Available online at www.sciencedirect.com
ScienceDirect
Neural Networks 20 (2007) 479–483
Neural Networks
www.elsevier.com/locate/neunet

2007 Special Issue

Model inversion by parameter fit using NN emulating the forward model — Evaluation of indirect measurements

Helmut Schiller

GKSS Forschungszentrum, PF 1160, 21494 Geesthacht, Germany

Abstract

The usage of inverse models to derive parameters of interest from measurements is widespread in science and technology. The *operational* usage of many inverse models became feasible just by emulation of the inverse model via a neural net (NN).

This paper shows how NNs can be used to improve inversion accuracy by minimizing the sum of error squares. The procedure is very fast as it takes advantage of the Jacobian which is a byproduct of the NN calculation. An example from remote sensing is shown. It is also possible to take into account a non-diagonal covariance matrix of the measurement to derive the covariance matrix of the retrieved parameters.

Keywords: Inverse model; Parameter fit; Neural net; Jacobian; Model emulation; Levenberg–Marquardt; Covariance matrix

1. Introduction

The problem of the evaluation of indirect measurements is widespread in (but not restricted to) remote sensing. Examples are:

- a sensor onboard a satellite measures spectral resolved radiances emanating from a certain area of the ocean surface — the parameters of interest are the concentrations of the water constituents;
- flood discharge to be derived from water levels and surface velocities.

A common method for tackling the problem is first to model the causal dependency of the measured quantities from the parameters of interest: in our former example the parameters of interest, the concentrations of water constituents, cause a certain coloration of the water which is registered by the measurement of spectral resolved radiances by a sensor onboard a satellite. In the second step, to derive the parameters of interest from the measurements, the model must be inverted. Nowadays, in the case of high model complexity, it is common practice to emulate the inverse model by a neural network (NN). In order not to apply the inverse NN outside its scope, a domain check (Schiller & Krasnopolsky, 2001) can be performed by combining the inverse NN with an NN emulating the forward model (forward NN). A comprehensive discussion of problems related to NNs emulating forward/inverse models can be found in Krasnopolsky and Schiller (2003).

The idea of the scope check can be extended: the output of the inverse net is not only fed into the forward net to check if its output complies to the original measurements but it is also used to improve iteratively the accuracy of the parameters of interest (Schiller & Doerffer, 2005). This becomes feasible since the NN emulation of the forward model allows us to calculate the Jacobian of the forward model efficiently and thus the Levenberg–Marquardt optimization scheme can be used to determine the parameters of interest best fitting the measurements.

The paper is organized as follows. In Section 2 the construction of the inverse/forward NN is sketched and then the implementation of the optimization loop is presented. A realization of this procedure is discussed in Section 3. A generalization of this scheme for the case of a non-diagonal covariance matrix of measurement errors is given in Section 4. Conclusions are drawn in Section 5.

2. Minimization of sum of error squares

We assume that the result of a measurement is a vector of quantities **r** and that changes in **r** are caused by changes of

E-mail address: schiller@gkss.de.

0893-6080/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2007.04.022

some underlying variables $\mathbf{c}$ (cause). The model describing the relation $\mathbf{r} = \mathbf{m}(\mathbf{c})$ might be physically based but could also be an empirical one. In most relevant cases there will be a certain region in the $\mathbf{c}$ space where the inverse function $\mathbf{c} = \mathbf{m}^{-1}(\mathbf{r})$ exists. (This is the necessary condition which must be fulfilled for whatever retrieval procedure is to be used.)

The model $\mathbf{m}$ is used to compute a large set of $\{\mathbf{r}_i, \mathbf{c}_i\}$ pairs. This set is used for the training/test of two NNs: one NN to emulate the forward model $\mathbf{r} = \mathbf{m}_{\mathrm{NN}}(\mathbf{c})$ and a second NN which emulates the inverse model $\mathbf{c} = \mathbf{m}_{\mathrm{NN}}^{-1}(\mathbf{r})$.

In practice, for various reasons (measurement errors, flaws of the preprocessing, deficiencies in the NN training, ...), the application of $\mathbf{m}_{\mathrm{NN}}^{-1}$ to a measurement $\mathbf{r}_M$ will not lead to the perfect solution ($\mathbf{m}_{\mathrm{NN}}(\mathbf{m}_{\mathrm{NN}}^{-1}(\mathbf{r}_M)) = \mathbf{r}_M$) but rather it will lead to an estimate $\mathbf{c}_0 = \mathbf{m}_{\mathrm{NN}}^{-1}(\mathbf{r}_M)$ with $\mathbf{m}_{\mathrm{NN}}(\mathbf{c}_0) = \mathbf{r}_0 \neq \mathbf{r}_M$.

At this stage it is natural to consider an iterative procedure to improve an estimate $\mathbf{c}_k$ (starting from $\mathbf{c}_0$) by minimization of the sum-of-squares error function, (Bishop, 1995)

$$J(\mathbf{c}) = \frac{1}{2}\sum_n (\epsilon_n)^2 = \frac{1}{2}\|\boldsymbol{\epsilon}\| = \frac{1}{2}(\mathbf{r}_k - \mathbf{r}_M)^{\mathrm{T}}(\mathbf{r}_k - \mathbf{r}_M) \qquad (1)$$

where $\mathbf{r}_k = \mathbf{m}_{\mathrm{NN}}(\mathbf{c}_k)$. As we emulate the forward model by an NN, it is not much overhead to calculate the Jacobian $\mathbf{M}$:

$$(M)_{ni} \equiv \left.\frac{\partial m_n}{\partial c_i}\right|_{\mathbf{c}=\mathbf{c}_k}.$$

This allows us to use the Levenberg–Marquardt algorithm to update the estimate

$$\mathbf{c}_{k+1} = \mathbf{c}_k - (\mathbf{M}^{\mathrm{T}}\mathbf{M} + \lambda\mathbf{1})^{-1}\mathbf{M}^{\mathrm{T}}(\mathbf{r}_k - \mathbf{r}_M). \qquad (2)$$

The parameter λ controls the step size, gently switching between the steepest descent and the Gauss–Newton formula. The recipe is to start with some arbitrary value such as $\lambda = 0.1$. At each step of the procedure, one checks if the parameter given by Eq. (2) results in a decrease in the error function. If the error decreases, the $\mathbf{c}_{k+1}$ is accepted and the λ is decreased by a factor of 20 and the process is repeated. Otherwise, $\mathbf{c}_{k+1}$ is discarded and λ is increased by a factor of 20 before entering Eq. (2). This is repeated until the error decreases.

The replacement of the original model (and its inverse) by NN's in the approach described above does not solve any of the problems connected with model inversion and function minimization in general (ill conditioned, local minima, ...). The crucial point of the above approach is just the replacement itself: in many cases the operational application of the original model is prohibitive — not to mention the many uses of the model in the iteration steps necessary to fit measured data.

3. Example from remote sensing

This section presents a remote sensing application of the algorithm described. It is used to derive the concentrations of water constituents from MERIS (the Medium Resolution Imaging Spectrometer). The primary mission of MERIS is the measurement of sea color in the oceans and in coastal areas. The aim is to convert such measurements of the sea color into a measurement of concentrations of chlorophyll pigment, suspended sediment and *gelbstoff* (dissolved organic material). The measurement of the ocean color relies on the measurement of the radiance of reflected sun light in different spectral bands in the visible range arriving at the satellite. Most (>90%) of the light arriving at the satellite has been reflected by the atmosphere. Therefore, to calculate the directional water leaving radiance reflectances (*i.e.* ocean color) from the measured radiances, a careful atmospheric correction is necessary. The color of a given water mass also depends on the viewing and illumination geometry, *i.e.* the zenith angles of the sun and the satellite and their difference in azimuth. Therefore the forward model for this problem reads $\mathbf{r} = \mathbf{m}(\mathbf{c}, \mathbf{g})$: the water leaving radiance reflectances $\mathbf{r}$ depend on concentrations $\mathbf{c}$ of water constituents and on three angles describing the geometry $\mathbf{g}$ of the situation. The inverse model $\mathbf{c} = \mathbf{m}^{-1}(\mathbf{r}, \mathbf{g})$ derives the concentrations of chlorophyll pigment, suspended sediment and *gelbstoff*.

For the ground-segment of MERIS a retrieval procedure based on NN technology was developed to transform directional water leaving radiance reflectances (≡angular-dependent radiation leaving the water divided by the downwelling irradiance entering the water) measured in eight spectral bands and the three angles pixel by pixel with high efficiency into concentrations of the water constituents: suspended matter, phytoplankton and *gelbstoff* (Schiller & Doerffer, 1999).

Since measurements do not cover the data space with sufficient density, the construction of the NN is based on a large table (130 K entries) of simulated data generated by a Monte Carlo radiative transfer code. For given concentrations of water constituents the Monte Carlo radiative transfer code calculates the angular distribution of water leaving radiance reflectance in eight visible MERIS bands. These angular distributions are sampled in the appropriate angle ranges to derive the entries of the training/test tables for building the NNs: three concentrations, three angles and eight water leaving radiance reflectances. An in-depth description of the radiation transfer calculations is given in Doerffer and Schiller (2007).

The knowledge of the three concentrations is not sufficient to simulate the ocean color: from measurements it was found that a variety of absorption and scattering spectra are possible for each of the three consituents, respectively. In order to cover the range of naturally occuring water spectra, the natural variability of the inherent optical properties of the water constituents was built into the Monte Carlo code by sampling the parameters describing the spectral dependence of the inherent optical quantities from their measured distributions. Details of this bio-optical model can be found in Doerffer and Schiller (2007).

With the table of simulated data, two NNs are trained: (1) $\mathbf{m}_{\mathrm{NN}}^{-1}$ to emulate the inverse model to derive concentrations $\mathbf{c}$ from reflectances $\mathbf{r}$ and geometry information $\mathbf{g}$, and (2) $\mathbf{m}_{\mathrm{NN}}$ to emulate the forward model deriving reflectances $\mathbf{r}$ from concentrations $\mathbf{c}$ and geometry information $\mathbf{g}$.[1]

[1] The NNs have the following architecture: the $\mathbf{m}_{\mathrm{NN}}^{-1}$ has five hidden planes with 45, 16, 12, 8 and 5 neurons, respectively; the $\mathbf{m}_{\mathrm{NN}}$ has four hidden planes with 55, 20, 15 and 10 neurons, respectively.