A robust neural network based approach for microseismic event detection

Jubran Akram*, Oleg O. Ovcharenko and Daniel B. Peter, Extreme Computing Research Center (ECRC), Physical Science and Engineering Division (PSE), King Abdullah University of Science and Technology (KAUST), Saudi Arabia

SUMMARY

We present an artificial neural network based approach for robust event detection from low S/N waveforms. We use a feed-forward network with a single hidden layer that is tuned on a training dataset and later applied on the entire example dataset for event detection. The input features used include the average of absolute amplitudes, variance, energy-ratio and polarization rectilinearity. These features are calculated in a moving-window of same length for the entire waveform. The output is set as a user-specified relative probability curve, which provides a robust way of distinguishing between weak and strong events. An optimal network is selected by studying the weight-based saliency and effect of number of neurons on the predicted results. Using synthetic data examples, we demonstrate that this approach is effective in detecting weaker events and reduces the number of false positives.

INTRODUCTION

Microseismic monitoring is useful for evaluating hydraulic fracturing in unconventional resource studies. The microseismicity due to fracturing is continuously recorded on receivers placed in borehole, near-surface and/or surface settings. During the data processing stage, microseismic events in the form of P- and/or S-signals are extracted from the continuously recorded waveforms, a process known as event detection, and later located in the Euclidean space for fracture geometry and subsequent interpretation.

In practice, event detection is performed by automatic algorithms which are typically based on some attribute-computation in moving windows (for instance, STA/LTA; Allen, 1982), waveform cross-correlation (Song et al., 2010) or on artificial neural networks (ANNs) (Maity et al., 2014). An ANN based approach use effective models from data-driven learning for solving complex and ill-posed problems (Gentili and Michelini, 2006; Maity et al., 2014). Therefore, it can provide better microseismic event detection as compared to conventional algorithms for low S/N and complex waveforms.

Previously, a non-linear, multilayer and feed-forward network has been used in many studies for event detection and arrival picking (Dai and MacBeth, 1995; Gentili and Michelini, 2006; Maity et al., 2014). Here, we use a similar ANN but our main goal is to study the effect of various components of a neural network on its performance. In particular, we look at the effect of number of neurons in the hidden layer, weight distribution for network layers and the behavior of error minimization curve in training, testing and validating the network. Based on this, we choose an optimal network for event detection on the entire dataset. This network generates a probability curve as output which provides the confidence level for each detection.

In the subsequent sections, we describe brief details on theory and show some synthetic examples to demonstrate our results.

THEORY

An ANN provides a mathematical computing framework to model the operations of biological neural systems. McCulloch and Pitts (1943) set the platform for neurocomputing by demonstrating that any logical or arithmetic operation could theoretically be performed by a collection of connected processors that are loosely modeled on the organization of the brain (McCormack et al., 1993). A common building block of a neural network model is a simple processing unit called neuron, and the average human brain has approximately 10^{11} neurons, each of which have a soma (main body of the nerve cell) and an axon (a single path for transmitting signal). A neuron receives and combines signals from many other neurons through the dendrites which are filamentary input paths. These dendritic trees are connected with soma. A synapse is the connection between a neuron's axon and another neuron's dendrite. An electrical signal is transmitted along axon when a neuron is excited above a certain threshold (Kartalopoulos, 1996).

In an artificial neural network, many neurons are interconnected based on the choice of network topology (for instance, single-layer, multi-layer, feed forward, feed-backward etc.). Figure 1 shows a generalized sketch of a non-linear, single-layer and feed-forward network. The input features ($x_1, \dots, x_M$) are applied in the input layer (M nodes) and the outputs ($y_1, \dots, y_K$) are obtained from the output layer (K nodes). The intermediate layer is called hidden layer. There can be more than one hidden layer in a network. Each input x_i is weighted (multiplied by w_i) before reaching a neuron as depicted in the figure by different thicknesses of connections. Moreover, it includes a bias term which is used to model a threshold exceeding which a neuron produces a signal. An activation function, R , acts on the produced signal and an output, a , is generated which constitutes input to other neurons in the network. Here, we use a log-sigmoid activation function (as desired output ranges between 0 and 1) which is non-linear and is described as

$$a = \frac{1}{1 + e^{-n}}, \quad (1)$$

where n is the net input which goes into transfer function (Kartalopoulos, 1996; Hagan et al., 2014).

The selection of input features plays a key role in the network training and performance. We, therefore, choose the following amplitude and polarization based attributes for use in our analysis:

- Average of absolute amplitudes for each waveform component

Neural network based event detection

- Variance for each waveform component
- Energy-ratio for each waveform component
- Polarization-rectilinearity measure from 3-C waveforms

For consistency, we compute these features using moving windows of identical lengths. Other attributes such as STA/LTA and higher-order statistics (skewness and kurtosis) are ignored because LTA, skewness and kurtosis require a longer window size for providing stable values.

Input Layer

Figure 1: A feed-forward neural network with single hidden layer. Input layer comprises of M feature nodes whereas hidden layer and output layers have L and K nodes, respectively. Each node connection between layers has a specific weight as depicted in figure with variable thickness line-arrows. A log-sigmoid activation function is used in this study.

We use a weight-based saliency measure to optimize the network architecture, in particular the number of neurons for the hidden layer (Steppe and Bauer, Jr., 1997). In this approach, we study the sum of square of weights for the network layers. The input feature weights are computed as

$$W_i^{IF} = \sum_j (w_{ij})^2 \quad (2)$$

The weights associated with each single node (neuron) in the hidden layer corresponding to all features are determined as

$$W_j^{IH} = \sum_i (w_{ij})^2 \quad (3)$$

Similarly, the weights from hidden layer nodes to output nodes are obtained as

$$W_j^{HO} = \sum_k (w_{jk})^2 \quad (4)$$

In this case, we use only a single output node, which produces a relative probability response (between 0 and 1). Therefore, W^{HO} is a vector of size L . The distribution of weights allows easy identification of important features and layer nodes for a specific network. As a result, the number of features or hidden layer nodes can be adjusted to optimize the network's performance. Here, we are interested in finding a configuration (number of neurons in the hidden layer) that uses information from all input features as each is capable of identifying signal intervals. In addition, number of neurons parameter is optimized by evaluating the predicted output and the behaviour of performance measure (mean-squared error between desired and predicted outputs) for the training, testing and validation steps.

Figure 2: a) Acquisition geometry for synthetic data. A vertical array of 50 receivers is used to record microseismic events which are generated for different magnitude levels and moment tensors. b) An example of waveforms with gaussian noise for a 3-C microseismic event.

APPLICATION ON A SYNTHETIC DATASET

Figure 2a shows the acquisition geometry and microseismic source locations. A vertical array with 50 receivers is used for recording synthetic data from the source locations. The distribution of event size as depicted in the figure indicates different magnitude levels (M_w ranges between -2 and 1). Different

Neural network based event detection

moment tensors are also used randomly for the generation of synthetic waveforms using a homogeneous velocity model (P-wave velocity (α) = 5000m/s, S-wave velocity (β) = 2941m/s, density (ρ) = 2200kg/m³).

Figure 2b shows the waveforms after the addition of gaussian white noise. In this example, the S-wave signal is visible on x- and z-components whereas P-wave signal is visible on x-component only. Here, we are interested in identifying the time windows containing the P- and S-wave signals. Moreover, we want to use a target measure which provides a robust distinction between weak and strong signals. For this purpose, we prepare the desired output for the training dataset by assigning relative probabilities (between 0 and 1) based on the S/N of the waveforms as well as the respective response of feature attributes. Figure 3 shows an example of input features and target probability for the training dataset. Since S-wave signal is visible on all 3-C waveforms, a probability of 1 is assigned in this interval. However, a lower probability (0.75) is assigned for the P-wave signal as it has relatively weaker amplitudes and is visible on x- and y-components only. Also, some features are not able to respond in the P-wave signal interval. The probability assigned to background noise level is 0.1. For a stronger signal like S-wave interval in this example, all features respond equally well. We, therefore, want to choose a network model that uses information from all input features. The training dataset contains 15 such examples where events with different S/N waveforms are used.

Figure 3: A training example of feature inputs (X_i) and desired output (target probability, t). The x,y,z-component of waveforms are also shown. The subscripts 1-3 represent the average of absolute amplitudes for each component, 4-6 show variance for each component, 7-9 are the energy ratios for each component and 10 indicates the polarization rectilinearity. The target probabilities are assigned based on the response of feature curves and waveform S/N at the P- and S- signal intervals.

Figure 4 explains the network model selection. Since we use 10 features as input, network models are trained for a combination of number of neurons (N) = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}. The predicted probabilities show successful event detections

from each trained model, even for the intervals where signals are not easily visible. In this case, the non-visibility of the signal on the waveform components is due to the presence of a stronger event with very high S-wave amplitudes. However, a zoomed view around 3-5s shows the presence of weaker events. The S-wave signal around 3.14s is visible on all 3-C waveforms and is, therefore, assigned the highest probability in the output from all trained models. The S-wave amplitude at 3.9s is relatively weaker and is only visible on two waveform components. Therefore, the predicted probability for its detection is lower (0.85). Similarly, P-wave signals are detected with much lower probabilities because of weaker amplitudes and their absence from some of the waveform components. Although all models perform similarly, we choose model with 16 neurons in the hidden layer (p_{16}) because of its stronger response for the P-wave signal intervals. In order to avoid biased results, we check the weight-based saliency measure for this model. The feature weights are distributed randomly around a mean value and none of the weights are closer to zero, which suggests that this model uses information from all features. The predicted results can be biased when only a few input features contribute (much higher weights) in the network model. Similarly, the weights associated with hidden layer nodes show that all nodes are contributing in the model. The nodes with weights closer to zero do not contribute in the model's performance and suggest adjusting the number of neurons in the hidden layer. The presence of an outlier, a higher magnitude event, can impact the performance of the network. However, we have not investigated this in the current study whether removing the outlier (making training dataset uniform) will improve the network performance.

Figure 5 shows the detection probability for 16-neuron model. By choosing a simple criterion ($p \geq 0.15$), we are able to detect all events (weak and strong) successfully (0.1 is the probability assigned in the training to background noise).

CONCLUSIONS

We have presented a neural network based approach for robust event detection from low S/N waveforms. We find that weight-based saliency and model's performance for different number of neurons can be useful in the network model selection. Also, by setting the output function as a relative probability curve, we can obtain a confidence measure on the detected event. The proposed algorithm, when applied on noisy synthetic data, successfully detected all events, thus proving its robustness for low S/N waveforms.

ACKNOWLEDGMENTS

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST). For computer time, this research used the resources of the Supercomputing Laboratory, Information Technology Division and Extreme Computing Research Center at King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia.

Neural network based event detection

Figure 4: Network model selection. Detection probabilities are shown. Due to presence of a stronger event, weaker events are not visible in the bottom waveform panel but can be seen in the zoomed view (top right). The subscript of p indicates the number of neuron used in the model. Weight-based saliency is shown for 16-neuron model (top left)

Figure 5: Detection probability curve from the neural network with 16 neurons. All events are successfully detected (red).

EDITED REFERENCES

Note: This reference list is a copyedited version of the reference list submitted by the author. Reference lists for the 2017 SEG Technical Program Expanded Abstracts have been copyedited so that references provided with the online metadata for each paper will achieve a high degree of linking to cited sources that appear on the Web.

REFERENCES

- Allen, R., 1982, Automatic phase pickers: their present use and future prospects: *Bulletin of the Seismological Society of America*, **72**, S225–S242.
- Dai, H. C., and C. MacBeth, 1995, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network: *Geophysical Journal International*, **120**, 758–774, <http://dx.doi.org/10.1111/j.1365-246X.1995.tb01851.x>.
- Gentili, S., and A. Michelini, 2006, Automatic picking of P and S phases using a neural tree: *Journal of Seismology*, **10**, 39–63, <http://dx.doi.org/10.1007/s10950-006-2296-6>.
- Hagan, M. T., Demuth, H. B., Beale, M. H., and O. De Jesus, 2014, *Neural network design*, 2nd ed.: Ebook, 1012.
- Kartalopoulos, S. V., 1996, *Understanding neural networks and fuzzy logic: Basic concepts and applications*: Wiley-IEEE Press, 205.
- Maity, D., F. Aminzadeh, and M. Karrenbach, 2014, Novel hybrid artificial neural network based autopicking workflow for passive seismic data: *Geophysical Prospecting*, **62**, 834–847, <http://dx.doi.org/10.1111/1365-2478.12125>.
- McCormack, M. D., D. E. Zaucha, and D. W. Dushek, 1993, First-break refraction event picking and seismic data trace editing using neural networks: *Geophysics*, **58**, 67–78, <http://dx.doi.org/10.1190/1.1443352>.
- McCulloch, W. S., and W. Pitts, 1943, A logical calculus of the ideas immanent in nervous activity: *The bulletin of mathematical biophysics*, **5**, 115–133, <http://dx.doi.org/10.1007/BF02478259>.
- Song, F., H. S. Kuleli, M. N. Toksöz, E. Ay, and H. Zhang, 2010, An improved method for hydrofracture-induced microseismic event detection and phase picking: *Geophysics*, **75**, no. 6, A47–A52, <http://dx.doi.org/10.1190/1.3484716>.
- Steppe, J.M., and K. W. Bauer, Jr., 1997, Feature saliency measures: *Computers & Mathematics with Applications*, **33**, 109–126, [http://dx.doi.org/10.1016/S0898-1221\(97\)00059-X](http://dx.doi.org/10.1016/S0898-1221(97)00059-X).