–16
[22] Zhao H *et al* 2015 Visible-to-near-infrared octave spanning supercontinuum generation in a silicon nitride waveguide *Opt. Lett.* **40** 2177–80
[23] Gondarenko A, Levy J S and Lipson M 2009 High confinement micron-scale silicon nitride high Q ring resonator *Opt. Express* **17** 11366–70
[24] Sun X, Alam M Z, Aitchison J S and Mojahedi M 2016 Compact and broadband polarization beam splitter based on a silicon nitride augmented low-index guiding structure *Opt. Lett.* **41** 163–6
[25] Chen L, Doerr C R and Chen Y-K 2011 Compact polarization rotator on silicon for polarization-diversified circuits *Opt. Lett.* **36** 469–71
[26] Levy J S, Foster M A, Gaeta A L and Lipson M 2011 Harmonic generation in silicon nitride ring resonators *Opt. Express* **19** 11415–21
[27] Hagan M T and Menhaj M B 1994 Training feedforward networks with the Marquardt algorithm *IEEE Trans. Neural Netw.* **5** 989–93
[28] Kisi O and Uncuoglu E 2005 Comparison of three back-propagation training algorithms for two case studies *Indian J. Eng. Mater. Sci.* **12** 434–42
[29] https://mathworks.com/help/nnet/ref/trainlm.html
[30] Heaton J 2008 *Introduction to Neural Networks for Java* 2nd edn (Chesterfield: Heaton Research)
[31] https://mathworks.com/help/nnet/ug/multilayer-neural-networkarchitecture.html
[32] https://mathworks.com/help/matlab/ref/spline.html