Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians

F. Aires

Department of Applied Physics and Applied Mathematics, Columbia University/NASA Goddard Institute for Space Studies, New York, USA

CNRS/IPSL/Laboratoire de Météorologie Dynamique, École Polytechnique, Palaiseau, France

C. Prigent

CNRS, LERMA, Observatoire de Paris, Paris, France

W. B. Rossow

NASA Goddard Institute for Space Studies, New York, USA

Received 22 September 2003; revised 17 February 2004; accepted 18 March 2004; published 21 May 2004.

[1] Used for regression fitting, neural network (NN) models can be used effectively to represent highly nonlinear, multivariate functions. In this situation, most emphasis has been on estimating the output errors, but almost no attention has been given to errors associated with the internal structure of the NN model. The complex relationships linking the inputs to the outputs inside the network are the essence of the model and assessing their physical meaning makes all the difference between a "black box" model with small output errors and a physically meaningful model that will provide insight on the problem and will have better generalization properties. Such dependency structures can, for example, be described by the NN Jacobians: they indicate the sensitivity of one output with respect to the inputs of the model. Estimating these Jacobians is essential for many other applications as well. We use a new method of uncertainty estimate developed in the work of *Aires* [2004] to investigate the robustness of the quantities that characterize the NN structure. A regularization strategy based on principal component analysis is proposed to suppress the multicolinearities that are a major concern when analyzing the internal structure of such a model. The theory is applied to the remote sensing application already presented in the work of *Aires* [2004] and *Aires et al.* [2004]. *INDEX TERMS:* 0933 Exploration Geophysics: Remote sensing; 3210 Mathematical Geophysics: Modeling; 3260 Mathematical Geophysics: Inverse theory; 3399 Meteorology and Atmospheric Dynamics: General or miscellaneous; *KEYWORDS:* remote sensing, uncertainty, neural networks

Citation: Aires, F., C. Prigent, and W. B. Rossow (2004), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians, *J. Geophys. Res.*, *109*, D10305, doi:10.1029/2003JD004175.

1. Introduction

[2] The Jacobian, or sensitivities, of a NN (neural network) model are defined as the partial first derivatives of the model outputs with respect to its inputs. These quantities are very useful. First, they allow one to investigate statistically how a trained NN model derives the outputs from the inputs. These Jacobians can identify nonrobust models and as such have been used for model selection [*Rivals and Personnaz*, 2003]. Second, determining such Jacobians can be important for a given application. For example, when modeling the Inverse Radiative Transfer Equation (IRTE) in the atmosphere, the NN Jacobians can be viewed as estimates of the actual physical Jacobians of the IRTE [*Aires et al.*, 1999] but this task requires some form of regularization. Recently, NN Jacobians have also been used to analyze feedback processes in a dynamical system [*Aires and Rossow*, 2003].

[3] A concern with these applications can be raised: the neural network model is trained to obtain good fit statistics for its outputs but, most of the time, no constraint is applied to structure the internal regularities of the model. Statistical inference is often an ill-posed inverse problem [*Tarantola*, 1987; *Vapnik*, 1997; *Aires*, 1999]: many solutions can be found for the NN parameters, i.e., the synaptic weights, for similar output statistics. One of the reasons of this nonunique solution comes from the fact that multicolinearities can exist among the variables. Such correlations on input or output variables are a major problem even for linear regressions: the

Published in 2004 by the American Geophysical Union.

parameters of the regression are very unstable and can vary drastically from one experiment to another. The Jacobians are the equivalent of the linear regression parameters so a similar behavior is expected: when multicolinearities are present, the Jacobians will probably be highly variable and unreliable, even if the output statistics of the NN are very good. The aim of this paper is to investigate this problem, analyze it, and to suggest a solution.

[4] We will use the application already presented in the work of [*Aires et al.*, 2001, 2004] and *Aires* [2004], concerning the retrieval of surface skin temperature (*Ts*), surface microwave emissivities (E_m), and integrated water vapor (*WV*) over land based on the combination of microwave and infrared satellite observations. In the work of *Aires* [2004], we provided a method to estimate the uncertainty of the neural network weights [*MacKay*, 1992; *Neal*, 1996; *Bishop*, 1996; *Nabney*, 2002]. These uncertainties can be used for a variety of applications, using theoretical derivations, when they are available [see *Aires et al.*, 2004], or they can be used in a Bayesian statistics analysis with simulations by Monte Carlo techniques [*Gelman et al.*, 1995]. The latter one will be used in this paper to obtain uncertainties of the NN Jacobians. Estimation of the Jacobian uncertainties is then used as a diagnostic tool to identify nonrobust regressions, resulting from unstable learning processes.

[5] The solution of the multicolinearity and all robustness problems in general, is some form of regularization [*Vapnik*, 1997]. Many regularization techniques exist to reduce the number of degrees of freedom in the model for the multicolinearity problem or for any other ill-posed problem. For example, one approach consists in reducing the number of inputs to the NN [*Rivals and Personnaz*, 2003], this is a model selection tool. However, the introduction of redundant information in the input of the NN can be useful for reducing the observational noise [e.g., *Aires et al.*, 2002b, 2002c] as long as the NN learning is regularized in some way. Furthermore, the input variables used in this work are highly correlated (among brightness temperatures, among first guesses, or between observations and first guesses) so it would be difficult to extract few of the original variables and avoid the multicolinearities by input selection. We propose to solve this nonrobustness by using a principal component analysis (PCA) regression approach that would suppress the multicolinearities. Such an approach was successfully used in the work of *Aires et al.* [2002b]. It is expected that based on this representation of the inputs and/or outputs, where correlations are suppressed, the solution to the regression problem would be unique meaning that the Jacobians would be more reliable and physically more meaningful.

[6] The Principal Component Analysis (PCA) of the NN input and output data is described in section 2. The regularization process by PCA is described in section 3. Network Jacobians are presented together with their uncertainties in section 4. Conclusions and perspectives are discussed in section 5.

2. Principal Component Analysis of Inputs and Outputs

2.1. Principal Component Analysis

[7] Let $\boldsymbol{C}_x$ be the $K \times K$ covariance matrix of inputs to a neural network, and $\boldsymbol{C}_y$ be the $M \times M$ covariance matrix of the outputs. (It is interesting to note here that the input data is quite particular, both observations and first guess information are mixed. A specific analysis of the PCA in this mixed information would be very interesting but this is beyond the scope of this paper.) We use the eigen-decomposition of these two matrices to obtain $\boldsymbol{F}_x$ and $\boldsymbol{F}_y$ the $K \times K$ and $M \times M$ matrices whose columns are the corresponding eigen-vectors.

[8] Instead of the full matrices, we can use the truncated $K' \times K$ matrix $\overline{\boldsymbol{F}_x}$ and the $M' \times M$ matrix $\bar{\boldsymbol{F}}_y$ ($K' < K$ and $M' < M$), to use only the higher-order components [*Aires et al.*, 2002a]. Inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ are projected using:

$$\bar{\boldsymbol{x}} = \overline{\boldsymbol{F}_x} \cdot \boldsymbol{S}_{1x}^{-1} \cdot (\boldsymbol{x} - \boldsymbol{m}_{1x}) \tag{1}$$

$$\bar{\boldsymbol{y}} = \overline{\boldsymbol{F}_y} \cdot \boldsymbol{S}_{1y}^{-1} \cdot (\boldsymbol{y} - \boldsymbol{m}_{1y}), \tag{2}$$

where $\boldsymbol{S}_{1x}$ and $\boldsymbol{S}_{1y}$ are the diagonal matrices with diagonal terms equal to the standard deviation of respectively inputs and outputs, and the vectors $\boldsymbol{m}_{1x}$ and $\boldsymbol{m}_{1y}$ are the input and output means. The vectors $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ are a compression of the real data but the inverse transformations of (1) and (2) go back from the compression to the full representation with, or course, some compression errors. PCA is optimum in the least squares sense: the square errors between data and its PCA representation is minimized.

[9] Using a reduced-PCA representation allows us to reduce the dimension of the data but a compromise needs to be found between a good compression level (i.e., smaller number of PCA components used) and a small compression error (i.e., larger number of PCA components used). The more PCA components used for compression, the lower the compression error is. Another advantage of the PCA representation is to suppress part of the noise during the compression process, when the lower-order principal components of a PCA decomposition describe the real variability of the observations or the signal and the remaining principal components describe higher frequency variabilities. The higher orders are more likely to be related to the Gaussian noise of the instrument or to very minor variability. We will consider in the following that the higher-order components describe noise (instrumental plus unimportant information) and use the reduced instead of the full PCA representation. We will not comment on compression or denoising considerations in this study [see *Aires et al.*, 2002a].

2.2. PCA Results on Inputs and Outputs

[10] In this section, we analyze the inputs and the outputs of the NN learning database $\mathcal{B}$ in order to define a representation that will optimize the NN processing.

[11] The NN outputs are the surface skin temperature (*Ts*), the integrated water vapor (*WV*), and the land surface emissivities (E_m). The microwave surface emissivities are described by 7 output values (i.e., the 7 frequency-polarization channels of the SSM/I instrument) in the neural network whereas *WV* and *Ts* are each described by only 1 value. In order to give the same importance to each of these 3 physical quantities, we use a additional weights for each of the neural outputs: 1 for *Ts* and *WV*, and 1/7 for each E_m. The interpretation of the components, and in particular

Table 1. Cumulative Percentage Explained Variance of Input, ***x***, and Output, ***y***, With Respect to the Number of PCA Components

Number of PCA Components Used	Cumulative Explained Variance for Inputs, %	Cumulative Explained Variance for Outputs, %
1	42.50	57.6844
2	68.23	94.1111
3	81.94	98.1895
4	86.38	99.4994
5	90.56	99.9346
6	92.64	99.9675
7	94.47	99.9910
8	96.09	99.9974
9	97.49	100.0000
10	98.35	
11	99.01	
12	99.46	
13	99.78	
14	99.87	
15	99.94	
16	99.98	
17	100.00	

the individual amplitudes of the anomalies for each PCA component, is not altered by this normalization.

[12] Table 1 describes the cumulative percentage of explained variance by a cumulative number of PCA components for the input and output data. In Figure 1, the PCA basis functions for the input data are shown. The first PCA component, which explains ~42% of the variance, is dominated by *Ts* and the microwave brightness temperatures (*TB*), with very similar weights for all frequencies. It is the second component that represents the differences between the information carried by the horizontal and vertical polarization channels. The water vapor variance only dominates in the fourth component. Figure 2 presents the PCA basis functions for the output data. The first component explains more than half of the signal and is dominated by *Ts* and E_m, with similar weights for all E_m. *WV* and the E_m polarization differences are represented in the second component.

[13] Even if the PCA is only optimal for Gaussian-distributed data sets, it can still be used with more complex distributions with satisfactory compression levels. Applying PCA to non-Gaussian data results in non-Gaussian distributions for the PCA components [*Aires et al.*, 2002d]. In Figure 3, the 6 first output PCA component distributions are shown. As can be seen, the distributions can be skewed (see Figures 3a, 3b, 3e, or 3f) or they can have positive kurtosis, i.e., platykurtic (see Figure 3d), or negative kurtosis, i.e., leptokurtic (see Figure 3c). This makes the use of a nonlinear model, such as the NN, even more important. Dealing with non-Gaussian-distributed data requires a

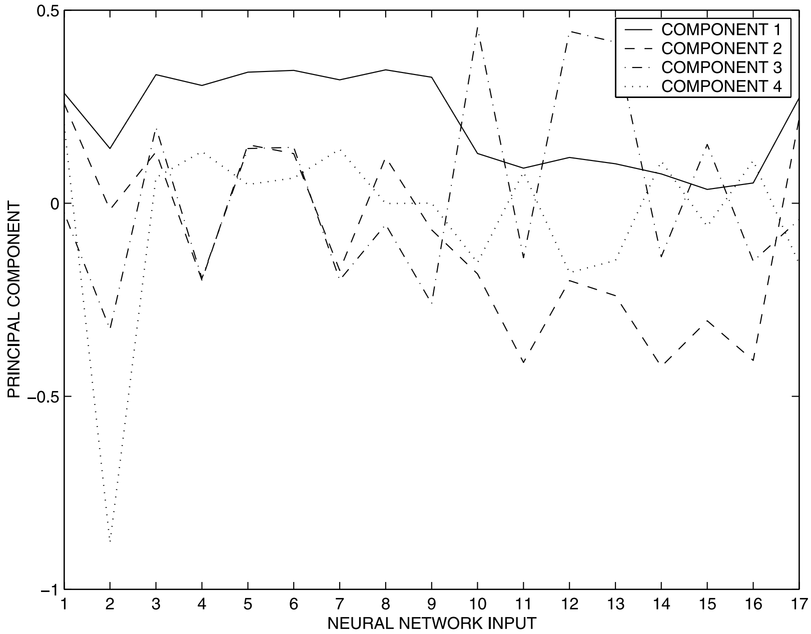

Figure 1. Eigen-decomposition of the covariance matrix $\boldsymbol{C_x}$ of the inputs, respectively *Ts*, *WV*, *TB*19*V*, *TB*19*H*, *TB*22*V*, *TB*37*V*, *TB*37*H*, *TB*85*V*, *TB*85*H*, E_m19*V*, E_m19*H*, E_m22*V*, E_m37*V*, E_m37*H*, E_m85*V*, E_m85*H*, *Tlay*.

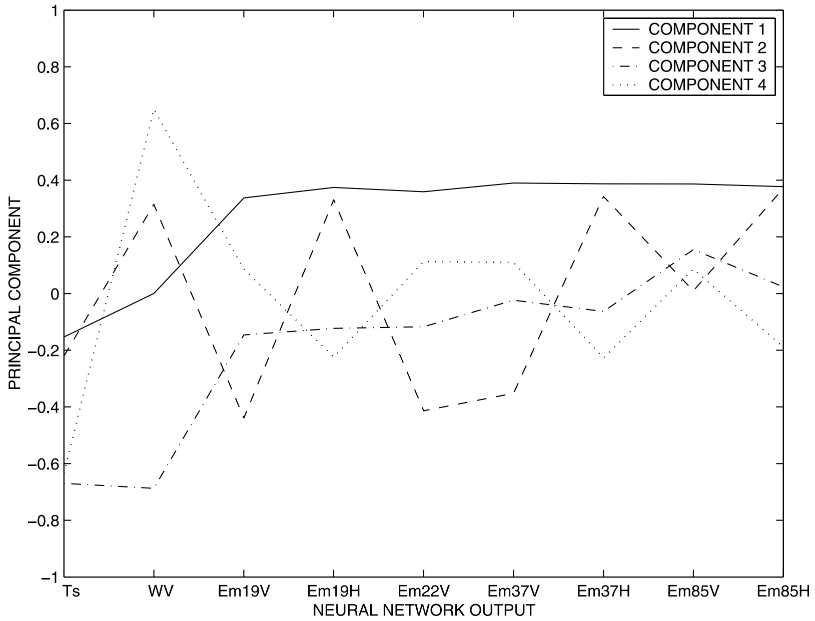

Figure 2. Eigen-decomposition of the covariance matrix C_y of the outputs.

model able to use the complex and nonlinear dependencies in the data or to represent its regime-dependency [*Aires et al.*, 2000]. Also note that extreme events can occur; such very strong absolute component value situations are particularly numerous for PCA components 5 and 6, see Figures 3e and 3f.

[14] To check the physical consistency of the PCA, samples of the database $\mathcal{B}$ are projected onto the map of the first two principal components that represent most of the variability (Figure 4). For display purposes, the clouds of points are indicated by the one-sigma contour line (i.e., the mean is the mean and the standard deviations of the represented Gaussians are the mean and the standard deviations of the cloud of points). The projection differentiates the different land surface types in a set of Gaussians modes that are well separated and that are physically consistent. Such PCA maps can be used for clustering or classification schemes. The negative first component values, such as for the rain forest, means that for this vegetation type *Ts* is above the mean value (the weight of *Ts* in the first component being negative, see Figure 2). In contrast, tundra has a positive first component, indicating *Ts* is lower than the mean value. By the same token, the second component is highly positive for

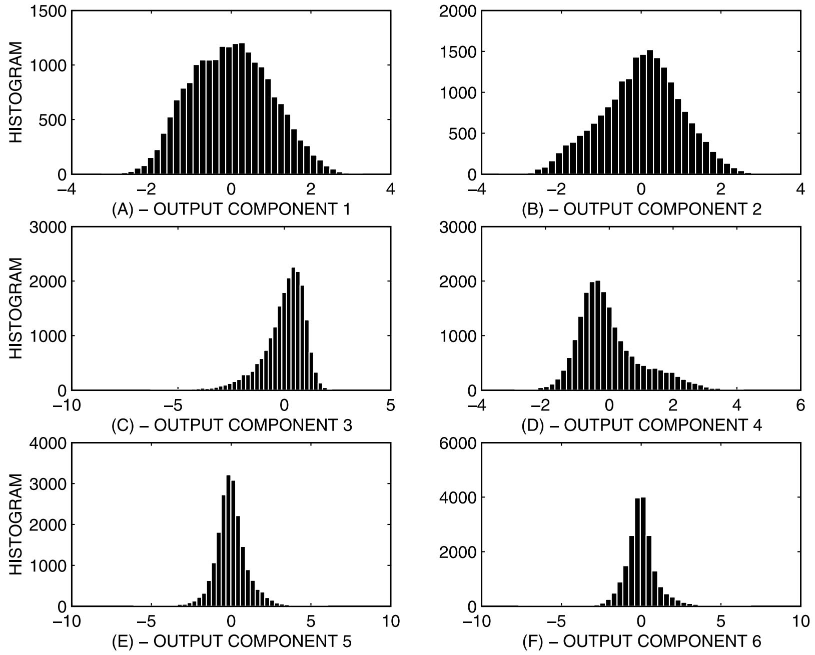

Figure 3. Histograms of the first 6 PCA components on outputs.

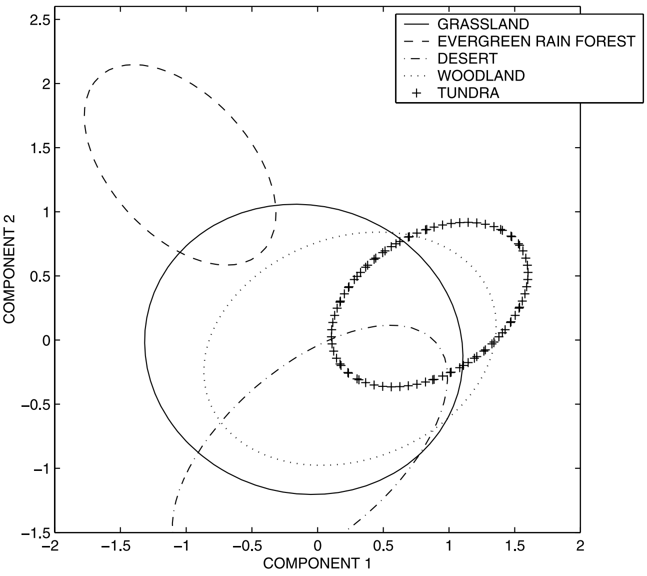

Figure 4. One-sigma contour of the distribution of data projected on the first 2 PCA components.

the rain forest, indicating *WV* higher than the mean in equatorial regions, as expected (note that the weight of WV in the second component is positive). Since surface types are known to represent a large part of the variability, the fact that the PCA is able to coherently separate them demonstrates the physical significance of the PCA representation. This is particularly important because the PCA will be used, in the following, to regularize the NN learning. The patterns that are found by the PCA will distribute the contribution of each input and each output for a given sensitivity. It is essential that these patterns have a physical meaning.

3. PCA Regression Approach

[15] This sections aims at using the PCA representation described earlier to perform the regression that is the global inversion model used for the remote sensing application. As it will be shown in section 4 this PCA regression regularized the NN Jacobians found.

3.1. PCA Regression

[16] The practical benefits of using PCA components instead of the raw inputs and outputs are that the NN method is faster because of the reduced data dimension and the reduced noise level in the observations. Furthermore, the learning stage is faster since the network has fewer inputs and outputs and fewer parameters to estimate.

[17] The fact that the dimension of the inputs is reduced decreases the number of parameters in the regression model (i.e., weights in the neural network), and consequently decreases the number of degrees of freedom in the model, which is good for any statistical techniques. The variance in determining the actual values of the neural weights is also reduced. The combination of PCA and NN has been used for example in the work of *Aires et al.* [2002b] where a 8461-channel spectrum from a high-resolution infrared interferometer has been compressed into a 100 component PCA representation to retrieve atmospheric profiles of temperature, water vapor, and ozone.

[18] The training of the NN is simpler because the inputs are decorrelated. Correlated inputs in a regression are called multicolinearities and they are well-known to cause problems for the model fit [*Gelman et al.*, 1995]. Suppressing these multicolinearities makes the minimization of the quality criterion more efficient: it is easier to minimize, with less probability of becoming trapped in a local minimum. Therefore it has the general effect of suppressing uncertainty in the determination of the parameters of the NN model. For a detailed description of PCA-based regression, see *Jolliffe* [2002].

3.2. Number of Components Used

[19] How many PCA components should the regression use? From section 2.1, it is preferable to use the optimal compromise between the best compression fit and denoising in terms of global statistics. This statement is related to the PCA representation, not taking into account how the NN uses these components. No theoretical results exist to define the optimal number of PCA components to be used in a regression, it entirely depends on the problem to be solved. Various tests can be performed. Experience with the NN technique shows that, if the problem is well regularized, once sufficient information is provided as input, adding more PCA components to the inputs does not have a large impact on the retrieved results; the processing just requires more computations because of the increased data dimension. Therefore we recommend being conservative and taking more PCA components than the de-noising optimum would indicate in order to keep all possibly useful information.

[20] In terms of output retrieval quality, the number of PCA components used in the input of the NN needs to be reduced for other reasons than just denoising or compression. In fact, during the learning stage, the NN is able to relate each output to the inputs that help predict it, disregarding the inputs that vary randomly. In some cases, the dimension of the network input is so big (few thousands) [*Aires et al.*, 2002a] that a compression is necessary. In our case, $K = 17$ is easily manageable so all the inputs variables could be used. For our study here, $K' = 12$ is chosen to reduce the number of degrees of freedom in the network architecture. This number of input PCA components is large enough for the retrieval, representing 99.46% of the total variance (see Table 1). No additional information would be gained from adding the higher-order PCA input components.

[21] The number of PCA components used in the NN output is related to the retrieval error magnitude for a nonregularized NN. If the compression error is minimal compared to the retrieval error of the nonregularized network, then M', the number of output components used, is satisfactory. It would be useless to try to retrieve something that is noise in essence. Furthermore, it could lead to numerical problems too and interfere with the retrieval of the other, more important, output components. In this application, $M' = 5$ has been chosen, representing 99.93% of the total variability of the outputs (see Table 1). It is particularly interesting to note that the ill-conditioning of

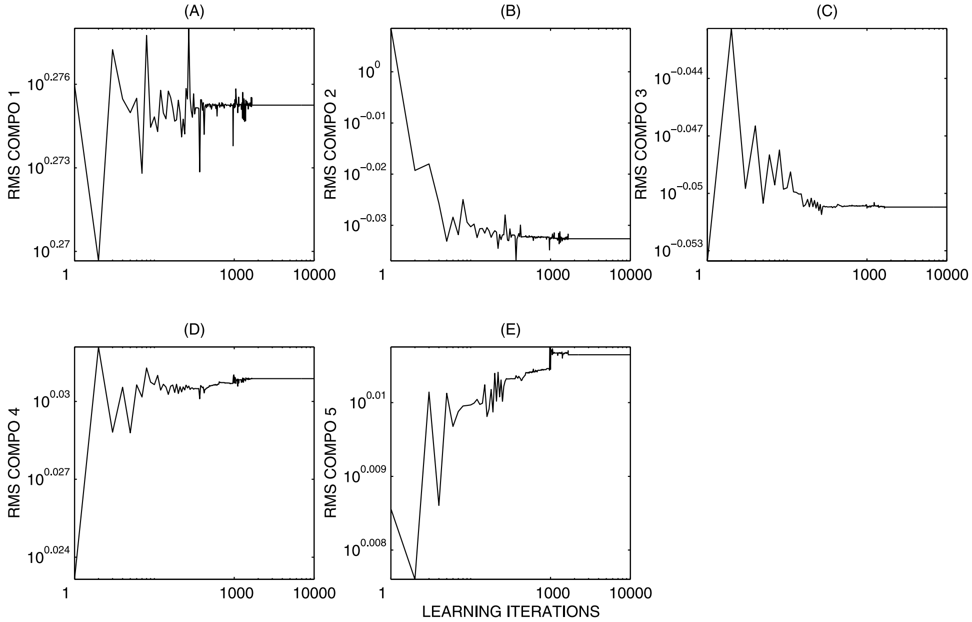

Figure 5. RMS error curves for the seven network output PCA components during the learning stage.

the Hessian matrix $\boldsymbol{H}$ [see *Aires*, 2004] is intimately related to the number of inputs and outputs chosen.

3.3. Postprocessing of the Data for the Quality Criterion

[22] Optimizing NN learning necessitates controlling correctly the system and diagnosing carefully the learning step, in contrast to the "black box" perception often associated with neural networks. In particular, emphasis must be put on the preprocessing and postprocessing of data.

[23] The normalization is performed before the PCA representation (section 2.2), but a post-PCA-processing normalization is also required. The outputs of the network, i.e., the PCA components, are not homogeneous, they have different dynamic ranges. The importance of each of the components in the output of the neural network is not equal. The first PCA component represents 52.68% of the total variance of the data, where the fifth component represents only 0.43% (Table 1). Giving the same weight to each of these components during the learning process would be misleading. To resolve this, we give a different weight to each of the network outputs in the "data" part, E_D, of the quality criterion used for the network learning [see *Aires*, 2004]. For an output component, this weight is equal to the standard deviation of the component. This is equivalent to using *Aires* [2004, equation (4)], where $\boldsymbol{A}_{in}$ is the diagonal matrix with diagonal terms equal to the standard deviation of the PCA components. Off-diagonal terms are zero since, by definition, no correlation exists between the components in $\varepsilon_y = (\boldsymbol{t}^{(n)} - g_{\boldsymbol{w}}(\boldsymbol{x}^{(n)}))$ (i.e., the output error, target or desired output minus the network output).

3.4. Retrieval Results

[24] The PCA-regularized NN has an architecture of 12 inputs, 30 hidden-layer neurons, and 5 outputs (the nonregularized NN architecture was 17-30-9). The mean RMS retrieval error for the new NN with PCA representation of its inputs and outputs is slightly higher than for the original nonregularized NN. For example, the surface skin temperature RMS error is 1.53 instead of 1.46 in the nonregularized NN [see *Aires*, 2004, Table 1]. This is expected because we know that reducing variance (overfitting) by regularization increases the bias (RMS error). This is known as the bias/variance dilemma [*Geman et al.*, 1992]. This dilemma describes the compromise that must be found between a good fitting on the learning database $\mathcal{B}$ and a robust model with physical meaning. The differences of RMS errors are, in this case, negligible. The error differences are also higher for the retrieval of the E_m. This is due to the normalization of *Ts*, *WV*, and the seven E_m (section 2.2): each emissivity has less weight in the retrieval statistics because of the 1/7 factor used.

[25] The evolution of the learning statistics are presented in Figure 5: these results describe how the RMS error of the output PCA components retrieval decreases with the number of learning iterations. The decrease is much more unstable than a normal learning process, like in the work of *Aires* [2004, Figure 2]. Large oscillations occur in the beginning of the learning process because of the complex mixing of the components that the NN tries to retrieve: each component mixes variability from each of the 9 original output physical variables. The NN can first decrease the error in one component but then, in order to decrease the error in another component, it makes compromises that can result in a sudden increase of the RMS error for some other components. In Figure 6 we translate these PCA component RMS error curves back to the original physical variables space. This confirms that the decrease/stabilization of the RMS error of the output PCA representation induces a steady decrease of the RMS error of the physical output variables. These curves looks much smoother and more regular than the corresponding ones in the PCA component space.

[26] In order to estimate the NN weight uncertainties, we use the approach described in the work of *Aires* [2004]: the Hessian matrix $\boldsymbol{H}$ must, first, be computed, and then

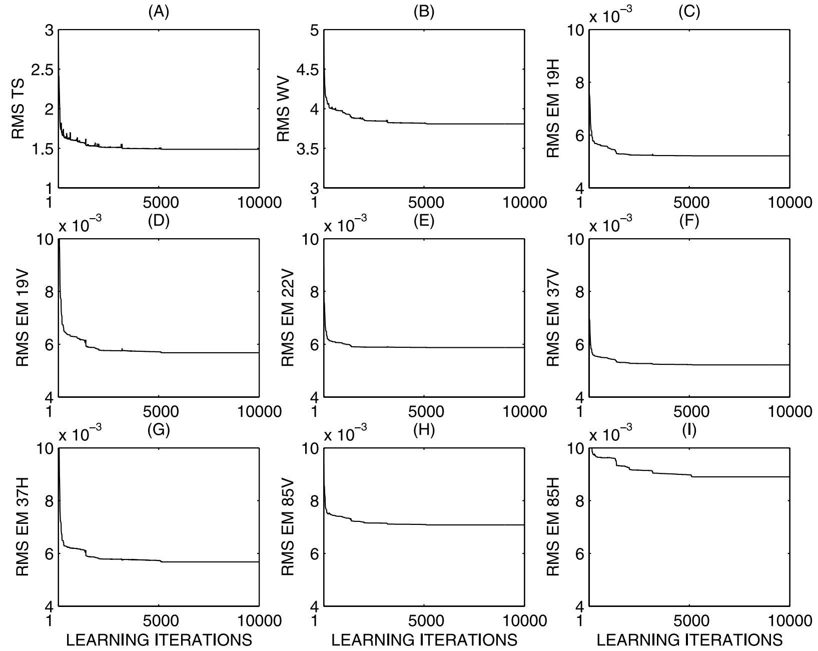

Figure 6. RMS error curves for the nine network outputs during the learning stage.

regularized, in order to obtain the covariance matrix of the weights PDF. This regularization of the Hessian matrix is done to make it positive definite which is not the same goal as the regularization of the NN behavior by the PCA representation. These two regularization steps should not be confused.

[27] Figure 7 presents the corresponding standard deviation for the neural network weights with various regularization parameters λ around the optimal value, λ = 660, which is determined as described in the work of *Aires* [2004] using various quality criteria. It is interesting that the ill-conditioning of the Hessian matrix shows large sensitivity to some particular network weights. For λ too small, the standard deviation is very chaotic and nonmonotonic with some values going from extreme large values to even negative ones. Increasing λ makes the standard deviation of the particular weights converging to a more acceptable, positive value, and coherent with the other standard deviations. At the same time, increasing λ uniformly decreases the standard deviation in all the network weights. The balance between a λ large enough to regularize ***H*** but without changing the standard deviation of well-behaved weights must be found [*Aires*, 2004]. This is probably the most important issue for the uncertainty

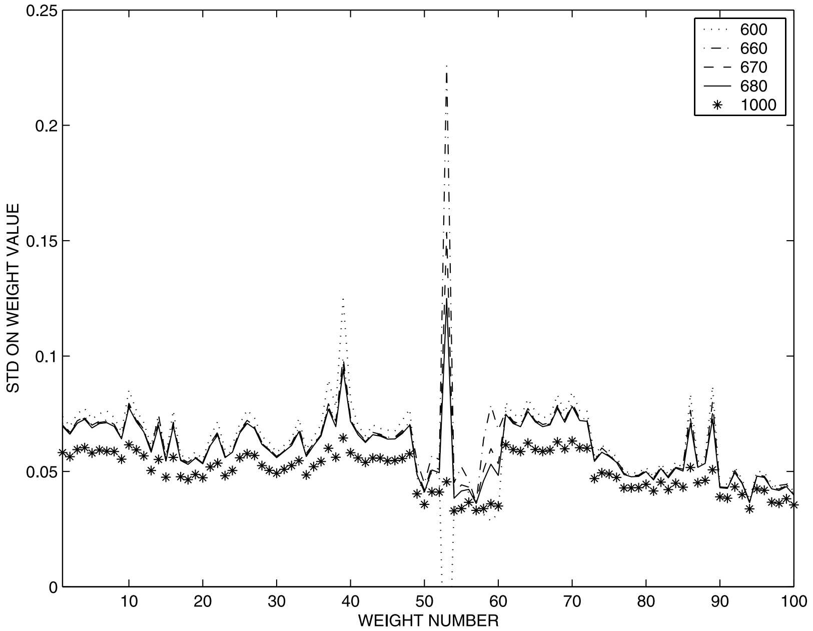

Figure 7. Standard deviation of NN weights with increased regularization parameter λ.

estimates described in this study. Another approach to obtain a well-conditioned Hessian would be, during the learning stage, to constrain the Hessian matrix $\boldsymbol{H}$ to stay definite positive.

4. Uncertainty of Neural Network Jacobians

[28] The a posteriori distribution of weights is useful to estimate the uncertainties of network outputs [*Aires et al.*, 2004]. We will now show that these distributions can be used for the estimation of very complex probabilistic quantities via Monte Carlo simulations. As an example of such an approach, we use it to estimate the uncertainties of the neural network.

4.1. Definition of Neural Network Sensitivities

[29] The neural network technique not only provides a statistical model relating the input and output quantities, it also enables an analytical and fast calculation of the neural Jacobians (just the derivative of the analytical expression of the NN model), also called the neural sensitivities or adjoint model [*Aires et al.*, 1999]. For example, the neural Jacobians for the two-layered MLP (a MLP network with one hidden layer) are

$$\frac{\partial y_k}{\partial x_i} = \sum_{j\in S_1} w_{jk} \cdot \frac{d\sigma}{da}\left(\sum_{i\in S_0} w_{ij}x_i\right) \cdot w_{ij}. \tag{3}$$

For a more complex MLP network with more hidden layers, a back-propagation algorithm exists that computes efficiently the neural Jacobians [*Bishop*, 1996]. Since the NN is nonlinear, these Jacobians depend on the situation defined by the particular input, $\boldsymbol{x}$.

[30] The neural Jacobian concept is a very powerful tool since it allows for a statistical estimation of the multivariate and nonlinear sensitivities connecting the input and output variables in the model under study, which can be very useful as a data analysis tool [*Aires and Rossow*, 2003]. The Jacobian matrix gives the global mean sensitivities for each retrieved parameter: they indicate the relative contribution of each input in the retrieval for a given output parameter. The Jacobian is situation-dependent which means that, depending on the situation $\boldsymbol{x}$, the NN uses the available information in different ways.

4.2. Preprocessing and Postprocessing of Data

[31] Before they are introduced as inputs and outputs of the neural network, the reduced-PCA representations, $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$, need to be centered and normalized. This is a requirement for the neural network method to work efficiently. The new inputs and outputs of the neural network are given by:

$$\boldsymbol{x}' = \boldsymbol{S}_{2x}^{-1} \cdot (\overline{\boldsymbol{x}} - \boldsymbol{m}_{2x}) \tag{4}$$

$$\boldsymbol{y}' = \boldsymbol{S}_{2y}^{-1} \cdot (\overline{\boldsymbol{y}} - \boldsymbol{m}_{2y}), \tag{5}$$

where the $\boldsymbol{S}_{2x}$ and $\boldsymbol{S}_{2y}$ are the diagonal matrices of the standard deviations of respectively $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$ (defined in equations (1) and (2)) and vectors $\boldsymbol{m}_{2x}$ and $\boldsymbol{m}_{2y}$ are the respective means.

[32] The NN formulation allows derivation of the network Jacobian $\left[\frac{\partial \boldsymbol{y}'}{\partial \boldsymbol{x}'}\right]$ for the normalized quantities of equations (4) and (5). To obtain the Jacobian in physical units, one should use equations (1), (2), (4), and (5) to find:

$$\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right] = \boldsymbol{S}_{1y} \cdot \overline{\boldsymbol{F}_y}^T \cdot \boldsymbol{S}_{2y} \cdot \left[\frac{\partial \boldsymbol{y}'}{\partial \boldsymbol{x}'}\right] \cdot \boldsymbol{S}_{2x}^{-1} \cdot \overline{\boldsymbol{F}_x} \cdot \boldsymbol{S}_{1x}^{-1}. \tag{6}$$

[33] Equation (6) gives the neural Jacobian for the physical variables $\boldsymbol{x}$ and $\boldsymbol{y}$. To enable comparison of the sensitivities between variables with different variation characteristics, the terms $\boldsymbol{S}_{1y}$ and $\boldsymbol{S}_{1x}^{-1}$ can be suppressed in this expression so that, for each input and output variable, a normalization by its standard deviation is used. The resulting nonlinear Jacobians indicate the relative contribution of each input in the retrieval to a given output variable.

4.3. Marginalization and Sampling Strategy

[34] Marginalization is defined, in Bayesian statistics, as the simulations performed to integrate a conditional probability. This approach was not used frequently in the past but is now more and more attractive since the large number of computations required are much more manageable with modern computers.

[35] To go beyond the "point estimation" approach where a learning algorithm is used to estimate only the optimal set of weights, the distribution of weights $\boldsymbol{w}$ uncertainty must be investigated. This distribution of weights can be used to estimate complex probabilitistic quantities like the confidence intervals of stochastic variables, the distribution of the outputs [*Aires et al.*, 2004], and other probabilities of quantities dependent on the output of the network. All these potential applications require the integration under the PDF of weights. Fortunately, the a posteriori distribution of weights is supposed to be Gaussian [*Aires*, 2004]: This means that the normalization term $\frac{1}{Z^N}$ is easily obtained (this is a main difficulty when integrating a PDF). The integration and the manipulation of a Gaussian PDF is particularly easy compared to other distributions. However, when faced with the estimation of complex quantities, the analytical solution of such integrations can still be difficult to obtain. The estimation of the network Jacobians PDF is such a situation. This is why simulation strategies have to be used. Simulations first sample the PDF of weights $\{\boldsymbol{w}^r;\ r = 1 \ldots, R\}$ and then use this sample to approximate the integration under the whole weight PDF.

[36] Using only $\boldsymbol{w}^\star$, the Maximum A Posteriori (MAP) parameters, to estimate some other dependent quantities directly may not be optimal, even if we are not interested in uncertainty estimates. In fact most of the mass of the distribution (i.e., location of the domain where the probability is higher), in a high dimension space, can be far from the most probable state (i.e., the MAP state): The high dimension makes the mass of the PDF more on the periphery of the density domain and less at its center. Nonlinearities can also distort the distribution of the estimated quantity. This is why it is good to use R samples of the weights $\{\boldsymbol{w}^r;\ r = 1 \ldots, R\}$ to estimate the density of the quantity of interest. See Appendix A for various sampling techniques in high dimension spaces.

[37] Concerning the network Jacobians, the MAP network Jacobian is given by using the most probable network

Table 2. Global Mean Nonregularized Neural Sensitivities $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$[a]

	Ts	*WV*	E_m19V	E_m19H	E_m22V	E_m37V	E_m37H	E_m85V	E_m85H
TB19V	0.26 ± 0.19	0.04 ± 0.23	**0.91** ± 0.18*	−0.20 ± 0.23	**0.57** ± 0.22*	0.02 ± 0.19	−0.29 ± 0.15	−0.17 ± 0.21	−0.12 ± 0.19
TB19H	0.08 ± 0.19	**0.42** ± 0.27	−0.16 ± 0.24	**1.26** ± 0.40*	−**0.46** ± 0.36	−**0.54** ± 0.23*	0.03 ± 0.18	−**0.30** ± 0.23	−0.43 ± 0.27
TB22V	0.11 ± 0.19	−**0.79** ± 0.27*	0.17 ± 0.21	−0.14 ± 0.25	**0.59** ± 0.28*	−0.15 ± 0.21	−0.09 ± 0.17	−**0.77** ± 0.21*	−0.26 ± 0.22
TB37V	0.20 ± 0.18	−0.16 ± 0.21	0.19 ± 0.18	−0.25 ± 0.21	0.25 ± 0.21	**1.12** ± 0.19*	0.05 ± 0.15	**0.63** ± 0.20*	0.01 ± 0.19
TB37H	0.15 ± 0.18	−**0.67** ± 0.23*	−0.28 ± 0.17	−0.00 ± 0.22	−0.13 ± 0.21	0.18 ± 0.19	**0.84** ± 0.15*	−0.20 ± 0.20	**0.61** ± 0.21*
TB85V	0.24 ± 0.16	−0.05 ± 0.20	−**0.54** ± 0.17*	−0.14 ± 0.19	−**0.61** ± 0.23*	−0.29 ± 0.18	−**0.33** ± 0.14*	**1.06** ± 0.18*	−0.15 ± 0.20
TB85H	−0.13 ± 0.15	**1.60** ± 0.18*	0.05 ± 0.15	−0.16 ± 0.17	0.09 ± 0.18	−0.12 ± 0.16	0.02 ± 0.13	−0.14 ± 0.16	**0.45** ± 0.17*
Ts	0.18 ± 0.08*	−0.15 ± 0.11	−0.27 ± 0.08*	−0.14 ± 0.09	−0.26 ± 0.10*	−**0.31** ± 0.08*	−0.12 ± 0.07	−0.26 ± 0.09*	−0.07 ± 0.09
WV	−0.04 ± 0.05	**0.33** ± 0.07*	0.03 ± 0.06	0.04 ± 0.08	−0.01 ± 0.09	0.03 ± 0.06	−0.04 ± 0.05	−0.06 ± 0.07	−0.15 ± 0.08
E_m19V	−0.07 ± 0.04	0.07 ± 0.06	0.12 ± 0.05*	0.11 ± 0.08	0.09 ± 0.07	0.15 ± 0.05*	0.06 ± 0.04	0.16 ± 0.05*	0.03 ± 0.05
E_m19H	−0.11 ± 0.08	−0.03 ± 0.10	0.22 ± 0.09*	−0.04 ± 0.15	0.29 ± 0.14*	0.19 ± 0.09*	0.09 ± 0.07	0.16 ± 0.10	0.18 ± 0.10
E_m22V	−0.06 ± 0.04	0.04 ± 0.05	0.11 ± 0.04*	0.04 ± 0.04	0.15 ± 0.04*	0.13 ± 0.04*	0.05 ± 0.03	0.13 ± 0.04*	0.06 ± 0.04
E_m37V	−0.07 ± 0.04	0.02 ± 0.05	0.11 ± 0.04*	0.07 ± 0.05	0.13 ± 0.05*	0.16 ± 0.04*	0.07 ± 0.03*	0.15 ± 0.04*	0.07 ± 0.04
E_m37H	−0.08 ± 0.06	−0.07 ± 0.07	0.14 ± 0.06*	0.09 ± 0.07	0.15 ± 0.07*	0.18 ± 0.06*	0.11 ± 0.05*	0.20 ± 0.06*	0.15 ± 0.06*
E_m85V	−0.04 ± 0.04	−0.05 ± 0.06	0.07 ± 0.04	0.07 ± 0.05	0.11 ± 0.05*	0.10 ± 0.05	0.04 ± 0.04	0.20 ± 0.05*	0.10 ± 0.05*
E_m85H	−0.03 ± 0.07	−0.18 ± 0.09	0.12 ± 0.09	−0.04 ± 0.11	0.17 ± 0.10	0.12 ± 0.08	0.05 ± 0.06	0.21 ± 0.08*	0.21 ± 0.07*
Tlay	−0.03 ± 0.06	0.13 ± 0.09	−0.04 ± 0.08	0.01 ± 0.09	−0.07 ± 0.09	−0.06 ± 0.08	−0.03 ± 0.06	−0.13 ± 0.08	−0.04 ± 0.07

[a] Columns are network outputs, $\boldsymbol{y}$, and rows are network Inputs, $\boldsymbol{x}$. Sensitivities with absolute value higher than 0.3 are in bold and positive 5%-significance tests are indicated by a star. The first part of this table is for SSM/I observations, the second part corresponds to first guesses.

weights $\boldsymbol{w}^\star$. The mean Jacobian is not sufficient for a real sensitivity analysis, a measure of the uncertainty in this estimate is required as well. In fact, the neural network is designed to reproduce the right outputs, but without any a priori information, the internal regularities of the network have no constraint. As a consequence, the internal regularities, such as the NN Jacobians, are expected to have a large variability. This variability needs to be monitored.

[38] To estimate the uncertainties of the Jacobians, we use R = 1000 samples from the weights PDF described in the work of *Aires* [2004]. Using an adequate sampling algorithm is a key issue here. To sample this Gaussian distribution in very high dimension space (about 800 network weights), the metropolis algorithm is used (see Appendix A). This method is also suitable for non-Gaussian PDFs.

[39] For each weight sample $\boldsymbol{w}^r$, we estimate the mean Jacobian over the entire data set $\mathcal{B}$. This means that we have at our disposal a sample of R = 1000 mean Jacobians. They are then averaged and a PDF for each individual term in the Jacobian matrix is obtained.

4.4. Multicolinearity Problem

[40] Table 2 gives the mean neural Jacobian values for the variables x_k and y_i for the neural network, as defined in equation (3). As described in section 4.2, the neural Jacobians are normalized by the standard deviation of the respective variables to enable comparison of the sensitivities between variables with different variation characteristics. These values indicate the relative contribution of each input in the retrieval of a given output parameter. The numbers correspond to global mean over $\mathcal{B}$ values which may mask rather different behaviors in various regions of the input space. The standard deviations of the uncertainty PDF are also indicated. The variability of the Jacobians is large: uncertainty of the neural sensitivities can be up to several times the mean value. For most cases, the Jacobian value is not in the confidence interval, which means that the actual value is not significant. In linear regression, obtaining nonsignificant parameters is often the signal that multicolinearities are a problem for the regression.

[41] A sample of sensitivities from the predictive distribution is presented (Figure 8) along with the distribution of these sensitivities, confirming the large uncertainties of the sensitivities. The distribution of the Jacobians shows that most of them are not statistically significant. The reason for such uncertainty can be the pollution of the learning process by multicolinearities in the data (inputs and outputs), which introduce compensation phenomena. For example, if two inputs are correlated and they are used by the statistical regression to predict an output component, then the learning has some indeterminacy: it can give more or less emphasis to the first of the inputs as long as it compensates this under- or over-allocation by, respectively, an over- or under-allocation in the second, correlated, input variable. This means that the two corresponding sensitivities will be highly variable from one learning cycle to another one. The output prediction would be just as good for both cases, but the internal structure of the model would be different. Since it is these internal structures (i.e., Jacobians) that are of interest here, this problem needs to be resolved.

[42] To see if the multicolinearities and consequent compensation phenomena are at the origin of the sensitivity uncertainties, the correlation between sensitivities is measured. If some of these sensitivities are correlated or anticorrelated it means that, from one learning cycle to another, the sensitivities will always be related following the compensation principle. The correlation of a set of sensitivities is shown in Table 3; some of the correlations are significant. For example, as expected, the correlation between the sensitivities of *Ts* to *TB*19*V* and *TB*22*V* is larger in absolute value than *Ts* with higher frequency *TB*. The negative sign of this correlation is explained by the fact that *TB*19*V* being highly correlated with *TB*22*V*, a large sensitivity of *Ts* to *TB*19*V* will be compensated for in the NN by a low sensitivity to *TB*22*V*, leading to a negative correlation. The absolute value of the correlations is not extremely high (about 0.3 or 0.4) but when added, all these correlations define a quite complex and strong dependency structure among the sensitivities. This is a sign that multicolinearities and subsequent compensations are acting in the network model.

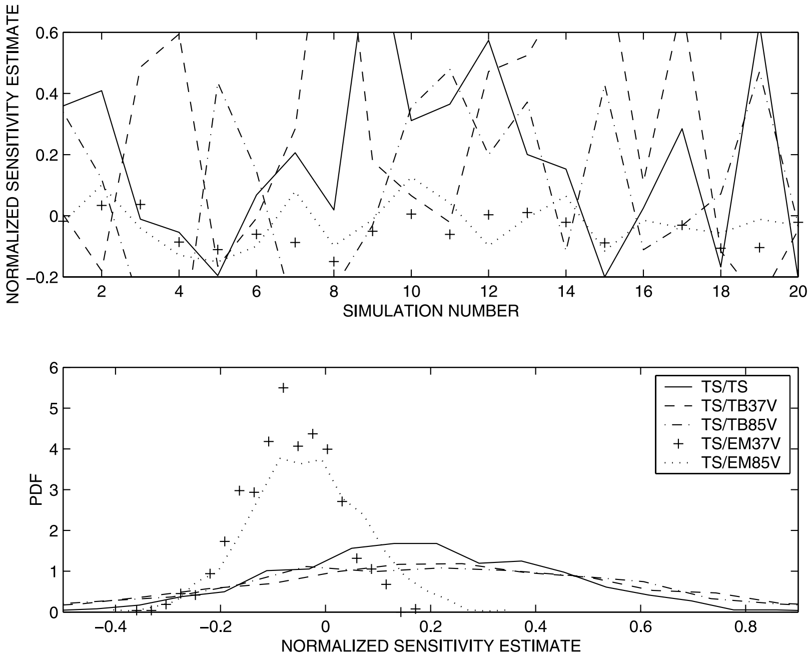

Figure 8. (top) Twenty samples of 5 neural network sensitivities $\left(\frac{\partial Ts}{\partial Ts}, \frac{\partial Ts}{\partial TB37V}, \frac{\partial Ts}{\partial TB85V}, \frac{\partial Ts}{\partial E_m 37V}\right.$, and $\left.\frac{\partial Ts}{\partial E_m 85V}\right)$, and (bottom) histogram of the same network sensitivities.

[43] To avoid such multicolinearity problems, the network learning needs to be regularized: (1) by using some physical a priori information to better constrain the learning, in particular in term of dependency structure among the variables; (2) or by employing some statistical a priori information that will help reduce the number of degrees of freedom in the learning process in a physically meaningful way. In the following sections, we investigate the latter regularization strategy by using principal component analysis.

4.5. PCA-Regularized Jacobians

[44] In Table 4, the PCA-regularized NN (see section 3.4) is used to estimate the mean Jacobian matrix $\left[\frac{\partial \mathbf{y}'}{\partial \mathbf{x}'}\right]$ of raw network outputs and inputs, together with the corresponding standard deviations. The standard deviations are much more

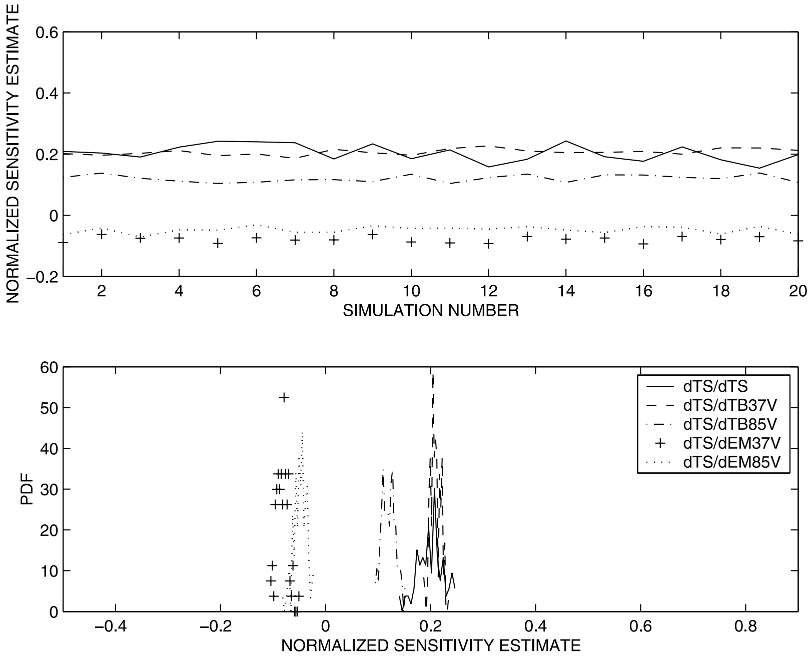

Figure 9. (top) Twenty samples of 5 regularized neural network sensitivities $\left(\frac{\partial Ts}{\partial Ts}, \frac{\partial Ts}{\partial TB37V}, \frac{\partial Ts}{\partial TB85V}, \frac{\partial Ts}{\partial E_m 37V}\right.$, and $\left.\frac{\partial Ts}{\partial E_m 85V}\right)$, and (bottom) histogram of the same network sensitivities.

Table 3. Correlation Matrix for a Sample of Neural Network Sensitivities[a]

	$\frac{\partial Ts}{\partial Ts}$	$\frac{\partial Ts}{\partial TB19V}$	$\frac{\partial Ts}{\partial TB19H}$	$\frac{\partial Ts}{\partial TB22V}$	$\frac{\partial Ts}{\partial TB37H}$	$\frac{\partial Ts}{\partial TB85V}$	$\frac{\partial Ts}{\partial TB85H}$	$\frac{\partial Ts}{\partial E_m 19V}$	$\frac{\partial Ts}{\partial E_m 19H}$	$\frac{\partial Ts}{\partial E_m 85H}$
$\frac{\partial Ts}{\partial Ts}$	1.00	...	...	...	...	...	...	...	...	...
$\frac{\partial Ts}{\partial TB19V}$	−0.19	1.00	...	...	...	...	...	...	...	...
$\frac{\partial Ts}{\partial TB19H}$	−0.15	−0.18	1.00	...	...	...	...	...	...	...
$\frac{\partial Ts}{\partial TB22V}$	−0.05	**−0.44**	−0.16	1.00	...	...	...	...	...	...
$\frac{\partial Ts}{\partial TB37H}$	0.13	−0.00	**−0.59**	−0.01	1.00	...	...	...	...	...
$\frac{\partial Ts}{\partial TB85V}$	−0.08	−0.04	0.12	−0.18	−0.03	1.00	...	...	...	...
$\frac{\partial Ts}{\partial TB85H}$	−0.01	0.18	−0.05	−0.08	**−0.38**	**−0.45**	1.00	...	...	...
$\frac{\partial Ts}{\partial E_m 19V}$	0.14	−0.17	0.25	−0.16	−0.06	0.15	0.04	1.00	...	...
$\frac{\partial Ts}{\partial E_m 19H}$	−0.00	0.14	**−0.44**	0.09	0.03	−0.03	0.01	**−0.41**	1.00	...
$\frac{\partial Ts}{\partial E_m 85H}$	−0.01	0.18	**−0.41**	0.17	0.06	0.03	−0.12	**−0.31**	0.26	1.00

[a]Correlations with absolute value higher than 0.3 are in bold.

satisfactory in this case: some high sensitivities are present but they are all significant to the 5% confidence interval. The structure of this sensitivity matrix is interesting and really illustrates the way the NN connects inputs and outputs together. For example, the first output component is related to the first input component (0.81 sensitivity value) but also the third input component (0.51). This shows that the PCA components are not the same in output and in input so that the NN needs to nonlinearly transform the input component to retrieve the output ones. With increasing output component number, the input component number used increases too. However, higher-order input components (more than 5) have limited impact. Even if the mean sensitivity is low, it does not mean that the input component has no impact on the retrieval for some situations. The nonlinearity of the NN allows it to have a situation-dependency of the sensitivities so that a particular input component can be valuable for some particular situations.

[45] Using equation (6), we obtain the corresponding Jacobian matrix $\left[\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right]$ for the physical variables instead of the PCA components, but normalized as discussed in section 4.2 to be able to compare individual sensitivities (Table 5). The uncertainty of the sensitivities is now very low and most of the mean sensitivities are significant to the 5% level. This demonstrates that the PCA regularization has solved, at least partially, the problem of Jacobian uncertainty, by suppressing the multicolinearities in the statistical regression. Interferences among variables are suppressed and the standard deviations calculated for each neural sensitivity are very small, as compared to the values previously estimated without regularization (Table 2). In addition, the sensitivities make more sense physically, as expected.

[46] The retrieved *Ts* is very sensitive to the brightness temperatures at vertical polarizations for the lower frequencies (see number in bold in the corresponding column). The emissivities being close to one for the vertical polarization (and higher than for the horizontal polarization), *Ts* is almost proportional to *TB* in window channels (i.e., those that are not affected by water vapor). Sensitivity to the *Ts* first guess is also rather high, but associated with a higher standard deviation. Sensitivities to the first guess emissivities are weak, regardless of frequency and polarization. *WV* information clearly comes from the 85 GHz horizontal polarization channel. It is worth emphasizing on the fact that the sensitivity of *WV* to *TB*85*H* is almost twice as large as to the *WV* first guess, meaning that real pertinent information is extracted from this channel. Sensitivity of the retrieved emissivities to the inputs strongly depends on the polarization, the vertical polarization emissivities being more directly related to *Ts* and *TBV* given their higher values generally close to one. Emissivities in vertical polarization are essentially sensitive to *Ts* and to the *TBV*, whereas the emissivities in the horizontal polarization are dominated

Table 4. Global Mean Neural Sensitivities $\frac{\partial \mathbf{y}'}{\partial \mathbf{x}'}$ of Raw Network Output and Input[a]

	NN Outputs				
NN Inputs	Compo 1[b]	Compo 2	Compo 3	Compo 4	Compo 5
Compo 1	**−0.81** ± 0.01	−0.25 ± 0.01	**0.53** ± 0.01	−0.05 ± 0.01	−0.03 ± 0.01
Compo 2	−0.21 ± 0.01	**−0.69** ± 0.01	**−0.62** ± 0.01	0.16 ± 0.01	0.10 ± 0.02
Compo 3	**0.51** ± 0.01	**−0.65** ± 0.01	**0.41** ± 0.01	**0.47** ± 0.01	0.02 ± 0.01
Compo 4	0.17 ± 0.01	**−0.46** ± 0.01	0.05 ± 0.01	**−0.44** ± 0.01	−0.07 ± 0.01
Compo 5	−0.06 ± 0.01	0.04 ± 0.01	−0.00 ± 0.01	−0.02 ± 0.01	**0.77** ± 0.04
Compo 6	−0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.07 ± 0.01
Compo 7	−0.01 ± 0.01	0.02 ± 0.01	−0.01 ± 0.01	−0.02 ± 0.01	0.10 ± 0.01
Compo 8	−0.03 ± 0.01	−0.10 ± 0.01	0.01 ± 0.01	−0.04 ± 0.01	−0.05 ± 0.01
Compo 9	0.01 ± 0.01	0.02 ± 0.01	−0.01 ± 0.01	−0.00 ± 0.01	−0.25 ± 0.01
Compo 10	−0.01 ± 0.01	0.03 ± 0.01	0.00 ± 0.01	0.02 ± 0.01	0.12 ± 0.01
Compo 11	−0.14 ± 0.01	0.22 ± 0.01	−0.01 ± 0.01	0.05 ± 0.01	0.25 ± 0.01
Compo 12	0.10 ± 0.01	−0.18 ± 0.01	0.10 ± 0.01	0.08 ± 0.01	−0.14 ± 0.01

[a]Columns are network outputs, $\mathbf{y}'$, and rows are network inputs, $\mathbf{x}'$. Sensitivities with absolute value higher than 0.3 are in bold.
[b]Compo refers to PCA component.

Table 5. Global Mean Regularized Neural Sensitivities $\frac{\partial y}{\partial \boldsymbol{x}}$ (Columns are Network Outputs, ***y***, and Rows are Network Inputs, ***x***)

	Ts	*WV*	E_m19V	E_m19H	E_m22V	E_m37V	E_m37H	E_m85V	E_m85H
TB19V	0.23 ± 0.02	−**0.52** ± 0.02	0.06 ± 0.00	−0.00 ± 0.00	0.06 ± 0.00	0.04 ± 0.00	−0.01 ± 0.00	−0.02 ± 0.00	−0.03 ± 0.00
TB19H	0.06 ± 0.01	0.14 ± 0.01	0.00 ± 0.00	0.03 ± 0.00	−0.00 ± 0.00	−0.01 ± 0.00	0.03 ± 0.00	−0.01 ± 0.00	0.02 ± 0.00
TB22V	0.21 ± 0.01	−**0.34** ± 0.01	0.05 ± 0.00	−0.01 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	−0.01 ± 0.00	−0.01 ± 0.00	−0.02 ± 0.00
TB37V	0.21 ± 0.01	−**0.27** ± 0.01	0.04 ± 0.00	−0.01 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	−0.01 ± 0.00	0.00 ± 0.00	−0.02 ± 0.00
TB37H	0.06 ± 0.01	0.28 ± 0.01	−0.02 ± 0.00	0.02 ± 0.00	−0.01 ± 0.00	−0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.02 ± 0.00
TB85V	0.12 ± 0.01	**0.39** ± 0.01	−0.02 ± 0.00	−0.01 ± 0.00	−0.02 ± 0.00	−0.00 ± 0.00	−0.01 ± 0.00	0.04 ± 0.00	0.01 ± 0.00
TB85H	0.01 ± 0.02	**0.80** ± 0.02	−0.06 ± 0.00	0.01 ± 0.00	−0.05 ± 0.00	−0.03 ± 0.00	0.01 ± 0.00	0.04 ± 0.00	0.04 ± 0.00
Ts	0.20 ± 0.02	−0.18 ± 0.02	−0.04 ± 0.00	−0.01 ± 0.00	−0.05 ± 0.00	−0.05 ± 0.00	−0.01 ± 0.00	−0.04 ± 0.00	−0.01 ± 0.00
WV	−0.06 ± 0.01	**0.42** ± 0.01	0.01 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	−0.00 ± 0.00	−0.01 ± 0.00	−0.02 ± 0.00	−0.01 ± 0.00
E_m19V	−0.09 ± 0.01	0.09 ± 0.01	0.03 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
E_m19H	−0.07 ± 0.02	−0.08 ± 0.02	0.03 ± 0.00	0.05 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.04 ± 0.00	−0.04 ± 0.00	0.01 ± 0.00
E_m22V	−0.07 ± 0.01	0.05 ± 0.01	0.02 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
E_m37V	−0.08 ± 0.01	0.02 ± 0.01	0.02 ± 0.00	0.00 ± 0.00	0.02 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.01 ± 0.00
E_m37H	−0.08 ± 0.02	−0.13 ± 0.02	0.02 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.02 ± 0.00
E_m85V	−0.05 ± 0.01	−0.06 ± 0.01	0.01 ± 0.00	−0.00 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.05 ± 0.00	0.02 ± 0.00
E_m85H	−0.05 ± 0.02	−0.16 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.04 ± 0.00	0.03 ± 0.00
Tlay	−0.03 ± 0.02	0.11 ± 0.02	−0.00 ± 0.00	−0.01 ± 0.00	−0.00 ± 0.00	−0.01 ± 0.00	−0.01 ± 0.00	−0.01 ± 0.00	−0.01 ± 0.00

Sensitivities with absolute value higher than 0.3 are in bold. The first part of this table is for SSM/I observations, the second part corresponds to first guesses.

by the emissivity first guess. The sensitivity matrix clearly illustrates how the NN extracts the information from the inputs to derive the outputs. In Figure 9, a sample of sensitivities from the predictive distribution is presented along with the distribution of these sensitivities. The comparison with similar plots in Figure 8 confirms the robustness of the Jacobians from the regularized NN.

[47] Experiments (not shown) establish that such PCA-regularized NNs have robust Jacobians even when the NN architecture is changed, for example with a different number of neurons in hidden layer. This shows how robust and reliable the new NN Jacobians and the NN model have become with the help of the PCA representation regularization.

[48] The sensitivity matrix is not only an efficient tool to help understand the NN inversion procedure, it can also help refine the inversion method, for instance by identifying inputs that would not significantly contribute to the retrieval (inputs for which all sensitivities are very low) [*Rivals and Personnaz*, 2003].

5. Conclusion and Perspectives

[49] The Jacobians of a nonlinear model, such as the NN, are a very powerful concept. In terms of the NN model, it allows us to obtain a robust model that will generalize well and suffer less from over-fitting deficiencies. Jacobians can also be used to analyze how the NN model links the inputs and the outputs, in a nonlinear way. This capacity is especially important when the NN as an analysis tool as proposed by *Aires and Rossow* [2003].

[50] In this paper, we show how to estimate the Jacobians of a nonlinear regression model, in particular for a NN model. New tools are provided to check how robust and stable these Jacobians are by estimating their uncertainty PDF by Monte Carlo simulations. A tool is provided to identify situations where regularization needs to be used. As it is often the case, regularization is a fundamental step of NN learning, especially for inverse problems [*Badeva and Morosov*, 1991; *Tikhonov and Arsenin*, 1977]. We propose a regularization method based on the PCA regression (using a PCA representation of input and output data for the NN) to suppress the problem of multicoilnearities in data. Our approach is able to make the learning process more stable, the Jacobians more reliable and can be more easily interpreted physically. All these tools are very general and can be used for other nonlinear models of statistical inference.

[51] Our work provides a framework for the characterization, the analysis, and the interpretation of Jacobians and their uncertainties in any neural network-based retrieval scheme. A large range of applications can benefit from such Jacobian estimates in meteorology and climatology. New technical developments can be pursued to solve additional problems. For example, we proposed in the work of *Aires et al.* [2004] to use the NN Jacobians to analyze in even more detail the different sources of uncertainty in the NN outputs (e.g., instrument noise, direct model errors). This could use the approach of *Rodgers* [1990] presented for classical remote sensing techniques such as variational assimilation.

[52] Another domain of application is the next generation of satellite sounders like IASI or AIRS. It would be particularly interesting to estimate the Jacobians of the radiative transfer equation (direct and inverse models) to characterize vertical resolution results for atmospheric temperature, water vapor, and ozone.

[53] Analysis of dynamical systems is another type of application. In [*Aires and Rossow*, 2003] the NN Jacobians are estimated to analyze feedback processes. The reliable estimation of physical Jacobians through the NN model is an ideal candidate for the study of climate feedback in both numerical models and observation data sets. The new ideas and techniques presented in this paper will directly benefit such studies.

Appendix A: Sampling a PDF in the High-Dimensional Space of Network Weights

[54] The simplest way of sampling a PDF is to define a multidimensional regularly spaced grid on the weight space. The probability of weights ***w*** is then estimated at each grid points. Clearly, this strategy becomes very inefficient when the number of network weights becomes too large. In the following application, we see that the number of weights involved is 819 and can be as large as tens of thousands. A

regular sampling in M intervals for each coordinate would require M^{819} samples!

[55] An alternative is to use the Cholesky decomposition of the covariance matrix [*Press et al.*, 1992] $\boldsymbol{H}^{-1} = \boldsymbol{L} \cdot \boldsymbol{L}^T$. (Here $\boldsymbol{H}^{-1}$ needs to be positive definite; see *Aires* [2004] for regularization techniques that make this matrix positive definite.) This can be used to sample the PDF: a random weight sample $\boldsymbol{w}^r$ is defined by

$$\boldsymbol{w}^r = \boldsymbol{L} \cdot \boldsymbol{r}, \tag{A1}$$

where $\boldsymbol{r}$ is a vector of normalized Gaussian random numbers. This approach is simple and elegant, but it can still be quite laborious since for each sample $\boldsymbol{w}^r$, a total of W (i.e., the dimension of $\boldsymbol{w}$) Gaussian random numbers needs to be calculated which is very time-consuming.

[56] Eigen-value decomposition can avoid this time-consuming problem. It uses the decomposition of the covariance matrix $\boldsymbol{H}^{-1}$ into its eigen-vectors:

$$\boldsymbol{H}^{-1} \cdot \boldsymbol{f}_i = s_i \cdot \boldsymbol{f}_i. \tag{A2}$$

The vectors $\boldsymbol{f}_i$, columns of $W \times W$ matrix $\boldsymbol{F}$, are the eigen-vectors and the scalars s_i are the eigen-values of matrix $\boldsymbol{H}^{-1}$. Then, each Gaussian sample is defined as:

$$\boldsymbol{w}^r = \boldsymbol{F} \cdot \boldsymbol{S}^r, \tag{A3}$$

where $\boldsymbol{S}^r$ is a diagonal matrix whose elements are drawn from a Gaussian distribution of mean zero and variance one. To reduce the time, one can use only the first few, more important, eigen-vectors and draw only a few random numbers in $\boldsymbol{S}^r$. This truncation makes this technique much faster. The compromise is that we lose some of the variability implied by higher-order eigen-vectors.

[57] Stochastic methods [*Duflo*, 1996] are particularly interesting for sampling high-dimension PDFs. Markov Chain Monte Carlo (MCMC) algorithms are good candidates [*Gelman et al.*, 1995]. These techniques are based on the idea that it is not useful to sample the space of parameters $\boldsymbol{w}$ where the distribution $P(\boldsymbol{w})$ is very small. Markov Chain Monte Carlo methods (like a random walk) are designed to sample mostly the significant part of the parameter space $\boldsymbol{w}$. The metropolis algorithm is one of these methods:

[58] 1. Modify $\boldsymbol{w}_{old}$ to $\boldsymbol{w}_{new} = \boldsymbol{w}_{old} + \Delta\boldsymbol{w}$ using a jump model $P(\Delta\boldsymbol{w})$.

[59] 2. If $P(\boldsymbol{w}_{new}|\mathcal{D}) > P(\boldsymbol{w}_{old}|\mathcal{D})$ accept.

[60] 3. If $P(\boldsymbol{w}_{new}|\mathcal{D}) < P(\boldsymbol{w}_{old}|\mathcal{D})$ accept with probability $\frac{P(\boldsymbol{w}_{new}|\mathcal{D})}{P(\boldsymbol{w}_{old}|\mathcal{D})}$.

[61] 4. Go to (1) until the number of samples $\boldsymbol{w}$ is sufficient, or a stopping criterion is satisfied.

[62] A very important feature of MCMC algorithms is that they do not require the evaluation of the normalization term of the PDF (which can be extremely complex to estimate for non-Gaussian distributions). This gives MCMC methods the huge advantage of being able to deal with non-Gaussian distributions where eigen and Cholesky decompositions are designed for Gaussian distributions only.

[63] As sophisticated as the optimization algorithm used during the learning stage is, the process can still be trapped in local minima $\boldsymbol{w}^\star$. It is well-known that many local minima exist in the quality criterion for the optimization of a neural network. For example, the permutation of all the neurons in the hidden layer would not change the results in the network outputs (i.e., same value for the criterion) but would change radically the weight vector. This proves that there exists a large number of local minima with equivalent criterion values. This does not affect the quality of the network but increases artificially the variability of weights. This is the reason why it can be necessary to integrate over different local minima. One approach to do that is using multiple learnings of the neural network, starting from different initial conditions for the network weights. For example, bootstrap methods take a different part of the learning database for each of the learnings. The various learning results estimate different final network weights $\boldsymbol{w}^\star$. This approach is perfectly valid, but it has a drawback in the neural network context: the learning step is computationally an intensive process. It is thus extremely difficult, or even unrealistic, to try to estimate the a posteriori distribution of the network weights based on such a scheme. However, it should be noted that it is possible to use a combination of the bootstrap and the metropolis methods. This will be the subject of a further study.

Notation

$\boldsymbol{y}$	vector of physical variables to retrieve, outputs of the NN.
$\boldsymbol{M}$	dimension of vector of $\boldsymbol{y}$, outputs of the NN.
$\boldsymbol{t}$	target vector of physical variables in data set $\mathcal{B}$.
$\boldsymbol{y}^b$	first guess a priori information for $\boldsymbol{x}$.
$\boldsymbol{x}$	observations vector, inputs of the NN.
$\boldsymbol{K}$	dimension of vector of $\boldsymbol{x}$, inputs of the NN.
ε_v	generic error symbol for variable v.
P_v	generic probability measure of variable v.
$\boldsymbol{C_x}$	covariance matrix of NN inputs $\boldsymbol{x}$.
$\boldsymbol{C_y}$	covariance matrix of NN outputs $\boldsymbol{y}$.
$\boldsymbol{H}$	$= \nabla\|_{\boldsymbol{w}}\ (\nabla\|_{\boldsymbol{w}}\ (E_{\mathcal{D}}(\boldsymbol{w})))$, the Hessian matrix of the log likelihood.
$\boldsymbol{S}_{1x}$	the diagonal matrices which diagonal terms are the standard deviation of input variable $\boldsymbol{x}$.
$\boldsymbol{S}_{1y}$	the diagonal matrices which diagonal terms are the standard deviation of output variable $\boldsymbol{y}$.
$\boldsymbol{m}_{1x}$	mean vector of input variable $\boldsymbol{x}$.
$\boldsymbol{m}_{1y}$	mean vector of output variable $\boldsymbol{y}$.
$\boldsymbol{x}'$	PCA representation of observations $\boldsymbol{x}$.
$\boldsymbol{y}'$	PCA representation of physical variables $\boldsymbol{y}$.
$\boldsymbol{S}_{2x}$	the diagonal matrix whose diagonal terms are the standard deviation of input PCA representation $\overline{\boldsymbol{x}}$.
$\boldsymbol{S}_{2y}$	the diagonal matrix whose diagonal terms are the standard deviation of output PCA representation $\overline{\boldsymbol{y}}$.
$\boldsymbol{m}_{2x}$	mean vector of PCA representation PCA $\boldsymbol{x}$.
$\boldsymbol{m}_{2y}$	mean vector of PCA representation PCA $\boldsymbol{y}$.
$\boldsymbol{r}$	vector of normalized Gaussian random numbers.
$\boldsymbol{S}^r$	diagonal matrix whose elements are drawn from a Gaussian distribution of mean zero and variance one.
$\boldsymbol{F}_v$	eigen-vector matrix associated to covariance matrix of vector v, with columns equal to $\boldsymbol{f}_i$.
$\overline{\cdot}$	truncation operator.

$.^T$ transposition operator.
a_i activity of neuron i.
σ sigmoid function of the neural network.
S_i number of neurons in network layer i.
$g_{\mathbf{w}}$ neural network model, or transfer function for our application.
$\mathbf{w}$ $\{w_i;\ i = 1, \ldots, W\}$, the vector of the network weights.
W dimension of $\mathbf{w}$.
R number of samples in $\{\mathbf{w}^r;\ r = 1, \ldots, R\}$, the sample of network weights.
$\mathcal{B}$ learning database, that includes outputs $\mathcal{D}$.
$\mathcal{D}$ target or network output database.
$E_{\mathcal{D}}(\mathbf{w})$ data term of the quality criterion.
$E_r(\mathbf{w})$ regularization term in the quality criterion.
λ regularization factor for the inversion of $\boldsymbol{H}$.

[64] **Acknowledgments.** We would like to thank Ian T. Nabney for providing the Netlab toolbox from which some of the routines have been used in this work. Filipe Aires would like to thank Andrew Gelman for very interesting discussion about modern Bayesian statistics. This work was partly supported by NASA Radiation Sciences and Hydrology Programs.

References

Aires, F. (1999), Problèmes inverses et réseaux de neurones: Application à l'interféromètre haute résolution IASI et à l'analyse de séries temporelles, Ph.D. thesis, 220 pp., Paris IX-Dauphine, Univ. Paris, Paris.

Aires, F. (2004), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 1. Network weights, *J. Geophys. Res.*, *109*, D10303, doi:10.1029/2003JD004173.

Aires, F., and W. B. Rossow (2003), Inferring instantaneous, multivariate and nonlinear sensitivities for the analysis of feedback processes in a dynamical system: The Lorenz model case study, *Q. J. R. Meteorol. Soc.*, *129*, 239–275.

Aires, F., M. Schmitt, N. A. Scott, and A. Chédin (1999), The weight smoothing regularisation for MLP for resolving the input contribution's errors in functional interpolations, *IEEE Trans. Neural Networks*, *10*, 1502–1510.

Aires, F., A. Chédin, and J.-P. Nadal (2000), Independent component analysis of multivariate times series: Application to the tropical SST variability, *J. Geophys. Res.*, *105*(D13), 17,437–17,455.

Aires, F., C. Prigent, W. B. Rossow, and M. Rothstein (2001), A new neural network approach including first guess for retrieval of atmospheric water vapor, cloud liquid water path, surface temperature and emissivities over land from satellite microwave observations, *J. Geophys. Res.*, *106*(D14), 14,887–14,907.

Aires, F., W. B. Rossow, N. A. Scott, and A. Chédin (2002a), Remote sensing from the infrared atmospheric sounding interferometer instrument: 1. Compression, denoising, first-guess retrieval inversion algorithms, *J. Geophys. Res.*, *107*(D22), 4619, doi:10.1029/2001JD000955.

Aires, F., W. B. Rossow, N. A. Scott, and A. Chédin (2002b), 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), 4620, doi:10.1029/2001JD001591.

Aires, F., A. Chédin, N. A. Scott, and W. B. Rossow (2002c), A regularized neural network approach for retrieval of atmospheric and surface temperatures with the IASI instrument, *J. Appl. Meteorol.*, *41*, 144–159.

Aires, F., W. B. Rossow, and A. Chédin (2002d), Rotation of EOFs by the independent component analysis: Towards a solution of the mixing problem in the decomposition of geophysical time series, *J. Atmos. Sci.*, *59*(1), 111–123.

Aires, F., C. Prigent, and W. B. Rossow (2004), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 2. Output errors, *J. Geophys. Res.*, *109*, D10304, doi:10.10292003JD004174.

Badeva, V., and V. Morosov (1991), *Problèmes Incorrectement Posés*, Masson, Paris.

Bishop, C. (1996), *Neural Networks for Pattern Recognition*, 482 pp., Clarendon Press, Oxford, UK.

Duflo, M. (1996), *Algorithmes Stochastiques: Mathématiques et Applications*, Springer-Verlag, New York.

Gelman, A. B., J. S. Carlin, H. S. Stern, and D. B. Rubin (1995), *Bayesian Data Analysis*, Chapman and Hall, New York.

Geman, S., E. Bienenstock, and R. Doursat (1992), Neural networks and the bias-variance dilemma, *Neural Comput.*, *1*(4), 1–58.

Jolliffe, I. T. (2002), *Principal Component Analysis*, 2nd ed., 487 pp., Springer-Verlag, New York.

MacKay, D. J. C. (1992), A practical Bayesian framework for back-propagation networks, *Neural Comput.*, *4*(3), 448–472.

Nabney, I. T. (2002), *Netlab: Algorithms for Pattern Recognition*, Springer-Verlag, New York.

Neal, R. M. (1996), *Bayesian Learning for Neural Networks*, Springer-Verlag, New York.

Press, W. H. P., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992), *Numerical Recipes in Fortran*, Cambridge Univ. Press, New York.

Rivals, I., and L. Personnaz (2003), MLPs (mono-layer polynomials and multi-layer perceptrons) for nonlinear modeling, *J. Machine Learning Res.*, *3*, 1383–1398.

Rodgers, C. D. (1990), Characterization and error analysis of profiles retrieved from remote sounding measurements, *J. Geophys. Res.*, *95*, 5587–5595.

Tarantola, A. (1987), *Inverse Problem Theory: Models for Data Fitting and Model Parameter Estimation*, 613 pp., Elsevier Sci., New York.

Tikhonov, A., and V. Arsenin (1977), *Solutions of Ill-Posed Problems*, V. H. Vinsten, Washington, D. C.

Vapnik, V. (1997), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

F. Aires, Department of Applied Physics and Applied Mathematics, Columbia University/NASA Goddard Institute for Space Studies, 2880 Broadway, New York, NY 10025, USA. (faires@giss.nasa.gov)

C. Prigent, CNRS, LERMA, Observatoire de Paris, 61, av. de l'Observatoire, Paris F-75014, France. (catherine.prigent@obspm.fr)

W. B. Rossow, NASA Goddard Institute for Space Studies, 2880 Broadway, New York, NY 10025, USA. (wrossow@giss.nasa.gov)