Neural network cloud screening algorithm Part I: A synthetic case over land surfaces using micro-windows in O_2 and CO_2 near infrared absorption bands with nadir viewing

Thomas E. Taylor,[a] and D. M. O'Brien[b]

[a]Department of Atmospheric Science, Colorado State University, Fort Collins, CO 80523
tommy@atmos.colostate.edu

[b]Cooperative Institute for Research in the Atmosphere, Fort Collins, CO 80523
obrien@atmos.colostate.edu

Abstract. A neural network is presented for estimating cloud water and ice paths, effective scattering heights of cloud water and ice, and column water vapor. The cloud water and ice are then used to classify scenes as either clear or cloudy using a simple threshold test of 2 gm^{-2} for water and 10 gm^{-2} for ice. Training of the neural networks was performed using high resolution spectra in micro-windows of O_2 and CO_2 near infrared absorption bands generated from an ensemble of analyzed meteorological fields from ECMWF and surface properties from MODIS. An independent test data set was generated using the same radiative transfer model, but coupled with atmospheric profiles derived from CloudSat and Calipso data. Analysis indicates that the algorithm provides approximately 75–90% accuracy with a 95-99% confidence level for classifying scenes as either cloudy or clear over land surfaces in nadir viewing geometry. These estimates are shown to be robust, in the sense that they are insensitive to realistic instrumental errors, errors in the meteorological analyses and surface properties, and errors in the simulations used for training.

Keywords: neural networks, clouds, remote sensing, carbon dioxide, radiative transfer, satellites.

1 INTRODUCTION

As the importance of atmospheric chemistry increases, so too will measurements of trace gases from space using high resolution spectroscopy. Examples include the European Space Agency's SCIAMACHY, Japan's greenhouse gases observing satellite (GOSAT) and NASA's ill-fated Orbiting Carbon Observatory (OCO), which failed to reach orbit. Each of these instruments measures high resolution spectra of reflected sunlight in the near infrared, from which trace gas amounts can be inferred in clear (or near clear) sky conditions.

High resolution spectroscopy from space also opens new opportunities for cloud detection. In observations deep within gas absorption lines, photons reflected from the surface are strongly absorbed, leaving as the dominant component of the reflected radiance those photons scattered by clouds and aerosols. Thus, detection of clouds against a bright background, historically a difficult problem in remote sensing, should be possible with high resolution spectra. For measurements of CO_2, where high absolute accuracy is required and bright surfaces (such as sun-glint) offer a signal-to-noise advantage, this is an important feature.

In this paper we examine the feasibility of cloud detection in the context of measurements of CO_2 by a hypothetical sensor, such as an etalon spectrometer, with high spectral resolution over narrow spectral windows. As noted in Refs. [1, 2] and others, the measurement of CO_2

Received 28 May 2009; accepted 24 Aug 2009; published 9 Sep 2009 [CCC: 19313195/2009/$25.00]

Table 1. Frequency windows and resolutions assumed for the spectrometer.

	O_2 A-band	strong CO_2 band	weak CO_2 band
Minimum wavenumber (cm^{-1})	13056.7	6203.6	4835.1
Maximum wavenumber (cm^{-1})	13061.3	6207.3	4838.8
Gaussian FWHM (cm^{-1})	0.05	0.05	0.05
Monochromatic samples per band	1001	801	801
Convolved channels per band	47	37	37

requires not only spectra in a CO_2 absorption band but also spectra in a nearby band of a well-mixed gas, usually O_2, to provide the normalization needed to compute the column averaged CO_2 dry air mole fraction. We will suppose that the measurements are in the O_2 A-band at 0.76 μm, the weak CO_2 band at 1.6 μm and the strong CO_2 band at 2.0 μm. Furthermore, we will investigate the ability of neural networks to detect the subtle changes in the spectra caused by cloud. Our principal objective will be to determine how reliably neural networks can detect the presence of cloud, rather than the more difficult task of predicting its optical properties and distribution in the atmosphere. The numerical experiments here are conducted with simulated data over land surfaces only and in the nadir viewing mode. Part II of the manuscript will extend the technique to lower resolution spectra and to the glint viewing mode as would have been acquired by the OCO. Later studies will explore the application to measurements acquired by the Japanese satellite GOSAT, which was launched in January 2009.

In Sec. 2 we describe the spectra used to train the neural networks. Mostly this draws upon analyzed meteorological profiles from ECMWF [3], together with surface properties from MODIS. Sec. 3 is devoted to the architecture and training of the hierarchy of neural networks needed to span the wide range of cloud conditions encountered in the simulations. Sec. 4 presents the results for two data sets, one a subset of the simulated spectra withheld from the training (the 'hold-over' data set), and the other a completely independent set of spectra derived from CloudSat observations with collocated and simultaneous meterological fields from ECMWF analyses. Finally, Sec. 5 contains the conclusions.

2 SYNTHETIC SPECTRA

Spectra were simulated for the hypothetical etalon spectrometer with the bands and spectral resolutions shown in Table 1. In order to ensure natural variations in the surface, the spectrometer was assumed to follow Worldwide Reference System-2 (WRS-2) orbit paths [4]. The specific path used for the training data was that for September 13, 2006 at 19:35 UTC as shown in Fig. 1. Simulated radiances were calculated for the nadir viewing direction only. Sun glint calculations will be addressed in Part II. Land surfaces were assumed to be polarizing and non-isotropic, with the bidirectional reflectance distribution function (BRDF) and polarizing properties synthesized from MODIS BRDF and POLDER data. Over the oceans the reflectance was calculated using a facet model [5] with a small, additional component from reflection by whitecaps [6]. The solar position was also taken from the WRS-2 orbit path, providing a range of solar zenith angles typical of a sun-synchronous orbit.

Atmospheric profiles (12,991 in total) were drawn from the ECMWF data base [3] sequentially as a function of orbit position. Formal selection criteria were not imposed to force matches between the profile type and the surface type defined by the location in the orbit, so mismatches between atmospheric and surface properties do exist in the training data. This in no way invalidates the results presented here, but the retrievals might be improved by forcing the training data to be consistent.

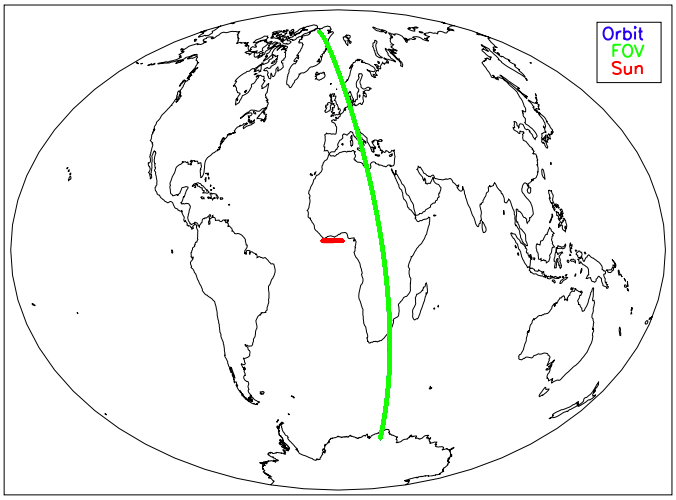

Fig. 1. The WRS-2 orbit (starting at 19:25 UTC on September 13, 2006) used in selecting surface properties for the simulations. The ground track of the satellite is shown in green, while the track in red is that of the sun.

Statistics of the cloud liquid water and ice water paths* for the 12,991 profiles are shown in Fig. 2. Note that for clarity the profiles with cloud water and ice identically zero have been ommitted from the plots (1714 profiles for water and 1402 for ice). Furthermore, profiles with cloud amounts greater than 1000.0 g/m^{-2} (1075 profiles for water and 100 for ice) have been reassigned to this value for clarity in the plots. Also shown are the effective pressure heights of water and ice, defined by

$$p_{cw} = \sum_j p_j c_{w,j} / \sum_j c_{w,j} \qquad \text{and} \qquad p_{ci} = \sum_j p_j c_{i,j} / \sum_j c_{i,j},$$

where $c_{w,j}$ and $c_{i,j}$ denote the mass densities of water and ice in layer j, expressed in units of gm^{-2}, p_j is the corresponding layer pressure, and the summations extend over the sixty layers of the ECMWF data set. The ensemble means for the column totals of water and ice,

$$c_w = \sum_j c_{w,j} \qquad \text{and} \qquad c_i = \sum_j c_{i,j},$$

are approximately 270 gm^{-2} and 100 gm^{-2}, but these quantities are extremely variable. It is important to capture this variability when training the neural networks, for otherwise the predictive skill of the networks will be impaired.

Similarly, Fig. 3 presents histograms of surface pressure, surface albedo in the O_2 A-band, column water vapor w (in units of kgm^{-2}) and mean temperature, the last defined by

$$\overline{T} = \sum_j T_j \delta p_j / \sum_j \delta p_j,$$

where T_j and δp_j denote the temperature and the pressure thickness of the j^{th} atmospheric layer. Again, each of these variables covers a wide range, representative of real world statistics.

*The term 'cloud liquid water path' will be abbreviated to simply 'cloud water' or even 'water' when there is little chance of confusion. Similar abbreviations will be used for ice.

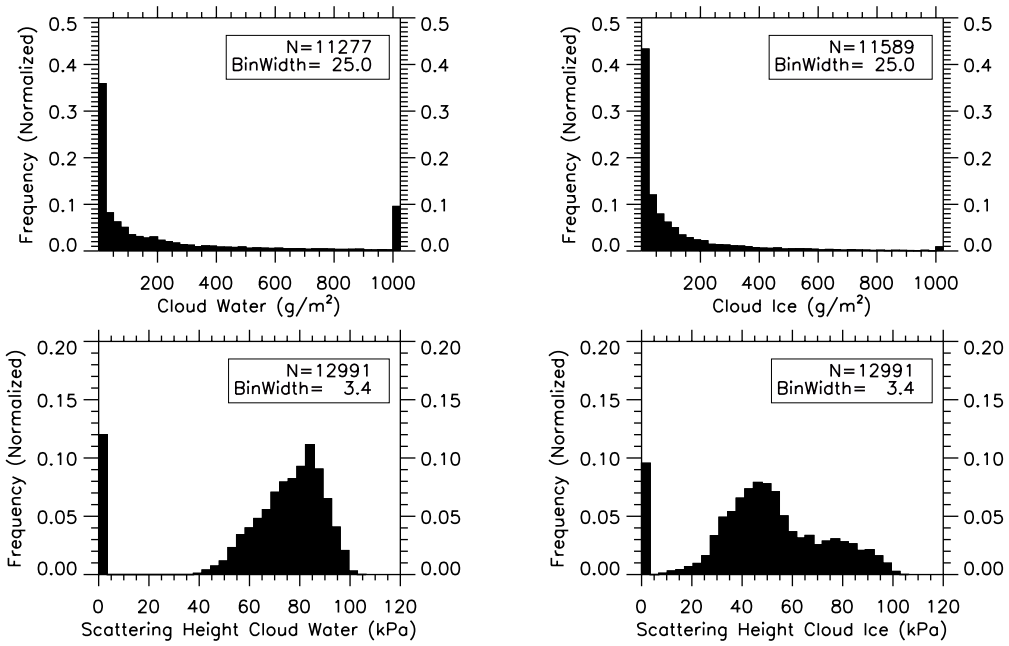

Fig. 2. Histograms of cloud water (upper left), cloud ice (upper right), effective scattering height of cloud water (lower left) and effective scattering height of cloud ice (lower right) derived from the 12,991 ECMWF profiles.

Spectra were computed using a realistic solar model with a multiple-scattering radiative transfer code [7, 8]. In order to expedite the process, the low-streams interpolator [9] was employed to provide fast radiative transfer with minimal impact upon accuracy. Monochromatic radiances were computed at 1001 points in the window of the O_2 A-band, and at 801 points in each of the weak and strong CO_2 bands. The monchromatic spectra were convolved with gaussian line shape functions to produce 47 'measurements' in the O_2 A-band and 37 in each of the CO_2 bands.

The optical properties of ice clouds assumed the distributions of ice crystals to be bi-modal, with the large mode temperature dependent [10]. Scattering properties were taken from Refs. [11, 12] and [13]. For water clouds, the effective radius was modeled as in Ref. [14], which in turn was derived from aircraft observations. The absorption and scattering properties were calculated using Mie theory and an assumed gamma distribution for the droplet radius. Finally, the vertical profile of CO_2 volume mixing ratio was assumed to be constant with height and fixed at 385 ppmv. Because the spectra are more sensitive to cloud than CO_2, and because CO_2 is approximately well-mixed in the atmosphere, this assumption has little impact upon the results. Examples of the spectra for the O_2 A-band (top), weak CO_2 band (middle) and strong CO_2 band (bottom) are shown in Fig. 4. Only part of an O_2 A-band doublet is covered, while the weak and strong CO_2 windows span only two lines.

The spectra are affected by many factors, the most important being the surface albedo and the vertical distribution of cloud. Generally, cloudy spectra are easily distinguished from clear, as illustrated in the left-hand panel of Fig. 5, where a cloudy O_2 A-band spectrum has been rescaled to match as closely as possible a clear spectrum, both with the same viewing geometry, surface pressure and surface albedo. Such is not always the case, however, as can be seen in the right-hand panel of the figure, which shows a case where a cloudy sky spectrum closely matches

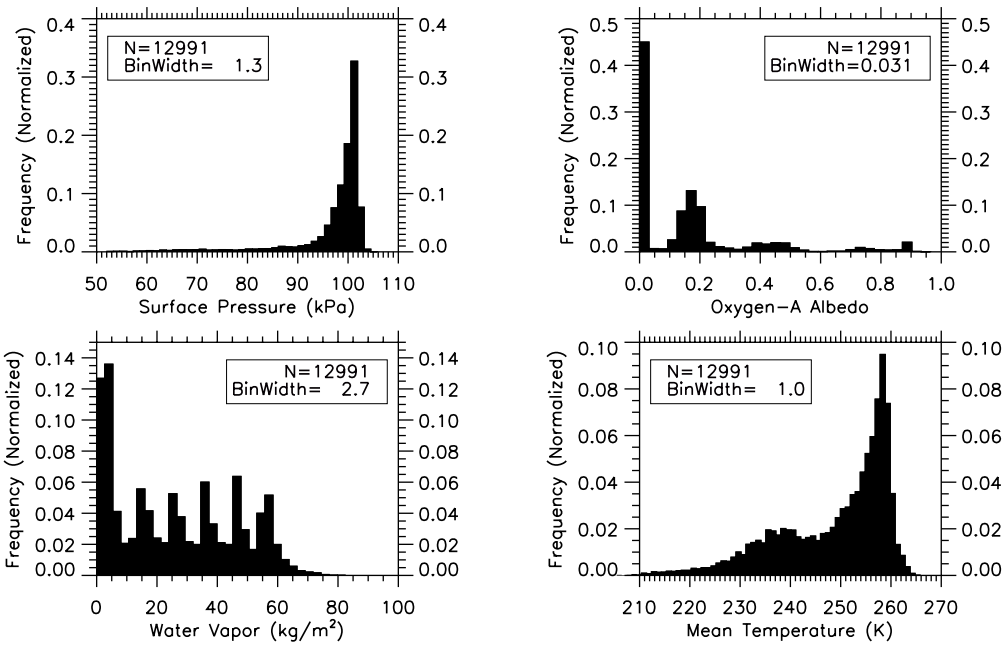

Fig. 3. Histograms of surface pressure (upper left), O_2 A-band surface albedo (upper right), column water vapor (lower left) and mean temperature (lower right) derived from the 12,991 ECMWF profiles.

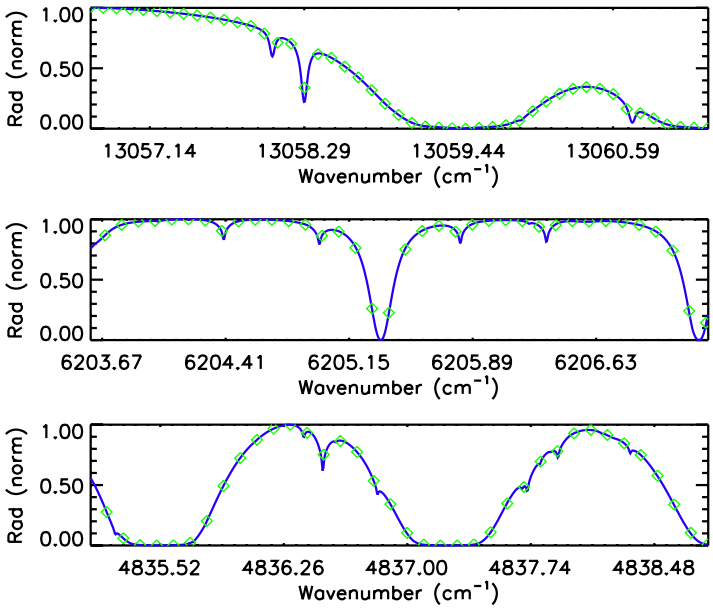

Fig. 4. Typical normalized spectra for the windows of the O_2 A-band (top), weak CO_2 band (middle) and strong CO_2 band (bottom). The monochromatic spectra are shown as solid lines, while the convolved spectra are shown as diamonds.

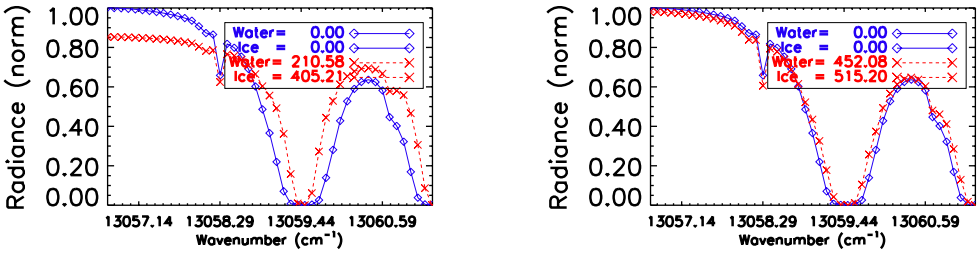

Fig. 5. Comparisons of a clear sky spectrum (blue) in the O_2 A-band to a cloudy sky spectrum (red) that has been scaled to fit. The scaled spectrum in the left-hand panel is clearly distinguishable from the clear sky case, while in the right-hand panel the two spectra are very similar even though the cloud water and ice amounts are larger.

a clear sky spectrum after scaling. Because the rescaling would be almost indistinguishable from a change in surface albedo for the clear atmosphere, such cases are challenging for the neural network algorithm, unless independent estimates are available for the surface reflectance. Therefore, in the analyses that follow, we assume that the surface properties can be prescribed with sufficient accuracy using data from sensors like MODIS. Equally, we assume that the surface pressure and the mean temperature of the atmospheric profile are available from reliable meteorological analyses.

3 NEURAL NETWORK CLOUD SCREENING ALGORITHM

Neural networks are a logical choice for cloud screening for two reasons. First, they can identify the subtle variations in reflected radiances caused by slight changes in the atmosphere. Second, once the off-line neural network training has been completed, the inversion process requires minimal computation, yielding fast results relative to typical forward model run times, an important feature when processing large data sets. It should be noted, however, that delays in processing may occur if the algorithm relies on ancillary data, such as atmospheric properties from ECMWF reanalysis or MODIS surface properties, as are used here.

This work employs neural networks with standard multi-layer perceptron architecture, trained via back-propagation of sensitivities [15]. Although many flavors of neural networks exist, common to the feed-forward architecture is preservation of the mapping in sets of weights, biases, transfer functions and other conditioning variables (also known as network parameters) that are determined during training by presenting to the neural network a series of matched predictor-target pairs. Once the training has been performed, the inversion of future inputs is then a matter of applying the network parameters to the set of inputs.

Examples of using neural networks to retrieve atmospheric variables from satellite measurements include estimation of surface wind speeds from SSMI measurements [16], retrieval of temperature and moisture profiles from Atmospheric InfraRed Sounder (AIRS) and Advanced Microware Sounding Unit (AMSU) data [17], as well as retrieval of temperature, water vapor and ozone from the Infrared Atmospheric Sounding Interferometer (IASI) [18] (applied to simulated data). Multi-layer perceptrons (MLP) have been applied to the problem of land classification with heuristic methods for determining the ideal number of layers, nodes and other *a priori* network parameters [19].

3.1 Network architecture

The basic architecture for a single layer of a feed-forward neural network is shown in Fig. 6. The input to layer m consists of a vector a_j^{m-1} of predictors that are combined linearly with

weights w_{ij}^m and biases b_i^m,

$$n_i^m = \sum_j w_{ij}^m a_j^{m-1} + b_i^m.$$

Each n_i^m is subjected to a transfer function to produce the layer output a_i^m,

$$a_i^m = f^m(n_i^m),$$

where m indicates layer number, i is the node index, j varies over the layer inputs and f represents the transfer function.

The transfer function in the hidden layer, used to scale n_i^m to values between 0 and 1, commonly takes the form of the log-sigmoid function,

$$f(x) = 1/(1 + \exp(-cx)),$$

or the hyperbolic tangent,

$$f(x) = \tanh(cx/2),$$

which differs only by a constant offset and a scale factor. In this work we chose the former, and set $c = 0.3$ on purely empirical grounds.

The architecture of a neural network begins with a set of predefined inputs, commonly called predictors, which are ultimately to be mapped into the output space. There are two primary criteria for selection of the predictors. Firstly, they should be the variables that capture the variability in the parameters that are to be retrieved and secondly, they should be variables that are readily available. For this work the predictors are the measured (or simulated) radiances, as well as the surface pressure, surface reflectance and the mean atmospheric temperature (described previously) that are obtained from ECMWF reanalysis. In a two-layer neural network the results from the first layer (the hidden layer) serves as the input to the second layer (the output layer), giving rise to the term 'feed-forward'. Normally a linear transfer function is used in the output layer, as was done in this work. The outputs from the second layer in a two-layer architecture are the estimates of the retrieval variables, often referred to as the targets. Generally the targets are simply defined by the problem at hand, in this case being cloud water, cloud ice, effective pressure heights of cloud water and ice, and column water vapor ($c_w, c_i, p_{cw}, p_{ci}, w$). To classify scenes as either cloudy or clear a simple threshold test is then performed on the estimated values of c_w and c_i as will be discussed in Sec. 4.1.

3.2 Network training

Fundamental to neural networks is the training method used to optimize the weights and biases that define the mapping between measurement and state space. This work uses back-propagation of sensitivities, which is a form of least mean square minimization via steepest descent, as it has been shown to be fast, robust and relatively simple [15]. Proper network behavior is presented to the algorithm in the form of matched predictor-target pairs, and the routine seeks the solution by following the negative of the gradient of the cost function, $\hat{F}$, defined to be the L^2-norm of the difference between the modeled data a_r^M from the final layer (indexed M) and the observed (target) data t_r,

$$\hat{F} = \sum_r [a_r^M - t_r]^2.$$

The gradient of the cost function with respect to the weights in layer m is

$$\frac{\partial \hat{F}}{\partial w_{ij}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \frac{\partial n_i^m}{\partial w_{ij}^m} = s_i^m a_j^{m-1},$$

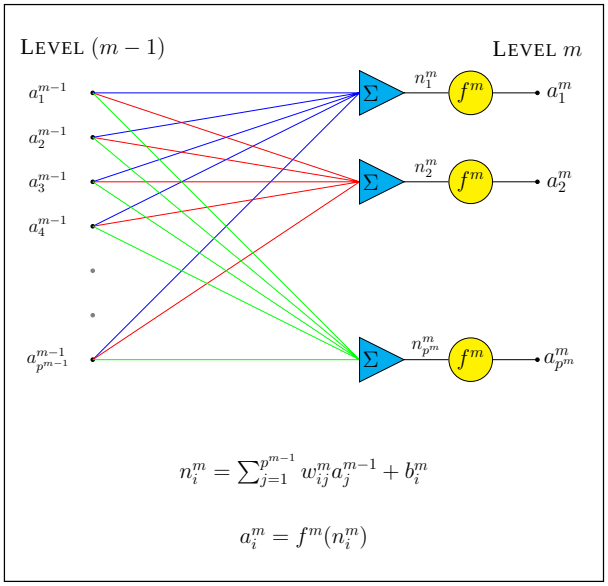

Fig. 6. Structure of one layer of a feed-forward neural network. The inputs a_j^{m-1} at the left are combined linearly with weights w_{ij}^m and biases b_i^m to produce n_i^m. Each n_i^m then is subjected to a non-linear transfer function f^m to produce the output a_i^m on the right. Here subscripts represent node number, and superscripts represent layer number.

where s_i^m denotes the sensitivity of the network error to a change in the i^{th} element of the input at layer m,

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m}.$$

Similarly, the gradient with respect to the biases in layer m is

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m}\frac{\partial n_i^m}{\partial b_i^m} = s_i^m.$$

In the case of a single layer, when there are no hidden nodes, the sensitivites are easily computed, but for a multi-layer network the calculation is slightly more complicated. However, the recurrence relation, derived from the chain rule of differentiation,

$$s_i^m = \sum_j \left(\frac{\partial \hat{F}}{\partial n_j^{m+1}}\right)\left(\frac{\partial n_j^{m+1}}{\partial n_i^m}\right) = \sum_j s_j^{m+1}\left(\frac{\partial n_j^{m+1}}{\partial n_i^m}\right),$$

shows that the sensitivities can be propagated backwards from the output layer. Furthermore, the derivative above is easily calculated, since

$$\frac{\partial n_j^{m+1}}{\partial n_i^m} = \sum_k \frac{\partial n_j^{m+1}}{\partial a_k^m}\frac{\partial a_k^m}{\partial n_i^m} = \frac{\partial n_j^{m+1}}{\partial a_i^m}\frac{\partial a_i^m}{\partial n_i^m} = w_{ji}^{m+1}\frac{\partial f^m(n_i^m)}{\partial n_i^m},$$

allowing the sensitivities to take the simple form

$$s_i^m = \frac{\partial f^m(n_i^m)}{\partial n_i^m}\sum_j s_j^{m+1} w_{ji}^{m+1}.$$

The recursion starts in the last layer M, where

$$s_i^M = 2(a_i^M - t_i)\frac{\partial a_i^M}{\partial n_i^M} = 2(a_i^M - t_i)\frac{\partial f^M(n_i^M)}{\partial n_i^M}.$$

Computationally the weights and biases are determined iteratively via

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha(k)\frac{\partial \hat{F}}{\partial w_{ij}^m(k)}, \tag{1}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha(k)\frac{\partial \hat{F}}{\partial b_i^m(k)}, \tag{2}$$

where k represents the iteration number and $\alpha(k)$ is the learning rate. Inserting the expressions for the partial derivatives of the cost function into Eqs. (1) and (2), we obtain

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha(k)\, s_i^m(k)\, a_j^{m-1}(k), \tag{3}$$

and

$$b_i^m(k+1) = b_i^m(k) - \alpha(k)\, s_i^m(k). \tag{4}$$

In practice, Eqs. (3) and (4) were replaced by

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \tanh(\alpha(k)\, s_i^m(k)\, a_j^{m-1}(k)) \tag{5}$$

and

$$b_i^m(k+1) = b_i^m(k) - \tanh(\alpha(k)\, s_i^m). \tag{6}$$

The modification significantly improves the stability of the training algorithm, because the hyperbolic tangent limits steps that are unrealistically large, as often occurs in the first few iterations. For small arguments,

$$\tanh(x) \approx x,$$

so there is no impact upon the final convergence. Note that the application of the hyperbolic tangent function is completely independent of the transfer function f that was described previously.

After each iteration of training, the new weights and biases are used to make updated predictions of the output parameters. Then the mean squared error (MSE) between the predicted and the target value is calculated, providing an indicator of improvement or degradation in the new set of weights and biases. Iterations proceed until preselected convergence criteria are met, as will be discussed in the following subsections. Several techniques designed to balance speed and accuracy are discussed extensively in the literature; see, for example, Refs. [19] and [20]. The following subsections discuss the implementation of these methods to achieve the best neural network performance.

3.2.1 Data division and normalization

When training neural networks it is standard practice to withhold a subset of the predictor-target data from the training session to be used to check the network predictive skill. In this work 75% of the simulated data was used for training, while the remaining 25% was split roughly in half to create selection and validation subsets. The selection set is used to pick the best of a series of trained neural networks, to be discussed in Sec. 3.3, while the validation set is used to quantify the predictive skill, as will be shown in Sec. 4.2. Also withheld from the training routine is a set called the hold-over data, whose composition and use will be discussed in more detail in Sec. 4.2.

After division of the data, the mean and standard deviation of each predictor were calculated from the training data set and used to normalize the predictors in order to provide numerical stability in the training routine. These normalization constants now become an inherent part of the algorithm and must be applied to any future predictors prior to performing a retrieval.

3.2.2 Shuffles

Once the training data set has been decided upon, the iterative method desribed by Eqs. (5) and (6) is used to determine the set of weights and biases. During one iteration all predictor-target pairs are presented to the neural network before the MSE is calculated. Note that the input is necessarily a rank two matrix with dimensions equal to the number of predictors and number of training cases. In an attempt to increase the generality of the neural network, the order in which the predictor-target pairs are presented changes for each iteration by scrambling the input matrix in a process called shuffling. Therefore, future references to the iteration number will be replaced with shuffle number.

3.2.3 Update of learning rate

As the algorithm searches state space for a solution that minimizes the MSE, the speed and accuracy of the convergence can be maximized by dynamically adjusting the learning rate, $\alpha(k)$. A performance increment, denoted $\gamma(k)$ and defined to be the ratio of the new MSE to the old MSE, is calculated after each shuffle, and then compared with a performance threshold γ_0, taken to be 1.04 in practice. If $\gamma(k) > \gamma_0$, the algorithm is penalized via a reduction in $\alpha(k)$,

$$\alpha(k) \rightarrow \alpha(k-1)\,\alpha_2, \quad \text{where} \quad \alpha_2 = 0.7,$$

and the newly calculated weights and biases are rejected in favor of the values from the previous shuffle. On the other hand when $\gamma(k) < \gamma_0$, the new weights and biases are retained and $\alpha(k)$ is increased,

$$\alpha(k) \rightarrow \alpha(k-1)\,\alpha_1, \quad \text{where} \quad \alpha_1 = 1.05.$$

3.2.4 Momentum coefficient

To reduce oscillations in the search trajectory and to help avoid the algorithm halting in local minima of the cost function, the second term on the right-hand side of Eq. (5) is rewritten as a linear combination of the gradient and the previous weights using the momentum coefficient, $\epsilon(k)$, as follows:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - [(1-\epsilon(k))\tanh(\alpha(k)\,s_i^m(k)\,a_j^{m-1}(k)) + \epsilon(k)(w_{i,j}^m(k) - w_{i,j}^m(k-1))]. \tag{7}$$

The second term in the bracket on the right-hand side of Eq. (7) is the momentum term, which serves as a low pass filter, damping out oscillations in the search trajectory. The value of $\epsilon(k)$ is updated after each training shuffle based on the MSE performance in a way similar to adjustments in the learning rate. That is, $\epsilon(k)$ is increased when $\gamma(k) < \gamma_0$, but is decreased when the new MSE is more than 4% larger than the MSE from the previous shuffle (when $\gamma(k) > \gamma_0$). However, the penalty when $\gamma(k) > \gamma_0$ is more severe for $\epsilon(k)$ than it was for $\alpha(k)$; $\epsilon(k)$ is reset to zero, in which case Eq. (7) reverts to the form of Eq. (5), since the momentum term has effectively been removed.

Table 2. Convergence criteria used for training of the neural networks.

Variable	Purpose	Value
Minimum shuffles	Minimum number of training shuffles	100
Maximum shuffles	Maximum number of training shuffles	2000
`NumRunMean`	Number of shuffles to calculate mean MSE	5
`MaxErrorUp`	Maximum number of shuffles for which the MSE ratio is allowed to increase	8
α_0	Initial learning rate	0.2
α_1	Increase in learning rate	1.05
α_2	Decrease in learning rate	0.7
γ_0	Performance threshold (tolerated increase in MSE)	1.04
$\gamma(k)$	Performance increment ($\mathrm{MSE}_{new}/\mathrm{MSE}_{old}$)	Varies at each shuffle
ε_0	Initial momentum coefficient	0.5
$\varepsilon(k)$	Calculated momentum coefficient	Varies from 0.0 to 1.0 at each shuffle

3.2.5 Early stopping

The predictor-target combinations that are presented to a neural network during the training routine are meant to be examples of the ideal relationship between input and output space. However, it is important to avoid lack of generalization, or over-training, such that the network simply memorizes the training data set. A straightforward method for identifying the onset of this condition is to monitor the difference between the MSE of the training and validation data sets. Although fluctuations in MSE are expected as the algorithm iterates towards a solution, the running mean value of the MSE for both training and validation data sets should continue to decrease. The major symptom of over-training is a continued decrease in the training data MSE with simultaneous increase in the validation data MSE. When this occurs the training session should be halted, a technique referred to as early-stopping.

The basic criteria for halting training is to set both a minimum and maximum allowed number of shuffles, the values of which are given in Table 2. Two additional training criteria were introduced to provide early-stopping. First, the MSE is averaged over `NumRunMean` shuffles to produce a running mean, which helps to filter noise in the MSE from one shuffle to the next. The second variable for early stopping, `MaxErrorUp`, is the maximum number of shuffles that the ratio of the running mean of the MSE of the validation and training data ($\mathcal{R} = \overline{\mathrm{MSE}}_{val}/\overline{\mathrm{MSE}}_{trn}$) is allowed to increase. After each shuffle $\overline{\mathrm{MSE}}_{val}$ and $\overline{\mathrm{MSE}}_{trn}$ are updated and $\mathcal{R}$ is calculated. If $\mathcal{R}$ continues to grow sequentially for a number of shuffles equal to `MaxErrorUp`, then the training is halted, and the set of weights and biases from the shuffle with the lowest training MSE are retained. Note that the minimum number of shuffles must be achieved before training is allowed to stop, regardless of the value of $\mathcal{R}$. Some experimentation with the parameters listed in Table 2 was performed and the neural networks were found to be fairly insensitive to minor changes in the values, while major deviations from these recommended values yielded inferior results.

3.2.6 Determining the number of nodes

As discussed previously, the function of the neural network training session is to determine the optimum set of weights and biases by minimizing the mean squared error (MSE) between the predicted and the target value of each variable. In theory the combination of multiple layers of

Table 3. Results from neural network training with a varying number of nodes in the hidden layer. Results are given for an average of fifteen trained networks in each case.

Number of nodes	MSE_{trn}	Number of shuffles	Time per shuffle [sec]	Run Time [sec (min)]
2	371.8	344	0.5	166 (3)
5	290.7	590	1.6	916 (15)
10	257.5	518	4.5	2344 (39)
20	205.3	738	14.9	11,029 (184)
30	198.4	452	30.4	13,728 (229)
40	192.8	681	53.3	36,263 (604)

weights and biases with the associated transfer functions allows for infinitely complicated relations between the measurement and state space [15], although in practice there are limitations due to imperfect training data sets, measurement noise and other complications.

In a two-layer network the matrix of weights and vector of biases in the first (hidden) layer have the dimensions

$$W^1 = [P, N] \qquad \text{and} \qquad b^1 = [N],$$

where P is the number of predictors and N is the number of nodes in the hidden layer. For the second (output) layer the dimensions are

$$W^2 = [N, T] \qquad \text{and} \qquad b^2 = [T],$$

where T is the number of target variables. Therefore, the total numbers of weights and biases for the two-layer model are

$$W_{total} = NP + NT = N(P + T) \qquad \text{and} \qquad b_{total} = N + T,$$

respectively.

Although some rules of thumb exist for determining N based loosely on the independent pieces of information in P and T, often an empirical tack is employed. Table 3 provides a summary from training sessions performed on the same set of predictor-target data, but with the number of nodes in the hidden layer varying between 2 and 40. The data for each case is an average of fifteen training runs, i.e., committee members (to be discussed in Sec. 3.3). It is evident that as the number of nodes is increased, thereby allowing for more elaborate mappings between measurement and state space, the MSE in the training data is reduced, as should be expected. However, there is not a clear correlation between an increase in the number of nodes and a reduction in the number of shuffles required to satisfy the stopping criteria of the training process. It is obvious though that the time required per shuffle increases steadily as nodes are added to the hidden layer, and the total training time increases substantially.

While the MSE is the sum of the squared differences between the predicted and target values for all the retrieval variables ($c_w, c_i, p_{cw}, p_{ci}, w$), it is instructive to examine a few cases to see the effect on the retrieval of cloud water and cloud ice. Shown in Table 4 are values of c_w and c_i predicted by networks with different numbers of nodes for a clear sky case, a moderately cloudy case and a thick cloud case. As can be seen from the top section of the table, and as will be demonstrated in Sec. 4, the neural networks have very little skill in estimating cloud amounts for clear sky cases. In general the predicted value is always an overestimate of the target value. For the moderately thick cloud case (middle section of Table 4), the differences between the predicted and target value for cloud water are minimized by the neural network with twenty nodes, while forty nodes provided a minimum difference for cloud ice. However,

Table 4. Predicted values of cloud water and ice from neural network training with a varying number of nodes in the hidden layer. A clear sky case is shown at the top, a moderately thick cloud case in the middle, and a thick cloud case at the bottom.

Number of nodes	Target c_w	Predicted c_w	(P-T)/T (%) c_w	Target c_i	Predicted c_i	(P-T)/T (%) c_i
2	0.0	99.0	–	0.0	48.7	–
5	0.0	23.5	–	0.0	15.0	–
10	0.0	21.1	–	0.0	-5.5	–
20	0.0	7.4	–	0.0	-8.3	–
30	0.0	-18.3	–	0.0	-6.7	–
40	0.0	20.2	–	0.0	-2.4	–
Number of nodes	Target c_w	Predicted c_w	(P-T)/T (%) c_w	Target c_i	Predicted c_i	(P-T)/T (%) c_i
2	73.8	105.7	43.2	12.8	52.3	309
5	73.8	86.8	17.6	12.8	31.1	143
10	73.8	84.1	14.0	12.8	20.8	62.5
20	73,8	75.8	2.7	12.8	16.9	32.0
30	73.8	86.2	16.8	12.8	18.0	40.6
40	73.8	82.8	12.2	12.8	15.2	18.8
Number of nodes	Target c_w	Predicted c_w	(P-T)/T (%) c_w	Target c_i	Predicted c_i	(P-T)/T (%) c_i
2	241.2	226.9	6.3	103.8	78.9	-24.0
5	241.2	150.3	-37.7	103.8	87.6	-15.6
10	241.2	153.5	-36.4	103.8	113.4	9.2
20	241.2	154.7	-35.9	103.8	108.6	4.6
30	241.2	187.1	-22.4	103.8	107.8	3.9
40	241.2	186.4	-22.7	103.8	100.1	-4.2

for the thick cloud water case (bottom section of Table 4), there is little difference in the predictions between the five, ten and twenty node networks, while the thirty and forty node networks provide marginally better results. For the thick cloud ice case the thirty node network again provides the most accurate result. Although these results suggest that more nodes in the hidden layer yields higher retrieval accuracy, for the developmental work we have selected ten nodes as a compromise for its faster training time. For an operational code, probably more nodes would be an advantage.

3.2.7 Summary of neural network training

Once the overall network architecture has been decided, the number of hidden nodes chosen, and training has been halted, either by the early stopping criteria or by reaching the maximum number of allowed shuffles, the final set of weights and biases are used to predict the target parameters. These predictions are regressed against the true target values. The slope and bias of the linear fit for each trained neural network is retained, and used later to adjust retrieved values, as will be discussed in Sec. 3.3.

Shown in Fig. 7 are results of the mean squared error for a typical training session after implementation of the techniques discussed above. The abscissa indicates the number of iterations, or shuffles, that occurred before training criteria were satisfied. The left panel shows results for

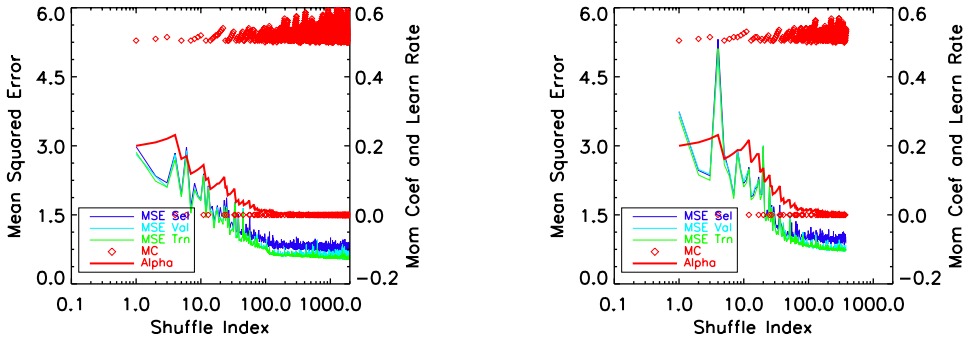

Fig. 7. Each panel shows neural network mean squared errors (MSE) against the left ordinate for the selection (thin blue), validation (thin teal) and training (thin green) data sets as functions of training shuffle number for a single training case. Shown against the right ordinate are the corresponding values of the momentum coefficient, $\varepsilon(k)$ (red diamonds), and the learning rate, $\alpha(k)$ (thick red). The left panel is for a case that required about 2000 shuffles to satisfy the stopping criteria, while the right panel shows a case that stopped after just over 300 shuffles. The only difference between the two cases is the initial set of weights and biases used to seed the neural network. Note the occasional reset of the momentum coefficient to zero due to violation of the γ_0 performance threshold.

a case that required about 2000 shuffles to satisfy the stopping criteria, while the case shown in the right panel stopped after only 300 shuffles. Note that the only difference between these two cases is the set of initial weights and biases used to seed the algorithm, as will be discussed in the next section. It is clear that the MSE in each of the training, selection and validation data sets converges asymptotically as the algorithm iterates, suggesting that further shuffles would provide only insignificant reductions in the MSE. Also shown in Fig. 7 are the values of the learning rate, $\alpha(k)$, plotted as the thick red line against the right ordinate. The decrease of this value towards zero is desirable behavior, as it indicates that the algorithm is taking smaller and smaller steps in solution space as iterations proceed, thus increasing the accuracy of the final solution. The momentum coefficient, $\varepsilon(k)$, generally has values between 0.5 (the initial setting) and 0.6, but occasionally is reset to zero when the performance increment is violated.

3.3 Consulting a committee of neural networks

As with any minimization problem, it is difficult to distinguish between local and global minima. A convenient way to deter the algorithm from converging on solutions that correspond to local minima is to perform multiple searches, each beginning from a randomly selected point in state space. To this effect, a congress of fifteen neural networks are trained, each using a different random set of initial weights and biases.

After a training session has been performed on a congress of neural networks and all of the weights, biases and other necessary conditioning variables stored, predictions of the target parameters can be made from a set of predictors. However, it is critical that the predictors be conditioned in the same way as for the training session, including normalization by the mean and standard deviation obtained from the training data prior to training (see Sec. 3.2.1). After predictions have been made by each congress member, the correction from the linear regression mentioned in Sec. 3.2.7 is applied to each, a committee of the best five neural networks is selected, based on the members that return the lowest MSE for the selection subset defined in Sec. 3.2.1. The final prediction for each target is the average of the predictions by the selected committee. This process helps minimize random errors in the predictions.

3.4 Sensitivity to predictors and measurement noise

In addition to the spectra, the neural networks require estimates of the surface pressure, the surface albedo and the mean of the atmospheric temperature profile. As the latter variables will be obtained from meteorological analyses and satellite observations, they will be subject to error, so it is important to test the impact of such errors upon the predictions of the neural networks. To this end, random errors were drawn from gaussian distributions with mean zero and standard deviations equal to 2%, 10% and 2% of the estimated values of surface pressure, surface reflectance and mean temperature, and then were added to those variables.

Another source of retrieval error that requires quantification is the random instrument noise, which for the hypothetical etalon spectrometer we assume to be dominated by photon noise. Thus, the signal-to-noise ratio has the simple form

$$\mathrm{SNR} = \mathrm{SNR}_0 \sqrt{I/I_0},$$

where SNR_0 is the signal-to-noise ratio at the reference radiance I_0. We assume the latter to be the radiance reflected in a clear, non-absorbing atmosphere from a surface with unit albedo when the sun is at 45° zenith angle. For the corresponding SNR_0, we assume a modest value of 100. Noise from a gaussian distribution with mean zero and standard deviation

$$N = \sqrt{II_0}/\mathrm{SNR}_0$$

is added to the radiance spectra. The noise for each spectral sample is assumed to be independent of all other samples.

The results from the sensitivity testing described here will be presented in Secs. 4.2.3 and 4.3, where it is shown that the neural network algorithm is robust to variations in all prediction variables, including instrument noise.

4 ANALYSIS OF NEURAL NETWORK PREDICTIONS

In this section we describe experiments of increasing complexity to test the skill of the neural network in classifying clear and cloudy scenes. First, a single neural network is used to illustrate general features of the results. Next the concepts of neural network committees and hierarchies are described, and it is shown how these improve the retrieval of cloud properties. These techniques then are applied to an independent data set to show how the algorithm performs under more realistic conditions.

4.1 Predictions using a single neural network

First a single neural network was trained using the techniques described in Sec. 3.2, and then predictions were made using the validation subset as input. The true and predicted values of cloud water and ice are shown as a scatter plot in Fig. 8. The horizontal and vertical bisection lines drawn at $2\,\mathrm{gm}^{-2}$ for cloud water and at $10\,\mathrm{gm}^{-2}$ for cloud ice, represent thresholds for clear and cloudy scenes.

These thresholds were derived from the expression relating cloud optical thickness, τ, and column water, W (either liquid or solid in units kgm^{-2}), in the visible and near infrared;

$$\tau = \frac{3}{2}\frac{W}{\rho r_e},$$

where ρ is the density of liquid water or solid ice in units kgm^{-3} and r_e denotes the effective radius in units meters [21]. Effective radii of 10 μm and 50 μm were assumed for water and ice particles, respectively, as well as an optical thickness of 0.3, the suggested cutoff for retrievals of X_{CO_2} from measurements in the near-infrared [2].

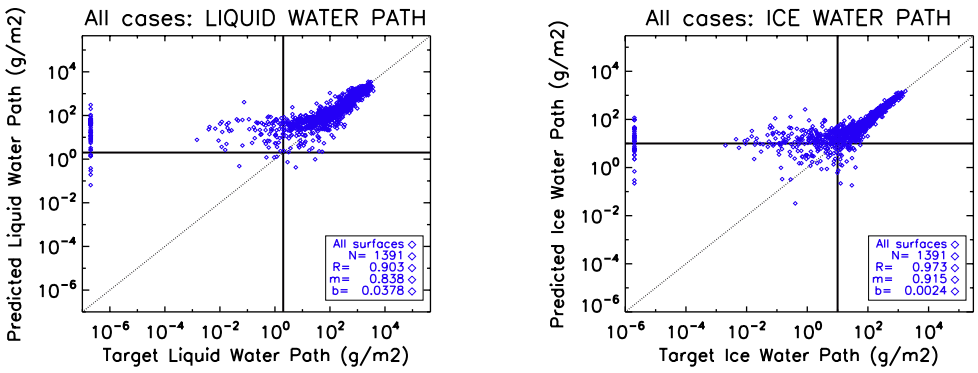

Fig. 8. Predicted versus target values of cloud water (left) and cloud ice (right) determined by a single neural network on the validation data set that has not been truncated into hierarchy ranges, nor stratified by surface type. Note the logarithmic scale of both axes. The linear regression coefficient (R), the slope (m) and the bias (b) are given in the legend, along with the number of data points (N). The horizontal and vertical bisection lines indicate appropriate thresholds for distinguishing cloudy scenes from clear.

For both of the plots shown in Fig. 8, the correlation between the predicted and the target values improves with increasing cloud amount. Certainly for points with target cloud amounts below the thresholds, the predicted value has almost zero correlation as well as high deviation, while for large cloud amounts the correlation is tight. For scenes with no cloud water,[†] the predicted values span a wide range of values.

4.2 Predictions using neural network committees and hierarchies

The predictions in Fig. 8 were made using neural networks trained with data spanning the full breadth of cloud water and ice (approximately 0–5000 $\mathrm{gm}^{-2}$and 0–2000 gm^{-2}). The issue of decreasing accuracy with decreasing optical thickness is presumably due to the difficulty in identifying small changes in signal that occur in thin cloud versus the much larger changes that occur when the optical thickness is large.

One technique for overcoming this difficulty is to train a series of neural networks, each using training data that spans a limited range of cloud water. The distinct neural networks, hereafter referred to as hierarchy members, can be thought of as specialists in their range of expertise. Experimentation indicated that selecting four training ranges, listed in Table 5, provided enough resolution to solve the aforementioned problem, while still providing enough training spectra for each hierarchy from the original set of 12,991 spectra.

Prior to truncation of the training data, the hold-over data set, mentioned in Sec. 3.2.1, is withheld from the full synthetic data set. This produces a collection of predictor-target pairs including profiles with cloud water and ice spanning approximately the full range of values found in the training data set. This hold-over set is used to test the retrieval skill in situations where the correct hierarchy member is not known *a priori*, as occurs when processing real soundings.

4.2.1 Stratifying scenes by IGBP surface type

Although the sensitivity study described in Sec. 3.4 indicated that the neural network retrievals are insensitive to variations of order 10% in surface reflectance, due to large changes in reflectance over natural surfaces, it was found that neural network predictions were improved if

[†]A small offset has been added to allow identical zeros to be plotted on a logarithmic scale.

Table 5. Thresholds of M, the combined cloud liquid and ice water in units gm^{-2}, used to build neural network hierarchy members.

Hierarchy index	Training range (gm^{-2})
H0	$0 \leq M \leq 1$
H1	$1 < M \leq 40$
H2	$40 < M \leq 300$
H3	$300 < M$

scenes were stratified according to the International Geosphere-Biosphere Programme (IGBP) surface type. Four surface categories were distinguished:

(1) permanent snow and ice (IGBP=15);

(2) barren or sparsely vegetated (IGBP=16);

(3) water (IGBP=17);

(4) all other surface types.

Although not strictly correct terminology, these categories are referred to as ice, desert, ocean and land in the remainder of this document for convenience. For brevity, only results over land are reported in the work presented here, the performance for other surface categories being similar. In Part II of the manuscript the results will be extended to include all surface types.

4.2.2 Predictions on the validation data set

After the 12,991 ECMWF profiles were split according to the four surface types and then into the four hierarchy regimes based on cloud amount, the subsets of synthetic spectra were used to train the corresponding sixteen congresses of neural networks. The results from the validation subsets are shown in the left and right panels of Fig. 9 in the form of scatter plots of the predicted values of cloud amount versus the target values from the four hierarchy members. In these plots each hierarchy member is plotted in a different color. Although the accuracy of the retrieved cloud amount is still proportional to the cloud water, the correlation is much tighter than the results shown in Fig. 8 when no hierarchy was used and no stratification by surface type was made.

Having provided four hierarchy members, each with a given range of expertise, a method for selecting the most reliable prediction from a set of measurements with unknown cloud amount is needed. The tight correlation with the hierarchy member trained using cloud water greater than $300\,\mathrm{gm}^{-2}$ (H3 in Table 5) suggests the following decision tree. First make a prediction using the largest hierarchy member (H3). If the predicted value falls within the range of expertise of H3, then retain the predicted value. Otherwise, make a second prediction using the next hierarchy member (H2) trained using cloud between $40\,\mathrm{gm}^{-2}$ and $300\,\mathrm{gm}^{-2}$. If the predicted value falls within the range of expertise of H2, then retain the predicted value. Repeat for H1 and H0, retaining the appropriate prediction. This method is referred to hereafter as the 'trickle-down' technique. For cases when none of the hierarchy members return a predicted value within its range of expertise, the scene is flagged and a dummy value (-999.9) is returned. For the retrievals on the entire independent data set discussed below, only 78 of 12,991(0.5%) of the scenes were indeterminate, and none occurred over land.

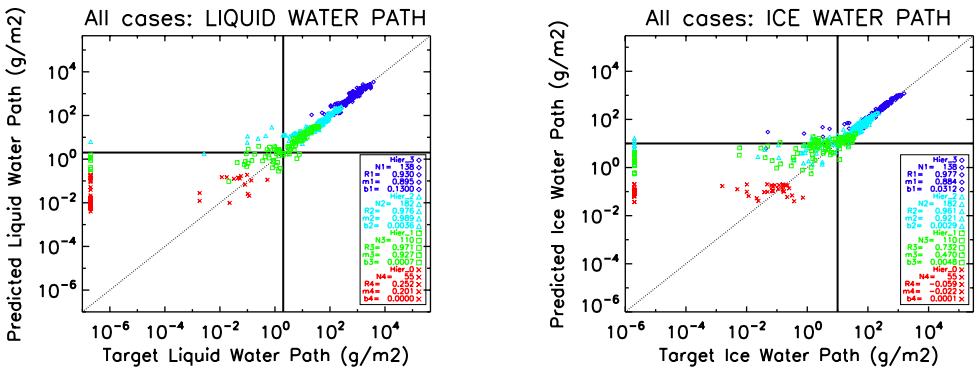

Fig. 9. Predicted vs target values of cloud water (left) and cloud ice (right) for four individual hierarchy members on the validation data set using a committee of five neural networks. The linear regression coefficient (R), the slope (m) and the bias (b) are given in the legends, along with the number of data points (N). Results are for measurements over land surfaces only.

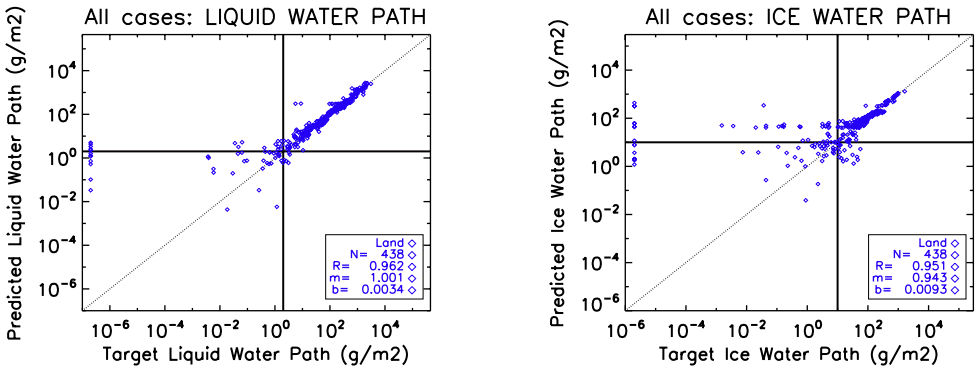

Fig. 10. Predicted vs target values of cloud water (left) and cloud ice (right) on the hold-over data set using the trickle-down strategy with a four member hierarchy, each consisting of a five member neural network committee. The linear regression coefficient (R), the slope (m) and the bias (b) are given in the legends, along with the number of data points (N). Results are for land surfaces only. Corresponding contingency table results are summarized in Table 6.

4.2.3 Predictions on the hold-over data set

Shown in Fig. 10 are predictions for cloud water (left) and cloud ice (right) on the hold-over data set for scenes over land surfaces. It is evident that the trickle-down technique provides a good retrieval for cloud water, especially when the water exceeds the $2\,\mathrm{gm}^{-2}$ threshold. In the case of cloud ice there are horizontal striations, indicating that the wrong hierarchy member was selected for the final prediction, generally leading to an overestimate of the true value. Overall the classification skill is reasonable. For both cloud water and cloud ice, the linear correlation coefficient is high ($R = 0.96$ and $R = 0.95$, respectively), while the slope is at or near one ($m = 1.00$ and $m = 0.94$, respectively) and the bias is small ($b = 3.4\,\mathrm{gm}^{-2}$ and $b = 9.3\,\mathrm{gm}^{-2}$, respectively) relative to the range of values.

Although the regression analysis provides a useful metric of accuracy, it is desirable to quantify the skill of the clear and cloudy classifications statistically. This is done through a contingency table, which bins each retrieval into one of four logical categories, true positive (*TP*), true negative (*TN*), false positive (*FP*) or false negative (*FN*), where true (*T*) refers to

Table 6. Contingency tables for classifications on the hold-over data set (corresponding to Fig. 10) for cloud water and ice. The results are for land surfaces, excluding IGBP types 15, 16 and 17.

Variable	Clear atmospheres					Cloudy atmospheres					*ACC*	*PPV*
	Total cases	Predicted clear		Predicted cloudy		Total cases	Predicted clear		Predicted cloudy			
		#	*TPR*	#	*FNR*		#	*FPR*	#	*TNR*		
Water	110	94	86%	16	14%	328	8	2%	320	98%	95%	92%
Ice	172	135	79%	37	21%	266	42	16%	224	84%	82%	76%

correct and false (*F*) to incorrect classification, while positive (*P*) refers to clear and negative (*N*) to cloudy conditions, based on the $2\,\mathrm{gm}^{-2}$ and $10\,\mathrm{gm}^{-2}$ thresholds introduced above. For a perfect algorithm, all cases will be correctly classified into the two diagonal elements of the contingency table ($TP + TN = 100\%$), but in practice the non-zero off-diagonal elements (*FP* and *FN*) indicate the frequency of error in the classification. The performance is summarized by the true positive rate $TPR = TP/P$, the false negative rate $FNR = FN/P$, the false positive rate $FPR = FP/N$ and the true negative rate $TNR = TN/N$.

Several useful diagnostic statistics can be calculated from the contingency table values. Here we report the value of the accuracy,

$$ACC = (TP + TN)/(P + N),$$

which represents the fraction of all scenes that were correctly classified, and the positive predictive value,

$$PPV = TP/(TP + FP),$$

which gives the confidence level that scenes reported as clear are actually clear. Since the most egregious behavior of the cloud screening is passing cloudy scenes as clear if the subsequent numerical cost of analyzing scenes is high, it is best to maximize the number of *TP* while simultaneously minimizing the number of *FP*. In general this result also leads to high values of the *PPV*. Note however, that a subtlety of the *PPV* is that it can be weighted heavily by either the *TP* or the *FP*, depending on the relative number of clear to cloudy scenes contained in the data set. For example, if there are ten times more clear scenes than cloudy ones, then it is likely that a classification algorithm that has only modest skill still will produce very high values of *PPV*. In general it is important that the training data set contains profiles covering the full range of realistic scenarios, while the independent data set should contain realistic frequencies of occurence of cloudy versus clear scenes.

The contingency table corresponding to Fig. 10 for cloud water and cloud ice is presented in Table 6. The relatively high values of *TPR*, *ACC* and *PPV*, along with the low value of *FPR*, indicate good classification skill, although the results are significantly better for cloud water than for cloud ice, a result that persists through all data sets.

Next, the sensitivity tests described in Sec. 3.4 were performed on the hold-over data set, first by adding noise to the variables separately and then simultaneously. Contingency tables were calculated for each scenario, with results summarized in Tables 7 and 8. Each of the individual perturbations has only a small effect on the statistics. Comparison of the results for the combined uncertainties with the results made with no uncertainties added (Table 6) shows that the degradation in skill is slight, with a decrease in accuracy and positive predictive value of about 2% for both cloud water and ice. These results show that the neural network algorithm is robust to realistic uncertainties in the predictors. The results presented in the following section

Table 7. Contingency tables for the sensitivity tests performed on the hold-over data set for cloud water. The results are for land surfaces, excluding IGBP types 15, 16 and 17.

Perturbation	Clear atmospheres					Cloudy atmospheres					*ACC*	*PPV*
	Total cases	Predicted clear		Predicted cloudy		Total cases	Predicted clear		Predicted cloudy			
		#	*TPR*	#	*FNR*		#	*FPR*	#	*TNR*		
Mean temp	110	90	82%	20	18%	328	8	2%	320	98%	94%	92%
Surf pres	110	94	86%	16	15%	328	8	2%	320	98%	95%	92%
Surf Refl	110	94	86%	16	15%	328	11	3%	317	97%	94%	90%
Inst noise	110	93	85%	17	15%	328	9	3%	319	97%	94%	91%
Combined	110	90	82%	20	18%	328	10	3%	318	97%	93%	90%

Table 8. Contingency tables for the sensitivity tests performed on the hold-over data set for cloud ice. The results are for land surfaces, excluding IGBP types 15, 16 and 17.

Perturbation	Clear atmospheres					Cloudy atmospheres					*ACC*	*PPV*
	Total cases	Predicted clear		Predicted cloudy		Total cases	Predicted clear		Predicted cloudy			
		#	*TPR*	#	*FNR*		#	*FPR*	#	*TNR*		
Mean temp	172	137	80%	35	20%	266	44	17%	222	83%	82%	76%
Surf pres	172	136	79%	36	21%	266	42	16%	224	84%	82%	76%
Surf refl	172	135	79%	37	21%	266	44	17%	222	83%	82%	75%
Inst noise	172	134	78%	38	22%	266	43	16%	223	84%	82%	76%
Combined	172	132	77%	40	23%	266	47	18%	219	82%	80%	74%

on the independent data set were performed with these uncertainties added to simulate real world conditions.

4.3 Predictions on independent data set

The tests thus far have assumed perfect simulations of spectra, whereas any numerical model can provide only an approximation to reality. To assess the impact of errors in the simulations, we generated an independent set of spectra that used CloudSat observations of clouds, coupled with meteorological analyses from ECMWF interpolated in time and space to the vertical grid used by CloudSat (typically with 125 layers). The aerosol and surface models were similar to those used for training, namely, a two-layer model of aerosol, with continental and oceanic types [22], and polarized, non-isotropic reflection, with parameters derived from MODIS and POLDER. A more thorough description of model implementation can be found in [23].

Scatter plots of the predictions of cloud water and ice are presented in Fig. 11 for land surfaces. Comparison of the results with and without uncertainties (blue diamonds and teal triangles) indicates only minor differences in the retrieved values, again supporting the robustness of the algorithm. Overall, the accuracy of the retrievals compares well to results from the hold-over data set (Fig. 10), although there is a much higher percentage of profiles with true cloud amounts below the threshold level, i.e., clear cases. For cloud water (left panel) there is a tight correlation for high cloud amounts, while the values below the clear-cloudy threshold are essentially uncorrelated and have high variance. Again, the horizontal striations in the retrieved values are caused by incorrect assignment of the hierarchy member to make the final prediction,

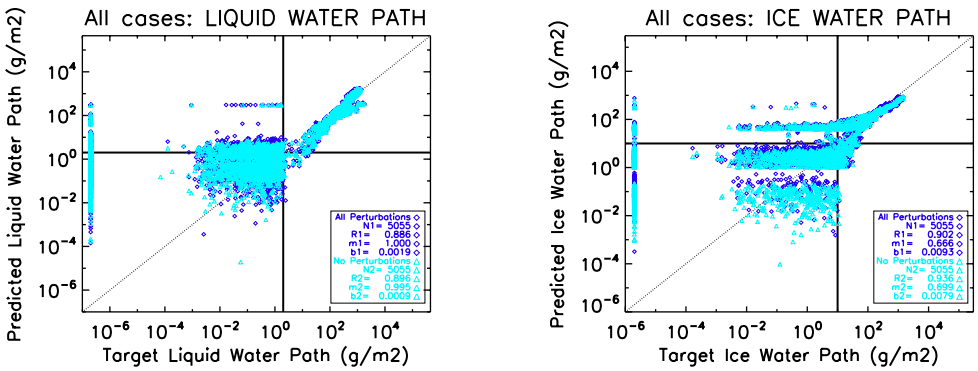

Fig. 11. Predicted versus target values of cloud water (left) and cloud ice (right) on the independent data set using the trickle-down technique with a four member hierarchy, each consisting of a five member neural network committee. The linear regression coefficient (R), the slope (m) and the bias (b) are given in the legends, along with the number of data points (N). The results are for measurements over land surfaces. The corresponding contingency tables are shown in Table 9.

Table 9. Contingency tables for retrievals on the independent data set for cloud water (top) and cloud ice (bottom) corresponding to the data presented in Fig. 11. The results for all land surfaces.

Variable/ error	Clear atmospheres					Cloudy atmospheres					*ACC*	*PPV*
	Total cases	Predicted clear		Predicted cloudy		Total cases	Predicted clear		Predicted cloudy			
		#	*TPR*	#	*FNR*		#	*FPR*	#	*TNR*		
Water/none	4422	3356	76%	1066	24%	633	22	4%	611	96%	79%	99%
Water/all	4422	3157	71%	1265	29%	633	19	3%	614	97%	75%	99%
Ice/none	4117	3770	92%	347	8%	938	189	20%	749	80%	89%	95%
Ice/all	4117	3736	91%	381	9%	938	193	21%	745	79%	89%	95%

a persistent problem with the trickle-down technique. Also of concern is the large variation in the predicted values when the target cloud amount is identically zero.[‡] These issues are even more apparent in the cloud ice retrievals (right panel).

Despite these deficiencies, the neural network still provides accurate classifications of scenes as either clear or cloudy. The contingency tables presented in Table 9 show that the accuracy and positive predictive value remain high, the latter with values of 99% and 95% for cloud water and ice, respectively, even with uncertainties added. Thus, we may have high confidence that scenes predicted as clear are in fact clear at the $2\,\mathrm{gm}^{-2}$ and $10\,\mathrm{gm}^{-2}$ thresholds. Furthermore, the relatively high true positive rates (71% and 91%) mean that the algorithm is not simply screening out all of the scenes, as might be done if the clear-cloudy thresholds were set unrealistically low.

Although not essential parameters for identifying clear scenes, the remaining three parameters predicted by the neural network might be useful to more sophisticated retrieval algorithms, such as that described in Ref. [24], for determining X_{CO_2} from high resolution spectral mea-

[‡]A small offset has been added to allow plotting on the logarithmic scale.

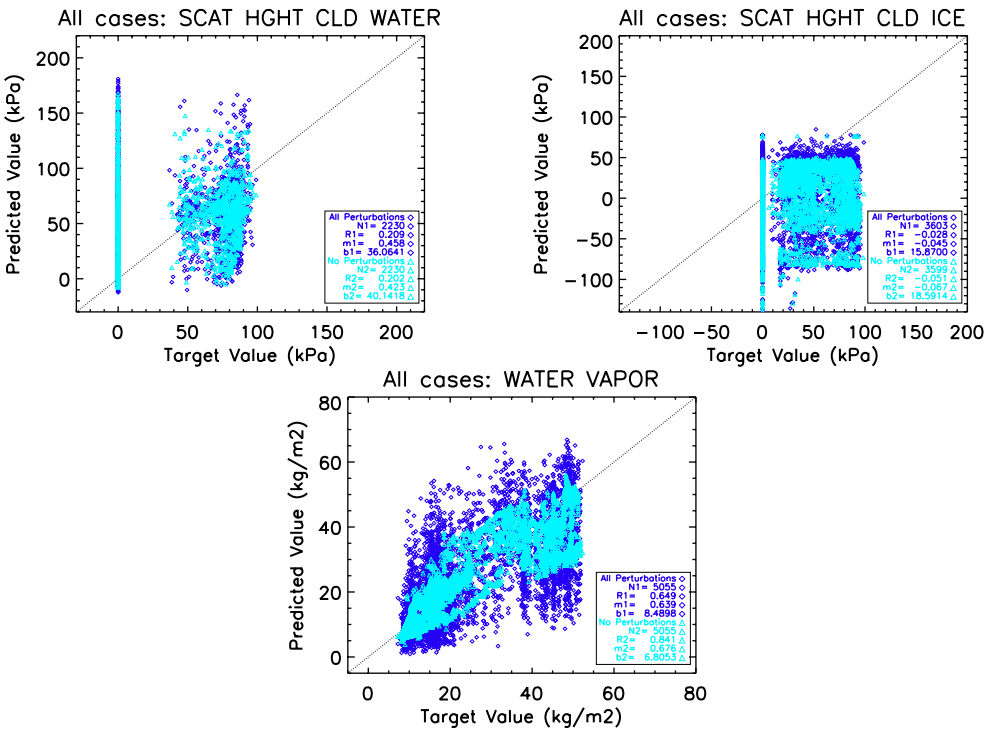

Fig. 12. Neural network predictions of the effective height of cloud water (left), effective height of cloud ice (right) and column water vapor (lower), as determined by the four member hierarchy using the trickle-down method on the independent data set.

surements in the near infrared. Shown in Fig. 12 are scatter plots for the effective heights of cloud water (left) and ice (right), and also column water vapor (lower). Overall the variance in these variables is high, and they are more susceptible to uncertainties than cloud water and ice. The retrievals of cloud water and ice are not significantly affected by removing these additional parameters from the neural network algorithm, although it does reduce the size of the neural network weight and bias matrices, thereby decreasing the training time and storage requirements. Although future training sessions might benefit from removal of these variables, since the number of nodes in the hidden layer could be increased to offset the time savings, they have been retained in the current exploratory version of the algorithm. It is hoped that future improvements will allow more accurate retrieval of these parameters.

5 CONCLUSIONS

A methodology using neural networks has been developed for classifying scenes as either clear or cloudy using high resolution spectra in micro-windows of the near infrared absorption bands of O_2 and CO_2. The neural networks have a two-layer, feed-forward architecture. Training of the neural networks was performed using predictor-target pairs synthesized with a multiple-scattering radiative transfer code using sixty-layer analyzed meteorology profiles from ECMWF and surface properties from MODIS. Committees of neural networks were trained over four cloud water ranges and four distinct IGBP surface types to provide better accuracy of the predictions. A methodology was developed to select the hierarchy member to make the final prediction.

Increasingly complicated experiments were conducted to show that the algorithm provides

satisfactory skill in distinguishing cloudy from clear scenes, even when realistic uncertainties were added to the predictor variables. The accuracy and confidence levels of the neural network predictions were confirmed with an independent data set generated using CloudSat clouds, ECMWF meteorology interpolated in time and space to the fine vertical resolution of CloudSat, and surfaces that were both polarizing and non-isotropic. Typically, the accuracy and confidence levels for predictions of water cloud were 75% and 99%, while those for ice cloud were 89% and 95%.

Acknowledgments

The authors would like to thank Adam Carheden at CSU/CIRA for providing invaluable support with the OCO cluster and to Natalie Tourville for her assistance with general computing issues. Thanks also are due to Chris O'Dell and Igor Polonsky for their assistance with various aspects of the radiance simulations. In addition we would like to thank anonymous reviewers for providing valuable feedback on this work. This research was funded by JPL contract 1280999.

References

[1] D. M. O'Brien and P. J. Rayner, "Global observations of the carbon budget 2. CO_2 column from differential absorption of reflected sunlight in the 1.61 μm band of CO_2," *J. Geophys. Res.* **107**(D18), 4354 (2002). [doi:10.1029/2001JD000617].

[2] D. Crisp, R. Atlas, F.-M. Breon, L. R. Brown, J. Burrows, P. Ciais, B. J. Connor, S. C. Doney, I. Y. Fung, D. J. Jacob, C. E. Miller, D. O'Brien, S. Pawson, J. T. Randerson, P. Rayner, R. J. Salawitch, S. P. Sander, B. Sen, G. L. Stephens, P. P. Tans, G. C. Toon, P. O. Wennberg, S. C. Wofsy, Y. L. Yung, Z. Kuang, B. Chudasama, G. Sprague, B. Weiss, R. Pollock, D. Kenyon, and S. Schroll, "The Orbiting Carbon Observatory (OCO) mission," *Adv. Space Res.* **34**, 700–709 (2004). [doi:10.1016/j.asr.2003.08.062].

[3] F. Chevallier, "Sampled databases of 60-level atmospheric profiles from the ECMWF analyses," SAF Programme Research Report 4, EUMETSAT/ECMWF, Am Kavalleriesand 31, Postfach D-64297 Darmstadt, Germany (2001).

[4] "Committee on Earth Observation Satellites: The Worldwide Reference System (WRS)," (2009). http://ceos.cnes.fr:8100/cdrom-00/ceos1/satellit/landsat7/wrs.htm.

[5] C. Cox and W. H. Munk, "The measurement of the roughness of the sea surface from photographs of the sun's glitter," *J. Opt. Soc. Amer.* **44**, 838–850 (1954). [doi:10.1364/JOSA.44.000838].

[6] E. C. Monahan and I. ÓMuircheartaigh, "Optimal power-law description of oceanic whitecap coverage dependence on wind speed," *J. Phys. Oceanogr.* **10**, 2094–2099 (1980). [doi:10.1175/1520-0485(1980)010¡2094:OPLDOO¿2.0.CO;2].

[7] A. K. Heidinger, C. O'Dell, R. Bennartz, and T. Greenwald, "The successive-order-of-interaction radiative transfer model: Part I: Model development," *J. Appl. Meteorol. Clim.* **45**, 1388–1402 (2006). [doi:10.1175/JAM2387.1].

[8] C. O'Dell, A. K. Heidinger, T. Greenwald, P. Bauer, and R. Bennartz, "The successive-order-of-interaction radiative transfer model: Part II: Model performance and applications," *J. Appl. Meteorol. Clim.* **45**, 1403–1413 (2006). [doi:10.1175/JAM2409.1].

[9] C. W. O'Dell, "Acceleration of multiple-scattering, hyperspectral radiative transfer calculations via low-streams interpolation," *J. Geophys. Res.* (2009). submitted.

[10] D. Ivanova, D. L. Mitchell, W. P. Arnott, and M. Poellot, "A GCM parameterization for bimodal size spectra and ice mass removal rates in mid-latitude cirrus clouds," *Atmospheric Research* **59–60**, 89–113 (2001). [doi:10.1016/S0169-8095(01)00111-9].

[11] B. A. Baum, A. J. Heymsfield, P. Yang, and S. T. Bedka, "Bulk scattering properties for the remote sensing of ice clouds. Part I: Microphysical data and models," *J. Appl. Meteorol.* **44**, 1885–1895 (2005). [doi:10.1175/JAM2308.1].

[12] B. A. Baum, P. Yang, A. J. Heymsfield, S. Platnick, M. D. King, Y.-X. Hu, and S. T. Bedka, "Bulk scattering properties for the remote sensing of ice clouds. Part II: Narrowband models," *J. Appl. Meteorol.* **44**, 1896–1911 (2005). [doi:10.1175/JAM2309.1].

[13] B. A. Baum, "The development of ice cloud scattering models for use in remote sensing applications," (2009). http://www.ssec.wisc.edu/ baum/Cirrus/IceCloudModels.html.

[14] K. N. Bower, T. W. Choularton, J. Latham, M. B. Baker, and J. Jensen, "A parameterization of warm clouds for use in atmospheric general circulation models," *J. Atmos. Sci.* **51**, 2722–2732 (1994). [doi:10:1175/1520-0469(1994)051¡2722:APOWCF¿2.0.CO;2].

[15] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Company (1996).

[16] V. M. Krasnopolsky, B. H. Gemmill, and L. C. Breaker, "A neural network multiparameter algorithm for SSM/I ocean retrievals: Comparisons and validations," *Remote Sens. Environ.* **73**, 133–142 (2000). [doi:10.1016/S0034-4257(00)00088-2].

[17] W. J. Blackwell, "A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data," *IEEE Trans. Geosci. Remote Sens.* **43**(11), 2535–2546 (2005). [doi:10.1109/TGRS.2005.855071].

[18] F. Aires, W. B. Rossow, N. A. Scott, and A. Chédin, "Remote sensing from the infrared atmospheric sounding interferometer instrument: 2. Simultaneous retrieval of temperature, water vapor and ozone atmospheric profiles," *J. Geophys. Res.* **107**(D22) (2002). [doi:10.1029/2001JD0001591].

[19] T. Kavzoglu and P. M. Mather, "The use of backpropagating artificial neural networks in land cover classificaton," *Int. J. Remote Sens.* **24**(23), 4907–4938 (2003). [doi:10.1080/0143116031000114851].

[20] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*, SPIE — The International Society for Optical Engineering, Bellingham, Washington, U.S.A. (2005).

[21] G. L. Stephens, *Remote Sensing of the Lower Atmosphere, An Introduction*, Oxford University Press, New York (1994).

[22] E. P. Shettle and R. W. Fenn, "Models for the aerosols of the lower atmosphere and the effects of humidity variations on their optical properties," Technical report Environmental research papers, No. 676, AFGL-TR-79-0214, Air Force Geophysics Laboratory, Hanscom Air Force Base, Massachusetts 01731 (1979).

[23] D. O'Brien, I. Polonsky, P. Stephens, and T. E. Taylor, "Feasibility of cloud screening using proxy photon path length distributions derived from high resolution spectra in the near infrared," *J. Atmos. Oceanic Technol.* (2009). accepted.

[24] H. Bösch, G. C. Toon, B. Sen, R. A. Washenfelder, P. O. Wennberg, M. Buchwitz, R. de Beek, J. P. Burrows, D. Crisp, M. Christi, B. J. Connor, V. Natraj, and Y. L. Yung, "Space-based near-infrared CO_2 measurements: Testing the Orbiting Carbon Observatory retrieval algorithm and validation concept using SCIAMACHY observations over Park Falls, Wisconsin," *J. Geophys. Res.* **111** (2006). [doi:10.1029/2006JD007080].