We_P05_17

Microseismic Hypocenter Location Using an Artificial Neural Network

Q. Hao[1], U.B. Waheed[1]*, M. Babatunde[1], L. Eisner[2]

[1] King Fahd University of Petroleum and Minerals; [2] Seismik s.r.o.

Summary

The sharp increase in the occurrence of human induced earthquakes globally requires real-time source location capabilities, particularly in areas where no prior seismic activity occurred. Recent advances in the field of machine learning coupled with available computational resources provide a great opportunity to address the challenge. Researchers have started looking into using convolutional neural networks (CNNs) for hypocenter determination by training on already located seismic events. We propose an alternate approach to the problem. We train a feed-forward neural network on synthetic P-wave arrival time data (based on a velocity model or empirical data). Once trained, the neural network can be deployed for real-time location of seismic events using observed P-wave arrival times. The use of a feed-forward neural network allows fast training compared to CNNs. We show sensitivity of the proposed method to the training dataset (density and distribution of the training sources), noise in the arrival times of the detected events, and size of the monitoring network.

Introduction

Seismicity induced by mining and impounding of dams have been recognized for decades (Foulger et al., 2018). More recently, hydraulic fracturing and waste water injection have resulted in a sharp increase in the number of earthquakes observed in historically quiet tectonic areas around the world (Ellsworth, 2013). In addition to causing considerable economic losses, such events are increasingly becoming a threat to public safety. For threat mitigation, a traffic light system (TLS) is implemented around the world, which requires real-time location and magnitude determination of induced seismic events (Anikiev et al., 2014).

Numerous techniques exist in the literature on automatic earthquake localization. With the increasing availability of data and computational resources, researchers have started exploiting the capabilities of machine learning algorithms to detect, locate, and interpret seismic events. Perol et al. (2018) used a convolutional neural network (CNN) to study induced seismicity in Oklahoma, USA. They train the network on data from 2709 events recorded on two stations to roughly locate earthquakes belonging to one of six regions. Kriegerowski et al. (2018) also trained a CNN using more than 2000 swarm events from West Bohemia, recorded on nine local stations, to locate clustered earthquakes precisely.

Although these studies have demonstrated the potential of supervised machine learning for event localization, they also signify the need for a large historical dataset needed to train these CNNs. In this abstract, we explore an alternative approach to the source localization problem. First, instead of using waveforms as input, we use only the P-wave arrival times and feed them to a fully connected feed-forward neural network. Furthermore, unlike the afore-mentioned papers, we train the network on synthetically generated data. Once trained, the network can be deployed to locate real events by feeding observed P-wave arrival times as input.

The focus of this study is to understand different factors affecting the accuracy of source localization using a neural network. Through numerical tests, we explore the accuracy of the proposed method as a function of several parameters, including velocity model complexity, receiver network distribution, and the size of the training data. It is worth noting that the proposed workflow does not depend on the availability of historical data for training.

Methodology

A feed-forward neural network is a set of neurons organized in layers. Each neuron represents a mathematical operation, whereby it takes a weighted sum of its inputs plus a bias term and passes it through an activation function. The output of a neuron is then fed to subsequent neurons in the sequence. Mathematically, the output, z, of a neuron is given as:

$$z = f\left(\sum_i w_i\, x_i + b\right), \tag{1}$$

where w_i is the weight associated with the input x_i, b is the bias term, and $f()$ represents the activation function. A nonlinear activation function is typically used to learn nonlinear relationships between the input and the output. Training of a neural network refers to the process whereby a network adjusts its weights to correctly map the input to the output provided in the training data.

For the seismic event location problem, the input layer of the neural network comprises one neuron per station for the total number of recording stations in the monitoring network. The output layer will contain one neuron for each coordinate axis. Hidden layers of neurons are used to learn nonlinear relationships between the input and the output. While increasing the hidden layers and/or the number of neurons in each hidden layer may result in improved performance on the training set, beyond a certain point it leads to the problem of overfitting, causing poor performance on test data. Therefore, careful selection of network parameters is important for optimal performance.

The training data for our network is generated synthetically. To create the training dataset, we compute P-wave arrival times corresponding to sources distributed on a regular grid throughout the volume of interest. For each event, we create a set of traveltimes by subtracting the traveltime recorded at each station by the average traveltime recorded across all stations for the event. Doing so helps us remove location errors caused by ambiguity in the event origin time. This set of times is used as the input to the neural network. The output of the neural network are the coordinates of the source location used to compute the corresponding P-wave arrival times. The training of the network finishes when the network's weights and biases are adjusted to match the input to the output in the training data. To compute location of a real event, we feed observed arrival times measured from an arbitrary point in time reduced by the average time over all stations as input to the trained network.

Numerical Tests

In this section, we test the accuracy of the proposed methodology. Our focus is on understanding different factors that affect the accuracy of the proposed method in accurately inverting for source locations using P-wave arrival data. Figure 1 shows the velocity model considered for the tests. The rectangular box (in black) shows the zone of interest where the events are expected to be located. We choose a neural network consisting of four hidden layers with 40 neurons in each layer. The number of neurons in the input layer equals the number of stations while we have two neurons in the output layer – one for each coordinate axis. The activation function for the hidden layer is the rectified linear unit while the final layer is linear. First, we train the network using synthetic data generated for equispaced sources inside the rectangular box. Having trained the network on this, we test the method by feeding synthetic arrival times from 100 randomly generated test sources from within the rectangular box. Below we present a systematic study of the different factors affecting the accuracy of the proposed approach.

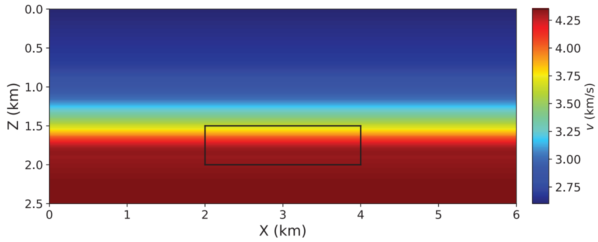


Figure 1 *P-wave velocity model considered for the tests. The rectangular box shows the zone of interest where events are expected to be located.*

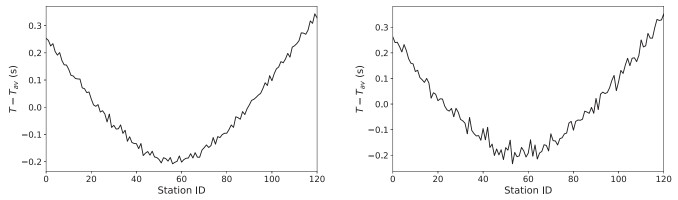
1. Effect of noise in test data

First we test the network by feeding in noisy input data. This is done to simulate roughly the effects of misspecifications in the velocity model and/or errors in picking P-wave arrival times from the observed data. We train the network using traveltime data from 451 sources located inside the rectangular box shown in Figure 1 and spaced at an interval of 50 m along both x and z axes. The traveltimes are recorded at 121 receivers on the surface, with a 50 m interval. We consider two noise levels by adding to the arrival times random Gaussian noise corresponding to 10 ms and 20 ms standard deviations (σ) and zero mean (μ). Figure 2 shows arrival times from a single source in the test data corresponding to $\sigma = 10$ ms (Figure 2(a)) and $\sigma = 20$ ms (Figure 2(b)) noise.

Figures 3 and 4 show error distribution in x and z coordinates of the source locations for the two cases. We observe that errors increase as the noise in the test data increases. However, even when noise is Gaussian with 20 ms standard deviation, the maximum locations error is only 60 m. We also fit a Gaussian distribution to the error histograms and show the mean (μ) and standard deviation (σ) of the fit in each case. It is worth noting that it took only 28.2 s to train the network on a CPU and merely 45 ms to predict locations for 100 test sources.

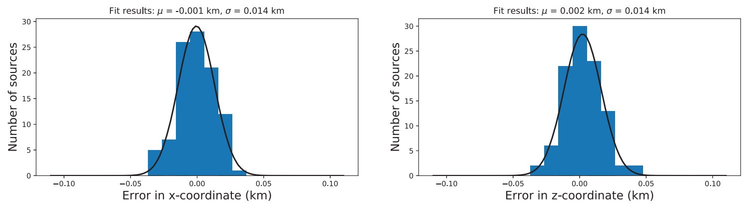
2. Effect of the number of receivers

Next we study the effect of the number of receivers on location errors. The training and test data are reduced to 31 receivers on the surface, spaced at an interval of 200 m. Figure 5 shows location errors for test data from the same 100 sources with a Gaussian noise corresponding to $\sigma = 10$ ms. By comparing



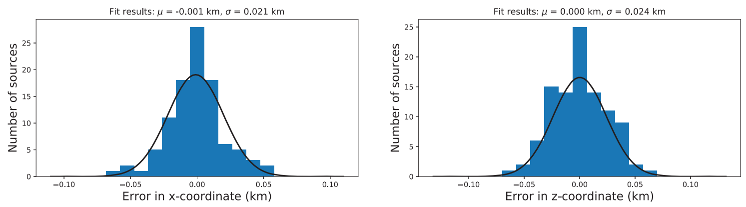
(a) (b)

Figure 2 *Input from a source in test data added with random Gaussian noise corresponding to 10 ms (a) and 20 ms (b) standard deviation and zero mean.*



(a) (b)

Figure 3 *Location errors for 100 test sources with Gaussian noise corresponding to* $\sigma = 10$ *ms added to input data recorded on 121 equispaced receivers on the surface.*

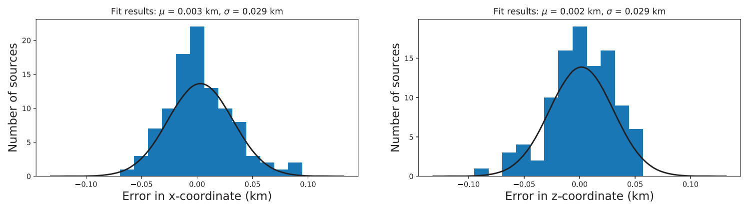


(a) (b)

Figure 4 *Location errors for 100 test sources with Gaussian noise corresponding to* $\sigma = 20$ *ms added to input data recorded on 121 equispaced receivers on the surface.*

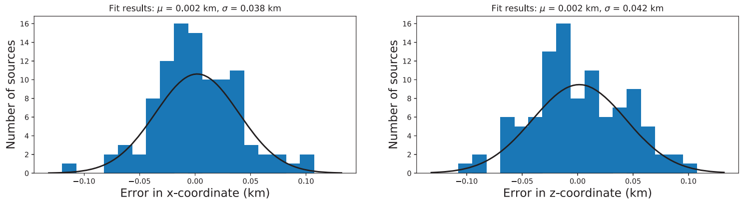
Figures 3 and 5, we observe considerable reduction in location accuracy when the number of receivers is reduced. We observe that the standard deviation of location errors increases by a factor of two.

Next, for the tests with 31 receivers, Figure 6 shows location error histograms for the arrival time noise level of $\sigma = 20$ ms. We observe, as in the previous case, that increased noise worsens the location accuracy but when the number of stations in the monitoring array is reduced, the reduction in accuracy is greater, indicating increased sensitivity to noise. Since the monitoring array is horizontal, the vertical location is less constrained, and therefore the errors in location along depth increases more than the lateral error. However, even in the worst considered scenario, the maximum location error observed is around 120 m.



(a) (b)

Figure 5 *Location errors for 100 test sources with added Gaussian noise corresponding to* $\sigma = 10$ *ms added to input data recorded on 31 equispaced receivers on the surface.*



(a) (b)

Figure 6 *Location errors for 100 test sources with added Gaussian noise corresponding to $\sigma = 20$ ms added to input data recorded on 31 equispaced receivers on the surface.*

3. Effect of the number of training sources

Finally, we study the effect of the number of sources used to train the network. Table 1 shows the training time and standard deviation of x and z location errors for a network trained using 451, 126, and 27 sources. In each case, the sources are placed at regular intervals along both dimensions. We use 121 recording receivers and the test data is contaminated by Gaussian noise with $\sigma = 10$ ms. It is obvious that the training time reduces as we decrease the number of training sources but the reduction in accuracy is significant. Since training is performed on synthetic data, this observation suggests using a higher number of sources to improve the accuracy of the trained network. Furthermore, since even with 451 sources the training time is quite small, there is no point in considering a reduction in training data.

Number of training sources	Training time (s)	Std. dev. of x error (m)	Std. dev. of z error (m)
451	28.2	14	14
126	10.9	23	25
27	4.9	70	95

Table 1 *Training time and standard deviation of location errors for different number of training sources.*

Conclusions

We show that feed-forward neural networks can be used to obtain fast and accurate source locations for microseismic monitoring. We train the network on P-wave arrivals of a synthetic dataset. This allows us to use the proposed approach even for cases when no historical data exist. We show that the source area needs to be adequately sampled to achieve higher accuracy of the located events as the accuracy decreases with lower number of receivers and higher noise. Alternatively, the approach can also be used for cases when a sufficient amount of historical data is available for training. In that case, instead of training on synthetics, one could use the already located events and their P-wave arrival picks to train the network. Obviously for this case, the accuracy of the network will vary depending on the location density of those already located events.

References

Anikiev, D., Valenta, J., Stanek, F. and Eisner, L. [2014] Detection, location and source mechanism determination of microseismic events: evaluating seismicity related to hydraulic fracturing. *First Break*, **32**(5), 59–65.

Ellsworth, W.L. [2013] Injection-induced earthquakes. *Science*, **341**(6142), 1225942.

Foulger, G.R., Wilson, M.P., Gluyas, J.G., Julian, B.R. and Davies, R.J. [2018] Global review of human-induced earthquakes. *Earth-Science Reviews*, **178**, 438–514.

Kriegerowski, M., Petersen, G.M., Vasyura-Bathke, H. and Ohrnberger, M. [2018] A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms. *Seismological Research Letters*, **90**(2A), 510–516.

Perol, T., Gharbi, M. and Denolle, M. [2018] Convolutional neural network for earthquake detection and location. *Science Advances*, **4**(2), e1700578.