A general formulation of nonlinear least squares regression using multi-layered perceptrons

Fouad BADRAN,Yann STÉPHAN,Nabil METOUI and Sylvie THIRIA *†‡§

November 22, 2000

Abstract

Non linear regression and non linear approximation are widely used for data analysis. In many applications, the aim is to build a model linking observations and parameters of a physical system. Two cases of increasing complexity have been studied: the case of deterministic inputs and noisy output data and the case of noisy input and output data. We present in this paper a general formulation of non linear regression using multilayered Perceptrons. Regression algorithms are derived in the three cases. In particular, a generalized learning rule is proposed to deal with noisy input and output data. The algorithm enables not only to build an accurate model but also to refine the learning data set. The algorithms are tested on two real-world problem in Geophysics. The good results suggests that multilayered Perceptrons can emmerged as an efficient nonlinear regression model for a wide range of applications.

Non linear regression, Multi-layered Perceptrons, Back-propagation, Uncertainties.

1 Introduction

Non linear regression and non linear approximation are widely used for data analysis in particular in Geophysical sciences as Meteorology and Oceanography. In many applications, the aim is to build a non linear model between two characteristic parameters $\mathbf{x}$ and $\mathbf{y}$ of a physical system such as:

$$\mathbf{y} = \mathbf{G}(\mathbf{x}) \tag{1}$$

Generally, it is possible to collect observations to have a relevant statistical set of couples $(\mathbf{x}^{obs}, \mathbf{y}^{obs})$. Both parameters are obtained by some experimental devices and are

*Fouad BADRAN is at the Centre d'Etude et De Recherche en Informatique du CNAM (CEDRIC) 292, rue Saint-Martin 75003 PARIS, FRANCE and in LODYC.

†Yann STÉPHAN is in the Centre Militaire d'Océanographie EPSHOM, BP 426 29275 BREST cedex, FRANCE.

‡Nabil METOUI is at CEDRIC.

§Sylvie THIRIA is at the Laboratoire d'Océanographie DYnamique et de Climatologie (LODYC) 4, Place Jussieu 75005 PARIS, FRANCE.

approximated measurements of actual values. It can be assumed that the observed data ($\mathbf{x}^{obs}$, $\mathbf{y}^{obs}$) are a realization of random variables with probability density functions

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}/\mathbf{x})p(\mathbf{x}) \tag{2}$$

Denoting respectively by $\mathbf{y}^{true}$ and $\mathbf{x}^{true}$ the "true" fields, it is generally assumed that a particular observation is written as the sum of the actual value $\mathbf{y}^{true}$ of $\mathbf{y}$ affected by a noise $\boldsymbol{\delta}$:

$$\mathbf{y}^{obs} = \mathbf{y}^{true} + \boldsymbol{\delta} \tag{3}$$

In the same way, $\mathbf{x}$ can be described as the "true" parameters spoilt by an additive noise $\boldsymbol{\epsilon}$, such as:

$$\mathbf{x}^{obs} = \mathbf{x}^{true} + \boldsymbol{\epsilon} \tag{4}$$

The central point of this study is to built a model $\mathbf{G}$ (which is called the forward model hereafter) which is optimal in the sense of a nonlinear least-square criterion and which takes into account the uncertainties on the observations. This implies that the problem to solve is to determine $\mathbf{G}$ such as:

$$\mathbf{y}^{true} = \mathbf{G}(\mathbf{x}^{true}) \tag{5}$$

This is a nonlinear regression problem which can be tackled by several techniques.The Bayesian formalism allows a general formulation of the problem, taking into account most of the different uncertainties appearing during the regression process. By doing specific hypothesis on the regression function, on the nature of the different noises ($\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$) and on the accuracy of the estimator, it is possible to retrieve the classical regression methods used in Statistics. They give rise to different modelizations which include parametric and non parametric models.

The hypothesis on the noise probability density functions are determinant. Most of the time, it is supposed that these density functions are Gaussian. In such a case, the maximum likelihood model provides simple expressions derived from the least mean squares criterion. Many regression methods are based upon this approach. Their main difference lies in the functions used during the regression process (linear, polynomial ...). The Neural approach by using Multi-layered Perceptrons (MLP) have proved to be efficient nonlinear approximators [4] [10] [19]. If many theoretical results on these models are now available, there is still a need to formalize the neural approach and to show its ability to systematically account for uncertainties. This is the basic motivation of this paper.
In this work, we show how the Bayesian formalism with Gaussian assumptions leads to different neural models of increasing complexity. Different cases are considered depending

on whether the noise appears on data: only the output $\mathbf{y}$ can be noisy or both the input $\mathbf{x}$ and the output $\mathbf{y}$, we consider the general case where the output noise depends on the input $\mathbf{x}$. The paper introduces the different neural models which allow to deal with the different cases together with their learning algorithms. As the handling is sometimes difficult, we detail the operational phases for each of them.

The simplest neural model corresponds to weighted least square regression minimization and can be used when the input $\mathbf{x}$ is deterministic and there is no specific knowledge about the density function of the output noise. Some fundamental results about this case will be presented in section 2.

A more sophisticated model allows us to introduce different hypothesis on the noise data distributions. Section 3 presents the maximum likelihood formulation and shows how such a modelization allows to estimate the output uncertainties (covariance matrix of the noise) in case of non noisy input observations $\mathbf{x}^{obs}$ and noisy ouput observations $\mathbf{y}^{obs}$.

The more general problem is to determine the regression function when dealing with both noisy $\mathbf{x}$ and $\mathbf{y}$ data. As this case requires a more sophisticated neural model, the models' behavior is presented in details using simulated data.This problem is presented in Section 4 where a general methodology to use MLP regression is derived.

The application of non linear regression using MLPs is discussed in Section 5. The main theoretical results and the efficiency of the neural algorithms are illustrated in two actual geophysical problems (the NSCAT scatterometer transfer function and the inversion of Ocean color) .

2 Least Mean Square and MLP

In this section, we address the easiest problem which is to assume that the inputs $\mathbf{x}$ are known without error ($\boldsymbol{\epsilon} = 0$).

It is well-known that Multi-layered Perceptrons represent a family of functions from R^p to R^q:

$$\begin{array}{rcl} R^p & \longrightarrow & R^q \\ \mathbf{x} & \longmapsto & \mathbf{y} = \mathbf{F}(\mathbf{W}, \mathbf{x}) \end{array} \tag{6}$$

where $\mathbf{W}$ is the matrix of weights, which allows the regression of a multi-dimensional variable $\mathbf{y}$ with respect to a multi-dimensional variable $\mathbf{x}$. The regression consists in determining the weights $\mathbf{W}$ of the function $\mathbf{F}$ to estimate $\mathbf{G}$. In the following s_k will denote the input of neuron k, f_k its transfer function and o_k its output. So we have:

$$o_k = f_k(s_k) \tag{7}$$

If every transfer function f_k is continuous and derivable, the MLPs represent a family of functions which is suited for non linear regression. Since the late 1980, several papers described how they could be used [7] [8] [11] [18] [30]. The basic results of these works

are that any multi-dimensional continuous function defined on a compact set can be approximated by a MLP with (at least) one hidden layer and linear output neurons. Learning generally consists in determining the weights $\mathbf{W}$ by minimizing a cost function i.e. a measure of the mismatch between target values and predicted values [4]. A simple expression of this function is the generalized least square error function:

$$R(\mathbf{W}) = \iint \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{M}(\mathbf{x}) \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \tag{8}$$

Where $\mathbf{M}(\mathbf{x})$ is a symetric define positive matrix.

Often the matrix $\mathbf{M}(\mathbf{x})$ is set to the inverse of the covariance matrix $\mathbf{C}_y(\mathbf{x})$ of the conditional random variable $\mathbf{Y}/\mathbf{x}$. In general, the covariance matrix $\mathbf{C}_y^{-1}(\mathbf{x})$ is a function of $\mathbf{x}$, its diagonal coefficients representing the variance of the noise components added to $\mathbf{x}$ and the others terms their covariance. In appendix, it is shown that the minimization of (8) provides a minimum of:

$$\int \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_{\mathbf{y}}^{-1}(\mathbf{x}) \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{x}) d\mathbf{x} \tag{9}$$

Where $\mathbf{E}(\mathbf{Y}/\mathbf{x})$ is the conditional mean vector of the observation $\mathbf{y}^{obs}$.
In equation (9) it is shown that a good minimum of (8) gives a good approximate of:

$$\mathbf{E}(\mathbf{Y}/\mathbf{x}) = \int \mathbf{y} p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \tag{10}$$

and that the outputs of the MLP are such that:

$$\mathbf{F}(\mathbf{W},\mathbf{x}) \approx \mathbf{E}(\mathbf{Y}/\mathbf{x}) \tag{11}$$

In some cases, when $\mathbf{C}_y(\mathbf{x}) = \mathbf{C}_y$ñ does not depend on $\mathbf{x}$, it is possible to give some information on the accuracy of the approximation. The cost function $R(\mathbf{W})$ becomes:

$$R(\mathbf{W}) = \sum_{j=1}^{q} \int \Big(\frac{\mathbf{u}_j^T.\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{u}_j^T.\mathbf{F}(\mathbf{W},\mathbf{x})}{\sigma_j}\Big)^2 p(\mathbf{x})\, d\mathbf{x} + \text{constant} \tag{12}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q)$ is the orthonormal matrix of the eigenvectors of $\mathbf{C}_y$ and σ_j^2 the eigen value associated to $\mathbf{u}_j$. In this case too, equation (12) shows that the MLP approximates the mean value of $\mathbf{y}$ conditionally to $\mathbf{x}$ and that the accuracy of this approximation can be measured on the principal axes $\mathbf{u}_j$ with variance σ_j^2. The maximum of accuracy can be obtained on the axes where σ_j^2 are the smallest. However, the theoretical cost function R given by of equation (12) is oftendifficult to get. Rather, a finite set of independant observations $D = \big\{(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs}), i = 1 \cdots N^{obs}\big\}$ can be obtained. Learning is done on this set by minimizing the empiric risk defined by:

$$R_{emp}(\mathbf{W}) = \sum_{i=1}^{N^{obs}} R_i \tag{13}$$

where

$$R_i = \Big(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{obs})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}_i^{obs}) \Big(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{obs})\Big) \tag{14}$$

which is a discrete approximation of the theoretical cost function R given by equation (9).

When the regressor $\mathbf{F}(\mathbf{W}, \mathbf{x})$ is an MLP, the accuracy of the approximation may be affected for two reasons. On one hand, the model may be badly chosen (too few or too many weights, inappropriate set of functions, inacurate values of the regularization parameters ...). On the other hand, the observation set may be inconsistent with the true distribution of the variable to regress. However, if these two constraints are overcome, the general result of relation (11) holds. The minimum of (13) with repect to the weights ($\mathbf{W}$) is determined by using back-propagation-to-weights for which several algorithms exist, as the steepest-gradient descent [21] [4] [10] [19] the conjugate-gradient descent and (quasi)Newton methods [2].

The underlying idea of back-propagation with MLP is to reach the minimum of an appropriate cost function R_{emp} using a gradient procedure. This is done by computing:

- Phase 1: the partial derivatives with respect to the input of the neurons of the output layer (denoted initial errors hereafter)

- Phase 2: in a recursive way, the partial derivatives with respect to the MLP's parameters .

Phase 2 which is the recursive part of the algorithm depends only on the MLP's architecture. The process is initialized during phase 1 which computes the initial errors. So the minimization of two different cost functions R which uses the same MLP's architecture only differs on the initial values of the recursion.

Assuming that the output units are linear, for a given $\mathbf{x}_i^{obs}$ the initialization of the (Phase 1) is given by :

$$\frac{\partial R_i}{\partial \mathbf{S}} = -2\, \mathbf{C}_y^{-1}\big(\mathbf{x}_i^{obs}\big)\, \big(\mathbf{y}_i^{obs} - \mathbf{F}\big(\mathbf{W}, \mathbf{x}_i^{obs}\big)\big) \tag{15}$$

where $\mathbf{S} = (s_1, \ldots, s_q)^T$ denotes the vector whose components are the inputs of the neurons of the last layer. As shown by equation (15), minimizing the cost function $R_i(\mathbf{W})$ requires the knowledge of the variance-covariance matrix $\mathbf{C}_y(\mathbf{x}_i^{obs})$. If the covariance matrix $\mathbf{C}_y(\mathbf{x}_i^{obs})$ is known the learning back-propagation algorithm is no more than a stochastic gradient algorithm which minimizes the weighted Least Mean Square expression.

When the output noise does not depend on $\mathbf{x}$ and is constant ($\mathbf{C}_y(x_i^{obs}) = \sigma^2 I$), σ^2 is estimated at the end of the learning phase by R_{emp}. For an output noise depending on the inputs, this leads to errors when estimating $\mathbf{W}$, the same strengh are imposed to each output range without paying any particular attention to its variability. The learning algorithm tends to overfit the regions with high variability at the expense of regions presenting small variability.

3 Output noise determination

The main goal of this section is to introduce the Bayesian formalism and to address the problem of determining the conditional probability density function of the output noise $p(\mathbf{y}/\mathbf{x})$. This paper will be restricted to the case of Gaussian noises and to MLP regressors. Other neural models dedicated to more general probability density function are available, the most popular are the multi-expert models proposed by Jordan [13], Weigend [28] and the mixture density networks proposed by Bishop [3] which allow to model more general conditional probability density functions. Their major advantages appear when dealing with multi-valued regression funtions. As these models can be used to resolve ambiguities, they are good candidate for solving inverse problems [26] [16] [20]. However, the paper focus on the direct problem where only single-valued function are considered and the more difficult problem of multi valued regression will not be addressed here.

3.1 The general Bayesian framework

In the following, we assume that there is no uncertainties on the input ($\boldsymbol{\epsilon} = 0$). Then, for each input$\mathbf{x}_i$, we have:

$\mathbf{x}_i^{obs} = \mathbf{x}_i^{true}$. On the other hand, $\mathbf{y}$ is spoilt by an additive noise such as:

$$\mathbf{y}_i^{obs} = \mathbf{y}_i^{true} + \boldsymbol{\delta}_i \tag{16}$$

The true field of $\mathbf{y}$, $\mathbf{y}_i^{true}$, represents the mean of the conditional random variable $\mathbf{Y}/\mathbf{x}_i^{obs}$ and our goal is to estimate the theoretical relation between $\mathbf{x}_i^{true}$ and $\mathbf{y}_i^{true}$.
The bayesian paradigm [14] [24] aims to estimate $\mathbf{W}$ by maximizing the probability corresponding to most likely parameters conditionally to the observations. Using the Bayes criterion, one have:

$$p(\mathbf{W}/D) = \frac{p(D/\mathbf{W})p(\mathbf{W})}{p(D)} \tag{17}$$

Under this assumption, maximizing Eq. (17) is equivalent to minimizing:

$$-2\ln\Big(p(\mathbf{W}/D)\Big) = -2\ln\Big(p(D/\mathbf{W})\Big) - 2\ln\Big(p(\mathbf{W})\Big) + \text{ constant} \tag{18}$$

The probability of data $p(D)$ does not depend on $\mathbf{W}$ and can be removed from the minimization process. In the case of independent observations, we can write:

$$\ln\Big(p(D/\mathbf{W})\Big) = \sum_{i=1}^{N^{obs}} \ln\Big(p\big((\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs})/\mathbf{W}\big)\Big) \tag{19}$$

which yields:

$$\ln\Big(p(D/\mathbf{W})\Big) = \sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs},\mathbf{W})\Big) + \sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{x}_i^{obs}/\mathbf{W})\Big) \quad (20)$$

The first term of the right hand side of Eq. (20) is the probability that $\mathbf{y}_i^{obs}$ comes from $\mathbf{x}_i^{obs}$ when generated by the neural model $\mathbf{F}(\mathbf{W},.)$. Assuming an acceptable model we have:

$$\mathbf{E}(\mathbf{Y}/\mathbf{x}_i^{obs}) \approx \mathbf{F}(\mathbf{W},\mathbf{x}_i^{obs}) \quad (21)$$

Since $\mathbf{x}_i^{obs}$ are not spoiled by an additive noise, the second term in the right hand side of equation (20) does not depend on $\mathbf{W}$. The assumption of a Gaussian output noise with zero mean and covariance matrix $\mathbf{C}_y(\mathbf{x})$ leads to the simplified expression of $\ln\Big(p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs},\mathbf{W})\Big)$:

$$-2\ln\Big(p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs},\mathbf{W})\Big) = \Big(\mathbf{y}_i-\mathbf{F}(\mathbf{W},\mathbf{x}_i)\Big)^T \mathbf{C}_y{}^{-1}(\mathbf{x}_i)\Big(\mathbf{y}_i-\mathbf{F}(\mathbf{W},\mathbf{x}_i)\Big) - \ln\Big(\det\big(\mathbf{C}_y{}^{-1}(\mathbf{x}_i)\big)\Big) + q\ln 2\pi \quad (22)$$

Using the previous equations, one can define the empirical risk:

$$R_{emp}(\mathbf{W}) = -2\ln\Big(p(\mathbf{W}/D)\Big) = \sum_{i=1}^{N^{obs}} R_i - 2\ln p(\mathbf{W}) + \text{ constant} \quad (23)$$

with

$$R_i(\mathbf{W}) = \Big(\mathbf{y}_i^{obs}-\mathbf{F}(\mathbf{W},\mathbf{x}_i^{obs})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}_i^{obs})\Big(\mathbf{y}_i^{obs}-\mathbf{F}(\mathbf{W},\mathbf{x}_i^{obs})\Big) - \ln\Big(\det\big(\mathbf{C}_y^{-1}(\mathbf{x}_i^{obs})\big)\Big) \quad (24)$$

The additional term $\ln p(\mathbf{W})$ which appears in the expression of $R_{emp}(\mathbf{W})$ (23) is a regularization term. It corresponds to an additional constraint on the weight distribution. If one assumes a Gaussian prior information on $\mathbf{W}$, this term corresponds to a weight decay, for uniformly distributed $\mathbf{W}$ it can be removed. The first term in the right-hand side of R_i is the quadratic error defined in equation (14). Equation (24) appears as a generalization of equation (14), the second term of the expression being introduced to take into account the output noise. According to the case, this term can be known or not: If not, it has to be considered and estimated during the minimization process.

3.2 Determination of the covariance-matrix

3.2.1 General formulation

The resolution of the general regression problem is equivalent to the estimation of the mean vectors $\mathbf{y}$ and the covariance matrix $\mathbf{C}_y(\mathbf{x})$. It leads to the determination of an adequate MLP architecture together with the configuration of weights which minimizes equation (24) under the hypothesis of unknown covariance matrix.

Under the same hypothesis (a Gaussian output noise) a general framework for the determination of the covariance-matrix using MLP has been proposed by Williams [29]. The learning algorithm consists in estimating the mean value $E(\mathbf{Y}/\mathbf{x}_i^{obs})$ on one hand and each coefficient of $\mathbf{C}_y^{-1}(x_i^{obs})$ on the other hand which gives the parameters of the Gaussian distribution $p(\mathbf{Y}/\mathbf{x})$. This type of algorithm was independently studied in [17] and [3], the only difference being that these authors give the results for independent variables (zero covariance coefficients). To solve this problem, Williams estimates the coefficients of the Cholesky decomposition of the variance-covariance matrix:

$\mathbf{C}_y^{-1}(\mathbf{x}) = \mathbf{A}^T(\mathbf{x})\mathbf{A}(\mathbf{x})$ where $\mathbf{A}(\mathbf{x}) = [a_{ij}(\mathbf{x}); i < j)]$ is an upper triangular matrix with strictly positive coefficients on the diagonal. Using these notations, equation (24) becomes:

$$\begin{aligned} R_i(\mathbf{W}) &= \left(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{obs})\right)^T \mathbf{A}^T(\mathbf{x}_i^{obs})\mathbf{A}(\mathbf{x}_i^{obs})\left(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{obs})\right) \\ &\quad - 2\sum_{j=1}^{q} \ln\left(a_{jj}(\mathbf{x}_i^{obs})\right) \end{aligned} \tag{25}$$

The architecture is designed with respect to the complexity of the problem : the size of the input layer being the dimension of $\mathbf{x}$.The adequate architecture of the MLP has an output layer made of three different sets of neurons.
In the output layer, three different sets of neurons (M, D, C) represent the different values to be estimated:

- M estimate the q distinct mean values $\mathbf{E}(\mathbf{Y}/\mathbf{x}_i^{obs})$ using linear neurons.
- D stand for the q diagonal coefficients $\mathbf{a}_{ii}(\mathbf{x})$ of $\mathbf{A}$; as these q values are positive the neurons of D use exponential transfert function in order to ensure positive values.
- C estimates the $\frac{q(q-1)}{2}$ correlation coefficients $(a_{kj}(\mathbf{x});\ \ k < j)$ using linear neurons.

The output layer has thus $\left(2q + \frac{q(q-1)}{2}\right)$ neurons whose outputs are distributed in three different sets which are denoted by:

$$O^M = \{o_j^M(\mathbf{W}, \mathbf{x}^{obs}) \ , \qquad j = 1 \cdots q\}$$
$$O^D = \{\, o_j^D(\mathbf{W}, \mathbf{x}^{obs}) \ , \qquad j = 1 \cdots q\}$$
$$O^C = \{\, o_{kj}^C(\mathbf{W}, \mathbf{x}^{obs}) \ , \ \ k < j,\ j = 1 \cdots q\}$$

For the inputs of the neurons of the output layer we use similar notations, replacing o by s and O by S, for the three different sets $S = \{S^M,\ S^D,\ S^C\}$. As neurons of type M and

C have linear transfer function and neurons of type D have exponential transfer function we have:

- $o_k^M = s_k^M$
- $o_{kj}^C = s_{kj}^C$
- $o_k^D = e^{s_k^D}$

To simplify the notations we denote by $\mathbf{x}^{obs}$ the particular observation $\mathbf{x}_i^{obs}$ and R the related cost function, omiting the index i.

For $\mathbf{x}^{obs}$ we introduce the related vector of error:
$\mathbf{\Delta} = \left[\delta_j^M\right] = \left(\mathbf{y}^{obs} - \mathbf{O}^M\right), \quad j = 1 \cdots q.$

If we denote by $\widetilde{\mathbf{A}} = \left[\widetilde{a}_{kj}\right]$ the estimated matrix (whose elements are the output of C and D) equation (25) becomes:

$$R(\mathbf{W}) = \mathbf{\Delta}^T \widetilde{\mathbf{A}}^T(\mathbf{x}^{obs}) \widetilde{\mathbf{A}}(\mathbf{x}^{obs}) \mathbf{\Delta} - 2 \sum_{k \in D} s_k^D \tag{26}$$

The learning algorithm which minimizes (26) proceeds by back-propagation. For each output neuron, the back-propagation algorithm requires the computation of the initial errors: $\frac{\partial R}{\partial \mathbf{S}}$

$$\frac{\partial R}{\partial S^M} = -2\ \widetilde{\mathbf{A}}^T \widetilde{\mathbf{A}} \Delta^M \tag{27}$$

$$\frac{\partial R}{\partial s_{kj}^C} = 2\delta_j^M \Big(\sum_{l=1}^{k-1} o_{ik}^C \delta_l^M + o_k^D \delta_k^M \Big) \tag{28}$$

$$\frac{\partial R}{\partial s_k^D} = 2 o_k^D \delta_k^M \Big(\sum_{j=1}^{k-1} o_{kj}^C \delta_j^M + o_k^D \delta_k^M \Big) - 2 \tag{29}$$

3.2.2 The algorithm

In the following we describe and comment the algorithm used in order to minimize equation (26). The algorithm proceeds in three phases, the phase 1 and 2 being run iteratively. Then, phase 3 is completed.

- **Phase 1** : the aim of this phase is to provide a good estimate of the mean value $E(\mathbf{Y}/\mathbf{x}^{obs})$ before going to phase 2 and 3. So this phase deals with neurons of type M. During this phase, it is assumed that $\widetilde{\mathbf{A}}(\mathbf{x}^{obs})$ does not depend on the weights $\mathbf{W}$. According to that its coefficients are left constant. Only the first term of equation (25) is minimized. It corresponds to the quadratic error (14). The initial errors on the q outputs of type M are initialized with equation (27) before running the back-propagation procedure. The first time phase 1 is run it is assumed that $\widetilde{a_{kj}}(\mathbf{x}^{obs}) = 0$ for $k < j$ and that $\widetilde{a_{jj}}(\mathbf{x}^{obs}) = 1$, R_{emp} is just the classical square error. In further iterations, the coefficients of $\widetilde{\mathbf{A}}(\mathbf{x}^{obs})$ are fixed at values computed from phase 2.

- **Phase 2** : In this phase, the outputs of the q neurons of type M are frozen assuming that they do not depend on $\mathbf{W}$. They are frozen at the values $\mathbf{F}_j(\mathbf{W}^*, \mathbf{x}^{obs})$, where $\mathbf{W}^*$ are the weights computed during phase 1. The remaining $\frac{q(q+1)}{2}$ outputs of type C and D compute the corresponding coefficient of $\widetilde{\mathbf{A}}(\mathbf{x}^{obs})$. Their initial errors are initialized with equation (28, 29) before running back-propagation. Therefore, this phase aims to improve the accuracy of the coefficient of matrix $\widetilde{\mathbf{A}}$.

- **Phase 3** : During phase 3, we minimize the error function (26). All network outputs are variable and we initialize the back-propagation using equation (27) for output neurons of type M, Equation (28) for output neurons of type C and equation (29) for neurons of type D.

Learning gives an approximation of $\mathbf{E}(\mathbf{Y}/\mathbf{x}^{obs})$ with an accuracy which can be analysed as in Appendix.
In the particular case where the noise components are independent, the matrix $\mathbf{C}_y^{-1}$ is diagonal and the algorithm for noise determination is more simple. The matrix $\mathbf{A}$ is diagonal with diagonal coefficients $a_{jj} = \frac{1}{\sigma_j}$, where σ_j is the standard deviation of the j^th component of the output noise. In this case the output layer has no neurons of type C. The initialization of the backpropagation algorithm becomes:

$$\begin{aligned} \text{for } k \in \text{ M}: \quad \frac{\partial R}{\partial s_k^M} &= -2\frac{\delta_k^M}{\left(o_k^D\right)^2} \\ \text{for } k \in \text{ D}: \quad \frac{\partial R_i}{\partial s_k^D} &= 2\left(o_k^D\right)^2\left(\delta_k^M\right)^2 - 2 \end{aligned}$$

This case was handled by [17]

Different architecture can be used for the required estimation which allows to introduce some knowledge about the particular problem. For example, when the output noise δ depends on both the input $\mathbf{x}$ and the output $\mathbf{y}$, it is more convenient to use two separate MLPs. The first MLP estimates the mean and the second the covariance matrix. An exemple of the method is presented in section (5). Other architectures have been proposed. For example, [17] propose a single network with three types of weights:

- weights accounting for all outputs of the network (both $\mathbf{F}_j(\mathbf{W}, \mathbf{x}_i^{obs})$ and $\sigma_j(\mathbf{x}_i^{obs})$)

- weights accounting only for outputs $\mathbf{F}_j(\mathbf{W}, \mathbf{x}_i^{obs})$

- weights accounting only for $\sigma_j(\mathbf{x}_i^{obs})$.

The first type of weights enable to correlate $\mathbf{F}_j(\mathbf{W}, \mathbf{x}_i^{obs})$ and $\sigma_j(\mathbf{x}_i^{obs})$, since, in practical, such a correlation exists. On the other hand, the two other types enables to give a freedom degree to each output cells group.

3.2.3 Validation procedures

In this section we propose some qualitative approach for validating the first and second order moments given by the networks at the end of the learning processes. In the following we assume that the number of observations is large enough and is representative of the local dispersion of the data, without such assumption, estimating the covariance matrix is not statisticaly significant.

At the end of the learning procedure, an important task is to validate the accuracy of the results given by the MLP. If the size of the learning data set is large enough, the use of multivariate statistical tests can give some relevant informations about it. In particular, they allow to check the accuracy of the MLP, that is its ability to predict the conditional mean $\mathbf{E}(\mathbf{y}/\mathbf{x})$ and the covariance matrix $\mathbf{C}_y(\mathbf{x})$.
This is done by:

- partitioning in small bins the inputs of the learning data set $\{\mathbf{x}_i^{obs}, i = 1, \ldots, N^{obs}\}$ which corresponds to the projection of the learning set onto the R^p space. The center of bin B is denoted $\mathbf{x}_B$.
- estimating the empirical mean $\overline{\mathbf{y}}_B$ and the empirical covariance matrix $\overline{\mathbf{\Sigma}}_B$ using the sample $\{\mathbf{y}_i^{obs} \backslash \mathbf{x}_i^{obs} \in B\}$ of size N_B.
- computing the MLP outputs at the center of the bin: $\boldsymbol{\mu}_B = \mathbf{F}(\mathbf{W}, \mathbf{x}_B)$.
- if the covariance matrix has been estimated by the MLP, using its outputs for determining the estimated covariance matrix $\mathbf{\Sigma}_B$ at the center of the bin $\mathbf{x}_B$.
- performing test hypothesis.

Clearly the bin size is an important factor, according to the results concerning Kernel regression technics [9]. One has to choose a good compromise between the bandwith of the bin and the number of observations lying into the bin. In the following we assume that each bin has enought data in order to estimate the mean and the covariance matrix: estimating second order statistics requires more data than determining the mean. We use multivariate statistical tests [1] to validate the network results.

The first problem is thus to test the hypothesis that $\mathbf{F}(\mathbf{W}, \mathbf{x}_B)$ represents an estimate of $\mathbf{E}(\mathbf{y}/\mathbf{x}_B)$ at a confidence level of α. This is done by performing a *Hotelling's* T^2 test with a significance level α. This test consists in computing: $T^2 = N^B \left[\overline{\mathbf{y}}_B - \boldsymbol{\mu}_B\right]^T \overline{\mathbf{\Sigma}}_B^{-1} \left[\overline{\mathbf{y}}_B - \boldsymbol{\mu}_B\right]$
The critical value of T^2 can be determined by the relation: $F = \frac{N_B - q}{q(N_B - 1)} T^2$ which is a Fisher statistic with q and $(N_B - q)$ degrees of freedom and allows to test the null hypothesis that $\boldsymbol{\mu}_B$ is equal to the mean of the gaussian distribution $\left\{\mathbf{y}_i^{obs} \backslash \mathbf{x}_i^{obs} \in B\right\}$. Such a test gives an adequate way to appreciate the validy of the results provided that the size of the learning data set is large enough. For MLP's with a single output (scalar regression), this test is no more than the classical *Student* t-test.
When the covariance matrice $\mathbf{C}_y^{-1}(\mathbf{x})$ has been estimated by MLPs it is possible to use classical multivariate tests as *Hotelling's* T^2 [1] which tests the hypothesis that the computed covariance matrix is equal to the covariance matrix of the distribution. However, this test presents some numerical drawbacks. A good compromise is to decorrelate the

sample related to each bin B with respect to its empirical covariance matrix defining the sample: $\left\{\mathbf{z}_i = \overline{\Sigma_B}^{1/2}(\mathbf{y}_i^{obs} - \overline{\mathbf{y}}_B) \backslash \mathbf{y}_i^{obs} \in B\right\}$. and to use the classical univariate χ^2 tests. As $\mathbf{z}$ is assumed to be Gaussian with unit covariance matrix, one can perform the univariate T^2 test on each sample made with one component of $\mathbf{z}$. In this case one test the hypothesis that the sample has unit variance.

4 The generalized regression problem

4.1 The general Bayesian framework

An important generalization of the regression problem is to take into account uncertainties on both input $\mathbf{x}$ and output $\mathbf{y}$. As in the previous section, the Bayesian formalism provides a methodology to achieve the estimation of the maximum likelihood using assumptions on the uncertainties distribution laws. A first formalisation of the problem was proposed by Weigend and Zimmermann [27] In this section, it is assumed that both noises have known densities:

$$\mathbf{x}_i^{obs} = \mathbf{x}_i^{true} + \boldsymbol{\epsilon}_i \tag{30}$$
$$\mathbf{y}_i^{obs} = \mathbf{y}_i^{true} + \boldsymbol{\delta}_i \tag{31}$$

where $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\delta}_i$ are zero mean random variables with known covariance matrix $\mathbf{C}_x$ and $\mathbf{C}_y(\mathbf{x})$.

The problem is to estimate the theoretical relation $\mathbf{G} : \mathbf{x}^{true} \longmapsto \mathbf{y}^{true}$ attainable by a set $D = \left\{(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs}), i = 1 \ldots N^{obs}\right\}$, using the available knowledge about noises. This relation is modeled by a MLP denoted$\mathbf{F}(\mathbf{W}, .)$, the parameters $\mathbf{W}$ being unknown. In fact, to solve this problem, one have to estimate the set $\mathbf{W}$ together with the values $\mathbf{x}^{true}$. Estimating the $\mathbf{y}^{true}$ is straightforward since, following the results previously given, $\mathbf{y}^{true}$ is obtained by $\mathbf{F}(\mathbf{W}, \mathbf{x}^{true})$. If a "good" estimate $\mathbf{x}_i^{est}$ of $\mathbf{x}_i^{true}$ can be obtained, the function $\mathbf{F}(\mathbf{W}, .)$ is derived by learning the set of observations: $D' = \left\{(\mathbf{x}_i^{est}, \mathbf{y}_i^{obs}), i = 1 \ldots N^{obs}\right\}$.This enables us to have a good estimate of the true observations $\mathbf{y}^{true}$. The problem is therefore to have both good estimates for $\mathbf{x}_i^{est}$ and good values for the $\mathbf{W}$ parameters.

The Bayesian approach consists, in this case, in maximizing:

$$p\big(\mathbf{W}, \{\mathbf{x}_i^{est}, i = 1 \ldots N^{obs}\}/D\big) = \frac{p\big(D/\mathbf{W}, I^{est}\big)p\big(\mathbf{W}, I^{est}\big)}{p(D)} \tag{32}$$

where we have to estimate $\mathbf{W}$ and $I^{est} = \{\mathbf{x}_i^{est}, i = 1 \ldots N^{obs}\}$.

Eq. (32) can be written:

$$\begin{aligned}\ln\Big(p(\mathbf{W}, I^{est}/D)\Big) &= \sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs}, \mathbf{W}, I^{est})\Big) \\ &+ \sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{x}_i^{obs}/\mathbf{W}, I^{est})\Big) \\ &+ \sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{W}, I^{est})\Big) - \ln\big(p(D)\big) \end{aligned} \tag{33}$$

This can be simplified, noticing that:

$$\begin{aligned} p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs}, \mathbf{W}, \{\mathbf{x}_i^{est}, i = 1 \ldots N^{obs}\}) &= p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs}, \mathbf{W}, \mathbf{x}_i^{est}) \\ p(\mathbf{x}_i^{obs}/\mathbf{W}, \{\mathbf{x}_i^{est}, i = 1 \ldots N^{obs}\}) &= p(\mathbf{x}_i^{obs}/\mathbf{x}_i^{est}) \\ p(\mathbf{W}, I^{est}) &= p(\mathbf{W}/I^{est}).p(I^{est}) \end{aligned}$$

We assume that $\ln\Big(p(\mathbf{W}/I^{est})\Big)$ does not depends on $\mathbf{x}^{est}$ and corresponds to the regularization factor discussed in the previous section. As I^{est} is assumed to be uniformly distibuted and $\ln\big(p(D)\big)$ is also a constant, maximizing Eq. (33) is equivalent to minimize:

$$\begin{aligned}-2\ln\Big(p(\mathbf{W}, I^{est}/D)\Big) &= -2\sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{y}_i^{obs}/\mathbf{x}_i^{obs}, \mathbf{W}, \mathbf{x}_i^{est})\Big) \\ &-2\sum_{i=1}^{N^{obs}} \ln\Big(p(\mathbf{x}_i^{obs}/\mathbf{x}_i^{est})\Big) + \text{constant} \end{aligned} \tag{34}$$

The cost function takes the form of:

$$R_{emp}(\mathbf{W}, \mathbf{x}^{est}) = \sum_{i=1}^{N^{obs}} R_i(\mathbf{W}, \mathbf{x}^{est}) - 2\ln\big(p(\mathbf{W})\big) + \text{constant} \tag{35}$$

with

$$\begin{aligned} R_i(\mathbf{W}, \mathbf{x}_i^{est}) &= \Big(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}_i^{est})\Big(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est})\Big) \\ &+ \ln\Big(\ det\ \big(\mathbf{C}_y(\mathbf{x}_i^{est})\big)\ \Big) \\ &+ \Big(\mathbf{x}_i^{obs} - \mathbf{x}_i^{est}\Big)^T \mathbf{C}_x^{-1}\Big(\mathbf{x}_i^{obs} - \mathbf{x}_i^{est}\Big) \end{aligned} \tag{36}$$

As for the algorithm presented in section (3.2.2), this cost function is iteratively minimized by estimating successively the weights, computing the true observation $\mathbf{y}_i^{true} = \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est})$ and improving subsequently $\mathbf{x}_i^{est}$.

The optimization scheme can be done by a gradient back-propagationalgorithm using the computation of the partial derivatives with respect to $\mathbf{W}$ and $\mathbf{x}_i^{est}$ of the cost function. The third term of Eq. (36) does not depend on $\mathbf{W}$, the computation of the partial derivatives with respect to the weights corresponds to formula presented in section 3.2.1. The partial derivatives of R_i with respect to the model parameters $\mathbf{x}^{est}$ are computed as follows:

$$\begin{aligned}\frac{\partial R}{\partial \mathbf{x}}\left(\mathbf{x}_i^{est}, \mathbf{W}\right) &= -2\,\mathbf{J}^T \mathbf{C}_y^{-1}\left(\mathbf{x}^{est}\right)\left(\mathbf{y}_i^{obs} - \mathbf{F}\left(\mathbf{W}, \mathbf{x}_i^{est}\right)\right) \\ &\quad -2\,\mathbf{C}_x^{-1}\left(\mathbf{x}_i^{obs}\mathbf{x}_i^{est}\right) \end{aligned} \tag{37}$$

where $\mathbf{J} = \left[J_{kj} = \frac{\partial \mathbf{F}_k}{\partial x_j}\right]$ is the Jacobian matrix of $\mathbf{F}$ with respect to the model parameters $\mathbf{x}^{est}$.This formula is an approximation of the true value of $\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\left(\mathbf{x}_i^{est}, \mathbf{W}\right)$ where the partial derivatives of

$$\frac{\partial \mathbf{C}_y^{-1}}{\partial \mathbf{x}}\left(\mathbf{x}_i^{est}, \mathbf{W}\right) \quad \text{and} \quad \frac{\partial \ln\left(\det\left(\mathbf{C}_y\left(\mathbf{x}_i^{est}\right)\right)\right.}{\partial \mathbf{x}}\left(\mathbf{x}_i^{est}, \mathbf{W}\right)$$

are neglected. At this time, it is assumed that the coefficients of the covariance matrix $\mathbf{C}_y\left(\mathbf{x}_i^{est}\right)$ are constant. The Jacobian $\mathbf{J}$ is computed by back-propagation to input [12].

Accounting for the input and output noises leads to a revised version of the Gradient Back-Propagation Algorithm (GBP) which proceeds in two steps: the first one estimates the weights and the second one estimates the values of $\mathbf{x}^{true}$.

4.2 The Generalized Back-Propagation (GBP)

In the following we describe the GBP algorithm, which has two different phases. The first phase aims to update the weights of the neural model. The second phase aims to update I^{est} and to clean the learning set, giving rise to a new learning set $D_{current}$. The two phases are run iteratively, allowing to minimize $R_{emp}(\mathbf{W}, \mathbf{x}^{est})$. As the observations have to be modified with care, the learning gain used for updating the inputs has to be significantly smaller than the one used when learning the weights.

- **Initialization phase** : This phase gives the first I^{est} and the first learning set $D_{current}$:
  For every $i = 1, \ldots, N^{obs}$, $\mathbf{x}_i^{est}$ is initialized to $\mathbf{x}_i^{obs}$ and
  $D_{current} = \left\{(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs}), i = 1 \cdots N^{obs}\right\}$

- **Phase 1** : This phase computes the new weights $\mathbf{W}^*$, an estimate of the mean $\mathbf{E}(\mathbf{Y}/\mathbf{x})$ and if necessary an estimate of the covariance matrix giving the new $\mathbf{C}_y^{-1}(\mathbf{x})$ (by applying the algorithm of section 3.2.2). It uses the learning set $D_{current}$ available at this iteration.

- **Phase 2** : This phase computes the new I^{est} and $D_{current}$.
  The parameters $\mathbf{W}$ are frozen and the $\mathbf{x}_i^{est}$ are adapted by minimizing R_{emp} (35) with

respect to I^{est}. The minimization is initialized using the current I^{est}, the minimum is reached using a stochastic gradient :

$$\mathbf{x}_i^{est} = \mathbf{x}_i^{est} - \varepsilon \frac{\partial R}{\partial \mathbf{x}}(\mathbf{x}_i^{est}, \mathbf{W}) \tag{38}$$

where the partial derivatives are estimated by formula (37). At the end of the minimization process, the outliers can be removed from the current learning set $D_{current}$ by applying the rejection test presented in subsection 4.3, which gives rise to the new I^{est} and $D_{current} = \left\{(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs}), i \in J, J \subseteq [1 \cdots N^{obs}]\right\}$

Phases 1, 2 are iterated

4.3 Rejection test

Under the Gaussian distribution hypothesis, the general Bayesian formulation presented in section 4 allows to test the accuracy of the neural computation and to propose some reject procedure in order to detect outliers. The test of rejection which checks the consistency of the learning data set, is used at the end of phase 2 and provides the new $D_{current}$ used in the next iteration of phase 1. The expression of the conditional joint distribution, $p(\mathbf{x}^{obs}, \mathbf{y}^{obs}/\mathbf{W}, \mathbf{x}^{est})$ with respect to the input and output Gaussian distibutions gives:

$$\begin{aligned} p(\mathbf{x}^{obs}, \mathbf{y}^{obs}/\mathbf{W}, \mathbf{x}^{est}) &= p(\mathbf{y}^{obs}/\mathbf{x}^{obs}, \mathbf{W}, \mathbf{x}^{est}) * p(\mathbf{x}^{obs}/\mathbf{x}^{est}) \\ &= K_x K_y(\mathbf{x}) e^{-\frac{1}{2}Q} \end{aligned} \tag{39}$$

where K_x and $K_y(\mathbf{x})$ are normalization constants and Q is defined by:

$$\begin{aligned} Q &= \left(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est})\right)^T \mathbf{C}_y^{-1}(\mathbf{x}_i^{est}) \left(\mathbf{y}_i^{obs} - \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est})\right) \\ &\quad + \left(\mathbf{x}_i^{obs} - \mathbf{x}_i^{est}\right)^T \mathbf{C}_x^{-1} \left(\mathbf{x}_i^{obs} - \mathbf{x}_i^{est}\right) \end{aligned} \tag{40}$$

A particular observation of the learning set $(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs})$ is assumed to be a realization of $(p+q)$ Gaussian random variables with mean $(\mathbf{x}_i^{est}, \mathbf{F}(\mathbf{W}, \mathbf{x}_i^{est}))$ and covariance matrix $\left[\mathbf{C}_\mathbf{x}^{-1}, \mathbf{C}_\mathbf{y}^{-1}(\mathbf{x}_i^{est}, \mathbf{W})\right]$. Under this assumption, the variable Q is a χ^2 random variable with $(p+q)$ degrees of freedom. The use of the χ^2 test allows to reject some observations $(\mathbf{x}_i^{obs}, \mathbf{y}_i^{obs})$ as outliers with a desired confidence level α. This procedure can be used during the learning process to clean the data, improving in the same time the learning accuracy. So, the joint utilization of the GBP algorithm and the preceeding rejection procedure gives both the required regression function and a "clean" learning data set [27].

4.4 An experimental study of GBP

The interest of the GBP is proved by giving extended comparisons with the Standard Back-Propagation Algorithm (SBP). We used here one dimensional simulated data set

which enables to vary all the parameters of the experiments and to show the effect of the different phases when using GBP (phase 1, phase 2 and rejection on the learning process). Moreover, we discuss some of the interesting characteristics of GBP : robustness, regression with small sample, detection of outliers.
In the following, let us consider the following example where the function **G** is of the form:

$$\begin{aligned} [\,0,\pi\,] &\rightarrow [\,-1,1\,] \\ x &\mapsto y = \mathbf{G}(\mathbf{x}) = 0.5\Big(\cos(\mathbf{x}) - \cos(4\mathbf{x})\Big) \end{aligned} \tag{41}$$

We determine four distinct sets of independant observations uniformly sampling the interval $[\,0,\pi\,]$, adding or not a normally distributed noise (A) to each input x and considering two distinct output noises. The input noise is derived from the normal distribution with zero mean and constant standard deviation 0.1. The first output noise (B) is also normally distributed with zero mean and constant standard deviation 0.2, the second one (C) is a normaly zero mean input dependant noise with non-constant variance: $\sigma_y^2(x) = 0.01 + 0.25\,[1 - \sin(2.5x)]^2$. Mixing the different input and output distributions allows to define four distinct observation sets:

1. D1 : no noise on x and noise B on y
2. D2 : noise A on x and noise B on y
3. D3 : no noise on x and noise C on y
4. D4 : noise A on x and noise C on y

We add to all these four data sets 3 outliers taken at random in order to investigate the behavior of GBP when facing to erroneous data. These points whose y-coordinate ($y_1 = -2,\ y_2 = 1.5,\ y_3 = -1.5$) are choosen in order to keep the 3 choosen pairs far from the different learning data sets. They can be considered as outliers and have to be removed during the rejection procedure (see 4.3 which presents some mechanism for the detection of these outliers).
The four different data distributions are presented in Figure (1).
All the experiments presented below use the same MLP with 2 hidden layers of 3 neurons each and a linear output. All the computations use stochastic second order gradient in order to minimize the required cost function. For each experiment, learning has been run until convergence. All the comparisons are made in the same numerical conditions. A regularization term depending on a parameter γ and corresponding to a Gaussian prior on the weights, is added to the neural cost functions involved in the comparison :$\gamma \sum_{i,j} w_{ij}^2$. This hyper-parameter γ, taken to $\gamma = 0.01$, has been optimized to allow the best performances for each of the methods.
In the following, in order to interpret the different results, we compute three distinct indicators using the test set. For comparison purpose we introduce the three following indices defined for a data set $D = \big\{(x_i, y_i), i = 1 \ldots N\big\}$:

1. The classical RMS giving the mean value of the discrepancy between the neural value $y^{NN}(x)$ and the data y:

$$\text{RMS}_{(D)} = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - y^{NN}(x_i)\right)^2}{N}}$$

2. The mean discrepancy between the theoretical function $\mathbf{G}$ (41), which has to be retrieved, and the estimated neural function $y^{NN}(x)$. To achieve this,we compute the RMS between the two functions:

$$\text{Real-RMS}_{(D)} = \sqrt{\frac{\sum_{i=1}^{N} \left(\mathbf{G}(x_i) - y^{NN}(x_i)\right)^2}{N}}$$

3. The empirical standard deviation of the final set of x^{est} given by GBP :

$$\sigma(x^{est} - x) = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i^{est} - x_i\right)^2}{N-1}}$$

The learning data set changes with the different experiments, but we compute the generalization performances RMS from an independent test set(T_1) of $N = 300$ samples generated according to the distribution of the related learning set. The Real-RMS is computed on a test set (T_2) made of $N = 300$ x-values regularely spaced on $[\,0, \pi\,]$. The third indicator is computed from the related learning set.

Figure (2) presents the function provided by SBP and GBP when learning from 200 samples of D3. In this experiment we assume that $\mathbf{C}_{\mathbf{y}}^{-1}(x)$ is known, possibly estimated by applying the algorithm of section (3.2.2). As there is no noise on x and the output noise is input dependent, only the initialization phase of GBP is run. Clearly, the knowledge of the output variance used by GBP improve the retrieval of function $\mathbf{G}$. The Real-RMS error is 0.163 for SBP and 0.142 for GBP, the use of the output variance during the learning phase allows a better restitution of the theoretical function $\mathbf{G}$.

The second experiment shows the efficiency of GBP when dealing with known noisy input data: the knowledge of the input noise always improves the estimation of $\mathbf{G}$. In these experiments, we use D2 which has been simulated with constant input and output noises and we determine four learning sets of different sizes (25, 50, 100, 200 patterns) uniformly distributed in the range $[\,0, \pi\,]$. The function $\mathbf{G}$ is estimated using SBP and GBP for these four learning sets of increasing size. Table (1) gives the RMS and the Real-RMS of the second experiment for both GBP and SBP with respect to the size of the learning set. At this stage we do not make use of the rejection test.

The RMS value represents an estimate of the standard deviation of the noise (0.2) added to the theoretical relationship $\mathbf{G}$. The performances are strongly related to ratio of outliers in the learning data set, as there are still 3 outliers the performances decrease

withe the size of the learning data set. For a learning data set of size 25 the percentage of outliers in the learning data set exeeds 10 percent and both algorithms SBP and GBP are heavily perturbated. It is clear that the presence, in the learning set, of more than 5 percent of outliers heavily disturbs the behavior of the learning algorithm. Nevertheless GBP always gives better results than SBP and the estimates of the output noise are never far from the 0.2 theoretical value. As shown in the second raw of table (1) ($\mathrm{RMS}_{(D)}$), the GBP algorithm, which uses the knowledge of the different noises, clearly improves the retrieval of the theoretical function when dealing with a large set ($N^{obs} > 50$).
The last row of table (1) gives for GBP an estimate of $\sigma(x^{est} - x^{obs})$ at the end of the learning stage. It can be seen that a part of the noise introduced on the input data is recovered, and that the difference between the theoretical noise of 0.1 and the estimated noise slowly decreases with the size of the learning set. We determine the outliers using the

rejection procedure presented in section 4.3, at a confidence level of 95%. Table (2) gives the number of detected outliers with respect to the sample size. It can be mentionned that the three actual ouliers are always detected. According to this test, we apply the selection procedure and reject all the data detected as outliers from the learning data set. Training is done again using the remaining data. The new performances are given in table (2) and can be compared with those of table (1).

In the last experiment, we illustrate the behavior of GBP in case of an unknown, non constant output noise. To achieve this, we compare the behavior of SBP and GBP on a learning set of 200 samples extracted from D4. As the the accuracy on the measurements is supposed to be unknown, the covariance matrix has been determined, during phase 1, using the algorithm of section 4.1. The estimated variances are used during phase 2 together with the knowledge of the input noise.The sucessive iterations allows the refinement of the regression function. Table 3 gives the performances for SBP and compare with those obtained at the end of the first iteration of phase 1 and at the end of the algorithm for GBP. Figure(5) gives the two neural functions given by SBP and GBP, clearly, the estimation of the output noise improves the retrieval of the real function ($\mathbf{G}$).This result will be confirmed later on on the real-world application presented in section (5). The use of the uncertainty on the observations acts as a smoothing factor and allows the estimated function to be less corrupted by outliers.As a consequence, the performances in interpolation but also in extrapolation are better.

5 Geophysical applications

In the following, we apply the neural network methodology presented above to actual problems to show the efficiency of the approach. We choose two different problems which come from operational Oceanography. In both of them, the problem is to approximate a highly non linear relationship between noisy sets of observations.

The first illustration is taken from the NSCAT scatterrometer transfer function estimation. It deals with a large amount of remote sensing data, offering a good opportunity

to evaluate the maximum of accuracy which can be reached by neural regression and covariance estimations. The second illustration is taken from the problem of ocean color and deals with a small set of very noisy in situ data. It shows the efficiency of GBP to take into account the uncertainties related to the observations and its ability to extrapolate the desired regression function.

5.1 The NSCAT scatterometer transfer function

5.1.1 The problem

NSCAT is a dual swath, Ku-band, scatterometer which was designed by NASA and constructed under its supervision. Its goal was to determine the wind vector over the ocean at global scale with an optimum space and time coverage. NSCAT uses 6 antennas, three for each swath giving a very large and unique data set which is used to determine the wind vector at the global scale. The two mid antennas operate in a dual polarized mode (vertical and horizontal mode) while the four others operate in a vertical polarized mode only. Most of the algorithms which have been proposed to compute the wind from scatterometer measurements are based on the inversion of the Geophysical Model Function (GMF) [15]. The GMF is the transfer function of the scatterometer, it gives the scatterometer signal (sig_0) as a function of the wind vector and the incidence angle (which is the angle between the radar beam and the vertical at the illuminated cell).Figure (6) presents three of the six antanna and the different variables:the wind speed (U), the azimuth angle (χ) and the incidence angle (θ). In a first order approximation one can assume that the wind vector and the incidence angle allows a determination of the scatterometer signal, so we assume in the following that $sig_0(obs) = sig_0 + \delta(\theta,\ U,\ \chi, sig_0)$ where δ is a zero mean gaussian noise accounting for the hiden phenomena and the instrumental noise which is function of the value of sig_0

The determination of an accurate GMF and of the covariance-matrix are then of a fundamental interest. Moreover, as the mid beam is dual polarized, the determination and the interpretation of the wind vector and the incidence angle dependences can be improved by the use of a multivariate regression. In the following, a large North Atlantic data set of NSCAT's mid-beam antenna colocated with the European Center for Medium-Range Weather Forecasts (ECMWF) wind fields is used to determine a synthetic NSCAT geophysical model function (GMF) providing the vertical (sig_0^{VV}) and horizontal (sig_0^{HH}) polarized radar cross sections and the conditional covariance-matrix.

5.1.2 The method

The scatterometer signal is a function of the wind vector and the incidence angle, it also depends on the amplitude of the signal itself. According to these considerations, we apply the algorithm presented in (3.2.2) but trained two different MLPs($(\mathbf{F})_1$ and $(\mathbf{F})_2$ as mentionned at the end of the section. We assume here that the wind vector is known without errors and do not make use of RGB. The first one, $\mathbf{F}_1$, uses the classical quadratic error cost function and gives an estimate of the conditional mean of the measurements:

$$\mathbf{F}_1\Big((\theta,\ U,\ \chi),\mathbf{W}\Big)\ \approx\ \mathbf{E}\big[\ (sig_0^{VV},\ sig_0^{HH})/\ \theta,\ U,\ \chi\big] \tag{42}$$

The second function $\mathbf{F}_2\Big((\theta,\ U,\ \chi),(sig_0^{VV},sig_0^{HH}),\mathbf{W}\Big)$ which uses the values of sig_0^{VV} and sig_0^{HH} given by $\mathbf{F}_1$ minimizes the maximum likelihood and provides an estimate of the conditional covariance matrix given the wind vector, the incidence angle and the scatterometers signals. $\mathbf{F}_2$ allows to compute the variances of sig_0^{VV} and sig_0^{HH} together with their covariance. The architecture of $\mathbf{F}_1$ has:

- an input layer with four neurons corresponding to the wind speed U, the incidence angle θ, $\cos\chi$ and $\cos(2\chi)$ in order to benefit of the bi-harmonic variation with respect to the azimuth angle,
- two hidden layers with ten neurons on each,
- and two linear neurons for the outputs which provide the estimates of the desired measurements $sig_0^{VV}$and sig_0^{HH}.

The architecture of the $\mathbf{F}_2$ has:

- an input layer with six neurons corresponding to U, θ, $\cos\chi$, $\cos(2\chi)$ and (sig_0^{VV},sig_0^{HH})
- two hidden layers with 20 neurons on each
- and three neurons for the outputs which provide the three coefficient of the Choleski decomposition.

The parameters $\mathbf{W}$ of $\mathbf{F}_1$ and $\mathbf{F}_2$ are estimated by using SBP implemented with a second order gradient descent algorithm. Their accuracy is related to the quality of the learning data set. The global data set includes a large range of situations which enables to take into account the effect of extra variables such as sea surface temperature or long wave modulation. The learning set has about 290,000 colocated vectors $\Big((U,\ \chi,\theta),\big(sig_0{}^{VV},sig_0{}^{HH}\big)\Big)$ extracted from the global data set where we tried to equally represent all speeds and directions in order to get a statistically representative data set. An independent test set of 220,000 colocated data extracted from the global data set and covering the whole globe was used for estimating the performances of $\mathbf{F}_1$ and $\mathbf{F}_2$.

5.1.3 The results

To test the accuracy of $\mathbf{F}_1$ and $\mathbf{F}_2$, we apply the statistical tests presented in section (4.3). The ECMWF wind vectors collocated with the observed $\big(sig_0^{VV},sig_0{}^{HH}\big)$ are partitioned in $11\times36\times8$ bins of azimuth angle of 10^o, wind speed of $2\,ms^{-1}$ and incidence angle of 5^o each. The wind speed ranges between 2 and $24\,ms^{-1}$ and the incidence angle from 15^o and 55^o. We obtain for each bin an observed sample $\big(sig_0^{VV},sig_0{}^{HH}\big)$. According to the assumption on the output noise, this sample can be considered as normally distributed. We check on each bin the validity of the estimated means, variances and covariances using the two tests presented in section (4.3) with a confidence level of $\alpha=5\%$. The hypothesis

that the computed means are equal to the empirical ones is accepted for 100% of the bins, the hypothesis on the covariance matrix is accepted 80% of the time.

The physical behavior of $\mathbf{F}_1$ is presented in Figure (7). These figures display the variations of $\mathbf{F}_1$ for the two polarizations with respect to the wind azimuth for different wind speeds and at the incidence angle of 35°. For the two polarizations, $\mathbf{F}_1$ exhibits the biharmonic dependence with respect to the azimuth and the upwind-downwind modulation. The up-wind and down-wind maxima are at 0^o and 180^o and the two minima are at 90^o and 270^o. These results corresponds to what is known about the physics of the scatterometer measurement. The variances and the correlation of the two output noises are computed according to the outputs of $\mathbf{F}_2$. They are shown in figure (8) with respect to the azimuth angle for the same wind speeds and incidence angle. Figure (9) gives the correlations with respect to the incidence angle. The finction $\mathbf{F}2$ allows us to determine the error bars. When compared to classical GMFs, the neural network GMF is of better quality. Moreover for the first time we are able to estimate the variance and consequently the errror bars of the GMF. These errors bars will be usefull to interpret the scatterometer measurements and the accuracy of the wind vector retrieval procedure.

5.2 Ocean colour: Regression with noisy in-situ data

5.2.1 The problem

Quantitative assessment of oceanic primary production and its role in the global carbon cycle is a critical environmental and scientific issue. Knowledge of primary production is necessary to calculate new production, derive the effect of biological processes on the partial pressure of carbon dioxide (CO_2), and, therefore, better understand how phytoplankton carbon fixation affects the net CO_2 flux accross the air-sea interface. Primary production depends on light availability and other environmental factors (temperature, nutrients) and on the amount of phytoplankton present for photosynthesis. The amount of phytoplankton and their optical properties (absorption, scattering) affect the spectral diffuse reflectance of the ocean, defined as the ratio of upwelling to downwelling irradiance at a given depth. Since phytoplankton pigments generally absorb more in the blue than in the green, the greener the water, the more phytoplankton. Thus by measuring ocean color, meaning the spectral reflectance at zero depth, R(λ), one can obtain estimates of phytoplankton pigment concentration, one of the key variables affecting primary production [5].

A variety of optical transfer functions (bio-optical models) have been proposed to quantify the influence of chlorophyllous pigments on spectral reflectance. The bio-optical relationships are generally established by analyzing concomitant reflectances and pigment data. Empirical relationship exist which relate the spectral reflectance at a given wavelenght λ with the absorbtion coefficient $a(\lambda)$

$$R(\lambda) = C\frac{b(\lambda)}{a(\lambda)} \tag{43}$$

Where $a(\lambda)$ is a non linear function of the pigment concentration, C and $b(\lambda)$ being independant of it. An accurate determination of the non linear relation $\mathbf{G}$ giving the

pigment concentration with respect to $a(\lambda)$ is thus of interest to retrieve $R(\lambda)$ and thus ocean color.

The bio-optical data set we use here is taken off shore the California coast and includes the phytoplankton absorption in the blue part of the spectrum (443 nm), it is part of the CALCOFI experiment [5]

5.2.2 Performances and comparisons

The global set of observations (the CALCOFI data) is made of 112 pairs of concomitant absorbtion coefficient $a(\lambda)$ and pigment concentration, see (Fig. (10)). According to the experts 20% of the two signals of each pair are supposed to be noisy. Learning with GBP, which takes into account the variability of both the input and the output noises could improve the determination of $\mathbf{G}$. As the data set of observations is small, we use of cross validation technique: the algorithms are run using a learning set of 104 observations, the performances being tested using the 8 remaining data (denoted by T). The learning procedures are repeated 14 times (112/8), Table (4) gives the averaged performances for the 14 distinct learning procedures:

$$\text{Relative-RMS} = \sqrt{\frac{1}{8}\sum_{T}\left(\frac{y^{obs}-y^{NN}}{y^{obs}}\right)^2}$$

As the desired relationship exhibit quite a linear trend in the log domain (see Figure (11)), the physicists choose for the required model F, the linear regression estimated in the log domain. This model gives a good results at low concentration, but fail to retrieve the high concentrations. The log linear model does not reproduce the saturation phenomenon and generally overestimate the observations. We compare the performances obtained by the log linear model, GBP and SBP. The optimal MLP architecture used for SBP and GBP has one input, one linear ouput and 2 hidden layers of 3 neurons each, we take as smoothing parameter $\gamma = 0.01$. Using GBP we detect 4 outliers which where removed from the observation set, one of them the upper point in the right corner was removed by the physicists as defective, performances are given in table (4) which presents the average of the RMS and the Relative RMS during the $112/8 = 14$ learning phases. Figure (12) presents the functions given by SBP, GBP and the log linear function used by the physicists. SBP and GBP retrieve the saturation phenomena, and GBP present a more realistic behavior as the saturation gives smaller absorbtion coefficients.

6 Conclusion

The general formulation of nonlinear least squares regression using multilayered Perceptrons has been presented in this paper. The aim is to build a non linear model between a set of data and a set of model parameter which represent a physical system. Two cases of increasing complexity have been treated: the case of noisy output data and deterministic input data and the case of noisy data and noisy model parameters. Multilayered Perceptrons can be efficient tools in all cases. In particular, a Generalized learning rule has been

proposed to deal with noisy data and noisy model parameters. The main interest lies in the fact that the algorithm allows to refine the learning set, which consitutes the basic new results of the work reported here. More generally, this paper provides the basis of a general MLP regression theory. The algorithms have been successfully tested on two real-world problem in Geophysics. The good results suggests that multilayered Perceptrons can emmerged as an efficient nonlinear regression model for a wide range of applications. Other examples using the same methology have proved their efficiency to deal with non li near inverse problems in Geophysics [25] [6] [16] [20]. Several other work are in progress.

Acknowledgments

This work was funded by French Service Hydrographique et Océanique de la Marine in the framework of the Exploratory Development in Ocean Acoustic Tomography (nňo 99.87.027..00.470.29.25). This work was also funded by the ECC in the framework of the NeuroSAT program (env94-CT96-0314).

References

[1] Anderson T.W. (1958). *"An introduction to multivariate statistical analysis"*, John Wiley & Sons, Inc, New York 374p

[2] Battiti, R. (1992) First- and second-order methods for learning: Between steepest descent and Newton's Method, Neural Computation, Vol. **4**, pp. 141–166.

[3] Bishop, C. (1994) Mixture density networks , technical report NCRG4288, Aston University, Birmingham.

[4] Bishop, C. (1995) *"Neural Networks for Pattern recognition"*, Calendon Press - Oxford.

[5] Bricaud, A., Babin, M., Morel, A. and Claustre, H. (1995) Variability in the chlorophyll-specific absorbtion coefficients of natural phytoplantankon : Analysis and parameterization, Journal of geophysical Research, vol. 100, No. C7, pp. 13,321-13,332.

[6] Cornford, D., Ramage, G., Nabney, I. (1999), A scatterometerneural network neural model with input noise, Neurocomputing (30) 1-4 pp13-21

[7] Cybenko, G. (1989), Approximation by superposition of a sigmoidal function, Math. Control Signal Systems, **2**, pp. 303–314.

[8] Funahashi, K.I. (1989), On the approximate realization of continuous mapping by neural networks, Neural Networks, Vol. **2**, pp. 185–192.

[9] Härdel, W. (1990), *Applied Nonparametric regression.* Cambridge university press.

[10] Haykin, S., (1996) *S Neural Networks a comprehensive foundation*, Prentice Hall.

[11] Hornik, K., Stinchcomb, M. and White, H. (1989) Multi-Layered feedforward networks are universal approximators, Neural Networks, **2**, pp. 359–366.

[12] Jordan, M. and rumelhard, D. (1992) Forward models: Supervised learning with a distal teacher, Cognitive Science, **16**, pp. 307–354.

[13] Jordan, M.I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm.Neural Computation, **16**: pp 181-214.

[14] MacKay, D. (1992a) Bayesian interpolation, Neural Computation, **4**, pp. 415–447.

[15] Mejia, C., Badran, F., Bentamy, A., Crepon, M., Thiria, S. and Tran, N. (1999) Determination of the geophysical model function of NSCAT and its corresponding variance by the use of the neural networks, Journal of geophysical Research, vol. 104, No. C5, pp. 11,539-11,556.

[16] Nabney, I.T., Cornford, D.,Williams, C.K.I. (1999) Structured neural network modelling of multi-valued functions for wind vector retrieval from satellite scatterometer measurements, Neurocomputing (30) 1-4 pp 3-11.

[17] Nix, D. and Weigend, A. (1995) Learning local error bars for nonlinear regression, in Advanced in neural Information Processing Systems, G.Tesauro & Al. Eds, MIT Press, Cambridge, pp. 489–496.

[18] Poggio, T. and Girosi, F. (1990) Networks for approximation and learning, Proceedings of the IEEE, Vol. **78**, pp. 1481–1497.

[19] Reed, Russel D.and Marks, Robert J (1998) *"Neural Smithing. Supervised Learning in Feedforward Neural Networks"* A Bradford Book MIT Press.

[20] Richaume, P.,Mejia, C.,Thiria, S., Tran, N., Crepon, M., Roquet, H., Badran, F.(1999) Neural Network Wind retrieval from ERS1 Scatterometer Data. To appear in Journal of Geophysical Research.

[21] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Parallel Distributed Processing, Vol. **1**, MIT Press, Cambridge.

[22] Stéphan, Y., Thiria, S. and Badran, F. Application of multi-layered neural networks to ocean tomography inversions,Inverse Problems in Engineering, Vol. **1**, pp. 181–304.

[23] Stéphan, Y., Démoulin, X. and Sarzeaud, O. (1998) Neural direct approaches for Geoacoustic inversion, Journal of Computational Acoustics, Vol.**6**, No 1-2, 151-166.

[24] Thodberg, H. (1996) A review of Bayesian neural networks with an application to near infrared spectroscopy, IEEE Transactions on Neural Networks, Vol. **7**, no 1, pp. 56–72.

[25] Thiria, S., Mejia, C., Badran F. and Crépon, M. A neural approach for modeling nonlinear transfer function: Application for wind retrieval from spaceborn scatterometer data, Journal of Geophysical Research, 98, C12, 22,827–22,841 (1993).

[26] Tarantola, A. (1987) *"Inverse Problem Theory"*, Elseiver Science Publisher, Amsterdam, 613 p.

[27] Weigend, A.S., Zimmermann, H.G.,Neuneier, R. (1996). Clearning, Neural Networks in Financial Engineering, Proceedings of the Third International Conference on Neural Networks in the Capital Markets, NNCM-95, pp. 511-522

[28] Weigend, A.S., Mangeas, M. and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. International Journal of Neural Systems. **6**: 373-399.

[29] Williams, P.M. (1996). Using Neural Networks to Model Conditional Multivariate Densities. Neural Computation**8**, 843–854.

[30] White, H. (1990). Connectionnist Nonparametric Regression: Multi-Layer Feedforward Networks Can Learn Arbitrary Mappings. Neural Networks **3**, 535–549.

Let us consider the case of the regression of a multi-dimensional variable $\mathbf{y}$ with respect to a multi-dimensional variable $\mathbf{x}$. The general case in which the variance on $\mathbf{y}$ depends on $\mathbf{x}$ values will be considered. This variance is given by a variance-covariance matrix denoted $\mathbf{C}_y(\mathbf{x})$. Learning generally consists of minimizing a cost function defined on the problem description set. This function is the generalized least square error function:

$$R(\mathbf{W}) = \iint \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \tag{44}$$

which can also be written, using Bayes rule:

$$R(\mathbf{W}) = \int \Big(\int \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \Big)\, p(\mathbf{x})\, d\mathbf{x} \tag{45}$$

Let us write $\mathbf{R}(\mathbf{W})$ in the form

$$R(\mathbf{W}) = \int I(\mathbf{W}) p(\mathbf{x})\, d\mathbf{x} \tag{46}$$

Elementary manipulations in the internal integral lead to:

$$\begin{aligned} I(\mathbf{W}) = & \int \Big(\mathbf{y} - \mathbf{E}(\mathbf{Y}/\mathbf{x}) + \mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \\ & \Big(\mathbf{y} - \mathbf{E}(\mathbf{Y}/\mathbf{x}) + \mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \end{aligned} \tag{47}$$

This can be written:

$$\begin{aligned} I(\mathbf{W}) \;=\; & \int \Big(\mathbf{y} - \mathbf{E}(\mathbf{Y}/\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \Big(\mathbf{y} - \mathbf{E}(\mathbf{Y}/\mathbf{x})\Big) p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \\ & + 2\Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) \mathbf{C}_y^{-1}(\mathbf{x}) \int \Big(\mathbf{y} - \mathbf{E}(\mathbf{Y}/\mathbf{x})\Big) p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \\ & + \int \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{y}/\mathbf{x})\, d\mathbf{y} \end{aligned} \tag{48}$$

The aim is to minimize $R(\mathbf{W})$. As the first term in Eq. (48) does not depend on $\mathbf{W}$ and the second term is zero, minimizing $R(\mathbf{W})$ is equivalent to minimizing the third term such as:

$$\min_{\mathbf{W}}(R) = \min_{\mathbf{W}} \Big(\iint \mathbf{A} p(\mathbf{y}/\mathbf{x}) p(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y} \Big)$$

$$\mathbf{A} = \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}) \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - (\mathbf{F})(\mathbf{W},\mathbf{x})\Big) \tag{49}$$

Since $\mathbf{E}(\mathbf{Y}/\mathbf{x})$ does not depend on $\mathbf{y}$, we have:

$$\min_{\mathbf{W}}(R) = \min_{\mathbf{W}}\left(\int Bp(\mathbf{x})\, d\mathbf{x} \right) \tag{50}$$

$$\mathbf{B} = \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{C}_{\mathbf{y}}^{-1}(\mathbf{x}) \Big(\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) \tag{51}$$

However, the theoretical cost function, as defined in Eq. (44), is often unavailable. Rather, a finite set of independant observations $L = \left\{(\mathbf{x}^i, \mathbf{y}^i), i = 1 \dots N^{obs}\right\}$ can be obtained. Learning is done on this set by minimizing the empirical risk defined by:

$$R_{emp}(\mathbf{W}) = \sum_{i=1}^{N^{obs}} \Big(\mathbf{y}^i - \mathbf{F}(\mathbf{W},\mathbf{x}^i)\Big)^T \mathbf{C}_y^{-1}(\mathbf{x}^i) \Big(\mathbf{y}^i - \mathbf{F}(\mathbf{W},\mathbf{x}^i)\Big) \tag{52}$$

which is a discrete approximation of the theoretical cost function R. Eq. (50) shows that, for a given value of $\mathbf{x}$, the minimum value of R is obtained when the network achieves an approximation of the mean field of variable $\mathbf{y}$. The accuracy of the approximation (regression) depends on the value of R. In particular, the accuracy of the approximation may be affected for two reasons. On one hand, the architecture may not be well-chosen (too few or too many weights, inappropriate set of functions, ...). On the other hand, the observation set may not be consistent enough with the true field of the variable to regress. However, if these two constraints are kept, it can be admitted that, for any $(\mathbf{x}, \mathbf{C}_y(\mathbf{x}))$ value, the output of the network is such as:

$$\mathbf{F}(\mathbf{W},\mathbf{x}) \approx \mathbf{E}(\mathbf{Y}/\mathbf{x}) \tag{53}$$

the accuracy can be computed on each principal axes of the covariance matrix $\mathbf{C}_y^{-1}$, as demonstrated in appendix.
In the previous computation, it is shown that the output of a MLP approximates the mean value of $\mathbf{y}$ conditionally to $\mathbf{x}$. The purpose of this annex is to show that the accuracy of this approximation can be obtained by the principal axes of the covariance matrix.
Let us re-write the generalized least-square functions in the form:

$$R(\mathbf{W}) = \iint \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{M}^{-1} \Big(\mathbf{y} - \mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \tag{54}$$

where $\mathbf{M}$ is symmetrical positive definite matrix.
Let $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_q)$ the orthonormal matrix of the eigenvectors and σ_j^2 the eigen value associated to $\mathbf{u}_j$. We can write that:

$$\mathbf{M}^{-1} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^T \quad \text{where} \quad \mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_q^2} \end{bmatrix} \tag{55}$$

The cost function can be written as:

$$R(\mathbf{W}) = \iint \Big(\mathbf{U}^T\mathbf{y} - \mathbf{U}^T\mathbf{F}(\mathbf{W},\mathbf{x})\Big)^T \mathbf{D}^{-1} \Big(\mathbf{U}^T y - \mathbf{U}^T\mathbf{F}(\mathbf{W},\mathbf{x})\Big) p(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \tag{56}$$

or, in the discrete form:

$$R(\mathbf{W}) = \sum_{j=1}^{q} \iint \Big(\frac{\mathbf{u}_j.\mathbf{y} - \mathbf{u}_j.\mathbf{F}(\mathbf{W},\mathbf{x})}{\sigma_j}\Big)^2 p(\mathbf{x},\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \tag{57}$$

Denoting $\mathbf{Y} = \mathbf{U}^T\mathbf{y} = (Y_1, \cdots, Y_p)$, it yields:

$$R(\mathbf{W}) = \sum_{j=1}^{q} \iint \Big(\frac{Y_j - \mathbf{u}_j.\mathbf{F}(\mathbf{W},\mathbf{x})}{\sigma_j}\Big)^2 p(\mathbf{x},\mathbf{U}\mathbf{Y})\, d\mathbf{x}\, d\mathbf{Y} \tag{58}$$

From computations described in subsection 2.2 (Eq. (47)–Eq. (50)), we obtain:

$$R(\mathbf{W}) = \sum_{j=1}^{q} \int \Big(\frac{\mathbf{E}(Y_j/\mathbf{x}) - \mathbf{u}_j.\mathbf{F}(\mathbf{W},\mathbf{x})}{\sigma_j}\Big)^2 p(\mathbf{x})\, d\mathbf{x} + \text{constant} \tag{59}$$

or, equivalently:

$$R(\mathbf{W}) = \sum_{j=1}^{q} \int \Big(\frac{\mathbf{u}_j.\big[\mathbf{E}(\mathbf{Y}/\mathbf{x}) - \mathbf{F}(\mathbf{W},\mathbf{x})\big]}{\sigma_j}\Big)^2 p(\mathbf{x})\, d\mathbf{x} + \text{constant} \tag{60}$$

Eq. (60) shows that the MLP approximates the mean value of $\mathbf{y}$ conditionally to $\mathbf{x}$ and that the accuracy of this approximation can be measured by the principal axes $\mathbf{u}_j$ and variance σ_j^2.

Table 1: Performances reached on the test set of $N = 300$ samples using SBP and GBP, each row presents a different indicator (RMS, Real-RMS and $\sigma(x^{est} - x^{obs})$) . Learning has been done using four distinct learning sets taken at random from D2, the four learning sets differ by their size $N^{obs} = 25, 50, 100, 200$

	SBP	SBP	SBP	SBP	GBP	GBP	GBP	GBP
N^{obs}	25	50	100	200	25	50	100	200
RMS	0.706	0.384	0.307	0.269	0.736	0.286	0.292	0.252
Real-RMS	0.613	0.343	0.191	0.137	0.668	0.145	0.150	0.102
$\sigma(x^{est} - x^{obs})$	-	-	-	-	0.124	0.136	0.115	0.089

1 Tables

Table 2: Performances reached by GBP on the test set of $N = 300$ samples after retraining. At the end of the experiment reported in table 1, outliers where detected using the χ^2 test with a confidence level of 95%. The second row gives the number of detected outliers which ever contains the 3 ones added to each learning set.

	GBP	GBP	GBP	GBP
N^{obs}	25	50	100	200
$N^{outliers}$	2	3	6	9
RMS	0.414	0.266	0.258	0.242
Real-RMS	0.376	0.139	0.085	0.063
$\sigma(x^{est} - x^{obs})$	0.079	0.079	0.074	0.067

Table 3: Performances of GBP on the test set of $N = 300$ samples when dealing with a non constant unknown output noise. The learning set is made of 200 samples of D4 GBP has been used after the estimation during phase 1 of $\mathbf{C}_y(x)$ using the algorithm of section (3.2.2)

	SBP	GBP(phase1)	GBP
RMS	0.607	0.638	0.612
Réelle-RMS	0.229	0.254	0.169

Table 4: Performances of the MLP trained with GBP and the log-linear model. Learning is made using the 112 observations of concomitant absorbtion coefficient and pigment concentration. The RMS and the relative-RMS are the average of the 14 performances obtained on the test sets during the 14 distinct learning. During the cross validation, on average we detected 4.7 outliers.

	Log-linear.	SBP	GBP
N^{obs}	112	112	112
RMS ($\times 10^{-2}$)	1.16	1.30	1.23
Relative RMS	21.53 %	25.71 %	21.20 %

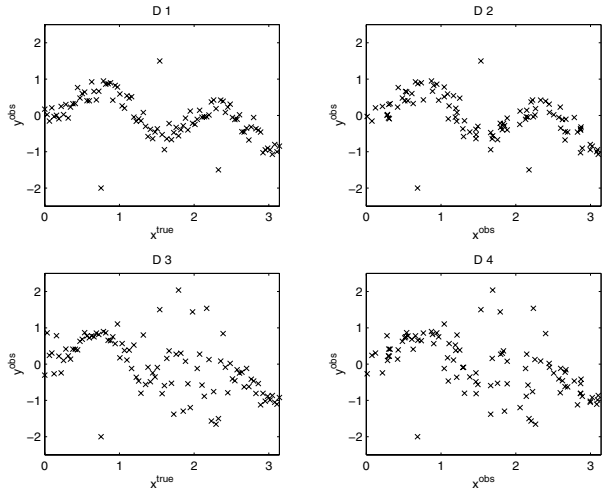

Figure 1: Presentation of the four distinct distributions D1,D2,D3,D4.

2 Figures

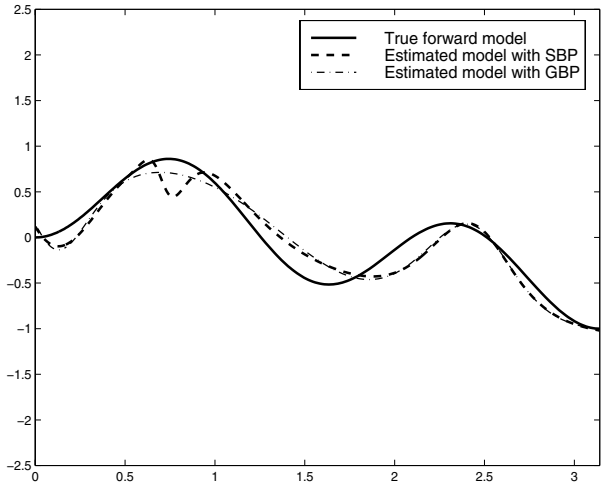

Figure 2: MLP Models obtained after learning with SBP (bold dashed line) and GBP (Light dashed line),plain bold line represent the theoretical relationship to be retrieved . The learning set is made of 200 samples of D3 (deterministic $\mathbf{x}$ and noisy $\mathbf{y}$). SBP and GBP only differ by the cost function they use, Least square for SBP and weighted least square for GBP

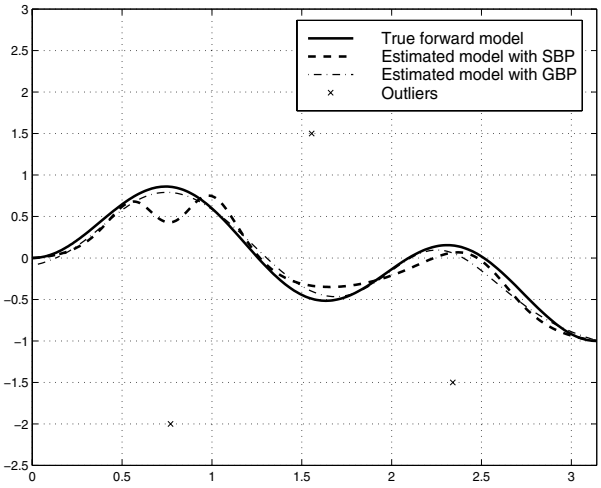

Figure 3: MLP Models obtained after learning with SBP (bold dashed line) and GBP (Light dashed line), plain bold line represent the theoretical relationship to be retrieved. The learning set is made of 200 samples of D2 (noisy $\mathbf{x}$ and noisy $\mathbf{y}$).

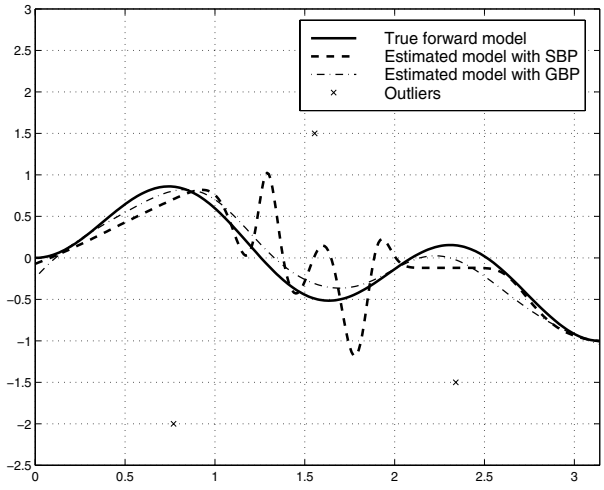

Figure 4: MLP Models obtained after learning with SBP (bold dashed line) and GBP (Light dashed line), plain bold line represent the theoretical relationship to be retrieved. The learning set is made of 200 samples of D4 (noisy $\mathbf{x}$ and noisy $\mathbf{y}$). The known output noise is input dependant $\sigma_y^2(x) = 0.01 + 0.25\,[1 - \sin(2.5x)]^2$.

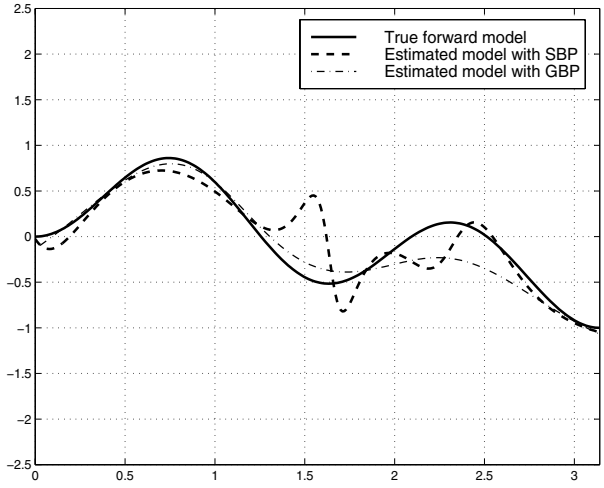

Figure 5: MLP Models obtained after learning with SBP (bold dashed line) and GBP (Light dashed line), plain bold line represent the theoretical relationship to be retrieved. The learning set is made of 200 samples of D4 (noisy $\mathbf{x}$ and noisy $\mathbf{y}$), the output noise is input dependant: $\sigma_y^2(x) = 0.01 + 0.25\left[1 - \sin(2.5x)\right]^2$. As the noise is now supposed to be unknown, GBP has been used after the estimation, during phase 1, of $\mathbf{C}_y(x)$ using the algorithm of section (3.2.2).

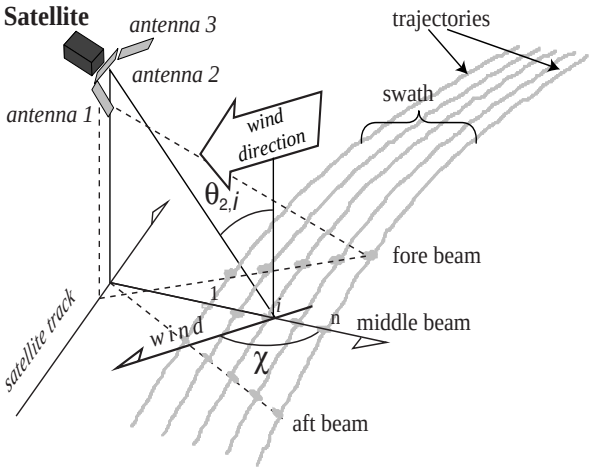

Figure 6: Definition of geophysical parameters; θ is the incidence angle, and χ is the azimuth angle.

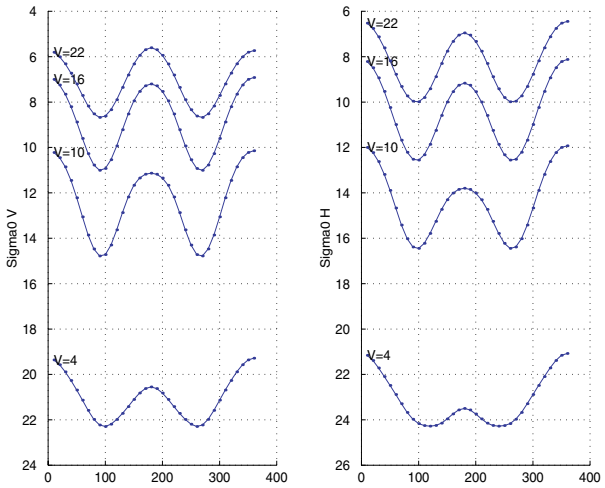

Figure 7: Variations of F_1 for the two polarizations with respect to the wind azimuth angle for different wind speeds and at the incidence angle of 35 degrees.

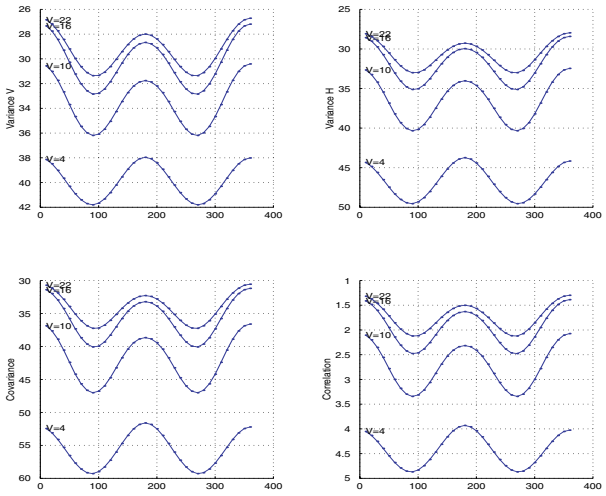

Figure 8: The variances and the correlation of the two output noises with respect to the wind azimuth angle for different wind speeds and at the incidence angle of 35 degrees.

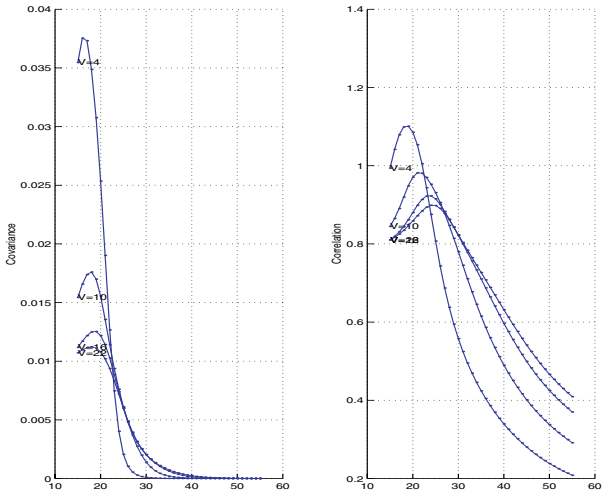

Figure 9: the correlations with respect to the incidence angle for different wind speeds and at the wind azimuth angle of 180 degrees.

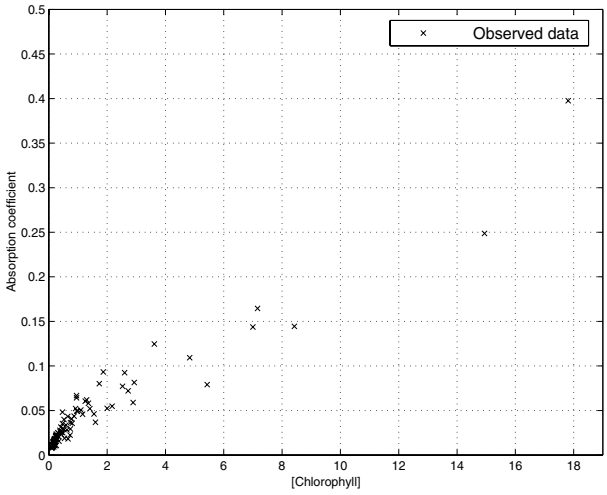

Figure 10: Global set of CALCOFI data in linear scale: 112 pairs of concomitant absorbtion coefficient $a(\lambda)$ and pigment concentration. For each pair 20% of the signals are supposed to be noise.

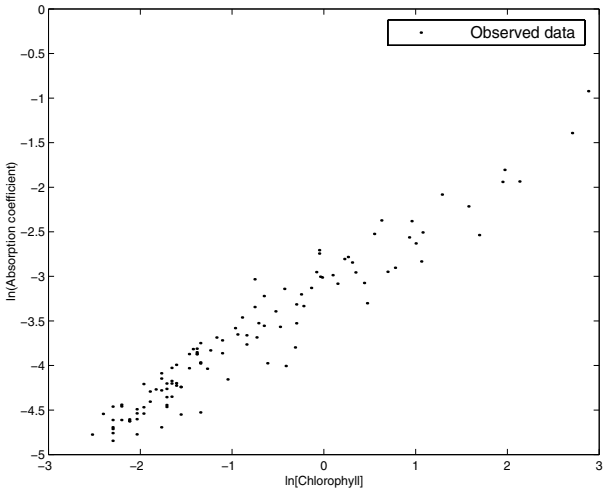

Figure 11: Global set of CALCOFI data in Log-Log coodinates: 112 pairs of concomitant absorbtion coefficient $a(\lambda)$ and pigment concentration.

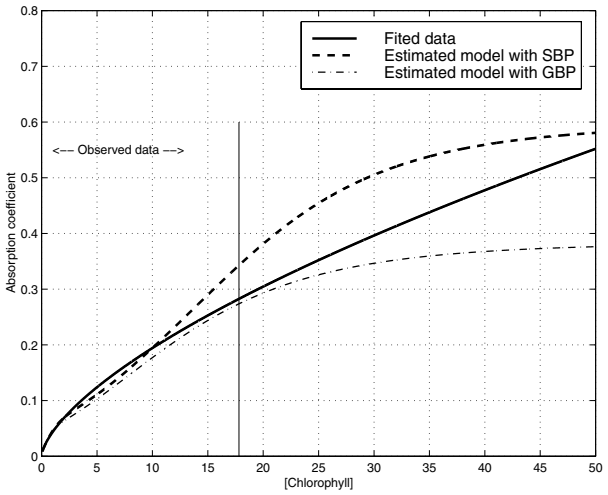

Figure 12: MLP Models obtained after learning with SBP (bold dashed line) and GBP (Light dashed line), plain bold line represent the log linear model.The learning set is made of the 112 observations of the CALCOFI data