Neural network analysis of neutron and x-ray reflectivity data: pathological cases, performance and perspectives

Alessandro Greco1 ©, Vladimir Starostin1 ©, Alexander Hinderhofer1 ©, Alexander Gerlach1 ©, Maximilian W A Skoda1 ©, Stefan Kowarik1 © and Frank Schreiber1 ©

1 Institute of Applied Physics, University of Tübingen, 72076 Tübingen, Germany
2 ISIS Pulsed Neutron and Muon Source, Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Science and Innovation Campus, Didcot OX11 0QX, United Kingdom
3 Department of Physical Chemistry, University of Graz, 8010 Graz, Austria

* Authors to whom any correspondence should be addressed.

E-mail: alexander.hinderhofer@uni-tuebingen.de and frank.schreiber@uni-tuebingen.de

Keywords: thin films, reflectivity, data analysis, neural networks

Supplementary material for this article is available online

Abstract

Neutron and x-ray reflectometry (NR and XRR) are powerful techniques to investigate the structural, morphological and even magnetic properties of solid and liquid thin films. While neutrons and x-rays behave similarly in many ways and can be described by the same general theory, they fundamentally differ in certain specific aspects. These aspects can be exploited to investigate different properties of a system, depending on which particular questions need to be answered. Having demonstrated the general applicability of neural networks to analyze XRR and NR data before (Greco et al 2019 J. Appl. Cryst. 52 1342), this study discusses challenges arising from certain pathological cases as well as performance issues and perspectives. These cases include a low signal-to-noise ratio, a high background signal (e.g. from incoherent scattering), as well as a potential lack of a total reflection edge (TRE). By dynamically modifying the training data after every mini batch, a fully-connected neural network was trained to determine thin film parameters from reflectivity curves. We show that noise and background intensity pose no significant problem as long as they do not affect the TRE. However, for curves without strong features the prediction accuracy is diminished. Furthermore, we compare the prediction accuracy for different scattering length density combinations. The results are demonstrated using simulated data of a single-layer system while also discussing challenges for multi-component systems.

1. Introduction

To investigate the structural, morphological or magnetic properties of surfaces and layered structures, such as solid and liquid thin films [1–8], x-ray and neutron reflectometry (XRR and NR) are often employed due to an array of benefits. Reflectometry measurements are an excellent non-invasive method for gaining access to the layer thickness, interface roughness and scattering length density (SLD) of a large variety of thin films [9]. Although neutrons and x-rays behave similarly in many ways, they also show some key differences regarding their elementary scattering process [10]. While x-rays interact with electrons, neutrons mainly interact with the nuclei (except for magnetic effects, which we neglect here), which allows them to be employed as probes for different types of samples, and thus answer different questions in a complementary manner [11]. Importantly, these differences are also reflected in the data the two methods produce and they must be taken into account during data analysis in order to extract the correct information from a given measurement [12].

The measured scattering signal is generally based on a Fourier transform of the probed structure in combination with Fresnel reflection coefficients; however, due to multiple scattering and the loss of the scattering phase in the detection process, it is not trivial to reconstruct the original real space structure, since a direct inverse transformation of the data is not possible. For reflectometry, a common way to extract
information from the measured data is to use a recursive mathematical model [13–16] to simulate a reflected intensity profile \( R(q) \) for different scattering vectors \( q \) which is in agreement with the measurement. This is usually done via iterative least means square fitting algorithms which have been implemented in various free software packages [17–21]. Increasingly sophisticated ways of optimizing the search for a local minimum are continuously developed to make the fitting process as fast and reliable as possible. However, depending on the quality of the data and the complexity of the studied system, finding a suitable model still often requires prior knowledge, a considerable amount of expertise, and is generally time-consuming.

A promising alternative could be machine learning (ML) techniques, which recently have been demonstrated for various scientific questions on related scattering techniques, such as small-angle scattering [22–24] and crystal structure or symmetry determination [25–27]. As shown in a recent study, mostly focused on real-time XRR [28], fully-connected neural networks can be trained to determine thickness, roughness and SLD parameters directly from the measured reflectivity data with very high speed and good accuracy within a comparatively large parameter range, thereby reducing the need for user input.

In this paper, we discuss the analysis of reflectivity data using neural networks in the light of three types of challenges: (1) Reflectivity curves without strong features that have a low information content, (2) curves without a total reflection edge (TRE), and (3) data with significant noise or background. We demonstrate that by applying different types of random noise and background intensity to the training data during the training process, the resulting neural network model is robust toward most types of perturbations when determining thin film properties, but struggles with certain particularly difficult edge cases. We test this approach on simulated reflectivity data of a single layer plus substrate within the same parameter ranges.

2. Methods

2.1. Reflectivity data simulation

The training and validation data were simulated using a model of a single layer on a substrate in air as an ambient medium. The model had five open parameters: substrate roughness, substrate SLD, layer thickness, layer roughness and layer SLD. We generated \( 3 \times 10^6 \) parameter sets for training and additionally \( 2 \times 10^6 \) sets for validation. The values of each set were generated within the ranges given in table 1 with a higher sampling density toward the limits of each range. This was done to make the local density of sampled values near the limits more similar to that toward the center of the distribution. The number of generated parameter sets was chosen as a compromise to cover as much of the large parameter space as possible while still maintaining technical feasibility in terms of training time and occupied memory. The range of possible SLD values for the substrate and layer was specifically designed to encompass a large spectrum of negative and positive SLDs of the most common elements. This allowed us to investigate the effects of different combinations of negative and positive SLDs on the prediction performance of the neural network. Furthermore, SLD combinations with contrasts between the layer and the substrate and the layer and the ambient SLD of less than \( 1 \times 10^{-6} \text{ Å}^{-2} \) were excluded from the training and testing data. These are known to produce curves without strong features and excluding them boosted the performance even on more feature-rich data. Also, for reasons of practicality due to the chosen \( q \) range, we focus on film thicknesses larger than 20 Å. A brief performance comparison between models trained with and without those exclusions is shown in figure S1 of the supporting information (available online at stacks.iop.org/MLST/2/045003/mmedia).

From the generated parameter sets, reflectivity curves were simulated using the implementation of the Matrix method [15] in the refl1D package [17]. The simulated \( q \) range was restricted to a range of 0.01–0.3 Å\(^{-1}\) in order to avoid \( q \) ranges where Bragg reflections and Lauve oscillations might appear in real measurements since they are not described by our slab model. Thus, we can be sure that the neural network predictions are only based on Kiessig fringes and other features related to the layer structure which would be present in an experimentally measured curve. Within this \( q \) range, the reflected intensity values \( R_q \) were simulated at 100 equally-spaced discrete points \( q \). This number was chosen to be comparable with common point densities of experiments. Of course experimental data would also be subject to a finite \( q \) resolution, however, to limit the number of noise sources under study, we have chosen to approximate this with uniform noise as described in section 2.1.2.
During training, different types of noise and background intensity were added to each curve every time a mini-batch was drawn from the training set. This means that every time the neural network encounters one of the training curves, the curve is modified with different noise and background. This step is crucial to avoid overfitting and to prevent the network from learning heuristics that do no work on imperfect data by forcing it to learn how to denoise the input data before interpreting it. The perturbations added to the data were Poisson noise, uniform noise, curve scaling and a constant background. Each type of curve modification is described in detail in the following.

2.1.1. Poisson noise due to counting statistics
Statistical noise in scattering data results from the counting statistics of scattered particles arriving at the detector and is dependent on the expected counting rate \( N_i \), i.e. the recorded intensity. Since this noise generally follows a Poisson distribution, the noise of a simulated reflectivity curve can be calculated by replacing each intensity value \( R_i \) with a random value picked from the distribution

\[
f_i(x) = \frac{f(x,sR_i)}{s}
\]

(1)

where \( s \) is the theoretical maximum number of counts at total reflection (for a monochromatic experiment) and \( f \) is the Poisson distribution

\[
f(x; N) = \frac{N^x e^{-N}}{x!}.
\]

(2)

Since the simulated intensities \( R \) only range from 0 to 1, they must be scaled to values which could occur in an experiment \( N_i = sR_i \) before calculating the noise. In this study, for every curve a scaling factor (corresponding to the flux) was randomly chosen on a logarithmic scale within \( s = [10^6, 10^9] \) to represent different experimental conditions. For the upper limit of \( s = 10^9 \), there is no noticeable noise in the chosen range anymore. An example of a curve with a noise level of \( s = 10^6 \) is shown in figure 1(a).

2.1.2. Uniform noise
Since the statistical noise mainly affects low-intensity regions in a reflectivity curve, the first half of the curve, i.e. low-\( q \) features and in particular the TRE, remain mostly unaffected by it. Despite that, experimental data may contain noise or other small deviations in this region. For example, in time-of-flight (TOF) neutron scattering experiments, the intensity given by the normalized reflectivity curve is not necessarily proportional to the counts on the detector, since the incoming beam generally has an energy spectrum with a non-uniform distribution (e.g. Maxwell–Boltzmann). The final intensity of the curve is then obtained by normalizing the number of counts in each channel (i.e. \( q \) value) with the corresponding incoming intensity of that energy. This can effectively lead to worse counting statistics in regions with seemingly higher intensity, such as near the TRE, compared to lower intensity regions. Furthermore, the beam shape and random errors on the measurement angle typically lead to non-negligible deviations of the intensity. This is particularly pronounced near the TRE where slight errors on the angle might translate to large errors in intensity.

To make the neural network robust against these types of errors, a random, uniformly distributed scaling factor \( \alpha \), is multiplied to each intensity value \( R_i \) of each input curve \( R \), so that the new intensity is given by

\[
R^*_i = \alpha_i R_i
\]

(3)

where \( \alpha = [0.7, 1.3] \). An example of a curve with uniform noise applied to it is shown in figure 1(b).

2.1.3. Curve scaling
In order to analyze reflectivity data, the measured intensity is typically normalized to the intensity at total reflection. For monochromatic experiments, this step is preceded by an angular dependent footprint correction. For polychromatic experiments (e.g. TOF), the normalization must in addition take into account the above mentioned energy spectrum, usually obtained via measuring the direct beam. Both of these corrections produce an error on the normalization procedure (which itself has a finite accuracy) and may result in distortions of the data. This effect is further exacerbated if there is no TRE, since the intensity at total reflection is not directly available, i.e. the naturally given absolute scale of the TRE is missing.

To make the neural network robust against these slight distortions, during training, every input curve \( R \) is multiplied by a random, uniformly distributed scaling factor \( \beta \), so that the new curve \( R^* \) is given by

\[
R^* = \beta R
\]

(4)

where \( \beta = [0.9, 1.1] \). An example of a curve with random scaling applied to it is shown in figure 1(b).
2.1.4. Residual background

Both x-ray and neutron scattering experiments contain background intensity stemming from various sources, such as background radiation or detector noise. Within the $q$ range discussed in this study (max. 0.3 Å$^{-1}$), for XRR these effects are usually negligible compared to the measured intensity of the reflected beam.

In addition, neutron reflectivity data usually contains background resulting from incoherent scattering [10]. In practice, most of this background is already removed during the data reduction step, e.g. via calibration with a pure transmission measurement. During data analysis, the residual background is then routinely approximated by a constant value, although more complex models exist [12].

To account for this, the residual background in the data was approximated by a $q$-independent constant $b$ with normally distributed fluctuations with a standard deviation of $\sigma_b = 0.1b$. The fluctuations were added so that the original curve cannot be fully reconstructed by just subtracting a constant value. Thus, for a given curve, the background added to each intensity value was randomly picked from the normal distribution

$$p_b(x; b, \sigma_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x-b)^2}{2\sigma_b^2}\right).$$

(5)

In this work, the background level of each curve was randomly chosen on a logarithmic scale within $b = [10^{-7}, 10^{-4}]$. An example of an added background with $b = 10^{-5}$ shown in figure 1(a).
2.2. Neural network design and training

The neural network used in this study was a fully-connected model with 100 input neurons, three hidden layers with 1000 neurons each and five output neurons as shown in figure 2. It was written in Python 3.7 with the help of the TensorFlow (2.1) framework [29]. The input corresponds to the reflected intensity values \( R \in \mathbb{R}^{100} \) at 100 discrete points in \( q \) space as described in the previous section. The output corresponds to the five open thin film parameters \( y \in \mathbb{R}^5 \) as shown in table 1. As an activation function, a simple ReLU (rectified linear) unit was chosen for all layers. During training, whenever a mini batch of 512 curves is drawn from the training set, curve modifications are applied as described in section 2.1. Then, each input \( \tilde{R}_i \) is independently standardized by subtracting the mean \( \bar{R}_i \) and dividing by the standard deviation \( \tilde{R}_i \) of all values of that input in the entire, randomly modified training set. The standardized input is thus given by

\[
\tilde{R}_i = \frac{R_i - \bar{R}_i}{\tilde{R}_i}
\]

with

\[
\bar{R}_i = \frac{1}{N} \sum_{n=1}^{N} R_{n,i} \quad \text{and} \quad \tilde{R}_i = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (R_{n,i} - \bar{R}_i)^2}
\]

where \( R_{n,i} \) is a curve from the training set of size \( N = 3 \times 10^6 \).

The output values \( y \) were normalized by the greater absolute value of either the minimum or maximum of their respective ranges given in table 1. This effectively confined all output values to a range from \(-1\) to 1.

The ADAM algorithm [30] was used as an optimizer with the recommended default parameters and a starting learning rate of \( 10^{-3} \). Furthermore, the learning rate was reduced by half each time the validation loss stagnated for ten epochs in a row in order to avoid skipping over narrow minima in the loss function space. The mean squared error (MSE) of the normalized outputs was used as the loss function. The neural network was trained on a GeForce RTX 2080 Ti GPU and an Intel® Core™ i5-9600K CPU for 175 epochs with a training time of about 6.5 min per epoch, amounting to a total training time of about 19 h. During training the training and losses were monitored as shown in figure 3. Overall, the training and validation loss were almost identical, with the validation loss only being slightly higher after 70 epochs. The reason why this difference is so small is that the training data is randomly modified with noise and background during every
epoch. This means that the network sees ‘fresh’ curves every time and is thus forced to generalize more. Since epoch 147 showed the lowest validation loss, we chose the neural network model corresponding to that epoch for all further testing.

3. Results and discussion

3.1. Definition of the prediction accuracy

The performance of the trained neural network was tested using 10 000 simulated reflectivity curves that were generated within the same ranges as the training data, excluding cases with low SLD contrast. The prediction error as a function of the ground truth for each of the five thin film parameters is shown in figure 4. In order to better quantify the performance of the neural network predictions, we separated all predictions into two classes: those that are near the ground truth were classified as ‘correct’ whereas all others were classified as ‘incorrect’. For this, we defined a condition for each parameter under which we consider a prediction ‘correct’. For the thickness and roughnesses, all errors smaller than 10% of the ground truth or smaller than 3 Å were considered ‘correct’. The absolute condition was added to avoid divergence for small