Neural network parameterisation of the mapping of wave–spectra onto nonlinear four–wave interactions

Kathrin Wahle * , Heinz Günther, Helmut Schiller

GKSS Research Center, Max–Planck–Str. 1, 21502 Geesthacht, Germany

Abstract

A new approach to parameterise the exact nonlinear interaction source (snl) term for wind wave spectra is presented. Discrete wave spectra are directly mapped onto the corresponding snl–terms using a neural net (NN). The NN was trained with modeled wave spectra varying from single mode spectra to highly complex ones. The specification of training data was based on a classification of the wave spectra by cluster analysis. In course of the structuring of the NN the intrinsic dimensionalty of the spectra was estimated with an auto-associative neural net (AANN). The AANN might be used for a scope check of the method.

Key words: Neural networks, Nonlinear wave–wave interaction, Wave–spectra, Wave models, Cluster analysis

1 Introduction

Wave–models compute the dynamics of the sea–surface by solving the action balance equation describing the evolution of interacting weakly–nonlinear waves, *e.g.* Komen et al. (1994). Besides propagation of waves this involves growth of waves due to wind, the dissipation of wave energy due to white-capping and the transfer of wave energy due to nonlinear wave–wave interaction. The computation of the last–mentioned process is by far the most time consuming step in the modelling. Operational wave–models therefore use approximations as the well known discrete interaction approximation (DIA) Hasselmann et al. (1985). The DIA is computational fast but has also known deficiencies.

* Corresponding authors.
Email address: kathrin.wahle@gkss.de (Kathrin Wahle).

The further development of third generation operational wave–models requires a replacement of the approximative methods for the calculation of the nonlinear wave-wave interaction by fast and accurate methods. Attempts have been made with extensions of the DIA but they are so far not generally applicable since tuning for different classes of wave spectra is needed. The same problem occurs for methods based on diffusion operators. For a comprehensive discussion see Cavaleri et al. (2007).

In this paper a neural net (NN) parametrisation of the mapping between wave–spectra and the corresponding nonlinear wave–wave interaction will be presented. The idea of applying neural networks in this context was first introduced by Krasnopolsky et al. (2001), Krasnopolsky et al. (2002). There the wave spectra as well as the nonlinear interaction source terms are assumed to be separable functions of frequency and direction which are approximated by expansion series, respectively. The neural network is used to map the two sets of expansion coefficients. This assumption was dropped in a successive work Tolman et al. (2005) where the authors used twodimensional Empirical Orthogonal Functions for the expansion of single peaked spectra. In contrast to this approach the feasibility of the direct mapping of the discretised wave spectra onto the nonlinear interaction source term will be demonstrated here. The study will not be restricted to single peaked spectra but will also include multi–modal ones.

Two design decisions are crucial for the successfull construction of a NN: the choice of the training data and of the NN architecture, respectively.

The wave spectra used for the training of the neural network (NN) are simulated spectra from a hindcast with the wave model WAM cycle 4 WAMDI group (1988). Although WAM uses the DIA the simulated spectra are suited to investigate the approach of a direct mapping. (Once the feasability has been shown one should make the effort of calculating all spectra with a wave model that uses the exact nonlinear interaction.) We expect that the training for wave spectra representing multi–modal wave systems will be more difficult then for single peaked spectra. Therefore the more complex cases should be well represented. To be able to enrich the number of multi–modal spectra in the training data set we perform a classification of the wave spectra. For the classification we use a cluster algorithm.

To fix the NN structure it is important to have an estimate of the intrinsic dimensionality of the spectra, *i.e.* the minimum number of variables needed to represent the spectra. For this we constructed auto-associative neural nets (AANN) which map the spectra onto themselves while compressing them in between. The difference of the original spectrum with the reproduced one gives a quality measure of the reduced representation. This difference can be used to decide wether the compression was successfull for a given spectrum (wether

it is in the scope of the algorithm). The finding of the intrinsic dimensionality led to the decision about the actual NN structure for the mapping of the wave spectra onto the nonlinear interaction source terms.

The results of the cluster analysis are given in section 2 whereas the technical details of the method are described in appendix A. Section 3 starts with a short introduction to neural nets and the special case of auto-associative neural nets. The concrete applications follow: in 3.1 some AANN architectures are tested in order to get an estimate of the intrinsic dimensionality of wave spectra. The design and training of the actual neural network for the nonlinear interaction source term is described in section 3.2. It includes the application of the AANN for outlier identification. We then discuss the relevant steps towards an operational usage of the procedure in an outlook given in section 4. Finally, the results are summarised in some concluding remarks in section 5.

2 The cluster analysis of wave–spectra

The wave spectra used for the training of the neural network (NN) are simulated spectra from a hindcast with the wave model WAM cycle 4 WAMDI group (1988). The data span over 22,000 points in the north-atlantic for the one month period of january 1995. Spectra were saved every three hours at an equidistant resolution of 15° in direction and an equidistant relative resolution in frequency ranging from approximately 0.04 Hz to 0.4 Hz. The resultant dimensionality of the spectra is 600 (24x25).

From this dataset of over five million spectra a representative subset containing all classes of spectra had to be selected. By random sampling the simple single– and bi–modal wave system cases would outweight any other class of spectra. As a result the neural networks would well map these cases but would badly perform for other classes. Thus a classification method should first of all be applied to the whole dataset.

Important characteristics for the classification of the wave spectra are the number and shapes of wave systems in them. The classification method should identify these wave systems. This clearly is a pattern recognition problem which we solved by applying a cluster algorithm.

As the spectra are represented using an azimuthal angle it happens that a wave system crossing 0° /360° is splitted artifically. To avoid this the wave spectra are preprocessed: the point of the origin of the direction is shifted if necessary to the direction with lowest energy. This is the only preprocessing needed here. Tolman et al. (2005) showed possiblities how to normalize the spectra further for other approaches.

Input to the cluster algorithm are the 600 points of a wave spectrum. Each of these points is defined by its coordinates (frequency and direction indices) and by the energy at this coordinate bin of the spectrum. The energies serve as weights of the points. The algorithm assigns the points into groups (called clusters) so that points from the same cluster are more similar (according to a distance measure) to each other than points from different clusters.

The cluster algorithm CLUCOV Schiller (1980) is used which was adapted for the twodimensional problem of parametrising the wave spectrum. Each cluster is characterised by a twodimensional Gaussians. For each cluster its respective center and covariance matrix are estimated and the resulting Mahalanobis distance is used to assign points to the closest cluster. This allows for varying orientations of the different clusters. In the iteration procedure it is checked if the clusters are unimodal. If not, the cluster is split into two. The number of clusters also can decrease: if two clusters strongly overlap these clusters are combined. In the appendix A the algorithm is described in more detail. One can summarize the method as performing a point density function approximation with Gaussians as radial basis functions (RBF).

Figure A.1 shows three examples. The left panel shows the input to the algorithm (original wave spectra). The right panel shows its parametrisation by the cluster algorithmus. It can be seen that the main characteristics are well reproduced. But also it should be noted that the number of clusters is often greater then the number of wave systems. In the first example the number of wave systems is two, but three clusters were found, the second example shows three wave systems and four clusters and the last spectrum has at least three wave systems in it and six clusters were found. This is due to the structure of the wave systems which are best approximated by two Gaussians, one around the peak and one for the tail (see again figure A.1).

To summarize the cluster algorithm is well suited to classify wave spectra. The main characteristics are well reproduced. The number of clusters is often larger than the number of wave systems in a spectrum but the two numbers are highly correlated. High number of clusters indicate complex structures in the corresponding spectrum.

Table 1 lists the result of the cluster analysis, *i.e.* the number of clusters found for each spectrum in the dataset. On this basis a representative subset of 791,570 spectra was choosen. The number of selected spectra increases with the number of clusters (more complex wave spectra). Above a number of clusters of five all spectra were taken and included twice into the dataset. Table 1 also quantifies the selection. This representative dataset was randomly subdevided in a training set (755,173 spectra) and a testing set (36,397 spectra) for the neural networks.

number of clusters	number of spectra	percentage selected
1	406,332	9.8
2	1,536,057	5.2
3	1,907,143	6.3
4	1,033,269	15.5
5	354,712	56.4
6–12	95,804	200

Table 1
Spectra used for the representative dataset.

3 Neural Networks

Wave spectra and the nonlinear interaction source term are related via a six–dimensional Boltzmann integral. A computational efficient method to parametrise this complex functional relation is the usage of artificial neural networks (NN). This is possible since a NN with at least one hidden layer – a layer between the input and the output layer – is able to approximate any continuous function (Universal Approximation Theorem) as was shown by Haykin (1999).

So a NN — in this context — is a computational tool for function approximation whose generalities will be described briefly (see Bishop (1995)).

A NN is organized in layers: one input layer, one output layer and one or more hidden layers inbetween. Each layer consists of neurons. The number of neurons in the in– and output layer are given by the dimensions of the in– and output vectors. (In our application the two–dimensional wave spectra and nonlinear interaction source terms will therefore be arranged as vectors with 600 components, respectively.) The number of neurons in the hidden layer(s) are not preset and depend upon the problem. Each neuron in a layer is linked to each neuron in a neighboring layer with a weight.

Neural nets 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 is repeated itself 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 approximation. They are fixed during the training phase of the NN. To do so a dataset — the training set — is needed consisting of N pairs of input vectors (the 600 components of the wave spectra) and corresponding desired output vectors (the correspond-

ing 600 components of the exact nonlinear interaction source term). At the beginning of the training the outcome of the NN will differ largely from the desired output. Let $\vec{y}$ and $\vec{y'}$ be the m–dimensional output vectors as emulated by the NN and as given in the dataset respectively. Then the difference of the two will result in an error e defined by

$$
\begin{aligned}
e &= \frac{1}{N} \sum_{p=1}^{N} e_p \\
e_p &= \frac{1}{m} \sum_{i=1}^{m} \left(\frac{{y'}_i - y_j}{\max({y'}_i^p, p \in [1,N]) - \min({y'}_i^p, p \in [1,N])} \right)^2 .
\end{aligned} \tag{1}
$$

This mean squared relative error per neuron e is iteratively minimized during the training by backpropagating it through the NN and adjusting the biases and weights according to a gradient descent scheme. It is measure for the quality of the function approximation by the NN.

It is good practise to have an additional independent dataset — the testing set — which is needed 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 is time consuming. But it needs to be done only once. Whereas the usage of a NN is very fast.

The abovementioned holds true for an auto–associative NN which is a particular NN. Its especialness is that it maps the input vector onto itself. But at one stage of the mapping — in the so called bottleneck layer — the dimensionality is reduced. So when applying AANN's with different degrees of dimensionality reduction to the 600 components of the wave spectra the number of variables needed to capture the information contained in the spectra can be determined.

Technically speaking the number of neurons in at least one hidden layer in an AANN is less than the dimension of the in- and output vector $\vec{x}$ and $\vec{x'}$. The AANN part $m(\vec{x})$ which maps the input vector onto the neurons in the bottleneck layer $\vec{p}$ is also called mapping part and the backprojection $\vec{x'} = d(\vec{p})$ is called demapping part, accordingly (see figure A.2).

The mapping part can be considered as performing a nonlinear generalization of the Principal Component Analysis (Nonlinear Principal Component Analysis, NLPCA) Kramer (1991). It retains the maximum possible amount of information from the original data, for a given degree of compression. Like PCA also NLPCA can serve important purposes, *e.g.* filtering noisy data, feature extraction, outlier identification, the compression of data, and the restauration of missing values Hsieh (2001), Schiller (2003).

3.1 Auto–associative neural network of wave–spectra

In order to achieve the goal of directly mapping the wave spectra onto the corresponding nonlinear interaction source term it is essential to find their intrinsic dimensionality (*the number of independent variables*) to be able to fix an appropriate NN structure.

The cluster analysis in the last section was not only useful for the creation of a representtive dataset but also it gives a hint about the intrinsic dimensionality of the wave spectra. This can be reasoned since the parametrisation of the wave spectra by gaussians is a first compressed representation. Since each spectrum can be parametrised by a few clusters and each cluster is characterised by six numbers (energy of the cluster, its center, its orientation and shape) the intrinsic dimensionality is expected to be of the order of a few tens. This will now be further investigated by means of auto–associative neural networks (AANN's).

To find an optimal AANN–parametrisation of the wave spectra AANN's with different number of neurons in the bottleneck layer had to be trained. The optimization problem consists of balancing the loss of information with decreasing bottleneck neuron number and the usability of the NN. At optimum the compression is as loss–free as possible with additional bottleneck neurons giving only slight improvement.

For the training the neural net program developed by Schiller (2000) was used. The AANN's have 600 neurons in the in– and output layer since the wave spectrum is interpreted as a 600-dimensional vector. Each of the trained AANN's had three hidden layers with 80 neurons in the first and third hidden layer and with differing number of neurons in the middle bottleneck layer. Figure A.3 shows the mean squared relative error per neuron e (see equation 1) of the AANN's for the testing data set as a function of the number of bottleneck neurons.

As expected the error e decreases with increasing number of bottleneck neurons. Clearly two different regimes are visible (the straight lines were fitted by linear regression): one for the number of bottleneck neurons below approximately 33 and one above. From this it was decided to choose the smallest AANN belonging to the second regime with 39 bottleneck neurons in it.

Figure A.4 shows three examples of the performance of this AANN. The left panel shows the original wave spectra. The middle panel shows the output of the AANN — the mapping of the original wave spectra onto themselves. The right panel shows the directionally integrated wave spectra. The examples exhibit an increasing complexity of the wave spectra. The AANN output strongly resembles the original wave spectra throughout the spectral space.

The residual average error $\sqrt{e}$ is 1.4, 1.5 and 1.8% from top to bottom.

To summarize the construction of different AANN's has shown that the intrinsic dimensionality of the analysed wave spectra is about 40. An AANN with 39 bottleneck neurons gives a good parametrisation for all different classes of wave spectra with a residual average error of $\sqrt{e}$ =1.3%.

3.2 Nonlinear wave–wave interaction

So far we have selected a dataset which incorporates all classes of wave spectra (see section 2) and we have found the typical number of independent variables describing each of these wave spectra (see section 3.1).

After these preparatory works one can now start with the construction of a neural net (NN) for the direct derivation of the nonlinear interaction source terms from wave spectra. As a first approximation we decided to use the same architecture already used for the AANN: In– and output layers consist of 600 neurons as the discretized wave spectra and the snl–terms are interpreted as 600–dimensional vectors. Inbetween there are again three hidden layers with 80, 39 and 80 neurons, respectively.

To train and test the NN the (exact) nonlinear interaction source (snl) terms corresponding to each of the spectra of the selected dataset of 791,570 wave spectra (see section 2) had to be calculated first. The method first suggested by Webb (1978) and known as the WRT method with further improvements by Vledder (2006) was applied for this purpose.

The pairs of wave spectra and snl–terms were again subdivided into the training and testing dataset. The resulting residual average error $\sqrt{e}$ (see equation 1) of the resulting NN was 1.1% for both the testing– and training.

Figure A.5 shows results of the performance of the NN. The top row shows the original wave spectra which served as input for the NN. The next row shows the corresponding exact snl–terms calculated with the WRT–method and below it the NN results (the WRT emulation by the NN) are shown. The bottom row shows the directionally integrated snl–terms. In all cases the NN emulation is very similar to the exact solution. The directionally integrated plots highlight the quality of the fit. The three wave spectra vary from single to three wave systems to demonstrate the broad applicability of the method to different classes of wave spectra.

The choice of a represantative training dataset is crucial since a NN has good interpolation properties but produces unpredictable output when forced to extrapolate. The approach described here offers a 'in range' check Schiller et

al. (2001), *i.e.* the AANN can be used to check if a given spectrum belongs to the class of spectra chosen for the training of the AANN. If the error e of the AANN is big for a given wave spectrum then it did not succeed to represent the spectrum through the bottleneck layer. Then it is questionable that the outcome of the snl–term NN will be in good agreement with the exaxt snl–term.

Figure A.6 exemplifies the resulting outlier identification: A wave spectrum which gave a large of error (here $\sqrt{e} = 3.6\%$) when parametrising it with the AANN was choosen randomly (upper panel). The emulation of the corresponding snl–term with the NN is poorly (lower panel). The resultend relative error when comparing the exact with the NN solution is $\sqrt{e} = 10.2\%$.

The NN method described here differs in mainly two points from the NN based method first presented in Krasnopolsky et al. (2001) and with latest results from Tolman et al. (2005): There the wave spectra as well as the snl-terms are first decomposed and represented by a linear superposition of two sets of orthogonal functions. Two different approaches are used for the basis functions. The former approach is based on a mathematical basis where the frequency and direction dependence was separated (an assumption beeing not satisfied by spectra having more than one wave system at one frequency). The later one is based on principal components but it is restricted to single peaked spectra. In either case the neural net is used to map the expansion coefficients onto each other. The training and testing sets used were generated based on theoretical spectral descriptions.

So firstly, the approach presented here is not restricted to special cases of wave spectra with e.g. only one wave system in it since a set of modeled spectra of a high variety of complexity has been used for its development. And secondly, no assumptions (as separabilty or convergence of an expansion series) about the wave spectra and nonlinear interaction terms are made.

Finally an important advantage of any NN parametrisation over other implementations of complex physical processes is its computational efficiency. It is very time consuming to train a NN and to find an appropriate net architecture but its usage is fast. A runtime comparison of the exact WRT–method with the NN method presented here gives a speedup factor of roughly 500.

4 Outlook

We have presented results of a neural net (NN) mapping directly wave spectra onto the corresponding exact nonlinear interaction source term. We now want to describe how we imagine to distinctly improve the method in order to

operationally incorporating it.

To start with, a new training/testing dataset of wave spectra should be calculated with a model that uses the exact nonlinear interaction. The datasets should again contain all classes of complexity of wave spectra.

With these datasets the training of the neural nets should be repeated and different net architectures (e.g. different mapping/demapping part) should be tried. In particular we assume it promising to train a AANN for both, the wave spectra and the nonlinear interaction source terms and to map the both bottleneck layers onto each other. This *ansatz* has some advantages: Firstly, more complex net architectures can be tried out for the NN which maps the two bottleneck layers onto eachother since the number of in– and output variables is one order of magnitude smaller than in the present NN. Secondly, with two AANN's the 'in range' check can be performed twice: one time for the wave spectra before applying the actual NN and one time afterwards for the emulated nonlinear interaction. Still, this is a necessary condition but not a sufficient one. Thirdly, when changing the resolution of the wave model only the two AANN's have to be retrained, the NN for the actual mapping stays the same. This can be easily done, because the number of neurons in the bottleneck layers of the AANN's is already known since the complexity of the spectra and the source terms does not change with resolution.

Finally, it has to be shown that the NN emulation gives robust and accurate results when implemented in a numerical wave model. As suggested by Krasnopolsky et al. (2005), Krasnopolsky et al. (2007) parallel runs of the wave model with the original parametrization of the exact nonlinear interaction and with its NN emulation should be performed to do so. When operationally running the wave model a quality control block as suggested by Krasnopolsky et al. (2008) could determine whether the NN emulation will be used or not.

5 Conclusions

The feasability of a neural network (NN) based method for direct mapping of discrete wave spectra onto the corresponding (exact) nonlinear wave–wave interaction source (snl) terms has been demonstrated.

In a preparatory step the intrinsic dimensionalty of the wave spectra was estimated by means of an AANN to be about 40, *i.e.* 40 variables allow to capture the variability of the wave spectra with a residual average error of 1.3%. The composition of the training data set for the NN was based on automated classification of the wave spectra by means of a cluster analysis.

The NN for the mapping of the wave spectra onto the corresponding snl–terms shows good performance. It is able to emulate the WRT method calculations for single and multi mode wave spectra with a much higher accuracy then the approximations implemented in nowadays operational wave models. The quality of the emulation might be controlled using the corresponding AANN.

6 Acknowledgements

The authors like to thank Gerbrant Ph. van Vledder for providing his improved version of the WRT–method for calculating the nonlinear interaction source terms and for offering his help on running it. Further, we thank the reviewers for their valuable comments on an earlier version of this paper.

A The cluster algorithm CLUCOV: twodimensional case

The CLUCOV algorithm is used assign the 600 points of a wave spectrum to clusters. The k–th cluster G^k is characterized by the moments of order zero, one and two of the distribution of the N points contained in this cluster. The points are defined by their frequency and direction coordinates $X^m = (X_f^m, X_\theta^m)$ and their energies which serve as weights w^m. Then the three moments are

– the total weight I^k of points X^m contained in the cluster G^k

$$I^k = \sum_{m=1}^{N^k} w^m$$

– the centroid Q^k of the cluster G^k

$$Q_i^k = \frac{1}{I^k} \sum_{m=1}^{N^k} w^m X_i^m \qquad \text{with } X^m \in G^k, \quad i \in \{f, \theta\}$$

– the covariance matrix C^k of the cluster G^k

$$C_{ij}^k = \frac{1}{I^k} \sum_{m=1}^{N^k} w^m (X_i^m - Q_i^k)(X_j^m - Q_j^k) \qquad \text{with } X^m \in G^k, \quad i, j \in \{f, \theta\}$$

The eigenvectors of the covariance matrix point into the direction of the main axes of the ellipsoids by which the shape of the clusters is approximated. The square root of the eigenvalues of the covariance matrix denote the lengths of the main axes, and the square root of the determinant of the covariance matrix thus measures the volume of the clusters.

All these three moments enter the definition of the distance f_m^k of a point m at X^m from the k–th cluster G^k. Thus, each cluster builds its 'own' metric: The (Mahalanobis) distance ρ_m^k of a given point X^m from the centroid Q^k of the k–th cluster enters the exponent of a Gaussian $g^k(X^m)$

$$g^k(X^m) = \frac{1}{2\pi\sqrt{|C^k|}} \exp[-\frac{1}{2}(\rho_m^k)^2] \quad \text{with}$$
$$\rho_m^k = \sqrt{(X^m - Q^k)^T (C^k)^{-1} (X^m - Q^k)}.$$

The total weight I^k of points in cluster k is included into the distance measure f_m^k as a linear weight factor, so that big clusters 'attract' further points

$$f_m^k = I^k \cdot g^k(X^m).$$

In the direction of the main axes distances are measured in units of the square root of the corresponding eigenvalue of the covariance matrix (which is suggested by the quadratic form in the exponent of the gaussian). This distance measure is also invariant under linear transformations such as translation and rotation.

The determinant in the denominator of f_m^k favours compact clusters against voluminous clusters of the same content.

In the algorithm CLUCOV it is also possible to split and merge clusters, that is to change the number of clusters.

To achieve this, a measure t for the overlap of two clusters G^k and G^l is defined:

$$t = \frac{h_0}{\sqrt{h_k h_l}}$$

The quantities h_k and h_l are superpositions of the gaussians $f^k(X)$ and $f^l(X)$ of the clusters G^k and G^l in their centroids Q^k and Q^l:

$$h_k = f^k(Q^k) + f^l(Q^k) \quad \text{and} \quad h_l = f^k(Q^l) + f^l(Q^l)$$

The quantity h_0 is the minimum of this superposition along the distance vector $(Q^k - Q^l)$ between the clusters:

$$h_0 = min[f^k(X) + f^l(X)]$$

If for a pair of groups this measure t exceeds a limit t_{merge} (parameter of the merging procedure) this pair of groups is united into one group.

The compactness of the clusters is tested by arbitrarily subdividing the clusters with straight lines through the cluster. If the overlap of the two parts of the clusters is smaller than t_{split} (parameter of the splitting procedure) this cluster is split by the corresponding line.

References

G.J. Komen, L. Cavaleri, M. Donelan, K. Hasselmann, S. Hasselmann, P.A.E.M. Janssen (1994) Dynamics and Modelling of Ocean Waves. Cambridge University Press,

S. Hasselmann, K. Hasselmann, J.A. Allender, T.P. Barnett (1985) Computations and parametrizations of the non–linear energy transfer in a gravity wave spectrum. Part II: Parametrization of the non–linear transfer for application in wave models. Journal of Physical Oceanography 15, p.1378–1391

The Wise Group, L. Cavaleri, J.-H.G.M. Alves, F. Ardhuin, A. Babanin, M. Banner, K. Belibassakis, M. Benoit, M. Donelan, J. Groeneweg, T.H.C. Herbers, P. Hwang, P.A.E.M. Janssen, T. Janssen, I.V. Lavrenov, R. Magne, J. Monbaliu, M. Onorato, V. Polnikov, D. Resio, W.E. Rogers, A. Sheremet, J. McKee Smith, H.L. Tolman, G. van Vledder, J. Wolf, I. Young (2007) Wave modelling The state of the art. Progress in Oceanography 75(4), p. 603–674

V.M. Krasnopolsky, D.V. Chalikov, H.L. Tolman (2001) Using neural network for parametrization of nonlinear interactions in wind wave models. In: International Joint Conference on Neural Networks, 15–19 July, 2001, Washington DC, p.1421–1425

V.M. Krasnopolsky, D.V. Chalikov, H.L. Tolman (2002) A Neural Network Technique to Improve Computational Efficiency of Numerical Oceanic Models. Ocean Modelling 4, p.363–383

H.L. Tolman, V.M. Krasnopolsky, D.V. Chalikov (2005) Neural network approximation for nonlinear interactions in wave spectra: direct mapping for wind seas in deep water. Ocean Modelling 8, p.253–278

WAMDI group (1988) The WAM model — a third generation ocean wave prediction model. Journal of Physical Oceanography 18, p.1775–1809

H. Schiller (1980) Cluster Analysis of Multiparticle Final States for Medium–Energy Reactions. Fizika elementarnyk chastits i atomnogo yadra 11, p.182–235, translated in Soviet Journal of Particles and Nuclei 11(1), p.71–90

S. Haykin (1999) Neural Networks — A comprehensive Foundation. Prentice Hall International, Inc.

C.M. Bishop (1995) Neural Networks for Pattern Recognition. Clarendon Press, Oxford

M.A. Kramer (1991) Nonlinear principal component analysis using autoassociative neural networks. American Institute of Chemical Engineers Journal 37(2), p.233–243

W.W. Hsieh (2001) Nonlinear principle component analysis by neural networks. Tellus A 53(5), p.599–615

H. Schiller (2003) Neural Net Architectures for Scope Check and Monitoring. CIMSA 2003, Lugano, Switzerland, 29-31 July 2003, ISBN 0-7803-7784-2

H. Schiller (2000) Feedforward–backpropagation neural net program ffbp1.0. GKSS–report 2000/37, ISSN 0344–9629

D.J. Webb (1978) Non–linear transfers between sea waves. Deep Sea Research 25(3), p.279–298

G.Ph. van Vledder (2006) The WRT method for the computation of non–linear four–wave interactions in discrete spectral wave models. Coastal Engeneering 53, p.223–242

H. Schiller, V.M. Krasnopolsky (2001) Domain Check for Input to NN Emulating an Inverse Model. International Joint Conference on Neural Networks, 15-19 July 2001, Washington DC, p.2150–2152, ISBN 0-7803-7044-9

V.M. Krasnopolsky, M.S. Fox-Rabinovitz, D.V. Chalikov (2005) New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Long Wave Radiation in a Climate Model. Monthly Weather Review 133(5), p.1370–1383

V.M. Krasnopolsky (2007) Neural Network Emulations for Complex Multidimensional Geophysical Mappings: Applications of Neural Network Techniques to Atmospheric and Oceanic Satellite Retrievals and Numerical Modeling. Reviews of Geophysics 45, RG3009

V.M. Krasnopolsky, M.S. Fox-Rabinovitz, H.L. Tolman, A.A. Belochitski (2008) Neural network approach for robust and fast calculation of physical processes in numerical environmental models: Compound parameterization with a quality control of larger errors. Neural Networks 21, p.535–543

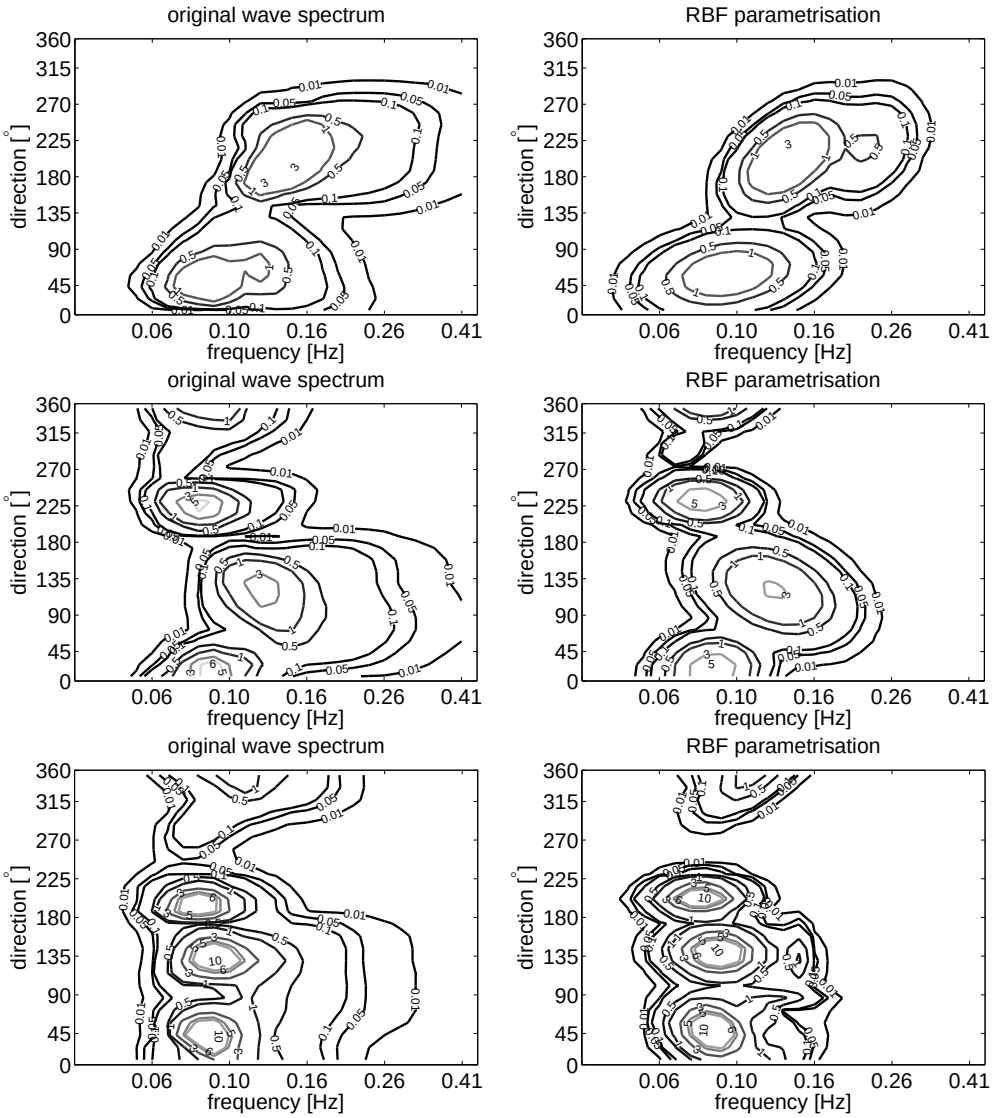

Fig. A.1. Examples of cluster analysis: left panel shows original wave spectra, right panel shows corresponding RBF parametrisation.

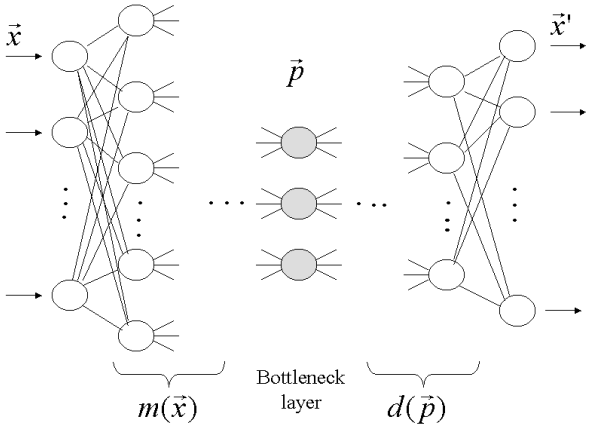

Fig. A.2. Autoassociative NN with bottleneck: the input is mapped by $\vec{p} = m(\vec{x})$ onto a lower dimensional ($dim(\vec{p}) < dim(\vec{x})$) space and is approximately reconstructed by the demapping NN part $\vec{x'} = d(\vec{p}) \approx \vec{x}$.

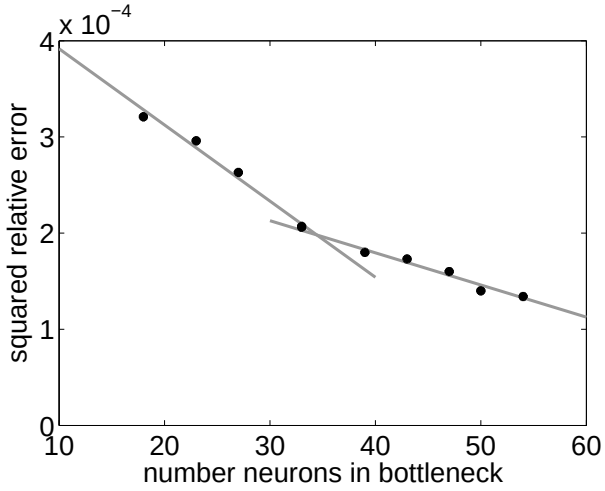

Fig. A.3. Squared relative error of the AANN's as function of the number of bottleneck neurons.

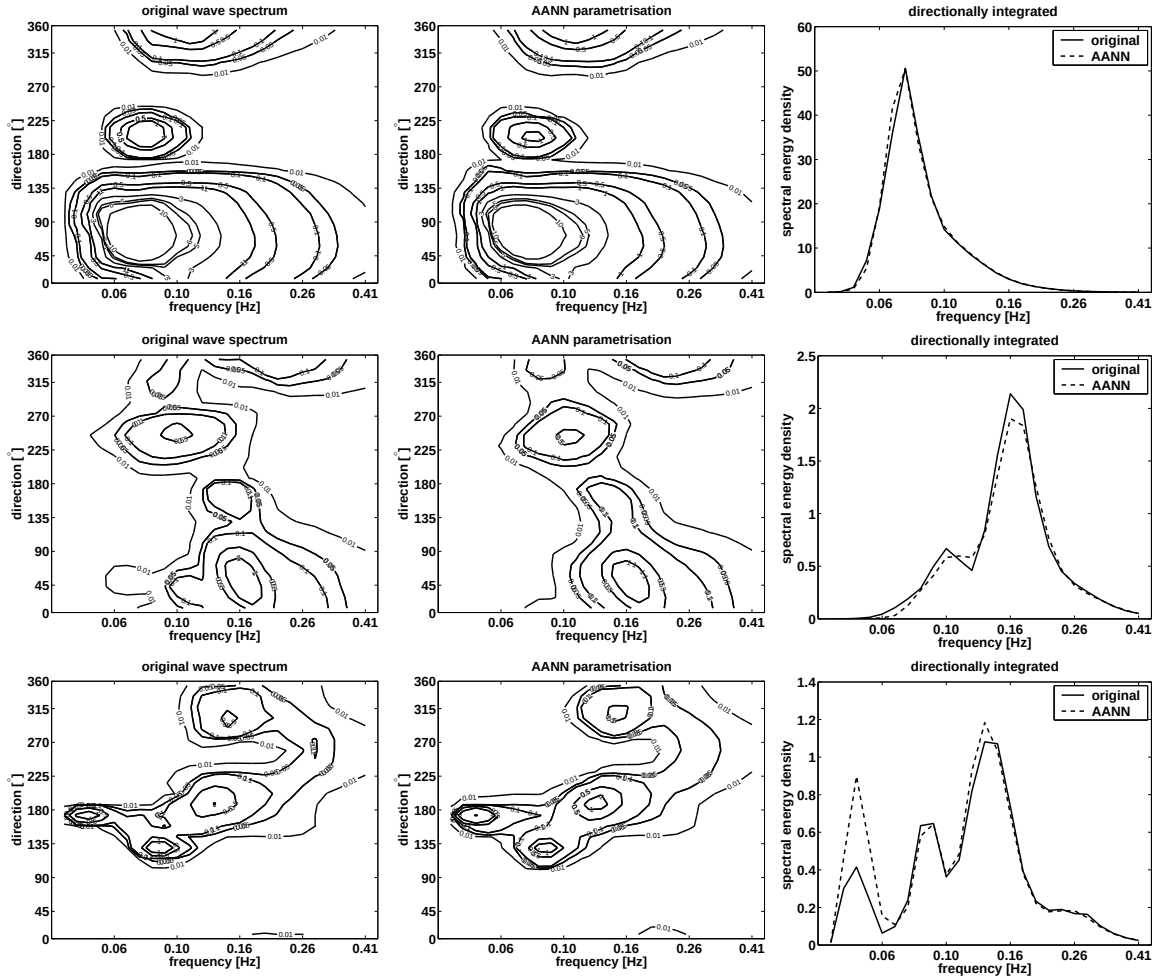

Fig. A.4. Examples of the performance of the AANN: left panel shows original wave spectra, panel in the middle shows corresponding AANN parametrisation, and right panel shows the directionally integrated spectra.

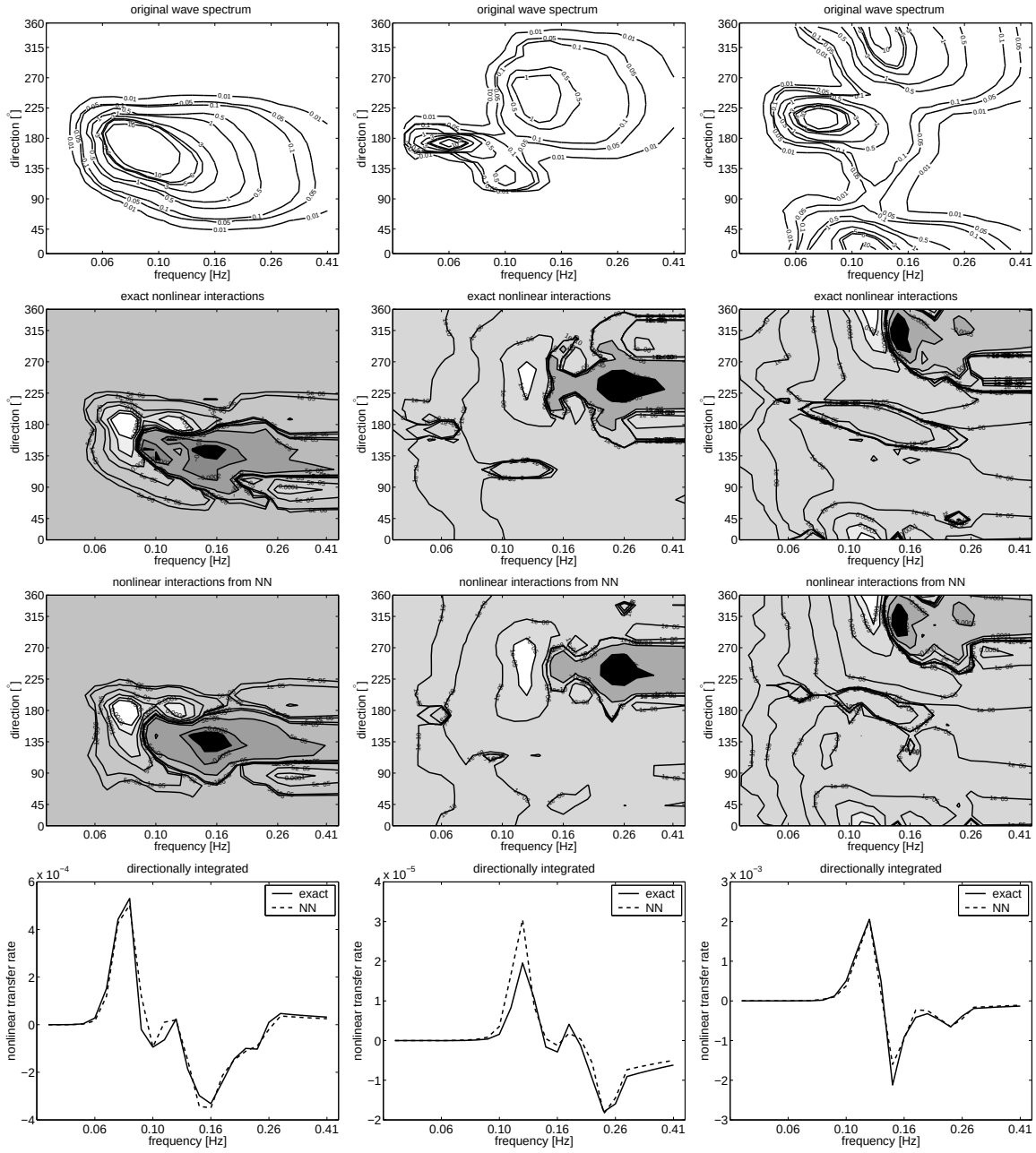

Fig. A.5. Examples of the performance of the NN for emulating the WRT method: upper row shows original wave spectra, next row the corresponding exact snl–term and third row shows its emulation by the NN. In the plots of the snl–term the black areas correspond to most negative values and the white areas to the biggest positive values. The gray areas inbetween cover ranges of four orders of magnitude for the snl–terms for either sign. The lower row shows the directionally integrated snl–terms.

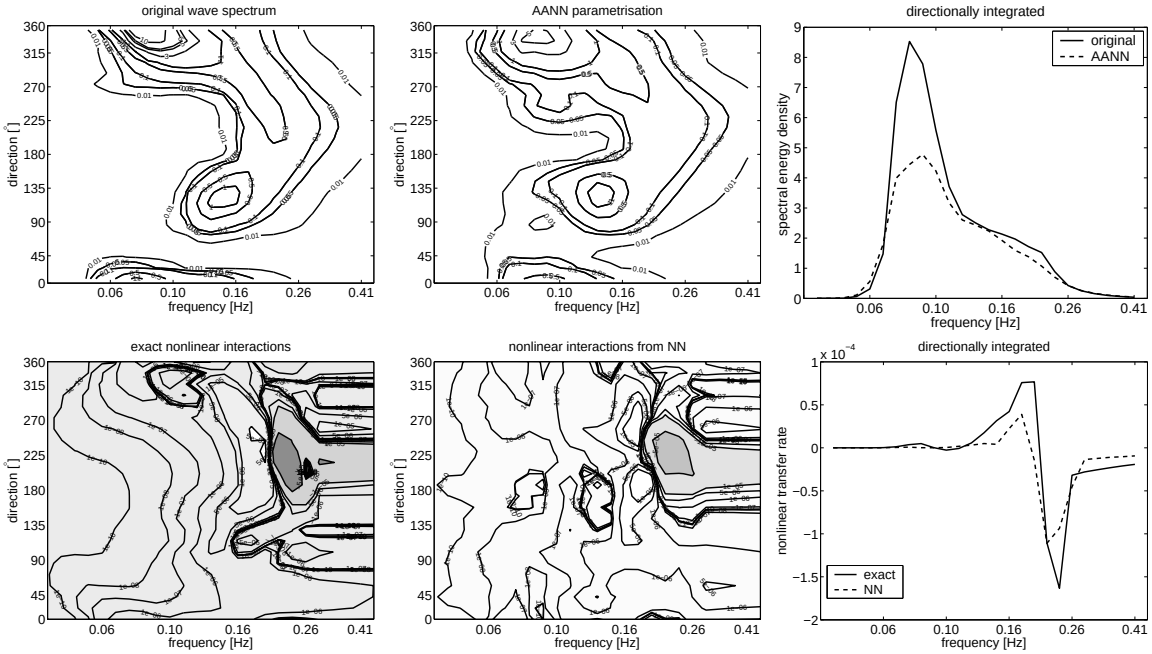

Fig. A.6. The upper panel shows the orinal wave spectra and the output from the AANN for a case where the error of the AANN is large. The lower panel shows the corresponding exact and NN emulated snl–term (contours as in figure A.5).