Ocean Modelling 96 (2015) 117–125

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

Virtual Special Issue
Ocean Surface Waves

Data assimilation of ocean wind waves using Neural Networks. A case study for the German Bight

Kathrin Wahle*, Joanna Staneva, Heinz Guenther

Helmholtz-Zentrum Geestahcht, Max–Planck–Str. 1, Geesthacht 21502, Germany

ARTICLE INFO

Article history:
Received 29 January 2015
Revised 15 July 2015
Accepted 17 July 2015
Available online 26 July 2015

Keywords:
Neural networks
Inverse modeling
Data assimilation
WAM
HF-radar

ABSTRACT

A novel approach of data assimilation based on Neural Networks (NN's) is presented and applied to wave modeling in the German Bight. The method takes advantage from the ability of NN's to emulate models and to invert them. Combining forward and inverse model NN with the Levenberg–Marquardt algorithm provides boundary values or wind fields in agreement with measured wave integrated parameters. Synthesized HF-radar wave data are used to test the technique for two academic cases.

1. Introduction

Data assimilation (DA) into regional wave models are often hampered due to two reasons: when simple assimilation schemes, such as optimal interpolation, are used the model innovation rapidly decreases. Advanced schemes like Ensemble Kalman Filters (EnKF, Evensen (1994, 2003)), variational methods (DIMET and Talagrand (1986)), and particle filters (Van Leeuwen (2009)) on the other hand are very time consuming. Data assimilation was first invented for numerical atmospheric and hydrodynamic modeling but nowadays is also standard in third generation wave modeling. Since the pioneering work of Lionello et al. (1992) several advanced schemes have been implemented with regional applications (e.g. Voorrips et al. (1997)).

One of the major goals of DA is to improve the accuracy of the model forecasts by incorporating data from observations through different methodology. Variational methods attempt to find the maximum a posteriori probability by computation of the cost function gradient and applying optimization methods to search for the cost function maximum. Note that the most common approach is to minimize, for convenience, minus the logarithm of the a posteriori probability. Combined with the common assumption of Gaussian a priori probabilities this leads to a quadratic minimization problem. Kalman filtering takes another approach. The model and observation operators are assumed linear, so that the optimization can be solved analytically. Variational data assimilation systems using adjoint wave models so far suffered from the simplicity of the adjoint. Either errors were introduced due to discretizing the analytical adjoint as in Walker (2006) or the adjoint was based on stationary solutions of the governing equations as in Orzech et al. (2013, 2014). Only recently a 4D-VAR system based on SWAN is under development (Veeramony et al. (2014)).

The formulation of the EnKF is quite similar to the 3D-Var method. The main difference is that the static background error covariance is replaced by the sample covariance. This sample covariance is computed from an ensemble of model forecasts in a procedure very similar to Monte Carlo methods. Since data assimilation is often quite computationally demanding many approximations have been proposed, such as Proper Orthogonal Decomposition (POD, Altaf et al. (2009)). Hunt et al. (2007) proposed algorithmic changes to EnKF and suggested that the new scheme should be called the Local Ensemble Transform Kalman Filter (LETKF). LETKF is an advanced data assimilation method, which has been tested with numerical weather prediction (NWP) models, from storm to global scales (Szunyogh et al. (2008)). The method also introduces changes that improve the computational efficiency of the algorithm and adds flexibilities that are beneficial when non-local observations are assimilated (Fertig et al. (2007)). However, high resolution coastal wave models require a further increase of memory by using EnKF/LETKF data assimilation

* Corresponding author. Tel.: +49 4152871559.
E-mail address: kathrin.wahle@hzg.de (K. Wahle).

http://dx.doi.org/10.1016/j.ocemod.2015.07.007
1463-5003/

approach, making this method a serious barrier for using it in real operational wave model coastal applications.

Here we present a novel assimilation technique based on Neural Networks (NNs) which combines the computational efficiency of sequential methods with non-locality of Kalman and adjoint methods. In general the NN aims to explore an extensive parallel network of simple elements in order to obtain result in a very short time and, at the same time, with insensitivity to loss and failure of some of the elements of the network. After training NN has a lower computational cost than extended and linear KF, variational method, and particle filter. NNs will be used to emulate the regional wave model integrated parameters from boundary values and wind fields and also to emulate the inverse. When applied to measured wave data a combination of the forward and inverse NN will provide improved boundary values or wind fields for use in the wave model. The method can thus be summarized as a statistical adjoint method.

NNs have also been used in the context of data assimilation in wave modeling: Zhang et al. (2006) used NN to emulate model errors in order to improve the forecast skill. Zamani et al. (2010) emulated the forward model and coupled it with an Ensemble Kalman Filter. But so far, NNs have not been used for the assimilation itself.

To explore the feasibility of the assimilation technique we used as a test case the German Bight. The Coastal Observing System for Northern and Arctic Seas (COSYNA) aims at the construction of a long-term observatory for southern North Sea (German Bight). COSYNA integrates near real-time measurements with numerical models in a pre-operational way and provides continuously state estimates and forecasts of the coastal ocean state. Observations consist of in-situ measurements from fixed (piles and buoys) and mobile platforms (FerryBox) as well as of remotely sensed data from shore by HF-radar and from space by satellites. The nested-grid modeling system estimates pre–operationally ocean state variables concerning ocean waves, hydrodynamics and suspended matter in the North Sea and German Bight. The main characteristics of COSYNA however, are the integrated approach of combining observations and numerical modeling by data assimilation. In the future, the system of three HF-radars will also measure wave parameters for the German Bight area. We have thus decided to study additionally the possible impact of assimilation of those high-resolution spatial data into the wave model data. So far, the HF–radar data of surface currents have been analyzed concerning their upscaling potential (Wahle and Stanev (2011)) and data is successfully assimilated in the pre-operational hydrodynamic German Bight model (Stanev et al. (2015)). For a similar regional area - the Liverpool Bay - HF–radar wave data have already be assimilated into a regional wave model using an optimal interpolation (Waters et al. (2013)).

In the following Section 2 we will frame the basic idea of the NN method. In Section 3 the model set–up is specified. Training and testing of the NNs is described in Section 4 and the application of the scheme to two academic test cases follows in Section 5. We then summarize our results and give some outlook in 6.

2. Method

In the proposed method, we use Neural Networks to emulate a physical model and its adjoint. Given some measurements $\vec{r}_{\rm m}$ a first (statistical) estimate of model forcings $\vec{c}$ are derived by NN emulating the adjoint model: $\vec{c}_1 = {\rm NN}^{-1}(\vec{r}_{\rm m})$. Subsequent application of forward NN gives an emulated model output $\vec{r}_1$ error χ_1^2 which can be subsequently minimized using the Levenberg–Marquardt algorithm:

$$\chi_k^2 = (\vec{r}_m - \vec{r}_k)^{\rm T}\,\mathbf{C}\,(\vec{r}_m - \vec{r}_k) \quad (1)$$

$$\vec{c}_{k+1} = \vec{c}_k + (\mathbf{M^T C^{-1} M} + \lambda\mathbf{1})^{-1}\,\mathbf{M^T C^{-1}}\,(\vec{r}_m - \vec{r}_k) \quad (2)$$

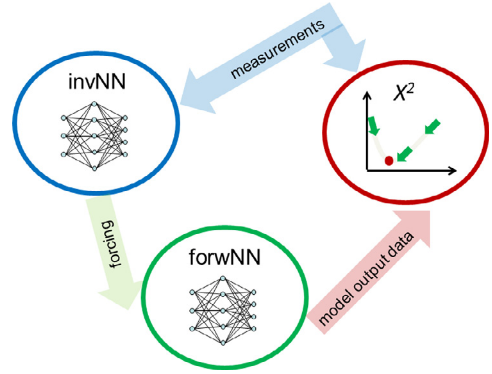

Fig. 1. Sketch of assimilation scheme. The combination of forward and backward NN is used to minimize error between measurement and model output.

where **C** is the covariance matrix and **M** is the Jacobian matrix with $\mathbf{M} \equiv (\frac{\partial \vec{r}(\vec{c})}{\partial \vec{c}})|_{\vec{c}=\vec{c}_k}$. $\lambda \in [0, 1]$ is a control parameter allowing to gently adjust between a Gauss-Newton ($\lambda = 0$) and a gradient descent ($\lambda = 1$) scheme.

The combination of emulating forward/inverse models with NNs and applying Levenberg–Marquardt algorithm is not a new idea. It has been used in remote sensing for parameter retrieval and out of scope check (Schiller (2007), Schiller and Krasnopolsky (2001), Krasnopolsky and Schiller (2003)). However, the methodology is novel for data assimilation. Its basic principle is visualized in Fig. 1.

We use feedforward backpropagation networks. Their characteristics will be described briefly, more details can be found in e.g. Bishop (1995) and Haykin (1999). The NNs are organized in layers: one input layer, one output layer and one or more hidden layers in between. Each layer consists of neurons. The number of neurons in the input and output layer are given by the number of their variables. The number of neurons in the hidden layer(s) is problem specific and its fixation needs some experience. Each neuron in a layer is linked to each neuron in a neighboring layer by a weight.

The NNs work sequentially: each element of the input vector serves as entry for one of the neurons of the input layer. The output of the first hidden layer is computed by summation of the weighted inputs, shifting it by a bias and applying a nonlinear function (a sigmoid here). The procedure repeats until the output layer is reached where the outcome of each neuron gives one element of the output vector.

Weights and biases are the free parameters of the network. They are fixed during the training phase of the NN by supplying a training dataset consisting of pairs of input vectors and corresponding desired output vectors. At the beginning of the training, the outcome of the NN will differ largely from the desired output. The mean squared relative error per neuron is iteratively minimized during the training by backpropagating it through the NN and adjusting the biases and weights according to a gradient descent scheme. This back propagation of errors is also exploited in the Levenberg–Marquardt algorithm: the first guess model output data $\vec{r}_0$ are adapted to better suit the measurements $\vec{r}_{\rm m}$.

It is good practice to have an additional independent testing dataset to check the generalization–power of the NN after the training, i.e. to test if reasonable output is produced for input not included in the training.

The training and testing phase of the NN methodology is time consuming. However, it needs to be done only once, whereas the subsequent usage of a NN is very fast.

3. Model Set-up

Within COSYNA WAM Cycle 4.5.4 runs pre–operationally twice a day with a three day forecast period. The model is an update