Global crustal thickness from neural network inversion of surface wave data

Geophys. J. Int. (2007) **169,** 706–722 doi: 10.1111/j.1365-246X.2007.03373.x

Global crustal thickness from neural network inversion of surface wave data

Ueli Meier,[1] Andrew Curtis[2,*] and Jeannot Trampert[1]

[1] *Utrecht University, Department of Earth Sciences, Budapestlaan* 4, 3584 *CD Utrecht, the Netherlands. E-mail: meierue@geo.uu.nl*
[2] *The University of Edinburgh, School of GeoSciences, Grant Institute, West Mains Road, Edinburgh,* EH9 3JW, *UK.*

Accepted 2007 January 24. Received 2006 November 9; in original form 2006 August 11

SUMMARY

We present a neural network approach to invert surface wave data for a global model of crustal thickness with corresponding uncertainties. We model the *a posteriori* probability distribution of Moho depth as a mixture of Gaussians and let the various parameters of the mixture model be given by the outputs of a conventional neural network. We show how such a network can be trained on a set of random samples to give a continuous approximation to the inverse relation in a compact and computationally efficient form. The trained networks are applied to real data consisting of fundamental mode Love and Rayleigh phase and group velocity maps. For each inversion, performed on a $2^\circ \times 2^\circ$ grid globally, we obtain the *a posteriori* probability distribution of Moho depth. From this distribution any desired statistic such as mean and variance can be computed. The obtained results are compared with current knowledge of crustal structure. Generally our results are in good agreement with other crustal models. However in certain regions such as central Africa and the backarc of the Rocky Mountains we observe a thinner crust than the other models propose. We also see evidence for thickening of oceanic crust with increasing age. In applications, characterized by repeated inversion of similar data, the neural network approach proves to be very efficient. In particular, the speed of the individual inversions and the possibility of modelling the whole *a posteriori* probability distribution of the model parameters make neural networks a promising tool in seismic tomography.

Key words: crustal structure, inversion, Moho discontinuity, surface waves, tomography.

1 INTRODUCTION

Crustal structure is an important global characteristic, which varies greatly over small length scales and has significant effects on fundamental mode surface waves. In surface wave tomography it is, therefore, common practice to remove the crustal contributions to surface wave measurements by applying crustal corrections. The computation of crustal corrections is still an issue of ongoing research and is problematic as outlined by Zhou *et al.* (2005).

Whatever the approach to compute the crustal corrections, the accuracy of the crustal thickness model is crucial. Crustal thickness varies from 5 km beneath oceans to 80 km under continents. The most widely used global crustal model is CRUST2.0 (Bassin *et al.* 2000) an updated model of CRUST5.1 (Mooney *et al.* 1998). This model is based on refraction and reflection seismics as well as receiver function studies. As a consequence, resolution of CRUST2.0 is high in regions with good data coverage but in regions with poor or no data coverage crustal thickness estimates are largely extrapolated. For the purpose of applying crustal corrections to surface wave measurements this is far from ideal and it is desirable to have a global crustal thickness model with a resolution similar to the data sets used in surface wave tomography. To our knowledge no global crustal thickness model solely constrained by surface waves exists. This forms one of the key motivations of this study: to invert fundamental mode surface wave data for crustal thickness and to present a global crustal thickness model.

Phase and group velocity measurements of fundamental mode Rayleigh and Love waves are most commonly used to constrain shear-velocity structure in the crust and upper mantle on a global scale (e.g. Zhou *et al.* 2006) or on regional scale (e.g. Curtis & Woodhouse 1997; Curtis *et al.* 1998; Ritzwoller & Levshin 1998; Villaseñor *et al.* 2001). Only a few studies used surface wave data to infer Moho thickness directly (Devilee *et al.* 1999; Das & Nolet 2001) on a regional scale and Shapiro & Ritzwoller (2002) globally. The aim of this study is to investigate how well Moho depth can be retrieved from current phase (Trampert & Woodhouse 2003) and group velocity (Ritzwoller *et al.* 2002) maps without the use of restrictive *a priori* constraints.

Inverting phase and group velocities for discontinuities within the earth forms a non-linear inverse problem. Linearization techniques around a reference model fail because: (1) non-linearities are too strong (i.e. varying the depth of a discontinuity alters the structure of the whole earth), (2) large variations in the depth of discontinuities, Moho depth for example varies from 5 to 80 km, make it

*ECOSSE (Edinburgh Collaborative of Subsurface Science and Engineering).

difficult to choose a reasonable reference model and (3) uncertainties estimated using linearized methods are inaccurate in non-linear problems. Montagner & Jobert (1988) demonstrated that variations in Moho depth clearly have a non-linear effect on the resulting phase and group velocity perturbations, and they proposed to use three different crustal reference models to remove most of this non-linearity. However, several fully non-linear inversion methods are available among which the most common ones are sampling based techniques (e.g. Mosegaard & Tarantola 1995; Sambridge 1999a,b). We focus on neural networks instead to solve the non-linear inverse problem, inverting Moho depth from phase and group velocity measurements.

Neural networks have been widely used in different geophysical applications, a good overview is given by van der Baan & Jutten (2000). Neural network techniques have been successfully applied to logging problems (e.g. Benaouda *et al.* 1999; Aristodemou *et al.* 2005). Roth & Tarantola (1994) used a neural network to invert seismic reflection data for 1-D velocity models and Devilee *et al.* (1999) were the first to use a neural network to invert surface wave velocities for Eurasian crustal thickness in a fully non-linear and probabilistic manner. In various other fields neural networks were successfully used to solve inverse problems. Thodberg (1996) for example used a neural network to predict fat content in minced meat from near infrared spectra, Cornford *et al.* (1999) retrieved wind vectors from satellite scatterometer data, and Lampinen & Vehtari (2001) investigated the use of neural networks in electrical impedance tomography.

In the current study we further develop the methods of Devilee *et al.* (1999), then invert surface wave data for global crustal thickness on a $2^\circ \times 2^\circ$ grid globally using a neural network. We show that for this particular application where many repeated inversions are required, the presented neural network approach significantly outperforms conventional sampling based inversion techniques.

The neural network approach for solving inverse problems is best summarized by three major steps: (1) proceed by randomly sampling the model space and solve the forward problem for all visited models (i.e. compute phase and group velocities for the sampled radially symmetric earth models using normal mode theory). This results in a collection of earth models and corresponding phase and group velocities (called the training data set). (2) Design a neural network structure that can accept phase and group velocities as input and compute the earth model as output, then use the training data to train the network (i.e. change the parameters of the network such that the network output represents the desired output, the earth model). (3) Once the network is trained it represents the non-linear inverse mapping from phase and group velocities to earth structure. For any observed dispersion curve the trained network will give an output that is close to the 'real earth'. Since the inverse mapping of this particular problem is multivalued (i.e. there exist many models that could produce the same specific dispersion curve), we propose to model the posterior model parameter distribution rather than only its mean and variance.

In what follows we first give a short introduction to neural networks, we show how neural networks can be used to model posterior model parameter distributions in general, and how such networks can be used to invert dispersion curves for *a posteriori* Moho depth distribution in particular. A thorough analysis is presented on how regularization is needed to train a network on a synthetic data set that interpolates well with a real data set corrupted by noise. Finally we compare our global crustal model with two other global crustal thickness models, CRUST2.0 from Bassin *et al.* (2000) and the CUB2 model from Shapiro & Ritzwoller (2002), and discuss the observed features.

2 NEURAL NETWORKS

There is no precise agreed definition as to what a neural network is. Originally neural networks were intended as an abstract model of the brain, consisting of simple processing units—similar to neurons in the human brain—connected together to form a network. Obviously resemblance to a human brain is rather limited; therefore, we prefer to think of a neural network as a graphical notation of a mathematical model defining a mapping from an input to an output space. The basic idea behind neural networks is to represent a non-linear function of many variables in terms of a composition of multiple, relatively simple component functions of a single variable, the so-called activation functions. Common choices for activation functions include the logistic sigmoid or the hyperbolic tangent, which result in equivalent mappings since these two functions only differ through a linear transformation. In our simulations we use the latter since it is often found that the hyperbolic tangent functions give rise to faster convergence of training algorithms than logistic functions.

The network diagram of the neural network we consider in this work is shown in Fig. 1. This is an example of a two-layer, feed-forward neural network often called a multilayer Perceptron (MLP). There are two layers of adaptive parameters. Those of the first and second layer are called weight matrices $\dot{w}_{ij}$ and $\ddot{w}_{jk}$ as well as the biases of the hidden $\dot{b}_j$ and output $\ddot{b}_k$ units, respectively; information flows only in the forward direction from the input to the output units. The output of a MLP as illustrated in Fig. 1 for a given input vector **d** can be computed as follows:

$$y_k = \sum_j \ddot{w}_{jk} z_j + \ddot{b}_k,$$

where

$$z_j = \tanh\left(\sum_i \dot{w}_{ij} d_i + \dot{b}_j\right). \tag{1}$$

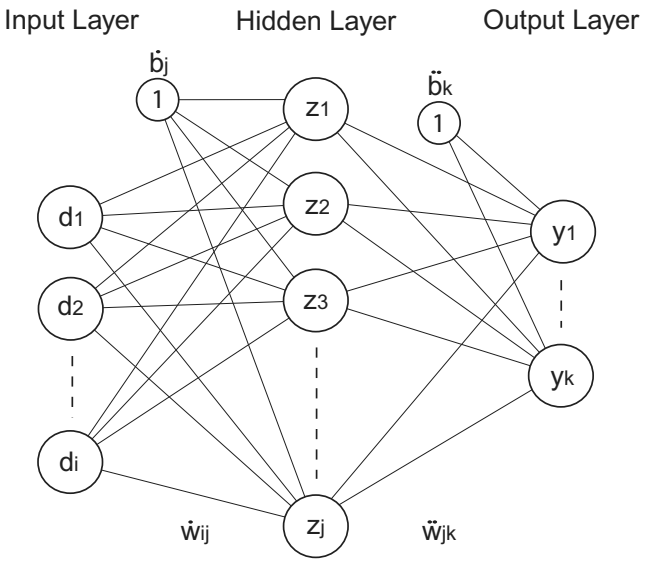

Figure 1. A multilayer Perceptron (MLP) with i input units, one hidden layer with j hidden units and k output units. There are two layers of adaptive parameters the first layer weights $\dot{w}_{ij}$ and bias $\dot{b}_j$ and the second layer weights $\ddot{w}_{jk}$ and bias $\ddot{b}_k$, indicated by the connections between the different units. The bias parameters $\dot{b}_j$ and $\ddot{b}_k$ are shown as weights from an extra input having a fixed value 1. For one specific input vector **d**, first the activations of the hidden units z_j are computed followed by the output values y_k.

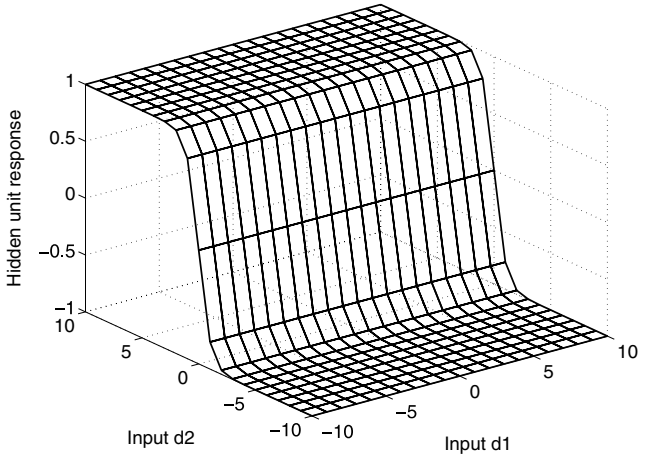

Figure 2. Output of a hidden unit as a function of its inputs. The first layer weight matrix is $\dot{w} = (0, 1)$ and the first layer bias term $\dot{b} = 0$.

Here $\dot{w}_{ij}$ is the weight on the connection between input i and hidden unit j, similarly $\ddot{w}_{jk}$ is the weight on the connection between hidden unit j and output k, while $\dot{b}_j$ and $\ddot{b}_k$ are biases of the hidden and output units, respectively. Note that it might be more convenient to put the first and second layer weight and bias terms into a single weight vector $\mathbf{w}$. Writing $y(\mathbf{d}; \mathbf{w})$ means that the network output y is a function of the input vector $\mathbf{d}$ and the network parameters $\mathbf{w}$, which we define to include the weight and bias terms of the first and second layer. A geometrical interpretation of the weight and bias terms in eq. (1) can be given by considering the output of each hidden unit z_j, the hyperbolic tangent, as a surface over the input space. Each hidden unit can then be regarded as a slope with orientation and steepness determined by the weight values $\dot{w}_{ij}$, the bias $\dot{b}_j$ determines the distance from the origin (Fig. 2). The second layer weight matrix $\ddot{w}_{jk}$ determines the relative importance of the individual slopes in the summation and the bias $\ddot{b}_k$ corresponds to a constant offset. From this perspective, a MLP is similar to a Fourier series where instead of various sine and cosine terms the desired function is approximated by summing up various hyperbolic tangents (although in this case there is no requirement for the various hyperbolic tangents to be linearly independent as in Fourier series).

Several people including Hornik *et al.* (1989) and Cybenko (1989) have shown that such an MLP can approximate arbitrarily well any continuous functional mapping from one finite-dimensional space to another, provided the number of hidden units is sufficiently large. However, interesting this property might be, what has attracted most interest in using MLP's is the possibility of learning a specific mapping from a finite data set. Learning in practice corresponds to the minimization of a cost function, which measures the error between the network output and the desired output. The problem then reduces to finding the set of network parameters which minimize the cost function. The back-propagation algorithm, introduced by Rumelhart *et al.* (1986), allows the efficient computation of the derivatives of the cost function with respect to the network parameters. Back-propagation forms the basis of conventional iterative optimization algorithms such as conjugate gradients and quasi-Newton methods. In fact the huge popularity of neural network applications over the last two decades can be traced back to the introduction of the back-propagation algorithm.

The central goal of network training is to learn the relationship between input and output parameters from a finite data set $D = \{\mathbf{d}^n, \mathbf{m}^n\}$, consisting of N data points, where $\mathbf{d}$ forms the network input and $\mathbf{m}$ is the desired output. In our application the input $\mathbf{d}$ consists of phase and group velocities at different periods and the desired output $\mathbf{m}$ is the corresponding radially symmetric earth model. This ordering is arbitrary and we can equally well design a network where the role of network input and output are interchanged. The successfully trained network is then applied to new inputs $\mathbf{d}$ with unknown outputs $\mathbf{m}$. This can be regarded as a non-linear regression task, where instead of polynomials or splines a neural network model is used.

2.1 Neural networks for solving inverse problems

Since we are interested in solving an inverse problem, we first state the solution of a general inverse problem within the probabilistic framework and then show how different types of neural networks can be used to provide statistical information about the solution. According to Tarantola & Valette (1982) and Tarantola (2005) the *a posteriori* state of information is given by the conjunction of *a priori* information and information about the theoretical relationship between models and data:

$$\sigma(\mathbf{d}, \mathbf{m}) = k \frac{\rho(\mathbf{d}, \mathbf{m})\theta(\mathbf{d}, \mathbf{m})}{\mu(\mathbf{d}, \mathbf{m})}, \tag{2}$$

where k is a normalization constant, $\rho(\mathbf{d}, \mathbf{m})$ represents the prior knowledge on data $\mathbf{d}$ and model parameters $\mathbf{m}$, $\theta(\mathbf{d}, \mathbf{m})$ represents the physical theory relating model parameters $\mathbf{m}$ to the observable parameters $\mathbf{d}$, $\mu(\mathbf{d}, \mathbf{m})$ represents an objective reference state of minimum information, and all quantities other than k in eq. (2) are probability density functions (Tarantola & Valette 1982; Tarantola 2005). The solution of the general inverse problem is then given by the marginal posterior distribution,

$$\sigma(\mathbf{m}) = \int_D \sigma(\mathbf{d}, \mathbf{m})\, dd, \tag{3}$$

which in the classical Bayesian framework is a conditional probability density, conditional on the observed data (Tarantola 2005).

Eq. (3) performs the task of transferring the information contained in the data to the model parameters. The solution of the inverse problem for a specific observation may be approximately represented by a set of models distributed according to $\sigma(\mathbf{m})$. From this set of models any desired statistic such as the mean and variance of any model parameter can be computed. Note, however, that such statistics are most useful if the solution has a single dominant maximum, and become less useful if the solution has many relevant maxima. Instead of providing a set of samples, we propose to train a neural network, whose outputs directly parametrize the form of $\sigma(\mathbf{m})$, providing fully probabilistic information about the solution.

2.1.1 Conventional use of neural networks

Assume that we have a data set $D = \{d^n, m^n\}$, shown as circles in Fig. 3, where d is given by some function of m with added Gaussian noise ϵ; $d = g(m) + \epsilon$. As a consequence the conditional probability distribution of d given m is Gaussian:

$$p(d \mid m) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{(d - g(m))^2}{2\sigma^2} \right), \tag{4}$$

where the mean is given by $g(m)$ and σ is the standard deviation of ϵ.

In conventional neural network applications the mean corresponding to the forward function $g(m)$, is approximated by a neural network model $y(m^n; \mathbf{w})$ and the network parameters $\mathbf{w}$ are inferred

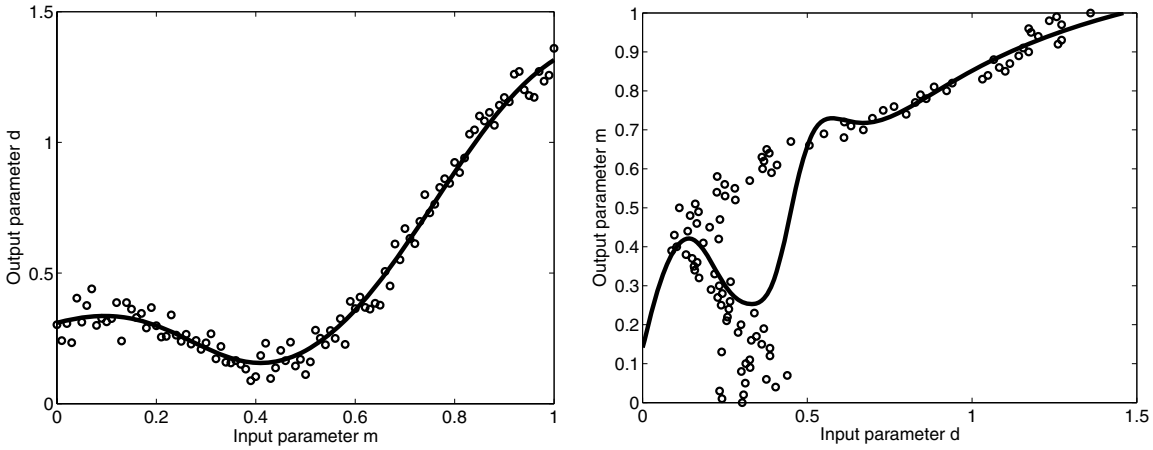

Figure 3. Two-parameter example of a network mapping which approximates the conditional average of the output parameter. (left) Single valued forward function; (right) multivalued inverse function obtained by interchanging the role of input and output variables. Circles indicate the training data and the solid line corresponds to the output of a trained MLP network with 10 hidden units.

from the data set D. This can be achieved by maximizing the likelihood of the data set D (or equivalently by minimizing its negative logarithm), which gives rise to the conventional least-square error measure (Bishop 1995),

$$E = \frac{1}{2}\sum_{n=1}^{N}(y(m^n;\mathbf{w}) - d^n)^2, \tag{5}$$

where the sum runs over the number of data points N in the training set.

Network training involves the minimization of eq. (5) with respect to the network parameters $\mathbf{w}$. Having found the optimal set of network parameters $\mathbf{w}^*$ which minimizes eq. (5), the neural network $y(m;\mathbf{w}^*)$ approximates the mean of $p(d \mid m)$, shown as the solid line in Fig. 3 (left), which indeed is a good approximation of the underlying function $g(m)$.

Imagine now that the roles of input and output parameters are interchanged (Fig. 3, right). Training a network by minimization of eq. (5) implicitly assumes that the conditional probability distribution $p(m \mid d)$, the solution to the inverse problem as stated in eq. (3), is Gaussian:

$$p(m \mid d) = \frac{1}{(2\pi)^{1/2}\sigma}\exp\left(-\frac{(m - g^{-1}(d))^2}{2\sigma^2}\right), \tag{6}$$

with mean given by the inverse function $g^{-1}(d)$.

Obviously the Gaussian assumption is violated, especially in the multivalued region (between [0.1, 0.5]). The trained network $y(d;\mathbf{w}^*)$ still approximates the mean of $p(m|d)$ as shown by the solid line in Fig. 3 (right). The mean of a multimodal distribution is, however, of limited significance (the average of various solutions is not necessarily itself a solution). This indicates that as long as $\sigma(m)$ the solution to the inverse problem for a specific observation as stated in eq. (3) is Gaussian or at least unimodal with a representative mean, the minimization of eq. (5) may be appropriate. If, however, $\sigma(m)$ is multimodal, the output of a trained network that minimizes an equation similar to eq. (5) is likely to give misleading results (Fig. 3, right).

2.1.2 The mixture density network

Devilee *et al.* (1999) introduced the Histogram and Median network, which provide a finite discretization of $\sigma(m)$. The k outputs of a Histogram network give an equidistantly sampled approximation of the solution whereas the k outputs of a Median network subdivide the solution into equal probability mass, but the required network outputs of such networks grow exponentially with increasing dimensionality of the solution distribution. We generalize their ideas and propose to model the solution as a mixture of Gaussians. This leads to the concept of the more compact mixture density network (MDN), a framework for modelling arbitrary probability distributions (in the same way as a conventional MLP can represent arbitrary functions (Bishop 1995)). The basic idea behind the MDN is to model the solution of the inverse problem as defined in eq. (3) as a sum of various Gaussians,

$$\sigma(\mathbf{m}) = \sum_{j=1}^{M}\alpha_j(\mathbf{d})\Theta_j(\mathbf{m}|\mathbf{d}), \tag{7}$$

where M is the number of Gaussian kernels, α_j are the mixing coefficients and can be interpreted as the relative importance of the j^{th} kernel, and Θ_j are the Gaussian kernels given by

$$\Theta_j(\mathbf{m}) = \frac{1}{(2\pi)^{(c/2)}\sigma_j^c(\mathbf{d})}\exp\left\{-\frac{(\mathbf{m} - \mu_j(\mathbf{d}))^2}{2\sigma_j^2(\mathbf{d})}\right\}, \tag{8}$$

where c is the dimensionality of $\mathbf{m}$. The parameters of this model such as the mixing coefficients α_j, the mean μ_j and variance σ_j^2 of the M Gaussians are taken to be the outputs of a conventional MLP. The total number of network outputs is $(c + 2) \times M$ as compared with the k^c outputs of a histogram network. The more complex the solution distribution we want to model, the more Gaussian kernels are required. A detailed description of the MDN is found in Bishop (1995).

Having decided on the specific parametric form of the probability distribution we want to model (eq. 7), the next stage is to use a data set $D = \{\mathbf{d}^n, \mathbf{m}^n\}$ to find the appropriate values of the network parameters and hence the parameters of the mixture model. From the principle of maximum likelihood, the following error measure is obtained (Bishop 1995):

$$E = -\sum_{n=1}^{N}\ln\left\{\sum_{j=1}^{M}\alpha_j(\mathbf{d}^n)\Theta_j(\mathbf{m}^n \mid \mathbf{d}^n)\right\}. \tag{9}$$

The same data set as in the above example is used to train a MDN network on the forward and the inverse mapping, respectively.

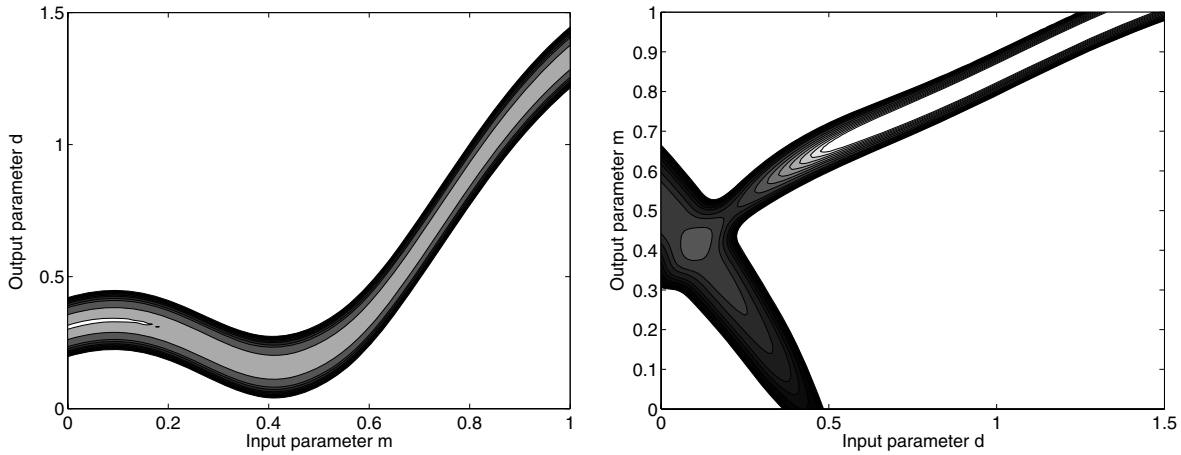

Figure 4. Contour plot of the solution distribution for various inputs. (left) of a MDN trained on the forward function and (right) of a MDN trained on the inverse. Both networks have three Gaussian kernels and 10 hidden units.

Having found the set of network parameters $\mathbf{w}^*$ which minimizes eq. (9), the *a posteriori* probability distribution of the output parameter can be computed for any given input according to eq. (7). This gives a far more complete description of the solution than the mean value alone. In Fig. 4 the probability density of the forward (left) and inverse (right) mapping is contoured. Note that the multivalued nature of the inverse mapping (in the region between [0.1, 0.5]) has been captured by the MDN. In the region (between [0, 0.1]) where no training data is available an extrapolation error is committed. From the outputs of a MDN network any desired statistic such as mean and variance can be computed. In this perspective the MLP network can be regarded as a special case of the more general MDN network. Compared to the Histogram and Median networks proposed by Devilee *et al.* (1999) which provide a finite discretization of the solution, the MDN gives a continuous approximation of the solution distribution.

2.2 Network training

As already mentioned network training corresponds to the minimization of an appropriate cost function (e.g. eqs 5 and 9). These cost functions are highly non-linear functions of the network parameters. Despite the complicated structure of the error surface good solutions are often found using gradient-based optimization methods. Gradient-based optimization algorithms proceed in an iterative way, starting from a user defined starting point. Obviously the starting values must be reasonably chosen in order to converge to a useful solution. Since we are using the hyperbolic tangent as the activation function, the summed inputs to the hidden units should be of order unity. Otherwise the activation functions are saturated and as a consequence the error surface becomes almost flat. In order to achieve this, it is common practice to normalize the input variables to have zero mean and standard deviation one. The network parameters are then drawn from a Gaussian with zero mean and standard deviation scaled by the number of input units feeding into each hidden unit for the first layer weights, and scaled by the number of hidden units feeding into each output unit for the second layer weights, respectively (Bishop 1995).

For the MDN as we consider it, the network parameters are initialized such that the solution $\sigma(\mathbf{m})$ in eq. (7) corresponds to the prior probability distribution $\rho(\mathbf{m})$. This ensures faster convergence and avoids ending up in poor local minima (Nabney 2002). Still each training run is sensitive to the initial set of network parameters. Therefore, it is common practice to train a particular network using different weight initializations.

In all our simulations we used the scaled conjugate gradient algorithm (Moller 1993), a recent variant of the conjugate gradient algorithm which avoids the expensive line-search procedure of conventional conjugate gradients. Conjugate gradient methods as well as quasi-Newton methods make use of second order information about the error surface and are, therefore, more efficient than simple gradient descent. Using quasi-Newton methods would require the storage of the inverse Hessian, which requires $O(w^2)$ storage, while conjugate gradients algorithm require only $O(w)$ storage. Since in our applications the number of network parameters w is rather large we opted for the latter.

3 INVERTING SURFACE WAVE DATA FOR CRUSTAL THICKNESS

In the previous section it was shown how neural networks can be used to learn a specific mapping from a finite training data set. We demonstrated the pitfalls of training a conventional MLP using the least-square error when the distribution of the output parameter is not Gaussian. Additionally we introduced the MDN network which allows to model any probability distribution as a sum of Gaussians. In this section we focus on the specific problem of using a MDN network to invert dispersion curves for Moho depth. In what follows we explicitly define the prior knowledge on data and model parameters as well as the theoretical relationship between model and data parameters. The solution to the inverse problem as stated in eq. (3) can then be given in a more explicit form.

3.1 *A priori* information

In this section the *a priori* information on data and model parameters are defined. By definition, the *a priori* information on model parameters is independent of the observations (Tarantola 2005). The joint probability density as in eq. (2) can thus be decomposed,

$$\rho(\mathbf{d}, \mathbf{m}) = \rho(\mathbf{d})\rho(\mathbf{m}). \tag{10}$$

In what follows the *a priori* probability densities of the data $\mathbf{d}$ and model parameters $\mathbf{m}$ are defined.

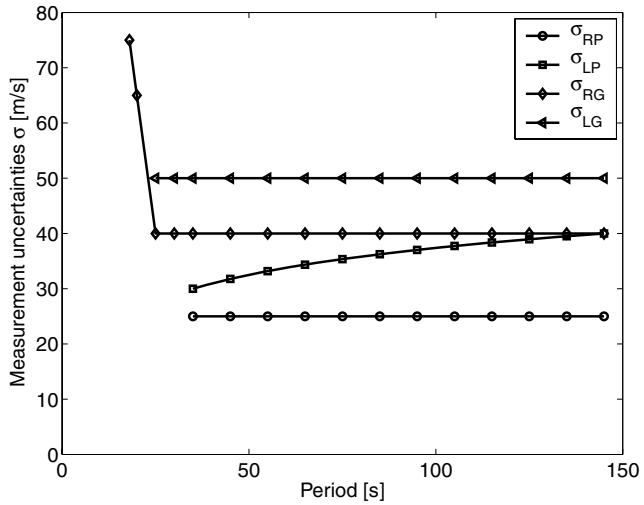

Figure 5. Measurement uncertainties of Rayleigh and Love phase and group velocities.

3.1.1 Data

In this study we consider fundamental mode Rayleigh and Love wave phase and group velocity models, respectively. The phase velocity models are from Trampert & Woodhouse (2003); the group velocity models from Ritzwoller *et al.* (2002). From these global phase and group velocity maps dispersion curves at discrete periods are constructed on a $2^\circ \times 2^\circ$ grid globally. Phase and group velocities are measured differently and thus provide two independent pieces of information to constrain crustal thickness.

Like all physical measurements the obtained dispersion curves are subject to uncertainties. We assume the uncertainties to be Gaussian and each dispersion curve can thus be represented as a probability density

$$\rho(\mathbf{d}) = \frac{1}{(2\pi)^{c/2}|\mathbf{C}_D|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{d}_{\rm obs} - \mathbf{d})^T \mathbf{C}_D^{-1}(\mathbf{d}_{\rm obs} - \mathbf{d}) \right\}, \quad (11)$$

where c is the dimensionality of $\mathbf{d}$, $\mathbf{d}_{\rm obs}$ is the observed dispersion curve, and $\mathbf{d}$ is the mean value of the distribution (i.e. the noiseless response of the unknown real Earth). A critical parameter is the covariance Matrix $\mathbf{C}_D$; we choose a diagonal covariance matrix (i.e. uncorrelated noise) with σ_{RP}, σ_{LP}, σ_{RG} and σ_{LG} as shown in Fig. 5, where the indices R, L, P and G refer to Rayleigh and Love waves and Phase and Group velocities, respectively. The error estimates for phase and group velocity maps are from Shapiro & Ritzwoller (2002).

3.1.2 Model parametrization

We tested various parametrizations and found that as long as we over-parametrize the model (given the potential resolving power of the data) the obtained solutions do not change. Thus, using an over-parametrized model does not introduce any implicit prior information and ensures that all the prior information is defined explicitly by defining the bounds of variations of all the model parameters. On each model parameter ($m^k_{\rm min} \leq m^k \leq m^k_{\rm max}$) hard bounds are imposed and further we assume that there exist no *a priori* correlations between individual model parameters.

Our model parametrization consists of 29 free parameters indicated by arrows in Fig. 6. Surface waves probe deeper parts of the Earth with increasing period. Dispersion curves in the period

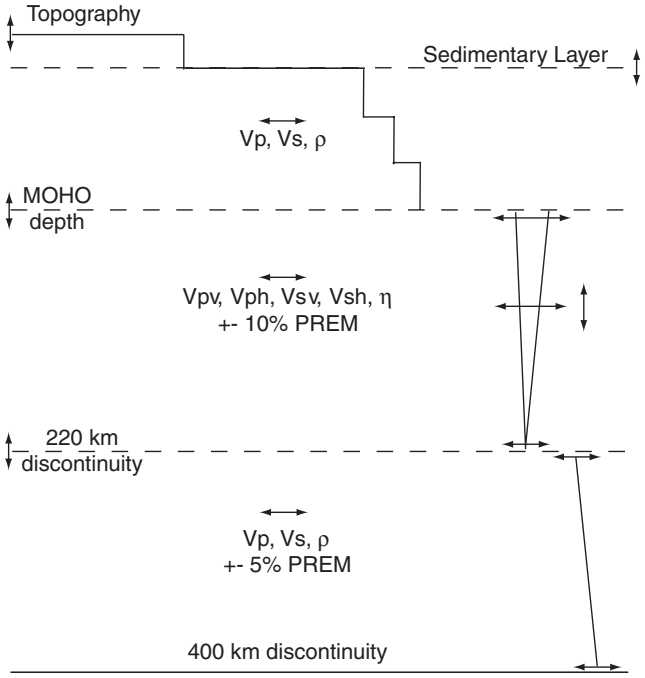

Figure 6. Model parametrization.

range considered in this study (from 18 s up to 145 s) are sensitive to Earth structure down to a depth of approximately 400 km. Therefore, below 400 km depth all model parameters are fixed to PREM (Dziewonski & Anderson 1981). From the Moho down to the 400 km discontinuity we use the same parametrization as PREM. From 220 km down to a depth of 400 km ρ, v_p and v_s are allowed to vary. These parameters are varied ± 5 per cent from PREM at the top and the bottom of this zone. Within this zone the depth profiles are then obtained by linear interpolation. The resulting depth profiles are linear with varying gradients. Within the anisotropic zone v_{pv}, v_{ph}, v_{sv}, v_{sh} and η are allowed to vary. These five parameters are drawn at the top, the bottom and at varying depths within that zone. Variations up to ± 10 per cent from PREM result in depth profiles consisting of two linear sections with varying gradients and intersection at varying depths. The 220 km discontinuity is allowed to vary ± 20 km. In Fig. 6 all the relevant parameters of the model parametrization are indicated by arrows. We distinguish between continental and oceanic models, below the Moho these two models are identical. The continental crust as well as the oceanic crust consist of three equally thick layers. In each of the three layers v_p, v_s and ρ are allowed to vary within a certain range. Since the oceanic crust is younger and more homogeneous than the continental crust the allowed range of variation is smaller for the oceanic models. The prior constraints on the crustal parameters were obtained by analysing CRUST2.0 (Bassin *et al.* 2000) an updated model of CRUST5.1 (Mooney *et al.* 1998). Additionally every second continental model has a sedimentary layer with varying thickness on top. The Moho depth and the topography are allowed to vary as well. For a summary of the explicit prior constraints on the model parameters see Table 1. All model parameters are drawn independently from a uniform distribution and each model realization $\mathbf{m}$ can be interpreted as a realization of the prior probability distribution over the model space $\rho(\mathbf{m})$. Note that in each successive realization all the model parameters are allowed to vary.

In Fig. 7 the partial derivatives of phase and group velocities with respect to Moho depth variations are plotted as a function of

Table 1. *A priori* information on model parameters as in CRUST2.1 within continental (top) and oceanic (bottom) crust.

	P wave [m s^{-1}]	S wave [m s^{-1}]	Density [g cm^{-3}]
Continental crust:			
Sedimentary layer	2850–3150	1700–1800	2295–2380
Top layer	5700–6300	3400–3600	2700–2800
Middle layer	6300–6600	3600–3800	2800–2900
Bottom layer	6600–7400	3600–4000	2900–3000
Oceanic crust:			
Sedimentary layer	–	–	–
Top layer	4950–5050	2500–2600	2600–2700
Middle layer	6500–6600	3600–3700	2800–2900
Bottom layer	7100–7200	3900–4000	3000–3100
	Topography [km]	Moho depth [km]	Thickness of sed. layer [km]
Continental:	0–8	10–100	1–10
Oceanic:	0–8 (Below sea level)	0–40	–

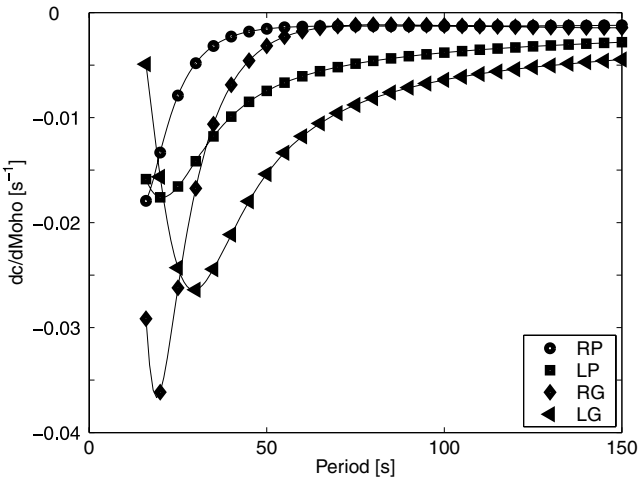

Figure 7. Sensitivity of Rayleigh and Love, phase and group velocities to variations in Moho depth as a function of period. Moho depth of our continental reference model is perturbed ±0.5 km and the fractional variations in phase and group velocities are shown. The indices R, L, P and G refer to Rayleigh and Love waves and Phase and Group velocities, respectively.

period for the continental reference model. An increase in Moho depth leads generally to a decrease in phase and group velocities. The shorter the period the more sensitive phase and group velocities become to variation in Moho depth. Group velocities have generally a higher sensitivity than phase velocities and Love waves a higher sensitivity than Rayleigh waves except for short periods (<30 s) where Rayleigh waves are more sensitive than Love waves.

3.2 Forward problem

In our particular application the forward problem consists of computing dispersion curves for a heterogeneous 3-D Earth. Instead of dispersion curves a neural network could equally well be trained on synthetic seismograms. Computing synthetic seismograms for a heterogeneous 3-D Earth using spectral-element methods (Komatitsch & Vilotte 1998; Komatitsch & Tromp 2002a,b) is computationally possible, but still a challenge for the large number needed for network training. Instead, we assume that a dispersion curve at a specific location is the result of a radially symmetric Earth and compute the corresponding dispersion curves using normal mode theory. For this normal mode approach, we use an algorithm developed by Woodhouse (1988) which allows the computation of Rayleigh and Love, phase and group velocities in a 1-D model. In terms of probability densities assuming an exact theory we obtain,

$$\theta(\mathbf{d}, \mathbf{m}) = \delta(\mathbf{d} - G(\mathbf{m}))\mu(\mathbf{m}), \tag{12}$$

where δ is the delta-function and G is the non-linear forward operator which computes synthetic data $\mathbf{d}$ for a specific radially symmetric Earth model $\mathbf{m}$. Note that by assuming an exact theory we implicitly assume that the inaccuracies in the forward relation are negligible compared to the uncertainties of the measurements.

3.3 The solution

So far our method does not differ from any sampling based inversion technique (e.g. Mosegaard & Tarantola 1995; Shapiro & Ritzwoller 2002). The prior knowledge on data and model parameters as well as the information about the physics relating data and model parameters were defined. Performing the integration in eq. (3) and using eqs (11) and (12) we obtain

$$\sigma(\mathbf{m}) = k\rho(\mathbf{m}) \exp\left\{ -\frac{1}{2}(\mathbf{d}_{\text{obs}} - G(\mathbf{m}))^T \mathbf{C}_D^{-1}(\mathbf{d}_{\text{obs}} - G(\mathbf{m}))\right\}. \tag{13}$$

This equation tells us how probable an earth model is, having made a specific observation. As already mentioned instead of using a sampling based approach we parametrize $\sigma(\mathbf{m})$ using a MDN as described in Section 2.1.2 The parameters of the MDN are learned from a finite synthetic data set, the training data. For this purpose we generate a data set of 500 000 continental and 500 000 oceanic models, drawn from the prior model distribution $\rho(\mathbf{m})$, and compute their corresponding dispersion curves. We assume that our prior

information for categorizing a location as oceanic or continental is 100 per cent accurate. A location is oceanic if water is present according to CRUST2.0. For this reason we train two networks, a continental and a oceanic one. Note that the synthetic dispersion curves contain the variations of all the model parameters. To simulate realistic measurement conditions, noise is added to the synthetic dispersion curves according to eq. (11) (see discussion below). Since we are only interested in Moho depth we ignore all the parameters except Moho depth from the sampled models and tabulate the resulting training set in terms of dispersion curves and the corresponding Moho depths. This can be seen as the marginalization step, integrating out all model parameters except Moho depth in eq. (13). One of the advantages of this approach is that once the parameters of the MDN are known, $\sigma(\text{moho})$ can be evaluated for any Moho depth and any observation without the need of (re)sampling model space and solving the forward problem for every visited model.

3.4 Efficiency of the MDN inversion

Inverting dispersion curves on a $2^\circ \times 2^\circ$ globally for Moho depth using a trained network solving 16 200 independent inverse problems takes only 2 s on a AMD Opteron(tm) Processor 242. The time-consuming part is the network training, i.e. the minimization of the error function. Since the contributions of each training pattern to the error and the gradient are independent it is straightforward to parallelize the training algorithm and achieved speed-ups scale almost linearly with the number of processors. On 50 processors training a MDN on 500 000 patterns takes 27 min.

Alternatively, the same training data could be used for a classical Monte Carlo inversion based on eq. (13). This involves comparing each of the 500 000 training patterns to the observed dispersion curve at each of the 16 200 locations of the $2^\circ \times 2^\circ$ grid and making a histogram. A single inversion performed that way takes 5 s on the same machine. Performing 16 200 individual inversions would then take 22.5 hr. This indicates that in this particular application, where 16 200 repeated inversions with similar prior information are required, the neural network approach significantly outperforms sampling based techniques.

4 REGULARIZATION

An important question is to investigate the implications of adding noise to the dispersion curves on the network mapping. We wish to derive an approximation to the inverse mapping which is valid at data points not necessarily contained in the training set D—the problem of generalization (interpolation). Obviously, the more flexible the network the smaller the discrepancy between network output and observations. However, this does not necessarily mean that a more flexible network interpolates better to unseen data points. This is generally known as the bias/variance trade off (Geman *et al.* 1992). If the network is not sufficiently flexible in terms of its ability to model non-linear relationships our approximation will exhibit a large bias (i.e. a systematic error); if on the other hand the network is too flexible the training data will be fit perfectly but interpolation performance will be poor (i.e. high variance). For this reason the effective complexity of the network has to be controlled. This can be done through the use of regularization which involves the addition of a penalty term to the error function

$$\tilde{E}_{reg} = E + \lambda E_r, \tag{14}$$

where E is the usual error measure, λ the regularization parameter describing the amount of regularization and E_r is the regularization term.

Within the Bayesian framework regularization corresponds to making specific assumptions about the prior distribution of the network parameters. The introduction of the Bayesian paradigm for neural network learning (e.g. MacKay 1992a,b; Neal 1996) offers an interesting view on regularization: the well-known minimum norm or weight decay regularization for example can be derived in the following way. Assume that the prior distribution of the network parameters is Gaussian with zero mean and variance $1/\lambda$

$$p(\mathbf{w}) = \frac{1}{(2\pi/\lambda)^{W/2}} \exp\left\{ -\frac{\lambda}{2}||\mathbf{w}||^2 \right\}, \tag{15}$$

taking the negative logarithm gives

$$\begin{aligned} -\ln p(\mathbf{w}) &= \lambda \frac{1}{2} \sum_{i=1}^{W} w_i^2, \\ &= \lambda E_r, \end{aligned} \tag{16}$$

where W is the number of network parameters. Eq. (16) shows that making the Gaussian assumption about the prior distribution of network weights results in the well-known minimum norm regularization where the regularization parameter λ is given by the inverse variance. By using this sort of regularization the network parameters are constrained to be of minimum norm. Neal (1996) shows that bigger network parameters $\mathbf{w}$ result in more complex network mappings. Constraining the network parameters to be small results as a consequence in a smoother mapping. In our case the network does not only have to interpolate between data points but additionally the data points are corrupted with noise. Because of the noise a regularizer is needed which constrains the mapping to be insensitive to variations in the input curves which are of the order of the noise.

Webb (1994) shows that the effect of noise on the input data can be compensated for by training a neural network on synthetic data by minimizing a regularized error measure. Under the assumption of uncorrelated Gaussian noise with zero mean and sufficiently small variance, the regularization term involves second order derivatives of the network output with respect to the inputs where the amount of regularization is governed by the noise variance. Computing the gradient of this modified error function with respect to the network parameters, is computationally too expensive to form the basis of a suitable training algorithm (involves third order terms). In a similar study Bishop (1995) shows that for the purpose of network training an equivalent regularizer can be derived which only depends on first order derivatives of the network outputs with respect to the inputs:

$$\tilde{E}^n_{\text{reg}} = E^n + \sigma_i^2 \left(\frac{\partial y_k^n}{\partial d_i^n} \right)^2, \tag{17}$$

where σ_i are the standard deviations of the assumed measurements error (i.e. standard deviations of the phase and group velocities at different periods (eq. 11)). The regularization term for the n^{th} pattern in eq. (17) is given by the squared derivative of the network output with respect to the network input and constrains the network mapping to be less sensitive to variations in the input data. In the same study Bishop (1995) shows that for small σ_i, adding uncorrelated Gaussian noise to the input data and minimizing the conventional least-square error function has the same effect as training a network on exact data but minimizing the regularized error function eq. (17). In what follows we perform synthetic tests to check if these two approaches are indeed similar.

4.1 Training with noise versus explicit regularization

In order to assess the generalization performance of different networks we generate a synthetic test set, consisting of 10 000 earth models and the corresponding synthetic dispersion curves. This test set was not included in the training set. Since we know the true Moho depth corresponding to each dispersion curve we can compare how well the trained network interpolates to unseen data points. In order to simulate the measurement errors we add uncorrelated Gaussian noise as described in eq. (11) to the synthetic dispersion curves of the test set. We consider three different networks: (I) a network trained on noiseless synthetic data, minimizing the conventional least-square error measure; (II) a network trained on noisy synthetic data, minimizing the conventional least-square error measure; (III) a network trained on noiseless synthetic data, minimizing the regularized error function eq. (17).

We test the interpolation performance of the three networks on (a) noiseless synthetic dispersion curves and (b) on noisy synthetic dispersion curves (i.e. simulating measurement errors). In Fig. 8 the mean Moho depth predictions of network I for exact (left) and noisy (right) dispersion curves are plotted against the true Moho depth of the underlying models. Obviously generalization performance of the network to synthetic dispersion curves is very good as indicated by the linear correlation. If we simulate the measurement errors and add uncorrelated Gaussian noise to the dispersion curves the network predictions become very poor and no obvious correlation is visible anymore. This indicates that a network trained on synthetic data approximates the exact inverse mapping well but performs badly on data corrupted by noise. Network I falsely interprets noise as coming from variations in the model parameters and hence the unrealistic network predictions in Fig. 8 (right). Without making assumptions about the measurement uncertainties or without any form of regularization a network trained on noiseless data will never be able to predict Moho depth for a real data set which is obviously corrupted by noise.

In Fig. 9 the predictions of network II (trained on noisy input curves) for the same test curves as before ((left) synthetic; (right) noisy) are plotted against the Moho depth of the true underlying model. As opposed to network I, network II predicts the correct Moho depth even if the dispersion curves are corrupted by noise. Through the addition of noise to the training data, network II is able to invert noisy dispersion curves.

In Fig. 10 the predictions of network III are plotted against the True Moho depth. As for network II, the performance for noisy data is very good. We thus showed that adding noise to the input data leads to an implicit regularization which has the same effect

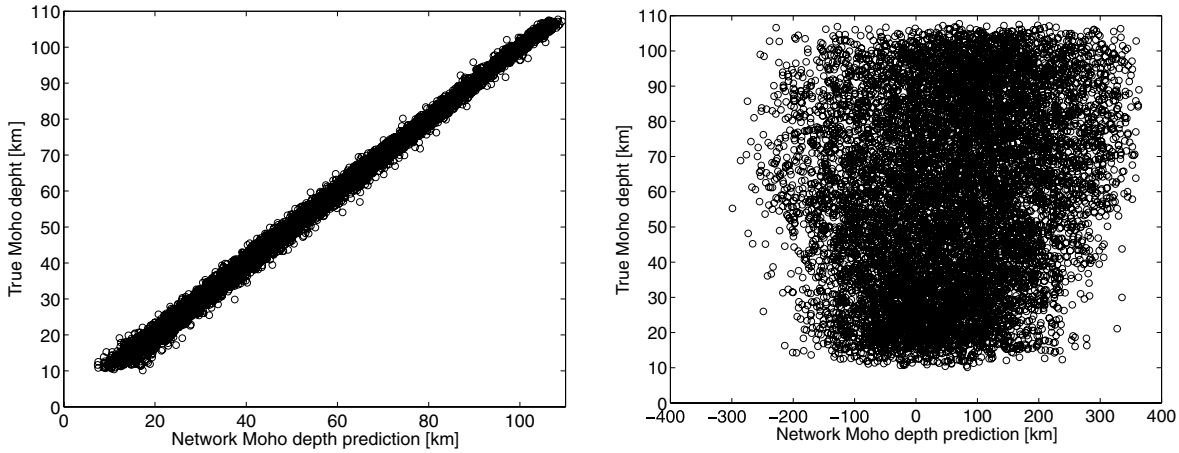

Figure 8. Network Moho depth predictions for 10 000 synthetic dispersion curves against the Moho depth of the underlying Models. The network was trained on exact noiseless data, minimizing the least-square error measure. (left) network predictions for noiseless data; (right) network predictions for noisy data.

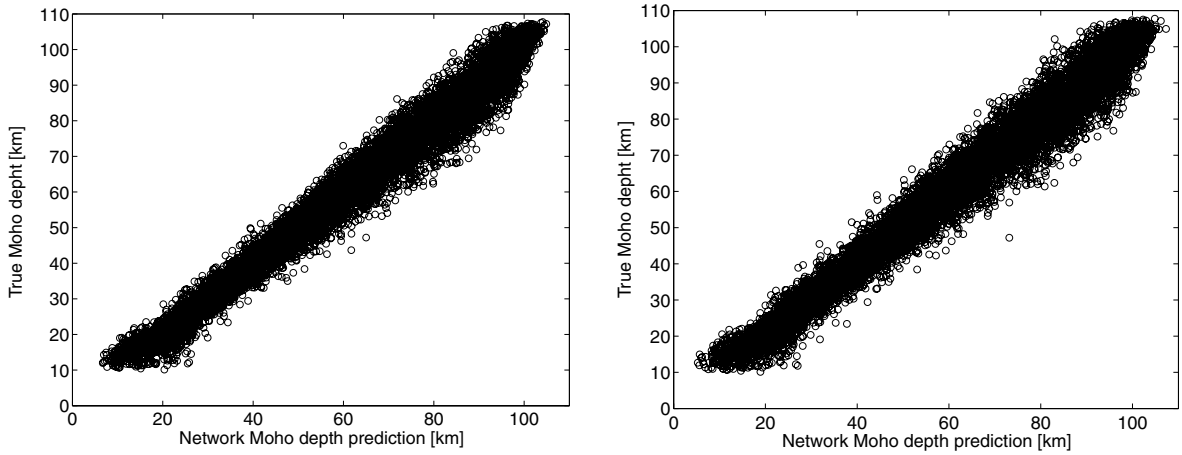

Figure 9. Network Moho depth predictions for 10 000 synthetic dispersion curves against the Moho depth of the underlying Models. The network was trained on noisy data, minimizing the least-square error measure. (left) Network predictions for noiseless data; (right) network predictions for noisy data.

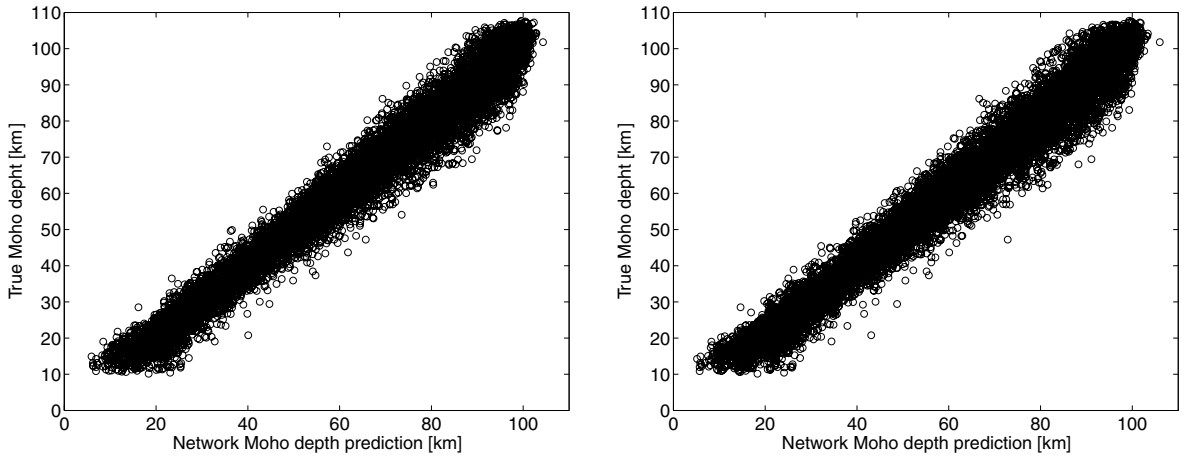

Figure 10. Network Moho depth predictions for 10 000 synthetic dispersion curves against the Moho depth of the underlying Models. The network was trained on exact data, minimizing the regularized error measure. (left) Network predictions for noiseless data; (right) network predictions for noisy data.

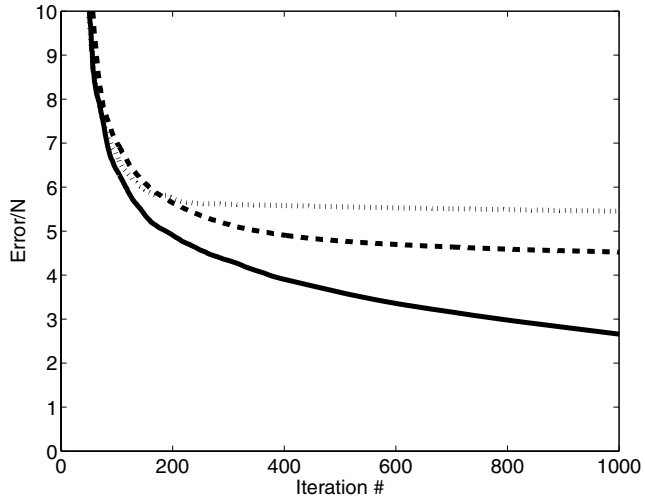

Figure 11. Learning curves for three different networks, network I (solid), network II (dashed) and network III (dotted).

as the explicit regularization in eq. (17). The regularization term in eq. (17) constrains the network mapping to be less sensitive to small variations in the dispersion curves, where the amount of regularization depends on the noise variance. As a consequence the network mapping becomes insensitive to variations in the dispersion curves which are within the measurement uncertainties as described by the standard deviations σ_i.

Another interesting feature can be observed by looking at the learning curves of the three different networks (Fig. 11). The solid line corresponds to the error of network I, the dashed line corresponds to network II and the dotted line to network III. Network I has the worst generalization performance of all the three networks even though the error is smallest. Even after 1000 iterations the error still decreases, this indicates that the network starts to fit the training data below the noise level. Network II and III on the other hand are characterized by learning curves which converge after 600 iterations for network II and already after 200 iterations for network III. Even though network III requires fewer iterations till convergence, it is more efficient to train network II from a computational point of view, because the computation of the derivatives of the regularization term in eq. (17) with respect to the network parameters is very expensive. Ten iterations for network II take 17 s on a AMD Opteron(tm) Processor 242 whereas ten iterations for network III on the same machine take 13 min. In what follows all the networks considered are trained on noisy dispersion curves.

5 RESULTS

We present global Moho depth maps with corresponding uncertainties inverted from phase and group velocities of Rayleigh and Love waves. The data set consists of azimuthally averaged global phase (Trampert & Woodhouse 2003) and group velocity (Ritzwoller *et al.* 2002) maps. From these maps we constructed dispersion curves at a $2^\circ \times 2^\circ$ grid globally. We considered phase velocities at discrete periods of 35, 45, . . . , 145 s; Rayleigh group velocities at discrete periods of 18, 20, 25, 30, 35, 45, . . . , 145 s and Love group velocities at discrete periods of 25, 30, 35, 45, . . . , 145 s. For all our simulations we used MDN's with three Gaussian kernels, resulting in nine output units. For the phase velocity inversion the networks had 24 input units, 50 hidden units and 9 output units; for the group velocity inversion the networks had 30 input units, 50 hidden units, 9 output units; for the joint inversion the networks had 54 input units, 100 hidden units and 9 output units. We found that the number of hidden units is not a crucial parameter and networks with different number of hidden units give similar results. By choosing three Gaussian kernels we allow the posterior Moho depth distributions to have up to three distinct maxima. In all our simulations we found that the resulting Moho depth distributions are characterized by a single well-defined maximum. Using more Gaussian kernels than the expected number of different maxima has little effect, since the network always has the option either to 'switch off' redundant kernels by setting the corresponding mixing coefficients to small values, or to 'combine' kernels by giving them similar mean and variance (Bishop 1995). This indicates that our results do not depend crucially on the inversion method; using a different amount of hidden units and/or Gaussian kernels will give results consistent with those presented. In order to avoid being stuck in a local minimum during the network training, we trained independent networks from different starting points. Again, the different networks produced very similar results and we chose the network with the smallest error.

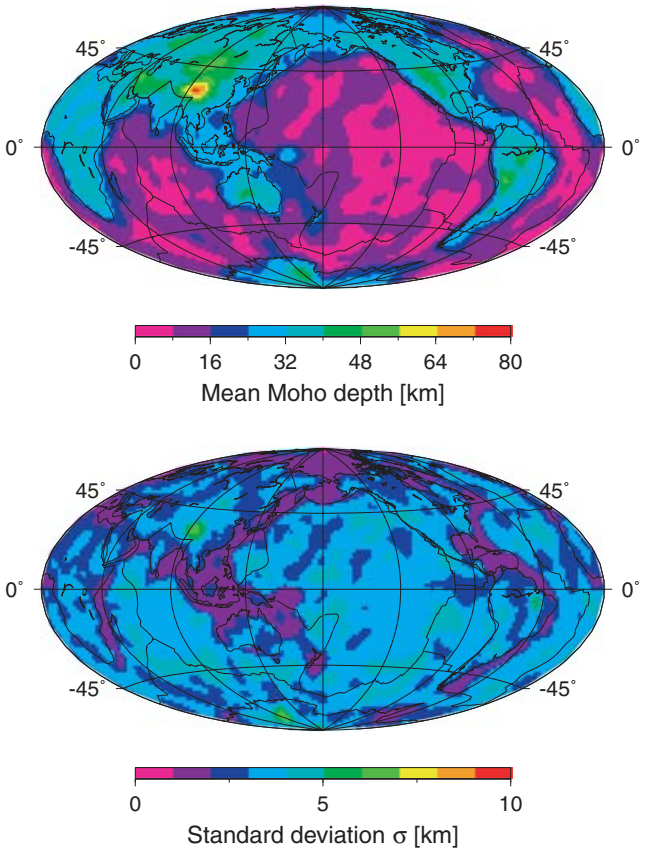

Figure 12. Global Moho depth map as a result of the joint inversion. (top) mean Moho depth [km]; (bottom) standard deviation σ [km], both extracted from the output of a MDN network.

5.1 Global map of crustal thickness

We used a MDN network to perform a joint inversion of phase and group velocities together. From the obtained Moho depth distributions mean Moho depths and standard deviations are computed. In Fig. 12, mean Moho depths and the corresponding standard deviations σ obtained from the joint inversion of phase and group velocities are plotted. Note how well the obtained mean Moho depth follows the topography. All the major features such as the contrast of continental and oceanic crust as well as thick continental roots beneath the main mountain ranges are retrieved. Fig. 12 (top) indicates that crustal thickness increases away from the mid-ocean ridges and hence with increasing age. Evidence for such an age signal in the oceanic crust is obtained by looking at the cumulative probability distribution of global oceanic crustal thickness belonging to four different age windows (Fig. 13). Mean Moho depth estimates above 70°N were excluded in this analysis. For instance, 22 per cent of global oceanic crust younger than 40 Myr is thicker than 10 km, whereas 50 per cent of the global oceanic crust between 120–150 Myr is thicker than 10 km. Age dependence of oceanic crustal thickness is generally considered to be weak or non-existent. However, there exist other studies which found evidence for an age signal in oceanic crustal thickness (Tanimoto 1995).

In Fig. 14 a histogram of all the standard deviations is shown; (left) for oceanic and (right) for continental regions. For oceanic regions all standard deviations are smaller than 5 km, whereas for continental regions, standard deviations up to 7 km are observed. The standard deviations depend on the prior information over the

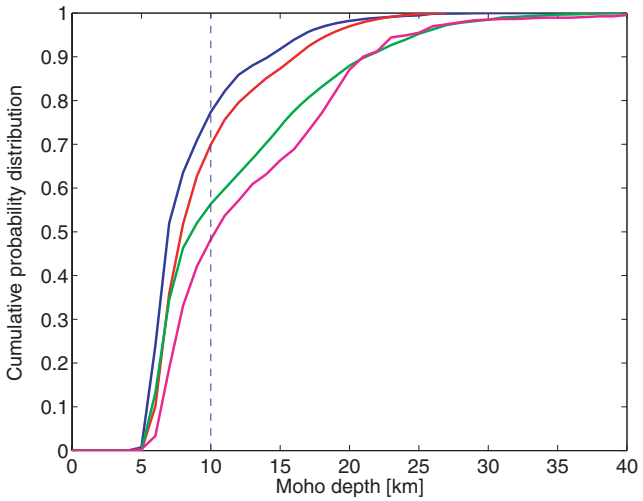

Figure 13. Cumulative probability distribution of mean Moho depth belonging to four different age windows; 0–40 [Myr] (blue), 40–80 [Myr] (red), 80–120 [Myr] (green), 120–150 [Myr] (magenta).

data space (i.e. the assumed uncertainties of the phase and group velocity maps) as well as the prior information over the model space. Increasing σ in eq. (11) leads to more homogeneous but on average higher standard deviations, since less importance is given to the data or more regularization is applied. Using a broader prior over the model space has a similar effect, since potentially more models might explain the data. We thus conclude that, given the prior information, defined in Section 3.1, fundamental mode surface waves in the period range considered constrain Moho depth with an average standard deviation of ± 3 km. An observed feature are small-scale variations of lower and higher standard deviations, indicating that the phase and group velocity maps are not everywhere in equally good agreement. However, on a global scale the presented results are in very good agreement with common knowledge about crustal thickness, indicating that overall the used data sets are reliable. Although we used a $2^\circ \times 2^\circ$ grid for convenient comparison with other crustal models, the lateral resolution of our Moho depth map is that of the combined resolution of the input phase and group velocity maps ranging between 500 and 1000 km. As a consequence the presented Moho depths and the corresponding standard deviations are averaged estimates over an area determined by the lateral resolution of the phase and group velocity maps and do not represent point estimates.

5.2 Moho depth distribution

To illustrate how well the different data sets constrain Moho depth it is instructive to look at the Moho depth distribution at specific locations. We chose four locations in Eurasia, one location near the Mid-Atlantic ridge and another location in the Pacific near the west coast of central America. In Fig. 15 the Moho depth distribution obtained by inverting phase (top) and group (middle) velocities alone and by inverting phase and group velocities together (bottom) are plotted. The broadness of the Moho depth distribution is mainly due to trade offs with all the other parameters. Generally group velocities constrain Moho depth better than phase velocities; the same was already observed by comparing the sensitivities of phase and group velocities to variations in Moho depth (Fig. 7). Additionally the group velocity data set includes lower periods which are more sensitive to crustal structure. In Central Tibet and to a lesser extent in

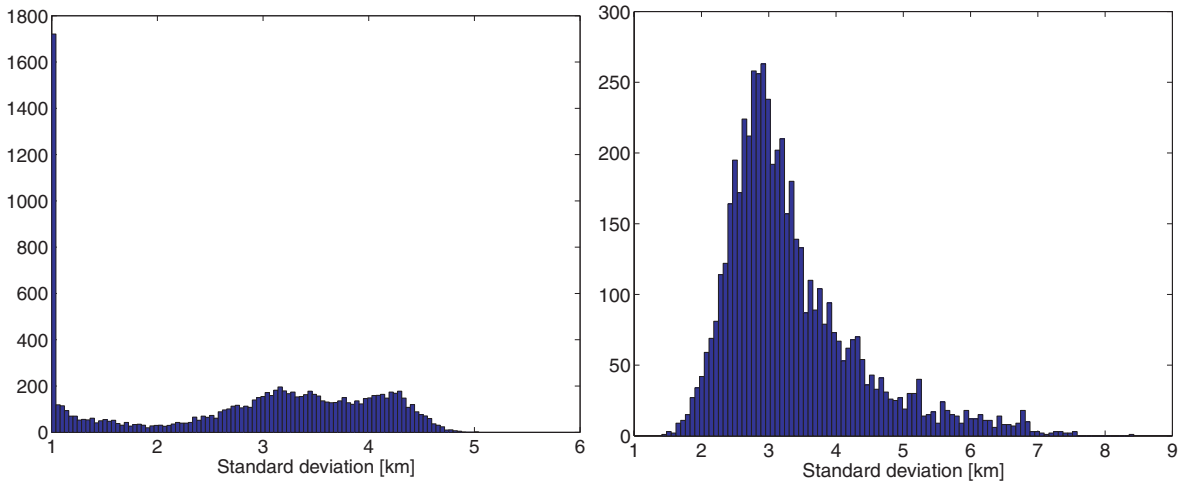

Figure 14. Histogram of all the standard deviations extracted from the joint MDN network inversion; (left) oceans, (right) continents.

the Tarim Basin the difference between the phase and group velocity inversion is large, indicating that the two data sets in this region are inconsistent. This observed discrepancy is further evidence for special structural features in the Tibetan region and might be explained by crustal anisotropy as proposed by Shapiro *et al.* (2004), but not included in our training set.

At locations in India and the Caspian/Aral region it is nicely visible that even though crustal thickness is mainly constrained by group velocities, phase velocities do contribute additional information, resulting in a tighter Moho depth distribution for the joint inversion. The same can be observed for the two locations in the oceans, at the Mid-ocean ridge and near the Pacific coast of central America.

Using the samples we trained the network on, we can construct a histogram of the Moho depth distribution according to eq. (13) by comparing the observed dispersion curve with each synthetic dispersion curve of the training data (referred to as Monte Carlo inversion here). The histograms for all three different inversions are superimposed on the Moho depth distribution obtained with the MDN (Fig. 15). Note that using the MDN network, consistent results are obtained compared to the Monte Carlo inversion. This indicates that the MDN network and Monte Carlo methods provides similar probabilistic information on the solution.

Compared with the results obtained by Devilee *et al.* (1999) who inverted two different phase and group velocity data sets at the same locations, our results are characterized by smaller uncertainties. This can be explained by the more recent and complete data set used in this study.

5.3 Comparison with other crustal models

We compared our crustal thickness estimates with the current knowledge about crustal thickness as in CRUST2.0 (Bassin *et al.* 2000). Additionally we compared our result with the model CUB2 from Shapiro & Ritzwoller (2002), who inverted a similar data set for crustal thickness using a Monte Carlo approach. In Fig. 16 the three crustal models, CRUST2.0, CUB2 and our own MDN model are compared. (top) MDN model is plotted against CRUST2.0; (middle) MDN model is plotted against CUB2; and (bottom) CUB2 is plotted against CRUST2.0. Note the linear correlation in all the three plots indicating overall agreement between the three models.

Interestingly, the agreement of our MDN model with CUB2 is better than with CRUST2.0. Shapiro & Ritzwoller (2002) restricted the model search to a small region ± 5 km around CRUST5.1, hence its better agreement with CRUST2.0. Keeping in mind that the allowed Moho depth variation in this study is between 0 and 110 km for continental regions and between 0 and 40 km for oceanic regions, some discrepancy between our MDN model and CUB2 has to be expected, still the same trend of deviation form CRUST2.0 is observed.

In Fig. 17 the difference between CRUST2.0 and our MDN model (top) and the difference between CUB2 and our MDN model (bottom) divided by the standard deviation of our MDN model are plotted. This allows to localize specific regions where the disagreement with CRUST2.0 and/or CUB2 is bigger than $\pm 1\sigma$. Almost everywhere our estimates are within $\pm 1\sigma$ of the two other models; however, in specific regions such as central Africa, the backarc of the Rocky Mountains, west Australia and underneath the Himalayas as well as the Andes according to our model the crust seems thinner than proposed by CRUST2.0 and CUB2. This can be explained partly because, due to the limited lateral resolution of the data set used, the really thick crust under the Himalayas and the Andes is not captured. In central Africa, the backarc of the Rocky Mountains and west Australia, there seems to be strong evidence that crustal thickness inferred from surface wave data is thinner than proposed by CRUST2.0. It is interesting to see that disagreement with CRUST2.0 and CUB2 coincidences geographically and that within these regions the disagreement with CUB2 is smaller than with CRUST2.0. This indicates that due to the tighter constraints CUB2 stays closer to CRUST2.0 than our MDN model, but the disagreement of both models with respect to CRUST2.0 has the same sign.

In Fig. 18 Moho depth of the MDN model (blue) with corresponding uncertainties (dashed), of CRUST2.0 (red) and of CUB2 (green) is shown along five different profiles, whose locations are indicated in Fig. 19 . Along profile AA′ crossing North America eastwards, Moho depth seems to be thinner as predicted by CRUST2.0. The difference increases towards the East coast. As already mentioned CUB2 shows the same trend but to a lesser extent. Along profile BB′ eastwards across Eurasia, all three models are generally in good agreement and show the same crustal thickness patterns. Profile CC′ runs across the Himalayas in northward direction, thickening of the crust beneath the Himalayas is the most dominant feature of this profile. Note the thinning of the crust underneath Tibet around 40N,

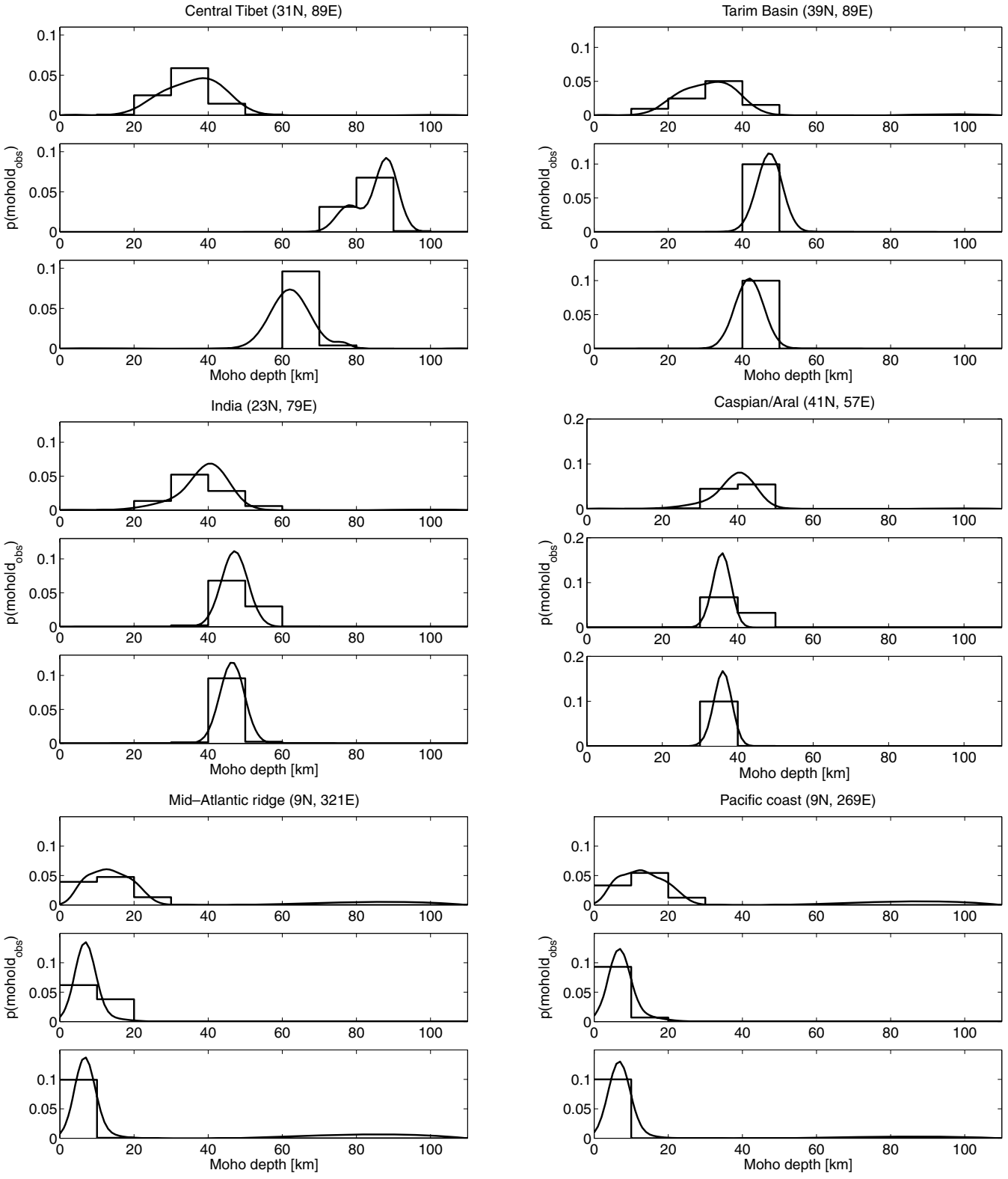

Figure 15. Moho depth distribution at six different locations. Each Fig. shows the Moho depth distribution obtained by phase velocity inversion (top); group velocity inversion (middle); joint inversion (bottom). (solid) Output of a MDN network; (histogram) result of a Monte Carlo inversion (see text).

which is not apparent in CRUST2.0 but visible in CUB2 to a lesser extent. At two locations along this profile the Moho depth distribution is shown in Fig. 15. The observed results along the two profiles BB′ and CC′ are consistent with the crustal thickness map across Eurasia obtained by Devilee *et al.* (1999). Along profile DD′ crossing Africa northwards, we observe a thinner crust than the other two models through most of the continent. Interestingly we observe a thickening of the crust beneath the Hellenic Arc between 35°–45°N,

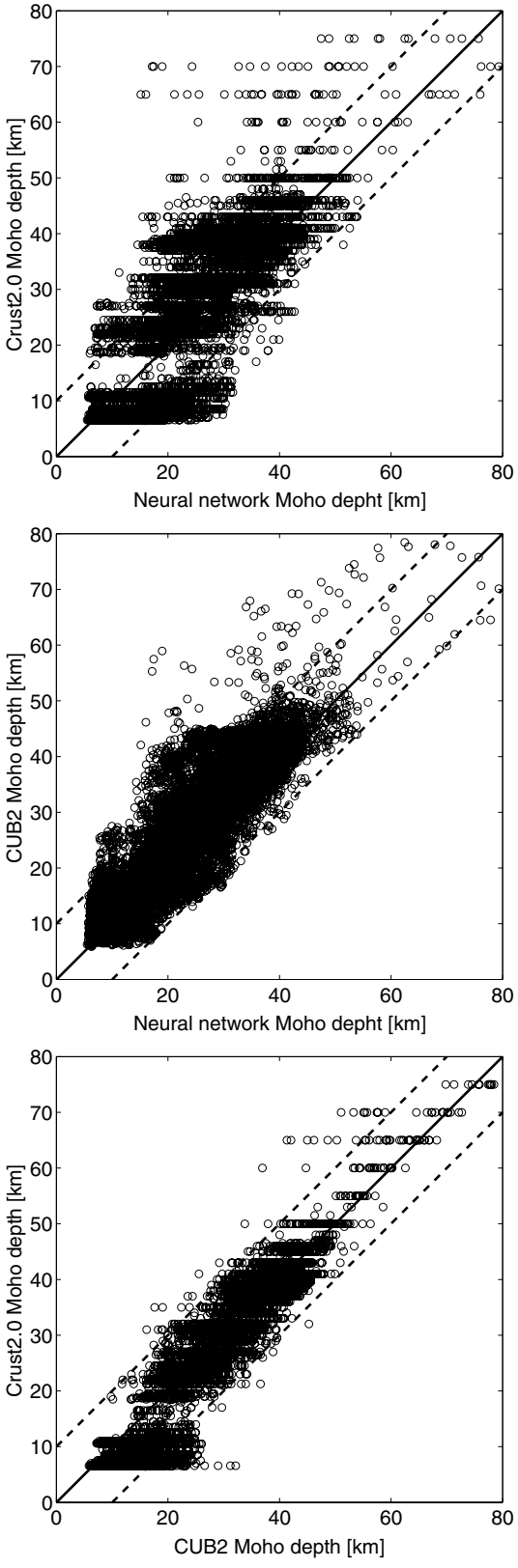

Figure 16. Scatter plot of three different global crustal models. (top) MDN versus CRUST2.0; (middle) MDN versus CUB2; (bottom) CUB2 versus CRUST2.0.

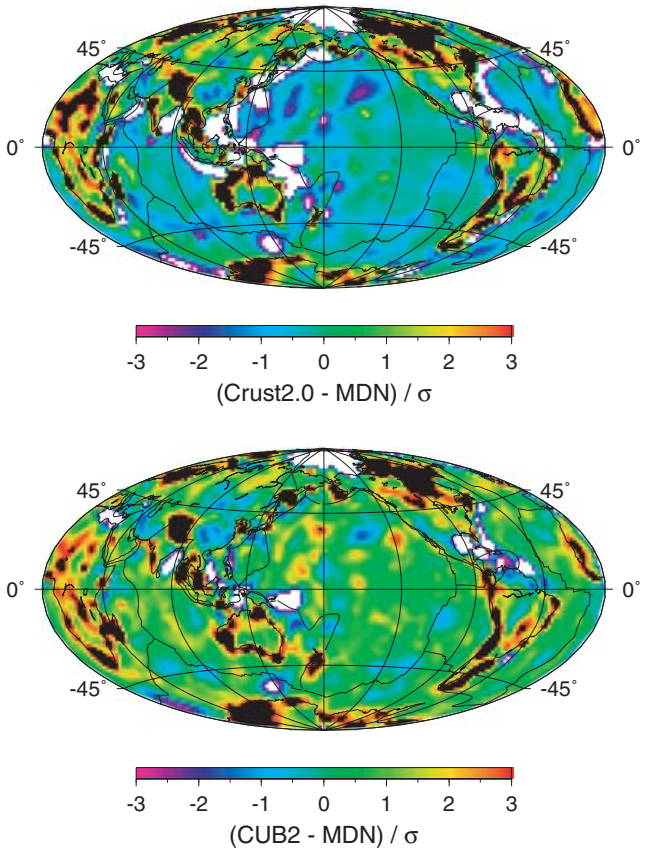

Figure 17. The difference between CRUST2.0 (top); CUB2 (bottom) and our MDN Moho depth estimates divided by the standard deviations.

whereas the other two models show a thinning of the crust in this region. Along profile EE′ through the Atlantic Ocean, crossing the Mid-ocean ridge, the three models are in good agreement.

6 CONCLUDING REMARKS

We presented a global crustal thickness model with corresponding uncertainties. The results were obtained using a neural network approach, the MDN, which allows one to model the posterior Moho depth probability distribution as a mixture of Gaussians. The whole procedure involves no linearization.

Generally non-linear inverse problems are solved using sampling based techniques. We have demonstrated that the MDN inversion has the following advantages over sampling based techniques: (1) if many repeated inversions are required the MDN inversion can be extremely efficient, inverting a dispersion curve using a trained network takes only a fraction of a second; (2) since the neural network interpolates between samples, far wider bounds on the model parameter values can be used, resulting in less biased results and (3) a continuous representation of the posterior model parameter probability distribution is obtained.

A large part of this work focused on the important concept of regularization. We made the link between regularization and measurement accuracy. The more exact a measurement is, the less regularization is required. All variations in the data in the order of the measurement noise provide no information. Without knowing the measurement error, noise will be falsely interpreted as variations in the model parameters. In our approach the assumed noise model determines the amount of regularization, which is either implicit

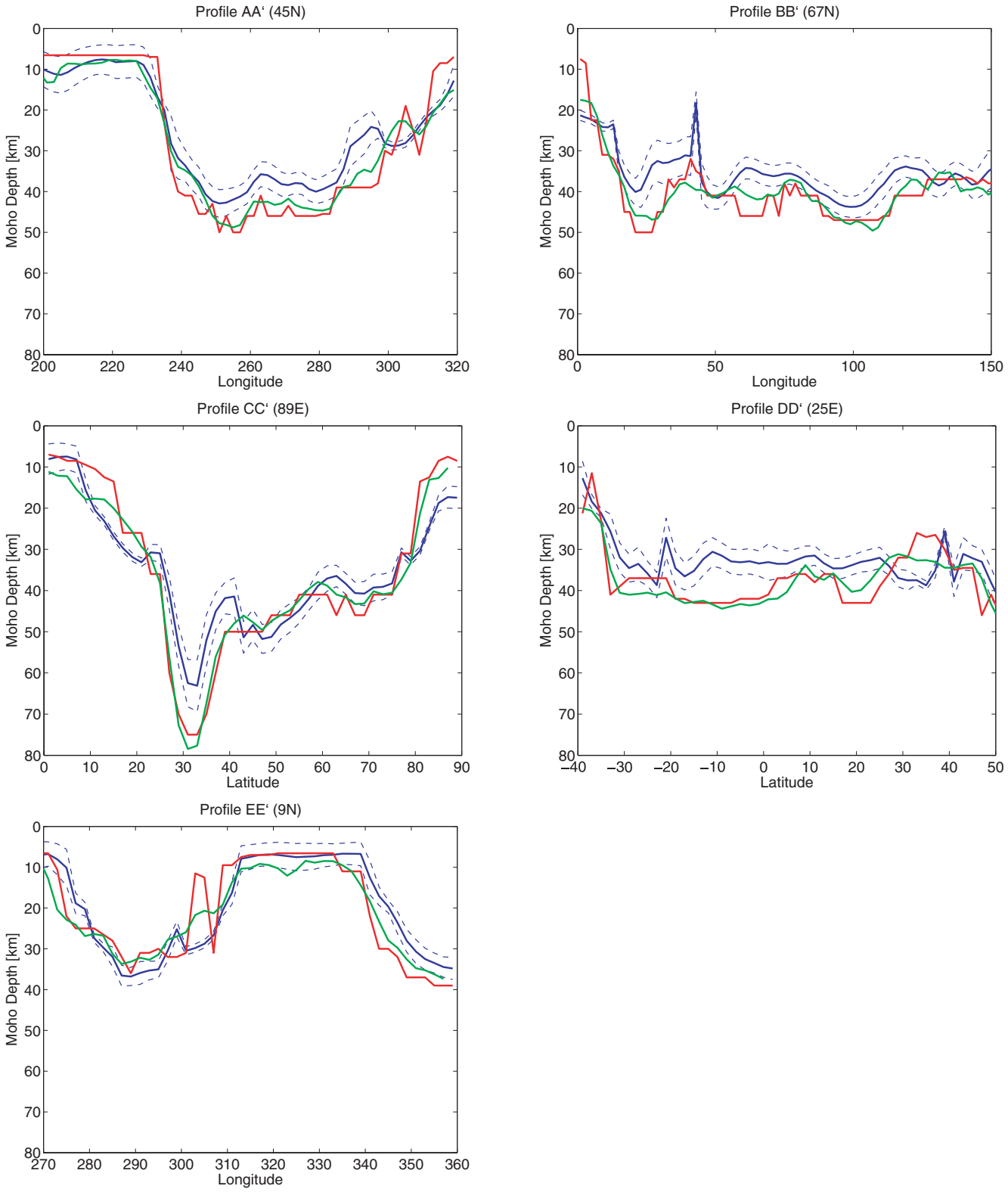

Figure 18. Five different Moho depth profiles. (blue) MDN Moho depth with corresponding uncertainties (dashed); (red) CRUST2.0 Moho depth; (green) Moho depth from CUB2.

through the addition of the noise to the synthetic training data, or explicit through the addition of a penalty term to the error measure to be minimized. Without any form of regularization the neural network approximation to the inverse mapping will not generalize well to observed data points which are corrupted by noise. This fits well with the theory of sampling based inversion techniques where prior assumptions have to be made in order to infer model parameters from noisy measurements.

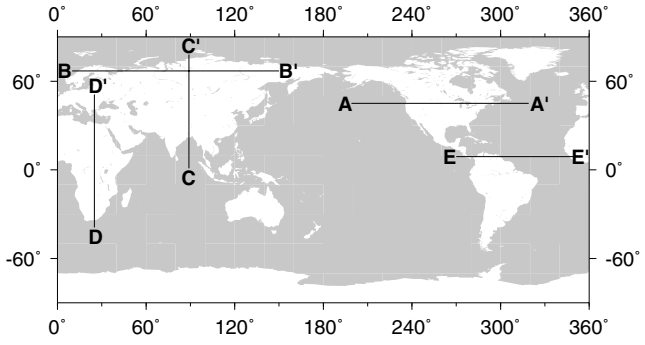

Figure 19. Map showing the location of five different profiles.

Finally we compared our model with current knowledge about crustal structure as represented by CRUST2.0 and CUB2 a recent model from Shapiro & Ritzwoller (2002). The overall agreement of $\pm 1\sigma$ with this two models is very good, where agreement is generally better with CUB2. The observed difference can be explained by different constraints applied to Moho depth variations. (Shapiro & Ritzwoller 2002, constrains Moho depth to vary ± 5 km around CRUST5.1 while we constrain Moho depth *a priori* to vary between 0 and 110 km for continental regions and between 0 and 40 km for oceanic regions). Our model shows generally the same trend as CUB2 with respect to differences from CRUST2.0. A notable new finding is that we see evidence for thickening of oceanic crust with increasing age.

ACKNOWLEDGMENTS

We would like to thank M.H. Ritzwoller for providing the group velocity maps, and N.M. Shapiro for making the model CUB2 available. Additionally we would like to thank H. Paulssen for helpful discussions and comments regarding the manuscript. UM is grateful for the support of the HPC-Europa programme, funded under the European Commission's Research Infrastructures activity of the Structuring the European Research Area programme, contract number RII3-CT-2003-506079. Part of the calculations were performed on a 64 node cluster financed by the Dutch National Science Foundation under grant number NWO:VICI865.03.007.

REFERENCES

Aristodemou, E., Pain, C., de Oliveira, C., Goddard, T. & Harris, C., 2005. Inversion of nuclear well-logging data using neural networks, *Geophys. Prospect.,* **53,** 103–120.

Bassin, C., Laske, G. & Masters, G., 2000. The current limits of resolution for surface wave tomography in north america, *EOS, Trans. Am. geophys. Un.,* **F897**.

Benaouda, D., Wadge, G., Whitmarsh, R., Rothwell, R. & MacLeod, C., 1999. Inferring the lithology of borehole rocks by applying neural network classifiers to downhole logs: an example from the ocean drilling program, *Geophys. J. Int.,* **136,** 477–491.

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK.

Cornford, D., Nabney, I.T. & Bishop, C.M., 1999. Neural network based wind vector retrieval from satellite scatterometer data, *Neural Computing and Applications,* **8,** 206–217.

Curtis, A. & Woodhouse, J., 1997. Crust and upper mantle shear velocity structure beneath the Tibetan plateau and surrounding regions from interevent surface wave phase velocity inversion, *J. geophys. Res.,* **102**(B6), 11 789–11 813.

Curtis, A., Trampert, J., Snieder, R. & Dost, B., 1998. Eurasian fundamental mode surface wave phase velocities and their relationship with tectonic structures, *J. geophys. Res.,* **103**(B11), 26 919–26 947.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems,* **2,** 304–314.

Das, T. & Nolet, G., 2001. Crustal thickness estimation using high frequency rayleigh waves, *Geophys. Res. Lett,* **123,** 169–184.

Devilee, R., Curtis, A. & Roy-Chowdhury, K., 1999. An efficient, probabilistic neural network approach to solving inverse problems: inverting surface wave velocities for Eurasian crustal thickness, *J. geophys. Res.,* **104**(B12), 28 841–28 857.

Dziewonski, A.M. & Anderson, D.L., 1981. Preliminary reference earth model, *Phys. Earth planet. Inter.,* **25,** 297–356.

Geman, S., Bienenstock, E. & Doursat, R., 1992. Neural networks and the bias/variance dilemma, *Neural Computation,* **4,** 1–58.

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

Komatitsch, D. & Tromp, J., 2002a. Spectral-element simulations of global seismic wave propagation-i. validation, *Geophys. J. Int.,* **149,** 390–412.

Komatitsch, D. & Tromp, J., 2002b. Spectral-element simulations of global seismic wave propagation-ii. three-dimensional models, oceans, rotation and self-gravitation, *Geophys. J. Int.,* **150,** 303–318.

Komatitsch, D. & Vilotte, J.-P., 1998. The spectral element method: an efficient tool to simulate the seismic response of 2d and 3d geological structures, *Bull. seism. Soc. Am.,* **88,** 368–392.

Lampinen, J. & Vehtari, A., 2001. Bayesian approach for neural networks - review and case studies, *Neural Networks,* **14**(3), 257–274.

MacKay, D.J., 1992a. Bayesian interpolation, *Neural Computation,* **4,** 415–447.

MacKay, D.J., 1992b. A practical Bayesian framework for backprop networks, *Neural Computation,* **4,** 448–472.

Moller, M., 1993. A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks,* **6,** 525–533.

Montagner, J.-P. & Jobert, N., 1988. Vectorial tomography—ii. Application to the indian ocean, *Geophys. J.,* **94,** 309–344.

Mooney, W.D., Laske, G. & Masters, T.G., 1998. Crust5.1: a global crustal model at 5 × 5 degrees, *J. geophys. Res.,* **103**(B1), 727–747.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. geophys. Res.,* **100**(B7), 12 431–12 447.

Nabney, I.T., 2002. *Netlab: Algorithms for Pattern Recognition*, Advances in Pattern Recognition, Springer Verlag, London, UK.

Neal, R.M., 1996. *Bayesian Learning for Neural Networks*, no. 118 in Lecture Notes in Statistics, Springer Verlag, New York, USA.

Ritzwoller, M.H. & Levshin, A.L., 1998. Eurasian surface wave tomography: Group velocities, *J. geophys. Res.,* **103**(B3), 4839–4878.

Ritzwoller, M.H., Shapiro, N.M., Barmin, M.P. & Levshin, A.L., 2002. Global surface wave diffraction tomography, *J. geophys. Res.,* **107**(B12), 2335.

Roth, G. & Tarantola, A., 1994. Neural networks and inversion of seismic data, *J. geophys. Res.,* **99,** 6753–6768.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986. Learning internal representations by error propagation, in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pp. 318–362, MIT Press, Cambridge, MA, USA.

Sambridge, M., 1999a. Geophysical inversion with a neighborhood algorithm-i. searching a parameter space, *Geophys. J. Int.,* **138,** 479–494.

Sambridge, M., 1999b. Geophysical inversion with a neighborhood algorithm-ii. appraising the ensemble, *Geophys. J. Int.,* **138,** 727–746.

Shapiro, N.M. & Ritzwoller, M.H., 2002. Monte-Carlo inversion for a global shear-velocity model of the crust and upper mantle, *Geophys. J. Int.,* **151,** 88–105.

Shapiro, N.M., Ritzwoller, M.H., Molnar, P. & Levin, V., 2004. Thinning and flow of Tibetan crust constrained by seismic anisotropy, *Science,* **305,** 233–236.

Tanimoto, T., 1995. Crustal structure of the earth, in *Global Earth Physics, A Handbook of Physical Constants*, AGU Reference Shelf 1, pp. 214–224, American Geophysical Union, Washington, USA.

Tarantola, A., 2005. *Inverse Problem Theory*, Siam, Philadelphia, USA.

Tarantola, A. & Valette, B., 1982. Inverse problems = quest for information, *J. Geophys.,* **50,** 159–170.

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

Trampert, J. & Woodhouse, J.H., 2003. Global anisotropic phase velocity maps for fundamental mode surface waves between 40 and 150 s, *Geophys. J. Int.,* **154,** 154–165.

van der Baan, M. & Jutten, C., 2000. Neural networks in geophysical applications, *Geophysics,* **65,** 1032–1047.

Villaseñor, A., Ritzwoller, M.H., Levshin, A.L., Barmin, M.P., Engdahl, E.R., Spakman, W. & Trampert, J., 2001. Shear velocity structure of central Eurasia from inversion of surface wave velocities, *Phys. Earth planet. Inter.,* **123,** 169–184.

Webb, A.R., 1994. Functional approximation by feed-forward networks: a least-squares approach to generalization, *IEEE Transactions on Neural Networks,* **5,** 363–371.

Woodhouse, J.H., 1988. The calculation of eigenfrequencies and eigenfunctions of the free oscillations of the earth and the sun, in *Seismological Algorithms, Computational Methods and Computer Programs*, pp. 321–370, Academic Press, London, UK.

Zhou, Y., Dahlen, F., Nolet, G. & Laske, G., 2005. Finite-frequency effects in global surface-wave tomography, *Geophys. J. Int.,* **163,** 1078–1111.

Zhou, Y., Nolet, G., Dahlen, F. & Laske, G., 2006. Global upper-mantle structure from finite-frequency surface-wave tomography, *J. geophys. Res.,* **111,** B04304.