Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 1. Network weights

F. Aires

Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 1. Network weights

F. Aires

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

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

Received 22 September 2003; revised 17 February 2004; accepted 15 March 2004; published 21 May 2004.

[1] Neural network techniques have proved successful for many inversion problems in remote sensing; however, uncertainty estimates are rarely provided. This study has three parts. In this article, we present an approach to evaluate uncertainties (i.e., error bars and the correlation structure of these errors) of the neural network parameters, the so-called ''synaptic weights'' on the basis of a Bayesian technique. In contrast to more traditional approaches based on ''point estimation'' of the neural network weights (i.e., only one set of weights is determined by the learning process), we assess uncertainties on such estimates to monitor the quality of the neural network model. Uncertainties of the network parameters are used in the following two papers to estimate uncertainties of the network output [*Aires et al.*, 2004a] and of the network Jacobians [*Aires et al.*, 2004b]. These new 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 microwave and infrared observations over land. *INDEX TERMS:* 0933 Exploration Geophysics: Remote sensing; 3260 Mathematical Geophysics: Inverse theory; 3210 Mathematical Geophysics: Modeling; 3399 Meteorology and Atmospheric Dynamics: General or miscellaneous; *KEYWORDS:* remote sensing, uncertainty, neural networks

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

1. Introduction

[2] Neural network techniques have proved very successful in developing computationally efficient algorithms for geophysical applications, in particular for remote sensing applications. Since the late 80s, the number of applications has been steadily increasing. First applications concerned classification (*Key et al.* [1989]; see also the review paper in the work of *Paola and Showengerdt* [1995]). Clustering techniques were also developed [*Gualtieri and Whithers*, 1988; *Wilkinson et al.*, 1993], and a recent application for the characterization of vegetation using multisource satellite observations can be found in the work of [*Prigent et al.*, 2001]. We are interested in this study in the application of the neural networks (NN) for remote sensing of surface and atmospheric variables where the inverse radiative transfer function is represented by a NN nonlinear multivariate regression [*Kamgar-Parsi and Gualtieri*, 1990; *Escobar et al.*, 1993]. 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 retrieved variables (output variables). Furthermore, the NN models the inverse radiative transfer function in the atmosphere once and for all (i.e., global inversion), where some classical methods use an inversion procedure for each observation (i.e., local inversion). Until recently, neural network techniques did not use first guess a priori information to regularize the solution, this was a major handicap of this technique compared to classical iterative methods. A study showing that it is possible to introduce in the neural network inversion scheme such important information was presented in the work of [*Aires et al.*, 2001].

[3] As 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. A rigorous statistical approach requires not only good estimation of the inverse model, but also an uncertainty estimate of the model parameters (i.e., individual uncertainties plus correlation structure of these uncertainties). This is the common way of investigating the reliability of an inversion model. Until now the uncertainty of an NN statistical model has rarely been provided, probably because of the lack of adequate tools.

[4] In this paper, we present uncertainty estimate tools for real-world applications. First of all, this approach provides

uncertainty estimates for the parameters of the neural network (i.e., the network weights) which are determined by the learning procedure. A similar approach is but in a simpler presentation (i.e., monovariate case) used for example in the work of *Bishop* [1996], *Neal* [1996], or *Nabney* [2002]. Any rigorous determination of a parameterized model should evaluate how robust its parameters are [*Saltelli et al.*, 2000]. The robustness of the NN parameters is assessed via the Hessian matrix (second derivative) of the log likelihood with respect to the NN weights. This Hessian matrix can also be used for a variety of applications including "weight pruning" algorithms to improve the learning regularization, "automatic relevance detection" for selecting the more informative inputs or "novelty detection" to monitor outliers.

[5] Our approach not only provides uncertainty estimates for the parameters of the neural network but can also be used for the determination of a variety of other probabilistic quantities related to the overall uncertainty of the NN model. These applications can, first, use theoretical derivations when they are available. One such analytical application provides uncertainty estimates of the network output (error bars plus their correlation structure). This is the subject of *Aires et al.* [2004a].

[6] Second, when a theoretical derivation is too complex to be obtained, another possible application of weight uncertainty is based on an empirical estimation by using modern Bayesian statistics. One such application uses Monte Carlo (MC) simulations [*Gelman et al.*, 1995] that take into account the stochastic character related to parameter uncertainty. Such MC simulations are called "marginalizations." This will be the subject of *Aires et al.* [2004b]. The intensive computations required by MC simulation were the limiting factor for Bayesian analysis of real-world applications until computers became fast enough to make possible such approaches.

[7] Our technological developments are illustrated by application to a neural network 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 et al.*, 1994; *Krasnopolsky et al.*, 1995, 2000]. Our scheme 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. A database is carefully designed to train and test the neural network with special attention to its statistical representativeness on a global basis. It is derived from a global collection of coincident surface and atmospheric parameters from the combination of National Center for Environmental Prediction (NCEP) reanalysis, the International Satellite Cloud Climatology Project (ISCCP) data [*Rossow and Schiffer*, 1991], and the microwave emissivity atlases previously calculated.

[8] The NN technique is described in section 2. The theoretical formulation of a posteriori distributions for the NN weights is developed in section 3. Section 4 describes the remote sensing application as an example to illustrate some first results of the application of our weight uncertainty analysis (section 5). Section 6 concludes this study by highlighting potential applications of this technique.

2. Multilayer Perceptron

2.1. Model

[9] The multilayer perceptron (MLP) network is a nonlinear mapping model composed of parallel processors called "neurons." These processors are organized in distinct layers: The first "layer" $\boldsymbol{x} = (x_i;\ i \in S_0)$ of the mapping. (The group of input neurons in S_0 is not called a layer in the classical terminology. In this article, vectors will be in lowercase/bold and matrices will be in capital/bold. Notations are summarized in the Notation section.) The last layer S_L represents the output mapping $\boldsymbol{y} = (y_k;\ k \in S_L)$. The intermediate layers S_m $(0 < m < L)$ are called the "hidden layers." These layers are connected via neural links [*Aires et al.*, 2001, Figure 1b]: Any neurons, i and j, in two consecutive layers are connected with a synaptic weight w_{ij}.

[10] Each neuron j executes two simple operations. First, it makes a weighted sum of all of its inputs z_i: This signal is called the activity of the neuron

$$a_j = \sum_{i \in \text{Inputs}(j)} w_{ij} \cdot z_i. \tag{1}$$

Then it transfers this signal to its output through a so-called "activation function," often a sigmoid function such as $\sigma(a) = \tanh(a)$. The output z_j of neuron j in the hidden layer is then given by

$$z_j = \sigma(a_j) = \sigma\left(\sum_{i \in \text{Inputs}(j)} w_{ij}\, z_i\right). \tag{2}$$

Generally, for regression problems, the output units have no activation function ($\sigma(a) = a$). For example, in a one-hidden-layer MLP, the kth output x_k of the network is defined as

$$y_k(\boldsymbol{x}) = \sum_{j \in S_1} w_{jk}\sigma(a_j) = \sum_{j \in S_1} w_{jk}\ \sigma\left(\sum_{i \in S_0} w_{ij}\ x_i\right). \tag{3}$$

We represent the output vector using

$$\boldsymbol{y} = g_{\boldsymbol{w}}(\boldsymbol{x}), \tag{4}$$

where the MLP with weights $\boldsymbol{w}$ is a function $g_{\boldsymbol{w}}$. Equation (3) is the only computation required in the operational mode (once the synaptic weights have been determined by the learning procedure). A bias term for each neuron has been deliberately omitted to simplify the notation, although it is used in the neural network. It has been demonstrated [*Hornik et al.*, 1989; *Cybenko*, 1989] that any continuous function can be represented by a one-hidden-layer MLP with sigmoid functions σ.

2.2. Quality Criterion

[11] In this section, 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 related to the theory of statistical inference [*Vapnik*, 1997]: This "data" term is expressed using the difference between the target data and the neural network 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 appropriate distance metric is the Mahalanobis distance [*Crone and Crosby*, 1995]: this distance uses the covariance matrix that characterizes the Gaussian PDF. The ideal covariance matrix for the Gaussian PDF, denoted $\boldsymbol{C_{in}} = \boldsymbol{A_{in}^{-1}}$, is called the "intrinsic noise" (or natural variability) of the physical variables $\boldsymbol{y}$ to retrieve. If this matrix is estimated using a database $\mathcal{B} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{t}^{(n)}); n = 1, \ldots, N\}$ of N matched input/output couples, then it represents observation or simulation noise, depending on how $\mathcal{B}$ is generated. If the data set $\mathcal{B}$ is based on colocated measurements (satellite radiance measurements and geophysical variables), then $\boldsymbol{C_{in}}$ includes all the sources of variability between the satellite measurements and the geophysical radiance variables, e.g., colocation errors, spatial resolution differences, observational errors, or nonuniqueness of the satellite measurements for the set of geophysical variables due to insufficient spectral resolution (i.e., null space errors). If the data set $\mathcal{B}$ is based on geophysical variables and the corresponding satellite measurements are calculated with a forward radiative transfer code, then the sources of discrepancies specified by $\boldsymbol{C_{in}}$ include transfer model errors but not the colocation or resolution errors. Note that $\boldsymbol{C_{in}}$ takes into account only the intrinsic variability and not the error associated to 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 in the work of *Aires et al.* [2004a] how to estimate this quantity, but we suppose here that it is known.

[12] When fixed, $\boldsymbol{C_{in}}$ can be used as a discrepancy measure to evaluate differences between desired, $\boldsymbol{t}$, and retrieved, $\boldsymbol{y} = g_{\boldsymbol{w}}(\boldsymbol{x})$, outputs. The "data" quality term becomes:

$$E_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left[\varepsilon_y^{(n)}\right]^T \cdot \boldsymbol{A_{in}} \cdot \varepsilon_y^{(n)}, \tag{5}$$

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:

$$E_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{S_L}\left(t_k^{(n)} - y_k^{(n)}\right)^2, \tag{6}$$

where t_k is the kth desired output component, $y_k^{(n)}$ is the kth component of the neural network output vector $\boldsymbol{y}^{(n)} = g_{\boldsymbol{w}}(\boldsymbol{x}^{(n)})$, and S_L is the output layer of the neural network.

[13] In order to link the classical NN learning theory with Bayesian statistics, we introduce here minus the log likelihood which is defined by:

$$-\log(P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})), \tag{7}$$

where the probability $P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})$ is the likelihood of target $\boldsymbol{t}$ given input variable $\boldsymbol{x}$ and model parameter $\boldsymbol{w}$. Minimizing (7) is equivalent to maximizing the likelihood. If the distribution of P is Gaussian, minus the log likelihood is equal to the term in equation (5). This means that minimizing the errors in the discrepancy measure of equation (5) using the Mahalanobis distance is equivalent to maximizing the likelihood $P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})$ used in Bayesian statistics.

[14] 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 et al.*, 1991] is probably the most common regularization technique for NN. It implies that we choose a Gaussian distribution of weights as a priori information for NN weight uncertainty. This constrains network weights w_i to avoid large absolute values which are often the consequence of unstable learning and result on bad generalization properties of the NN. The weight decay term is expressed by:

$$E_r(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A_r} \cdot \boldsymbol{w}, \tag{8}$$

where $\boldsymbol{C_r} = \boldsymbol{A_r^{-1}}$ is the covariance matrix of the Gaussian a priori distribution for the network weights. This matrix associates a different variability to each weight w_i and describes a structure of correlation between them. Having different a priori distributions for the different weights can be important, especially for MLP networks where the weights from the input to the hidden layers and the weights from the hidden to the output layers can be in fundamentally different ranges. If such knowledge is not available for a particular structure of network weights, the matrix $\boldsymbol{A_r}$ in equation (8) is set to the identity matrix and the weight decay term is simplified to:

$$E_r(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{W} {w_i}^2, \tag{9}$$

where W is the dimension of $\boldsymbol{w}$. This simple regularization technique avoids over-fitting of the NN by penalizing large absolute values of the NN weights.

[15] The overall quality criterion that is minimized during the learning stage is the sum of the "data" and the regularization terms:

$$E(\boldsymbol{w}) = E_{\mathcal{D}}(\boldsymbol{w}) + E_r(\boldsymbol{w}). \tag{10}$$

[16] 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 the simplified forms in equations (6) and (9). 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 (10), but the "hyperparameters" $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$ are difficult to guess a priori. A method to estimate this very important information will be presented in section 3.

[17] In order to train the NN, we use the conjugate gradient descent (i.e., a gradient descent with improved search directions) to minimize $E(\boldsymbol{w})$. See *Bishop* [1996] for a detailed description of this optimization algorithm. The learning of the NN is done using a database of samples $\mathcal{B}$ but an additional independent data set is used to test the ability of the NN to generalize its behavior. This will ensure that the overfitting problem (fitting too well the learning database but generalizing badly to other samples) is avoided [*Geman et al.*, 1992].

3. Posterior Distribution of Network Weights

[18] The developments described in the work of *Bishop* [1996, section 10.1.4] are adopted here but generalized to the multiple output case using full covariance matrices, instead of scalars, to weight the information from the data and the a priori information in $E(\boldsymbol{w})$. The following developments are general and do not depend on the use of the NN statistical model and could be applied to other statistical nonlinear models.

3.1. Intrinsic Uncertainty of Targets

[19] 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 unless those from the inversion model, represented by uncertainties on the network weights $\boldsymbol{w}$, that are fixed in the conditional probability.

[20] 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 $\boldsymbol{C_{in}} = \boldsymbol{A_{in}^{-1}}$. In other words:

$$P(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z} e^{-\frac{1}{2}\boldsymbol{\varepsilon_y}^T \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\varepsilon_y}}, \tag{11}$$

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 the trained NN $g_{\boldsymbol{w}}$, is $\mathcal{B}$ that includes $\mathcal{D} = \{\boldsymbol{t}^{(n)}; n = 1, \ldots, N\}$, the set of output samples:

$$P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w}) = \prod_{n=1}^{N} P\left(\boldsymbol{t}^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{w}\right) \tag{12}$$

$$= \prod_{n=1}^{N} \frac{1}{Z} e^{-\frac{1}{2}\boldsymbol{\varepsilon_y}^{(n)T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\varepsilon_y}^{(n)}} \tag{13}$$

$$= \frac{1}{Z^N} e^{-\frac{1}{2}\sum_{n=1}^{N} \boldsymbol{\varepsilon_y}^{(n)T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\varepsilon_y}^{(n)}} \tag{14}$$

that we simplify by:

$$P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z^N} e^{-E_{\mathcal{D}}}, \tag{15}$$

using the definition of $E_{\mathcal{D}}$ in (5). 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 $\boldsymbol{t}$).

[21] The conditioning of the previous probabilities like in equation (12) 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.

3.2. Theoretical Derivation of Weight PDF

[22] 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 Bayes theorem:

$$P(\boldsymbol{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{w})P(\boldsymbol{w})}{P(\mathcal{D})}. \tag{16}$$

$P(\mathcal{D})$ 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 (10). So we can use for $P(\boldsymbol{w}|\mathcal{D})$ the expression for $P(\mathcal{D}|\boldsymbol{x}, \boldsymbol{w})$ from equation (15), the other terms in equation (16) being considered as constant normalization factors.

[23] 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:

$$E_{\mathcal{D}}(\boldsymbol{w}) = E_{\mathcal{D}}(\boldsymbol{w}^{\star}) + \boldsymbol{b}^T \cdot \triangle\boldsymbol{w} + \frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}, \tag{17}$$

where $\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}^{\star}$, $\boldsymbol{b}$ is the Jacobian vector given by:

$$\boldsymbol{b} = \nabla|_{\boldsymbol{w}}(E_{\mathcal{D}}(\boldsymbol{w})),$$

and $\boldsymbol{H}$ is the Hessian matrix given by:

$$\boldsymbol{H} = \nabla|_{\boldsymbol{w}}\left(\nabla|_{\boldsymbol{w}}(E_{\mathcal{D}}(\boldsymbol{w}))\right)$$

(see section 3.3). The linear term $\boldsymbol{b}^T \cdot \Delta\boldsymbol{w}$ 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.

[24] The second-order approximation leads to:

$$P(\boldsymbol{w}|\mathcal{D}) = \frac{1}{Z^N} e^{-E_{\mathcal{D}}(\boldsymbol{w}^{\star}) - \frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}} \tag{18}$$

$$\propto e^{-\frac{1}{2}\triangle\boldsymbol{w}^T \cdot \boldsymbol{H} \cdot \triangle\boldsymbol{w}}. \tag{19}$$

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.

[25] If a regularization term, such as the one described in equation (8), is used, then this probability becomes:

$$P(\boldsymbol{w}|\mathcal{D}) \propto e^{-\frac{1}{2}\triangle \boldsymbol{w}^T \cdot (\boldsymbol{H}+\boldsymbol{A_r}) \cdot \triangle \boldsymbol{w}}. \tag{20}$$

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 5.1) since $\boldsymbol{A_r}$ is the inverse of a positive definite matrix.

[26] It is worth noting here that the posterior distribution of network weights has some important possible applications. In the work of [*MacKay*, 1992], the estimate Hessian matrix $\boldsymbol{H}$ is used to compare the efficacy of different network topologies or different cost functions used to train the network. This is done by evaluating the "evidence," $P(\mathcal{D}|\mathcal{A}, \mathcal{R})$, of data, $\mathcal{D}$, given a neural architecture, $\mathcal{A}$, and a generic regularizer, $\mathcal{R}$. This "evidence" framework allow to compare totally different models in the light of the data.

3.3. Hessian Matrix for a One-Hidden-Layer Network

[27] 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:

$$H_{ij}(\boldsymbol{x}) = \left.\frac{\partial^2 E(\boldsymbol{w})}{\partial w_i \partial w_j}\right|_{\boldsymbol{x}}, \tag{21}$$

where w_i and w_j are two weights from the set $\boldsymbol{w}$. The Hessian is obviously dependent on NN weights $\boldsymbol{w}$ but we will omit this when refering to $\boldsymbol{H}$ for simplicity of notations. The Hessian of minus the log likelihood 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} (this 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; (5) to give a posteriori distributions of the neural weights as we do here.

[28] There are many ways of estimating the Hessian matrix, some are generic methods, some are specific to the multilayer perceptron [*Bishop*, 1996]. In this work, it is possible to retrieve a mathematical expression for the Hessian based on the neural network 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 $\mathcal{O}(W^3)$.

4. A Remote Sensing Example

[29] A neural network inversion scheme, including first guess information, 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 et al.*, 2001; *Prigent et al.*, 2003a]. See [*Prigent et al.*, 2003b] for the snow covered land case. The present study aims, in part, to provide uncertainty estimates for these retrievals. Both cloudy and clear-sky versions of this retrieval scheme have been developed but only the clear-sky case will be discussed here as an application example.

4.1. Neural Network Model With First Guess

[30] To avoid nonuniqueness and/or 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]. Introduction of a priori first guess information into a neural network model was first proposed by *Aires et al.* [2001]. With the a priori information included in the input of the classical MLP network, the neural transfer function becomes:

$$\boldsymbol{y} = g_{\boldsymbol{w}}\left(\boldsymbol{y}^b, \boldsymbol{x}^\circ\right), \tag{22}$$

where $\boldsymbol{y}$ is the retrieval (i.e., retrieved physical parameters), $g_{\boldsymbol{w}}$ is the neural network 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].

4.2. Learning Algorithm With First Guess

[31] The error back-propagation algorithm [*Rumelhart et al.*, 1986] is the learning algorithm that estimates the optimal network parameters $\boldsymbol{w}$ by minimizing a cost function $E(\boldsymbol{w})$, approaching as closely as possible the desired function (i.e., inverse of the radiative transfer equation). The data term usually used to derive $\boldsymbol{w}$ is in the form

$$E_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2}\int\int\int D(\boldsymbol{t}, g_{\boldsymbol{w}}(\boldsymbol{t}+\boldsymbol{\varepsilon}_b, \boldsymbol{x}+\boldsymbol{\eta}))^2 P(\boldsymbol{x}) P_{\eta}(\eta) P_{\varepsilon_b}(\varepsilon_b), \tag{23}$$

where D is a distance between $\boldsymbol{t}$ the desired output, and $\boldsymbol{y} = g_{\boldsymbol{w}}(\boldsymbol{t} + \boldsymbol{\varepsilon}, \boldsymbol{x} + \boldsymbol{\eta})$, the network output. We used here a continuous formulation in contrast to equation (5). D, the Euclidean or Mahalanobis distance for example, is implicitly here in the space of the physical parameters $\boldsymbol{t}$ (same dimension as S_L). $P(\boldsymbol{t})$ is the probability distribution function of the physical variables $\boldsymbol{t}$ that depends on their natural variability. $P_{\eta}(\boldsymbol{\eta})$ is the probability distribution function of the observation noise $\boldsymbol{\eta}$ (i.e., a Gaussian distribution with covariance matrix $\boldsymbol{E}$). $P_{\varepsilon_b}(\varepsilon_b)$ is the probability distribution function of the first guess error $\boldsymbol{\varepsilon}_b = \boldsymbol{y}^b - \boldsymbol{t}$. A Gaussian distribution with covariance matrix $\boldsymbol{B}$ is used here. This is generaly what has been used in traditional approaches like variational assimilation. However, if a more complex first guess error model is known a priori, like a state-dependent first guess error, then this model can be used during the learning of the neural network. Since results obtained with simulated and

observed data are similar, the error specifications that we use seem to be valid.

[32] To minimize the criterion of equation (23), and more generally in equation (10), we create a learning database

$$\mathcal{B} = \left\{ \left(\boldsymbol{x}^{o(n)}, \boldsymbol{t}^{(n)}, \boldsymbol{y}^{b(n)} \right); \quad n = 1, \ldots, N \right\} \tag{24}$$

that samples as well as possible all the probability distribution functions in equations (23) or (10) (see next section). Then, the practical criterion used during the learning stage is given by:

$$E_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} D\left(\boldsymbol{t}^{(n)}, g_{\boldsymbol{w}}\left(\boldsymbol{y}^{b(n)}, \boldsymbol{x}^{o(n)} \right) \right)^2. \tag{25}$$

[33] To sample the probability distribution function, $P(\boldsymbol{t})$, we select geophysical states $\boldsymbol{t}^n$ that cover all natural combinations and their correlations and by calculating $\boldsymbol{x}^{(n)} = RTM(\boldsymbol{t}^{(n)})$ with a Radiative Transfer Model (i.e., physical inversion). Alternatively, we could obtain these relationships from a ''sufficiently large'' set of colocated and coincident values of $\boldsymbol{x}$ and $\boldsymbol{t}$ (i.e., empirical inversion). To sample the first guess variability with respect to state $\boldsymbol{t}$ (i.e., sampling $P(\boldsymbol{y}^b|\boldsymbol{t})$), we use a first guess data set $\{\boldsymbol{y}^{b(n)}\}$; $n = 1, \ldots, N\}$. This data set can be a climatological data set or a 6-hour prediction (which might have better statistics of the errors, but would add model dependencies). The balance between reliance on the first guess and the direct measurements is then made automatically and optimally by the neural network during the learning stage.

[34] Once trained, the neural network $g_{\boldsymbol{w}}$ represents the inverse of the radiative transfer equation statistically. The neural network 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). For more details see [*Aires et al.*, 2001].

4.3. Neural Network Learning Stage

[35] 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 the work of *Aires et al.* [2001].

[36] The database is produced from global clear-sky data collected from a whole year of data, sampled every 3 hours, from July 1992 to June 1993 over land between 60°S and 80°N. Cloudy, snow- and ice-covered pixels are not considered (see [*Aires et al.* [2001] and *Prigent et al.* [2003b] for these cases). Over the 9,830,211 samples we have used only $N = 20{,}000$ samples, chosen randomly, to construct the learning database $\mathcal{B}$. We limit the number of samples because the computation of Hessian matrices are time consuming. Among the 20,000 samples, 15,000 are used for the training stage, and 5000 for the generalization test. The number of neurones in the hidden layer is estimated by monitoring the generalization errors of the NN for different choices, the optimal one is kept.

[37] The architecture of the neural network is a MLP with 17 inputs coding the seven SSM/I observations, and the first guess for *Ts*, *Ta*, *WV* and the 7 emissivities. The hidden layer is composed of 30 neurons. The output layer uses 9 neurones to represents the 9 physical variables to retrieve: *Ts*, *Ta*, *WV* and the 7 emissivities (see Table 1). The seven SSM/I channels have a Gaussian instrumental noise with 0.6 K standard deviation. First guess errors are described in the first column of Table 1. Inputs and outputs are first centered, and then normalized. The weights of the NN are initialized prior to training using a uniform distribution between −1 and 1.

Table 1. First Guess, Linear, and Neural Network (NN) Retrieval Root Means Square (RMS) Errors

	First Guess	Linear Retrieval	NN Retrieval
Ts, K	3.52	1.74	1.46
WV, Kg m^{-2}	7.99	4.94	3.83
E_m19V	1.50 10^{-2}	0.58 10^{-2}	0.49 10^{-2}
E_m19H	1.84 10^{-2}	0.59 10^{-2}	0.49 10^{-2}
E_m22V	1.66 10^{-2}	0.62 10^{-2}	0.56 10^{-2}
E_m37V	1.43 10^{-2}	0.58 10^{-2}	0.49 10^{-2}
E_m37H	1.80 10^{-2}	0.59 10^{-2}	0.50 10^{-2}
E_m85V	1.76 10^{-2}	0.77 10^{-2}	0.68 10^{-2}
E_m85H	2.14 10^{-2}	0.95 10^{-2}	0.82 10^{-2}

[38] The learning algorithm and the network architecture are able to infer the inverse radiative transfer equation with the $N = 20{,}000$ samples. Among this data set, 3/4 are used for the learning data set, the other 1/4 is used as a validation data set to test the generalization properties of the neural network. All the learning errors decrease extremely fast and stabilize after few thousand iterations (each iteration involving the whole learning database $\mathcal{B}$). Training is stoped when the RMS errors in the training data set and in the generalization data set stabilize. Over-training is avoided by controling the generalization curves. This shows how fast and efficient the conjugate gradient optimization algorithm is. This learning stage determines the optimal weights $\boldsymbol{w}^\star$.

[39] In Figure 1, the first guess and the retrieved quantities are compared to the target data. For each of the 9 outputs, the neural network is able to concentrate the scatterplot toward the diagonal which means that the RMS errors of the retrieved variables are reduced compared to those of the first guess. This is summarized in Table 1 that 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.

5. Posterior Distribution of Network Weights

[40] A rigorous model parameterization should always be followed by a sensitivity analysis [*Saltelli et al.*, 2000]. It can only be trusted and used when its sensitivity to all the hypotheses used is known. We propose, in this section, to perform this sensitivity analysis by estimating the uncertainty of the parameters of the neural network, i.e., the network weights.

5.1. Neural Network Hessian Regularization

[41] Figure 2 illustrates the Hessian $\boldsymbol{H}$ computed using the data set $\mathcal{B}$. The grey scaling goes from black to white, where black and white are, respectively, for minimum and maximum values in $\boldsymbol{H}$. The features of this matrix are related to the neural network structure. The network weights between the input and hidden layer are more related than those between the hidden and output layer. Once this matrix is inverted, $\boldsymbol{H}^{-1}$ (not shown) becomes the covariance matrix

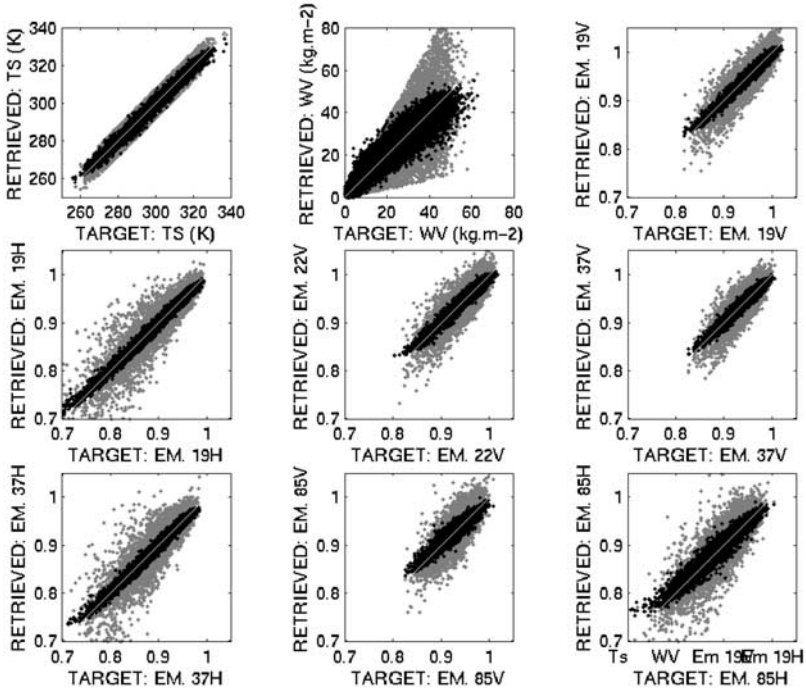

Figure 1. Scatterplot of the nine network outputs: first guess (gray) or retrieved (black) against actual variables.

of the network weights $\boldsymbol{w}$. The structure of $\boldsymbol{H}$ shown in Figure 2 means that the uncertainty of the weights between the hidden and output layer is larger than the uncertainty of the weights between the input and hidden layer. Of course, this matrix should always be symmetric.

[42] A few comments about the inversion of $\boldsymbol{H}$ are required. This matrix can be very large when the neural network 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, the eigen-values of $\boldsymbol{H}$ too close to zero or even negative, making the inversion numerically impossible.

[43] We commented already in section 3.2 that the gradient $\boldsymbol{b}$ in equation (17) 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 neural network converges close enough toward the optimal solution $\boldsymbol{w}^{\star}$. Monitoring the convergence algorithm could be enhanced by checking in parallel the positive definite character of the corresponding Hessian of the network.

[44] 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 neural network can produce an ill-conditioned Hessian matrix $\boldsymbol{H}$. A possible solution 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.

[45] To determine the factor λ representing the right trade-off, we use three regularization criteria together with

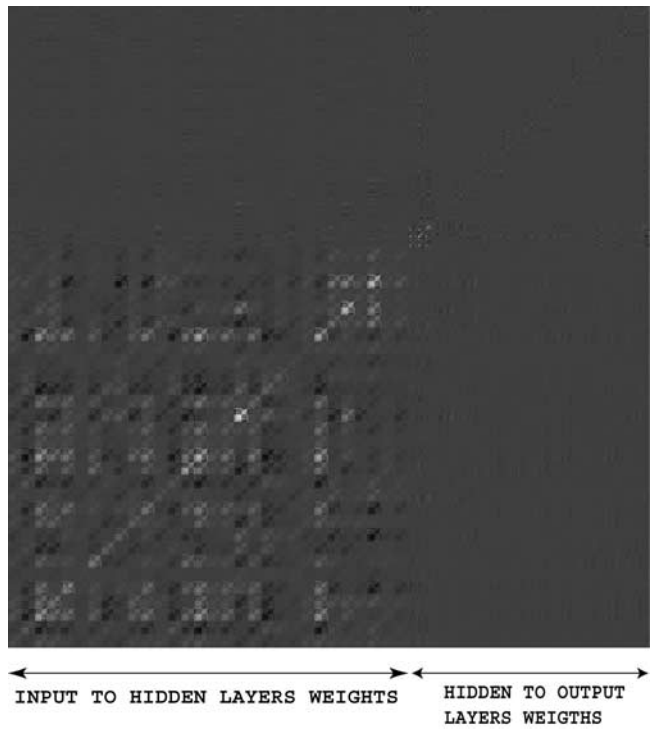

Figure 2. Hessian matrix of the log likelihood with respect to network weights: The gray scaling goes from black to white, where black and white are for the minimum and maximum value in $\boldsymbol{H}$, respectively.

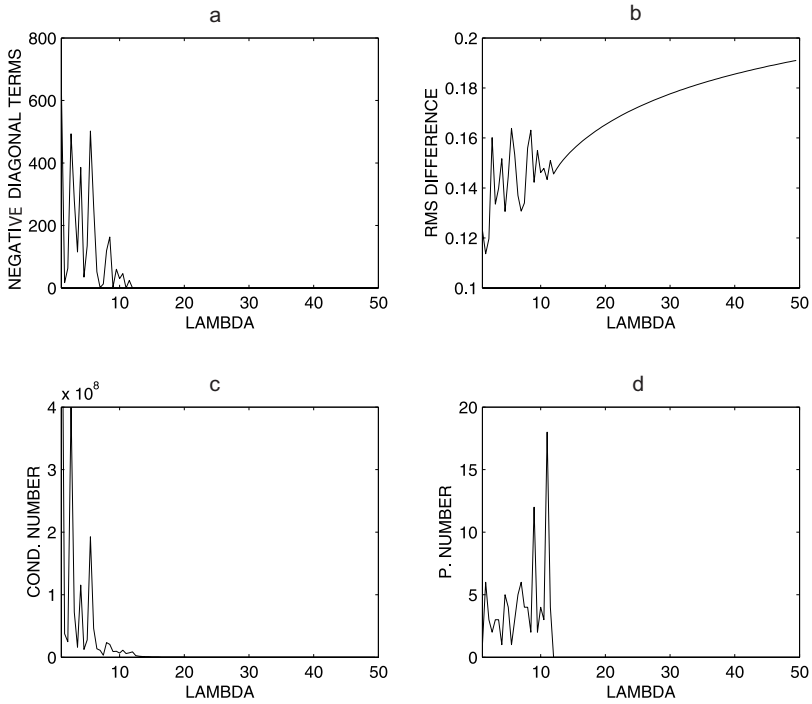

Figure 3. Quality criteria for variable λ: (a) the number of negative diagonal terms in the matrix, (b) RMS differences between the square root of the positive diagonal elements of the matrices, (c) the condition number with respect to inversion, and (d) the P number which is a positive integer if the matrix is not positive definite and zero otherwise. See text.

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 latter quantity measures the differences that the regularization has introduced in the standard deviations of the two covariance matrices.

[46] In Figure 3, the variations of these four quantities for an increasing λ, from 0 to 50, are shown. The good compromise is found to be $\lambda = 12.0$: the regularization criteria are satisfactory (positive definite matrix, all diagonal terms positive, minimum condition number) while the discrepancy measure is still small.

[47] The validity of the Gaussian approximation is a key point in this work [*Neal*, 1996]. The high number of samples in our experiment is important and could explain our good and coherent results [*Thodberg*, 1996]. The main condition is that the weights of the NN are close to the optimal weights $W^\star$. This might not be enough because the problem is so complex and the parameters interdependent, that the weights after the learning can be optimum for most of the weights but not for some of them. In that case, the weights can be in the saddle point of the error function instead of a perfect global minima. The fact that we need to regularize the Hessian matrix by adding a diagonal term in order to obtain a definite positive matrix is a signal that the approximation is not totally satisfying. We think that this issue is related to the approximation of the hyperparameter matrix A_r that could regularize the Hessian matrix H (they are added together in the technique). We will investigate these points in a future study on the estimation of the hyperparameters (through integration, maximum likelihood, or our simple iterative scheme). This will provide another solution for the regularization of $\boldsymbol{H}$, probably preferable, by using a regularization term in the quality criterion as already mentioned in this paper in section 2.2. Other regularization techniques could be used, some are statistical, other are more physically oriented [*Tikhonov and Arsenin*, 1977; *Badeva and Morosov*, 1991; *Aires et al.*, 1999, 2002]. This will be the subject of future work.

5.2. PDF of Network Weights

[48] 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 3 that this distribution follows a Gaussian PDF with mean $\boldsymbol{w}^\star$ and covariance matrix $\boldsymbol{H}^{-1}$.

[49] In Figure 4, 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

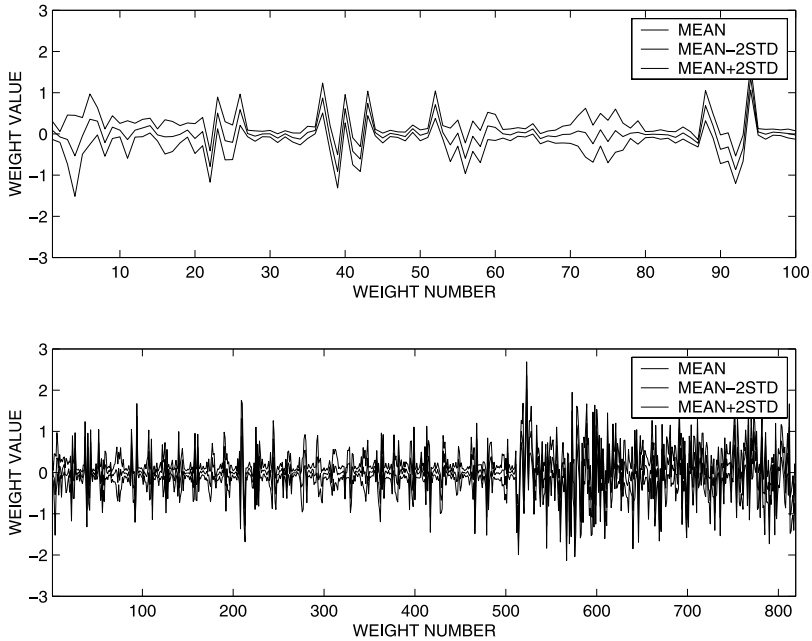

Figure 4. (top) Mean network weights $\boldsymbol{w}^{\star} \pm 2$ standard deviation: the first 100 NN weights corresponding to input/hidden layer connections, and (bottom) all 821 NN weights with weight 510 to 819 for hidden/output layer connections.

layer to the output layer, is just a linear postprocessing of the hidden layer.

5.3. Interpretation of Weight Uncertainty

[50] It is possible to know much more than just the output estimates of a NN. From the distribution of weights, samples $\{\boldsymbol{w}^r; r = 1, \ldots, R\}$ of NN weights can be chosen. Each of the R samples $\boldsymbol{w}^r$ represent a particular NN. Together, they represent the uncertainty on the NN weights. These samples can be used later on 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 [*Aires et al.*, 2004b, Appendix A].

[51] Figure 5 presents a few samples from this simulation using the eigen-decomposition-based sampling approach. 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 a 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. The 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 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.

[52] These are just a few reasons to explain why the network weights can vary and still provide a good general fitting model. Variability of the network weights is considered as a natural variability, inherent to the neural technique. Furthermore, what is important for the NN user is not directly the variability of the weights but rather the uncertainty that this variability produce in the network outputs [*Aires et al.*, 2004a] or in even more complex quantities such as the network Jacobians [*Aires et al.*, 2004b].

6. Conclusion and Perspectives

[53] This study advances the use of neural networks beyond the black box conception often associated with them by proposing a way to provide insight into how the NN model actually works and how the NN outputs are estimated. These developments draw the NN technique closer to 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 neural networks can be a natural alternative to traditional regression methods, with its particular advantage of nonlinearity. An important application of the posterior distribution of network weights is that different neural network models can be compared in the light of the observations available [*MacKay*, 1992].

[54] As a consequence, this study also draws the NN inversion method closer to classical inversion techniques. Such links were already investigated in the work of *Aires et al.* [2001], introducing a priori first guess information into

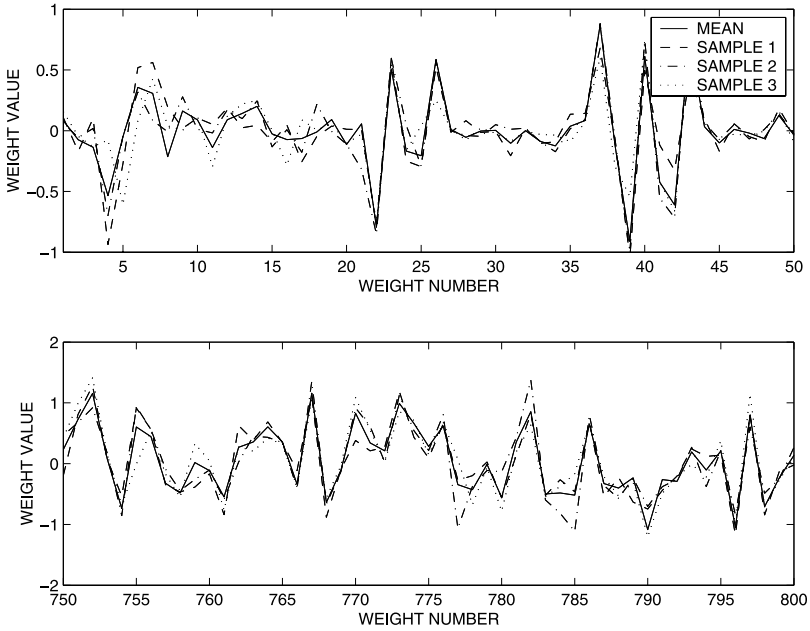

Figure 5. (top) Mean network weights $\boldsymbol{w}^\star$ plus five simulation samples from the weights a posteriori distribution: NN weights corresponding to input/hidden layer connections, and (bottom) weights corresponding to hidden/output layer connections.

the neural network. The parameter uncertainty estimate, together with the applications developed in the work of *Aires et al.* [2004a, 2004b], are additional bridges between neural networks and techniques like variational assimilation. This should benefit a large community of neural network users in meteorology/climatology. Furthermore, the tools that we have developed are generic and have been used, or can be used, for other nonlinear models in the context of statistical inference. Its generality will allow future development like a Bayesian estimation of the hyperparameters.

[55] Many applications can be derived from this study. A few suggestions for future developments were mentioned in the section 3.3. 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 paper but for some purposes a few other techniques can also be used and we will comment on some of them in the next two companion papers.

[56] These results about the characterization of the NN weight uncertainties are very important and will be used in the work of *Aires et al.* [2004a, 2004b]. The uncertainty of the 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 [see *Aires et al.*, 2004a]. Interestingly, even if the PDF of the NN weights is Gaussian, the PDF of outputs can be non-Gaussian since the NN is nonlinear. The second application of the weight PDF will be the estimation of neural network Jacobians uncertainty. This will be useful when the NN is used for diagnostic purposes: a regularization technique will be proposed to extract realistic Jacobians.

Notation

- $\boldsymbol{y}$ vector of physical variables to retrieve, outputs of the NN.
- $\boldsymbol{t}$ target vector of physical variables in data set $\mathcal{B}$.
- α weight for the regularization term linked to the a priori general variance of the weights.
- β inverse of the observation noise variance for all outputs.
- $\boldsymbol{y}^b$ first guess a priori information for $\boldsymbol{x}$, inputs of the NN.
- ε_v generic error symbol for variable v.
- $\boldsymbol{x}$ observations vector, inputs of the NN.
- $\boldsymbol{x}(\boldsymbol{y})$ = $RTM(\boldsymbol{y})$, radiative transfer function for the physical variable $\boldsymbol{x}$ (also a vector).
- $\boldsymbol{x}^\circ$ SSM/I brightness temperature observations.
- $\boldsymbol{\eta}$ SSM/I instrumental noise.
- P_v generic probability measure for variable v.
- $\boldsymbol{B}$ = $\langle \boldsymbol{\varepsilon}_b^T \cdot \boldsymbol{\varepsilon}_b \rangle$, covariance matrix of the first guess errors.
- $\boldsymbol{E}$ = $\langle \boldsymbol{\eta}^T \cdot \boldsymbol{\eta} \rangle$, covariance matrix of the measurement errors.
- $\boldsymbol{C_0}$ (= $\boldsymbol{A_0}^{-1}$), covariance matrix of total error on retrieved physical variables y.
- $\boldsymbol{C_{in}}$ (= $\boldsymbol{A_{in}}^{-1}$), covariance matrix of intrinsic noise on physical variables $\boldsymbol{y}$, equivalent to $1/\beta$ in traditional Bayesian formulation.
- $\boldsymbol{C_r}$ (= $\boldsymbol{A_r}^{-1}$), covariance matrix for weight regularization, equivalent to $1/\alpha$ in traditional Bayesian formulation.
- $\boldsymbol{A}$ generic inverse matrix of associated covariance matrix.
- $\boldsymbol{I}$ identity matrix.

$\boldsymbol{b}$ $= \nabla|_{\boldsymbol{w}} (E_{\mathcal{D}}(\boldsymbol{w}))$.
$\boldsymbol{H}$ $= \nabla|_{\boldsymbol{w}} (\nabla|_{\boldsymbol{w}} (E_{\mathcal{D}}(\boldsymbol{w})))$, the Hessian matrix of the log likelihood.
$.^T$ transposition operator.
$\delta(\cdot)$ Kronecker operator.
D generic distance.
$\mathcal{A}$ generic neural architecture.
$\mathcal{R}$ generic regularizer.
a_i activity of neuron i.
σ sigmoid function of the neural network.
z_i output of the neuron i.
S_i number of neurons in network layer i.
L number of layers in the network network.
$g_{\boldsymbol{w}}$ neural network model, or transfer function for our application.
$\boldsymbol{w}$ $\{w_i;\ i = 1, \ldots, W\}$, the vector of the network weights.
W dimension of $\boldsymbol{w}$.
R number of samples in $\{\boldsymbol{w}^r;\ r = 1, \ldots, R\}$, the sample of network weights.
$\mathcal{B}$ learning database, that includes outputs $\mathcal{D}$.
$\mathcal{D}$ target or network output database.
N number of samples in $\mathcal{D}$ and $\mathcal{B}$.
$E(\boldsymbol{w})$ quality criterion for classical neural network learning phase.
$E_{\mathcal{D}}(\boldsymbol{w})$ data term of the quality criterion.
$E_r(\boldsymbol{w})$ regularization term in the quality criterion.
λ regularization factor for the inversion of $\boldsymbol{H}$.

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

References

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

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

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

Aires, F., C. Prigent, and W. B. Rossow (2004a), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 2. Output errors, *J. Geophys. Res.*, *109*, D10304, doi:10.1029/2003JD004174.

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

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

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

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

Cybenko, G. (1989), Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.*, *2*, 303–314.

Escobar, J., A. Chédin, F. Chéruy, and N. A. Scott (1993), Réseaux de neurones multicouches pour la restitution de variables thermodynamiques atmosphériques à l'aide de sondeurs verticaux satellitaires, *C. R. Acad. Sci. Paris*, *317*(2), 911–918.

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

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

Gualtieri, J. A., and J. Whithers (1988), Goddard researchers simulate neural networks using parallel processing, *NASA Inf. Syst. Newslett.*, *15*, 12–15.

Hertz, J., A. Krogh, and R. C. Palmer (1991), *Introduction to the Theory of Neural Computation, Santa Fe Institute Studies in the Sciences of Complexity: Lecture Notes*, vol. 1, Addison-Wesley-Longman, Reading, Mass.

Hornik, K., M. Stinchcombe, and H. White (1989), Multilayer feedforward networks are universal approximators, *Neural Networks*, *2*, 359–366.

Kamgar-Parsi, B., and J. A. Gualtieri (1990), Solving inversion problems with neural networks in *Proceedings of the International Joint Conference on Neural Networks*, vol. III, pp. 955–960, Int. Neural Net Soc., Inst. of Electr. and Electron. Eng., New York.

Key, J., A. Maslanic, and A. J. Schweiger (1989), Classification of merged AVHRR and SMMR arctic data with neural network, *Photogramm. Eng. Remote Sens.*, *55*(9), 1331–1338.

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

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

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

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

Paola, J. D., and R. A. Showengerdt (1995), A review and analysis of backpropagation neural networks for classification of remotely sensed multi-spectral imagery, *Int. J. Remote Sens.*, *16*(16), 3033–3058.

Prigent, C., F. Aires, W. B. Rossow, and E. Matthews (2001), Joint characterization of the vegetation by satellite observations from visible to microwave wavelengths: A sensitivity analysis, *J. Geophys. Res.*, *106*, 20,665–20,685.

Prigent, C., F. Aires, and W. B. Rossow (2003a), 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. Geophys. Res.*, *108*(D10), 4310, doi:10.1029/2002JD002301.

Prigent, C., F. Aires, and W. B. Rossow (2003b), Retrieval of surface and atmospheric geophysical variables over snow and ice from satellite microwave observations, *J. Appl. Meteorol.*, *42*, 368–380.

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.

Rossow, W. B., and R. A. Schiffer (1991), ISCCP cloud data products, *Bull. Am. Meteorol. Soc.*, *72*, 2–20.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. I, *Foundations*, edited by D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, pp. 318–362, MIT Press, Cambridge, Mass.

Saltelli, A., K. Chan, and E. M. Scott (2000), *Sensitivity Analysis*, John Wiley, Hoboken, N. J.

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

Thodberg, H. H. (1996), A review of Bayesian neural networks with an application to near infrared spectroscopy, *IEEE Trans. Neural Networks*, *7*(1), 56–72.

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

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

Wilkinson, G. G., C. Kontoes, and C. N. Murray (1993), Recognition and inventory of oceanic clouds from satellite data using an artificial neural network technique, in *Demethylsulphide: Oceans, Atmosphere and Climate*, *Proceedings of the International Symposium*, pp. 392–399, Kluwer Acad., Norwell, Mass.

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