Seismic Data Processing Using Nonlinear Prediction and Neural Networks

Nicoleta Enescu

Tampere University of Technology,
Signal Processing Laboratory
P.O. Box 553, Fin 33101 Tampere, Finland
E-mail: enescu@cs.tut.fi

ABSTRACT

In this paper a simplified but formalized version of the "human interpreter approach" for model driven seismic interpretative processing is proposed. We consider the task of recognizing as such various seismic events and accurately timing their arrival. The novelty of our approach consists in relating nonlinear prediction and neural networks for analyzing real data sets. A seismic event being inherently unpredictable, the performance of a sensible predictor will be perturbed and this is what we will count on. Based on the generalization power of a neural network, the goal is to allow a certain degree of automatic traceability of the criteria used for the identification and timing of seismic events.

1. INTRODUCTION

Model driven seismic interpretative processing requires consistent velocity models of the investigated rock volume. In turn, these models rely on the ability to recognize as such various seismic events and on the accurate timing of their arrival. The capability of various computer aided waveform recognition and automated time picking methods has been well proven for synthetic signals. However, when real data are analyzed, it is still common to rely on the experience of human interpreters. The often invoked reason is that stable criteria for detecting events and determining their arrival times must be kept, in spite of potentially large variations of the signal characteristics through the data set.

The main disturbing factors with real data are:

- Poor signal-to-noise ratio, SNR_{trace}, to be defined latter, due to high absorption in seismically "opaque" media;
- The presence of accidental bursts of man-made noise;
- Changes in the phase and amplitude of the signals while traveling through the medium, due to scattering on micro-inhomogeneities and multipath effects;
- Variation through the data set of the frequency and amplitude of the signals, due to different coupling to the medium for different positions of the sources and detectors.

The combined effect of the factors listed above is that the automatic identification of the seismic events and the result of the corresponding time picking exercise tend to depend significantly on the "path" followed through the data set, that is the order in which the traces in the data set are processed. The approach proposed in this paper, employing nonlinear prediction and neural networks, is an attempt to produce a simplified but formalized version of the "human interpreter approach" and thus allow a certain degree of automatic traceability of the criteria used for the identification and timing of seismic events.

1.1. Description of Data Acquisition

In the following we will analyze a real seismic data set. This consists of 64 traces obtained from measurements at eight detectors of the waves produced by eight sources. The length of one trace is 8.192 ms, at a sampling rate of $f_s = 125\ kHz$. The trigger delay was $t_d = -400\ \mu s$. Data acquisition was made using on-line A/D conversion. A preliminary spectral analysis followed by "soft" band-pass filtering was performed on the raw data set. At this point, based on the maximum frequency in the signal and velocity analysis, we determine the period of the wave we are interested in, here the P-wave. Two typical examples of the waveforms that will be further processed are presented in Figure 1.

2. NONLINEAR PREDICTION BASED PREPROCESSING

As we shall point out in the following, one way of detecting a seismic event given a measurement set is through nonlinear prediction. This will represent one

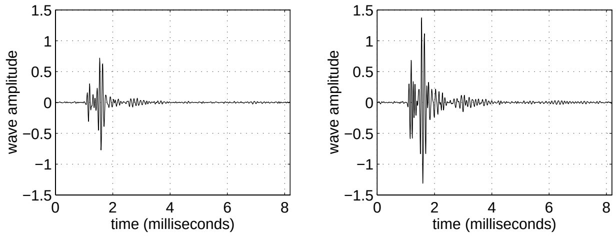

Figure 1: *Examples of real seismic waves.*

step in the processing scheme. Different traces can be independently analyzed by passing them through a nonlinear predictor. If we deem that the apparition of a burst represents a fast transition in the signal, it is reasonable that a predictor will try to "cope" with this, but its performance will be first disturbed. If we assume a rather complex prediction model (for example a second order Volterra polynomial), the disturbance produced by the apparition of the burst will be quite strong and this is exactly what we are looking for. In other types of applications such a behavior could be disturbing but here it is a plus. A complex (but not too complex) nonlinear predictor is found to be more sensible to fast transitions in the signal than its linear counterpart.

Let us consider a time series $\{x(n) \mid n = 1, \ldots, N\}$. The model used for nonlinear prediction of a trace signal is linear in parameters and is described by:

$$\hat{x}(n) = h_0 + \sum_{i_1=1}^{M-1} h_1(i_1)x(n-i_1) + \sum_{i_1=1}^{M-1}\sum_{i_2=i_1}^{M-1} h_2(i_1,i_2)x(n-i_1)x(n-i_2), \quad (1)$$

where M represents the length of the processing window. This is chosen such that it corresponds to approximative one period of the signal to be predicted.

The nonlinear predictor was implemented adaptively. The sum of squared prediction errors, $e(n) = x(n) - \hat{x}(n), n = 1, \ldots, N$, was iteratively minimized using the least-squares approach. For estimating the prediction parameters we used a QR-decomposition based RLS algorithm.

To detect the apparition of a burst we test the "predictability" of the seismic wave by estimating an "instantaneous" signal-to-noise ratio, $SNR(n)$ curve. To define such a performance indicator is a tricky task. We will compute the $SNR(n)$ as the ratio of the average power of the desired signal, $x(n)$, and the average power of the prediction error, $e(n)$, over few (one to three) periods of the signal after (length h) and respectively before (length l) the current sample, n, in decibels:

$$SNR(n) = 10\log_{10}\frac{\sum_{i=n-l}^{n+h} x^2(i)}{\sum_{i=n-l}^{n+h} e^2(i)} \text{ (dB)} \quad (2)$$

The $SNR(n)$ curve, obtained this way, will have a rapid variation when the first break occurs and therefore the location of the maximum of its derivative will (indicate) be close to the position of the first break, as it is shown in Figure 2. For better illustrating this we construct a step signal whose step transition will correspond to the maximum of the $SNR(n)$ derivative.

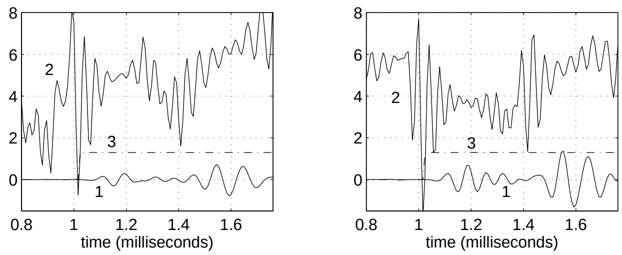

Figure 2: *Illustration of the variation of the SNR(n) (dB) curve (2) given the test signals (1), that were presented in Figure 1. The transition of the step signal (3) indicates the apparition of the burst.*

Because the $SNR(n)$ curve estimated by (2) will have rapid variations due to noisy character of the predicted trace signal, we will compute the derivative of the $SNR(n)$ just on a short interval. The bounds of this interval can be determined through velocity analysis, if we assume as known the velocity range of the P-wave propagation. In this way we reduce the sensibility of the detection of arrival time for noisy signals.

From experimental investigations resulted that in the case of "clean" test signals (with good SNR_{trace}) it is actually possible to determine accurately the position of the first break based only on nonlinear prediction. The signal to noise ratio that we refer to here is characteristic to a trace and is estimated as the ratio of the power of the signal and the power of the noise on the trace, in decibels:

$$SNR_{trace} = 10\log_{10}\frac{\sum_{i=t_0-w}^{t_0} x^2(i)}{\sum_{i=t_0}^{t_0+w} x^2(i)} \text{ (dB)}, \quad (3)$$

where t_0 is the estimated arrival time and w is the width of the window we consider for computing the SNR_{trace}. When considering noisy signals, the detection is not exact but allows us to retrieve an approximate position of the burst, which will be good enough if we consider the nonlinear prediction stage just as preprocessing before detection using a neural network.

3. NEURAL NETWORKS BASED ANALYZE

For an accurate detection of the arrival time of the first break (a burst wave on a noisy background) we consider appropriate to use a neural network. This choice is justified by a considerable amount of data that is to be taken into account for seismic interpretative processing. The proposed architecture is that of a three layers feed-forward neural network trained using the fast back-propagation algorithm.

3.1. The Proposed Processing Chain

Given a data set, which in practice could be represented by thousands of traces, we regard of interest the following processing chain:

1. From the initial data set, isolate (using SNR based criteria and clustering) a representative set of "clean" traces (10 to 20% from the whole set),

2. Detect the arrival time of the burst wave by using nonlinear prediction on the traces in the representative set,

3. From each of these traces isolate a window of interest of 5 or 6 periods centered on the detected time of arrival. Construct the corresponding target signals as described below and form in this way the basic training set for the neural network.

4. Train a neural network to match the input signals in the training set with the target ones.

5. Use the neural network resulted from the training process for the estimation of the arrival times for the whole set of traces to be analyzed.

3.2. Designing the Training Set and Network Architecture

Few considerations about choosing a window of interest on a trace and constructing the target signal for it are in order now. First, for the neural network to "learn" the characteristics of the noise segment and those of the signal segment and, moreover, to accurately "recognize" the transition between them, it is necessary to present to the network's input a window containing both segments, noise and signal. The length of the segments is to be considered in number of periods of the signal. Taking two periods as the length of the noise segment and three as the length of the actual signal, has been experimentally proved to be "enough". The design of the neural network architecture should take into consideration economical considerents regarding the complexity.

For our experimental investigations we considered that each layer of the network consists of 50 neurons,

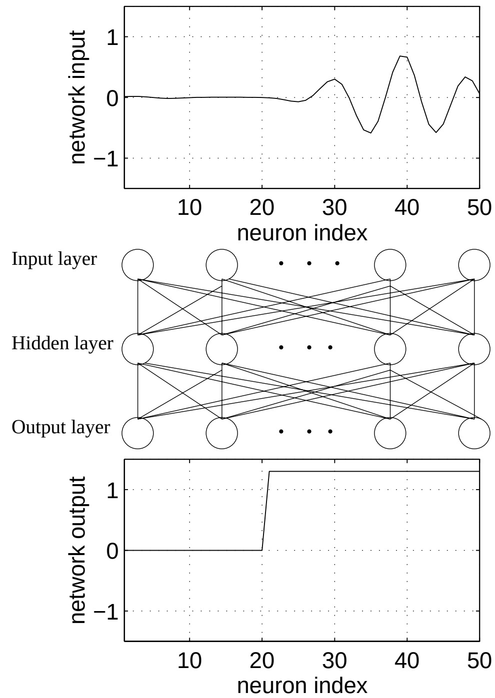

Figure 3: *A three layers feed-forward neural network architecture together with examples of input and target signals.*

which correspond to processing a 50 samples window of the current signal. The window of interest is one that contains the begining of the burst wave. As it was already explained, in order to "isolate" the probable range where the burst occurs one can preprocess the signal through nonlinear prediction. The correspondence between the 50 neurons length of one layer and the 5 periods length of the signal window, obtained as explained above, may be assured through a resampling process of the original signal. The resulted signal will be considered as an input to the network.

The target signal for one input to the network will be a step signal with the step transition in the transition point between the noise and signal segments of the input. Isolating the input signals for the neural network and constructing the target signals as just described, we obtain the basic training set.

An input and an output of the network, together with a three layers architecture are exemplified in Figure 3.

3.3. Training the Neural Network

The neural network was trained using a matrix implementation of the backpropagation (BP) algorithm. As the BP algorithm we considered a classical but effective gradient descent method with adaptive learning rate [3]. By rearranging the BP algorithm in order to operate only with matrices it was possible to use fast matrix multiplication techniques. The result represents the so called Matrix Back Propagation (MBP) algorithm [1]. Using this algorithm and the training set, we adaptively estimated the weights of the connections among the neurons of different layers. The learning process was performed during 10 different runs, with different starting points. The final configuration of the network used the weights obtained from the "best" run. The MBP being a gradient descent method, tries to minimize a cost function, in our case the mean square error between the target and the output of the network.

It is reasonable to think that a "small", but representative, training set is appropriate for assuring fast adaptation of the network and "performant learning" of the signals' characteristics.

4. SIMULATIONS

The basic training set consisted of 64 segment traces of 50 samples each. This was extended by presenting the input segments with different relative ordering, with and without added noise (uniform distributed in the range $-1 \cdots 1$ (33.33%, reported to the dynamic range of the signal, which corresponds to a mean of ≈ 5.95 (dB) SNR between the original and noisy versions), with and without random shifting of the position of the first break (few samples in future or in past). The extended set of target signals was constructed to correspond to the signals in the input set.

Let us note that the reliability of the estimates obtained using a neural network is in direct relation with the design of the training set (when assuming a competitive design of the neural network itself). We choose to extend the basic training set, as just described, in order to better simulate the case of a big initial data set, given that we considered here just 64 initial traces.

After adaptation of the network, the weights associated with the connections among neurons of different layers will reflect the knowledge extracted by the neural network from the training set. At this point, it is possible to optimize the configuration of the network, using a pruning algorithm [3]. In the testing stage, in order to determine the position of the first break for a certain trace, as a sample index, one simply should estimate the maximum of the first derivative of the output signal of the network.

Assuming that the training was successful, we can test the accuracy of the neural network response for signals that were not presented in the training set. In order to do so, we constructed three different testing sets and presented them to the input of the neural network. We started from the basic set of 64 traces and extended it in a similar manner as the training set. The levels of the added noise were 8.33%, 16.67% and 25%. The shifting of the position of the first break was again random, as was the relative ordering of the traces in the resulting testing sets. Some typical results, for the three testing sets are shown in figure 4. In the case of big initial data set, as we already mentioned, it will be possible to isolate "clean" and representative traces to form the basic training set and also, based on SNR_{trace} estimates, to construct different testing sets. They may be used for cross-validation of the network performance during learning process and therefore improve the overall performance of the network.

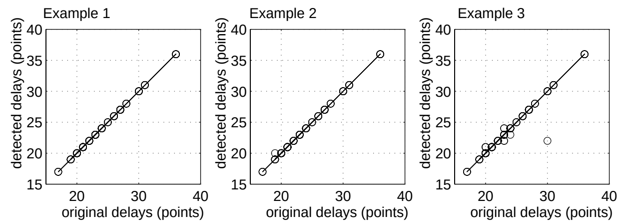

Figure 4: *Original versus detected delays (expressed as positions of first sample of the burst wave). Detection was made using a 3 layers NN trained by MBP. This figure presents the results for three different noise levels: Example 1* - 8.33%, *Example 2* - 16.67% *and Example 3* - 25%.

5. CONCLUSIONS

Since automatic identification of the seismic events is still a problem, in this paper we proposed an automated processing chain for seismic real data analysis, based on nonlinear prediction and neural networks, as an aid to the human interpreter. We presented the basic design considerations for a successful implementation of this approach and illustrated some typical results obtained through computer simulations.

ACKNOWLEDGEMENTS

We would like to thank to Călin Cosma from Vibrometric Oy, Helsinki, Finland, for initiating this research and providing us with sets of real data.

REFERENCES

[1] Davide Anguita "Matrix Back Propagation - An efficient implementation of the BP algorithm" *Technical Report*, DIBE - University of Genova, November 1993

[2] Yuzuru Ashida "Data Processing of Reflection Seismic Data by Use of Neural Networks", *to appear in*

[3] S. Haykin *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, 1994

[4] V.J. Mathews "Adaptive Polynomial Filters", *IEEE Signal Processing Magazine*, vol. 8, no. 3, July, 1991