

Article

Inverse Design of Distributed Bragg Reflectors Using Deep Learning

Sarah Head and Mehdi Keshavarz Hedayati *

Department of Engineering, Durham University, Durham DH1 3LE, UK; sarah.b.head@durham.ac.uk
* Correspondence: mehdi.keshavarz-hedayati@durham.ac.uk

Abstract: Distributed Bragg Reflectors are optical structures capable of manipulating light behaviour, which are formed by stacking layers of thin-film materials. The inverse design of such structures is desirable, but not straightforward using conventional numerical methods. This study explores the application of Deep Learning to the design of a six-layer system, through the implementation of a Tandem Neural Network. The challenge is split into three sections: the generation of training data using the Transfer Matrix method, the design of a Simulation Neural Network (SNN) which maps structural geometry to spectral output, and finally an Inverse Design Neural Network (IDNN) which predicts the geometry required to produce target spectra. The latter enables the designer to develop custom multilayer systems with desired reflection properties. The SNN achieved an average accuracy of 97% across the dataset, with the IDNN achieving 94%. By using this inverse design method, custom-made reflectors can be manufactured in milliseconds, significantly reducing the cost of generating photonic devices and thin-film optics.

Keywords: Machine Learning; inverse design; tandem network; chromaticity; Distributed Bragg Reflector



Citation: Head, S.; Keshavarz Hedayati, M. Inverse Design of Distributed Bragg Reflectors Using Deep Learning. *Appl. Sci.* **2022**, *12*, 4877. <https://doi.org/10.3390/app12104877>

Academic Editor: Alessandro Belardini

Received: 8 April 2022
Accepted: 10 May 2022
Published: 11 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The application of optical devices is extremely broad, and over the last few decades, their prevalence has grown due to the advances in manufacturing techniques and computation power [1]. Metamaterials and nanophotonics can control the light–matter interaction precisely, enabling them to be used for a wide variety of applications. Their numerous applications include invisibility cloaking [2], structural and sustainable colour generation [3,4], high-resolution printing, anti-reflectors [5], bio-sensors [6] and nanophotonic lasers [7]. Distributed Bragg Reflectors (DBRs) are one of the classical optical systems, and are formed of alternating layers with different refractive indices, where each material interface gives rise to a partial reflection of the incident light waves [8]. A typical DBR is formed of an alternating $\lambda/4$ structure of high and low refractive index materials [9]. There are a number of applications for DBRs in photonics, such as LEDs and vertical-cavity surface-emitting lasers [10].

There exist several numerical methods and corresponding computer programs that serve to predict the responses of these thin-film structures, and over time, this information has been collated into a number of libraries. However, in practice, it is common to desire a response that has not yet been associated with a particular structure. This poses a need for inverse design (ID), which proves far more complex than direct structural analysis [11].

Traditional ID methods have yielded successful results, and are often highly accurate for small-scale modelling; however, when handling a large number of parameters, these methods simply are not feasible due to their complexity and time cost [12]. For these reasons, Machine Learning (ML) has moved to the forefront of pioneering inverse design methods [13]. ML employs a range of algorithms to analyse large amounts of data and map complex relationships between information, in order to make future predictions about new data [11,14]. Although the input data are still produced through simulations, this

computational cost occurs only once and the burden can be spread across several devices in parallel [15]. Additionally, the analytical as opposed to the numerical approach adopted by ML algorithms is far more efficient, allowing huge amounts of data to be analysed in a relatively small amount of time [4]. Table 1 presents a summary of the most prevalent inverse design methods in the literature, and their associated drawbacks.

Table 1. A summary of existing methods of inverse design and their drawbacks.

Approach	Details	Drawbacks
Systematic	By making intuitive design decisions based on photonics theory and existing structures, a starting structure is chosen and parameters are systematically cycled through until a desirable response is happened upon.	The links between structures and spectra are not always intuitive, making this method computationally costly and extremely slow.
Removal of non-unique instances followed by DL	Liningier et al. implement a feedforward neural network to design a 1–5-layer system using up to 5 different materials, with a user-imposed ‘similarity metric’ [14]. This is used to filter out training instances deemed as non-unique, effectively reducing the design challenge to a one-to-one problem.	The ‘similarity threshold’ chosen by the user is arbitrary and may not catch all similar spectra, which could prevent model convergence.
DL tandem model	Works by Liu et al. and Xiaopeng et al. use a Tandem Neural Network to address non-uniqueness [15,16]. This is composed of two networks: the first designing a reflector to produce a target spectra, while the second pretrained network determines how well the target and predicted spectra match, regardless of the intermediate structural predictions.	Struggles with classification design problems, as the intermediate network output needs to be in an acceptable form to be fed into the secondary network.
DL mixture-density model	Unni et al. also use two networks, the second again being pretrained to map structural components to spectral response [17]. However, the two are not connected, and instead the first network outputs the probability of a number of structures, and a separate optimisation process evaluates each possibility with reference to its performance in the second network.	Requires an optimisation process on top of a second pretrained network, and multiple possible designs must be tested and verified for each design case.

Within recent years, the tandem approach has become the most popular method of addressing non-uniqueness. To give a more comprehensive overview, a tandem network is established first with an Inverse Design Neural Network (IDNN), the output of which acts as the input of a forward modelling network, as shown in Figure 1.

This forward model (sometimes referred to as the simulation neural network (SNN) [16] to distinguish between the forward design and forward propagation) has been pretrained, meaning that its parameters are locked throughout the tandem model training. The calculated network loss is taken as the difference between the desired spectral response (fed into the inverse model as an input), and the predicted spectral response given by the pretrained forward model [15]. Any structure produced by the inverse model is directly fed into the pretrained SNN, which is known to converge. In taking this approach, we can minimise the non-uniqueness problem, as, regardless of the predicted material structure, the model will be assessed only on its ability to produce a structure with the desired spectral response [15].

It is clear from this that the SNN model is an integral part of the tandem network, as inaccuracies here will prevent the inverse model from producing valid results [11].

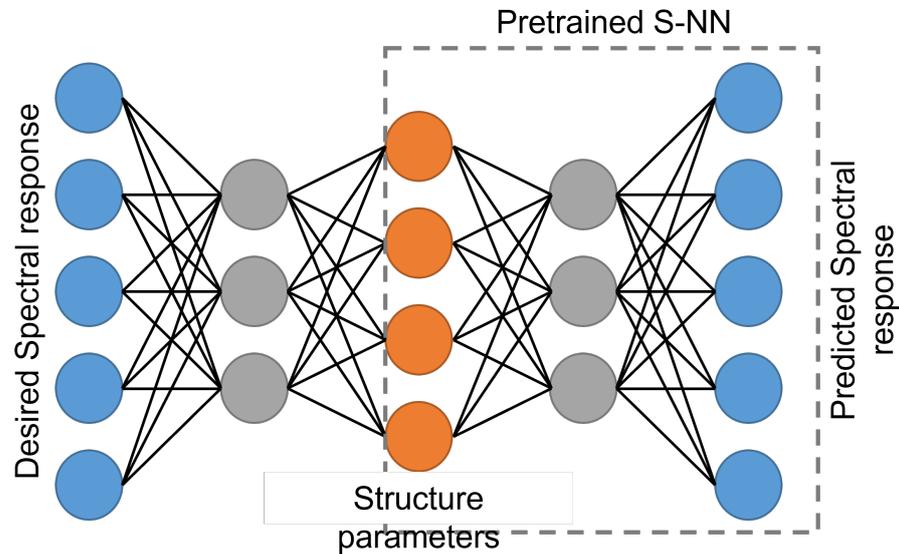


Figure 1. A simplified tandem network showing the pre-trained SNN network as connected to the inverse network. The hidden layers are represented by the grey nodes, where, in practice, there would be multiple layers.

The aim of this study is to accurately produce on-demand reflector designs using a Deep Learning approach to analyse a tightly constrained thin-film structure. A tandem network approach is adopted to address the non-uniqueness problem, and to achieve an accurate and versatile model using a reduced dataset.

2. Materials and Methods

The reflector design has been simplified into a six-layer structure composed of alternating layers of SiO₂ and Si₃N₄, with each layer ranging from 10 to 60 nm. An example of this is shown in Figure 2.

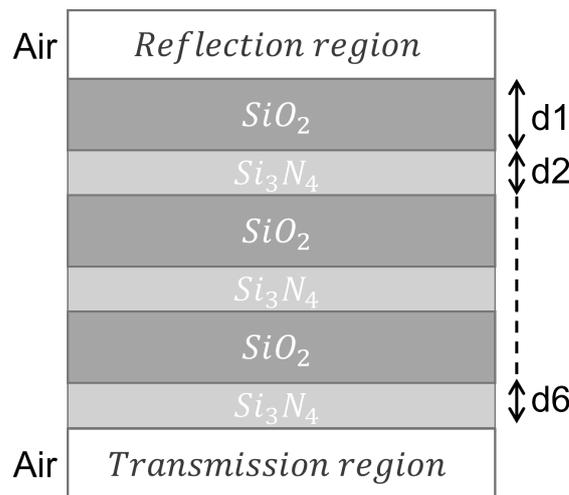


Figure 2. A 6-layer thin-film structure of alternating layers of SiO₂ and Si₃N₄, all of some depth d_i . The transmission and reflectance media surrounding the structure are both half-infinite free space.

These are a traditional choice of dielectrics, and for good reason. SiO₂ is a low-refractive-index material (n in the range of 1.45–1.47 within the visible spectrum), and

is usually used alongside a high-index material. In this case, this is Si_3N_4 , which has a corresponding refractive index in the range of 1.99–2.08 [18]. Together, they are able to produce highly reflective surfaces when in the right combination. Furthermore, several methods have been developed for the stable production of both SiO_2 and Si_3N_4 films, making them a practical choice [19].

The following methodology analyses a range of six-layer thin-film structures, bounded on each side by free space and subject to normal incident light waves.

2.1. Data Generation

The desire for relatively small-scale data generation requires careful attention to the choice of samples, to ensure enough diversity to train a useful model and simultaneously encapsulating enough similar data that accurate input–output mapping can occur. The examination of possible material depths within the chosen range yielded 46,656 samples, where material depths were constrained to multiples of 10 nm. The intention was to split these data in half, where alternating samples were taken, so that both the SNN and IDNN could be trained on similar yet unique data. Following this, information regarding refractive indices (n) and extinction coefficients (k) was compiled from an online refractive index database [18] for the two materials. Data were taken for wavelengths between 380 and 780 nm, over which range both materials had a negligible k value. The raw data were then processed through a simple program *wavelengths.m* (see data availability for information on accessing code), which linearly extrapolated between the data points to produce uniformly spaced refractive index values for each material.

Finally, these values were input to a program based on the works by Rao [20], *tmm-Comp.m* (see data availability for information on accessing code), along with a file containing the structural combinations, and the TMM used to evaluate each case. The output of this program was the reflectance spectrum of each structure within the visible light range.

2.2. Feature Encoding

The outputs from the TMM code were taken as the spectral data (with each feature of length 81), whereas the structural data were the same as those used in the TMM process. The resulting features were of the form $[d_1, d_2, \dots, d_6]$, where each d_i refers to the depth at layer i . To make bounding the depth predictions easier later on, all structural values were scaled to between 0 and 1.

Depending on the network in question, the features can be read as either inputs or desired outputs. To save confusion, from herein, they will be referred to as either the structural properties or reflectance spectra, where these can be both predicted or target values.

2.3. Simulation Neural Network

A fully connected neural network (FCNN) with four hidden layers was implemented, where network inputs were processed through a batch normalisation layer to improve convergence, and eliminate the need for further standardisation [21]. The input and output layers hold 6 and 81 neurons, which represent the 6 layer thicknesses and the discretised reflectance spectra, respectively. The mean squared error (MSE) loss function was chosen to suit the regression task, and the Adam optimiser was chosen for combined convergence speed and quality [22]. Finally, a learning rate of 0.001 was used. All layers initially used the ReLU activation function, except for the final layer, which used a sigmoid function to bound the predictions to between 0 and 1.

Further refinements looked at adding model complexity through altering layer sizes and adding dropout. Table 2 summarises some of the key metrics produced by different layer architectures. It was found that optimal performance was achieved by increasing only the final layer to 1000 neurons, but that additional neurons saw no significant improvements.

Table 2. A sample of trialled SNN model architectures and their associated accuracy metrics.

Hidden Layer Sizes	Dropout	Mean Absolute Error
500, 500, 500, 500	None	0.0047
500, 500, 500, 1000	None	0.0039
500, 500, 1000, 1000	None	0.0039
500, 500, 500, 1000	2%, 2%, 1%, 0	0.0041
500, 500, 500, 1000	2%, 1%, 1%, 0	0.0044
500, 500, 500, 1000	5%, 2%, 1%, 0	0.0040
500, 500, 500, 1000	2%, 1%, 0, 0	0.0037

Following these trials, a structure of 6-500-500-500-1000-81 neurons was chosen, with 2% and 1% dropout in hidden layers 1 and 2, respectively, as this presented the best trade-off between predictive accuracy and overfitting.

Dropout was not added to the later layers, as, this late in the network, there is less opportunity for the model to correct the potential errors that it could introduce.

To help optimise the convergence, the model was updated to take an adaptive learning rate. This makes use of the accuracy of a small learning rate where needed, whilst reducing the chance of getting trapped in local minima at an early stage [21]. It works by reducing the learning rate if the validation loss plateaus for more than a specified number of epochs (known as the patience) [22]. After some experimentation, this was chosen as 6, and the learning rate decreased logarithmically each time it updated until reaching 1×10^{-6} , below which the network training became unfavourably slow.

Lastly, to try and reduce the final training errors, the ReLU activation functions were swapped for leaky ReLU functions. The difference between these is that the leaky variety has a user-defined non-zero gradient for negative values. This helps to solve the ‘dying neuron’ problem, as, now, if certain weights enter the negative region, their training will not completely stop [23]. The hyperparameter α specifies how much the function ‘leaks’—in other words, the gradient of the function for negative values of x . Here, α has been chosen as 0.1.

A summary of the current network architecture is included in Table 3.

Table 3. Summary of final SNN model architecture.

Layer	Details	Number of Parameters
Input shape	Output shape = 6	0
Dense layer	Output shape = 500	3500
Dropout layer	2% dropout	0
Dense layer	Output shape = 500	250,500
Dropout layer	1% dropout	0
Dense layer	Output shape = 500	250,500
Dense layer	Output shape = 1000	501,000
Dense layer	Output shape = 81	81,081

The model was trained on the first half of the dataset mentioned in Section 2.1, where the data were first shuffled and then split into training, validation, and test data at roughly a 6:1:0.8 split.

Following this, the model was saved so that it could be imported into a new file later, with its parameters already determined.

2.4. Tandem Neural Network

To form the tandem network, first, an IDNN model was created in much the same way as for the SNN. The main differences were that the layers tapered from largest to smallest, to correspond with the larger spectral inputs and smaller material outputs, and the number of hidden layers was increased from 4 to 6 to deal with the slightly more complex problem. For the same reason, the first three layers hold 1000 neurons each, and the remaining three

have 500. This required the dropout values to be tweaked accordingly to be 2% and 1% in hidden layers 3 and 4, respectively. Finally, the output layer used a sigmoid activation function to bound the material predictions to a form recognisable by the SNN.

After freezing the parameters to prevent further training, the tandem model was formed by cascading the SNN and IDNN, and taking the MSE between the IDNN input and the SNN output as the model loss function. The final IDNN model architecture can be seen in Table 4.

Table 4. Summary of final IDNN model architecture.

Layer	Details	Number of Parameters
Input shape	Output shape = 81	0
Dense layer	Output shape = 1000	82,000
Dense layer	Output shape = 1000	1,001,000
Dense layer	Output shape = 1000	1,001,000
Dropout layer	2% dropout	0
Dense layer	Output shape = 500	500,500
Dropout layer	1% dropout	0
Dense layer	Output shape = 500	250,500
Dense layer	Output shape = 500	250,500
Dense layer	Output shape = 6	3006

3. Results and Discussion

3.1. SNN Model Performance

The mean absolute errors for the network discussed in Section 2.3 are shown in Figure 3, both with and without the addition of dropout. Without dropout, the training and validation curves converge at a very similar level, with the validation dataset slightly underperforming compared to the training dataset, suggesting a small amount of overfitting. With the application of dropout, the curves switch places, and the training curve converges at a noticeably higher level. This is unlikely due to the selection of validation samples as this is a direct result of applying dropout, and so is likely caused due to the effective ‘shrinking’ of the network layers that occurs when training the model. When testing on validation and test data, all neurons are used again, which can result in a marginal increase in performance.

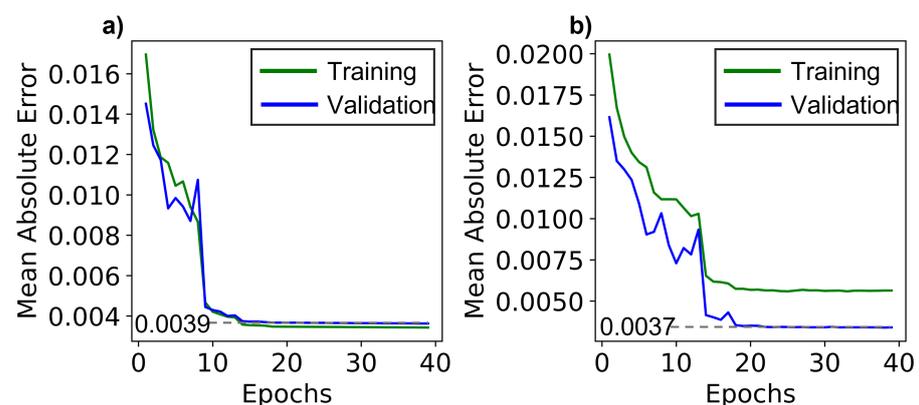


Figure 3. Learning curves showing the mean absolute error (MAE) of the SNN network without (a) and with (b) the use of dropout. Curves representing both the training and validation datasets are shown, and the final MAE of the validation curves highlighted.

The training begins to slow after around 12 epochs, where one can see that the curves begin to smooth out as a result of the adaptive learning rate. With dropout, the mean absolute error for the validation dataset is 0.0037, and it is 0.0036 for the test dataset. Considering that the range of possible reflectance values is 1, this is a strong result; however,

to further analyse the model performance, predicted spectra were examined. Predictions were made for each sample in the test dataset, and by comparing these to the targets, the resulting percentage accuracy was calculated using (1).

$$Accuracy = \frac{1}{N} \sum_{i=1}^N \frac{|R'_i - R_i|}{R_{avg}} \times 100 \quad (1)$$

where N is the number of test vectors, R'_i is the predicted spectrum, R_i the ground truth, and R_{avg} is the average value of each spectrum. The 2328 test inputs yielded an average accuracy of 97.2%, with a possible accuracy of up to 99.4%. Figure 4 shows two sample spectra, one representing a 99% accurate graph, and one showing one of the lower-performing graphs. One can see that although the accuracy is only 77% for Figure 4a, the absolute errors are small. This is due to the way in which the accuracy was defined in (1). By dividing by the average reflectance value for each case, spectra with consistently low levels of reflectance will yield a higher percentage error for the same absolute errors. Nevertheless, this was preferable to dividing by the actual value, which could produce huge errors for expected values close to zero.

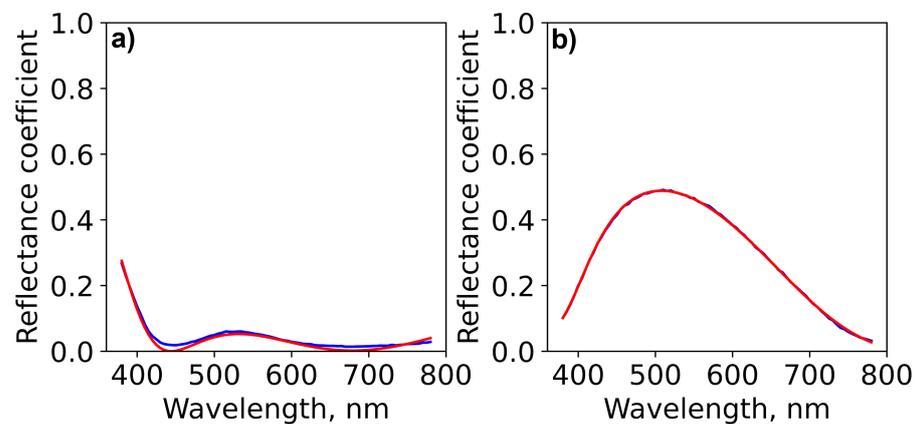


Figure 4. The predicted (blue) and anticipated (red) reflectance spectra produced by the SNN for two different structures, demonstrating its ability to predict a variety of spectral shapes. The percentage accuracies for (a,b) are 77% and 99%, with layer depths [21 10 50 30 60 60] and [40 50 30 10 60 50] nm, respectively.

3.2. Tandem Model Performance

The IDNN network was analysed by first looking at the tandem network performance, and then validating the predictions made by the IDNN.

Similarly to the SNN, the tandem network training curves were evaluated for a range of model architectures. It was seen that even without the application of dropout, the validation curve converged to a lower error. This is most likely due to the presence of dropout in the SNN model; however, it was seen that low levels of dropout in the IDNN did still increase the performance further by helping model generalisation.

The final mean absolute errors were 0.00770 and 0.00745 for the validation and test datasets, respectively. As expected, this is marginally higher than for the SNN, as this is both a more complex system and must incorporate the errors of both the IDNN and the SNN.

The average accuracy was found to be 94.0%, with a maximum accuracy of 99.8% and a minimum accuracy of 89.52%, which appears to outperform the network presented in [16]. Breaking it down further, it was found that 68.4% of test samples were predicted with greater than 95% accuracy, and 79.5% above 90%. Figure 5 shows these accuracies presented visually.

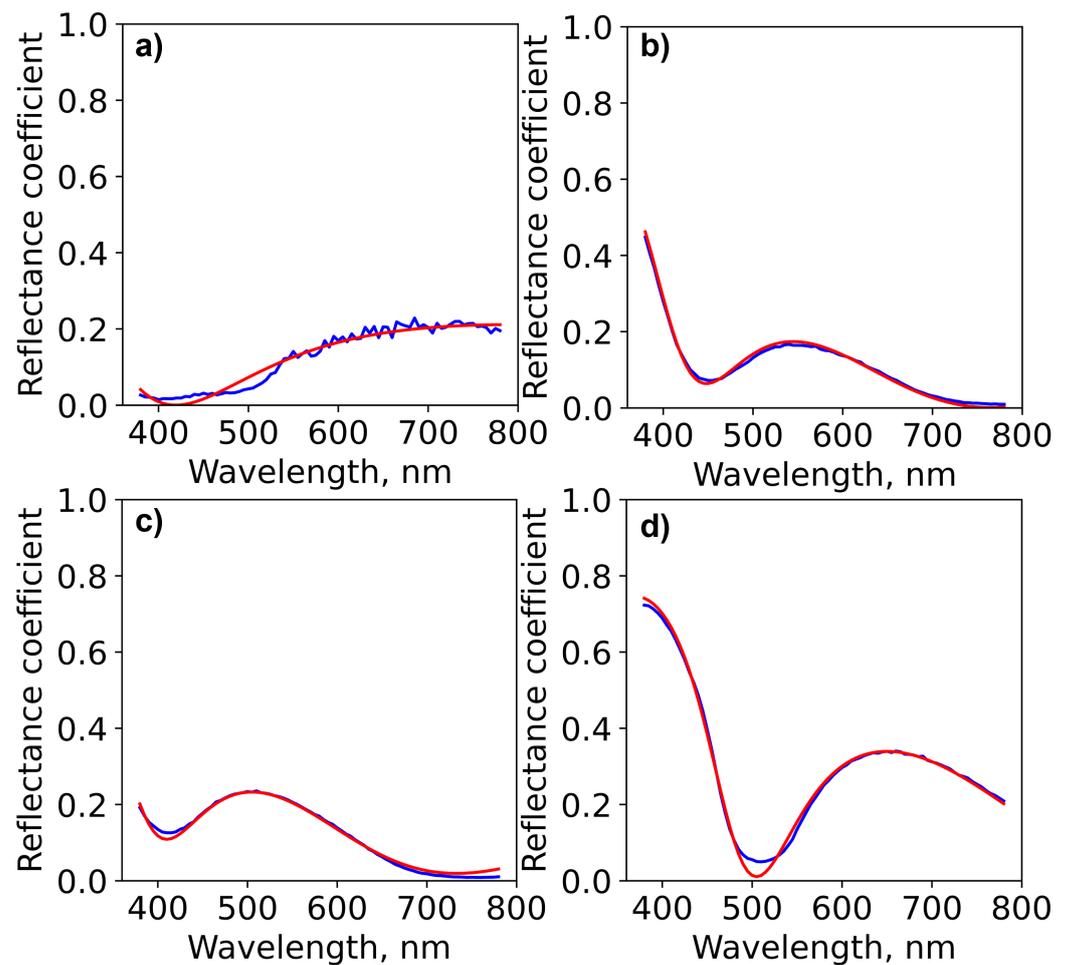


Figure 5. A selection of graphs showing example plots corresponding to various levels of accuracy. These are 90%, 95%, 94%, and finally 97%, where the predicted layer depths were [83.3 56.4 89.6 46.0 66.7 3.2], [27.1 25.6 58.8 49.0 65.4 26.9], [27.4 43.8 31.2 42.5 52.9 14.8] and [69.5 63.5 61.5 38.9 17.9 21.9] nm for (a–d), respectively.

Figure 5a represents an accuracy of 90%, and shows some deviation from the target curve. However, much of the error here could be reduced though smoothing the predicted curve, as the overall shape is relatively close. Figure 5b represents a 95% accurate graph, and one can see that the predicted values match the target well, both in terms of the reflectance value and the location of the peaks and troughs along the x -axis. The latter is of more concern, as the overall shape of the spectrum has a greater effect on the perceived colour than relatively small changes in reflectance intensity. Moving to Figure 5c shows a 94% accurate plot set to represent the average accuracy. There is a small divergence between the curves at around 380–400 nm. Lastly, Figure 5d was chosen to demonstrate the model behaviour with higher reflectance value targets. This was achieved here with 97% accuracy, suggesting that the model was exposed to several instances of high reflectance.

To fully evaluate the IDNN, the material parameters were extracted for a number of test samples and checked for validity. The predictions were already bounded by the sigmoid function, meaning that negative values would not occur, but unfeasibly small or large depth predictions were also checked for. Additionally, several predictions were fed back into the TMM code to check that the analytical and numerical methods were in agreement. A few such examples are given in Figure 6, and have the material depth values of [35.6 52.7 36.9 22.1 46.1 72.0] and [47.1 25.3 23.8 22.6 51.8 68.4] nm, respectively. Here, the dashed curves represent the Matlab simulated responses, and the solid curves are the tandem model predictions.

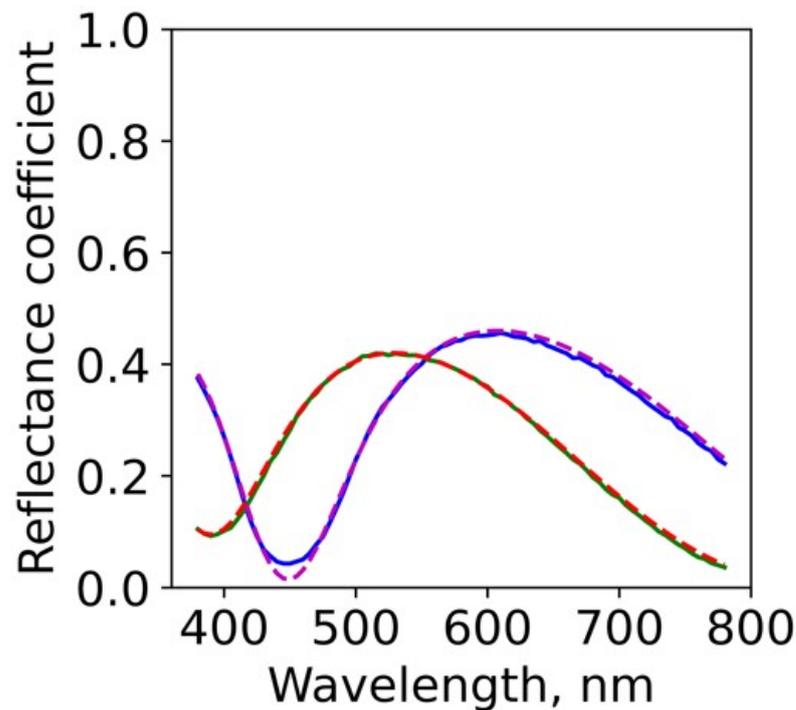


Figure 6. Model output (solid lines) compared with TMM calculations using the model predictions (dashed lines) for two sample structures with material depths [35.6 52.7 36.9 22.1 46.1 72.0] and [47.1 25.3 23.8 22.6 51.8 68.4] nm, respectively. The strong similarities indicate good agreement between the analytical and numerical methods.

Overall, the predictions and calculations match well, which, alongside the high spectral prediction accuracy, means that the model will perform well when designing structures within the scope of the materials present. On top of this, the model is able to make predictions in between 100th and 10th of a second, which is vastly faster than any other inverse design method.

However, examining a number of spectra from the training and test datasets, it is apparent that there are many similarly shaped curves, often with shallow peaks and troughs. This brings into question the ability of the model to predict more diverse spectra. To simulate the model being used, a number of arbitrary curves were used as input spectra. These were Gaussian-shaped curves with a range of peak-heights, positions, and widths.

The results of this were mixed, with some targets being met well, and others performing poorly. This is unsurprising, as the model is only able to predict within the constraints of the patterns observed in the training data, which in turn are constrained by material and structural properties. This meant that relatively shallow spectra, such as that shown in Figure 7, were predicted well with accuracies upwards of 85/90%, but that spectra including high-intensity narrow peaks had significantly lower accuracies.

This was verified by making predictions on a number of shallow spectra only, where it was seen that the model was able to predict all targets to above 74% accuracy, with many above 94%, where the larger errors were a result of the low reflectance values as opposed to greater absolute errors. This strongly suggests that the observed limitations are due to the choice of material structure, rather than inaccuracies within the model. Exploring more complex material arrangements would likely increase the range of spectra that could be produced without needing to alter the core model architecture. There is also the potential to adjust the two materials used, where differences in refractive index could create different responses.

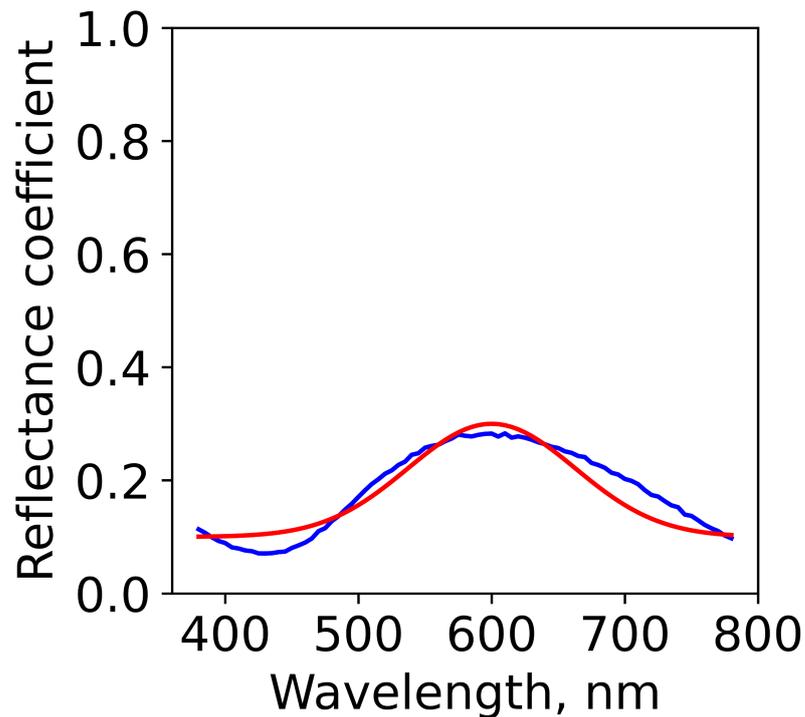


Figure 7. An example of model performance when asked to design a structure to produce an arbitrary response within the material capabilities. The predicted (blue) curve shows good similarity to the target (red) curve, with the peaks and troughs in good agreement and a percentage accuracy of 88%. The predicted material depths were [61.4 49.4 65.3 46.5 52.6 12.1] nm.

The case should also be considered where a DBR spectrum is desired, for which there is no possible solution given photon absorption within the chosen materials. Training data are produced using the TMM, which takes n and k values into consideration, and therefore should account for the effect of absorption where applicable. When designing a DBR, training data should be provided that cover a range of wavelengths, including those which would not produce viable DBRs. This will prevent the model from inferring responses based on only a few training cases. It follows that if the user requests a spectrum that is outside the capabilities of the material in use, then the model will fail to produce a reflector design which performs satisfactorily. Whether a spectrum is satisfactory or not is up to the user, where this can be assessed by comparing the tandem model output to the target spectrum. Where a spectrum is not feasible, it can be assumed that an alternative material pair is needed.

3.3. Colour Generation by Multilayers

The model was further assessed through exploring the range of colours produced by each structure. From here, only the training data for the desired configuration need be generated. This allows for far sparser data sampling, and is perhaps a more intuitive way of demonstrating the versatility of each structure. Additionally, only by making very small changes to the model structure is it likely that (other than a small adjustment to the input vector size) the DL model architecture would not need to be altered.

Using an adapted code by Scardina [24], the reflectance spectra data were converted into CIE chromaticity values of L^* , a^* , and b^* , which were then converted to the tristimulus values (XYZ) using Matlab. Finally, using the relations in Equation (3.3), the xy values were found, allowing the 2D chromaticity plot seen in Figure 8 to be constructed. The black data points represent a sample of the model predictions made for the test data, and therefore are an accurate indicator of the current model's true capabilities.

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z} \quad (2)$$

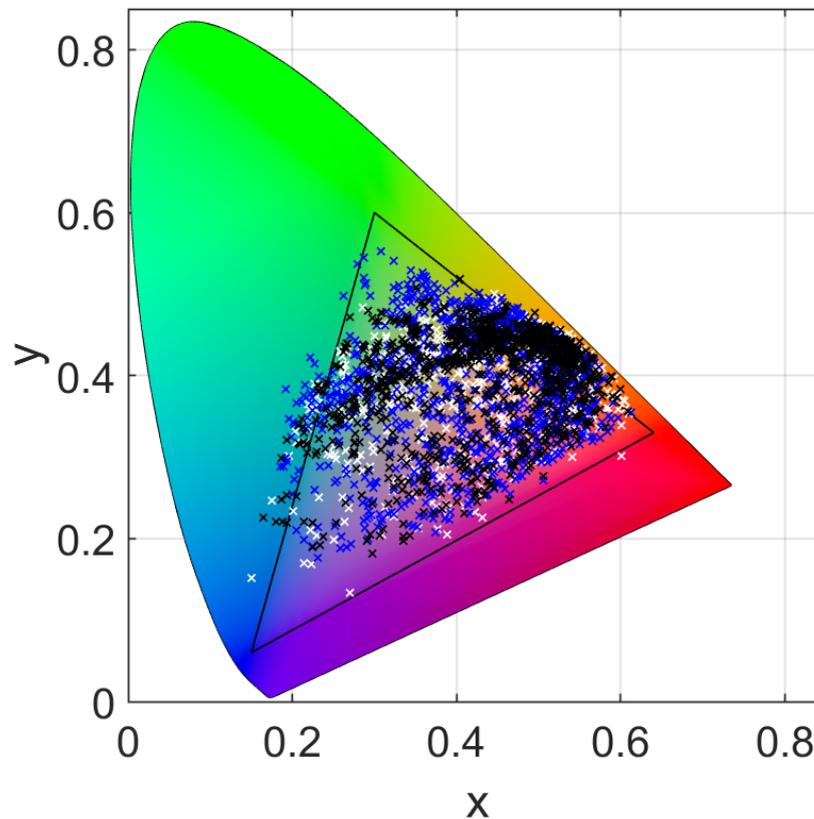


Figure 8. A 2D chromaticity plot showing the RGB colour space (triangle plot). Black crosses represent the unique colours produced by each of the model predictions made using the test data for the $\text{SiO}_2\text{-Si}_3\text{N}_4$ system. White crosses show the breadth of a 10-layer system, and blue crosses show the results of the same 6-layer system using MgO-ZrO_2 as the material pair.

The triangular plot represents the scope of the RGB colour space. It can be seen from this that the model was able to accurately produce colours across the yellow/orange and pink/purple ranges, as well as pale greens and turquoises. However, it struggles with vibrant greens and reds, and deep blue–purples, which lie at the corners of the RGB space. This is likely due to these colours being derived from narrow reflectance peaks, which are not producible using the tightly constrained model.

Further work could focus on assessing small structural changes which could be made in order to improve the range of the model predictions. The white and blue data points show the additional breadth of responses which can be produced by a 10-layer $\text{SiO}_2\text{-Si}_3\text{N}_4$ structure and a six-layer MgO-ZrO_2 structure, respectively. It is apparent that the 10-layer structure extends further into the purple–blue and red colour spaces, but also covers the regions achieved with the original structure. Additionally, the MgO-ZrO_2 data are able to achieve the intense greens that were otherwise lacking, and extends even past the RGB colour space into the blue–turquoise regions. Green colour is a huge advantage of this multilayer system over conventional metamaterial-based structures [4]. The simplicity of this model and the smaller quantity of data needed to train it as compared to previous studies [15,16] present the opportunity for the same model architecture to be used alongside a number of training datasets, each of which may be tuned to reflect the materials/structural constraints imposed by the user.

4. Conclusions

In this study, we used Deep Learning to design a network that can accurately produce a Distributed Bragg Reflector (DBR) on demand. We successfully developed a Simulation Neural Network (SNN) with a promising mean absolute error of 0.0036 and a percentage accuracy of 97% for the test dataset. This was combined with an Inverse Design Neural Network (IDNN) to form a tandem network, which was able to produce predictions with a test mean absolute error of 0.00791, and an average accuracy of 94% for a range of inputs. This proves the capability of Deep Learning in the inverse design of multilayer systems, and could be applied to the photonic device industry while keeping the design and processing costs low. To demonstrate the model's versatility, reflectance predictions for the test dataset were then converted into CIE colour space values. These were plotted on a chromaticity diagram to demonstrate the possible colours which were within the scope of the design constraints. It was shown that a broad colour variation was achieved, and that marginal changes to the material structure allow for the generation of vivid colours at the corners of the RGB colour space.

Author Contributions: Conceptualization, supervision, M.K.H.; methodology, writing—original draft preparation, formal analysis, data curation, S.H.; writing—review and editing, M.K.H. and S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this research are available to the public at <https://github.com/sblHead/Supplementary-code>, accessed on 8 April 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qu, Y.; Jing, L.; Shen, Y.; Qiu, M.; Soljačić, M. Migrating Knowledge between Physical Scenarios Based on Artificial Neural Networks. *ACS Photonics* **2019**, *6*, 1168–1174. [[CrossRef](#)]
2. Schurig, D.; Mock, J.J.; Justice, B.; Cummer, S.A.; Pendry, J.B.; Starr, A.F.; Smith, D.R. Metamaterial electromagnetic cloak at microwave frequencies. *Science* **2006**, *314*, 977–980. [[CrossRef](#)] [[PubMed](#)]
3. Caligiuri, V.; Tedeschi, G.; Palei, M.; Miscuglio, M.; Martin-Garcia, B.; Guzman-Puyol, S.; Hedayati, M.K.; Kristensen, A.; Athanassiou, A.; Cingolani, R.; et al. Biodegradable and insoluble cellulose photonic crystals and metasurfaces. *ACS Nano* **2020**, *14*, 9502–9511. [[CrossRef](#)] [[PubMed](#)]
4. Roberts, N.B.; Keshavarz Hedayati, M. A deep learning approach to the forward prediction and inverse design of plasmonic metasurface structural color. *Appl. Phys. Lett.* **2021**, *119*, 061101. [[CrossRef](#)]
5. Keshavarz Hedayati, M.; Abdelaziz, M.; Etrich, C.; Homaeigohar, S.; Rockstuhl, C.; Elbahri, M. Broadband anti-reflective coating based on plasmonic nanocomposite. *Materials* **2016**, *9*, 636. [[CrossRef](#)] [[PubMed](#)]
6. Zhou, C.; Keshavarz Hedayati, M.; Zhu, X.; Nielsen, F.; Levy, U.; Kristensen, A. Optofluidic sensor for inline hemolysis detection on whole blood. *ACS Sens.* **2018**, *3*, 784–791. [[CrossRef](#)] [[PubMed](#)]
7. Gaio, M.; Saxena, D.; Bertolotti, J.; Pisignano, D.; Camposeo, A.; Sapienza, R. A nanophotonic laser on a graph. *Nat. Commun.* **2019**, *10*, 226. [[CrossRef](#)] [[PubMed](#)]
8. Malekovic, M.; Bermúdez-Ureña, E.; Steiner, U.; Wilts, B.D. Distributed Bragg reflectors from colloidal trilayer flake solutions. *APL Photonics* **2021**, *6*, 026104. [[CrossRef](#)]
9. Sugawara, H.; Itaya, K.; Hatakoshi, G. Characteristics of a distributed Bragg reflector for the visible-light spectral region using InGaAlP and GaAs: Comparison of transparent- and loss-type structures. *J. Appl. Phys.* **1993**, *74*, 3189–3193. [[CrossRef](#)]
10. Schubert, M.F.; Xi, J.Q.; Kim, J.K.; Schubert, E.F. Distributed Bragg reflector consisting of high- and low-refractive-index thin film layers made of the same material. *Appl. Phys. Lett.* **2007**, *90*, 141115. [[CrossRef](#)]
11. Liu, Z.; Zhu, D.; Raju, L.; Cai, W. Tackling Photonic Inverse Design with Machine Learning. *Adv. Sci.* **2021**, *8*, 2002923. [[CrossRef](#)] [[PubMed](#)]
12. Molesky, S.; Lin, Z.; Piggott, A.; Jin, W.; Vucković, J.; Rodriguez, A. Inverse design in nanophotonics. *Nat. Photonics* **2018**, *12*, 659–670. [[CrossRef](#)]
13. Zhang, J.; Luo, Y.; Tao, Z.; You, J. Graphic-processable deep neural network for the efficient prediction of 2D diffractive chiral metamaterials. *Appl. Opt.* **2021**, *60*, 5691–5698. [[CrossRef](#)] [[PubMed](#)]

14. Lininger, A.; Hinczewski, M.; Strangi, G. General Inverse Design of Thin-Film Metamaterials with Convolutional Neural Networks. *arXiv* **2021**, arXiv:2104.01952.
15. Liu, D.; Tan, Y.; Khoram, E.; Yu, Z. Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures. *ACS Photonics* **2018**, *5*, 1365–1369. [[CrossRef](#)]
16. Xu, X.; Sun, C.; Li, Y.; Zhao, J.; Han, J.; Huang, W. An improved tandem neural network for the inverse design of nanophotonics devices. *Opt. Commun.* **2021**, *481*, 126513. [[CrossRef](#)]
17. Unni, R.; Yao, K.; Han, X.; Zhou, M.; Zheng, Y. A mixture-density-based tandem optimization network for on-demand inverse design of thin-film high reflectors. *Nanophotonics* **2021**, *10*, 4057–4065. [[CrossRef](#)]
18. Polyanskiy, M.N. Refractive Index Database. Available online: <https://refractiveindex.info> (accessed on 14 November 2021).
19. Lei, P.H.; Wang, S.H.; Juang, F.S.; Tseng, Y.H.; Chung, M.J. Effect of SiO₂/Si₃N₄ dielectric distributed Bragg reflectors (DBRs) for Alq₃/NPB thin-film resonant cavity organic light emitting diodes. *Opt. Commun.* **2010**, *283*, 1933–1937. [[CrossRef](#)]
20. Rao, S. Transmittance and Reflectance Spectra of Multilayered Dielectric Stack Using Transfer Matrix Method. MATLAB Central File Exchange. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/47637-transmittance-and-reflectance-spectra-of-multilayered-dielectric-stack-using-transfer-matrix-method> (accessed on 8 April 2022).
21. Chollet, F. *Deep Learning with Python*, 2nd ed.; Manning Publications: New York, NY, USA, 2018.
22. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reiley: Springfield, MO, USA, 2019.
23. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
24. Bottura Scardina, S. Colour Converter—Reflectance Spectra to CIE1964 Space. 2022. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/87467-colour-converter-reflectance-spectra-to-cie1964-space> (accessed on 8 April 2022).