Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2011) 1–10
Procedia Computer Science

Classification of Seismic Windows Using Artificial Neural Networks

Steve Diersen[a], En-Jui Lee[b], Diana Spears[c], Po Chen[b], Liqiang Wang[a]

[a] *University of Wyoming, Dept. of Computer Science, 1000 E. University Ave. Laramie, WY 82071, U.S.A.*
[b] *University of Wyoming, Dept. of Geology and Geophysics, 1000 E. University Ave. Laramie, WY 82071, U.S.A.*
[c] *Swarmotics, LLC, Laramie, WY 82070, U.S.A.*

Abstract

We examine the plausibility of using an Artificial Neural Network (ANN) and an Importance-Aided Neural Network (IANN) for the refinement of the structural model used to create full-wave tomography images. Specifically, we apply the machine learning techniques to classifying segments of observed data wave seismograms and synthetic data wave seismograms as either usable for iteratively refining the structural model or not usable for refinement. Segments of observed and synthetic seismograms are considered usable if they are not too different, a heuristic observation made by a human expert, which is considered a match. The use of the ANN and the IANN for classification of the data wave segments removes the human computational cost of the classification process and removes the need for an expert to oversee all such classifications. Our experiments on the seismic data for Southern California have shown this technique to be promising for both classification accuracy and the reduction of the time required to compute the classification of observed data wave segment and synthetic data wave segment matches.

Keywords:
full-wave tomography, machine learning, artificial neural network, importance-aided neural network

1. Introduction

One of the major concerns today is how to withstand natural disasters. For example, in Southern California a real concern is the ability of high rise buildings, roadways and bridges to withstand earthquakes. Another concern is where to obtain natural resources, for example, where we will find enough oil to drive the World's economy. Understanding the physical properties of the Earth's subsurface is an essential step towards making more secure high rise buildings, roadways and bridges as well as finding future reserves of oil. A recent advance in subsurface imaging technology is *full-wave tomography*, which uses waveform information as a means for providing the subsurface image. Full-wave tomography has shown itself to be more accurate and have higher resolution than other forms of tomography [1]. A current detractor to this form of tomography is the higher computational costs regarding both computer and human processing compared to other forms of tomography.

Seismic tomography is a technique that images the interior of the Earth. In the 1970s, seismologists developed a travel time tomography method that uses the *body wave* arrival times to investigate the lateral heterogeneousness of the Earth's interior [2]. Body waves are energy that propagates through the subsurface of the Earth. Travel time tomography has a low computational cost, but produces lower resolution images than full-wave tomography. Recent

advances in parallel computing technology and numerical methods (e.g. [3, 4, 5]) have made large-scale, three-dimensional simulations of the seismic wave-fields much more affordable, which has opened up the possibility of full-wave tomography (e.g. [6, 7, 8, 9, 10]) and seismic source parameter inversions (e.g. [11, 12, 13]). The main advantage of the full-wave method is in enhancing the resolution and accuracy of the structure model. The more accurate model can be used for both scientific and practical purposes. For example, high resolution models can provide reliable geological structures and/or process interpretation (e.g. [6, 7, 8, 9, 10]), and more accurate ground motion predictions for seismic hazard analysis (e.g. [6, 7]). In full-wave tomography, any type of waveform can be used to improve the model and it is not necessary to identify specific phases of waveforms. However, from the inversion point of view, not every segment on the seismogram is suitable to be used for extracting waveform discrepancies in full-wave inversions. In fact, the successes of some of the recent full-wave tomography studies at different geographic scales (e.g. [6, 8, 9]) are very dependent upon the proper segmentation of the complete seismogram and the proper selection of time-localized waveforms.

The motivation and also the necessity of seismogram segmentation and waveform selection are four-fold. First, the distribution of structural and/or source information on the seismogram is uneven. Consider, for example, the differences between phase and amplitude observations for body-waves that propagate through the body of the Earth. The phase data contained in travel-time measurements made on body-waves are quasi-linear with respect to structural parameters [14], while similar properties are not available for amplitude data. Second, seismogram segmentation reduces possible nonlinear effects due to the interference among different wave groups and allows us to make incremental changes to our structure model. As demonstrated in [7, 9], we can start from fitting portions of the observed seismograms that are not too different from our synthetic seismograms and gradually improve our structure model and try to fit more observed waveforms through iterations. Third, different types of seismic phases on the seismogram can have very different sensitivities to different types of structural parameters and the inverse problem can often be simplified through a judicious choice of the appropriate arrivals and corresponding structural parameters [15]. Fourth, seismogram segmentation allows us to separate signal from noise, which includes signal-generated noise resulting from inadequacies in modeling capabilities.

Our contribution through this paper is the elimination of the majority of the human processing element. This is accomplished through the combination of *CWT* and machine learning. CWT is a continuous wavelet transform that allows us to analyze waveforms in the time and frequency domains. Furthermore, we apply an Artificial Neural Network and a Knowledge-Based Artificial Neural Network to the human processing element of selecting good seismic window segments within the full-wave tomography algorithm. In the experiments presented in this paper, we have shown success when implementing our algorithm for data regarding the Southern California subsurface.

We will discuss the following topics in the remainder of this paper. Section 2 talks about current waveform analysis methods and reviews machine learning methodology. Section 3 describes the related work of machine learning that has been done with regard to seismic data. Section 4 describes the process used to pre-process seismic data for the machine learning algorithms and also describes how we apply machine learning to this problem. Section 5 goes through the design of our neural networks. Section 6 lists the results of our experiments. Section 7 is a discussion on our conclusions and future work.

2. Background

2.1. Current Waveform Analysis Methods

The full-wave tomography workflow (Figure 1) starts with an initial structure model that represents the study area. For our research the study area is the Southern California subsurface. Numerical methods (e.g. finite difference and the spectral-element method) are used to simulate seismic wave propagation in the model, which are then stored as synthetic waveforms containing the ground motion recordings generated. We then compare the observed waveforms and the synthetic waveforms in an effort to find similarities. Waveforms that are found to be similar are then used as the basis of the *kernel* calculation. The kernel is a 3D volume that represents the propagation of seismic energy that generates the waveform within the selected window. We then use an inversion process based upon the kernel and the synthetic and observed waveform differences to update the structural model such that it approaches the true model of the region's subsurface. The structural model is continually refined using this process in an incremental manner.

There are two difficulties in automating the seismogram segmentation and waveform selection procedure. First, the seismogram could be very complex, especially at local to regional distances, where complex wave effects caused by small-scale crustal heterogeneities are significant. Different types of waves recorded by a given sensor could overlap in the time and/or the frequency domains, which pose challenges to the design of automated seismogram segmentation algorithms. Second, the modeling assumptions, in particular, the reference structure and/or source models used in wave-propagation simulations, could be inadequate to describe many of the wave arrivals on the observed seismogram. This type of "signal-generated noise" poses difficulties for both automatically identifying robust features on the seismogram and identifying which specific features contribute to particular aspects of the structure and/or source models.

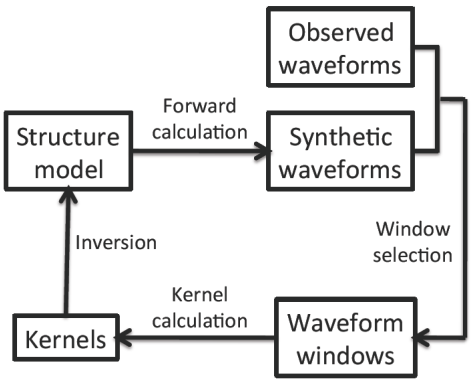


Figure 1: Full-Wave Tomography Workflow

We have developed an algorithm based on CWT that allows us to analyze waveforms in the time-frequency domain. Different from the purely time-domain segmentation method (e.g. [16]), a time-frequency domain algorithm allows extra freedom for separating waves arriving at overlapping time windows but with disjoint frequency domain or time-frequency domain support. Both the observed seismograms and synthetics are segmented in the same way and robust features that exist both on the synthetics and the observed seismograms are identified and extracted automatically using a machine-learning algorithm, which can be tuned to meet specific requirements imposed by an expert.

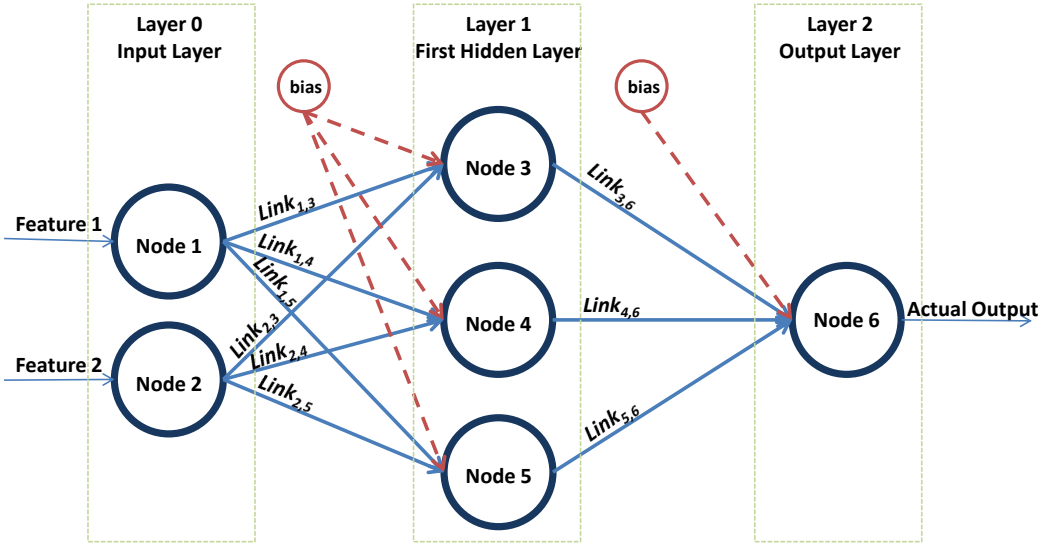
2.2. Machine Learning Methodology

Mitchell [17] defines machine learning as follows, "A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." For this research we used an *Artificial Neural Network* (ANN) and an *Importance-Aided Neural Network* (IANN) [18] to classify segments of observed data wave seismograms and segments of synthetic data wave seismograms as either a match or not a match. Importance-aided neural networks are a type of *knowledge-based artificial neural network*. Knowledge-based ANNs embed expert knowledge into the neural network a priori. The underlying architecture of an IANN is the same as an ANN. Subsection 4.2 discusses how expert knowledge is embedded into the IANN.

The underlying architecture of neural networks are *nodes* connected with directed *links*. Each node performs a three step computation: First, it sums all inputs into the node. Second, it applies an *activation function* to the summation total; where the activation function acts to squash the summed values into a range (in our case) of [0, 1]. Finally, it uses the result of the activation function as the output for the node. For example, Node 3 in Figure 2 first adds the values from $Link_{1,3}$, $Link_{2,3}$ and the *bias* link, then it applies an activation function to the total from the first step and finally the activation function result is sent to Node 6 via $Link_{3,6}$. The value Node 6 receives as input is not the actual output of Node 3, but the product of Node 3's output and $Link_{3,6}$'s *weight*. The link's weight is the "importance" of the sending node's output when determining the receiving node's output. The bias link acts as a threshold value, which must be exceeded in order for an activation function to produce a result greater than its minimum range value.

Individual ANNs can be described in many different ways, for instance, by its *topology*. The topology of a network is referenced as *x*-*y*-*z*, where x is the number of input *layer* nodes, y is the number of nodes in the first hidden layer and z is the number of output layer nodes. For instance, Figure 2 is a neural network with a topology of 2-3-1. The layer is used to describe the fewest number of links connecting an input node to a specific node. In Figure 2 nodes 1 and 2 are in layer 0 (they are input nodes); nodes 3, 4 and 5 are in layer 1 and node 6 is in layer 2.

The number of *features* (attributes) determines the number of input nodes of the network. A feature is a characteristic of the problem that can be represented as a range of values; features from our research include: frequency bandwidth, maximum energy over the whole window and correlation coefficient. The number of *expected outputs*



A standard two layer feed-forward artificial neural network with a 2-3-1 topology. Each example for this network contains exactly two features (attributes) and one expected output (value). Links carry the output of node *i* to node *j*. The bias acts as a threshold value, which must be exceeded for a node to activate. The actual output is generated by the network through the evaluation of a specific example's features.

Figure 2: An Artificial Neural Network

determines the number of output nodes. The set of feature values and expected output values constitute an *example*, which is also given a *class label*. Class labels define the example as being either "positive" (a match) or "negative" (not a match). The examples (including the class labels) for this research were compiled by our human expert (En-Jui) over a period of months.

The neural network learns by comparing the actual output of the network with the expected output of the example. The comparison is done using squared error, $\frac{1}{2}(y-a)^2$, where y is the expected output for the example and a is the actual output of the network. To learn from experience the error must propagate back through the network assigning a portion of blame to the influence of each node. This method, called back-propagation, adjusts each link's weight according to the *learning rate, momentum* and the error of the link's receiving node. The learning rate is how much a link weight is affected by the current example, while the momentum is how much the previous example affects the current link weight. The back-propagation equation is $\Delta\omega_{i,j}(n) = \alpha\delta_j x_{i,j} + \eta\Delta\omega_{i,j}(n-1)$, where $\omega_{i,j}$ is the weight for the link from node i to node j, α is the network's learning rate , δ_j is the error for node j, $x_{i,j}$ is the output from node i to node j, η is the network's momentum value and n is the n^{th} iteration.

A single example will allow a network to learn that example very well, but it will not allow the network to generalize to unseen examples. Therefore, to train a network properly, a large set of representative examples is required; this is called the *training set*. During the training of a neural network the training set may be further divided into a training set and a *test set*. The test set is used to determine when the network has been trained. In order to determine if a neural network generalizes to unseen examples another set, called the *validation set*, is used after training of the network is complete to judge the accuracy of the network.

In the back-propagation method, each example in the training set is evaluated by the neural network. Evaluating each example in the training set is called an *epoch*. After each epoch, the test set examples are evaluated, producing an error for each example. The average of all examples' errors in the test set is used to determine if the network has finished training. In our research two criteria were used to determine if our network was trained. The first criterion was the average error for the test set being within a predetermined tolerance for a specific number of epochs and the second criterion was a maximum epoch count. After the network is considered trained the validation set is evaluated. The

results of the evaluation are the network's classification of each example in the validation set. These classifications are then compared with the expected classification of the validation set to determine the accuracy of the neural network.

3. Related Work on ANNs for Supervised Classification of Seismology Data

We selected ANNs because there are many successful precedents in the literature of applications of ANNs to the supervised classification of seismic data, e.g., seismic waveforms/events. For example, Poulton edited an entire book on the subject of ANNs for geophysical data processing [19]. A wide range of applications is explored in this book, including the application of ANNs for geophysical inverse problems. Lopez et al. extracted wavelet features for classification and then used a multilayer feedforward ANN [20]. The original signals were acoustic and seismic. The ANN performed in a robust manner across a wide range of data characteristics. Enescu decided to use ANNs because of their generalization power [21]. The ANN was found to be especially useful for accurate detection of the arrival time of the first break (i.e., a burst wave on a noisy background) in seismograms. Dowla chose ANNs because they easily incorporate nonlinearities into a solution, and they are easily adaptable and generalize well [22]. His report discusses a wide variety of ANNs that are applicable to the discrimination and classification of seismic data. Romeo found ANNs to be a powerful seismic event classifier that was faster than other methods [23]. A variety of types of ANNs were explored in this research. Baaske et al. [24] and Williamson et al. [25] both successfully applied ANNs for automated seismic facies mapping. ANNs have been quite popular for discriminating natural earthquakes from manmade explosions. For example, Gitterman et al. [26] and Joswig [27] applied ANNs to distinguish earthquakes from nuclear explosions. Benbrahim et al. [28] and Dysart and Pulli [29] both used ANNs for discriminating earthquakes from chemical explosions. Benbrahim et al. got over 80% average classification accuracy, and Dysart and Pulli showed the superior performance of the ANN over an alternative linear discriminant algorithm. Abu-Elsoud et al. used an ANN for discriminating earthquakes from oil prospecting explosions with a 93.7% classification accuracy [30]. Pezzo et al. used a multilayer NN architecture for the discrimination of earthquakes and underwater explosions [31], and got an average classification accuracy of 92%. In addition to discrimination, ANNs have been highly successful for classification tasks. For example, Scarpetta et al. used an ANN with features extracted from spectrograms for classifying local seismic signals and earthquakes [32]. They got an average of 94-100% correct classification on test sets. Sharma and Arora got very low standard error estimates (around 0.1) when using an ANN for earthquke prediction [33], and Murphy and Cercone's ANN classified seismic events with an average accuracy exceeding 98% [34]. In addition to huge efficiency improvements, Langer et al. improved their classification accuracy on a large data set of seismic events from 70% to 80% by going from manual to ANN classification [35]. Furthermore, Wang and Teng's results showed that the accuracy of an ANN was superior to that of a more traditional threshold classifier, especially in the presence of noise, for seismic event detection [36]. Shimshoni and Intrator classified seismic signals with ensembles of ANNs [37]. The reason for an ensemble was to handle low energy and non-stationary signals. Their experimental results confirmed the robustness and accuracy of ANNs on the low-energy, non-stationary data. Likewise, Gravirov et al. demonstrated the robustness and accuracy of ANNs, despite the presence of high levels of noise, for seismic identification [38].

None of the above-mentioned research applied ANNs to the problem of selecting good windows for the refinement of full-wave tomography models. Furthermore, based on an extensive literature search (in addition to the above-mentioned papers), we have not found *any* prior application of knowledge-based ANNs to the field of seismology. The research described in this paper is therefore novel and pioneering.

4. Our Approach

4.1. Data Pre-processing

In our algorithm, the time-domain seismogram is transformed to the time-frequency domain through CWT. In principle, our algorithm can be based on any type of time-frequency transform. The reason we are adopting CWT in our algorithm is its linearity, which results in the absence of interfering cross-terms, and its dyadic pavement of the time-frequency space, which allows efficient and high-resolution representation of the time-frequency content of the seismogram. In our approach, we did not adopt the discrete wavelet transform due to the lack of redundancy in

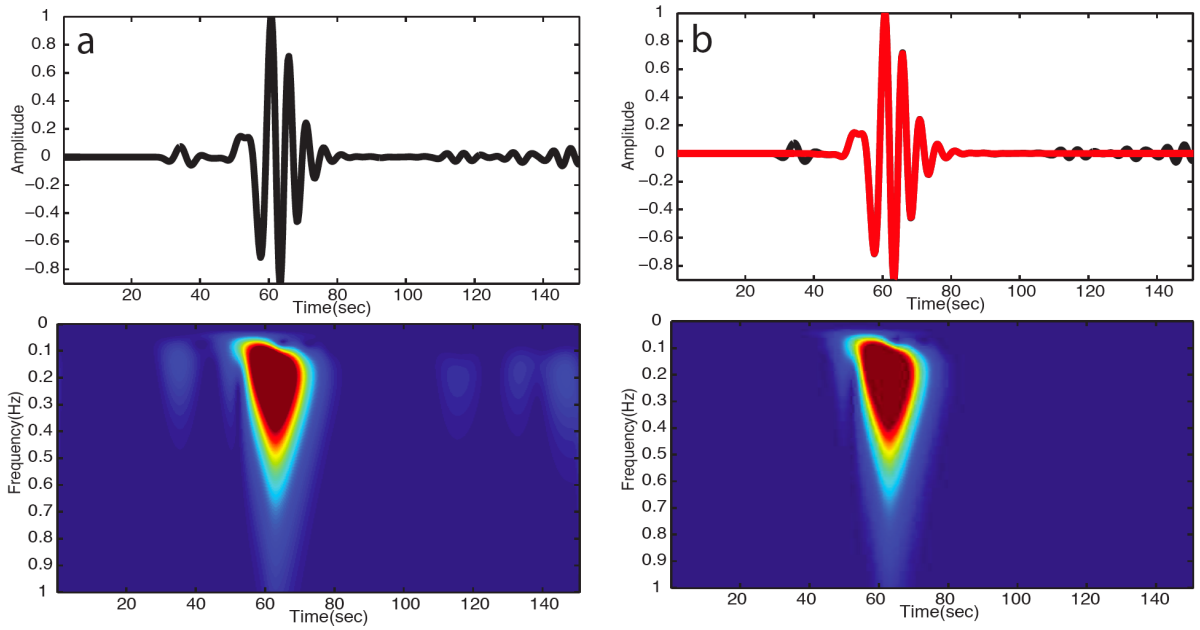


Figure 3: This figure is an example of waveform segmentation for the synthetic ambient-noise Green's function. (a) The time-domain synthetic waveform (top row) and the corresponding scalogram (bottom row) (b) A seed located at the local maxima of the original scalogram was selected and topological watershed code from [39] was adopted for segmentation. In the bottom row, the time-domain waveform of the segmented scalogram is in red.

the discrete wavelet bases, which could reduce the resolution of the resulting time-frequency domain image of the seismogram.

After CWT, seismogram segmentation is then performed automatically on the time-frequency domain scalogram using the topological watershed method, an algorithm designed to cluster all pixels that are connected to the same local extremum [39, 40]. The algorithm is based on the simulation of the immersion process. The two-dimensional scalogram image is reversed and the local maxima become local minima, which are called the catchment basins. The catchment basins are flooded through inlets (seeds) pierced at those local minima. As the flooding progresses, some regions could start to mix and at this point a dam is built to keep the regions separated. As the flood reaches the top of the reversed scalogram, all the dams that have been built during the flooding process form the watershed of the scalogram. The number and the locations of the seeds can be selected in advance. Catchment basins without seeds can be flooded by water coming from a neighboring catchment basin.

By selecting the location of the seeds, over-segmentation due to the existence of noise in the seismogram can be avoided. Figure 3 shows an example of the segmented scalogram and corresponding time-domain seismogram for the synthetic ambient-noise Green's function. The wave arrival selected in the time-frequency domain can be transformed back to the time domain through the inverse continuous wavelet transform (ICWT). The seeds used for segmenting the observed seismogram can be selected in the vicinities of the seed locations used for segmenting the corresponding synthetic seismogram.

4.2. Applying Machine Learning to Seismic Window Classificiation

The results of the data pre-processing discussed in Subsection 4.1 are the features used as the input vector for our machine learning algorithm. There are two advantages for using machine learning algorithms with this problem set. First, machine learning algorithms are able to discern patterns in the solution space, which would be difficult, if not impossible, for human experts to recognize in a timely manner. Second, the machine learning algorithm is faster than the current method. Once the machine learner is trained it will take a fraction of a second to produce a classification compared to the roughly five minutes it takes for an expert to produce the same result.

We used two types of neural networks for the research; the first is an ANN and the second is an Importance-Aided Neural Network (IANN). ANNs work exceptionally well with seismic data (see Section 3). The IANN embeds expert

knowledge into the neural network as Feature Relative Importance (*fri*) [18]. This value indicates an attribute's overall importance when compared to other attributes for determining a correct classification of an example. To determine the *fri* value for each feature, our expert was asked to assign a real-valued number in the range [0,1] based upon the combination of its overall importance to obtaining a correct classification and its importance when compared to all other features. These *fri* values were then used to train the IANN as described in Subsection 5.1. See [18] for more information on *fri* values. The embedding of expert knowledge into ANNs can have two advantages over standard ANNs. The first advantage is that knowledge-based ANNs tend to generalize better than ANNs [17]. The second advantage is that knowledge-based ANNs require fewer examples to achieve the same results as standard ANNs [41].

Table 1: *fri* values

Feature (Attribute)	*fri* value
Correlation Coefficient	0.8
Amplitude ratio of data and synthetic waveforms	0.7
Time shift between data and sythetic waveforms	0.6
Source-reciever distance	0.1
Window start time of data waveform	0.1
Window end time of data waveform	0.1
Start frequency point of data waveform	0.1
End frequency point of data waveform	0.1
Window start time of synthetic waveform	0.1
Window end time of synthetic waveform	0.1
Start frequency point of synthetic waveform	0.1
End frequency point of synthetic waveform	0.1
Mask overlay % (synthetic/data)	0.6
Correlation Coefficient between data waveform and recovered synthetic waveform	1.0
Phase time curve and broadband dt	0.7

The columns are the features contained in the example set and their associated *fri* values. The *fri* values are our expert's opinion of each feature's relative importance in the correct classification of a specific example.

5. Design

5.1. Neural Network Algorithms

We used a standard *feed-forward* ANN with back-propagation. In feed-forward networks all links are unidirectional from a lower numbered layer to a higher numbered layer. We chose the logistic function, $\frac{1}{1+e^{-x}}$, as our activation function due to its range of output, [0, 1]. The derivative of the logistic function, $\frac{e^x}{(1+e^x)^2}$, was used to calculate how much blame each link receives for any error in the network's output during back-propagation. The initial weights for all links were set individually to a small random number in the range [-1,1].

The IANN was identical to our ANN with one exception, using the *fri* values (Table 1) to embed expert knowledge into the network. To embed expert knowledge into the IANN, two adjustments to the mechanics of the network were required. First, the weights for the links from the input layer to the first hidden layer were initialized using one of two methods, as in [18]. The first method was to use the feature relative importance value for the input attribute with a randomly assigned positive or negative value. The second method was to assign a small random value in the range [-0.5,0.5]. To determine whether the first or second method was used for a given link $l_{i,j}$, we first assigned a random number p_j to node j with a value between 1 and the maximum number of attributes, inclusive. Then for each $l_{i,j}$ into node j, we randomly determined if the link was to be given either the first or second method. If the first method was selected, we then checked if the number of links assigned the first method was less than p_j. If the number assigned the first method was less than p_j, then the first method was used; otherwise the second method was used.

The second difference between the ANN and IANN was a change to the back-propagation algorithm [18]. This change affected only the links between the input layer and the first hidden layer. For these links, the *fri* value is multiplied with the learning rate to emphasize the more important attributes. The algorithm for the first link layer then became: $\Delta\omega_{i,j}(n) = \alpha fri_i \delta_j x_{i,j} + \eta\Delta\omega_{i,j}(n-1)$.

5.2. Topology Creation

In order to create the final topology of the neural network, we used a version of k-fold *cross-validation* [17], specifically 10-fold cross-validation. In 10-fold cross-validation the training set is partitioned into 10 roughly equal *folds*. A fold is a unique subset of examples from the training set. The network is then trained using 9 of the folds and tested using the remaining fold. This process is repeated such that each fold is used for testing exactly once. Our training set consisted of 1250 examples. Therefore each fold was exactly 125 examples. Thus each topology had exactly 10 unique runs where each run started with a different set of random link weights, a slightly different example set and a unique test set.

Table 2: Combined results of all topologies

	Average	Std. Dev.	Min Std. Dev.	Max Std. Dev
MSE	0.11427	0.007964	0.003463	0.019391
Avg. <= 5%	103.60	2.282577	101.32	105.88
Avg. > 25%	4.19	0.352235	3.84	4.54

The column headers are the overall values for all topologies tested, from left to right: Average is the average values for all folds of all topologies, Std. Dev. is the standard deviation for all folds of all topologies, Min Std. Dev. is the lowest possible value for a topology to be within one standard deviation and Max Std. Dev. is the largest possible value for a topology to be within one standard deviation. Row 1 is the Mean Squared Error, Row 2 is the number of actual outputs within 5% of the expected outputs and Row 3 is the number of actual outputs greater than 25% of the expected outputs.

The learning rate α was set to 0.5 and the momentum η was set to 0.05. We used two different stopping criteria for determining when the neural network was trained. The first stopping criterion dealt with the average error of one epoch of training. If the average error for an epoch was within 2% then a variable for the number in range was incremented, but if the average error was greater than 2% that variable was set to 0. When the variable reached a value of 200 the network was considered trained and testing would commence. We used a maximum epoch count of 4000 as the second criterion; of the 530 fold runs, only 6 individual runs stopped due to reaching the maximum epoch count.

5.3. Topology Choice

We created a total of 53 topologies, of which 13 had a single hidden layer and 40 had two hidden layers. The topologies with a single hidden layer had the number of hidden nodes ranging from 22 to 34. The 40 topologies using two hidden layers had the first hidden layer ranging from 12 to 19 hidden nodes and a second hidden layer ranging from 6 to 10 hidden nodes. We used three criteria for determining which of the 53 trained topologies we would use. They were the average mean squared error over all folds, the average number of test cases that were equal to or less than 5 percent from the expected output and the average number of test cases that were greater than 25 percent away from the expected output. The three best topologies according to these criteria were 16-19-7-1, 16-12-8-1 and 16-12-7-1. The results for these three topologies are shown in Table 3.

Table 3: Top three topology results

	16-19-7-1	16-12-8-1	16-12-7-1
Avg. MSE	*0.002905*	0.009703	0.006491
Avg. <= 5%	106.33	*110.92*	106.92
Avg. > 25%	3.96	3.88	*3.54*

The column headers are the three best topologies according to the criteria in rows 1, 2 and 3. The Italicized topology is the topology chosen for our experiments. The rows contain the averaged results for each topology's 10 fold runs. Row 1 is the Mean Squared Error, Row 2 is the number of actual outputs within 5% of the expected outputs and Row 3 is the number of actual outputs greater than 25% of the expected outputs. The Italicized value in each row is the best results for all topologies tested.

Table 2 shows the averaged results for all topologies trained as well as the first standard deviation and the range of the standard deviation. Using the results shown in both Table 2 and Table 3, we chose the 16-19-7-1 topology as the topology best suited to our problem set.

6. Experimental Results

The selected ANN and IANN topologies were trained using the entire training set of 1250 examples. For the ANN the initial link weights were individually set to a small random number in the range [-1, 1]. The IANN link weights were set as described in Subsection 5.1 with the *fri* values shown in Table 1. The ANN trained in approximately 5 minutes while the IANN trained in approximately 8 minutes. To test these networks we used a validation set of 504 previously unseen examples, which were applied to each network with the following results: the ANN correctly predicted 99.21% of the examples with 4 false positives and 0 false negatives, while the IANN correctly predicted 99.60% of the examples with 2 false positives and 0 false negatives. Table 4 shows the results of the experiments.

Table 4: Experimental Results

	ANN	IANN
Correct	500	502
False Negatives	0	0
False Positives	4	2
Total	504	504
Percentage Correct	99.21	99.60
Percentage Error	0.79	0.40

Comparison of the experimental results between ANN and IANN. There were 504 total examples in the validation set. The first row is the number of correct classifications for the ANN and IANN, respectively. False negatives are when actual output from either network is negative and the expected output is positive. False positives are when the actual output from either network is positive and the expected output is negative.

7. Conclusions and Future Work

Full-wave tomography gives geophysicists the best opportunity for producing accurate, high resolution images of the Earth's subsurface. Unfortunately, the time required to create an accurate model of a region's subsurface is prohibitive, leading many geophysicists to use other forms of tomography which have lower accuracy and resolution, but require less time for model creation. The goal of this research is to remove the human computational costs associated with the creation of models for full-wave tomography. To this end, we have used an artificial neural network and an importance-aided neural network as a means of classifying seismic windows. A real benefit of this research is the time savings when applied to full-wave tomography. As an example, to classify 1000 examples, an expert (using the 5 minutes per example from Subsection 4.2) would need roughly 3.5 days to classify all examples, whereas either the ANN or the IANN would classify all 1000 examples in less than a second. This is about a 300,000x speed up over the manual classification process.

Our experiments have shown that the ANN is very accurate (99.21% correct classification rate), which demonstrates that machine learning is a very promising approach to classifying segments of seismograms. The IANN is slightly more accurate (99.60% correct classification rate) than the ANN. This is also promising as it demonstrates the value of adding expert knowledge to machine learning algorithms and provides evidence showing that other types of expert knowledge may further increase the accuracy of the machine learner. For instance, as part of our future work we plan to add proportionality knowledge (PANN [41]) or rule-based Horn clause knowledge (KBANN [42]), which should increase the accuracy given larger training and validation sets. We also plan to include, in our experiments, support vector machines (SVM) as well as knowledge-based versions of SVMs corresponding to those used with ANNs.

References

[1] A. J. Brenders, R. G. Pratt, Full waveform tomography for lithospheric imaging: Results froma blind test in a realistic crustal model, Geophysics Journal International.
[2] S. Stein, M. Wysession, An Introduction to Seismology, Earthquakes and Earth Structure, Wiley-Blackwell, 2002.
[3] K. B. Olsen, Simulation of 3-d elastic wave propagation in the salt lake basin, PhD Dissertation, University of Utah, Salt Lake City, USA (1994).
[4] R. Graves, Simulating seismic wave propagation in 3d elastic media using staggered-grid finite differences, Bulletin of the Seismological Society of America 86 (1996) 1091–1106.
[5] D. Komatitsch, Q. Liu, J. romp, P. Süss, C. Stidham, J. H. Shaw, Simulations of ground motion in the los angeles basin based upon the spectral-element method, Bulletin of the Seismological Society of America 94 (2004) 187–206.
[6] P. Chen, L. Zhao, T. H. Jordan, Full 3d tomography for the crustal structure of the los angeles region, Bulletin of the Seismological Society of America 97 (2007) 1094–1120.
[7] C. Tape, Q. Liu, A. Maggi, J. Tromp, Adjoint tomography of the souther california crust, Science 325 (2009) 988–992.

[8] C. Tape, Q. Liu, A. Maggi, J. Tromp, Seismic tomography of the southern california crust based on spectral-element and adjoint methods, Geophysical Journal International 180 (2010) 433–462.
[9] A. Fichtner, B. L. N. Kennett, H. Igel, H. P. Bunge, Full seismic waveform tomography for upper-mantle structure in the australasian region using adjoint methods, Geophysical Journal International 179 (2009) 1703–1725.
[10] A. Fichtner, B. L. N. Kennett, H. Igel, H. P. Bunge, Full waveform tomography for radially anisotropic structure: New insights into present and past states of the australasian upper mantle, Earth and Planetary Science Letters 290 (2010) 270–280.
[11] L. Zhao, P. Chen, T. H. Jordan, Strain green's tensors, reciprocity, and their applications to seismic source and structure studies, Bulletin of the Seismological Society of America 96 (5) (2006) 1753–1763.
[12] P. Chen, T. H. Jordan, L. Zhao, Resolving fault plane ambiguity for small earthquakes, Geophysical Journal International 181 (2010) 493–501.
[13] A. Fichtner, H. Tkalčič, Insights into the kinematics of a volcanic caldera drop: Probabilistic finite-source inversion of the 1996 bárdarbunga, iceland, earthquake, Earth and Planetary Science Letters 297 (2010) 607–615.
[14] Jordan, T. H.. Earth structure from seismological observations. In Physics of the Earths Interior, Proc. Int. School Phys. Enrico Fermi, 77, eds Dziewonski, A. M. & Boschi, E., Soc. Italiana de Fisisca, Bologna, 1980, pp. 1-40.
[15] A. Sieminski, J. Trampert, J. Tromp, Principal component analysis of anisotropic finite-frequency sensitivity kernels, Geophysics Journal International 179 (2009) 1186–1198.
[16] A. Maggi, C. Tape, M. Chen, D. Chao, J. Tromp, An automated time-window selection algorithm for seismic tomography, Geophysics Journal International 178 (2009) 257–281.
[17] T. M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[18] R. A. Iqbal, Empirical learning aided by weak domain knowledge in the form of feature importance, Eprint arXiv:1005.5556.
[19] M. M. Poulton, Computational Neural Networks for Geophysical Data Processing, Volume 30 of Seismic Exploration, Pergamon/Elsevier, 2001.
[20] J. Lopez, H. H. Chen, J. Saulnier, Target identification using wavelet-based feature extraction and neural network classifiers, Tech. rep., CYTEL Systems, Inc. (1999).
[21] N. Enescu, Seismic data processing using nonlinear prediction and neural networks, in: Proceedings of the IEEE NORSIG Symposium, 1996.
[22] F. U. Dowla, Neural networks in seismic discrimination, Tech. rep., Lawrence Livermore National Laboratory (1995).
[23] G. Romeo, Seismic signals detection and classification using artifical neural networks, Annali Di Geofisica XXXVII (3) (1994) 343–353.
[24] U. P. Baaske, M. Mutti, F. Baioni, G. Bertozzi, M. A. Naini, Using multi-attribute neural networks classification for seismic carbonate facies mapping: A workflow example from mid-cretaceous persian gulf deposits, in: Seismic Geomorphology: Applications to Hydrocarbon Exploration and Production, The Geological Society of London, 2007, pp. 105–120.
[25] A. Williamson, R. Walia, R. Xu, M. Koop, G. Lopez, Quantitative interpretation of neural network seismic facies - oriente basin ecuador, in: Proceedings of the Canadian Society of Exploration Geophysicists, 2003.
[26] Y. Gitterman, V. Pinsky, A. Shapira, Spectral classification methods in monitoring small local events by the israel seismic network, Journal of Seismology 2 (1998) 237–256.
[27] M. Joswig, Seismic signal classification: Preprocessing, Tech. rep., Tel Aviv University (1999).
[28] M. Benbrahim, A. Daoudi, K. Benjelloun, A. Ibenbrahim, Discrimination of seismic signals using artificial neural networks, in: Proceedings of the World Academy of Science, Engineering and Technology 4, 2005.
[29] P. S. Dysart, J. J. Pulli, Regional seismic event classification at the noress array: Seismological measurments and the use of trained neural networks, Bulletin of the Seismological Society of America 80 (68) (1990) 1910–1933.
[30] M. A. Abu-Elsoud, F. E. Z. Abou-Chadi, A.-E. M. Amin, M. Mahana, Classification of seismic events in suez gulf area, egypt, using artificial neural network, in: Proceedings of the International Conference on Electrical, Electronic and Computer Engineering, 2004.
[31] E. D. Pezzo, A. Esposito, F. Giudicepietro, M. Marinaro, M. Martini, S. Scarpetta, Discrimination of earthquakes and underwater explosions using neural networks, Bulletin of the Seismological Society of America 93 (1) (2003) 215–223.
[32] S. Scarpetta, F. Giudicepietro, E. C. Ezin, S. Petrocino, E. D. Pezzo, M. Martini, M. Marinaro, Automatic classification of seismic signals at mt. vesuvius volcano, italy, using neural networks, Bulletin of the Seismological Society of America 95 (1) (2005) 185–196.
[33] M. L. Sharma, M. K. Arora, Prediction of seismicity cycles in the himalayas using artificial neural networks, Acta Geophysica Polonica 53 (3) (2005) 299–309.
[34] M. D. Murphy, J. A. Cercone, Neural network techniques applied to seismic event classification, in: Proceedings of the 25th Southeastern Symposium on Systems Theory, 1993, pp. 343–347.
[35] H. Langer, S. Falsaperla, T. Powell, G. Thompson, Automatic classification and a-posteriori analysis of seismic event identification at soufriere hills volcano, montserrat, Journal of Volcanology and Geothermal Research 153 (2006) 1–10.
[36] J. Wang, T.-L. Teng, Artificial neural network-based seismic detector, Bulletin of the Seismological Society of America 85 (1) (1995) 308–319.
[37] Y. Shimshoni, N. Intrator, Classification of seismic signals by integrating ensembles of neural networks, IEEE Transactions on Signal Processing 46 (1996) 1194–1201.
[38] V. V. Gravirov, K. V. Kislov, T. V. Ovchinnikova, Neural network method for identification of earthquake phases in increased noise level conditions, Geophysical Research Abstracts, EGU2010-2434-1 12.
[39] L. Vincent, P. Soille, Watersheds in digital spaces / an efficient algorithm based on immersion simulation, IEEE Trans. PAMI 13 (6) (1991) 583–598.
[40] G. Bertrand, On topological watersheds, Journal of Mathematical Imaging and Vision 22 (2-3) (2005) 217–230.
[41] G. Levine, U. Kuter, K. V. Sloten, G. DeJong, D. Green, A. Rebguns, D. Spears, Using qualitative domain proportionalities for learning mission safety in airspace operations, in: Proceedings of the IJCAI'09 Workshop on Learning Structural Knowledge From Observations, 2009.
[42] G. Towell, J. Shavlik, Knowledge-based artificial neural networks, Artificial Intelligence 70 (1994) 119–165.