Inversion of a velocity model using artificial neural networks

Aaron Moya a,b,*, Kojiro Irikura a,c,1

* Disaster Prevention Research Institute, Gokasho, Uji, Kyoto 611-0011, Japan
b Laboratorio de Ingeniería Sísmica, Instituto de Investigaciones en Ingeniería, Universidad de Costa Rica. Apdo. 3620-60 San Pedro de Montes de Oca, San José, Costa Rica
c Disaster Prevention Research Center, Aichi Institute of Technology Toyota, Aichi 470-0392, Japan

ARTICLE INFO

Article history:
Received 6 June 2008
Received in revised form
4 August 2009
Accepted 29 August 2009

Keywords:
Velocity model
Simulation
Neural networks
Synthetic waveforms
Inversion
Algorithm

ABSTRACT

We present a velocity model inversion approach using artificial neural networks (NN). We selected four aftershocks from the 2000 Tottori, Japan, earthquake located around station SMNH01 in order to determine a 1D nearby underground velocity model. An NN was trained independently for each earthquake-station profile. We generated many velocity models and computed their corresponding synthetic waveforms. The waveforms were presented to NN as input. Training consisted in associating each waveform to the corresponding velocity model. Once trained, the actual observed records of the four events were presented to the network to predict their velocity models. In that way, four 1D profiles were obtained individually for each of the events. Each model was tested by computing the synthetic waveforms for other events recorded at SMNH01 and at two other nearby stations: TTR007 and TTR009.

1. Introduction

Scientists use both forward and backward numerical methods to model an earthquake rupture process. Source inversion results become the input to other programs such as finite difference methods (FDM) to refine the velocity model and explain damage distribution areas in terms of ground motion amplification effects. Starting with an initial model, the source parameters are continuously adjusted until the differences between observed and synthetic seismograms are minimal.

The agreement between observation and synthetic data is much influenced by the propagation path, specifically the depths of the layers and also the P and S-wave velocities. Accurate knowledge of the velocity model becomes as important as the source itself for calculating synthetic seismograms. Seismic reflection surveys are a straightforward method to obtain underground velocity values and layer thicknesses. Such surveys often yield many detailed 2D sections that later can be interpolated to construct a 3D velocity model (e.g. Fisher et al., 2003; Stephenson et al., 2000). The surveys, though reliable, have the inconvenience that they are generally expensive and difficult to deploy in highly populated or underwater areas.

Alternatively, the P and S arrival times from small earthquakes can be used to construct velocity models using travel time inversion techniques (Musumeci et al., 2003). Those methods have the advantage that a lot of information is generally available from seismological centers and can be applied over extensive regions. In this study, we consider not only the arrival times to perform the inversion but the waveform itself. The idea of using waveforms to determine a velocity model has been applied by other researchers. Chen et al. (2000) used a combination of 1D and 2D models using whole seismograms to refine a 2D basin structure in the Los Angeles area, California, using genetic algorithms. Satoh et al. (2001) used a trial-and-error technique to fit synthetic waveforms in order to refine a 3D velocity model in the Sendai basin, Japan. Our approach deviates from other techniques in that we use a neural network (NN) as a pattern recognition tool that will be trained to associate waveforms to their specific velocity models.

NN has proven to be a powerful tool in pattern recognition applications (Rogers, 1997). In seismology, they have been applied in tasks such as arrival picking (Dai and MacBeth, 1997), discrimination between earthquake signals and explosions (Del Pezzo et al., 2003; Dysart and Pulli, 1990), and earthquake risk evaluation (Giacinto et al., 1997). Röth and Tarantola (1994) applied NN to determine velocity profiles from seismic sections used in exploration seismology. Their velocity profile consisted of eight layers each of which had a fixed thickness. They generated many synthetic seismograms to train the network by solving the wave equation. They used a ray-tracing approximation and a Ricker wavelet as the source. The network was expected to predict...
the corresponding velocity values for each layer when a given seismic section was presented as input.

Here we also trained an NN using synthetic data, but our purpose was to apply the method actually to earthquake records and not to seismic sections. We used the discrete wave number method (Bouchon, 1981) to generate the synthetic seismograms and included the focal mechanism solution as the source term. Our velocity model consisted of four layers: layer 1 (surface), layer 2, layer 3, and layer 4 (bottom layer). The interfaces between layers 1 and 2, and between layers 2 and 3 were set variables besides the velocities of layers 1, 2 and 3. Layer 4 was set constant at 20.0 km depth.

Four independent 1D models were obtained for previously selected earthquakes recorded at the same station. The earthquakes were located at different depths and azimuths. We were interested in observing whether the models would be very different among themselves or whether they would share enough similarities in order to get a compromised final model. Each model was tested also by simulating other earthquakes not used during the inversion. We compared the synthetic waveforms to the observed records at the target station as well as at TTR007 and TTR009 stations.

2. What are artificial neural networks?

Artificial neural networks (NN) can be better described as collections of mathematical models that mimic the way the brain learns. The brain and nervous system are composed of cells called neurons. It is estimated that the human brain contains as many as 100 billion neurons each one interconnected to thousand neighboring ones forming a biological neural network (Rogers, 1997).

The basic constituents of a neuron are the dendrites, the soma or cell body, the axon, and the synapse (Fig. 1a). Information is transmitted from one neuron to another by means of small electrical pulses. A neuron receives a stimulus through the dendrites. The cell body then decides how to respond, and after that the information is carried out through the axon. At the end of the axon stands the synapse whose task is to transmit the electrical impulses to nearby neurons. Learning occurs as the synapses increase or decrease the signals passed between neurons.

A schematic representation of an NN is shown in Fig. 1b. The network is composed of interconnected units that serve as model neurons. The units can be of type input, hidden, or output. The links connecting the units serve the same function as the synapse in the biological neuron. These links are made of weights that are adjusted by the network during a training phase. The vertical columns of neurons, as shown in the figure, are known as layers. A layer receives signals from the one before it and passes its output to the one which follows.

NN are usually configured for a specific application, such as pattern recognition or data classification. They are also applicable in every situation in which a relationship between the inputs and outputs exists, even when that relationship becomes very complex. NN can represent both linear and non-linear relationships by discerning patterns directly from the data being modeled.

NN learn by example during a training phase in which inputs and outputs are shown to the network sequentially and repeatedly. When numbers are introduced from the input layer (X), they are multiplied by the weights (W) at the interconnecting links and then summed at the hidden neurons (Fig. 2). The resulting sum is passed through a function, such as the sigmoid transfer function, before being sent to the next layer of nodes. The sigmoid transfer function, shown in Eq. (1), forces the sum of products in the range between 0.0 and 1.0. The number that emerges at the output node depends on the input values and the weights assigned to each interconnection.

\[
Y = \frac{1.0}{1.0 + e^{-Wo}}
\]

where Y correspond to the output value and Wo to the summation of the products of the input values and the weights as (in the case

![Fig. 1. Schematic representation of a neuron cell (a) and an artificial neural network (b). Input layer units accept input variables (independent variables). One or more hidden layers of units do majority of processing. Values from hidden layer are processed and presented as an output value at one or more output nodes (dependent variables).](Image)

![Fig. 2. Processing inside network. Input given to network has to be normalized in range 0.0–1.0.](Image)
of Fig. 2) given by

$$Wo = X_1W_1 + X_2W_2 + X_3W_3$$  (2)

At first, the outputs produced by NN consist of arbitrary numbers. Over time, as cases are reintroduced repeatedly hundreds or thousands of times, NN begin to get some of the answers right. The training algorithm continues to change the weights until most of the answers are correct. Training can be stopped when NN reach a certain minimum error (specified by the user) or after a given number of iterations (this is our case).

3. The backpropagation algorithm

NN use many different algorithms for training, we used the backpropagation algorithm. In backpropagation, the input pattern is presented at the input layer and the network runs normally to see what output it produces. The predicted output is compared to what output it produces. The predicted output is compared to the actual output of unit

$$j$$

is the actual output of unit

$$j$$

is the index of a predecessor to the current unit

$$j$$

is a hidden unit

$$i$$

is the index of a predecessor to the current unit

$$j$$

is weight of the link from unit

$$i$$

is a hidden unit

$$j$$

is weight of the link from unit

$$i$$

is a hidden unit

$$j$$

is weight of the link from unit

$$i$$

is a hidden unit

$$j$$

is a hidden unit

$$j$$

is the learning rate (constant value),

$$\delta$$

is the error (difference between predicted and actual output) of unit

$$j$$

is a hidden unit

$$j$$

is a hidden unit

$$j$$

is the index of the current unit,

$$k$$

is the index of the current unit, and

$$k$$

is the index of the current unit.

The learning rate, \(\eta\), is a quantity that specifies the proportion of the error derivative by which the weights will be adjusted during training. If the learning rate is chosen too large then the learning process may diverge, but if the learning rate is too low then convergence can take an extremely long time.

If local minima were along the path of the error surface, the network could get trapped. One way to overcome this problem is by using what is called a momentum term as part of the weights change. Each link has a given value assigned to it, its weight. That value changes every time a new pattern is presented at the input units. The momentum term, \(\mu\), is a percentage of a link value's previous weight change to the actual change. In other words, every time there is a change in the weights (link values), a percentage of the previous weight is added to the new weight.

$$\Delta w_{ij}(t + 1) = \eta \delta o_i + \mu \Delta w_{ij}(t)$$  (4)

where \(w_{ij}\) is weight of the link from unit \(i\) to unit \(j\), \(A\) corresponds to the change added to the previous link's value, and \(\mu\) is the momentum term.

4. Data preparation

As it was mentioned before, in order to train NN, a large data set was needed from which the network could learn to associate a particular waveform to its corresponding velocity model. Because such database was not available from observed records, it was necessary to use synthetic waveforms instead.

To compute synthetic seismograms, we used the discrete wave number method (Bouchon, 1981). The synthetics were computed for four selected aftershocks of the Tottori earthquake recorded at SMNH01 station. The Tottori earthquake occurred on October 6, 2000, and it had a Mw 6.6 (Fig. 3). The strike-slip fault did not reach the surface, but results from waveform inversion indicated that the rupture extended 33 km long by 21 km deep (Iwata and Sekiguchi, 2002; Pulido, 2004; Peyrat and Olsen, 2004) in a NW–SE direction. Two strong motion networks recorded the many aftershocks: KiK-net (which has both borehole and surface...
instruments) and K-NET (only surface instruments). To conduct this study, station SMNH01 was chosen for two reasons:

1. It was located in the middle of the rupture zone. This implied that many earthquakes were recorded there allowing us to retrieve single 1D models from different azimuths.
2. It was a KiK-net station. We could simulate the borehole records instead of the surface ones in order to avoid amplification effects due to surface geology.

Table 1a shows the source parameters for the earthquakes we selected. The selection was based on the proximity from the events to the station along the rupture area, and also on whether or not the waveform had a simple shape in the frequency range of interest. The acceleration records were integrated using SAC2000 (Goldstein et al., 2003) to obtain velocity and filtered in the range 0.3–1.0 Hz in time domain using a bandpass filter.

In determining a velocity model by waveform modeling, we have to consider that the fitting is not only sensitive to the velocity and depth of the layers, but also to the focal mechanism of the earthquake. For that reason, information from the focal mechanism was not set variable, and for each event it was fixed to the value reported by F-net project. The F-net project determines moment tensor solutions for earthquakes nationwide using a broadband network. The values for the strike, dip, and rake for the selected earthquakes are given in Table 1b.

Ito et al. (1995) determined the velocity model around the Tottori region (Table 2a). Based on their model, we constructed a four-layer velocity model from which synthetics were generated semi-randomly by changing the values of the target parameters within the ranges given in Table 2b. Our model considered the same Q values as Ito et al. (1995) where \( Q = 0.5Q_p \).

The target parameters were the depth at the interface between layers 1 and 2 (\( h_{1-2} \)), layers 2 and 3 (\( h_{2-3} \)), and the P-wave velocity of the first (\( v_1 \)), second (\( v_2 \)), and third (\( v_3 \)) layers. The S-wave velocity of the first layer was taken as \( \beta = 2\sqrt{2} \) (Kitsunezaki et al., 1990; Ludwig et al., 1970) and as \( \beta = 2\sqrt{3} \) for the rest. In total we wanted to determine five unknowns.

In our model the bottom layer reaches out to 20 km depth because our deepest event is located at 12.1 km (Table 1). The source was modeled as a ramp function with a half rise-time of 0.5 s. The duration of the synthetic was 38.5 s sampled every 0.15 s (i.e. the delta \( t = 0.15 \) s). If we inverted the three components as such, there would be 256 (38.5/0.15) units per component or 768 input units from the three components per event. In order to reduce that number, we took a time window of approximately 80 points (or 12 s) around the maximum amplitude of the vectorial summation for the three components (Fig. 4) estimated as

\[
O(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}
\]

where \( x(t) \) and \( y(t) \) correspond to the horizontal components of the records and \( z(t) \) to the vertical component. Using 12 s window would be enough to cover the whole S-wave arrival part of the signal.

<table>
<thead>
<tr>
<th>Table 1a</th>
<th>Source parameters for earthquakes used in this study. Event #5 and #6 are not used for inversion but for later testing (see section on results).</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>Event</td>
</tr>
<tr>
<td>1</td>
<td>Oct-07-06:22</td>
</tr>
<tr>
<td>2</td>
<td>Oct-17-22:17</td>
</tr>
<tr>
<td>3</td>
<td>Oct-18-23:39</td>
</tr>
<tr>
<td>4</td>
<td>Nov-03-16:53</td>
</tr>
</tbody>
</table>

* Not used for inversion, only for testing the results.

<table>
<thead>
<tr>
<th>Table 1b</th>
<th>Focal mechanism parameters for earthquakes used in this study.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>Event</td>
</tr>
<tr>
<td>1</td>
<td>Oct-07-06:22</td>
</tr>
<tr>
<td>2</td>
<td>Oct-17-22:17</td>
</tr>
<tr>
<td>3</td>
<td>Oct-18-23:39</td>
</tr>
<tr>
<td>4</td>
<td>Nov-03-16:53</td>
</tr>
</tbody>
</table>

* Not used for inversion, only for testing the results.

<table>
<thead>
<tr>
<th>Table 2a</th>
<th>Velocity model from Ito et al., 1995.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer no.</td>
<td>Depth (km)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>16.0</td>
</tr>
<tr>
<td>4</td>
<td>38.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2b</th>
<th>Depth and velocity ranges for constructing velocity models for inversion. It indicates depth ranges for corresponding layers (i.e. 1–2 means interface between first and second layer) as well as ( \beta ) to S-wave velocity. Values for density and Q factors were taken from Ito et al., 1995 model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer no.</td>
<td>Depth (km)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.1 &lt; ( h_{1-2} ) &lt; 1.0</td>
</tr>
<tr>
<td>3</td>
<td>1.0 &lt; ( h_{2-3} ) &lt; 6.0</td>
</tr>
<tr>
<td>4</td>
<td>20.0</td>
</tr>
</tbody>
</table>
5. Learning a task

In order to properly train the network, we constructed a total set of 1000 velocity models with their correspondent waveforms for each earthquake-station pair. We subdivided those 1000 models into three subsets. We used 800 models for training, 100 models for validating, and 100 models for testing. The meaning of each subset is explained hereafter.

The training data set is made of the models the network actually uses for learning. It is from that data set that the network derives the necessary rules to decrease the global mean squared error (MSE). The MSE is defined as

\[ \text{MSE} = \frac{\sum (x_o - x_p)^2}{n-p} \]  

where \( x_o \) is the actual output (original depth and velocity values), \( x_p \) is the predicted output (predicted depth and velocity values) by the network, \( n \) is the number of observations and \( p \) the number of parameters (weights) (Zell et al., 1991, SNNSv4.2). The network adjusts the weights (or links) between connecting units as the patterns are presented over and over again until training finishes.

The validation data set is composed of models not used during training. Using this set, we can gain insight into the performance of the network while it is still learning. As training goes on, every certain number of epochs (20 in our case) the process is momentarily paused. The validation set is presented to the network, which according to what it has learned up to that point, tries to compute the error. After that, the validating data set is removed and learning proceeds with the training data set. The validation error has no effect on the network’s weights, but it can be used to decide when to stop the learning process and avoid what is called overtraining. Overtraining occurs when an NN starts to memorize only a few patterns from the training data thus loosing its ability to generalize. In other words, overtraining can make the network yield the same result regardless of the input data set—i.e. yield a constant velocity model. By inspecting the error from the validating data set is how we can tell whether the network has started to memorize (overtrain) or not.

The testing data set is a small data set used at the end, when the network has finished the learning process and the weights do not change anymore. It was used to examine the performance of the network in its final stage. From the values predicted by the network (i.e. the depths and velocities of the predicted model), we computed the synthetic waveforms and compared them to the ones from the original models.

![Fig. 5.](image)

**Fig. 5.** Evolution of residual during learning process of NN for selected events. Plots correspond to the validation data set for each one of the events given.

**Table 3**

Comparison of original values and predicted values for selected patterns after network finished training. These models were obtained from testing (synthetic) data set.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Vp layer 1</th>
<th>Vp layer 2</th>
<th>Vp layer 3</th>
<th>Depth layer 1-2</th>
<th>Depth layer 2-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Origin</td>
<td>Predic</td>
<td>Origin</td>
<td>Predic</td>
<td>Origin</td>
</tr>
<tr>
<td>1</td>
<td>577.2</td>
<td>858.4</td>
<td>3519.8</td>
<td>3388.0</td>
<td>6635.9</td>
</tr>
<tr>
<td>12</td>
<td>1316.8</td>
<td>991.3</td>
<td>2432.2</td>
<td>2831.1</td>
<td>7267.0</td>
</tr>
<tr>
<td>36</td>
<td>1035.8</td>
<td>913.6</td>
<td>1698.3</td>
<td>1769.5</td>
<td>6887.8</td>
</tr>
<tr>
<td>74</td>
<td>1108.4</td>
<td>918.1</td>
<td>2125.5</td>
<td>2085.8</td>
<td>7514.9</td>
</tr>
</tbody>
</table>
Inputs and outputs needed to be scaled in the range 0.0–1.0. Stuttgart Neural Network Simulator User Manual, 2000 (Zell et al., 1991) software was used to train the network. We selected a backpropagation algorithm with a momentum term as the training function. We used a learning rate of 0.005 and momentum term of 0.001. Each network was trained for a total of 100,000 epochs or iterations. The error shown in Fig. 5 corresponds to the one from the validating data set as previously discussed. It is interesting to note that the tendency is similar for three events but very different for the Oct-17–22:17 one, which is also the farthest earthquake from the target station.

We used a single hidden layer with 10 hidden units. There are not specific rules to define the number of units in this layer except by trial-and-error. In general, using a large number of hidden units is not recommended since it increases the degrees of freedom (i.e. larger number of links between input–hidden–output layers to be determined) of the task. In this particular case of velocity model inversion, computation time significantly increased when numbers larger than 10 units were used. It was observed that the final error was only slightly smaller when larger the number of hidden nodes was used. Using 10 hidden units yielded satisfactory results.

6. Results

Once training was finished, we examined the performance of the network using the testing data set. We show the comparison between the original values for four selected patterns against the values predicted by using NN in Table 3. Fig. 6 shows the plot of pattern 1 where it is clear that the model proposed by NN is very close to the original one. The fitting is never exact, but the results given by the inversion are similar to the observed ones.

After inspecting the result with the testing data set, we presented the network the actual observed records of the first four aftershocks listed in Table 1. The values of each best fitting model are given in Table 4. Using those velocity models, we computed synthetic waveforms and compared the results with the observed records in Fig. 7.

The fitting between observed and synthetic records is good in all models for the NS and EW components for the S-wave part. The vertical component is more difficult to fit in models A, B, and C. Model D seems to be the one that best explains observation even in the vertical component.

Fig. 8 shows a map where we have drawn the profiles obtained for each of the four events in Table 1. The figure also shows the location of events Oct-12/17:07 and Oct-07/04:59 given in the same table as well as stations TTR007 and TTR009. Using station SMNH001, we calculated synthetic waveforms each event using the four models we obtained in the inversion. The idea was to test how well each model performed with different events. The result

Table 4
Resultant models predicted by NN after presenting observed waveforms as input values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Earthquake’s date</th>
<th>Model A</th>
<th>Model B</th>
<th>Model C</th>
<th>Model D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$ (m/s)</td>
<td>Nov-03/16:53</td>
<td>1839</td>
<td>1209</td>
<td>1985</td>
<td>1927</td>
</tr>
<tr>
<td>$x_2$ (m/s)</td>
<td>Oct-07/06:22</td>
<td>4212</td>
<td>5027</td>
<td>3957</td>
<td>5323</td>
</tr>
<tr>
<td>$x_3$ (m/s)</td>
<td>Oct-18/23:39</td>
<td>5877</td>
<td>5888</td>
<td>6330</td>
<td>6373</td>
</tr>
<tr>
<td>$h_{1-2}$ (m)</td>
<td>Oct-17/22:17</td>
<td>201</td>
<td>123</td>
<td>387</td>
<td>186</td>
</tr>
<tr>
<td>$h_{2-3}$ (m)</td>
<td></td>
<td>3734</td>
<td>4562</td>
<td>1950</td>
<td>6774</td>
</tr>
</tbody>
</table>

Fig. 6. Comparison of results for pattern # 1. Observed and predicted model are similar.

Fig. 7. Waveform fitting between observed (light-gray line) and predicted (solid black line) records for each one of four events given in Table 1.
after using event #5 (Oct-07/04:59) is shown in Fig. 9. The best fit is obtained when synthetics are computed using models A and D.

In a similar way, we calculated synthetics for event #6 (Oct-12/17:07 in Table 1). Unlike event #5, event #6 was also recorded at TTR007 and TTR009. The epicenter of this event was located between models A and B (Fig. 8), so we were expecting to observe a better fitting when using either model. However, as shown in Fig. 10, the fitting was slightly better when using models C or D instead. The amplitudes of the NS components are overestimated in all cases and the EW component at TTR009 shows the largest mismatch.

One possibility for those differences with respect to event #5 could be related to the magnitude and depth of event #6. As Table 1 shows, event #6 is smaller in magnitude and is located at a greater depth than event #5. The amplitude of the waveform should be smaller compared to other earthquakes used for training. NN might not have been properly trained to handle such smaller amplitudes.

The fitting of the four selected events, using their corresponding velocity models, was also tested at nearby TTR007 and TTR009 stations. Those sites are part of the K-Net network and have no borehole instruments only surface ones. Fig. 11 shows the waveform fittings. The results in those stations show clear differences between observation and synthetics except for event #4 (Model D). For event #4, the fitting is good in all three stations except for the NS component in TTR009.
7. Discussion and recommendations

NN were used as a tool to construct a velocity model using waveform data. In this study, we have assumed that the location of the hypocenter was fixed and inverted only the depth and velocities of the layers. However, we have to consider that the original hypocenter determination (the one reported by F-net) depends on the original velocity model. It would seem that this is a loop because hypocenter locations are model dependent and in this work a fixed hypocenter location is used to estimate a velocity model. This would be certainly a problem in the case we were inverting the original model used by F-net (Kubo et al., 2002), but we are not. We are using NN to estimate a different (or a more refined) velocity model in a local area that is constrained to the aftershock region of the Tottori earthquake. The validity of the refined model would be dependent on the seismic network spatial density or to the four profiles we have computed.

Moment tensor is another quantity that is model dependent and using a fixed value for our inversion technique could result in an unstable solution. This can be solved by considering that F-net uses long period surface waves to compute the moment tensor. The estimated number of broadband stations used in the calculation is around 70 according to their homepage (http://www.hinet.bosai.go.jp/f-net/event/dreger.php?LANG=en). Our inversion focuses on the waveform fitting of body waves from strong motion recordings near the rupture area. We are also working in a different frequency band (0.3–1.0 Hz).

NN were able to predict the corresponding velocity models in the 1D case when synthetic data were used. They were also able

Fig. 10. Waveform fitting for event #6 from Table 1 at station SMNH001, TTR007 and TTR009. Synthetic waveform is given by dotted line while observed record by solid line. Amplitude is given in cm/s and time in s. Each model is indicated below each set of stations for the NS, EW, and UD components from left to right, respectively. Stations names are indicated to left of figure. SMNH01 is plotted as SMND01 to indicate that borehole record was used.
to predict a velocity model from the observed records that yielded reasonable synthetic waveforms (particularly the first arrivals of P and S waves). Much of the different capabilities of NN were not explored, such as the use of different training algorithms, neural architecture, number of hidden units, etc., which may improve the results.

According to the different models obtained for the 1D case, model D seemed to be the one that better explained the synthetics and observed records near SMNH01 station. This, however, cannot be conclusive until further analyses are carried out using not only the records at SMNH01 but also the ones at TTR007 and TTR009 in the inversion procedure. The tests presented using earthquakes other than the ones for creating the models (i.e. the four events to train the network) do not show perfect agreement between observed and synthetic records. One of the reasons was probably the low amplitude of the waveform due to the small magnitude and greater depth for one of the earthquakes compared to the ones used for training the net. Those misfits probably suggest that a more realistic 2D or even 3D model could be used instead of a simple 1D model. In fact, the poor fitting for the NS component at TTR009 observed in Figs. 10 and 11 may be pointing in that direction.

Addition of white noise to the synthetic data sets used for training, validation, and testing should be taken into account in

![Waveform fitting for events #1, #2, #3, and #4 from Table 1 at stations TTR007 and TTR009 using velocity models computed from NN. Synthetic waveform is given by dotted line while observed record by solid line. Amplitude is given in cm/s and time in s. Each model is indicated below each set of stations for NS, EW, and UD components from left to right, respectively. Stations names are indicated to left of figure.](image-url)
future inversions with NN. In the present study this was not checked due to the low frequency band we considered.

8. Conclusions

The neural network approach that we have implemented in this study seems to be a good alternative to retrieve velocity models provided certain requirements are met (such as the availability of high quality data, initial information of a velocity model, etc.). The method was presented for a 1D case and it was tested using real (observed) records.

In our neural network approach we tried to obtain what is called continuous output which meant that we did not use neural networks to perform a classification or sorting task but to retrieve actual values. In general, we believe that the answer given by NN is in most cases a good approximation to the true solution as shown by the fitting of the waveforms.

In the case of 1D modeling, we obtained four different velocity profiles for the same station. We observed that the velocity model corresponding to model D seemed to be the one that best explained the data when different earthquakes (at different stations as well) were simulated. However, as the waveform fittings were not exact, just approximations, we decided not to adopt a single velocity model until further testing was conducted. Such testing could involve other profiles in addition to station SMNH01 and nearby stations as well.

In training neural networks for velocity model inversion, it is suggested that repetition of the same pattern be avoided into the training data set. In other words, it should be necessary that the training data set contained as many different patterns as possible for the network to properly generalize from it. Because this was not checked for in this study, we believe that the solutions could be further improved by careful examination of the input data set.

Given the fact that they are generally good at approximating a solution to its true value, combining the result of NN with a genetic algorithm could probably further enhance the effectiveness of the method as well as the use of different learning algorithms (not just backpropagation), different network designs (number of hidden units or hidden layers or full connecting versus shortcut connecting links), and number of parameters to be determined (output units).

It would be necessary to examine whether the learning rate or momentum term could be different for different events or whether it could have any relation with the magnitude and depth.

Acknowledgements

The waveform data used was provided by K-NET, KiK-net which belongs to the National Research Institute for Earth Science and Disaster Prevention (NIED). Seismic moments were obtained from F-net (NEID). We wish to thank two referees for their careful and critical review of the original manuscript. The study was completed while one of the authors (A. Moya) was a student at Kyoto University and it was partially supported by a Grant-In-Aid for Scientific Research from the Ministry of Education, Science, Sports, and Culture of Japan.

References


Downloaded from: http://www.ra.cs.uni-tuebingen.de/SNNS/.