Neural Network Uncertainty Assessment Using Bayesian Statistics: A Remote Sensing Application

F. Aires
faires@giss.nasa.gov
Department of Applied Physics and Applied Mathematics, Columbia University, NASA Goddard Institute for Space Studies, New York, NY 10025, U.S.A., and CNRS/IPSL/ Laboratoire de Météorologie Dynamique, École Polytechnique, 91128 Palaiseau Cedex, France

C. Prigent
catherine.prigent@obspm.fr
CNRS, LERMA, Observatoire de Paris, Paris 75014, France

W.B. Rossow
wrossow@giss.nasa.gov
NASA Goddard Institute for Space Studies, New York, NY 10025, U.S.A.

Neural network (NN) techniques have proved successful for many regression problems, in particular for remote sensing; however, uncertainty estimates are rarely provided. In this article, a Bayesian technique to evaluate uncertainties of the NN parameters (i.e., synaptic weights) is first presented. In contrast to more traditional approaches based on point estimation of the NN weights, we assess uncertainties on such estimates to monitor the robustness of the NN model. These theoretical developments are illustrated by applying them to the problem of retrieving surface skin temperature, microwave surface emissivities, and integrated water vapor content from a combined analysis of satellite microwave and infrared observations over land.

The weight uncertainty estimates are then used to compute analytically the uncertainties in the network outputs (i.e., error bars and correlation structure of these errors). Such quantities are very important for evaluating any application of an NN model.

The uncertainties on the NN Jacobians are then considered in the third part of this article. Used for regression fitting, NN models can be used effectively to represent highly nonlinear, multivariate functions. In this situation, most emphasis is put on estimating the output errors, but almost no attention has been given to errors associated with the internal structure of the regression model. The complex structure of dependency inside the NN is the essence of the model, and assessing its quality, coherency, and physical character makes all the difference between a blackbox model

with small output errors and a reliable, robust, and physically coherent model. Such dependency structures are described to the first order by the NN Jacobians: they indicate the sensitivity of one output with respect to the inputs of the model for given input data. We use a Monte Carlo integration procedure to estimate the robustness of the NN Jacobians. A regularization strategy based on principal component analysis is proposed to suppress the multicollinearities in order to make these Jacobians robust and physically meaningful.

1 Introduction

Neural network (NN) techniques have proved very successful in developing computationally efficient algorithms for geophysical applications. We are interested, in this study, in the application of the NN retrieval methods for satellite remote sensing (Aires, Rossow, Scott, & Chédin, 2002a): the NN is used as a nonlinear multivariate regression to represent the inverse radiative transfer function in the atmosphere. This is an application of the inverse theory: remote sensing requires the estimation of geophysical variables from indirect measurements by applying the inverse radiative transfer function to radiative measurements. NN are well adapted to solve nonlinear problems and are especially designed to capitalize more completely on the inherent statistical relationships among the input and output variables.

A rigorous statistical approach requires not only a minimization of output errors, but also an uncertainty estimate of the model parameters (Saltelli, Chan, & Scott, 2000). The reliability of the inverse model is as important as its answer, but until now, probably because of the lack of adequate tools, the uncertainty of an NN statistical model has rarely been quantified. Our work is based on the developments of Le Cun, Denker, and Sola (1990) and MacKay (1992). These studies introduced error bar estimates for neural networks using a Bayesian approach, but these tools were developed and tested in simple cases for a unique network output. In this article, we use a slightly different approach than the more traditional full Bayesian method, where scalar hyperparameters are estimated using the so-called evidence approach. A multiple output method is used in order to develop uncertainty tools for real-world applications. Our Bayesian methodology provides first uncertainty estimates for the parameters of the neural network (i.e., the network weights). A similar approach is, but in a simpler presentation (a monovariate case), used, for example, in Bishop (1996), Neal (1996), and Nabney (2002). The robustness of the NN parameters is assessed using the Hessian matrix (second derivative) of the log likelihood with respect to the NN weights.

Uncertainty estimates for the parameters of the neural network can then be used for the determination of a variety of other probabilistic quantities related to the overall stochastic character of the NN model. Such possible applications can use theoretical derivations when they are available. In this article, one such analytical application provides uncertainty estimates of the network output (error bars plus their correlation structure). Reliability of the NN predictions is very important for any application. Confidence intervals (CI) have been developed for classical linear regression theory with well-established results (e.g., Koroliouk, Portenko, Skorokhod, & Tourbine, 1983). For nonlinear models, such results are more recent (Bates & Watts, 1988), and in NN they are rarely available. Generally, only the root mean square (RMS) of the generalization error is provided, but this single quantity is not situation dependent. Other approaches use bootstrap techniques to estimate such CI, but they are limited by the large number of computations that such techniques require. Recently, Rivals and Personnaz (2000, 2003) introduced a new method for estimating CI by using a linear Taylor expansion of the NN outputs (which makes traditional estimation of CI for nonlinear models a tractable problem). In this article, we separate the errors that are due to the NN weight uncertainty and the errors from all remaining sources. Such additional sources of uncertainty can be, for example, noise in the inputs of the NN (Wright, Ramage, Cornford, & Nabney, 2000). We will comment on an approach to analyze in even more detail the various contributions to output errors. These errors are described in terms of covariance matrices that can be interpreted using eigenvectors called error patterns (Rodgers, 1990).

When a theoretical derivation is too complex to be obtained, another possible application of weight uncertainty is the empirical estimation of probabilistic quantities. Modern Bayesian statistics are used here together with Monte Carlo (MC) simulations (Gelman, Carlin, Stern, & Rubin, 1995) to estimate uncertainties on the NN Jacobian. These Jacobians, or sensitivities, of a NN model are defined as the partial first derivatives of the model outputs with respect to its inputs. These quantities are very useful. However, the NN 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 an ill-posed inverse problem (Tarantola, 1987; Vapnik, 1997; Aires, Schmitt, Scott, & Chédin, 1999): many solutions can be found for the NN parameters (i.e., the synaptic weights) for similar output statistics. One of the reasons for this nonunique solution comes from the fact that multicollinearities can exist among the variables. Such correlations on input or output variables are a major problem even for linear regressions: the 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 similar behavior is expected: when multicollinearities 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 article is to investigate this problem, analyze it, and suggest a solution.

Many regularization techniques exist to reduce the number of degrees of freedom in the model for the multicollinearity problem or for any other ill-posed problem. For example, one approach is to reduce the number of inputs to the NN (Rivals & 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., 2002a; Aires, Rossow, Scott, & Chédin, 2002b) 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 multicollinearities by input selection. We propose to solve this nonrobustness by using a principal component analysis (PCA) regression approach.

Our technological developments are illustrated by application to an NN inversion algorithm for remote sensing over land. Such NN methods have already been used to retrieve columnar water vapor, liquid water, or wind speed over ocean using special sensor microwave/imager observations (Stogryn, Butler, & Bartolac, 1994; Krasnopolsky, Breaker, & Gemmill, 1995; Krasnopolsky, Gemmill, & Breaker, 2000). Our algorithm includes for the first time the use of a first guess to retrieve the surface skin temperature Ts, the integrated water vapor content WV, the cloud liquid water path LWP, and the microwave land surface emissivities E_m between 19 and 85 GHz from SSM/I and infrared observations.

Neural network techniques have proved very successful in developing computationally efficient algorithms for remote sensing (e.g., Aires et al., 2002b), but uncertainty estimates on the retrievals have been a limiting factor for the use of such methods. Our technical developments on this remote sensing application provide a new framework for the characterization and the analysis of various sources of neural network errors. Estimation of the Jacobian uncertainties is then used as a diagnostic tool to identify nonrobust regressions, resulting from unstable learning processes.

The Bayesian approach for the estimation of NN weight uncertainty is presented in section 2. The NN technique is described, the theoretical formulation of a posteriori distributions for the NN weights is developed, and a remote sensing application is presented as an example to illustrate some first results of the application of our weight uncertainty analysis. The theoretical computation of the predictive distribution of network outputs is developed in section 3. Theoretical developments are used to characterize the NN output uncertainty sources. Section 4 presents the uncertainty estimate of NN Jacobians. The PCA of the NN input and output data is described together with its regularization properties. Network Jacobians are presented with their corresponding uncertainties. Conclusions and perspectives are discussed in section 5.

2 Network Weights Uncertainty

2.1 The Quality Criterion. In this study, we use a classical multilayer perceptron (MLP) trained by backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986). For the definition of the quality criterion to maximize, we present a general matrix formulation of the problem and link our derivation to the "classical" literature on Bayesian error estimation often introduced with a scalar formulation (MacKay, 1992; Bishop, 1996). The first and main term in the quality criterion used to train a neural network is the "data" term, expressed using the difference between the target data and the NN estimates as measured by a particular distance. Many distance measures can be used, but it is often supposed that the differences follow a gaussian probability distribution function (PDF), which means that the right distance is the Mahalanobis distance (Crone & Crosby, 1995). The ideal covariance matrix for the gaussian PDF, denoted $\boldsymbol{C}_{in} = \boldsymbol{A}_{in}{}^{-1}$, describes what we call the intrinsic noise (or natural variability) of the physical variables $\boldsymbol{y}$ to retrieve. Note that $\boldsymbol{C}_{in}$ takes into account only the intrinsic variability and not the error associated with the retrieval scheme itself: this makes this measure coherent physically. The information encoded in $\boldsymbol{C}_{in}$ is difficult to obtain a priori; we will see how to estimate this quantity, but we suppose here that it is known. The data quality term becomes

$$E_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N} [\varepsilon_y{}^{(n)}]^T \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y{}^{(n)}, \tag{2.1}$$

where $\varepsilon_y{}^{(n)} = (\boldsymbol{t}^{(n)} - \boldsymbol{y}^{(n)})$ is the output error and the index $^{(n)}$ indicates the sample number in database $\mathcal{B}$. This criterion leads to a weighted least squares when the matrix $\boldsymbol{C}_{in}$ is just diagonal. When no a priori information is available, $\boldsymbol{C}_{in} = \boldsymbol{I}$, and the criterion becomes the classical least squares.

In order to regularize the learning process, a regularization term is sometimes added to the data term in the quality criterion. The weight decay (Hertz, Krogh, & Palmer, 1991) is probably the most common regularization technique for NN:

$$E_r(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A}_r \cdot \boldsymbol{w}, \tag{2.2}$$

where $\boldsymbol{C}_r = \boldsymbol{A}_r{}^{-1}$ is the covariance matrix of the gaussian a priori distribution for the network weights.

The overall quality criterion that is minimized during the learning stage is the sum of the data and the regularization terms,

$$E(\boldsymbol{w}) = E_{\mathcal{D}}(\boldsymbol{w}) + E_r(\boldsymbol{w}). \tag{2.3}$$

The two matrices $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$ are called hyperparameters. They are generally simplified in the classical literature by using two scalars instead, respectively, β and α, so that the general quality criterion becomes $E(\boldsymbol{w}) = \beta E_{\mathcal{D}} + \alpha E_r$, where $E_{\mathcal{D}}$ and E_r are simplified quadratic forms. In this formulation, β represents the inverse of the observation noise variance for all outputs and α is a weight for the regularization term linked to the a priori general variance of the weights. This is obviously poorer and less general than our matrix formulation in equation 2.3, but the hyperparameters $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$ are difficult to guess a priori.

2.2 Intrinsic Uncertainty of Targets. The conditional probability $P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})$ represents the variability of target $\boldsymbol{t}$ for input $\boldsymbol{x}$ and network weights $\boldsymbol{w}$, due to a variety of sources like the errors in the model linking $\boldsymbol{x}$ to $\boldsymbol{t}$ in $\mathcal{B}$ or the observational noise on $\boldsymbol{x}$. This variability includes all sources of uncertainty except those from the NN regression model, represented by uncertainties on the network weights $\boldsymbol{w}$, that are fixed in the conditional probability.

If the neural network $g_{\boldsymbol{w}}$ fits the data well (after the learning stage), the intrinsic variability is evaluated by comparing the target values, $\boldsymbol{t}$, matched with each input $\boldsymbol{x}$ in the data set $\mathcal{B}$ to the NN outputs $\boldsymbol{y}$. Generally, this distribution can be approximated locally to first order by a gaussian distribution with zero mean and a covariance matrix $\boldsymbol{C}_{in} = {\boldsymbol{A}_{in}}^{-1}$:

$$P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z} e^{-\frac{1}{2} {\varepsilon_y}^T \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y}, \tag{2.4}$$

where Z is a normalization factor. The likelihood of the parameters $\boldsymbol{w}$, given the inverse model structure g of the trained NN $g_{\boldsymbol{w}}$, is expressed by evaluating this probability over the database $\mathcal{B}$ that includes $\mathcal{D} = \{\boldsymbol{t}^{(n)}\ ;\quad n = 1, \ldots, N\}$, the set of output samples,

$$P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w}) = \prod_{n=1}^{N} P(\boldsymbol{t}^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{w}) = \frac{1}{Z^N} e^{-\frac{1}{2} \sum_{n=1}^{N} {\varepsilon_y}^{(n)T} \cdot \boldsymbol{A}_{in} \cdot {\varepsilon_y}^{(n)}}, \tag{2.5}$$

that we simplify by

$$P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z^N} e^{-E_{\mathcal{D}}}, \tag{2.6}$$

using the defintion of $E_{\mathcal{D}}$ in equation 2.1. The smaller $E_{\mathcal{D}}$ is, the likelier the output data sample $\mathcal{D}$ is (i.e., the closer all $\boldsymbol{y}$ are to target $\boldsymbol{t}$). The conditioning of the previous probabilities as in equation 2.5 is dependent on the input $\boldsymbol{x}$, but since the distribution of $\boldsymbol{x}$ is not of interest here, this variable will be omitted in the following notation for simplicity.

2.3 Theoretical Derivation of Weight PDF. In classical regression techniques, a point estimate of the parameters $\boldsymbol{w}$ is searched for (i.e., only one estimate of the weight vector $\boldsymbol{w}$ is evaluated). In the Bayesian context, an uncertainty of $\boldsymbol{w}$ described by a PDF $P(\boldsymbol{w})$ can also be characterized. This distribution of the weights conditional on a database is given by the Bayes theorem:

$$P(\boldsymbol{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{w})P(\boldsymbol{w})}{P(\mathcal{D})}. \tag{2.7}$$

$P(\mathcal{D})$ does not depend on the weights, and the prior $P(\boldsymbol{w})$ is a uniform distribution in this application (since there is no prior information on $\boldsymbol{w}$), meaning that no regularization term $E_r(\boldsymbol{w})$ is used in equation 2.3. So we can use for $P(\boldsymbol{w}|\mathcal{D})$ the expression for $P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w})$ from equation 2.6, the other terms in equation 2.7 being considered as constant normalization factors.

Laplace's method is now used: it consists in using a local quadratic approximation of the log-posterior distribution. A second-order Taylor expansion of $E_{\mathcal{D}}(\boldsymbol{w}^\star)$ is performed, where $\boldsymbol{w}^\star$ is the set of the final optimized network weights (parameters of the neural network regression) found at the end of the learning process:

$$E_{\mathcal{D}}(\boldsymbol{w}) = E_{\mathcal{D}}(\boldsymbol{w}^\star) + \boldsymbol{b}^T \cdot \triangle\boldsymbol{w} + \frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}, \tag{2.8}$$

where $\triangle\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}^\star$, $\boldsymbol{b}$ is the Jacobian vector given by $\boldsymbol{b} = \triangledown|_{\boldsymbol{w}} (E_{\mathcal{D}}(\boldsymbol{w}))$, and $\boldsymbol{H}$ is the Hessian matrix given by $\boldsymbol{H} = \triangledown|_{\boldsymbol{w}} (\triangledown|_{\boldsymbol{w}} (E_{\mathcal{D}}(\boldsymbol{w})))$. The linear term $\boldsymbol{b}^T \cdot \triangle\boldsymbol{w}$ disappears because we are at the optimum $\boldsymbol{w}^\star$, which means that the gradient $\boldsymbol{b}$ is zero. For the local quadratic approximation to be valid, $\boldsymbol{w}^\star$ must be a real optimum (at least locally in the weight space), otherwise, the gradient $\boldsymbol{b}$ cannot be neglected anymore and the matrix $\boldsymbol{H}$ might not be positive definite, which will make its use difficult for subsequent uncertainty estimates.

The second-order approximation leads to

$$P(\boldsymbol{w}|\mathcal{D}) = \frac{1}{Z^N} e^{-E_{\mathcal{D}}(\boldsymbol{w}^\star) - \frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}} \propto e^{-\frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}}. \tag{2.9}$$

This means that the a posteriori PDF of the neural network weights follows a gaussian distribution with mean $\boldsymbol{w}^\star$ and covariance matrix $\boldsymbol{H}^{-1}$. This probability represents a plausibility (in the Bayesian sense) for the weight $\boldsymbol{w}$, not the probability of obtaining the weight $\boldsymbol{w}$ when using the learning algorithm.

If a regularization term, such as the one described in equation 2.2, is used, then this probability becomes:

$$P(\boldsymbol{w}|\mathcal{D}) \propto e^{-\frac{1}{2}\triangle\boldsymbol{w}^T \cdot (\boldsymbol{H} + \boldsymbol{A}_r) \cdot \triangle\boldsymbol{w}}. \tag{2.10}$$

These two terms are used to weight the contribution to the variability of the weights due to the network model and the variability of the weights due to the gaussian distribution of the a priori information on the weights. What is interesting about this formula is that to obtain the covariance matrix on the weights, we invert $\boldsymbol{H} + \boldsymbol{A}_r$ instead of $\boldsymbol{H}$ only, which is more robust (see section 2.6) since $\boldsymbol{A}_r$ is the inverse of a positive definite matrix.

2.4 Hessian Matrix for a One-Hidden-Layer Network. The Hessian, $\boldsymbol{H}$, of the previously defined log likelihood is a matrix of dimension $W \times W$ (W is the dimension of $\boldsymbol{w}$) whose components are defined by

$$H_{ij}(\boldsymbol{x}) = \left. \frac{\partial^2 E(\boldsymbol{w})}{\partial w_i \partial w_j} \right|_{\boldsymbol{x}}, \tag{2.11}$$

where w_i and w_j are two weights from the set $\boldsymbol{w}$.

There are many ways of estimating the Hessian matrix; some are generic methods, and some are specific to the MLP. For example, one generic approximation method uses finite differences, but in our case, it is possible to retrieve a mathematical expression for the Hessian based on the NN model. This theoretical Hessian is less demanding computationally—its scaling is $\mathcal{O}(W^2)$ (where W is the number of weights in the neural network)—than the previous approximation by finite differences, which scales like $\mathcal{O}(W^3)$.

2.5 A Remote Sensing Example. An NN inversion scheme has been developed to retrieve surface temperature (Ts), water vapor column amount (WV), and microwave surface emissivities at each frequency/polarization (E_m), over snow- and ice-free land from a combined analysis of satellite microwave (SSM/I) and infrared International Satellite Cloud Climatology Project (ISCCP) data (Aires, Prigent, Rossow, & Rothstein, 2001; Prigent, Aires, & Rossow, 2003). This study aims, in part, to provide uncertainty estimates for these retrievals.

To avoid nonuniqueness and instability in an inverse problem, it is essential to use all a priori information available. The chosen solution is then constrained so that it is physically more consistent (Rodgers, 1976). We introduce a priori first-guess information into the input of an NN model, so the neural transfer function becomes

$$\boldsymbol{y} = g_{\boldsymbol{w}}(\boldsymbol{y}^b, \boldsymbol{x}^\circ), \tag{2.12}$$

where $\boldsymbol{y}$ is the retrieval (i.e., retrieved physical parameters), $g_{\boldsymbol{w}}$ is the NN with parameters $\boldsymbol{w}$, $\boldsymbol{y}^b$ is the first guess for the retrieval of physical parameters, and $\boldsymbol{x}$ the observations. In this approach, the first guess is considered to be an independent estimate of the state obtained from sources other than the indirect measurements (here, the satellite observations). These are sometimes called virtual measurements (Rodgers, 1990).

The extensive learning database used in this study, together with the characteristics of the a priori first-guess information and related background errors, are presented in Aires et al. (2001). Over the 9,830,211 samples for clear, snow- and ice-free measurements from a whole year of data, we have used only N =20,000 samples, chosen randomly, to construct the learning database $\mathcal{B}$. The learning algorithm and the network architecture are able to infer the inverse radiative transfer equation with these N samples. The conjugate gradient optimization algorithm used to train the NN is fast and efficient: the learning errors decrease extremely fast and then stabilize after a few thousand iterations, each iteration involving the whole learning database $\mathcal{B}$. This learning stage determines the optimal weights $\boldsymbol{w}^{\star}$.

Once trained, the neural network $g_{\boldsymbol{w}}$ represents statistically the inverse of the radiative transfer equation. The NN model is then valid for all observations (i.e., global inversion), where iterative methods, such as variational assimilation, have to compute an estimator for each observation (i.e., local inversion). Table 1 gives the RMS scores for the first guesses and the retrievals. For each output, the retrieval is a considerable improvement compared to the first guess.

2.6 Neural Network Hessian Regularization. The Hessian $\boldsymbol{H}$ is computed using the data set $\mathcal{B}$. A few comments about the inversion of $\boldsymbol{H}$ are required. This matrix can be very large when the NN considered is big (W, the size of $\boldsymbol{H}$, and the number of parameters in the NN can reach a few thousand). This means that the inversion can be sensitive to numerical problems. As a consequence, the estimation of $\boldsymbol{H}$ needs to be done with enough samples from $\mathcal{B}$; otherwise, the subspace spanned by the samples describing $\boldsymbol{H}$ might be too small or the eigenvalues of $\boldsymbol{H}$ too close to zero or even negative, making the inversion numerically impossible.

We noted in section 2.3 that the gradient $\boldsymbol{b}$ in equation 2.8 is supposed to be zero; otherwise, the local quadratic approximation is not good enough, implying that the Hessian matrix $\boldsymbol{H}$ is not positive definite. As a consequence, it is very important that the learning of the NN converges close enough toward the optimal solution $\boldsymbol{w}^{\star}$. Monitoring the convergence al-

Table 1: First Guess and Retrieval RMS Errors.

	First Guess	Retrieval
Ts (K)	3.52411	1.46250
WV $(Kg.m^{-2})$	7.99485	3.83521
$E_m 19V$	0.01504	0.00494
$E_m 19H$	0.01837	0.00495
$E_m 22V$	0.01659	0.00562
$E_m 37V$	0.01425	0.00497
$E_m 37H$	0.01802	0.00501
$E_m 85V$	0.01764	0.00682
$E_m 85H$	0.02137	0.00820

gorithm could be enhanced by checking in parallel the positive definite character of the corresponding Hessian of the network.

Even when enough samples from $\mathcal{B}$ are used to estimate $\boldsymbol{H}$ and when the learning convergence is reached, numerical problems can still exist. This situation can be related to an inconsistency between the complexity of the NN versus the complexity of the desired function to be estimated: too many degrees of freedom in the NN can produce an ill-conditioned Hessian matrix $\boldsymbol{H}$. A possible solution often used in this context is to introduce a diagonal regularization matrix: $\boldsymbol{H}$ is replaced by $\boldsymbol{H} + \lambda \boldsymbol{I}$, where λ is a small scalar and $\boldsymbol{I}$ is the identity matrix. The regularization factor λ is chosen to be small enough not to change the structure in $\boldsymbol{H}$ but big enough to allow the inversion: a compromise must be found.

To determine the factor λ representing the right trade-off, we use four regularization criteria together with a discrepancy measure between the nonregularized $\boldsymbol{H}$ and the regularized matrix $\boldsymbol{H} + \lambda \boldsymbol{I}$. The regularization criteria are the condition number with respect to inversion (the lower the better); the P-number, which is a positive integer if the matrix is not positive definite and zero otherwise (the lower the better); and the number of negative diagonal terms in the matrix (the lower the better). For the discrepancy measure between $\boldsymbol{H}$ and $\boldsymbol{H} + \lambda \boldsymbol{I}$, we use the RMS differences between the square roots of the positive diagonal elements of the matrices (the lower the better). This quantity measures the differences that the regularization has introduced in the standard deviations of the two covariance matrices.

In Figure 1, the variations of these four quantities for an increasing λ, from 0 to 50 are shown. A good compromise is found to be $\lambda = 12.0$: the regularization criteria are satisfactory (positive definite matrix, all diagonal terms positive, minimum condition number), and the discrepancy measure is still small.

2.7 PDF of Network Weights.

To complete the analysis of the uncertainties due to the inversion algorithm, the posterior distribution of the network weights needs to be determined. As previously stated, this PDF represents a plausibility of weights $\boldsymbol{w}$, not a probability of finding the particular weights. We saw in section 2.3 that this distribution follows a gaussian PDF with mean $\boldsymbol{w}^\star$ and covariance matrix $\boldsymbol{H}^{-1}$.

In Figure 2, the optimum weights $\boldsymbol{w}^\star$ are shown together with ± two standard deviations. As previously noted, weights between the hidden and the output layers are more variable than weights between the input and the hidden layers. This is due to the fact that the first processing stage of the NN, at the hidden layer level, is a high-level processing that includes the nonlinearity of the network. The second processing stage of the NN, from the hidden layer to the output layer, is just a linear postprocessing of the hidden layer.

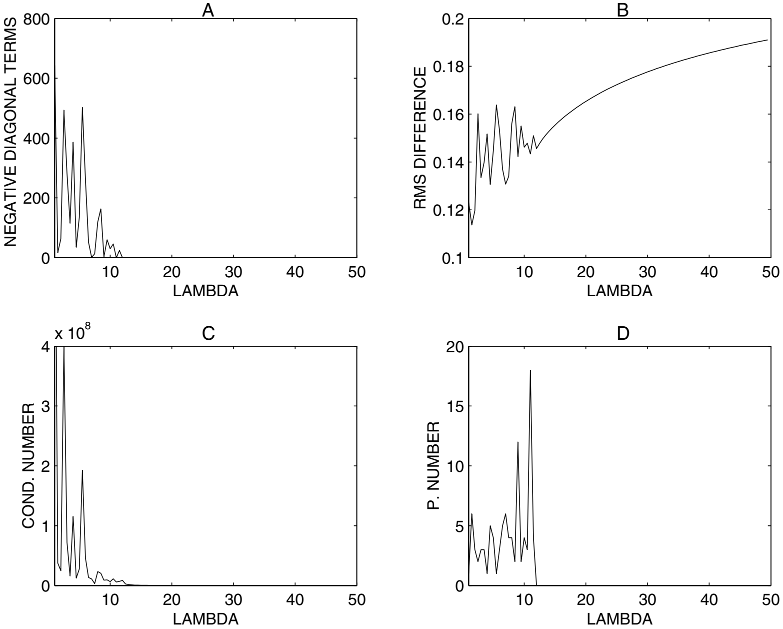

Figure 1: Quality criteria for variable λ. (A) The number of negative diagonal terms in the matrix. (B) RMS differences between the square root of the positive diagonal elements of the matrices. (C) The condition number with respect to inversion. (D) The P-number, which is a positive integer if the matrix is not positive definite and zero otherwise. See the text.

It is possible to know much more than just the output estimates of an NN. From the distribution of weights, samples $\{\boldsymbol{w}^r \; ; \quad r = 1, \ldots, R\}$ of NN weights can be chosen. Each of the R samples $\boldsymbol{w}^r$ represents a particular NN. Together, they represent the uncertainty on the NN weights. These samples can be used later to integrate under the PDF of weights in a Monte Carlo approach. For neural networks, the number of parameters (i.e., size of $\boldsymbol{w}$) is big, so it is preferable to use an advanced sampling technique. Even if these samples are included within the large variability of the two standard deviations envelope, correlation constraints avoid random oscillations from noise by imposing some structure on them. The weights have considerable latitude to change, but their correlations constrain them to follow a strong dependency structure. This is why different weight configurations can result in the same outputs. Most important for network processing is the structure of these correlations. For example, if the difference of two inputs is a good predictor, as long as two weights linked to the two inputs perform the

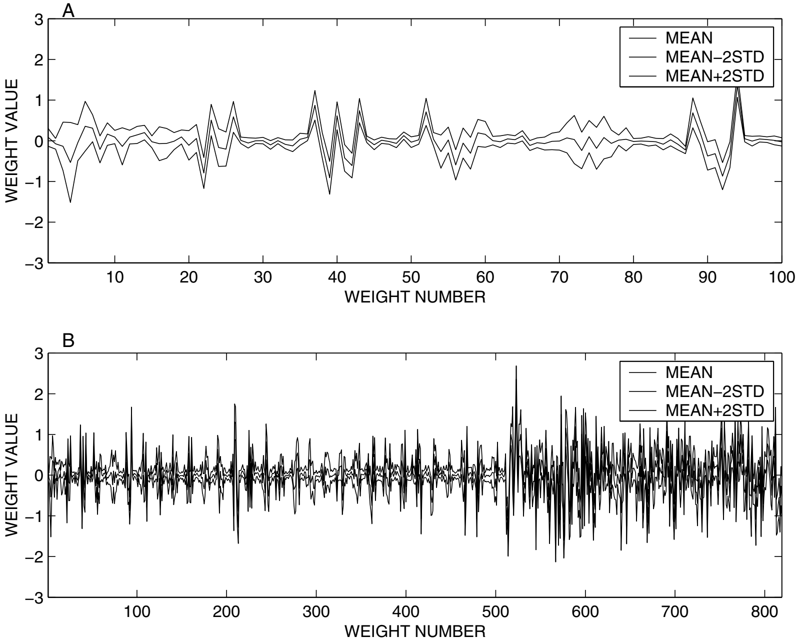

Figure 2: Mean network weights $w^{\star} \pm 2$ standard deviation: (A) The first 100 NN weights corresponding to input/hidden layer connections, and (B) all 821 NN weights with weight 510 to 819 for hidden/output layer connections.

difference, the absolute value of the weights is not essential. Another source of uncertainty for the weights is the fact that some permutations of neurons have no impact on the network output. For example, if two neurons in the hidden layer of the network are permuted, the network answer would not change. The sigmoid function used in the network is saturated when the neuron activity entering is too low or too high. This means that a change of a weight going to this neuron would have a negligible consequence.

These are just a few reasons that explain why the network weights can vary and still provide a good general fitting model. Variability of the network weights is considered a natural variability, inherent to the neural technique. Furthermore, what is important for the NN user is not the variability of the weights but the uncertainty that this variability produces in the network outputs or in even more complex quantities such as the network Jacobians.

3 Uncertainty in Network Outputs

3.1 Theoretical Derivation of the Network Output Error PDF. The distribution of uncertainties of the NN output, $\boldsymbol{y}$, is given by

$$P(\boldsymbol{y}|\boldsymbol{x},\mathcal{D}) = \int P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) \cdot P(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}, \tag{3.1}$$

where $\mathcal{D}$ is the set of outputs $\boldsymbol{y}$ in a data set $\mathcal{B} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{t}^{(n)})\ ;\ \ n = 1, \ldots, N\}$ of N matched input-output couples. Using equations 3.4 and 3.12, we find that this probability is equal to:

$$\frac{1}{Z}\int e^{-\frac{1}{2}(\boldsymbol{t} - g_{\boldsymbol{w}}(\boldsymbol{x}))^T \cdot \boldsymbol{A}_{in} \cdot (\boldsymbol{t} - g_{\boldsymbol{w}}(\boldsymbol{x}))} \cdot e^{-\frac{1}{2}\triangle \boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle \boldsymbol{w}} d\boldsymbol{w}, \tag{3.2}$$

where $\boldsymbol{A}_{in}$ is the inverse of $\boldsymbol{C}_{in}$, the covariance matrix of the "intrinsic noise" of physical variables $\boldsymbol{y}$, and $\boldsymbol{H}$ is the Hessian matrix of the quality criterion used by the learning process. Note that all the terms not dependent on $\boldsymbol{w}$ have been put together in the normalization factor Z. A first-order expansion of the NN function $g_{\boldsymbol{w}}$ about the optimum weight $\boldsymbol{w}^{\star}$ is now used:

$$g_{\boldsymbol{w}}(\boldsymbol{x}) = g_{\boldsymbol{w}^{\star}}(\boldsymbol{x}) + \boldsymbol{G}^T \cdot \triangle \boldsymbol{w}, \tag{3.3}$$

where

$$\boldsymbol{G} = \left.\bigtriangledown\right|_{\{\boldsymbol{w}=\boldsymbol{w}^{\star}\}} (g_{\boldsymbol{w}}) \tag{3.4}$$

is a $W \times M$ matrix. Introducing equation 3.4 into 3.2, and using $\varepsilon_y = (\boldsymbol{y} - g_{\boldsymbol{w}^{\star}}(\boldsymbol{x}))$, we obtain

$$\begin{aligned} P(\boldsymbol{t}|\boldsymbol{x},\mathcal{D}) &\propto e^{-\frac{1}{2}\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y} \int e^{-\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot (\boldsymbol{G}^T \triangle \boldsymbol{w})} \\ & e^{-\frac{1}{2}\triangle \boldsymbol{w}^T \cdot (\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T + \boldsymbol{H}) \cdot \triangle \boldsymbol{w}} d\boldsymbol{w} & (3.5) \\ &\propto e^{-\frac{1}{2}\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y} \int e^{\boldsymbol{h}^T \cdot \triangle \boldsymbol{w} - \frac{1}{2}\triangle \boldsymbol{w}^T \cdot \boldsymbol{O} \cdot \triangle \boldsymbol{w}} d\boldsymbol{w}, & (3.6) \end{aligned}$$

where $\boldsymbol{h} = [-\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T]^T$ and $\boldsymbol{O} = \boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T + \boldsymbol{H}$.

The integral term in equation 3.6 can be simplified by

$$(2\pi)^{\frac{dimW}{2}} |\boldsymbol{O}|^{-\frac{1}{2}} e^{\frac{1}{2}\boldsymbol{h}^T \cdot \boldsymbol{O} \cdot \boldsymbol{h}}. \tag{3.7}$$

We can rewrite equation 3.6 using this simplification to obtain

$$\begin{aligned} P(\boldsymbol{t}|\boldsymbol{x},\mathcal{D}) &\propto e^{-\frac{1}{2}\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y} \\ & e^{\frac{1}{2}\varepsilon_y{}^T \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T (\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T + \boldsymbol{H})^{-1} \cdot \boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \varepsilon_y} & (3.8) \\ &\propto e^{-\frac{1}{2}\varepsilon_y{}^T \cdot [\boldsymbol{A}_{in} - \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T (\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T + \boldsymbol{H})^{-1} \boldsymbol{G} \cdot \boldsymbol{A}_{in}] \cdot \varepsilon_y}. & (3.9) \end{aligned}$$

This means that the distribution of $\boldsymbol{t}$ follows a gaussian distribution with mean $g_{\boldsymbol{w}^\star}(\boldsymbol{x})$ and covariance matrix:

$$\boldsymbol{C}_0 = \left[\boldsymbol{A}_{in} - \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T (\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T + \boldsymbol{H})^{-1} \boldsymbol{G} \cdot \boldsymbol{A}_{in}\right]^{-1}. \tag{3.10}$$

This covariance matrix can be simplified by multiplying the numerator and denominator by

$$\boldsymbol{G} \cdot (\boldsymbol{I} + \boldsymbol{H}^{-1} \cdot \boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^T) \cdot \boldsymbol{G}$$

to obtain

$$\boldsymbol{C}_0 = \boldsymbol{C}_{in} + \boldsymbol{G}^T \cdot \boldsymbol{H}^{-1} \cdot \boldsymbol{G}. \tag{3.11}$$

We see that the uncertainty in the network outputs is due to the intrinsic noise of the target data embodied in $\boldsymbol{C}_{in}$ and the uncertainty described by the posterior distribution of the weight vector $\boldsymbol{w}$ embodied in $\boldsymbol{G}^T \cdot \boldsymbol{H}^{-1} \cdot \boldsymbol{G}$. This relation describes the fact that the uncertainties are approximately related to the inverse data density. As expected, uncertainties are larger in the less dense data space, where the learning algorithm gets less information.

3.2 Sources of Uncertainty. In equation 3.11, we have separated the sources of error in two terms: the intrinsic noise with covariance matrix $\boldsymbol{C}_{in}$ and the neural inversion term with covariance matrix $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$. Our neural inversion term refers to the errors due only to the uncertainty in the inverse model parameters, and all the remaining outside sources of errors are grouped in $\boldsymbol{C}_{in}$. The inversion uncertainty can itself be decomposed into three sources, corresponding to the three main components of an NN model:

1. The imperfections of the learning data set $\mathcal{B}$, which include simulation errors when $\mathcal{B}$ is simulated by a model, collocation and instrument errors when $\mathcal{B}$ is a collection of coincident inputs and outputs, null-space errors, and others. This is probably the most important source of uncertainty due to the inversion technique.
2. Limitations of the network architecture because the model might not be optimum, with too few degrees of freedom or a structure that is not optimal. This is usually a lower-level source of uncertainty because the network can partly compensate for these deficiencies.
3. A nonoptimum learning algorithm because as good as the optimization technique is, it is impossible in practice to be sure that the global minimum $\boldsymbol{w}^\star$ has been found instead of a local one. We think that this source of uncertainty is limited.

Matrix $\boldsymbol{C}_{in}$ includes all other sources of errors. Our approach allows for the estimation of the global $\boldsymbol{C}_{in}$, but if some individual terms are known, it is possible to subtract them from $\boldsymbol{C}_{in}$. For example, if the instrument noise is known, it is possible to measure the impact of this noise on the NN outputs. The individual terms can then be subtracted from the global $\boldsymbol{C}_{in}$. For simplification and because we do not use such a priori information, we adopt the hypothesis that $\boldsymbol{C}_{in}$ is constant for each situation; only the inversion term is situation dependent. But any a priori information about any nonconstant term in $\boldsymbol{C}_{in}$ could be used in this very flexible approach.

Note that the specification of the sources of uncertainty by the approach of Rodgers (1990) uses mainly the concept of Jacobians of either the direct or the inverse model in order to linearize the impact of each error source. Linearity and gaussian variables are easily manageable analytically, the algebra being essentially based on the covariance matrices—for example:

- $\boldsymbol{C}_M = \boldsymbol{D}_x \cdot \boldsymbol{E} \cdot \boldsymbol{D}_x{}^T$, the covariance of the errors due to instrument noise, where $\boldsymbol{D}_x = \frac{\partial g_w}{\partial x}$ is the contribution function and $\boldsymbol{E} = \langle \boldsymbol{\eta}^T \cdot \boldsymbol{\eta} \rangle$ is the covariance matrix of instrument noise $\boldsymbol{\eta}$; or
- or $\boldsymbol{F} = \boldsymbol{A}_b \cdot \boldsymbol{C}_b \cdot \boldsymbol{A}_b{}^T$, the covariance of the forward model errors, where $\boldsymbol{C}_b$ is the covariance matrix errors of the forward model parameter, $\boldsymbol{b}$, and $\boldsymbol{A}_b$ is the sensitivity matrix of observations $\boldsymbol{b}$ with respect to $\boldsymbol{b}$ (Rodgers, 1990).

Some bridges can be built to link our error analysis and the approach used in variational assimilation by Rodgers (1990). In section 4, such Jacobians are analytically derived in the neural network framework. This makes feasible the use of Rodgers' estimates. The difference would be that our linearization uses Jacobians that are situation dependent; this means that the estimation of the error sources would be nonlinear in nature. This will be the subject of another study.

In Wright et al. (2000), noise in NN inputs is considered an additional source of uncertainty. An approach for the empirical characterization of the various sources of uncertainties is to use simulations. For example, for the instrument noise-related uncertainty, it is easy to introduce a sample of noise into the network inputs and analyze the consequent error distribution of the outputs. The advantage of such simulation approach is that it is very flexible and allows for the manipulation of nongaussian distributions. This will be the subject of another study.

3.3 Distribution of Network Outputs.

After the learning stage, we estimate $\boldsymbol{C}_0$, the covariance matrix of network errors $\varepsilon_y = (\boldsymbol{t} - g_{\boldsymbol{w}}(\boldsymbol{x}))$, over the database $\mathcal{B}$. equation 3.11 shows that this covariance adds the errors due to neural network uncertainties and all other sources of uncertainty. Table 2 gives the numerical values of $\boldsymbol{C}_0$ for the particular example from Prigent et al. (2003). The right/top triangle is for the correlation, and the left/bottom triangle is for the covariance. The diagonal values give the variance of errors of quantity. The correlation part indicates clearly that some errors are highly

Table 2: Covariance Matrix C_0 of Network Output Error Estimated over the Database $\mathcal{B}$.

	Ts	WV	E_m19V	E_m19H	E_m22V	E_m37V	E_m37H	E_m85V	E_m85H
Ts	2.138910	−0.24	**−0.87**	**−0.72**	**−0.76**	**−0.84**	**−0.72**	**−0.49**	**−0.32**
WV	−1.392113	14.708836	0.16	−0.06	0.14	0.05	−0.15	−0.18	**−0.37**
E_m19V	−0.006294	0.003179	0.000024	**0.77**	**0.88**	**0.89**	**0.74**	**0.60**	**0.42**
E_m19H	−0.005261	−0.001143	0.000019	0.000024	**0.72**	**0.73**	**0.81**	**0.60**	**0.56**
E_m22V	−0.006274	0.003140	0.000024	0.000020	0.000031	**0.84**	**0.71**	**0.71**	**0.54**
E_m37V	−0.006121	0.001049	0.000021	0.000018	0.000023	0.000024	**0.81**	**0.70**	**0.50**
E_m37H	−0.005290	−0.002954	0.000018	0.000020	0.000020	0.000020	0.000025	**0.65**	**0.67**
E_m85V	−0.004895	−0.004945	0.000020	0.000020	0.000027	0.000023	0.000022	0.000046	**0.79**
E_m85H	−0.003906	−0.011933	0.000017	0.000022	0.000024	0.000020	0.000027	0.000044	0.000067

Notes: The right/top triangle is for correlation, and the left/bottom triangle is for covariance; the diagonal gives the variance. Correlations with absolute value higher than 0.3 are in bold.

correlated. This is why it would be a mistake to monitor only the error bars, even if they are easier to understand.

The correlations of errors exhibit the expected physical behavior. Errors in Ts are negatively correlated with the other errors, with large values of correlation with the vertical polarization emissivities, for the channels that are much less sensitive to the water vapor (E_m19V and E_m37V). The vertical polarization emissivities are larger than for the horizontal polarizations and are often close to one, with the consequence that the radiative transfer equation in channels that are much less sensitive to the water vapor (the 19 and 37 GHz channels) is quasi-linear in Ts and in E_mV. In contrast, errors in water vapor are weakly correlated with the other errors: the largest correlation is with the emissivity at 85 GHz in the horizontal polarization. The 85 GHz channel is the most sensitive to water vapor, and, since the emissivity for the horizontal polarization is lower than for the vertical, the horizontal polarization channel is more sensitive to water vapor. Correlations between the water vapor and the emissivities errors are positive or negative, depending on the respective contribution of the emitted and reflected energy at the surface (which is related not only to the surface emissivity but also to the atmospheric contribution at each frequency). Correlations between emissivity errors are always of the same signs and are high for the same polarizations, decreasing when the difference in frequency increases.

The correlations involved in the PDF of the errors described by the covariance matrix $\boldsymbol{C}_0$ make it necessary to understand the uncertainty in a multidimensional space. This is more challenging than just determining the individual error bars, but it is also much more informative: the diagonal elements of the covariance matrix provide the variance for each output error, but the off-diagonal terms show the level of dependence among these output errors. To statistically analyze the covariance matrix $\boldsymbol{C}_0$, this matrix can be decomposed into its orthogonal eigenvectors (not shown). These base functions constitute a set of error patterns (Rodgers, 1990).

3.4 Covariance of Output Errors Due to the Neural Inversion.

The matrix $\boldsymbol{H}^{-1}$ is the covariance of the PDF of network weights. The use of the gradient $\boldsymbol{G}$ transforms this matrix into $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$, the covariance error of the NN outputs associated with the uncertainty of weights. Note that multiplication by $\boldsymbol{G}$ partially regularizes $\boldsymbol{H}^{-1}$, so that for this particular purpose of the estimation of the output errors, $\boldsymbol{H}$ does not need to be regularized.

Table 3 represents this covariance matrix $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$ averaged over the whole learning database $\mathcal{B}$. Even if some of the bottom-left values representing the covariance matrix are close to zero (this is an artifact since the variability ranges of the variables are quite different from each other), structure is still present in this matrix, as is shown in the correlation part (top right). The error correlation matrix $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$, related to the NN inversion method, has relatively small magnitudes with a maximum of 0.55. However,

Table 3: Covariance matrix $G^T H^{-1} G$ of Error Due to Network Uncertainty, Averaged over the Database $\mathcal{B}$.

	Ts	WV	$E_m 19V$	$E_m 19H$	$E_m 22V$	$E_m 37V$	$E_m 37H$	$E_m 85V$	$E_m 85H$
Ts	0.493615	−0.14	−0.28	−0.14	−0.25	−**0.32**	−0.16	−0.19	−0.06
WV	−0.106484	1.063071	0.10	−0.02	0.09	0.02	−0.07	−0.15	−0.25
$E_m 19V$	−0.000325	0.000167	0.000002	**0.33**	**0.55**	**0.55**	0.28	0.27	0.08
$E_m 19H$	−0.000255	−0.000060	0.000001	0.000006	0.26	0.22	0.29	0.10	0.13
$E_m 22V$	−0.000268	0.000152	0.000001	0.000001	0.000002	**0.50**	0.26	0.28	0.12
$E_m 37V$	−0.000330	0.000033	0.000001	0.000000	0.000001	0.000002	**0.34**	**0.38**	0.14
$E_m 37H$	−0.000270	−0.000183	0.000001	0.000001	0.000000	0.000001	0.000005	0.16	0.26
$E_m 85V$	−0.000231	−0.000282	0.000000	0.000000	0.000000	0.000000	0.000000	0.000002	**0.43**
$E_m 85H$	−0.000128	−0.000681	0.000000	0.000000	0.000000	0.000000	0.000001	0.000001	0.000006

Notes: The right/top triangle is for correlation, and the left/bottom triangle is for covariance; the diagonal gives the variance. Correlations with absolute value higher than 0.3 are in bold.

it has a structure similar to the global correlation matrix, with the same signs of correlation and similar relative values between the variables.

3.5 Covariance of the Intrinsic Noise of Target Values. To estimate $\boldsymbol{C}_{in}$, we use equation 3.11,

$$\boldsymbol{C}_{in} = \langle \boldsymbol{C}_0 \rangle_{\mathcal{B}} - \langle \boldsymbol{G}^T \boldsymbol{H}^{-1} \boldsymbol{G} \rangle_{\mathcal{B}}, \tag{3.12}$$

where the two right-hand terms are the covariance matrix of the total output errors averaged over $\mathcal{B}$ (see section 3.3) and the covariance matrix of the output errors due to the network inversion scheme averaged over $\mathcal{B}$ (see section 3.4). Table 4 gives the numerical values of the matrix $\boldsymbol{C}_{in}$: The right/top triangle is for the correlation and the left/bottom triangle is for the covariance. Intrinsic error correlations can be very large (up to 0.99). The structure of $\boldsymbol{C}_{in}$ is also very similar to the structure of the global error correlation matrix, the only noticeable difference being the larger correlation values. An important remark is that the covariance error is dominated, in this application, by the intrinsic errors. This behavior is totally dependent on the particular application that is treated.

3.6 Network Outputs Error Estimate. Once $\boldsymbol{C}_{in}$ is available, we can estimate a $\boldsymbol{C}_0(\boldsymbol{x})$ that is dependent on the observations $\boldsymbol{x}$, the term $\boldsymbol{G}^T \boldsymbol{H}^{-1} \boldsymbol{G}$ varying with input $\boldsymbol{x}$. It should be noted that the use of the regularization for matrix $\boldsymbol{H}$ has virtually no consequences for the results obtained for the error bars in the following. Using no regularization for the Hessian matrix is possible since $\boldsymbol{H}$ is multiplied by the gradients in $\boldsymbol{G}^T \boldsymbol{H}^{-1} \boldsymbol{G}$. This is an additional argument that the regularization helps the matrix inversion without damaging the information in the Hessian.

$\boldsymbol{C}_0(\boldsymbol{x})$ is estimated for each of the 1,239,187 samples for clear-sky pixels in July 1992. Figure 3 presents the monthly mean standard deviations (square root of the diagonal terms in $\boldsymbol{C}_0(\boldsymbol{x})$) for four outputs: the surface skin temperature *Ts*, the columnar integrated water vapor *WV*, and the microwave emissivities at 19 GHz for vertical and horizontal polarizations.

The errors exhibit the expected geographical patterns. Large errors on *Ts* are concentrated in regions where the emissivities are lower or highly variable: inundated areas and deserts. In inundated areas, for instance (around the rivers like the Amazon or the Mississippi), or in coastal regions, the contribution from the surface is weaker, and sensitivity to *Ts* is lower because the emissivities are lower. In sandy regions through desert areas, due to higher transmission in the very dry sandy medium, microwave radiation does not come from the very first millimeters of the surface, but from deeper below the surface—the lower the frequency, the deeper (Prigent & Rossow, 1999). As a consequence, the microwave radiation is not directly related to the skin surface temperature (see Prigent & Rossow, 1999, for a detailed explanation) and *Ts* cannot be retrieved with the same accuracy.

Table 4: Covariance Matrix C_{in} of Intrinsic Noise Errors, Estimated over the Database $\mathcal{B}$.

	Ts	WV	E_m19V	E_m19H	E_m22V	E_m37V	E_m37H	E_m85V	E_m85H
Ts	1.645294	−0.27	**−0.99**	**−0.92**	**−0.86**	**−0.95**	**−0.88**	**−0.55**	**-0.37**
WV	−1.285629	13.645765	0.17	−0.06	0.14	0.05	−0.16	−0.19	**−0.39**
E_m19V	−0.005968	0.003011	0.000021	**0.89**	**0.91**	**0.92**	**0.83**	**0.63**	**0.46**
E_m19H	−0.005006	−0.001083	0.000017	0.000017	**0.83**	**0.86**	**0.98**	**0.71**	**0.66**
E_m22V	−0.006005	0.002988	0.000023	0.000019	0.000029	**0.87**	**0.80**	**0.75**	**0.58**
E_m37V	−0.005790	0.001015	0.000020	0.000017	0.000022	0.000022	**0.90**	**0.72**	**0.54**
E_m37H	−0.005019	−0.002770	0.000017	0.000018	0.000019	0.000019	0.000019	**0.74**	**0.76**
E_m85V	−0.004663	−0.004662	0.000019	0.000019	0.000026	0.000022	0.000021	0.000043	**0.82**
E_m85H	−0.003777	−0.011251	0.000016	0.000021	0.000024	0.000019	0.000026	0.000042	0.000060

Notes: The right/top triangle is for correlation, and the left/bottom triangle is for covariance; the diagonal gives the variance. Correlations with absolute value higher than 0.3 are in bold.

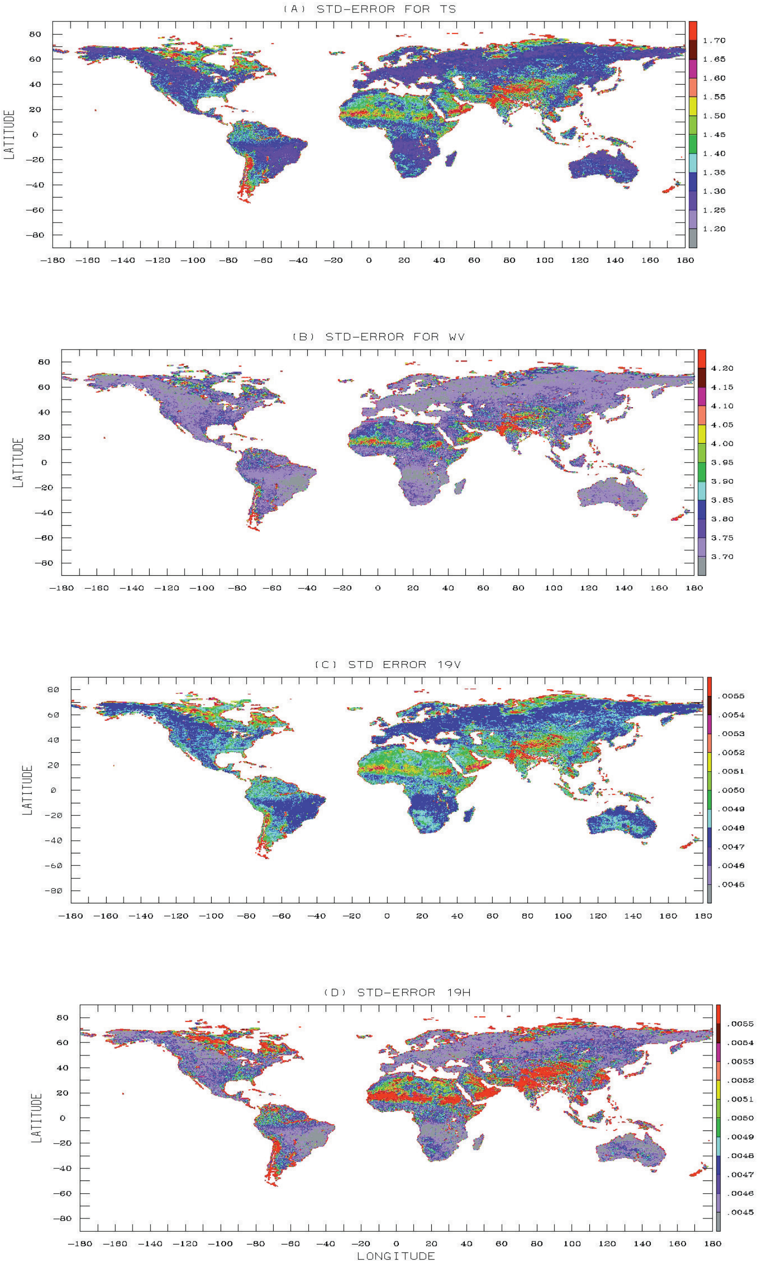

Figure 3: Standard deviation of error maps for (A) surface skin temperature *Ts*, (B) columnar integrated water vapor *WV*, (C) microwave emissivity at 19 GHz vertical polarization, and (D) microwave emissivity at 19 GHz horizontal polarization.

The same arguments hold for the errors in emissivity. All the parameters being tightly related for a given pixel, the water vapor errors are also rather large in inundated regions and in sandy areas.

3.7 Outlier Detection. What is the behavior of the neural retrieval when the situation is particularly difficult, as when the first-guess is far from the actual solution? In principle, the nonlinearity of the NN allows it to have different weights on the observations and first-guess information, depending on the situation. For example, if the first guesses are better in tropical cases than in polar cases, the NN will have inferred this behavior during the learning stage and then will give less emphasis to the first guess when a polar situation is to be inverted. This assumes once again that the training data set is correctly sampled. To understand the behavior of the uncertainty estimates better, a good strategy is to introduce artificial errors for each source of information and to analyze the resulting impact on the network outputs.

In Figure 4, the retrieval STD error change index is presented to show the effect of perturbating the mean inputs or the mean FGs by an artificial error. The impact of these artificial errors is measured in terms of the percentage of the regular STD retrieval error. For example, an impact index of 120% means that the regular STD retrieval error estimate increases by 20% when the input is perturbed. The impact indices can be compared for each of the nine network outputs. These results are obtained by averaging over the 20,000 samples in $\mathcal{B}$.

Figure 4A presents the error impacts when all 17 network inputs are changed by a factor ranging from −5% to +5%. Obviously, this will correspond to incoherent situations since the complex nonlinear relationships between vertical and horizontal brightness temperatures and first guesses will not be respected. As expected, the error increases monotically with the absolute value of the perturbation. However, the impact is not uniform among the output variables. For WV, which is retrieved with a rather low accuracy, changes in the inputs do not have a large influence. The impact on the emissivities is larger for horizontal polarizations than for vertical: horizontal polarization emissivities are much more variable than the vertical ones, and as a consequence, emissivities for vertical polarization have rather similar values in outputs whatever the situation and do not depend that much on the inputs. It can also be noted that positive perturbations have a slightly stronger impact than negative ones. This is to be related to the distribution of the variables in the training database. For the emissivities, for instance, the distribution has a steep cut-off for unit emissivity, above which the emissivities are not physical. On the contrary, a large range of emissivities exists in the training data base at lower values (see Figure 3 in Aires et al., 2001). As a consequence, decreasing the emissivity first guess will still be physically realistic, whereas increasing it will not be.

Figure 4B is the same except that the changes are made only for the first-guess inputs. We note a similar behavior (nonuniform impact among output

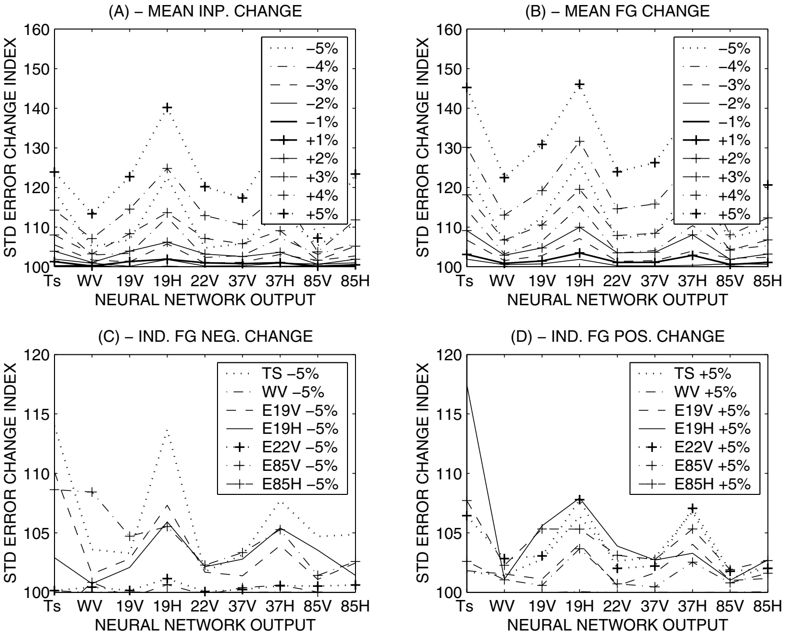

Figure 4: Estimated STD error change index for an artificial pertubation: (A) of the mean input, (B) of the mean first-guess input, (C) of individual first-guess negative changes, and (D) of individual first-guess positive changes. See the detailed explanation in the text. Statistics are performed over 20,000 samples from $\mathcal{B}$.

variables and with larger impact for positive perturbations), but we observe also that errors are larger than when all the inputs are perturbed in Figure 4A. This suggests that the error estimate is able to detect inconsistencies between observations and first-guess inputs.

In Figures 4C and 4D, the first-guess input variables are perturbed individually with, respectively, negative and positive amplitude of 5%. For negative perturbations, the biggest impact is produced by the Ts first-guess perturbation: it is noticeable that the Ts error impact is similar for the retrieval of $E_m 19H$ and for its own retrieval. For other variables, the impacts have lower levels, with almost no impact from the WV first guess. The WV first guess is associated with a large error (40%), and as a consequence the NN gives little importance to this first guess. For positive individual perturbations in Figure 4D, the results are similar to the negative errors. The magnitude of the positive changes as compared to the negative ones is related again to the distribution of the variables in the training data set (see Figure 3 in Aires et al., 2001): If the distribution is not symmetrical around

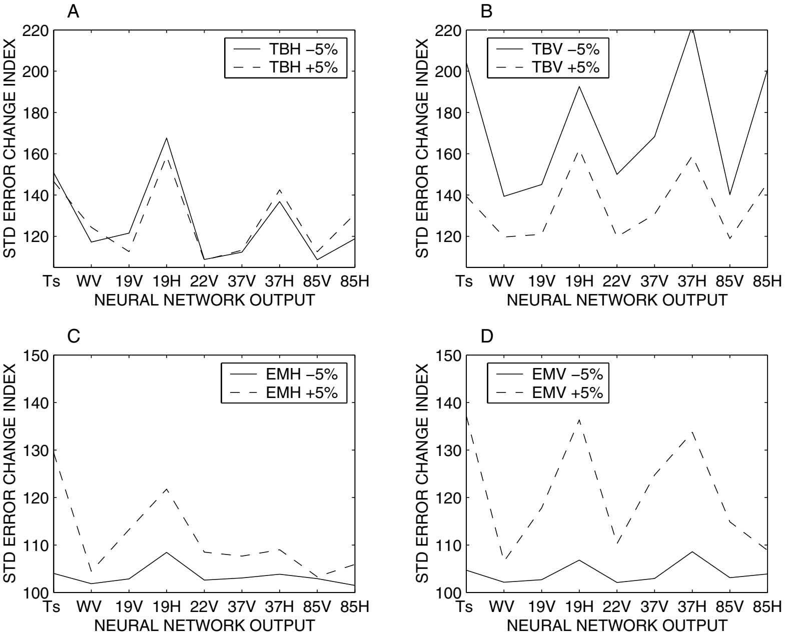

Figure 5: Estimated STD error change index for an artificial pertubation: (A) of horizontal polarization brightness temperatures, (B) of vertical polarization brightness temperatures, (C) of horizontal polarization first-guess emissivities, and (D) of vertical polarization first-guess emissivities. Statistics are performed over 20,000 samples from $\mathcal{B}$.

a mode value, depending on the shape of the distribution, increasing or decreasing the value can be more or less realistic.

In Figure 5, “incoherencies” have been introduced between the vertical and horizontal polarizations in the brightness temperatures (TB) observations and in the first-guess emissivities, E_ms, by increasing or decreasing one, keeping the other polarization constant. In Figure 5A, we increased and decreased artificially by 5% the horizontal TB, and in Figure 5B, the same has been done for vertical polarizations. Figures 5C and 5D are similar for first-guess emissivities instead of TB. Several comments can be made. First, the impact is larger for observations than for first-guess errors, which suggests that observations are more important for the retrieval, the first guess being used mostly as an additional constraint. Second, these polarization inconsistencies have a bigger impact than changes of the means in Figure 4. For example, the NN might emphasize the difference of polarization for the retrieval, and then these inconsistencies would have a very strong impact. This shows that the NN, using complex nonlinear multivariate relationships, is sensitive to inconsistencies among the inputs. It is encouraging to see that our error estimates are able to detect such situations. Finally, the relative impact of the positive and negative changes can be explained again by the distribution of the variables in the learning database. For the emissivities, whatever the polarization and the frequency, the histograms are not symmetrical, having a broad tail toward lower values and an abrupt end for the higher values. As a consequence, when artificially increasing the emissivities, unrealistic values are attained, which is not the case when decreasing the emissivities. (See Aires et al., 2001, for a complete description of the distributions of the learning database and the histograms of the inputs.)

The results shown in Figures 4 and 5 are consistent with a coherent physical behavior, confirming that the new tools developed in this study and its companion articles can be used to diagnose difficult retrieval situations such as might be caused by bad first guesses, inconsistent measurements, situations not included in the training data set, or uncertainties of the NN on the possible retrievals. Our a posteriori probability distributions for the NN retrieval define confidence intervals on the retrieved quantities that allow the detection of such situations.

It could be argued that a limitation of our retrieval uncertainty estimates comes from the fact that our technique is based on statistics over a data set $\mathcal{B}$. This could mean that the error estimate is valid only when we are inside the variability spanned by $\mathcal{B}$. On the contrary, it has been shown that the local quadratic approximation approach increases error estimates in sparsely sampled data space domains (see, e.g., MacKay, 1992).

4 Network Jacobian Uncertainties

The a posteriori distribution of weights is useful to estimate the uncertainties of network outputs (see section 3). We will now show that these distributions can also be used for the estimation of complex probabilistic quantities via Monte Carlo simulations. As an example of such an approach, we use it to estimate the uncertainties of the NN Jacobians.

4.1 Definition of Neural Network Sensitivities. The NN 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 (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

$$\frac{\partial y_k}{\partial x_i} = \sum_{j \in S_1} w_{jk} \cdot \frac{d\sigma}{da} \left(\sum_{i \in S_0} w_{ij} x_i \right) \cdot w_{ij}. \tag{4.1}$$

For a more complex MLP network with more hidden layers, a backpropagation 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, $\boldsymbol{x}$.

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 is a useful data analysis tool (Aires & Rossow, 2003). The Jacobian matrix with terms given by equation 4.1 describes the global 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 Sampling Strategy for Network Weights. 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, 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 gaussian (see section 2). This means that the normalization term $\frac{1}{Z^N}$ in equation 2.9 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 with $\{\boldsymbol{w}^r\ ;\ \ r = 1, \ldots, R\}$ and then use this sample to approximate the integration under the whole weight PDF.

Using only $\boldsymbol{w}^\star$, the MAP parameters, to estimate some other dependent quantities (such as NN Jacobians) 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., the 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.

Concerning the network Jacobians, the MAP network Jacobian is given by using the most probable network 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 NN 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.

To estimate the uncertainties of the Jacobians, we use R =1000 samples from the weights PDF described in section 2. 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. This method is also suitable for nongaussian PDFs.

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.

4.3 Multicollinearity Problem.

Table 5 gives the mean neural Jacobian values for the variables x_k and y_i for the neural network, as defined in equation 4.1. These values indicate the relative contribution of each input in the retrieval of a given output. 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 multicollinearities are a problem for the regression.

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 multicollinearities 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 overallocation by, respectively, an over- or underallocation in the second, correlated input variable. This means that the two corresponding sensitivities will be highly variable from one learning 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.

To see if the multicollinearities 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

Table 5: Global Mean Nonregularized Neural Sensitivities $\frac{\partial y}{\partial x}$.

	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

Notes: Columns are network outputs, y, and rows are network inputs, x. Sensitivities with absolute value higher than 0.3 are in bold, and positive 5% significance test are indicated by an asterisk. The first part of this table is for SSM/I observations; the second part corresponds to first guesses.

of a set of sensitivities is shown in Table 6; 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 multicollinearities and subsequent compensations are acting in the network model.

To avoid such multicollinearity problems, the network learning needs to be regularized by using some physical a priori information to better constrain the learning, in particular in terms of dependency structure among the variables, 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 PCA.

4.4 Principal Component Analysis of Inputs and Outputs. 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. We use the eigendecomposition 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 eigenvectors.

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

$$\overline{\boldsymbol{x}} = \overline{\boldsymbol{F}_x} \cdot {\boldsymbol{S}_{1x}}^{-1} \cdot (\boldsymbol{x} - \boldsymbol{m}_{1x}) \tag{4.2}$$

$$\overline{\boldsymbol{y}} = \overline{\boldsymbol{F}_y} \cdot {\boldsymbol{S}_{1y}}^{-1} \cdot (\boldsymbol{y} - \boldsymbol{m}_{1y}), \tag{4.3}$$

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 $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$ are a compression of the real data, but the inverse transformations of equations 4.2 and 4.3 go back from the compression to the full representation with, of course, some compression errors. PCA is optimum in the least-squares sense: the square errors between data and its PCA representation are minimized.

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 (a smaller number of PCA components used) and a small compression error (a larger number of PCA components used). The more PCA components that are used for compression, the lower the compression error is. Another advantage of the PCA representation is to suppress part of

Table 6: Correlation Matrix for a Sample of Neural Network Sensitivities.

	$\frac{\partial Ts}{\partial Ts}$	$\frac{\partial Ts}{\partial TB19V}$	$\frac{\partial Ts}{\partial TB19H}$	$\frac{\partial Ts}{\partial TB22V}$	$\frac{\partial Ts}{\partial TB37H}$	$\frac{\partial Ts}{\partial TB85V}$	$\frac{\partial Ts}{\partial TB85H}$	$\frac{\partial Ts}{\partial E_m19V}$	$\frac{\partial Ts}{\partial E_m19H}$	$\frac{\partial Ts}{\partial E_m85H}$
$\frac{\partial Ts}{\partial Ts}$	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	...	...	...
$\frac{\partial Ts}{\partial E_m19V}$	0.14	−0.17	0.25	−0.16	−0.06	0.15	0.04	1.00	...	...
$\frac{\partial Ts}{\partial E_m19H}$	−0.00	0.14	**−0.44**	0.09	0.03	−0.03	0.01	**−0.41**	1.00	...
$\frac{\partial Ts}{\partial E_m85H}$	−0.01	0.18	**−0.41**	0.17	0.06	0.03	−0.12	**−0.31**	0.26	1.00

Note: Correlations with absolute value higher than 0.3 are in bold.

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).

Figure 6 describes the cumulated percentage of explained variance by a cumulated number of PCA components for the input and output data. The first PCA components for the inputs and the outputs of the neural network are represented in Figures 7A and 7B, respectively. The physical consistency of the PCA has been checked (not shown) by projecting the samples of the database $\mathcal{B}$ onto the map of the first two principal components that represent most of the variability. Clusters of points are related to surface characteristics. 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

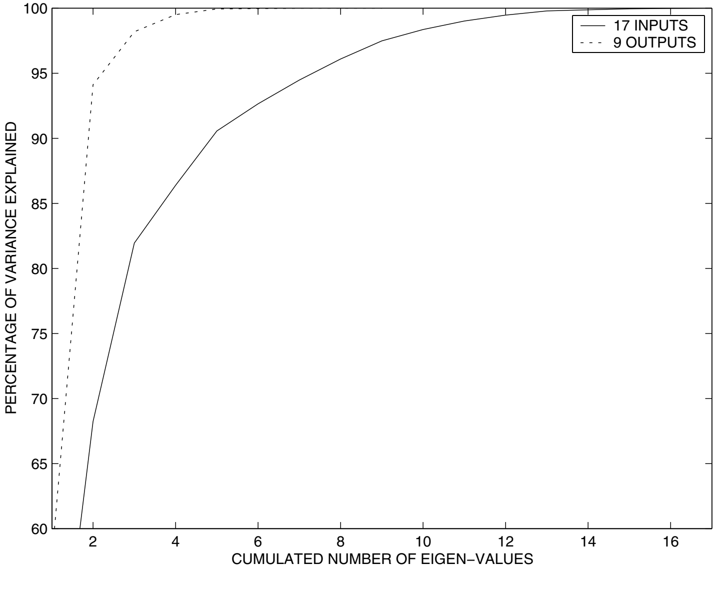

Figure 6: Percentage of variance explained by the first PCA components of inputs (solid) and outputs (dotted) of the neural network.

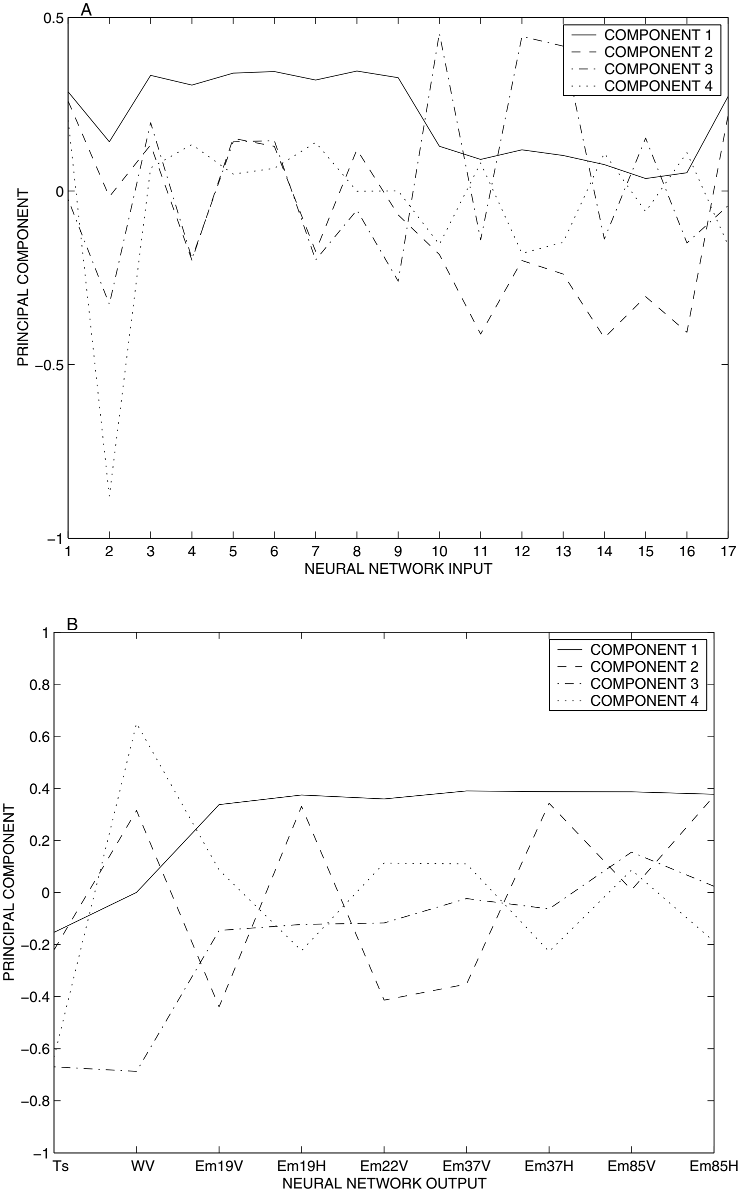

Figure 7: First PCA components of inputs (A) and outputs (B) of the neural network.

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.

4.5 PCA Regression Approach. 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 reduced.

The training of the NN is simpler because the inputs are decorrelated. Correlated inputs in a regression are called multicollinearities, and they are well known to cause problems for the model fit (Gelman et al., 1995). Suppressing these multicollinearities 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.)

How many PCA components should the regression use? From section 4.4, 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 depends entirely 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 denoising optimum would indicate in order to keep all possibly useful information.

In terms of output retrieval quality, the number of PCA components used in the input of the NN needs to be reduced for reasons other 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 input 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 7). No additional information would be gained from adding the higher-order PCA input components.

Table 7: Global Mean Neural Sensitivities $\frac{\partial y'}{\partial x'}$ of Raw Network Output and Input.

	NN Outputs				
NN Inputs	Compo 1	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

Notes: Columns are network outputs, y', and rows are network inputs, x'. Sensitivities with absolute value higher than 0.3 are in bold.

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 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.

The outputs of the network, that is, the PCA components, are not homogeneous; they have different dynamic ranges. The importance of each of the components in the output of the NN 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%. 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, $E_{\mathcal{D}}$, of the quality criterion used for the network learning (see section 2). For an output component, this weight is equal to the standard deviation of the component. This is equivalent to using equation 2.1, 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).

4.6 Retrieval Results of PCA Regression. 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. This is expected because we know that reducing variance (overfitting) by regularization increases the bias (RMS error). This is know as the bias/variance dilemma (Geman, Bienenstock, & Doursat, 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.

In order to estimate the NN weight uncertainties, we use the approach described in section 2: the Hessian matrix $\boldsymbol{H}$ must first be computed and then 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.

Figure 8 presents the corresponding standard deviation for the NN weights with various regularization parameters λ around the optimal value,

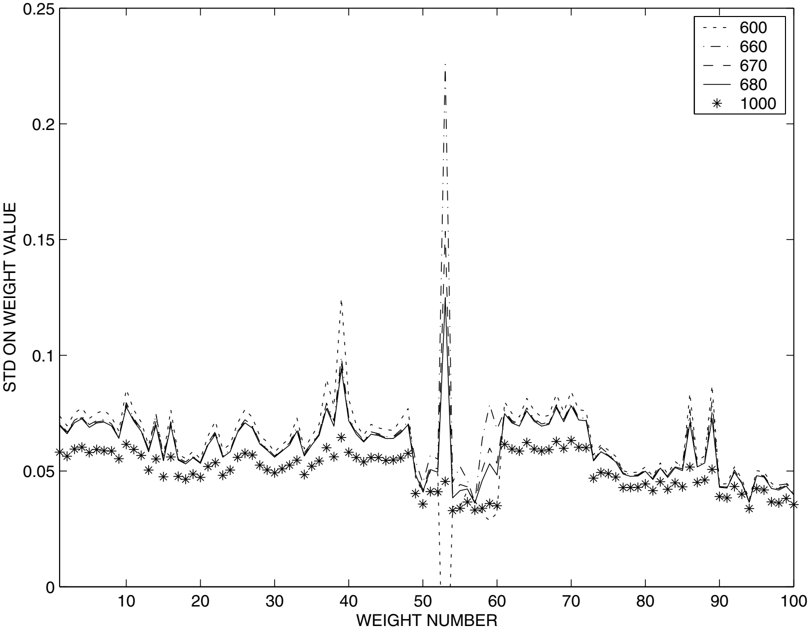

Figure 8: Standard deviation of NN weights with increased regularization parameter λ: λ = 600 (dotted), λ = 660 (dot-dash), λ = 670 (dashed), λ = 680 (solid), and λ = 1000 (asterisk).

λ = 660, which is determined as described in section 2.6 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 $\boldsymbol{H}$ but without changing the standard deviation of well-behaved weights must be found. This is probably the most important issue for the uncertainty estimates described in this study. Another approach to obtain a well-conditioned Hessian would be to constrain the Hessian matrix $\boldsymbol{H}$ to stay definite positive during the learning stage.

4.7 PCA-Regularized Jacobians. 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:

$$\boldsymbol{x}' = {\boldsymbol{S}_{2x}}^{-1} \cdot (\overline{\boldsymbol{x}} - \boldsymbol{m}_{2x}) \tag{4.4}$$

$$\boldsymbol{y}' = {\boldsymbol{S}_{2y}}^{-1} \cdot (\overline{\boldsymbol{y}} - \boldsymbol{m}_{2y}), \tag{4.5}$$

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 4.2 and 4.3) and vectors $\boldsymbol{m}_{2x}$ and $\boldsymbol{m}_{2y}$ are the respective means.

The NN formulation allows derivation of the network Jacobian $\left[\frac{\partial \boldsymbol{y}'}{\partial \boldsymbol{x}'}\right]$ for the normalized quantities of equations 4.4 and 4.5. To obtain the Jacobian in physical units, one should use equations 4.2 through 4.5 to find

$$\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right] = \boldsymbol{S}_{1y} \cdot {\overline{\boldsymbol{F}_y}}^{T} \cdot \boldsymbol{S}_{2y} \cdot \left[\frac{\partial \boldsymbol{y}'}{\partial \boldsymbol{x}'}\right] \cdot {\boldsymbol{S}_{2x}}^{-1} \cdot \overline{\boldsymbol{F}_x} \cdot {\boldsymbol{S}_{1x}}^{-1}. \tag{4.6}$$

Equation 4.6 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.

4.8 Uncertainty of Regularized NN Jacobians. In Table 7, the PCA-regularized NN is used to estimate the mean Jacobian matrix $\left[\frac{\partial \boldsymbol{y}'}{\partial \boldsymbol{x}'}\right]$ of raw network outputs and inputs, together with the corresponding standard deviations. The standard deviations are much more 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 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. But higher-order input components (more than five) 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.

Using equation 4.6, we obtain the corresponding Jacobian matrix $\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right]$ for the physical variables instead of the PCA components, but normalized to be able to compare individual sensitivities (see Table 8). 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 multicollinearities 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 (see Table 5). In addition, the sensitivities make more sense physically, as expected.

The retrieved *Ts* is very sensitive to the brightness temperatures at vertical polarizations for the lower frequencies (see the numbers 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 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 by the emissivity first guess. The sensitivity ma-

Table 8: Global Mean Regularized Neural Sensitivities $\frac{\partial y}{\partial x}$.

	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

Notes: Columns are network outputs, y, and rows are network inputs, x. 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.

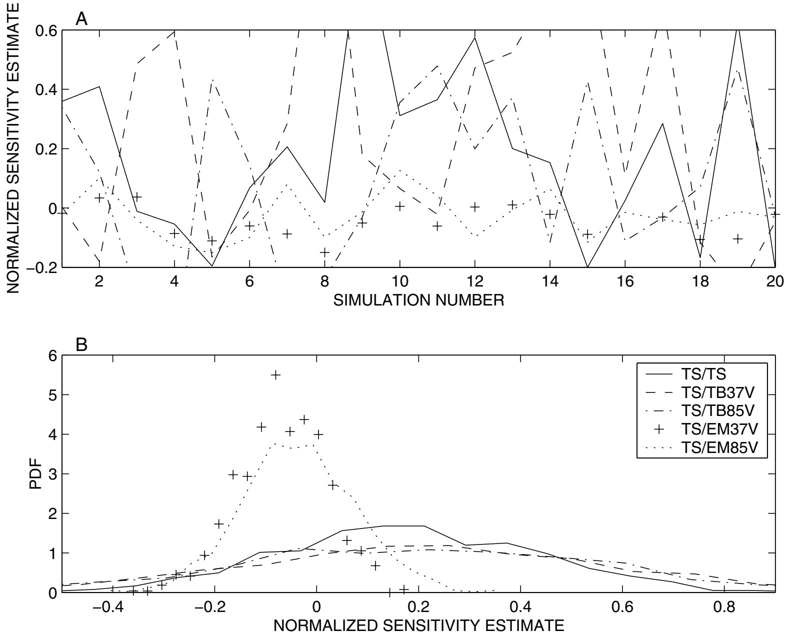

Figure 9: (A) Twenty samples of five neural network sensitivities ($\frac{\partial Ts}{\partial Ts}$, $\frac{\partial Ts}{\partial TB37V}$, $\frac{\partial Ts}{\partial TB85V}$, $\frac{\partial Ts}{\partial E_m 37V}$, and $\frac{\partial Ts}{\partial E_m 85V}$). (B) Histogram of the same network sensitivities.

trix clearly illustrates how the NN extracts the information from the inputs to derive the outputs.

In Figure 9 (resp. Figure 10), 20 samples of five NN sensitivities are represented. These samples are found using the Monte Carlo sampling strategy described in section 2.7. Associated with these samples are represented the histogram of the same NN sensitivities. As can be seen in these two figures, the uncertainty on the NN sensitivities has been largely reduced with the regularized NN (see Figure 10) compared to the nonregularized NN (see Figure 9).

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.

5 Conclusion and Perspectives

This study provides insight into how the NN model works and how the NN outputs are estimated. These developments draw the NN technique closer to

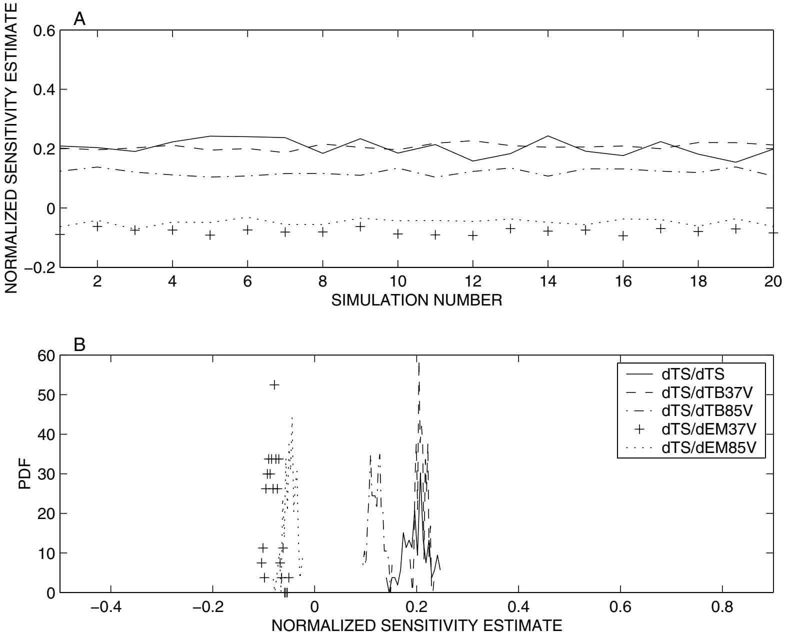

Figure 10: (A) Twenty samples of five regularized neural network sensitivities ($\frac{\partial Ts}{\partial Ts}$, $\frac{\partial Ts}{\partial TB37V}$, $\frac{\partial Ts}{\partial TB85V}$, $\frac{\partial Ts}{\partial E_m 37V}$, and $\frac{\partial Ts}{\partial E_m 85V}$) and (B) Histogram of the same network sensitivities.

better-understood classical methods, in particular linear regressions. With these older techniques, estimation of uncertainties of the statistical fit parameters is standard and is completely mandatory before the use of the regression model. Having at our disposal similar statistical tools for the NN establishes it on a stronger theoretical and practical basis so that NNs can be a natural alternative to traditional regression methods, with its particular advantage of nonlinearity. The tools are very generic and can be used for different linear or nonlinear regression models. A fully multivariate formulation is introduced. Its generality will allow future developments (like the iterative reestimation strategy or the fully Bayesian estimation of the hyperparameters).

The uncertainty of the NN weights can be large, but as we saw, the complex structure of correlation constrains this variability so that the NN outputs are a good statistical fit to the desired function. In the Bayesian approach, the prediction (estimation of the NN output) does not use just a specific estimation of the weights $\boldsymbol{w}^{\star}$ but rather integrates the outputs over the distribution of weights $P(\boldsymbol{w})$, the “plausibility” distribution. This approach is different in the sense that the prediction is given in terms of the PDF instead of the mode value. This article describes a technique to estimate the uncertainties of NN retrievals and provides a rigorous description of the sources of uncertainty.

The second application of the weight PDF is the estimation of NN Jacobians uncertainty. In this article, 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, providing the identification of situations where regularization needs to be used. As is often the case, regularization is a fundamental step of NN learning, especially for inverse problems (Badeva & Morosov, 1991; Tikhonov & 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 multicollinearities in data at the origin of the NN Jacobian variability. Our approach is able to make the learning process more stable and the Jacobians more reliable, and it can be more easily interpreted physically. All these tools are very general and can be used for other nonlinear models of statistical inference.

Together with the introduction of first-guess information first described in Aires et al. (2001), error specification makes the neural network approach even closer to more traditional inversion techniques like variational assimilation (Ide, Courtier, Ghil, & Lorenc, 1997) and iterative methods in general. Furthermore, quantities obtained from NN retrievals can now be combined with a forecast model in a variational assimilation scheme since the error covariances matrices can be estimated. These covariance matrices are not constant; they are situation dependent. This makes the scheme even better since it is now possible to assimilate only inversions of good quality (low-uncertainty estimates). Bad situations can be discarded from the assimilation or, even better, can be used as an "extreme" detection scheme that would, for example, signal the need for an increased number of simulations in an ensemble forecast. All these new developments establish the NN technique as a serious candidate for remote sensing in operational schemes, compared to the more classical approaches (Twomey, 1977). Our method provides a framework for the characterization, the analysis, and the interpretation of the various sources of uncertainty in any NN-based retrieval scheme. This makes possible improvements in the inversion schemes. Any fault that can be detected can be corrected: lack of data in the observation domains, errors of the model in some specific situations, or detection of extreme events. This should benefit a large community of NN users in meteorology or climatology.

Many new algorithmic developments can be pursued, and we provided a few ideas. For example, the network output uncertainties can easily be used for novelty detection (i.e., data that have not been used to train the network) or fault detection (i.e., data that are corrupted by errors, like instrument-related problems). Our determination of error characteristics can also be used with adaptive learning algorithms (i.e., learning when a small additional data set is provided after the main learning of the network has been done). The NN Jacobians can be used to express the various sources of uncertainty with even more detail, using Rodgers's approach (Rodgers, 1990). A source of uncertainty can be the presence of noise in NN inputs (Wright et al., 2000). Another technical development would be the optimization of the hyperparameters using an iterative reestimation strategy or evidence measure in a Bayesian framework (Neal, 1996; Nabney, 2002).

The Jacobians of a nonlinear model, such as the NN, are a very powerful concept. Many applications of the NN Jacobians can be derived from this study. The Hessian matrix can be used for many purposes (Bishop, 1996): (1) in several second-order optimization algorithms, (2) in adaptative learning algorithms (i.e., learning when a small additional data set is provided after the main learning of the network is done), (3) for identifying parameters (i.e., weights) not significant in the model as indicated by small diagonal terms H_{ii}, which is used by regularization processes like the "weight pruning" algorithm; (4) for automatic relevance determination, which is able to select the most informative network inputs and eliminate the negligible ones; and (5) to give a posteriori distributions of the neural weights as we do here. We also saw that the regularization of the Hessian matrix is essential if one wants to use it. A regularization solution is given in this article, but for some purposes, a few other techniques can also be used. In terms of the NN model, it allows us to obtain a robust model that will generalize well and suffer less from overfitting deficiencies. Jacobians can also be used to analyze how the NN model links the inputs and the outputs, in a nonlinear way. This capacity of measuring the internal structure of the NN is especially important when the NN is used as an analysis tool as proposed by Aires and Rossow (2003), where the NN Jacobians are estimated in order to analyze feedback processes in a dynamical. 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 article will directly benefit such studies.

Acknowledgments

We thank Ian T. Nabney for providing the Netlab toolbox from which some of the routines have been used in this work. F.A. thanks 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., Prigent, C., Rossow, W. B., & Rothstein, M. (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., & Rossow, W. B. (2003). Inferring instantaneous, multivariate and nonlinear sensitivities for the analysis of feedback processes in a dynamical system: The Lorenz model case study. *Q. J. Roy. Met. Soc., 129*, 239–275.

Aires, F., Rossow, W. B., Scott, N. A., & Chédin, A. (2002a). Remote sensing from the infrared atmospheric sounding interferometer instrument. 1: Compression, de-noising, and first guess retrieval algorithms. *J. Geophys. Res., 107*, no. D22, 4619.

Aires, F., Rossow, W. B., Scott, N. A., & Chédin, A. (2002b). Remote sensing from the IASI instrument. 2: Simultaneous retrieval of temperature, water vapor and ozone atmospheric profiles. *J. Geophys. Res., 107*, no. D22, ACH-7.

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

Badeva, V., & Morosov, V. (1991). *Problèmes incorrectement posés*. Paris: Masson.

Bates, D. M., & Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. New York: Wiley.

Bishop, C. (1996). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Crone, L., & Crosby, D. (1995). Statistical applications of a metric on subspaces to satelite meteorology. *Technometrics, 37*, 324-328.

Gelman, A. B., Carlin, J. S., Stern, H. S., & Rubin, D. B. (1995). *Bayesian data analysis*. London: Chapman & Hall.

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

Hertz, J., Krogh, A., & Palmer, R. C. (1991). *Introduction to the theory of neural computation*. Cambridge, MA: Perseus Books.

Ide, K., Courtier, P., Ghil, M., & Lorenc, A. C. (1997). Unified notation for data assimilation: Operational, sequential and variational. *J. Meteorol. Soc. J., 75*, 181–189.

Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). New York: Springer.

Koroliouk, V., Portenko, N., Skorokhod, A., & Tourbine, A. (1983). *Aide-mémoire de théorie des probabilités et de statistique mathématique*. Moscow: Edition Mir.

Krasnopolsky, V. M., Breaker, L. C., & Gemmill, G. H. (1995). A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager. *J. Geophys. Res., 100*, 11,033–11,045.

Krasnopolsky, V. M., Gemmill, G. H., & Breaker, L. C. (2000). A neural network multiparameter algorithm for SSM/I ocean retrievals: Comparison validations. *Remote Sens. Environ., 73*, 133–142.

Le Cun, Y., Denker, J. Y., & Solla, S. A. (1990). Optimal brain damage. In D. S. Touretzky (Ed.), *Advances in neural information processing systems, 2* (pp. 598–605). San Mateo, CA: Morgan Kaufmann.

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

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

Neal, R. M. (1996). *Bayesian learning for neural networks*. New York: Springer-Verlag.

Prigent, C., Aires, F., & Rossow, W. B. (2003). Land surface skin temperatures from a combined analysis of microwave and infrared satellite observations for an all-weather evaluation of the differences between air and skin temperatures. *J. Geophysical Research*, 108(D10), 4310.

Prigent, C., & Rossow, W. B. (1999). Retrieval of surface and atmospheric parameters over land from SSM/I: Potential and limitations. *Q.J.R. Met. Soc., 125*, 2379–2400.

Rivals, I., & Personnaz, L. (2000). Construction of confidence intervals for neural networks based on least squares estimation. *Neural Network, 13*, 463–484.

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

Rodgers, C. D. (1976). Retrieval of atmospheric temperature and composition from remote measurements of thermal radiation. *Rev. Geophys., 14*, 609–624.

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

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Saltelli, A., Chan, K., & Scott, E. M. (2000). *Sensitivity analysis*. New York: Wiley.

Stogryn, A. P., Butler, C. T., & Bartolac, T. J. (1994). Ocean surface wind retrievals from special sensor microwave imager data with neural networks. *J. Geophys. Res., 99*, 981–984.

Tarantola, A. (1987). *Inverse problem theory: Models for data fitting and model parameter estimation*. Amsterdam: Elsevier.

Tikhonov, A., & Arsenin, V. (1977). *Solutions of ill-posed problems*. Washington, DC: V. H. Vinsten.

Twomey, S. (1977). *Introduction to the mathematics of inversion in remote sensing and indirect measurements*. Amsterdam: Elsevier.

Vapnik, V. (1997). *The nature of statistical learning theory*. New York: Springer-Verlag.

Wright, W. A., Ramage, G., Cornford, D., & Nabney, I. T. (2000). Neural network modelling with input uncertainty: Theory and application. *Journal of VLSI Signal Processing, 26*, 169–188.

Received December 1, 2003; accepted April 29, 2004.