A Deep Learning Approach to Fast Radiative Transfer

Patrick G. Stegmann[a,1], Benjamin Johnson[a], Isaac Moradi[c,d], Bryan Karpowicz[d,e,f], Will McCarty[d]

[a]*Joint Center for Satellite Data Assimilation, NOAA Center for Weather and Climate Prediction, 5830 University Research Ct, 20740 College Park, MD, USA*

[b]*Joint Center for Satellite Data Assimilation, 3150 TAMU, 77840 Boulder, CO, USA*

[c]*University of Maryland, College Park, MD, USA*

[d]*NASA Goddard Space Flight Center, Greenbelt, MD, USA*

[e]*Goddard Earth Sciences Technology and Research, Greenbelt, MD, USA*

[f]*Universities Space Research Association, Columbia, MD, USA*

Abstract

Due to the sheer volume of data, leveraging satellite instrument observations effectively in a data assimilation context for numerical weather prediction or for remote sensing requires a radiative transfer model as an observation operator that is both fast and accurate at the same time. Physics-based line-by-line radiative transfer (RT) models fulfil the requirement for accuracy, but are too slow and too costly in computational terms for operational applications. Therefore, fast methods were developed to be able to perform fast RT calculations using techniques such as spectral sampling or pre-computed look-up tables. The operational fast models currently calculate the absorption and scattering coefficients from the pre-computed regression coefficients and atmospheric state and cloud profiles. As a novel solution to this problem, this work investigates a deep learning approach to replace the regression coefficients in the fast RT models. A selection of hidden-layer neural network configurations is trained against atmospheric transmittance profile data computed by an accurate line-by-line model and their performance is evaluated and their advantages and disadvantages are discussed.

Keywords: machine learning; deep learning; radiative transfer; infrared radiation; transmittance

1. Introduction

1.1. *Background*

A unique aspect of atmospheric radiative transfer in the thermal infrared (IR) region of the electromagnetic spectrum is the presence of sharp absorption lines from atmospheric trace gases, primarily water vapor and carbon dioxide, but also methane and ozone among others. The radiances measured by the satellite instruments are averaged over a spectral band according to the sensor response function. Therefore, computing corresponding radiances using a RT model requires conducting high resolution RT calculations over the sensor response domain for each instrument channel. Line-by-line radiative transfer models [1] directly resolve each absorption line in a brute-force approach. However, the necessary computational resources for this approach, both in terms of computation time and memory, are prohibitive in the context of inverse problems such as operational satellite radiance data assimilation and remote sensing retrievals. Similar problems arise for the computation of broadband radiative fluxes in thermodynamic models [2].

[1]* Corresponding author.
E-mail address: stegmann@ucar.edu

For this reason, a wide range of methods was developed in order to mitigate this issue. These methods can broadly be classified into three distinct categories: (i) band models, (ii) spectral sampling strategies, and (iii) statistical regressions. However, band models [3] are not in common use anymore. The Correlated-k distribution method (CKD) [4] is based on sorting the absorption coefficient spectrum in ascending order before the numerical convolution is applied to it. While this approach is quite elegant from a mathematical perspective, its generalization to inhomogeneous atmospheric profiles and multiple absorber gases is not straightforward. Both CKD and Optimal Spectral Sampling (OSS) [5] are categorized as spectral sampling strategies for the purposes of this work. The statistical method is the most common technique used in the current fast models. Accurate line-by-line calculations provide the basis for a fast gas transmittance parameterization using methods from statistical learning, or more specifically from statistical regression. A non-exhaustive selection of such models is mentioned in the following. An early model that predicted water vapor and ozone absorption using a linear regression on constant levels of cumulative absorber amount is the so-called OPTRAN model [6]. This was later developed into the compactOPTRAN model [7] where a polynomial regression is used to drastically reduce the number of necessary regression coefficients and ensure smooth water vapor Jacobians. This in turn provided the basis for the ODAS (Optical Depth in Absorber Space) algorithm of the Community Radiative Transfer Model (CRTM). The CRTM is a widely used fast radiative transfer model for satellite data assimilation and allows the choice between two different transmittance algorithms, one being ODAS, while the other one is Optical Depth in Pressure Space (ODPS) [8]. As the name suggests, ODPS uses a linear regression on a grid of constant pressure levels to predict the layer optical depth. Similar methods have been applied in the RTTOV (Radiative Transfer for TOVS) fast radiative transfer model [9], the SARTA (Stand-alone AIRS Radiative Transfer Algorithm) model [10], and the TAMU fast radiative transfer model [11]. Another approach includes Principal-Components Radiative Transfer Model (PCRTM) [12,13], where a singular value decomposition is applied to the transmittance training data.

1.2. *Neural Networks for Radiative Transfer*

Despite the increasing applications of neural networks in many other fields such as image processing and natural language processing, the application of neural networks in atmospheric radiative transfer has been limited by comparison. Likely the first influential application of neural networks to longwave radiative transfer was by Chevalier in Refs. [14,15]. Therein, the NeuroFlux model for climate studies was described. Neural networks for sounder radiance prediction are described in Ref. [16], where a predictor training set based on synthetic atmospheric profiles is utilized. The same approach is employed in the recent Ref. [17], where ECMWF forecast profiles are used as predictors to approximate VIIRS M-band (**V**isible **I**nfrared **I**maging **R**adiometer **S**uite – **M**oderate resolution band) brightness temperatures. General applications of radial basis function networks are discussed in Refs. [18] and references therein. A feed forward neural network for efficient radiative transfer simulation is developed and discussed in Ref. [19].

Lastly, while dimensionality reduction techniques are strictly speaking not a direct component of neural networks, they are nevertheless important and often applied to pre-processing of input- and training data and for this reason the work of Efremenko in Ref. [20] is also mentioned here, where a selection of such methods is studied w.r.t. to their applicability in a radiative transfer context.

1.3. *Purpose of this study*

The purpose of this study is to evaluate deep learning neural networks as a means of fast and accurate transmittance parameterization in radiative transfer models for satellite data assimilation and remote sensing, such as the Community Radiative Transfer Model.

1.4. *Manuscript Overview*

In the following, the manuscript is divided into 4 remaining sections. Section 2 provides an introduction to the problem of radiative transfer in a non-scattering atmosphere, and the associated difficulties. The need for a fast transmittance model is highlighted. Section 3 briefly describes the deep learning neural network approach, and the different specific network topologies that were chosen for this study. The subsequent Section 4 discusses the preparation of the training data for both predictors and predictands. The penultimate Section 5 presents the results of the training process and the trained neural network in a comparative approach. Both advantages and disadvantages of the different network topologies are highlighted. Lastly, Section 6 concludes the study with a discussion of the findings in the context of fast radiative transfer models for data assimilation.

2. Theoretical Background: Radiative Transfer in a Non-Scattering Atmosphere

The solution to the one-dimensional, stationary, and monochromatic radiative transfer equation for the upwelling radiance at the top of a non-scattering atmosphere (TOA) is given by Eq. (1):

$$I_{\nu,TOA} = \epsilon_\nu \cdot B_S(\nu,\theta_S) \cdot T(\nu,z_S) + (1-\epsilon_\nu) \cdot T(\nu,z_S) \cdot \int_1^{T(\nu,z_S)} B\big(\nu,\theta(z)\big)\, dT(\nu,z) + \int_{T(\nu,z_S)}^{1} B\big(\nu,\theta(z)\big)\, dT(\nu,z) + r \cdot \mu_0 \cdot \frac{F_{Sol}}{\pi} \cdot T(\nu,z_S,\mu_0) \cdot T(\nu,z_S,\mu)\,, \quad (1)$$

where $I_{\nu,TOA}$ is the monochromatic radiance at wavenumber ν and TOA, z is the geometric height, θ is the thermodynamic temperature in Kelvin, and $B\big(\nu,\theta(z)\big)$ is the Planck function. The quantities ϵ_ν, r, μ_0, μ, and F_{Sol} are the monochromatic surface emissivity, surface reflectivity, secant of solar zenith angle, secant of satellite zenith angle, and the incident solar irradiance. The symbol $T(\nu,z)$ is the monochromatic transmittance. Surface emission, upward atmospheric emission, and downward atmospheric emission reflected by the surface are represented by the first three terms in Eq. (1). Direct solar reflection is represented by the last term. Eq. (1) can further be simplified by assuming a surface emissivity $\epsilon_\nu = 1$:

$$I_\nu = B_S\big(\nu,\theta(z_S)\big) \cdot T(\nu,z_S) + \int_{T(\nu,z_S)}^{1} B\big(\nu,\theta(z)\big)\, dT(\nu,z) \quad (2)$$

In order to compute the channel-effective radiance measured by a satellite instrument, the monochromatic radiance given by Eq. (2) needs to be convolved with the spectral response function (SRF) $\Phi(\nu)$ over the spectral bandwidth $\Delta\nu$ of the instrument channel:

$$I_{ch} = \frac{\int_{\Delta\nu} \Phi(\nu)\cdot B_S(\nu,\theta)\cdot T(\nu,z_S)\, d\nu + \int_{\Delta\nu} \Phi(\nu) \int_{T(\nu,z_S)}^{1} B\big(\nu,\theta(z)\big)\, dT(\nu,z)\, d\nu}{\int_{\Delta\nu} \Phi(\nu)\, d\nu} \quad (3)$$

As is common practice [21], the quadrature of the integral over the transmittance profile in Eq. (3) is carried out using either the layer-to-space transmittance profile or the level-to-space transmittance profile for an atmospheric column that is

discretized into N layers or $(N+1)$ levels, respectively.

$$I_{ch} \approx \frac{\int_{\Delta\nu} \Phi(\nu)\cdot B_S(\nu,\theta)\cdot T(\nu,z_S)\,d\nu + \sum_{i=1}^{N} \int_{\Delta\nu} \Phi(\phi)\cdot \underline{B}(\nu,\theta)\cdot[T(\nu,z_{i-1}) - T(\nu,z_i)]\,d\nu}{\int_{\Delta\nu} \Phi(\nu)\,d\nu}, \tag{4}$$

where $\underline{B}(\nu,\theta)$ is commonly referred to as the *effective* Planck function [22] of a layer between the levels $i-1$ and i. In the simplest approximation to Eq. (4), the variation of the Planck function over a layer is not taken into account and the polychromatic channel-effective transmittance $T_{\Delta\nu,i}$ can be approximated as follows:

$$T_{\Delta\nu,i} = \frac{\int \Phi(\nu)\cdot T_i(\nu)\,d\nu}{\int \Phi(\nu)\,d\nu}. \tag{5}$$

Eq. (5) is a convolution between the instrument channel SRF $\Phi(\nu)$ and the monochromatic transmittance $T_i(\nu)$ for layer i. It betrays the difficulty in calculating the channel transmittance $T_{\Delta\nu,i}$. While the instrument channel SRF is often numerically cheap to integrate, such as in the case of a boxcar SRF, the monochromatic transmittance $T_i(\nu)$ is often a very irregular function of the wavenumber ν, particularly in the thermal infrared part of the electromagnetic spectrum. Consequently, carrying out the integral in Eq. (5) as a numerical quadrature requires a very high resolution in order to accurately resolve $T_i(\nu)$, which makes this direct approach too expensive for operational data assimilation. However, since the approximation Eq. (4) to the solution of the radiative transfer equation in Eq. (3) is otherwise computationally cheap, it makes sense to retain the physics-based solution in Eq. (4) and only find a parameterization for the numerically expensive part in Eq. (5). This project uses deep learning neural networks for this purpose and the details of this approach are given in Section 3.

3. Description of the Regression Model

This section describes the statistical regression approach towards accelerating the computation of the layer-to-space transmittance profiles required for the solution of Eq. (4) in a radiative transfer computer program. From a high-level view, the process is quite straightforward and a sketch is outlined in Fig. 1.

Figure 1: Flowchart of the transmittance regression approach.

The first step of the process consists of generating a synthetic dataset for the transmittance predictors and predictands. This is achieved by selecting a representative dataset of atmospheric states and feeding them to an accurate line-by-line radiative transfer model to compute the monochromatic transmittance profiles at a high spectral resolution. Step 2 consists of the convolution of the monochromatic transmittance profiles with the instrument spectral response function (SRF) to arrive at the

channel-resolution transmittances. Lastly, a regression fit of the channel-resolution transmittance is performed to develop a fast model. In this work, a hidden-layer neural network is applied to the last step. Further details on the neural network structure are given in the remainder of this section.

3.1. *Deep Learning Neural Networks*

Section 3 provides an overview of the essential properties and for further details we refer to the review article [23]. A *Hidden-Layer Neural Network* is a nonlinear statistical regression model typically illustrated by a network diagram of a directed graph exemplified in Fig. 2.

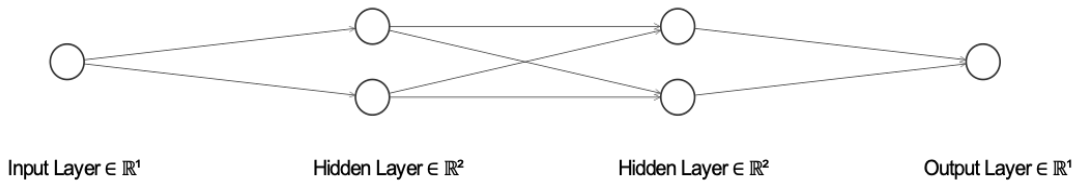


Figure 2: Illustration of a simple hidden-layer neural network with two hidden layers.

The network may contain two or more so-called *hidden* layers. Each node of the network is associated with an internal state z_i that is computed as a linear combination of the internal states y_j of the preceding network layer or the network input x_i respectively:

$$z_k = w_{jk} \cdot y_j + b_k, \tag{6a}$$

$$y_j = f(z_j), \tag{6b}$$

where w_{jk} is the matrix of the weighting coefficients of the network connections (“axons”), b_k are the layer bias coefficients and is the layer activation function $f(z_j)$. The activation function generally introduces a nonlinearity in the otherwise linear network. Possible choices for the activation function $f(z_j)$ include, but are not limited to, the *sigmoid function*, the *hyperbolic tangent* (abbreviated here as tanh) and the *rectified linear unit* (ReLU). Initially all types of activation functions were considered in this study, but the ReLU function could quickly be ruled out, as training results were consistently worse and the tanh activation function fared slightly better than the sigmoid. From a mathematical point of view this is easy to understand, as the channel-effective transmittance is a smooth function bounded by the range [0,1]. The sigmoid function is given here for the sake of completeness:

$$f(z) = \frac{1}{1+e^{-x}}. \tag{6c}$$

It remains to be emphasized that this outcome is specific to the current problem and network architecture. Ref. [19] for instance achieved better accuracy with a *LeakyReLU* activation function. Fitting the neural network model to the predictands of the training data set is achieved by minimizing the sum of squared errors quadratic functional as a measure:

$$J(\Omega) = \frac{1}{2}\sum_{i\in L_1} \quad \sum_{l\in L_{final}} \quad (T_{li} - y_l(x_i))^2 \,, \tag{6d}$$

where $J(\Omega)$ is the cost- (or alternatively loss-) function for a specific set of neural network parameters $\Omega \in \{w_{jk}, b_k\}$, T_{li} are the training data points, and y_l is the output of the neural network for the *l-th* output layer node. A variety of methods exist for the minimization of Eq. (6c), such as gradient descent [24], and the ADAM stochastic optimizer [25]. One thing these methods have in common is that they require not only the model y_l itself, but also the gradients $\left\{\frac{\partial J(\Omega)}{\partial w_{jk}}, \frac{\partial J(\Omega)}{\partial b_k}\right\}$ of the cost function. The necessary gradients of y_l w.r.t. the parameter space Ω are computed using *backpropagation* [26], which is a special case of reverse-mode algorithmic differentiation [27]. Consequently, backpropagation is the adjoint [28] of the hidden-layer neural network forward model w.r.t. the model parameters (but not the model input). This approach for computing the gradients is generally more efficient than the tangent-linear or forward-mode algorithmic differentiation as long as the number of model outputs $y_l(x_i)$ is smaller than the number of model parameters Ω.

3.2. *Neural Network Topologies for the Transmittance Regression*

Generally speaking, multilayer feedforward neural networks have been shown to be universal function approximators [29], but the hidden-layer neural network regression model described in the previous subsection 3.1 also allows for a great deal of flexibility in solving the problem at hand. Fig. 2 and Eqs. (6a-d) illustrate that the model consists of a number of basic, simple building blocks, namely the network nodes and edges, which can be combined in an almost arbitrary way, leading to a theoretically unlimited number of degrees of freedom for the model. It remains to be evaluated, which specific arrangement of the model leads to the best results in practice. The objective of this subsection is to describe the different model arrangements suitable to the transmittance regression problem at hand, and discuss their respective advantages and disadvantages. Fig. 3 provides a general sketch of the atmospheric gaseous transmittance regression process using a hidden-layer neural network. Irrespective of the exact details of the network, all proposed neural networks need to follow the general arrangement outlined in Fig. 3, with an input layer that accepts atmospheric property profiles relevant for gaseous absorption, such as temperature profile, pressure profile, water vapor concentration and other absorber concentration profiles, such as carbon dioxide, as well as possible derived quantities, such as cumulative absorber profiles. The network should include one or more hidden layers, in order to account for nonlinear relationships and provide an advantage over simple linear regression [23]. Lastly, the model output consists of a gaseous transmittance value. As the transmittance is a multivariate function, the number of input layer nodes of the network will always be a multiple of the number of output layer nodes.

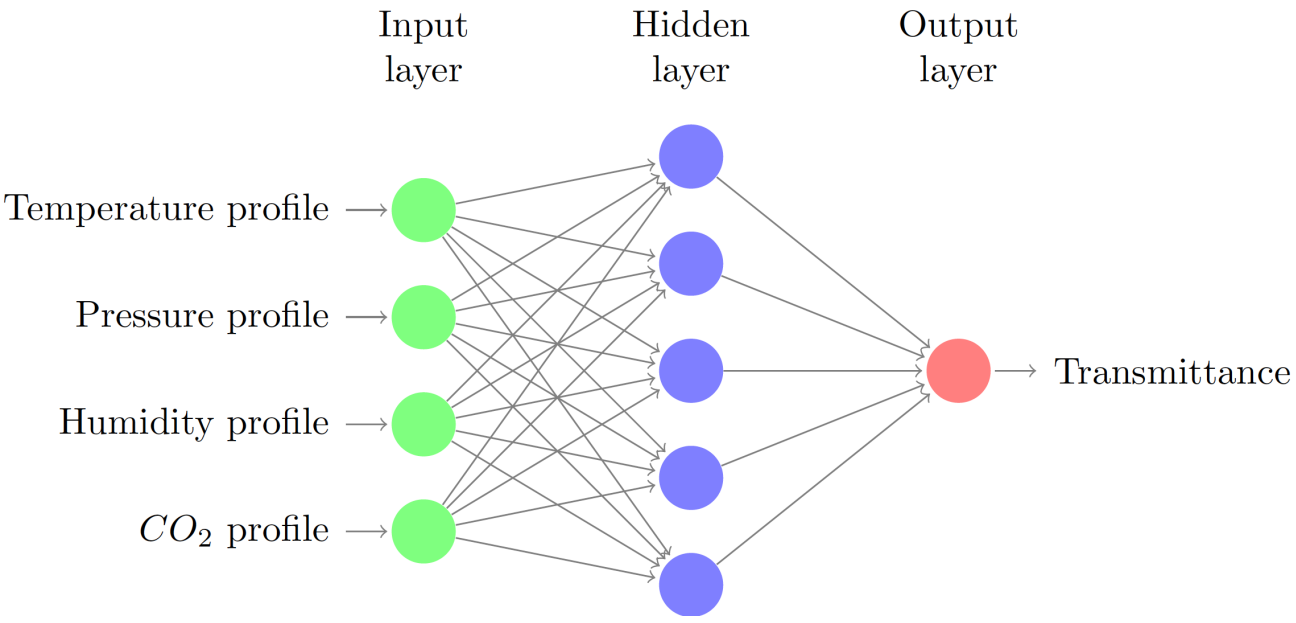


Figure 3: Sketch of the atmospheric transmittance regression using a hidden layer neural network.

As is common practice, both the atmospheric properties and the gaseous transmittance are defined on a discrete grid of either atmospheric pressure layers or levels [21], as described in Section 2. For the neural network model, quantities henceforth are defined as a function of pressure layers, unless indicated otherwise. Limiting the study to fully connected neural networks, two different model arrangements thus are conceivable in order to predict the layer-to-space gaseous transmittance from a given dataset of input atmospheric profiles:

1. A separate neural network for each pressure layer,
2. A global neural network for the entire atmospheric profile for all layers simultaneously,

henceforth called approach 1 and approach 2, respectively. Both approaches are permissible, as the gaseous layer-to-space transmittance of a given layer is completely independent from other layers, at least in the context of this problem. A direct illustration for an actual network used in approach 1 is given in Fig. 4. An illustration of the network for approach 2 is intractable due to the network size and thus omitted. From initial considerations, both advantages and disadvantages of approach 1 and approach 2 can be directly deduced. The advantages and disadvantages of approach 1 are given in Table 1 and the properties of approach 2 are given in Table 2.

Advantages	**Disadvantages**
Small individual network sizes	Long training times for all layers combined
Layers are completely independent	Larger size of all coefficients combined
Networks can be tuned on a layer-by-layer basis	Cannot directly account for non-local effects between different layers
Optimal for parallelization	

Table 1: Advantages and disadvantages of using a separate (small) neural network for each atmospheric layer (approach 1).

Overall the advantages and disadvantages of approach 1 and approach 2 are largely complementary. Approach 1 relies on a separation of individual layers. This allows to treat each layer separately and optimize the regression process and the neural network for each layer individually. This also allows to parallelize the transmittance computation in the final model in a straightforward manner, as the computations for each layer are completely detached from each other and thus embarrassingly parallel. This does not mean however, that it is not possible to parallelize the network of approach 2. Nevertheless, it becomes necessary to apply parallel linear algebra operation algorithms for the entire network, instead of simply parallelizing the individual layer networks. Furthermore, the size of the model coefficient arrays for each layer is significantly smaller than the corresponding size of the coefficient arrays for the model that treats all layers simultaneously. As a consequence, the coefficient arrays become easier to analyse, potentially leading to better insights on a per-layer basis. On the other hand, the training time for all separate layers (approach 1) combined will be longer than in the simultaneous case (approach 2). Conversely, the total size of the coefficient arrays for approach 1 is also larger than for the case of approach 2, leading to larger instrument-specific coefficient files and more computational time required for the coefficient I/O process while running the fast model. Lastly, this approach by construction does not permit to account for potential non-local effects in the transmittance computation, i.e. no constraints between layers can be considered without additional modifications of the scheme. In an extreme scenario, there would be no guarantee for instance that the layer-to-space transmittance is monotonously increasing while moving down from TOA to the surface. A more realistic conceivable issue is that in approach 1 there is no mechanism that ensures that the Jacobian of the channel-resolution transmittance profile w.r.t. the atmospheric properties is a smooth function.

Advantages	**Disadvantages**
Fast Training	Not straightforward to parallelize
Only one set of network parameters	Unable to analyse individual layers
Can fully account for non-local effects	Less accurate overall

Table 2: Advantages and disadvantages of using a single continuous neural network for the entire atmospheric profile (approach 2).

For approach 2 the overall training time is faster than for approach 1, under the assumption that the training in approach 1 is not done in parallel. In terms of coefficient storage, online I/O and scientific analysis there is only one coherent array of model coefficients that allow for smaller coefficient storage sizes, faster model I/O, and simpler analysis and visualization of the coefficients. In the same vein, approach 2 allows to directly account for constraints between atmospheric profile layers, as the layers are directly manifest as nodes of the model and the constraints or connections between different layers are reflected in the network connections between layers. Visualizing the connection weights as a matrix makes it possible to gain insight into how the model processes the atmospheric transmittance and if there are influences present between layers. Lastly, starting from considerations of overfitting, a third and last approach was derived from approach 2 in an effort to retain the advantages listed in Table 2, while simultaneously mitigating the disadvantages, in particular when it comes to the model accuracy. This is achieved by reducing the hidden layer size and number of trainable parameters of approach 2. While this approach may seem counterintuitive at first, reducing the complexity of the model can minimize the contribution of predictors that have little or no correlation with the predictands and reduce the influence of overfitting. More details on this issue are provided in subsection 5.4. The third approach is designated as *optimized approach 2*, and its advantages are given in Table 3.

Advantages	Disadvantages
Fast Training	Not straightforward to parallelize
Only one set of network parameters	Unable to analyse individual layers
Can fully account for non-local effects	Requires careful tuning
Improved accuracy	
Reduced model parameters and size	

Table 3: Advantages and disadvantages of using an optimized single continuous neural network for the entire atmospheric profile (optimized approach 2).

From Table 3 it can be seen that out of the proposed architectures, the optimized approach 2 is the most favourable for practical applications. A summary of the neural network model structure for all approaches is given in Table 4.

	Approach 1	Approach 2	Optimized Approach 2
Nodes	$8 \times 16 \times 16 \times 1$	$800 \times 1600 \times 1600 \times 100$	$800 \times 80 \times 100$
Activation Function	tanh	tanh	tanh
Input Nodes	8 per layer	800 in total	800 in total

Table 4: Neural network model parameters for approach 1 and approach 2.

Generally, the network structure for approach 2 is a simple scaling of the structure for approach 1 by the number of atmospheric profile layers. In all three cases the *tanh* activation was chosen for all layers. For both approach 1 and 2 it was found through trial and error that a satisfactory compromise between network size and achieved accuracy could be realized with two hidden layers and a number of nodes per hidden layer that is twice the number of input nodes.

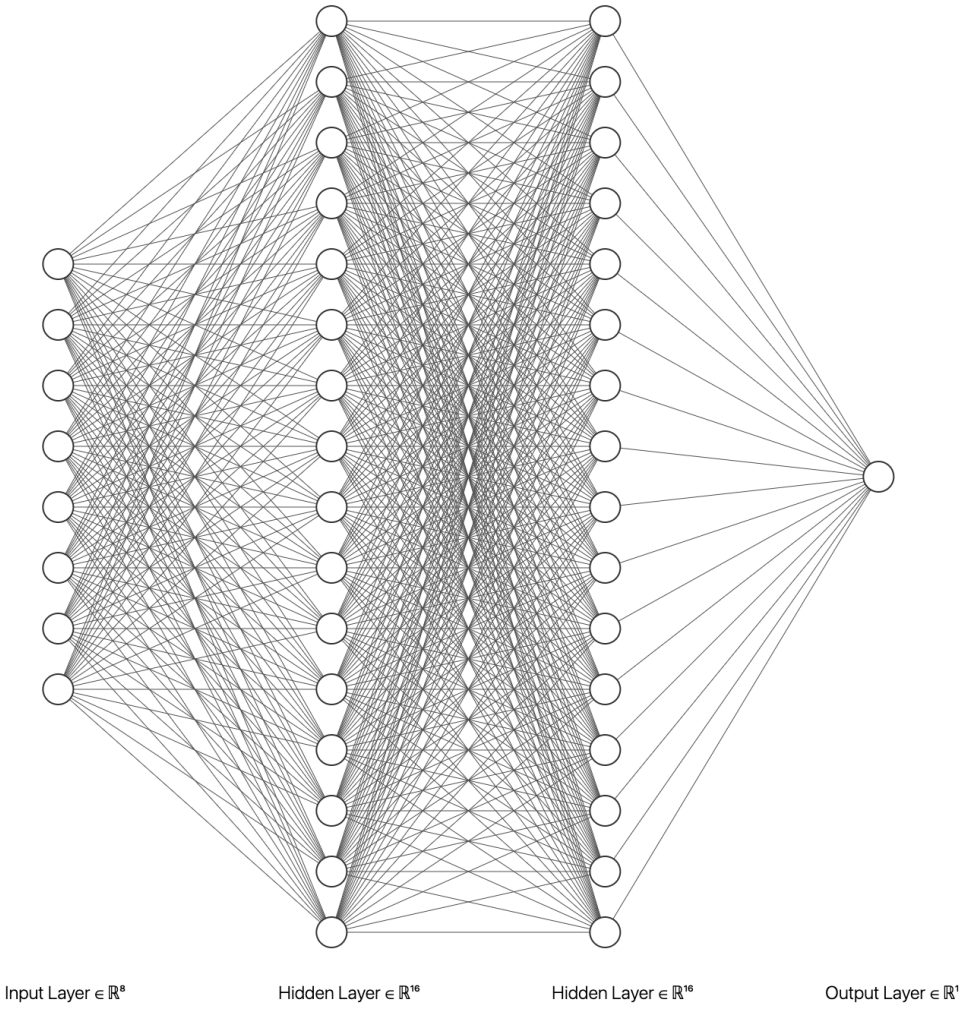


Figure 4: Feed-forward hidden-layer neural network for approach 1.

4. Preparation of the Training Data

This section discusses the preparation of the predictors and predictands necessary for the regression problem. The predictors consist of a representative set of atmospheric profiles with physical parameters that affect the atmospheric transmittance of the respective profile. The predictands are quite generally channel-resolution transmittance profiles computed from the predictor profiles using accurate line-by-line models. This approach is common practice for the development of fast radiative transfer models [8,9].

4.1. *Atmospheric Profiles as Predictors*

The predictors of the neural network model are based on a diverse, representative dataset of atmospheric profiles [30] that is also the basis for the CRTM [8] and RTTOV [9] transmittance models. Henceforth, the profile set will be referred to as the ECMWF83 profile set. It contains the following atmospheric physical properties:

- Pressure in Hecto-Pascal,
- Temperature in Kelvin,
- Water vapor absorber amount in g/kg,

- Carbon dioxide absorber amount in ppmv,
- Ozone absorber amount in ppmv.

The listed quantities are defined on a vertical discretization of both layers and levels which follow the AIRS pressure level definition [10]. As suggested by the name, the ECMWF83 profile set contains 83 atmospheric profiles that cover a representative range of variation of the atmospheric parameters listed above. The range of the profiles, together with mean and median are displayed in Fig. 5.

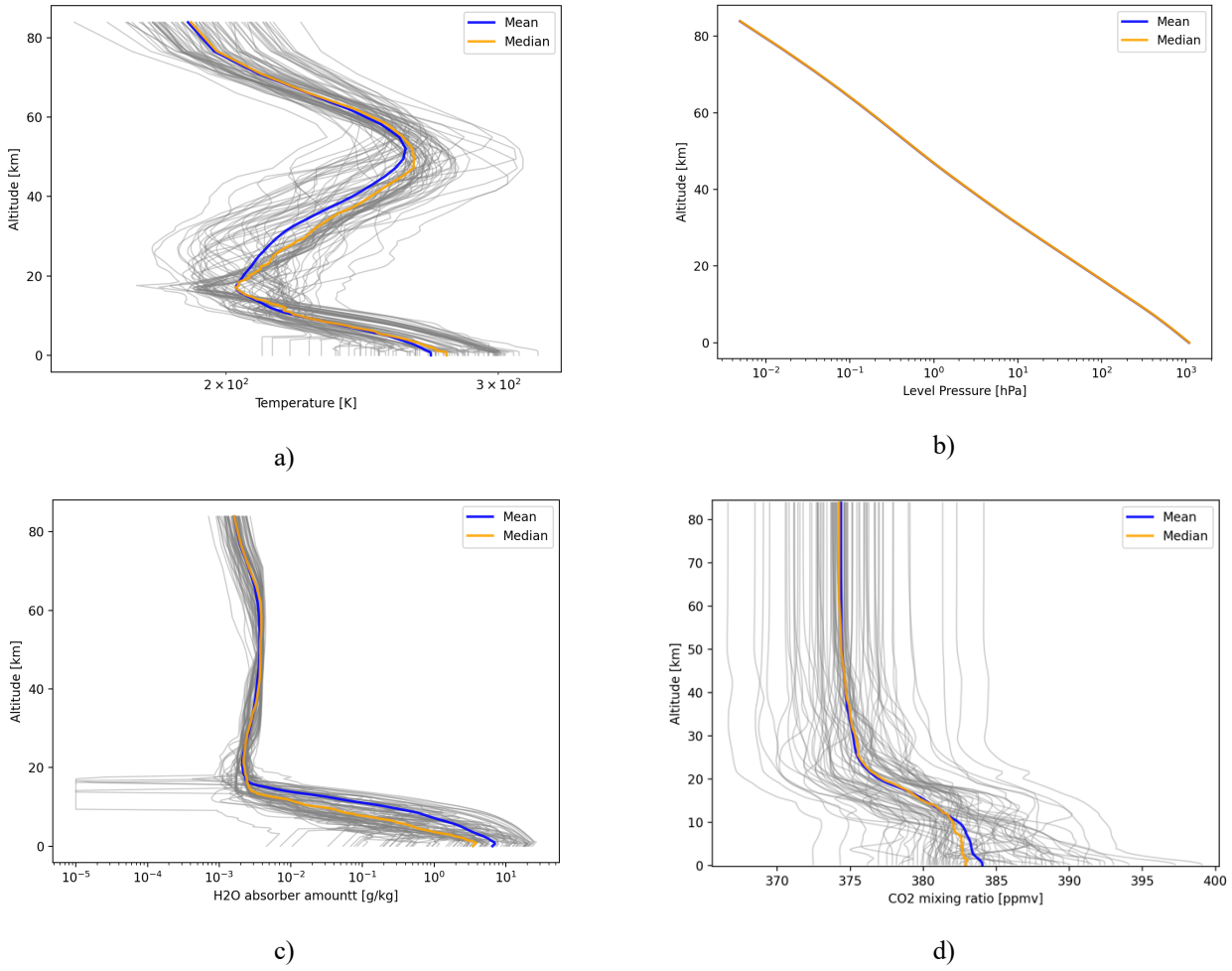


Figure 5: ECMWF83 atmospheric training data profile set.

There is no variation in the pressure variable which serves as the vertical coordinate. In this study the values of the carbon dioxide with a surface median value of 382.5 ppmv are taken from the original ECMWF83 profile set and not adjusted to 2021 concentration levels. An investigation of the impact of increasing carbon dioxide concentration values on the terrestrial atmosphere are beyond the scope of this study. As an independent tesing profile data set, the UMBC48 with 48 profiles is selected. The testing dataset is shown in Fig. 6. Note that as a significant caveat the testing dataset does not contain any carbon dioxide variation.

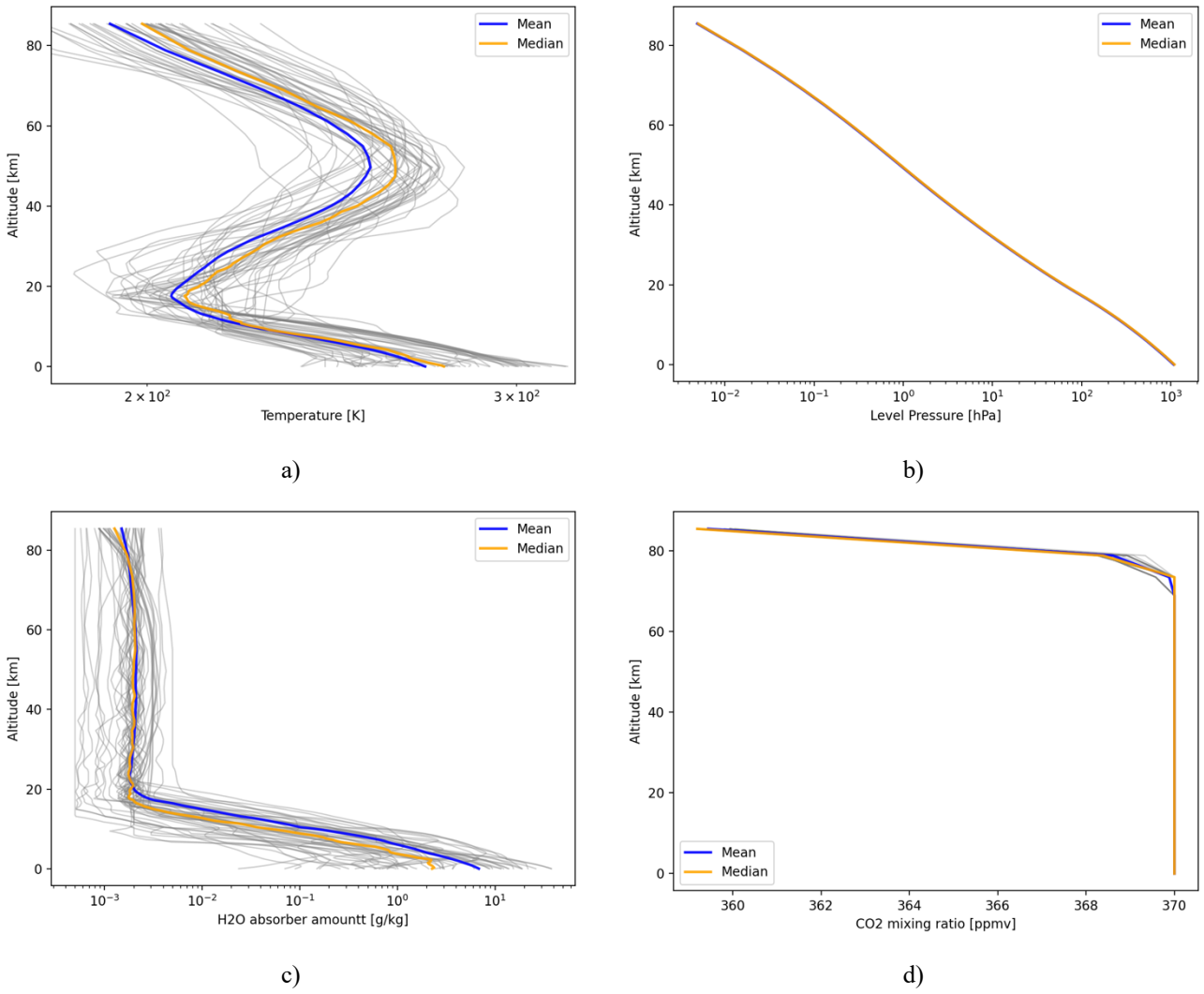


Figure 6: UMBC48 atmospheric training data profile set.

The ECMWF83 profiles shown in Fig. 5 are used directly as input for the line-by-line calculations to compute the atmospheric transmittance profiles, but a normalization procedure is applied before they are used as predictors in the neural network regression. First, the layer- and level minimum and maximum values X_{min} and X_{max} of the profile set for a given quantity $X(p)$ are computed for each layer and level separately. The normalized profiles with the quantity $\tilde{X}(p)$ are computed by subtracting the mean from a given profile and dividing the remainder by the standard deviation:

$$\tilde{X}(p) = \frac{X(p) - X_{min}}{X_{max} - X_{min}}. \qquad (7)$$

As such, the quantity $\tilde{X}(p)$ becomes normalized into the range $\tilde{X}(p) \in [0,1]$. This normalization step is critical in bringing the various atmospheric properties which can differ by orders of magnitude onto a common scale and performing the calculations with dimensionless numbers. However, this method is not unique, and other approaches for a normalization are conceivable, such as a normalization with the mean and standard deviation of the profile dataset.

The predictors used for the hidden-layer neural network are the normalized atmospheric properties listed in order below:

1. Temperature

2. Pressure
3. Carbon dioxide amount
4. Cumulative carbon dioxide amount
5. Ozone amount
6. Cumulative ozone amount
7. Water vapor amount
8. Cumulative water vapor amount

The list order corresponds to the neural network layout, i.e. input node 1 corresponds to predictor 1 etc., whereas for approach 2 the layer predictors repeat for each layer, that is input node 1 modulo 8 always contains the predictor 1 for a given layer. The neural network regression fit is performed separately for each instrument zenith angle, and as such the angle is also implicitly available as a predictor. Including the instrument zenith angle as a direct predictor in the neural network is straightforward and is not discussed here for the sake of brevity. A correlation plot between the listed normalized predictors and the total spectral layer-to-space transmittance for VIIRS-Moderate NPP channel 12 in layer 100 is shown in Fig. 7.

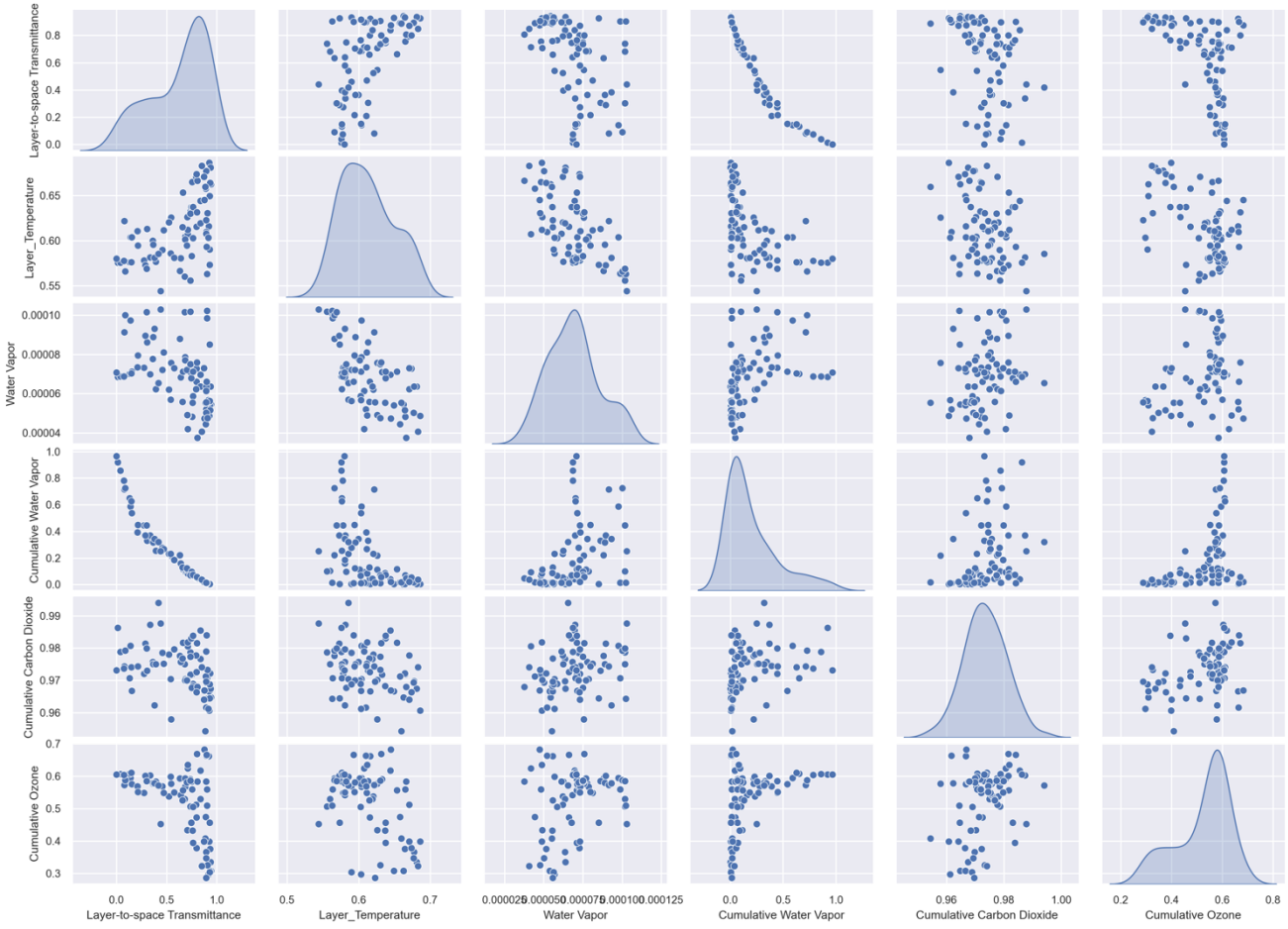


Figure 7: Correlation plot of ECMWF83 atmospheric training data profile set predictors for layer 100 and normalized via Eq. (7) and VIIRS-M NPP channel 12 total spectral layer-to-space transmittance (kernel density estimate on the diagonal). Note the clear and monotonic relationship between transmittance and cumulative water vapor amount at this layer and channel.

4.2. *Transmittance Profiles as Predictands*

The unnormalized ECMWF83 atmospheric profile data set discussed in the previous subsection is used as the input for line-by-line radiative transfer calculations to obtain a synthetic dataset of accurate monochromatic atmospheric gaseous transmittance profiles. Subsequently the monochromatic transmittance profiles are convolved with the instrument spectral response function as demonstrated in Eq. 5 to obtain the transmittance profiles at instrument channel resolution. A detailed sketch of the line-by-line calculation process is given in Fig. 8.

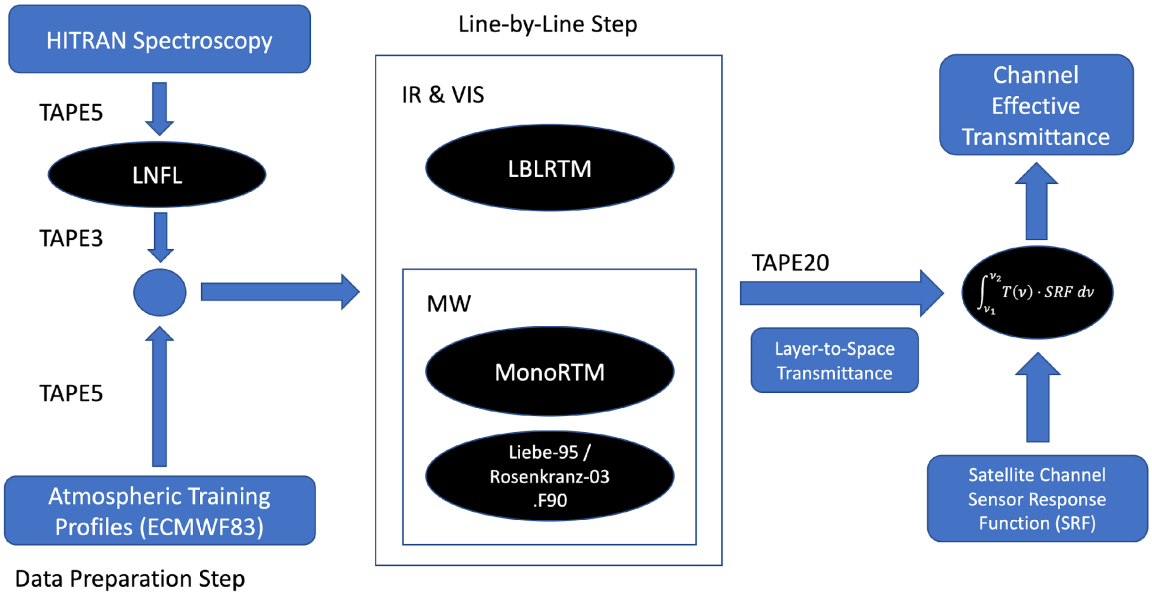


Figure 8: Flowchart of line-by-line computation and SRF-convolution workflow. The workflow is divided into a data preparation step that involves creating the appropriate I/O TAPE files for LBLRTM, a computationally expensive line-by-line computation step, and an SRF convolution step in order to transform the monochromatic transmittance data into instrument channel-resolution transmittance profiles.

The line-by-line radiative transfer model used for the monochromatic transmittance calculations in the infrared region is the *Line-By-Line Radiative Transfer Model* (LBLRTM) [1] maintained by *Atmospheric and Environmental Research* (AER). This study focuses on the SRF of the M-band of the Visible Infrared Imaging Radiometer Suite (VIIRS) onboard the Suomi NPP satellite. A plot of the SRFs for channels 12 and 13 of VIIRS-M S-NPP is shown in Fig. 9. As can be seen, the SRFs are comparatively smooth functions of the wavenumber. The convolution between the monochromatic gaseous transmittance profiles and the VIIRS-M NPP SRF is performed through numerical quadrature. The results of the total transmittance for channel 13 and the ECMWF83 profile set is shown in Fig. 10. The same transmittance for the UMBC48 testing data set is shown in Fig. 11. Similar to the predictors, the computed transmittance profiles at channel resolution should also be normalized accordingly. The resultant normalized transmittance profiles become the predictands of the neural network regression process. However, since the transmittance values already lie in the range between one and zero, the normalization step for the predictands is not necessary. In an operational radiative transfer model, the influence of variable gases needs to be calculated via effective transmittances [9], i.e. the predicted transmittances for a specific variable gas are the ratio of the total transmittance and the total transmittance without the variable gas. This must be done in order to ensure that the product of all variable gases is always equal to the total transmittance. Note that the layer transmittance normalized in this way cannot be expected to behave in the same way as the original transmittance. In particular, monotonicity constraints on the cumulative

layer-to-space transmittance and bounds on the positivity and absolute value of the layer transmittance do not hold. For the sake of simplicity this study focuses on the total transmittance alone, in order to evaluate the feasibility of the neural network approaches. The extension of the methods discussed here towards the aforementioned effective transmittances for individual gases is straightforward.

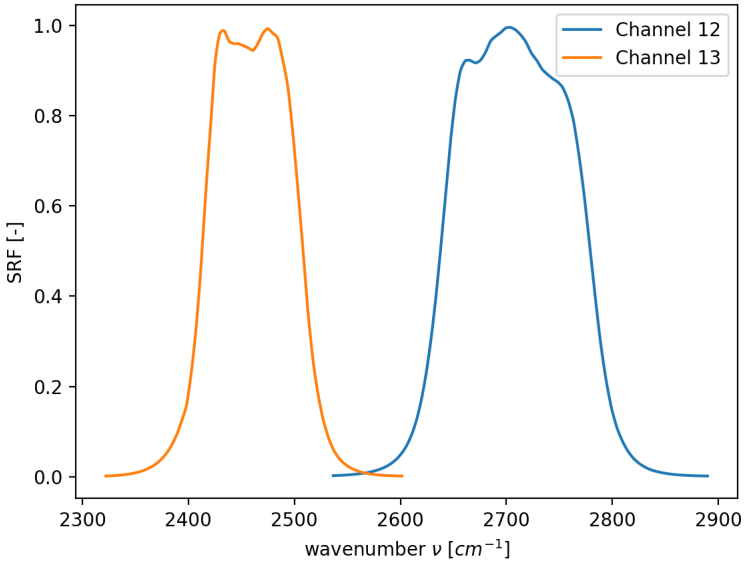


Figure 9: Channel 12 & 13 SRF for the VIIRS-M instrument onboard Suomi-NPP.

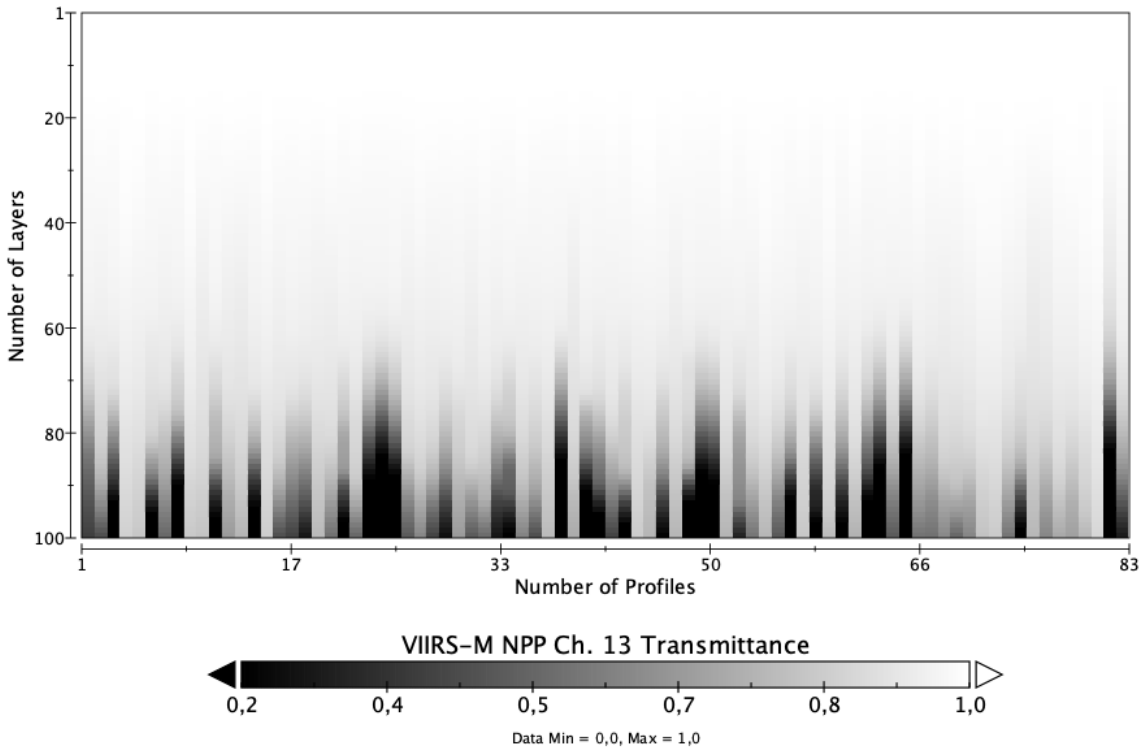


Figure 10: VIIRS-M NPP channel 13 total spectral transmittance profiles at nadir as a function of profile set number based on the ECMWF83 atmospheric profile training set. Layer 1 is at the top of the atmosphere (TOA) and layer 100 is the ground layer.

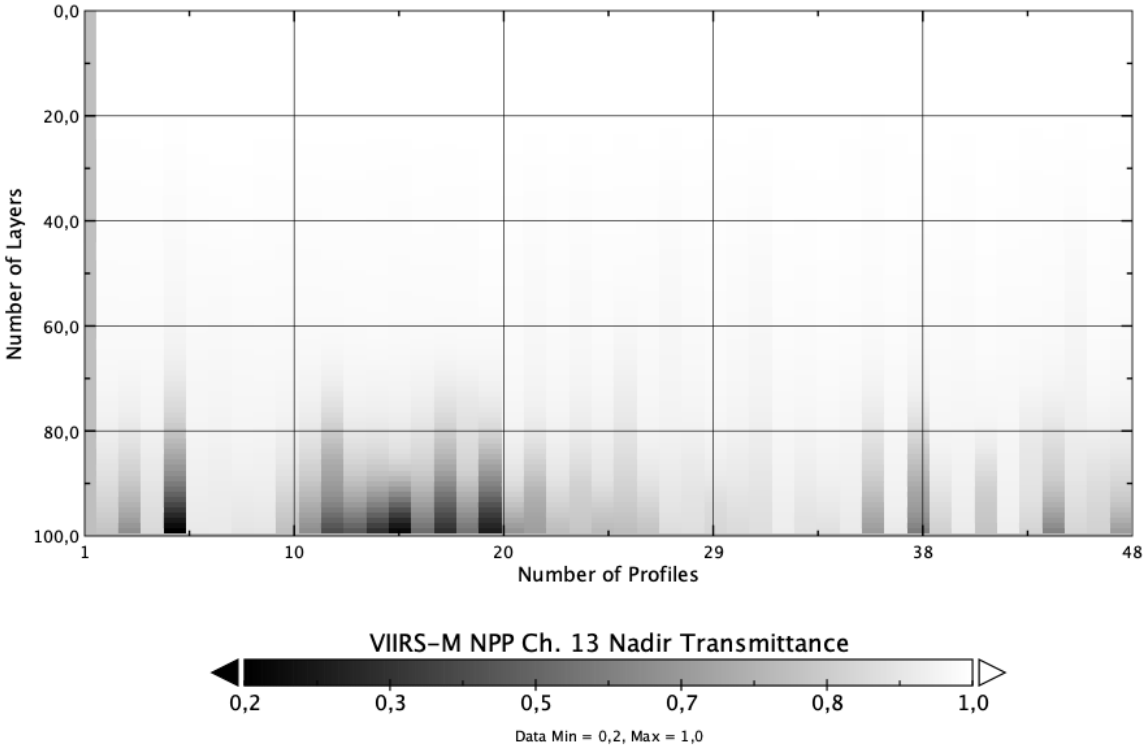


Figure 11: VIIRS-M NPP channel 13 total spectral transmittance profiles at nadir as a function of profile set number based on the UMBC48 atmospheric profile training set. Layer 1 is at the top of the atmosphere (TOA) and layer 100 is the ground layer.

5. Neural Network Results

This section discusses the results of the hidden-layer neural network regression of the layer-to-space gaseous transmittance profiles for both network topologies discussed in Section 3. This section is divided into an initial model validation, a description of the results for both networks, and a final comparison of the performance of both models. For the definition and training of the hidden-layer neural networks the TensorFlow Python API [31] was used. The loss function for the training is the mean-squared error.

5.1. *Initial Model Validation*

A simplified test problem was chosen to evaluate the practical capability of the selected hidden-layer neural network model to approximate a nonlinear multivariate function and to anticipate possible shortcomings. A suitable function for this purpose is the cosine function as a nonlinear mapping with multiple inflection points. For this preliminary test the deterministic cosine function is sampled with added Gaussian noise of mean zero and variance 0.1. The resulting dataset of x and cos(x) values is used to train the minimal hidden layer neural network shown in Fig. 2 over a limited region $x \in [0, 3\pi]$. A comparison of the deterministic cosine function and sample output from the trained neural network is shown in Fig. 12.

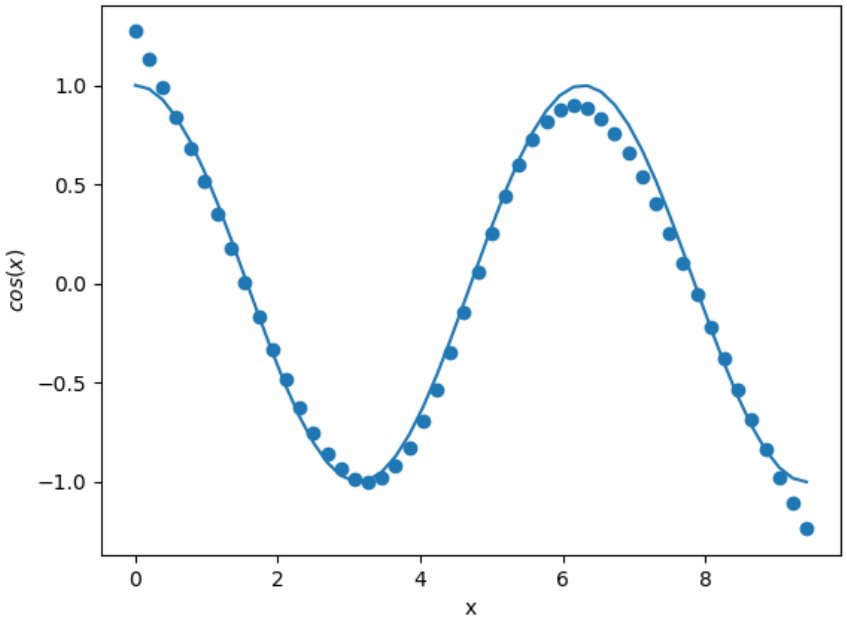


Figure 12: Comparison between a deterministic cosine function $cos\ (x)$ (blue curve) and the output of a neural network (blue dots) trained with sampled cosine data.

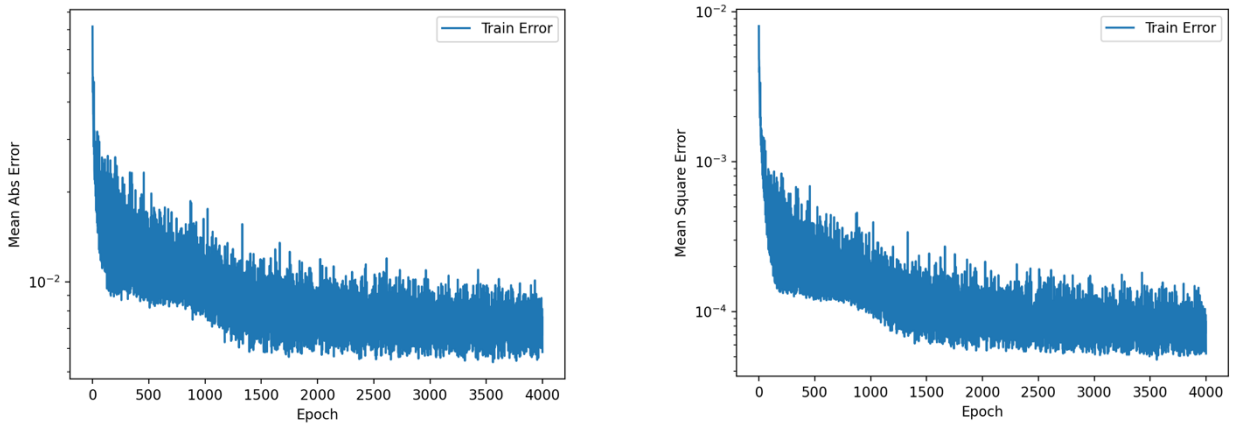
Fig. 12 demonstrates that even the minimal network example of Fig. 2 is able to reproduce the nonlinear relationship inherent in the cosine function to a specified degree. However, two key issues could be identified during the model training:

- Loss of accuracy in regions of sparse training data and data boundaries.
- Neural network settles into a local minimum.

The first issue is particularly evident in Fig. 12, where the output of the trained neural network deviates from the cosine function and becomes almost linear at the domain boundaries 0 and 3π respectively. This also means that the neural network model will likely perform poorly in the case of extrapolation into regions outside the volume of available training data. The second issue occurs because the model weights and biases are initialized randomly. Upon minimization of the loss function using gradient descent the model may settle into a local minimum where the mean-squared error is not reduced further but the regression fit nevertheless remains poor.

5.2. *Results for the Case of a Single Network per Layer*

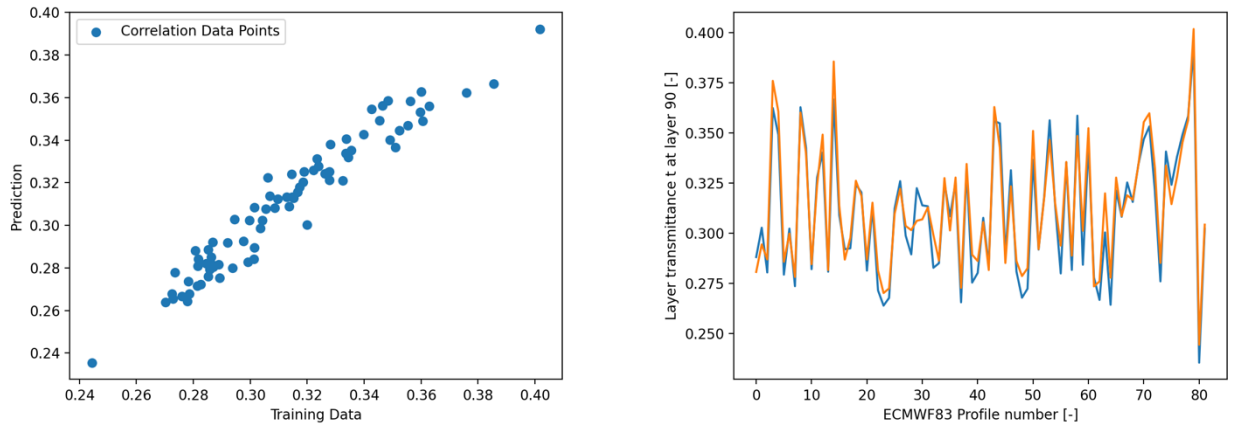
In approach 1 the regression for each atmospheric pressure layer is performed separately. The neural network in Fig. 4 is used, together with the parameters given in Table 3. As a first step, the training results for a single specified layer are investigated. Without loss of generality this layer is chosen to be layer 90, at an atmospheric pressure of 814.8 hPa. Mean-squared error and mean absolute error of the training of a single network for layer 90 and the associated convergence behaviour of the loss-function minimization are shown in Fig. 13.



a) b)

Figure 13: Mean absolute error (left) and mean square error (right) for total transmittance neural network training for Layer 90 only.

The output of the trained neural network for layer 90 is shown in Fig. 14. As can be seen, the training accuracy of the neural network model towards the normalized total transmittance as a predictand for a single layer is quite high, which is quite encouraging for the generalization towards all atmospheric layers.



a) b)

Figure 14: Correlation between normalized total transmittance training data and neural network output for Layer 90 (left) and normalized total transmittance as a function training profile number (training transmittance in blue, predicted transmittance in orange).

In order to analyze the structure of the network for a single layer and gain a measure of insight and interpretation of the network processes, it is possible to visualize the trained network weights as matrices. The absolute value of the trained network weights is shown in Fig. 15 for all layers. A higher absolute value of a particular weight consequently indicates that a particular model input has a bigger impact on the model output and the intermediate results. It is particularly illustrative to look at the weight matrix between the input layer and the first hidden layer. However, interpreting the weight matrices of subsequent hidden layers becomes progressively more difficult. In Fig. 15, the weight matrix indices correspond to the layer nodes, i.e. matrix index 0 corresponds to node 1 of a given layer. The nodes of any given layer are represented as the weight matrix row

index, whereas the nodes of the subsequent layer are given as the weight matrix column index in the TensorFlow output representation. Note that this definition is the transpose of w_{jk} in Eq. (6a). This also means that the matrix rows of the input layer correspond to the regression predictors and the matrix column of the last hidden layer corresponds to the regression predictand. Consequently, the matrix in Fig. 15a shows that the cumulative carbon dioxide amount and the level pressure have the biggest impact on the first hidden layer of the neural network for the given case, and conversely the weights of the layer pressure and cumulative water vapor amount are quite small. The highly-weighted cumulative carbon dioxide amount for instance is visible in Fig. 15a as element $w_{3,2}$ with a bright yellow color, whereas small weights are indicated as black and dark blue matrix elements. In this way it is also possible to identify predictors that do not significantly contribute to the regression and remove them entirely to selectively improve speed and memory footprint of the computations [32].

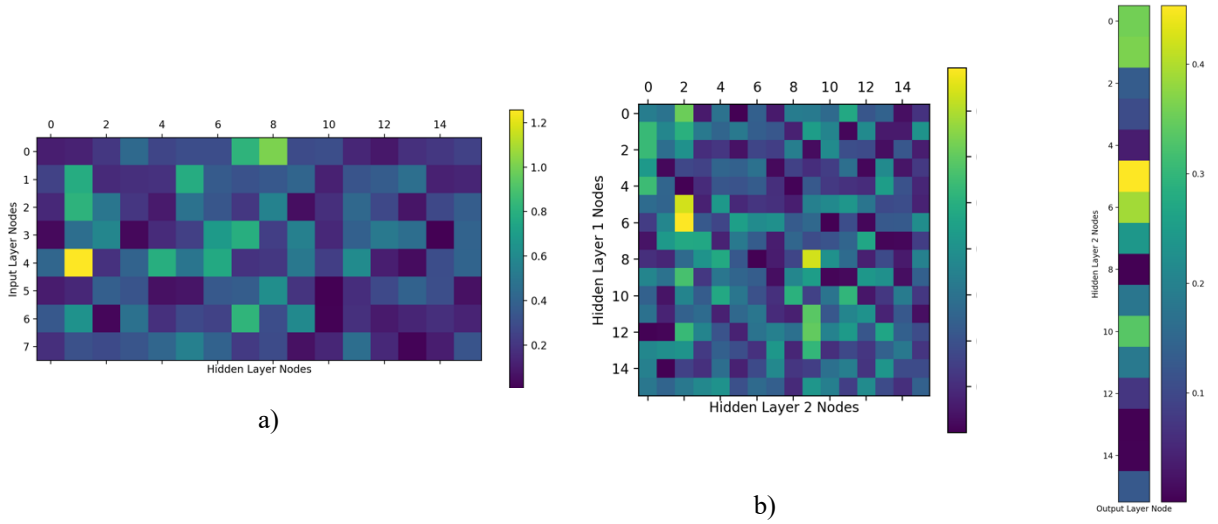


Figure 15: Node connection weight matrices w_{kj} of the network shown in Fig. 4 trained with the predictands in Fig. 13b. The corresponding layers are: a) input layer; b) hidden layer 1; c) hidden layer 2.

As the neural network model for approach 1 can successfully predict the normalized total layer transmittance for a single layer, the same principle can be applied to all layers of a given atmospheric profile. All that needs to be done is creating a separate neural network for each atmospheric layer and repeating the training process. The result for approach 1 is shown in Fig. 16. The convergence of the neural network training for every single layer is not reproduced here for the sake of brevity and Fig. 13 can be considered as representative. Fig. 16 shows the normalized total layer transmittance as a function of atmospheric pressure layers for ECMWF83 training profile number 3. The predictand profile is shown in red, while the neural network output is shown in green. The absolute difference $(\tau_{neural\ network} - \tau_{LBLRTM})$ between the two results is given as the dashed blue curve. It can be seen that the neural network model is overall very accurate and is able to reproduce all qualitative features of the line-by-line validation result. The biggest differences appear both in the lower, tropospheric layers, at TOA, and at the inflection points of the normalized layer transmittance. Overall however, the neural network result is not as smooth as the training profile and has a certain numerical noise level, as no constraints between layers are enforced. There

is a clear negative bias at TOA and a mostly positive bias near the ground. The drift of the difference changes sign near the center of the atmospheric column, where it is mostly oscillatory in nature.

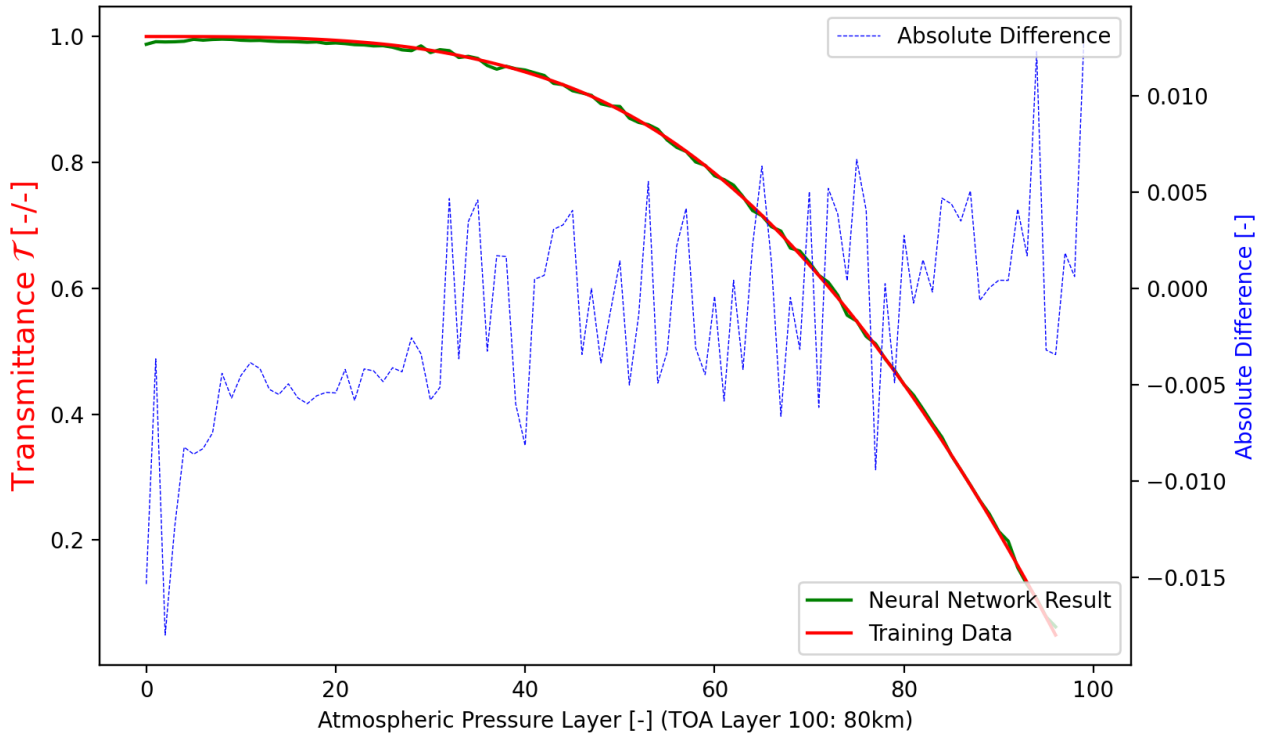
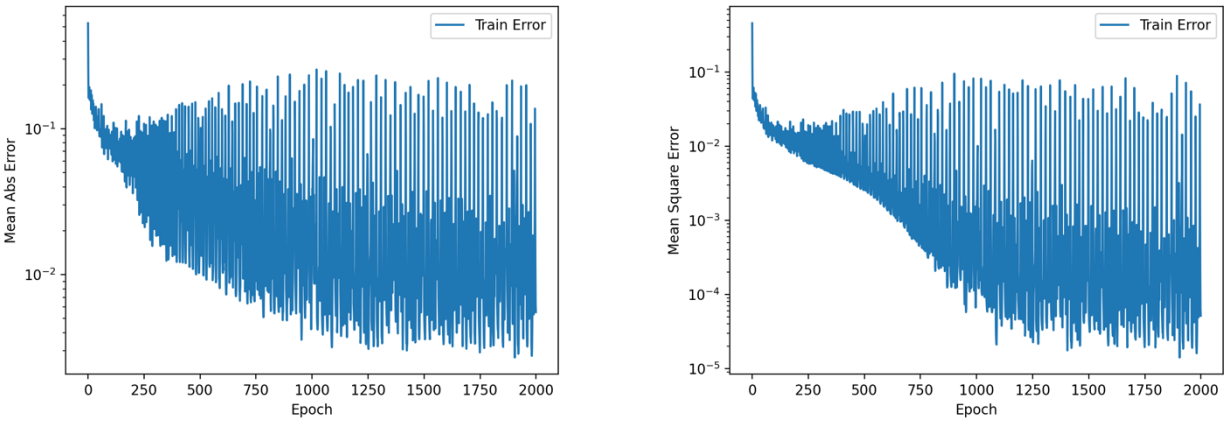


Figure 16: Normalized total layer transmittance as a function of atmospheric pressure layers for ECMWF83 training profile number 3 and approach 1.

5.3. *Results for the Case of a Single Network per Profile*

The method applied to individual atmospheric layers discussed in subsection 5.2 can be generalized to a single neural network for the entire atmospheric profile. In this way, the total training time can be reduced, relationships between atmospheric layers can be taken into consideration within the neural network and the management of the model parameters is considerably simplified. The network topology of approach 2 listed in Table 3 was again trained with the normalized ECMWF83 profile sets and the corresponding normalized total transmittance. The least-squares error convergence during the training is shown in Figure 17.



a) b)

Figure 17: Mean absolute error and mean square error convergence for the training of the neural network of approach 2 in Table 3.

The comparison of the normalized layer transmittance produced by the neural network of approach 2 and the line-by-line normalized transmittance of profile 3 is shown in Fig. 18. As in Fig. 16, the relative error is overall small and behaves very similar to the error of approach 1. There is significantly more noise in the error curves of approach 2, with high-frequency oscillations of higher error values. The numerical noise and the lack of smoothness of the neural network result are reduced for the example profile shown in Fig. 18, however.

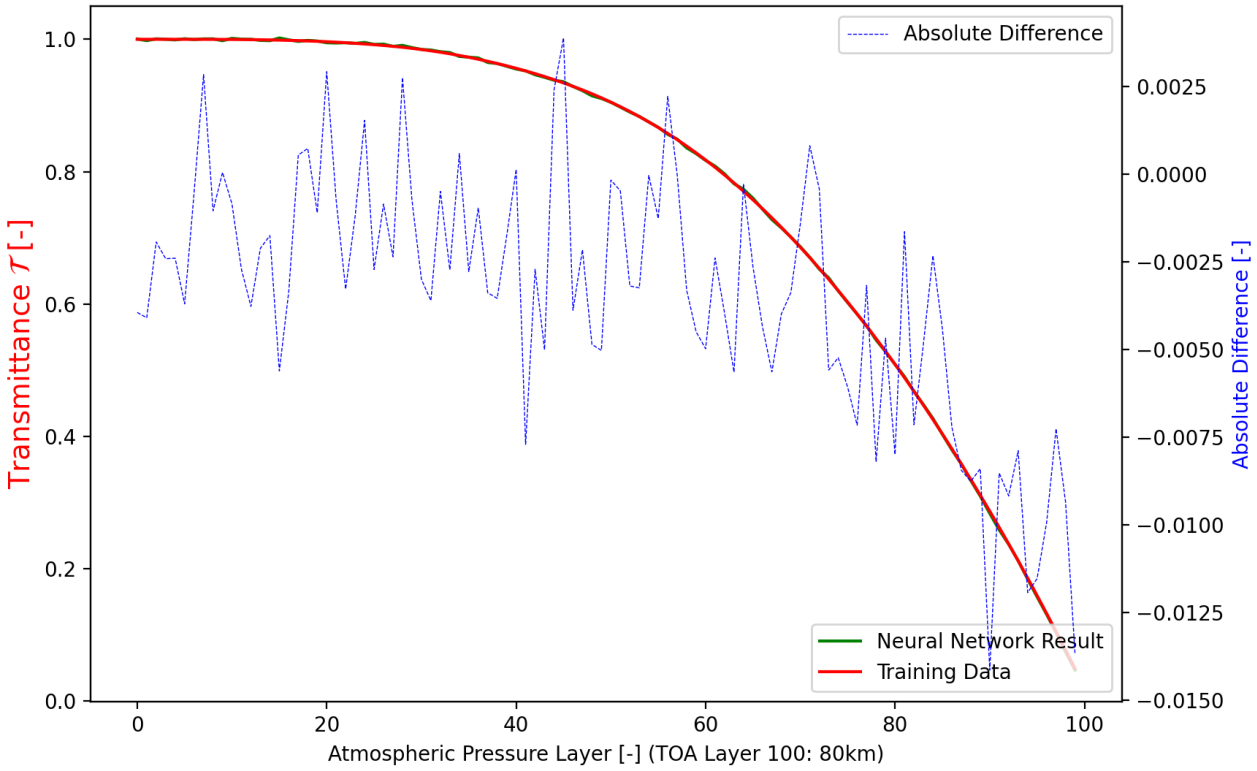
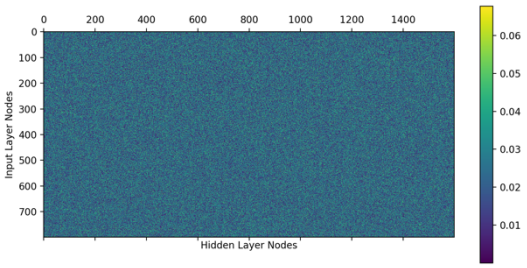
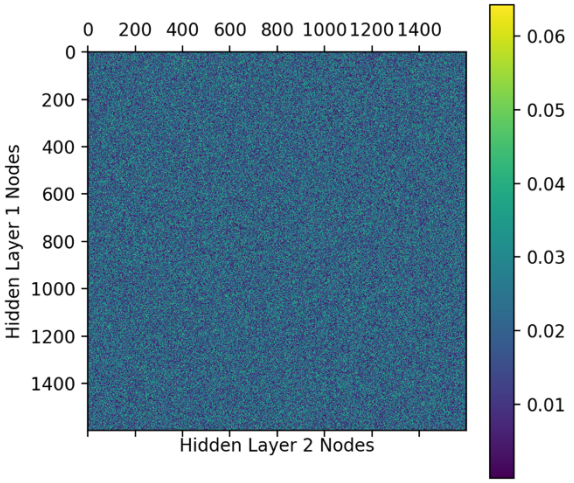


Figure 18: Normalized total layer transmittance as a function of atmospheric pressure layers for ECMWF83 training profile number 3 and approach 2.

As approach 2 leads to a smaller absolute error in the normalized layer transmittance than approach 1, looking at the *reconstructed* layer-to-space transmittance that is required for the solution of the radiative transfer equation Eq. (4) provides an upper bound of the reconstructed error for both approach 2 and approach 1. To this end, the transmittance normalization is reversed by applying the minimum and maximum values of the transmittance predictand training data set with the normalized transmittance. Fig. 18 shows the reconstructed layer-to-space transmittance as a function of atmospheric pressure, together with absolute error. It can be seen that the neural network result is highly accurate, with an absolute error below a value of 0.015. The weight matrices of the neural network in approach 2 can be inspected in the same way as for the case of approach 1, with the caveat that the matrices will be much larger. Looking at the weight matrices in Fig. 19 does not provide any insights, however. No immediate structure is visible in the matrices of Fig. 20 a) through c) and the weights appear as random noise. The weight matrices themselves seem sparse, with many values very small or close to zero. This picture is perpetuated when zooming into the weight matrix of the second hidden layer shown in Fig. 20 c) for a more detailed look. The zoom is displayed in Fig. 20 d) and also shows an apparently random and noisy pattern. Consequently, the weight matrices for approach 2 cannot be interpreted as easily as the ones of approach 1.


0
200
400
600
800
1000
1200
1400
0
100
200
300
400
500
600
700
Input Layer Nodes
Hidden Layer Nodes
0.06
0.05
0.04
0.03
0.02
0.01

a)


0
200
400
600
800
1000
1200
1400
Hidden Layer 1 Nodes
Hidden Layer 2 Nodes
0.06
0.05
0.04
0.03
0.02
0.01

b)

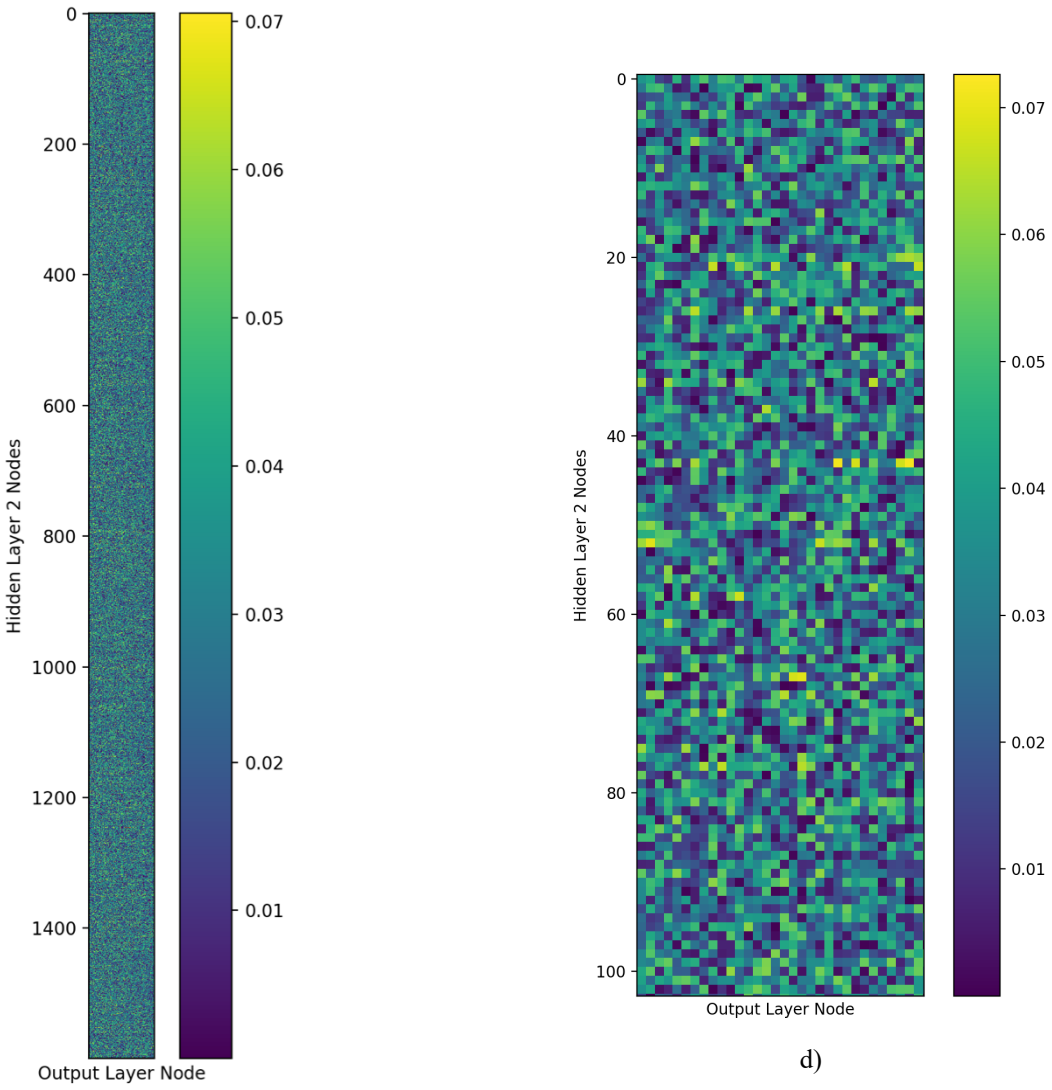


Figure 19: Analysis of the weight matrices of the trained neural network for approach 2: a) input layer; b) first hidden layer; c) second hidden layer; d) zoom of the first 100 elements of the weight matrix from c).

After checking for an acceptable fit of the model to the training data, its performance in practice needs to be gauged against an independent testing data set, which is based on the UMBC48 atmospheric profile set, as explained before. A direct comparison of an individual transmittance profile is shown in Fig. 20 for UMBC48 profile number 3. The overall absolute difference between the neural network transmittance and the line-by-line model result does not exceed a value of 0.06 in this case. Notable is a clear increase of the absolute difference value with atmospheric pressure. In other words, the neural network difference increases towards the bottom of the atmosphere. This result is somewhat surprising, as in approach 2 the entire transmittance profile is approximated simultaneously and no layer is treated preferentially. However, this may hint at a correlation between the transmittance error and physical variables such as pressure or water vapor concentration. The root mean square error (RMSE) for the entire UMBC48 testing data set is shown in Fig. 21. The RMSE value does not exceed a value of 0.025 and the same increase of the RMSE value with atmospheric pressure is observed as in the case of the individual profile, revealing a consistent trend.

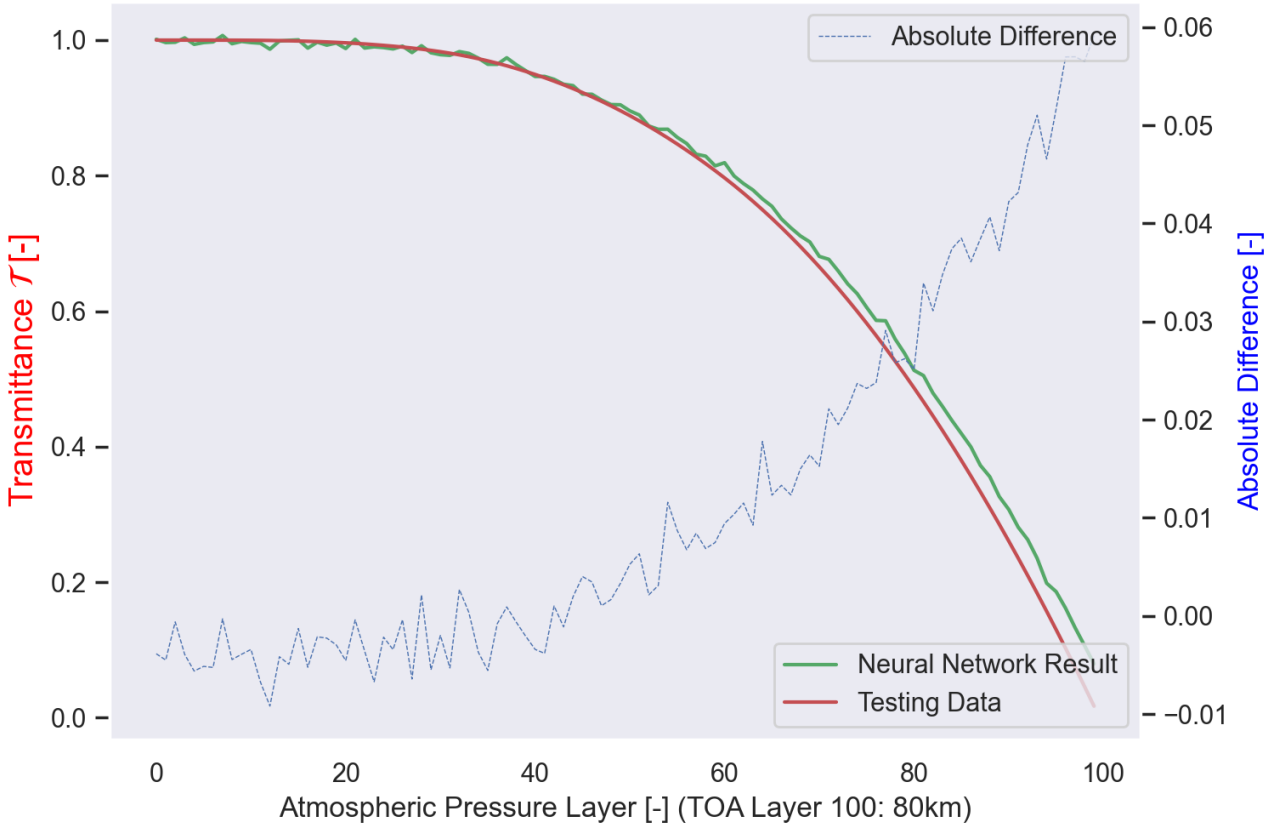


Figure 20: Normalized total layer transmittance as a function of atmospheric pressure layers for UMBC48 testing profile number 3 and approach 2.

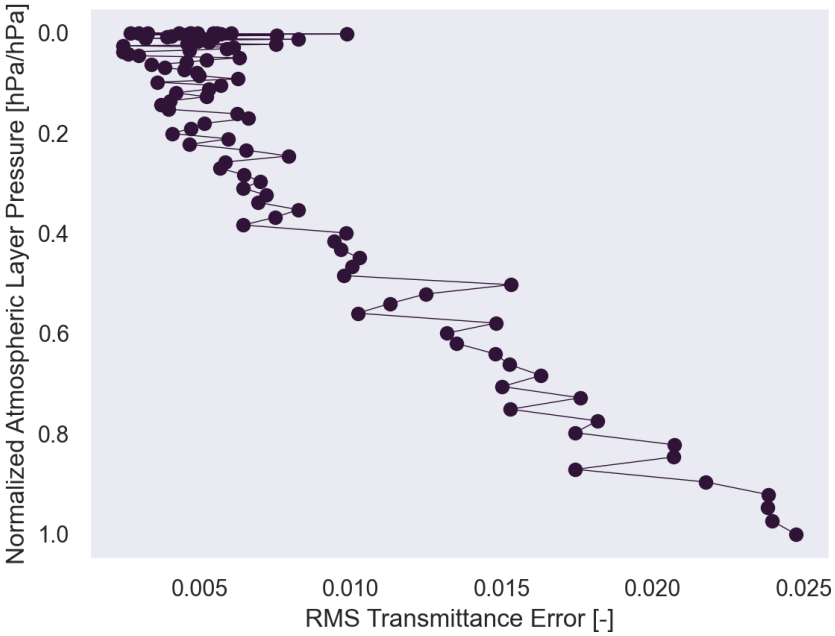


Figure 21: Root mean square error between the neural network (approach 2) and exact line-by-line model transmittance as a function of normalized atmospheric layer pressure for the UMBC48 testing data set.

5.4. *Overfitting and a reduced parameter space for approach 2*

Looking at the results of this section for approach 2 so far reveals two potentials areas of immediate improvement:

- Fig. 19 shows that a large number of parameters of the full neural network has values that are small, hinting at a potential redundancy of many of these parameters.
- The fit of the neural network towards the testing dataset is not as good as towards the training set, hinting at potential overfitting and model generalization issues.

There are several common methods in machine learning practice to address the issue of overfitting and the following solutions have been considered in this study:

- Early stopping of the training based on optimal accuracy for a validation dataset.
- L1 and L2 regularization of individual network layers.
- Dropout layers that set input units to zero as part of the network training process.
- Reducing the complexity of the neural network model by reducing individual nodes or entire layers.

Neither L1 and L2 regularization, or dropout layers individually, nor any combination of the three were found to improve the accuracy of a modified approach 2 network, even though regularization was found to slightly improve the smoothness of the result. A criterion for the early stop of the neural network training to avoid overfitting is conventionally determined based on the model accuracy towards a validation dataset, which is distinct from the previously discussed training and testing datasets. A common approach is to select the validation dataset as a fraction of the training dataset, where a value of 80% for training and 20% for validation is often used. The goal is to stop the training of the neural network not when convergence of the training error is achieved, but earlier, when the error of the model towards the validation dataset reaches its minimum, if such a minimum exists. Following this approach, the training for a modified approach 2 was conducted with an 80/20 split of the ECMWF83 profile set into a training and validation dataset respectively and the results of the corresponding training process for the neural network model were plotted to search for the minimum of the validation error. Both mean squared error and mean absolute error for both training and validation dataset over the training epochs are given in Fig. 22 a) and b) respectively.

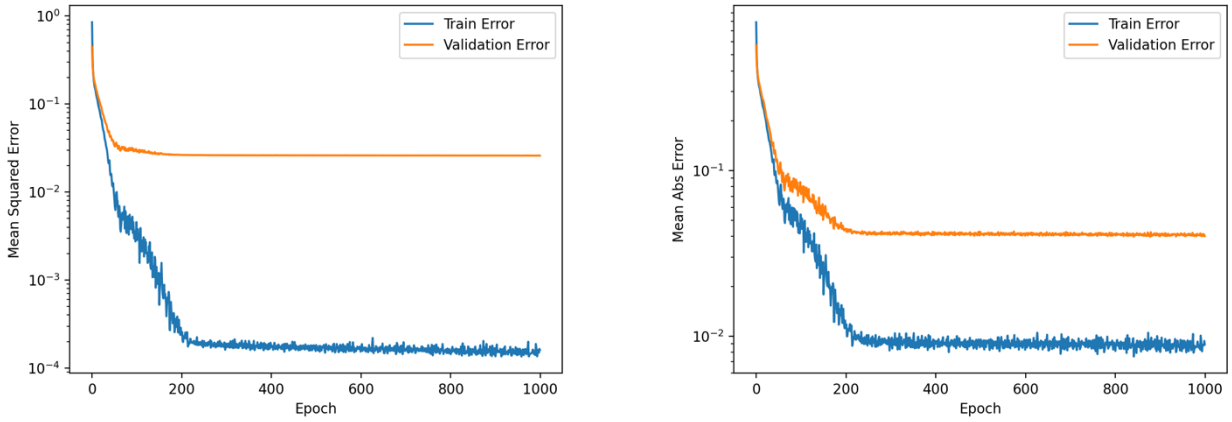


Figure 22: Mean absolute error and mean square error convergence for the training of the neural network for approach 2. Both training (blue) and validation (orange) dataset are shown. Note that the validation dataset does not obtain a minimum, but rather converges towards a steady value. This is contraindicative of overfitting through extended training times.

It can be seen that the validation dataset error does not obtain a minimum value during the training sequence, but rather converges towards a constant value, just like the training error. This means that a potential overfitting issue cannot be resolved by early stopping of the model training process in this case. Consequently, this method is also unsuitable to address the preceding issue.

The last option to address the first issue only requires reducing the model size. This was done aggressively and the full network size was reduced from a 800 x 1600 x 1600 x 100 model to 800 x 80 x 100 nodes (henceforth designated as the improved approach 2), i.e. the hidden layer was smaller in size than both the input and output layer, but still kept at a multiple of the number of different predictor quantities. The number of hidden layers was also reduced from 2 to 1. This reduced network architecture is very similar to the well-known *autoencoder* [33] neural network architecture, however, it is distinct from it as the output of an autoencoder is used to regenerate its input. Looking at the mean absolute error and mean square error of the training process in Fig. 23 shows that reducing the number of network parameters reduces the noise in the training error substantially, while at the same time maintaining a comparable mean level. Looking at the comparison of the network and line-by-line transmittance for a single UMBC48 validation profile in Fig. 24 paints a similar qualitative picture and overall trend as Fig. 20 for the full network of approach 2, albeit with a slightly reduced absolute difference value. The same is true for the RMSE of the full UMBC48 validation dataset for the optimized approach 2 in Fig. 25. As in Fig. 21, the RMSE value increases with atmospheric pressure, but is in fact slightly lower than for the full network of approach 2. This shows that not only was it possible to drastically reduce the size of the network for approach 2 due to inherent parameter redundancies, but it was also possible to slightly reduce the maximum RMSE value for the validation dataset with the optimized approach 2. Further reducing the network size did not lead to further improvements however, and the RMSE was found to increase again for smaller network sizes. The same improvement also couldn't be observed for approach 1.

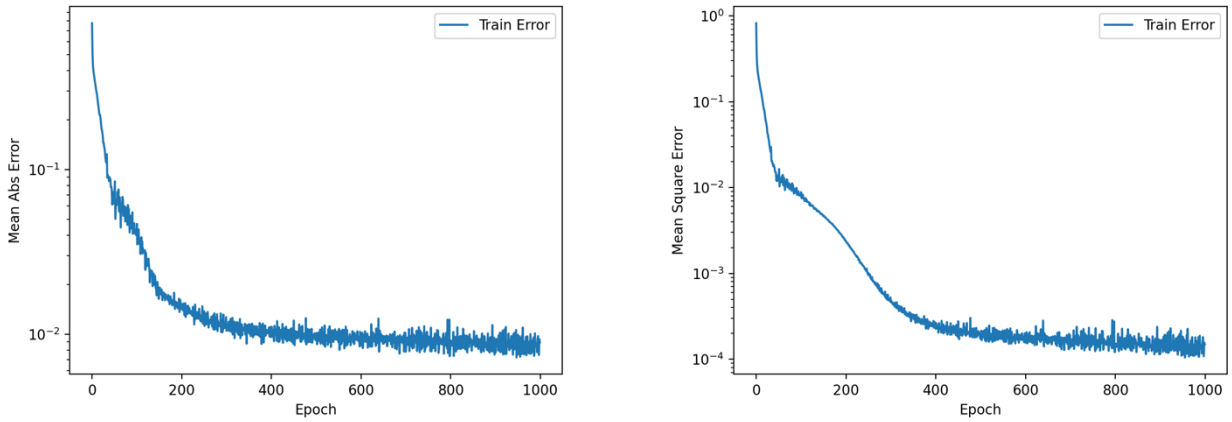


Figure 23: Mean absolute error and mean square error convergence for the training of the neural network of the optimized approach 2 in Table 3.

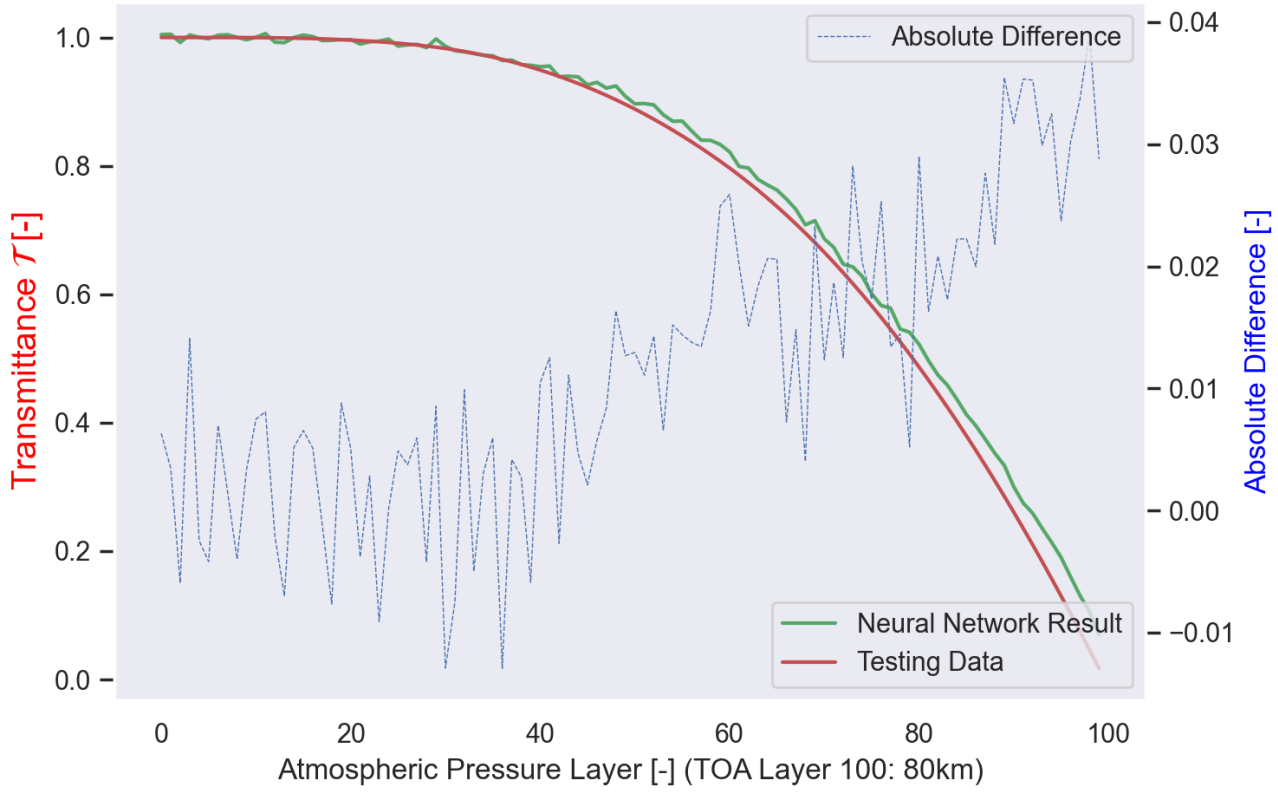


Figure 24: Normalized total layer transmittance as a function of atmospheric pressure layers for UMBC48 testing profile number 3 and optimized approach 2.

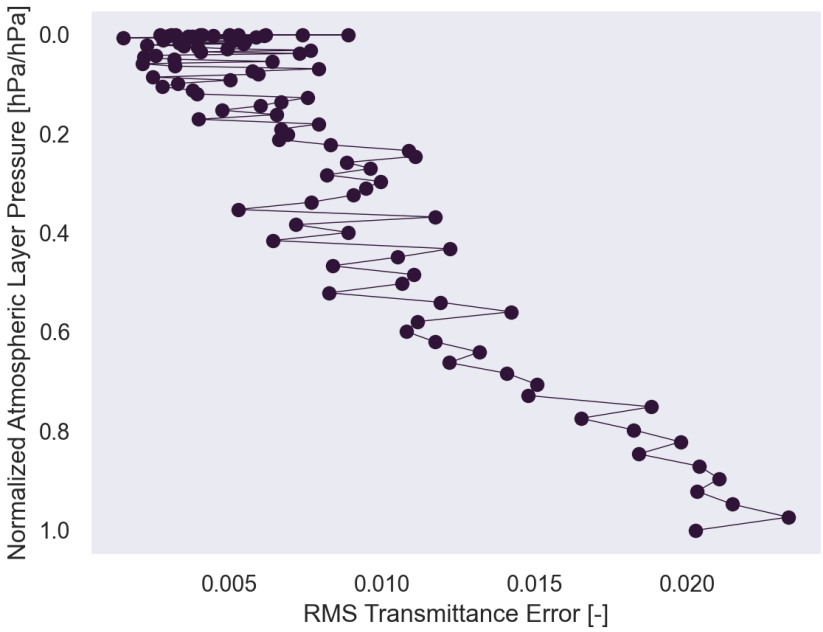


Figure 25: Root mean square error between the neural network (optimized approach 2) and exact line-by-line model transmittance as a function of normalized atmospheric layer pressure for the UMBC48 testing data set.

5.5. *Comparison of Approaches*

Both approach 1 and the optimized approach 2 allow for a very accurate computation of layer-to-space transmittances, with approach 1 being overall less accurate than the optimized approach 2. Additionally, the optimized approach 2 allows for a faster training and an easier handling of the neural network parameters, which is of particular concern for the implementation of a radiative transfer model. For the given normalization strategy, the optimized approach 2 also leads to more accurate results and the network size and performance can be further improved upon by reducing its size, leading to autoencoder approach 2. Approach 1 however makes it easier to understand the relationship between the predictors and the model output by analyzing the model weight matrices. Overall, approach 2 is simpler and more straightforward, as only one neural network is required for the entire problem, instead of one neural network per atmospheric layer, and leads to more accurate overall results. This again simplifies the implementation in a radiative transfer model. As a clear conclusion, autoencoder approach 2 with its small network size and most accurate outcome should be the preferred approach for this kind of physical problem and predictor normalization strategy. However, this conclusion may not be generalized and facing a different kind of physical problem or even using a different predictor normalization approach for the same problem may lead to a different outcome.

5.6. *Radiance Computation*

Ultimately, the purpose of a radiative transfer solver for satellite data assimilation and remote sensing is not the computation of transmittances, but radiances at instrument resolution. To demonstrate the suitability of the neural network approach

number 2 to this application, the network was integrated with a simple radiative transfer solver for an absorbing and emitting atmosphere [21]. Scattering was ignored, as were polychromatic effects in the Planck emission function. The radiance source was surface emission at a temperature of 283.15 K and an emissivity of 0.92, which is characteristic for an ocean surface. The radiative transfer solver is solving the clear-sky radiative transfer equation Eq. (4), and the trained neural network of method 2 is providing the necessary fast parameterization of the layer-to-space transmittance. The corresponding results are shown in Fig. 26.

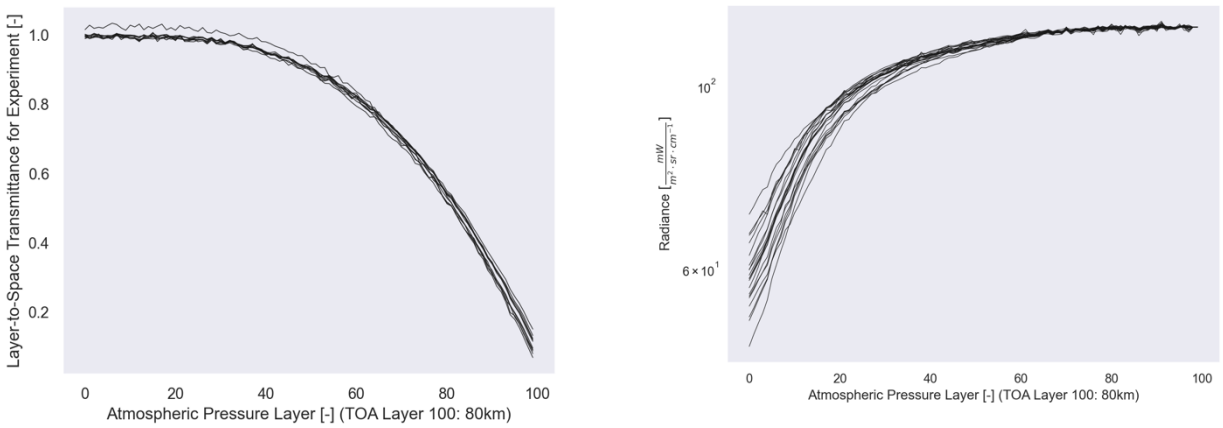


Figure 26: Radiative transfer calculations for selected UMBC48 profiles with a neural network spectral transmittance parameterization according to scheme 2. The predicted transmittance is shown in Fig. 26a and the resulting radiance in Fig. 26b.

A last issue to consider particularly for remote sensing and data assimilation applications are the computational resources required by the neural network radiative transfer solver. The normalized computation time for the neural network transmittance parameterization (2nd. approach) alone is shown in Fig. 27. In general, the time required by the neural network transmittance computation in its current naive implementation is about one order of magnitude faster than the radiative transfer solver part of the code and scales approximately linearly with the number of atmospheric profiles to be processed.

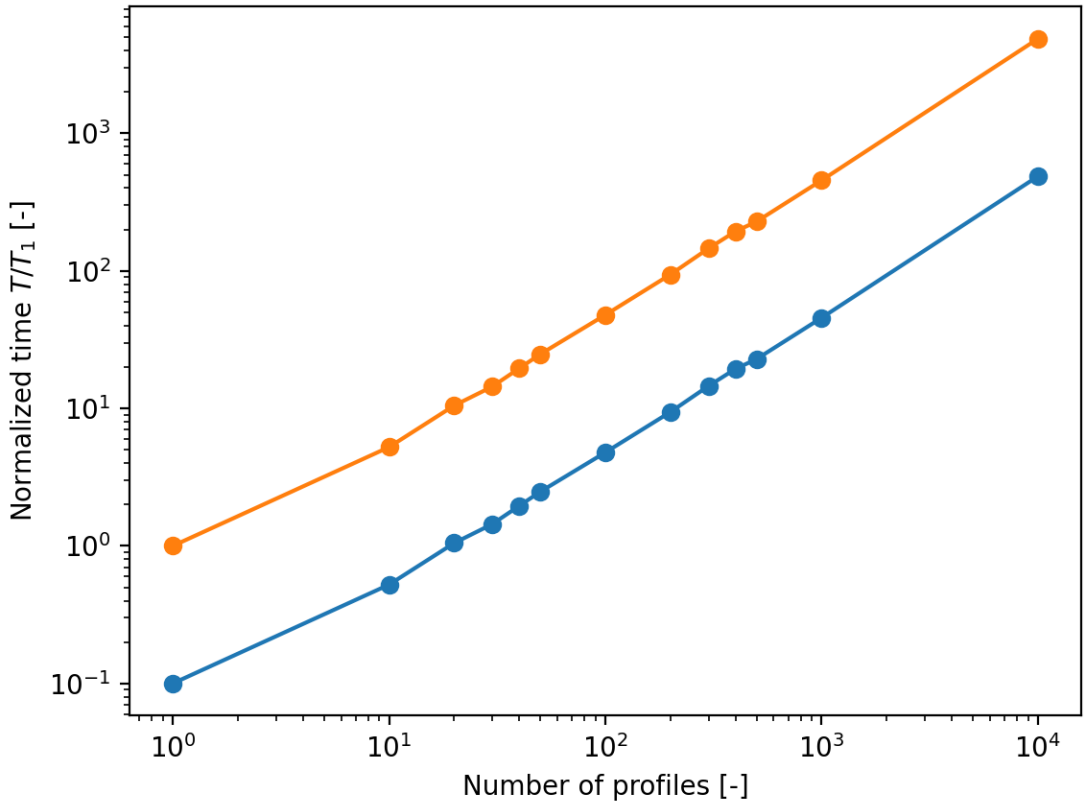


Figure 27: Normalized computation time of the optimized neural network model 2 for fast transmittance parameterization as a function of the number of atmospheric profiles (full radiative transfer solver in orange, neural network section in blue).

6. Conclusion

This work demonstrates the feasibility of employing neural networks for a fast transmittance parameterization in radiative transfer models for remote sensing or satellite data assimilation. To the knowledge of the authors, this is the first time that neural networks are investigated as a fast transmittance model, particularly in an operational context. The importance of the current findings cannot be understated, as they pave the way for further investigations towards more general applications, such as the computation of forward model Jacobians, the inclusion of physical constraints in the networks, challenges from hyperspectral instruments, and all-sky radiance assimilation with scattering clouds. Especially within the context of the CRTM, the development of new practical fast radiative transfer algorithms has effectively stagnated since 2010, and this manuscript is the first real attempt at modernizing the flagship radiative transfer model of NOAA and the JCSDA for satellite radiance assimilation since then.

Two initial approaches have been compared w.r.t. their accuracy and computational aspects. The first approach uses a neural network per atmospheric layer to predict its associated layer-to-space transmittance based on a given number of predictands. The second approach uses a single neural network for an entire atmospheric profile to do the same. It was found that approach 1 allows for easier analysis of the statistical relationships. However, approach 2 is simpler, more accurate and its training is faster. Lastly, the accuracy and efficiency of approach 2 can be further improved by decreasing the model size, leading to an optimized version of approach 2. The main strengths of both approaches are a very accurate parameterization of the

atmospheric layer-to-space transmittance, the ease of integration in existing radiative transfer solver frameworks for both clear-sky and all-sky scenes, and the lack of data-induced drift of the results that occurs with other approaches [17]. In general, it was found that both approaches were quite practical and useful. However, for the given physical problem and predictor normalization strategy, the optimized approach 2 is clearly preferable over approach 1.

Acknowledgements

This research of the NOAA OAR Office of Weather and Air Quality is supported by NOAA's Science Collaboration Program and administered by UCAR's Cooperative Programs for the Advancement of Earth System Science (CPAESS) under awards NA16NWS4620043, NA18NWS4620043B, and NA20NWS4620043C.

The computations were performed on the S4 system of the Space Science and Engineering Center of the University of Madison-Wisconsin.

We would like to thank Dr. Marco Matricardi from ECMWF and Dr. Liu Haixia from NOAA EMC for helpful discussions. Lastly, we would like to thank two anonymous reviewers for their comments and their help in improving the quality of this manuscript.

References

[1] Clough, S. A., and M. J. Iacono (1995): Line-by-line calculations of atmospheric fluxes and cooling rates II: Application to carbon dioxide, ozone, methane, nitrous oxide, and the halocarbons. J. Geophys. Res. 100(16), 519-16.

[2] Mlawer, E. J., S. J. Taubman, P. D. Brown, M. J. Iacono, and S. A. Clough (1997): RRTM, a validated correlated-k model for the longwave. J. Geophys. Res. 102(16), 663-16.

[3] Liou, K. N. (2002): *An Introduction to Atmospheric Radiation*, 2nd ed. Academic Press, San Diego.

[4] Edwards, D. P. and G. L. Francis (2000): Improvements to the correlated-k radiative transfer method: Application to satellite infrared sounding. J. Geophys. Res. 105(D14), 18135-56.

[5] Moncet J.-L., G. Uymin, P. Liang, and A. E. Lipton (2015): Fast and Accurate Radiative Transfer in the Thermal Regime by Simultaneous Optimal Spectral Sampling over all Channels. J. Atm. Sci. 72, 2622-41.

[6] McMillin, L. M., and H. E. Fleming (1976): Atmospheric transmittance of an absorbing gas: A computationally fast accurate transmittance model for absorbing gases with constant mixing ratios in inhomogeneous atmospheres. Appl. Opt. 15, 358-363.

[7] Chen, Y., Y. Han, P. Van Delst, and F. Wenig (2010): On water vapor Jacobian in fast radiative transfer model. J. Geophys. Res. 115(D12303).

[8] Chen, Y., Y. Han, and F. Weng (2012): Comparison of two transmittance algorithms in the community radiative transfer model: Application to AVHRR. J. Geophys. Res. 117(D06206).

[9] Matricardi, M., F. Chevallier, G. Kelly, and J.-N. Thepaut (2004): An improved fast general radiative transfer model for the observation of radiance observations. Q. J. R. Meteorol. Soc. 130, 153-73.

[10] Strow, L. L., S. E. Hannon, S. De Souza-Machado, H. E. Motteler, and D. C. Tobin (2003): An Overview of the AIRS Radiative Transfer Model. IEEE Trans. Geosci. Rem. Sens. 41(2).

[11] Ding, J., P. Yang, M. D. King, S. Platnick, X. Liu, K. G. Meyer, and C. Wang (2019): A fast vector radiative transfer model for the atmosphere-ocean coupled system. J. Quant. Spec. Rad. Tran. 239(106667).

[12] Yang, Q., X. Liu, W. Wu, P. Yang, and C. Wang (2015): Training and validation of the fast PCRTM_solar model. in 2015 AGU Fall Meeting Abstracts.

[13] Liu, X., Q. Yang, H. Li, Z. Jin, W. Wu, S. Kizer, D. K. Zhou, and P. Yang (2016): Development of a fast and accurate radiative transfer modeling the solar spectral region. Appl.Opt. 55(29), 8236-47.

[14] Chevallier, F., F. Cheruy, N. A. Scott, and A. Chedin (1998): A Neural Network Approach for a Fast and Accurate Computation of a Longwave Radiative Budget. J. Appl. Meteorol. 37, 1385-97.

[15] Chevallier, F., J.-J. Morcrette, F. Cheruy, and N. A. Scott (2000): Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. Q. J. R. Meteorol. Soc. 126, 761-776.

[16] Krishnan, P., K. S. Ramanujam, and C. Balaji (2012): An artificial neural network based fast radiative transfer model for simulating infrared sounder radiances. J. Earth Syst. Sci. 121(4), 891-901.

[17] Liang, X., and Q. Liu (2020): Applying Deep Learning to Clear-Sky Radiance Simulation for VIIRS with Community Radiative Transfer Model – Part 2: Model Architecture and Assessment. Rem. Sens. 12(3825).

[18] Stamnes K., and J. Stamnes (2015): Radiative Transfer in Coupled Environmental Systems: An Introduction to Forward and Inverse Modeling. Wiley-VCH Verlag Weinheim, Germany.

[19] Gao, M., B. A. Franz, K. Knobelspiesse, P.-W. Zhai, V. Martins, et al. (2021): Efficient multi-angle polarimetric inversion of aerosols and ocean color powered by a deep neural network forward model. Atmos. Meas. Tech. 14, 4083-4110.

[20] Efremenko, D., A. Doicu, D. Loyola, and T. Trautmann (2014): Optical property dimensionality reduction techniques for accelerated radiative transfer performance: Application to remote sensing total ozone retrievals. J. Quant. Spec. Rad. Trans. 133, 128-135.

[21] Rodgers, C. D. (2000): *Inverse Methods for Atmospheric Remote Sensing*. World Scientific Publishing Singapore.

[22] Chen, Y., F. Weng, Y. Han, and Q. Liu (2011): Planck-Weighted Transmittance and Correction of Solar Reflection for Broadband Infrared Satellite Channels. J. Atm. Oc. Tech. 29, 382-96.

[23] LeCun, Y., Y. Bengio, and G. Hinton (2015): Deep Learning. Nature 521, 436-444.

[24] Tarantola, A. (2005): Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM Publishing Philadelphia.

[25] Kingma, D. P., and J. L. Ba (2015): ADAM: A Method for Stochastic Optimization. In 2015 ICLR Conference Abstracts.

[26] Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986): Learning representations by back-propagating errors. Nature 323, 533-536.

[27] Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018): Automatic Differentiation in Machine Learning: a Survey. J. Mach. Learn. Res. 18, 1-43.

[28] Hascoet, L. and V. Pascual (2013): The Tapenade Automatic Differentiation tool: principles, model, and specification. ACM Transactions on Mathematical Software 39(3), 1-43.

[29] Hornik, K. (1991): Approximation capabilities of multilayer feedforward networks. Neural Networks 4(2), 251-257. doi:10.1016/0893-6080(91)90009-T

[30] Chevallier, F., S. Di Michelle, and A. P. McNally (2006): Diverse profile datasets from the ECMWF 91-level short-range forecasts. NWP SAF Report No. NWPSAF-EC-TR-010.

[31] Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al. (2015): TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Google White Paper.

[32] Del Frate, F., M. Iapaolo, S. Casadio, S. Godin-Beekmann, and M. Petitdidier (2005): Neural networks for the dimensionality reduction of GOME measurement vector in the estimation of ozone profiles. J. Quant. Spec. Rad. Trans. 92, 275-291. doi:10.1016/j.jqsrt.2004.07.028

[33] Kramer, M. A. (1991): Nonlinear Principal Component Analysis using Autoassociative Neural Networks. AIChE Journal 37(2), 233-243.