Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

LETTER • OPEN ACCESS

Determining optical constants of 2D materials with neural networks from multi-angle reflectometry data

To cite this article: Ergun Simsek 2020 *Mach. Learn.: Sci. Technol.* **1** 01LT01

View the article online for updates and enhancements.

MACHINE LEARNING
Science and Technology

Determining optical constants of 2D materials with neural networks from multi-angle reflectometry data

Ergun Simsek

University of Maryland Baltimore County, Department of Computer Science and Electrical Engineering, 1000 Hilltop Circle, Baltimore, MD 21250, United States of America

E-mail: simsek@umbc.edu

Abstract

Synthetically generated multi-angle reflectometry data is used to train a neural network based learning system to estimate the refractive index of atomically thin layered materials in the visible part of the electromagnetic spectrum. Unlike previously developed regression based optical characterization methods, the prediction is achieved via classification by using the probabilities of each input element belonging to a label as weighting coefficients in a simple analytical formula. Various types of activation functions and gradient descent optimizers are tested to determine the optimum combination yielding the best performance. For the verification of the proposed method's accuracy, four different materials are studied. In all cases, the maximum error is calculated to be less than 0.3%. Considering the highly dispersive nature of the studied materials, this result is a substantial improvement in terms of accuracy and efficiency compared to traditional approaches.

1. Introduction: deep learning and optical material characterization

With the availability of open-source and easy to use libraries [1, 2] and graphics processing units at affordable prices, researchers from various disciplines of science and engineering are using artificial neural networks to learn from and make predictions on data in various forms. Optical material characterization based on reflectometry (or ellipsometry) data is one of these applications, where deep learning has been implemented to identify two-dimensional (2D) nanostructures [3–6] and to obtain optical constants of particles [7], thin films [8, 9], solutions [10], tissues [11], and soils [12]. This work focuses on determining optical constants of atomically thin layered materials as follows.

In 2004, Novoselov *et al* experimentally showed that 2D atomic crystals of carbon could be obtained with a top-down exfoliation technique [13]. Later we have learned that other types of materials, such as transition metal dichalcogenides, metal oxides, and boron nitride can be obtained in the monolayer or a-few layers form [14–17]. Depending on their family, these 2D materials exhibit unique properties such as high electrical conductivity, thermal stability, and mechanical strength [13, 15, 18]. With more sophisticated synthesis methods, such as chemical vapor deposition (CVD) that can generate large area and high-quality samples, they become attractive materials with a promising potential to be used in future optical, opto-electronic, and quantum devices.

In terms of device design and realization, precise knowledge of optical and electrical properties is a must and accurate methods are needed to extract these properties. However, 2D materials exhibit another important feature: their optical properties change with wavelength, temperature, doping, and external factors such as applied gate voltage and magnetic fields [15–18]. Each time one or more of these parameters change, the refractive index changes. Hence, the method used for optical property extraction should be not only accurate but also efficient. As previously mentioned [3–11], deep learning is a promising field for this very purpose, which can be used in two different ways: regression [8] or classification [12, 19]. Since previous regression based

approaches have relatively high prediction errors, e.g. 5% in [8], we follow the second approach to predict a continuous output from discrete classes.

This paper is organized as follows. Section 2 explains how refractive index can be evaluated numerically from reflectometry data. Section 3 describes our neural network implementation. Section 4 discusses the accuracy and efficiency of the implementation. Conclusions and references are provided at the end. The data that support the findings of this study are openly available [20].

2. Numerical determination of optical constants of 2D materials

Previously, we have carried out both theoretical and experimental studies on various types of 2D materials including graphene [18], monolayers of MoS_2 [15, 16] and WSe_2 [17]. For example in [15], we develop a numerical method to extract the refractive index of monolayer MoS_2 that are synthesized on quartz and silicon substrates with CVD. Reflectance spectra are measured as functions of wavelength, incidence angle, and polarization. The two different lamps are used to cover the entire visible spectrum. Incidence angle is changed from $0°$ to $60°$ at the steps of $5°$. Two sets of measurements are carried out for *s*- and *p*-polarizations. In each case, we take two measurements: R_{wo} is the one from bare substrate (i.e. without MoS_2) and R_w is the one from coated substrate (i.e. with MoS_2). Then the differential reflectance spectrum is obtained with $\Delta R = (R_w - R_{wo})/R_{wo}$. One can obtain the refractive index of MoS_2 as a function of wavelength from these measurements as follows.

For a non-magnetic multilayered structure, where each layer is defined with its refractive index and thickness, electromagnetic waves' transmission (T) through and reflection (R) from that structure can be calculated numerically [21]. For a multilayered structure, where one of the layer's thickness is known but its refractive index is an unknown, one can calculate T and R for a wide range of complex refractive index values and compare these numerical values to experimental ones. The refractive index value that yields T and R closest to the experimental T and R values, can be taken as a reference point and new set of T and R values can be calculated numerically in the vicinity of this reference point. The permittivity value that yields T and R closest to the experimental results can be taken as the new reference point. This procedure can be repeated several times until the desired accuracy is reached (i.e. difference between numerical and experimental results is smaller than 10^{-5}). Since R–T calculations are needed for each iteration, this process is not only slow but also constraining from a computational point of view, such that the search algorithm and R–T calculator need either to be written in the same programming language or to have a communication protocol between different platforms (e.g. MATLAB Engine API for Python).

Here we would like to take advantages of neural networks for two purposes. First, we would like to create a framework which could calculate the refractive index of 2D materials accurately without requiring any additional R–T calculation. Second, this framework would achieve this prediction in a few seconds or less for any given set of measurements conducted on any type of atomically thin layered material fabricated on top of silicon and glass substrates. In this direction, we first create a training dataset [20] by calculating differential reflectance spectra in a similar fashion to our experimental work [15]. We assume two different substrates: quartz and SiO_2 coated silicon (Si) substrates. The wavelength dependent refractive indices of quartz, SiO_2, and Si are taken from [22]. The thickness of SiO_2 is assumed to be 285 nm, which is one of the two standard values used in 2D material research. For each substrate, we calculate the reflectance from these substrates with and without a 2D material on the top. The thickness of the 2D material is taken as 0.7 nm [14, 15, 18]. At a chosen wavelength, ranging from 400 nm to 800 nm, we calculate differential reflectance for incidence angles changing from $0°$ to $60°$ at the steps of $1°$ for both polarizations. We carry out these calculations for 3060 unique refractive index values where the real part (n) changes from 0.1 to 6 and imaginary part (k) changes from 0 to 5, both at the steps of 0.1. We treat each of these 3060 indices as classes.

In order to determine the best configuration for the neural network based learning system and testing, we primarily use the refractive index provided in [15] for a monolayer MoS_2 at room temperature. In section 4, we provide other results obtained on monolayers of $MoSe_2$, WS_2, and WSe_2. Their refractive indices are taken from [14].

3. Neural network implementation: activation functions (AFs) and optimizers

Neural-network implementation is carried out on Google Colaboratory using Keras' [1] sequential application program interface (API) running on top of TensorFlow [2]. Here we treat this optical constant determination problem as a classification problem and as shown in figure 1, we use three layers to determine the best performing AF and optimizer combination for this unique problem as follows.
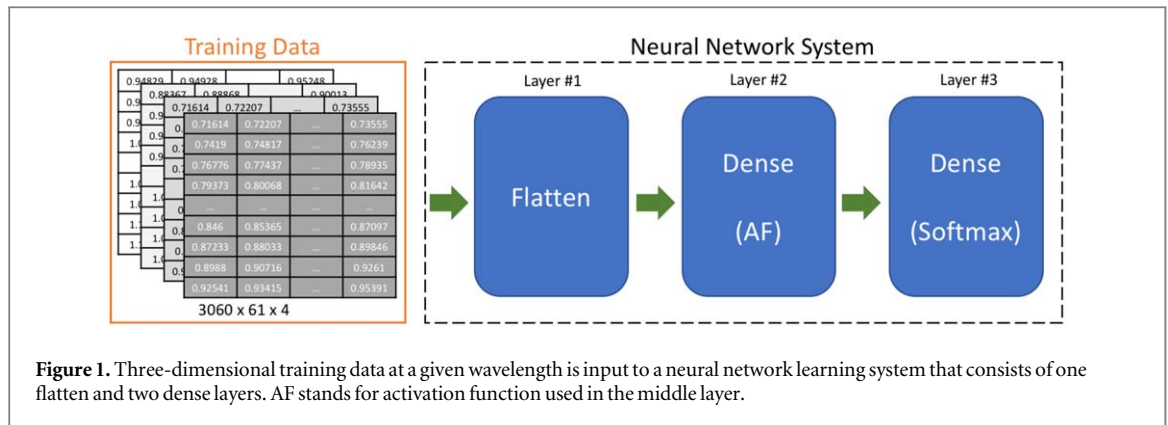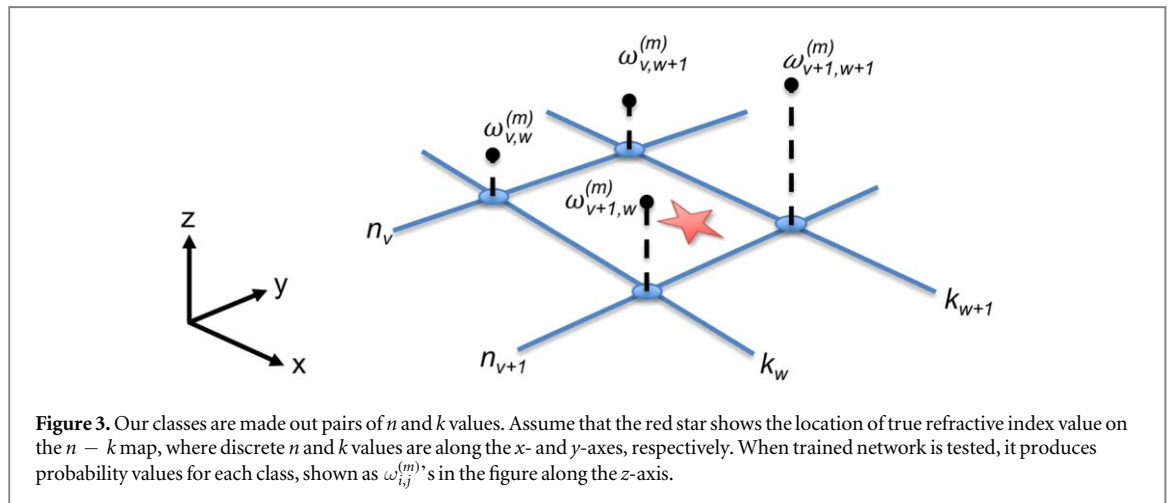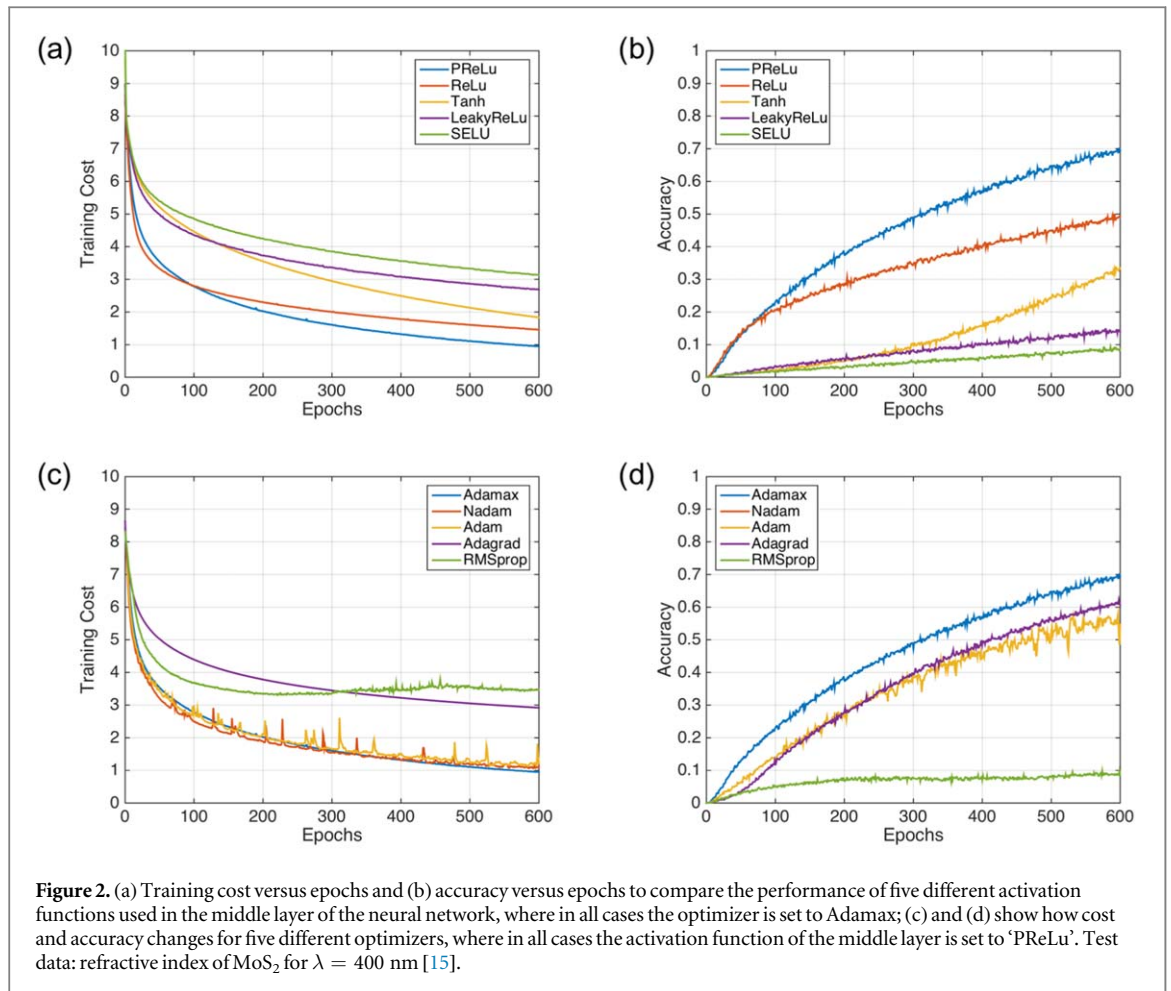
Figure 1. Three-dimensional training data at a given wavelength is input to a neural network learning system that consists of one flatten and two dense layers. AF stands for activation function used in the middle layer.

We start with differential reflectance data in a matrix form. We first use a 'flatten' layer to flatten the input. The second layer transforms the flattened input to achieve the learning by finding optimum parameters such as weights and bias. The selection of AF utilized in this layer is discussed in the following paragraph. The main role of the third and last layer is increasing the contrast among the possible labels (classes) determined by the previous layer. Since we are dealing with multiple classes, we use 'Softmax' function [23] for this last layer. The main advantage of using Softmax is that it produces an output probabilities range, where the sum of all the probabilities is equal to one. This property is used to increase the accuracy of the estimate as discussed in the next section.

To determine the best performing AF for the second layer of our sequential API implementation, we investigate how training cost and accuracy change as a function of epoch number for a broad range of AFs available in Keras API. Figures 2(a) and (b) show these trends for five different AFs, namely PReLu [24], ReLu [24], tanh, LeakyReLu [25], and SELU [26] on $MoS_2(\lambda = 400$ nm) dataset. 'Adamax' optimizer is used in all of these trainings. Among these five implementations, 'PReLu' AF yields the highest accuracy and lowest training cost which can be explained with following reasons. Since the absolute value of the differential reflectance ($|\Delta R|$) can become a significantly large number, when $|R_{wo}|$ is a very small compared to $|R_w|$, tanh and SELU activations functions experience the problem of vanishing gradients and do not perform well. Since the differential reflectance data has negative values and ReLu simply ignores them in the learning process by setting output to zero, ReLu's performance is also poor compared to PReLU. Leaky ReLU has a small slope for negative values instead of zero so it could have performed better than ReLu but apparently the used slope value, which is the default value of 0.01, is not the optimum one for our study. PReLu does not experience this problem because it allows the neural network to determine the slope. At the end of 600 epochs, the cost is less than 1 and accuracy is higher than 0.7 with the PReLu AF implementation. If the number of epochs is doubled, then the accuracy reaches to 0.9 and the training cost becomes smaller than 0.3.
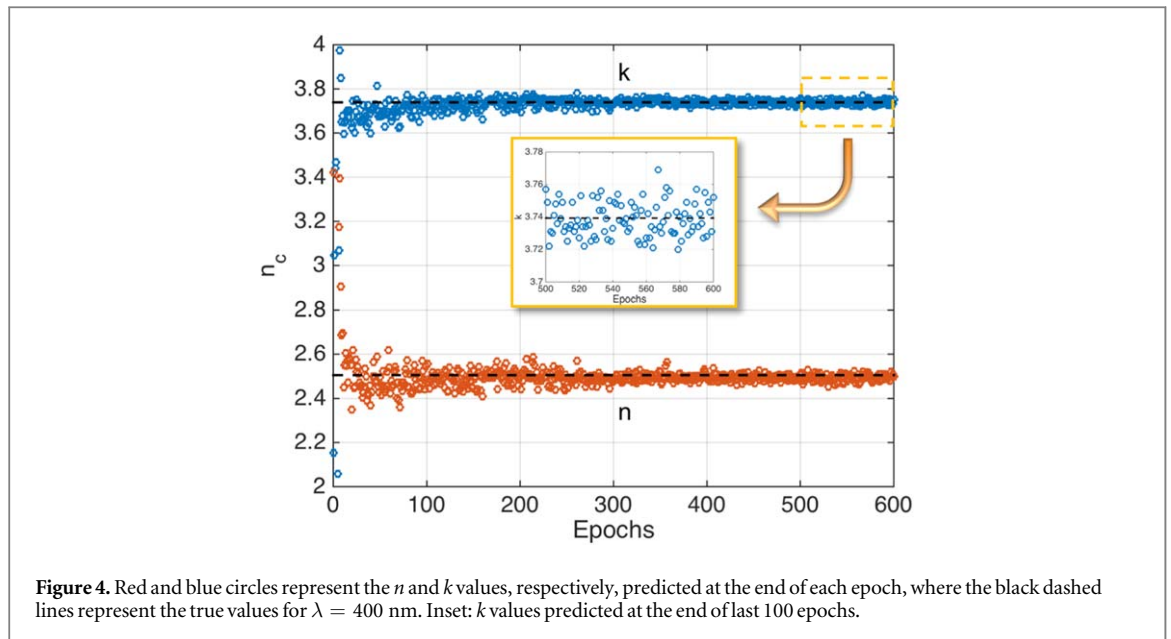
In terms of the optimizers, Keras provide several options, which are built upon slightly different versions of gradient descent method. Since the performance of these optimizers changes from one problem to another, we try several of them (Adamax [27], Nadam [28], Adam [27], Adagrad [29], and RMSprop [30]) and compare their performances in terms of training cost and accuracy. The results are plotted in figures 2(c) and (d). In terms of accuracy, RMSprop shows a very poor performance compared to the others. In our implementation, we use default parameters for learning rate ($\eta = 0.001$) and momentum term ($\rho = 0.9$). Different pairs of parameters can improve the accuracy of this optimizer but this is beyond the main focus of this work. When the default parameters are used, 'Adamax' performs slightly better than the other optimizers that compute adaptive learning rates for each parameter. It should be also noted that we run other sets of trainings using various AF—optimizer combinations on datasets generated at different wavelengths. At the end of this study, we conclude that PReLu and Adamax are the best performing AF and optimizer pair.

In problems as complex as this one, the learning requires several epochs. One might set the maximum number of these epochs to a certain number (e.g. 500) or choose a criterion to stop the training. For the former approach, if the chosen number is too small, the final set of parameters might be far from the optimum set; if it is too large, then vanishing or exploding gradients might lead to significant errors. The latter approach can be done in different ways: stopping the training when (i) loss function score becomes smaller than a certain value, (ii) defined metric, such as accuracy, reaches a certain threshold, or (iii) derivative of the loss function with respect to epoch number, which is basically the slope of the curves shown in figures 2(a) and (c), becomes smaller than a certain value. In this work, the third approach is followed: when the slope averaged over the last 20 epochs becomes less than 0.01, we stop the training.

Figure 2. (a) Training cost versus epochs and (b) accuracy versus epochs to compare the performance of five different activation functions used in the middle layer of the neural network, where in all cases the optimizer is set to Adamax; (c) and (d) show how cost and accuracy changes for five different optimizers, where in all cases the activation function of the middle layer is set to 'PReLu'. Test data: refractive index of MoS_2 for $\lambda = 400$ nm [15].

Figure 3. Our classes are made out pairs of n and k values. Assume that the red star shows the location of true refractive index value on the $n - k$ map, where discrete n and k values are along the x- and y-axes, respectively. When trained network is tested, it produces probability values for each class, shown as $\omega_{i,j}^{(m)}$'s in the figure along the z-axis.

4. Numerical results: number of epochs and improving accuracy

As previously mentioned, the output of the Softmax layer is a set of probabilities for each class to be the closest one to the ground truth for a given input test dataset. For a well-trained neural-network, one can expect to have high probability values for the classes, which are close to the true refractive index value, as depicted in figure 3. At the end of mth epoch, we utilize these probabilities, $\omega_{v,w}^{(m)}$'s, to have a more accurate prediction with the following formula
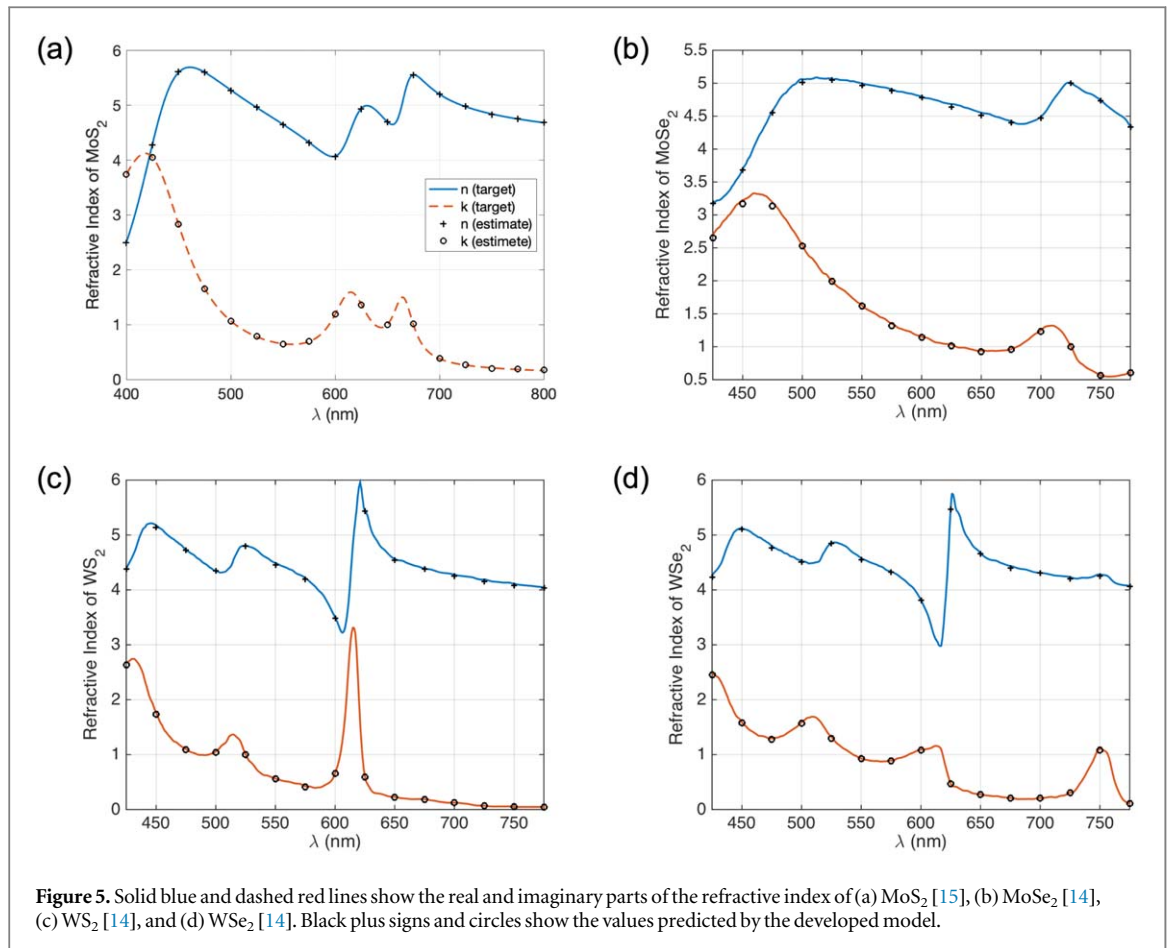
Figure 4. Red and blue circles represent the *n* and *k* values, respectively, predicted at the end of each epoch, where the black dashed lines represent the true values for $\lambda = 400$ nm. Inset: *k* values predicted at the end of last 100 epochs.

$$n_c^{(m)} = \frac{\sum_{v=1}^{V}\sum_{w=1}^{W}(n_v + \mathrm{i}k_w) \times \omega_{v,w}^{(m)}}{\sum_{v=1}^{V}\sum_{w=1}^{W}\omega_{v,w}^{(m)}}, \tag{1}$$

where $V = 60$ and $W = 51$ are number of discrete *n* and *k* values used in our training set, i.e. $n_v = v/10$ and $k_w = (w - 1)/10$. We observe that the difference between the results obtained using all the probabilities and using the highest *Q* probabilities, where $4 \leqslant Q \leqslant 100$, is less than 0.1%. So one might like to use fewer probabilities for a slightly faster computation. It should also be noted that when all probabilities are used, the denominator of the above formula is equal to 1 and does not require computation.

Figure 4 shows the predicted *n* and *k* values calculated with the above formula at the end of each epoch for $MoS_2(\lambda = 400$ nm) dataset. Interestingly, even though the training accuracy is estimated to be around 0.3 at the end of hundredth epoch, we can see that predicted values are close to the true values depicted with the dashed black lines in the same figure. Moreover, we observe the values predicted at the end of following epochs are distributed almost evenly around the true values, see the inset of figure 4. Based on this observation, we use another useful feature of Keras: saving neural network model and weights at the end of each epoch during the training session. Then we calculate the predicted refractive index values using weights of the last 20 epochs and we take the average of these values as our final prediction. Note that we stop our training when the slope averaged over the last 20 epochs becomes less than 0.01. Since at this level, the system is at a mature learning stage, averaging the last 20 predictions should not raise any concern over the accuracy. Instead, it is expected to provide a natural balance among over- and under-estimates.

In order to verify this expectation, we use this method to predict the refractive index of MoS_2 at 17 different wavelengths in the visible part of the electromagnetic spectrum. Figure 5(a) shows both predicted and expected (true) values. Indeed, the proposed neural network is capable of predicting refractive index of MoS_2 with a maximum error of 0.3%. A plain neural network implementation's maximum error is determined to be around 1.5%. Use of probabilities as weights via equation (1) reduces this error to 0.6% and the averaging step further reduces it to 0.3%. We test the accuracy of the developed method over other 2D materials: $MoSe_2$, WS_2, and WSe_2 [14]. As shown in figures 5(b), (c) and (d), there is again a very good agreement between the exact and predicted values and the maximum error is less than 0.3 % in all three cases.

A training period for one wavelength takes $\sim$ 40–50 min on Google Colaboratory. Initially all the model coefficients are saved. Once the training is completed, only the last 20 models' coefficients are hold and others are deleted to save memory. Testing for one wavelength takes less than 1 s. Considering the fact that the training needs to be done only once, the overall method can be considered as a very fast approach to predict the refractive index of any kind of 2D material under any circumstances, i.e. changing temperature, doping, applied voltage, etc. When we consider that the reflectance data changes 3% on average when *n* or *k* changes by 0.1, maximum error rate of 0.3% shows that neural networks not only bring the speed but also the accuracy compared to linear interpolation based approximations in which the maximum error rate is expected to be around 1.5%. Using neural-network estimated probabilities as weighting coefficients and averaging over final 20 predictions to balance over- and under-estimates bring this higher accuracy of the discussed neural-network based learning system.

Figure 5. Solid blue and dashed red lines show the real and imaginary parts of the refractive index of (a) MoS_2 [15], (b) $MoSe_2$ [14], (c) WS_2 [14], and (d) WSe_2 [14]. Black plus signs and circles show the values predicted by the developed model.

Even though it is not implemented in this work, it is also possible to use the model parameters successfully predicting at one wavelength (let us say $\lambda = 400$ nm) as initial guess of another model that is going to be used at a nearby wavelength (such as $\lambda = 401$ nm). Since refractive index is not a rapidly changing function of wavelength, the training fed by a model developed for a nearby wavelength is expected to require much smaller number of epochs to learn compared to that feeding model. One might benefit from this approach if the refractive index is needed as a very dense function of wavelength.

The developed neural-network based refractive estimator can easily be extended to 2D materials with multiple layers. In that case, the learning system can also predict the number of layers of those 2D materials, similar to [3–6].

When the number of layers existing in the medium where the light propagates increases, both reflectance and transmittance spectra are likely to have more features. This is why one might need to increase the number of layers implemented in the neural network to learn those features for such backgrounds. A final note is about the oscillatory nature of reflectometry measurements. Generally speaking, both reflectance and differential reflectance spectra are highly oscillatory, especially at room temperature experiments. For the sake of simplicity, such spectra are typically smoothed by fitting the data with some special functions, such as Gaussians or splines. However, one interesting fact we learn from the deep learning community is that such oscillations can lead to improvements in network generalization [31–33]. Hence the use of raw reflectance data, not the smoothed one, might be useful to reduce overfitting, which is encountered frequently in training neural networks with small datasets.

5. Conclusions

In this paper it is demonstrated that a neural-network based learning system trained on discrete data can predict the refractive index of atomically thin layered materials. The accuracy of the system is improved by (i) using the probabilities of each input element belonging to a label as weighting coefficients in a simple analytical formula and (ii) by averaging a finite set of predictions which are observed to be almost evenly distributed over- and under-estimates. Since the prediction does not require additional reflectance-transmittance calculation, the overall method is efficient from a computational perspective.

Acknowledgments

ORCID iDs

Ergun Simsek ⓘ https://orcid.org/0000-0001-9075-7071

References

[1] Chollet F *et al* 2015 Keras (https://github.com/keras-team/keras)
[2] Abadi M *et al* 2015 TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (https://tensorflow.org)
[3] Lin X, Si X, Fu W, Yang J, Guo S, Cao Y, Zhang J, Wang X, Liu P, Jiang K and Zhao W 2018 *Nano Res.* **11** 6316–24
[4] Li Y, Kong Y, Peng J, Yu C, Li Z, Li P, Liu Y, Gao C-F and Wu R 2019 *J. Materiomics* **5** 413–21
[5] Nolen C M, Denina G, Teweldebrhan D, Bhanu B and Balandin A A 2011 *ACS Nano* **5** 914–22
[6] Masubuchi S, Morimoto M, Morikawa S, Onodera M, Asakawa Y, Watanabe K, Taniguchi T and Machida T 2018 *Nat Commun.* **9** 1413
[7] Vladimir V B and Valery A L 2009 *Appl. Opt.* **48** 6178–87
[8] Castellano-Hernandez E and Sacha G M 2012 *12th Proc IEEE-NANO*
[9] Tabet M F and McGahan W A 2000 *Thin Solid Films* **370** 122–7
[10] Alexandridis A, Chondrodima E, Moutzouris K and Dimos T 2012 *J. Mater. Sci.* **47** 883–91
[11] Farrell T J, Wilson B C and Patterson M S 1992 *Phys. Med. Biol.* **37** 2281–6
[12] Budiman M and McBratney A B 2008 *Chemometr. Intell. Lab.* **94** 72–9
[13] Novoselov K S, Geim A K, Morozov S V, Jiang D, Zhang Y, Dubonos S V, Grigorieva I V and Firsov A A 2004 *Science* **306** 666
[14] Li Y, Chernikov A, Zhang X, Rigosi A, Hill H M, van der Zande Arend M, Chenet D A, Shih E-M, Hone J and Heinz T F 2014 *Phys. Rev. B* **90** 205422
[15] Mukherjee B, Tseng F, Gunlycke D, Amara K K, Eda G and Simsek E 2015 *Opt. Mater. Express* **5** 447–55
[16] Mukherjee B, Kaushik B, Tripathi R P B, Joseph A M, Mohapatra P K, Dhar S, Pavan Kumar G V, Lodha S and Simsek E 2017 *Sci. Rep.-UK* **7** 41175
[17] Mukherjee B, Leong W S, Li Y, Gong H, Sun L, Shen Z X, Simsek E and Thong J T L 2015 *Mater. Res. Express* **2** 065009
[18] Simsek E 2013 *Opt. Lett.* **38** 1437–9
[19] Weiss S and Indurkhya N 1995 *J. Artif. Intell. Res.* **3** 383–403
[20] Simsek E 2019 Reflectance: Differential Reflectance from SiO$_2$/Si and Glass Substrates (https://github.com/simsekergun/reflectance)
[21] Chew W C 1999 *Waves and Fields in Inhomogeneous Media* (Piscataway, NJ: Wiley-IEEE Press) pp 45–56
[22] Palik E D 1991 *Handbook of Optical Constants of Solids II* (Boston, MA: Academic)
[23] Kipf T N and Welling M 2016 ICLR
[24] Klambauer G, Unterthiner T, Mayr A and Hochreiter S 2017 arXiv:1706.02515
[25] Maas A L, Hannun A Y and Ng A Y 2013 *ICML (2013)*
[26] Jin X, Xu C, Feng J, Wei Y, Xiong J and Yan S 2015 arXiv:1512.07030
[27] Kingma D and Ba J 2014 arXiv:1412.6980
[28] Dozat T 2016 *ICLR Workshop* **1** 2013–6
[29] Duchi J, Hazan E and Singer Y 2011 *J. Mach. Learn. Res.* **12** 2121–59
[30] Tieleman T and Hinton G 2012 COURSERA: neural networks for machine learning *Technical Report* University of Toronto
[31] Reed R and ll Marks R J 1999 *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks* (Cambridge, MA: The MIT Press) p 273
[32] Bishop C M 1995 *Neural Comput.* **7** 108–16
[33] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: The MIT Press) p 241