SAND2018-13957C

Detecting Low Magnitude Seismic Events Using Convolutional Neural Networks

Robert Forrest[1], Jaideep Ray[1], and Chris Young[1] | *Sandia National Laboratories[1]*

1. Objective

Currently, many traditional methods are used to detect arrivals in three-component seismic waveform data collected at various distances. Accurately establishing the identity and arrival of these waves in adverse signal-to-noise environments is helpful in detecting and locating the seismic events. Autocorrelation and template matching techniques are just a few of the various methods that may be used, each with their own performance benefits and drawbacks. For example, template matching methods have been shown to be effective, but are restricted to repeating signals and become computationally intensive with increasing numbers of templates.

In this work, we move to convolutional neural networks (CNNs). Convolutional neural networks have been shown to significantly improve performance at local distances under certain conditions such as induced seismicity. In this work we expand the use of CNNs to more remote distances and lower magnitudes. We explore the advantages and limits of a particular approach and begin to understand requirements for expanding this technique to different types, distances and magnitudes of events in the future. We describe in detail results of this method tuned on a new dataset with expert defined arrival picks. The dataset used is from the Dynamic Network Experiment 2018 (DNE18) and comes from sensors in Utah. We demonstrate the ability to train the CNN on events from the dataset and achieve high test set performance. Furthermore, we examine performance on streaming data, including very low magnitude expert picked arrivals.

Goal: Use established CNN methods on a new expert picked arrival dataset to understand advantages and limitations of current CNN methods.

2. CNN Background

General:
CNNs are a particular type of feedforward artificial neural networks used for processing data. Unlike traditional neural networks in which all layers are fully connected, CNNs use a linear convolution in some layers as opposed to a full matrix multiplication. These convolutions have several advantages over traditional fully connected layers. Importantly, convolutions act as filters spatially invariant features and sharing these parameters throughout the data dramatically increasing the generalizability especially in the case of time series data. Additionally, because of the reduced connections between layers, CNNs severely reduce the parameters that need to be trained, improving statistical efficiency. Therefore, despite there prevalence in image recognition CNNs may also make a good candidates for pattern recognition in time series seismic data.

CNN Architecture:
In this work, we follow the lead of the architecture of ConvNetQuake (see reference 1) and expand to new datasets and lower magnitude signals. The architecture takes raw seismic waveforms from three component 100 Hz ground motion sensors and divides the input into 10 second windows (see figure 1). The network consists of eight convolutional layers that down sample the signal by a factor of two and contain 32 filters.

The output consists of a 1D vector of 128 features and then a fully connected layer containing the class scores. The original ConvNetQuake categorizes waveforms into clustered geographical regions. We replicate that in this work, although we are more interested in detection performance than location accuracy.

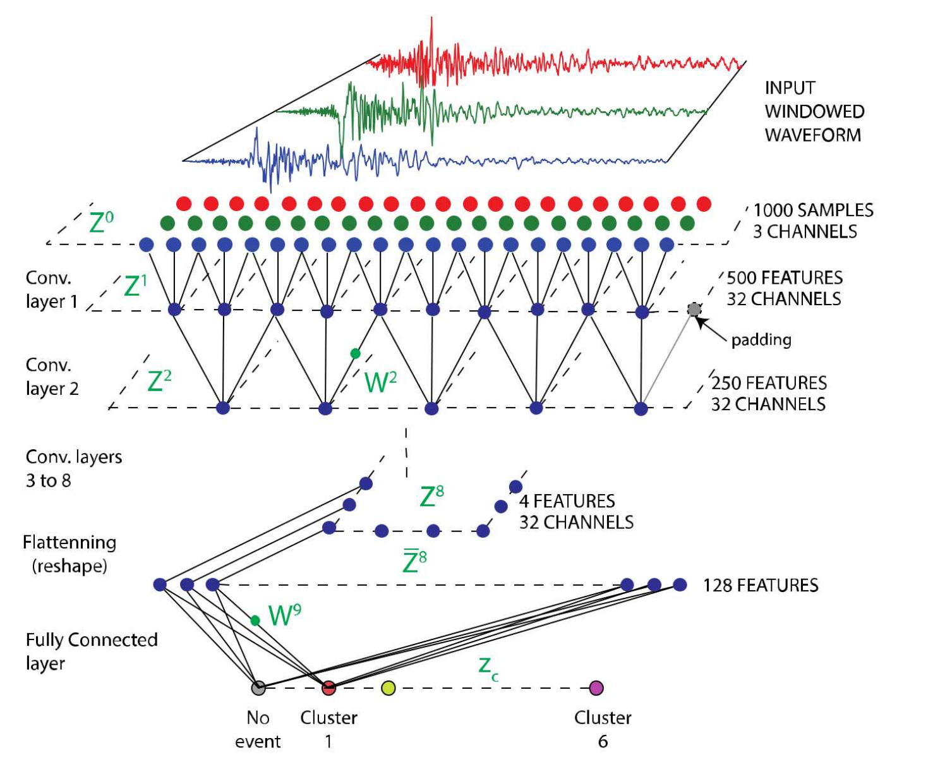


Figure 1: ConvNetQuake Architecture (Reference 1)

3. Seismic Data (DNE18 Dataset)

The data used in this study is the 2018 Dynamic Networks Data Processing and Analysis Experiment (DNE18) dataset. A more detailed description can be found elsewhere in this conference (see Reference 2).

This dataset consists of University of Utah Seismograph Stations (UUSS) waveform data spanning from 01/01/2011 through 01/14/11. Stations sample three component data (N, E, Z) at 100 Hz. Although there are many stations to choose from, we only use a select few for this study.

This catalog consists of a high quality, hand-picked seismic events derived from analyst Chip Brogan's picks with the Analyst Review Station (ARS) software as well as other automated event input sources. The event must be sensed on 3 or more stations in order to build an event. All events (man-made or natural) generated match this criteria. The advantage of this dataset is the low magnitude and expert curated events we will use as ground truth.

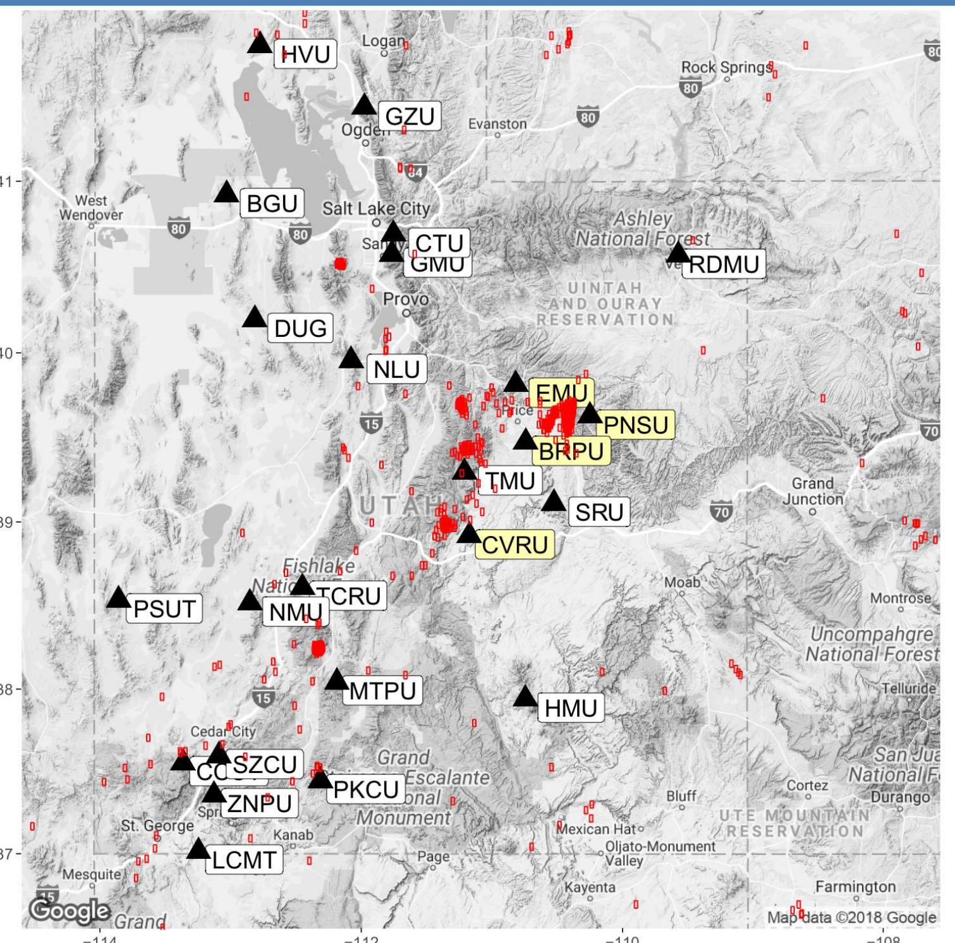


Figure 2: Map of DNE18 stations (black triangles) and seismic events (red circles)

4. CNN Training and Validation

To train and validate the CNN, there are three general steps. Data preparation, training and validation. We go through each of these here.

Data Preparation:
Streams of waveform data are normalized for each station and divided into 10 second windows. Each window is classified as having either seismic noise or a signal event based on labeled seismic arrivals in the event catalog for that station. Because we have exact arrival times, yet want to train the network to recognize events anywhere in the window, we introduce a time buffer into some events between the start of the window and event onset.

Training:
As in Reference 1, we train the CNN by minimizing with backpropagation the L2-regularized cross-entropy loss function which measures the difference between the predicted and actual class probability (signal or no signal). We form batches of 128 windows consisting of 64 noise and 64 signal events per batch. We use the ADAM optimizer and a learning rate of 10E-4.

The training curve for one station is shown here. Each step (X-axis) is a batch of 128 signal and noise events.

We train until the accuracy (Y-axis) saturates without any improvement. Here we can see accuracy saturating at about 93%

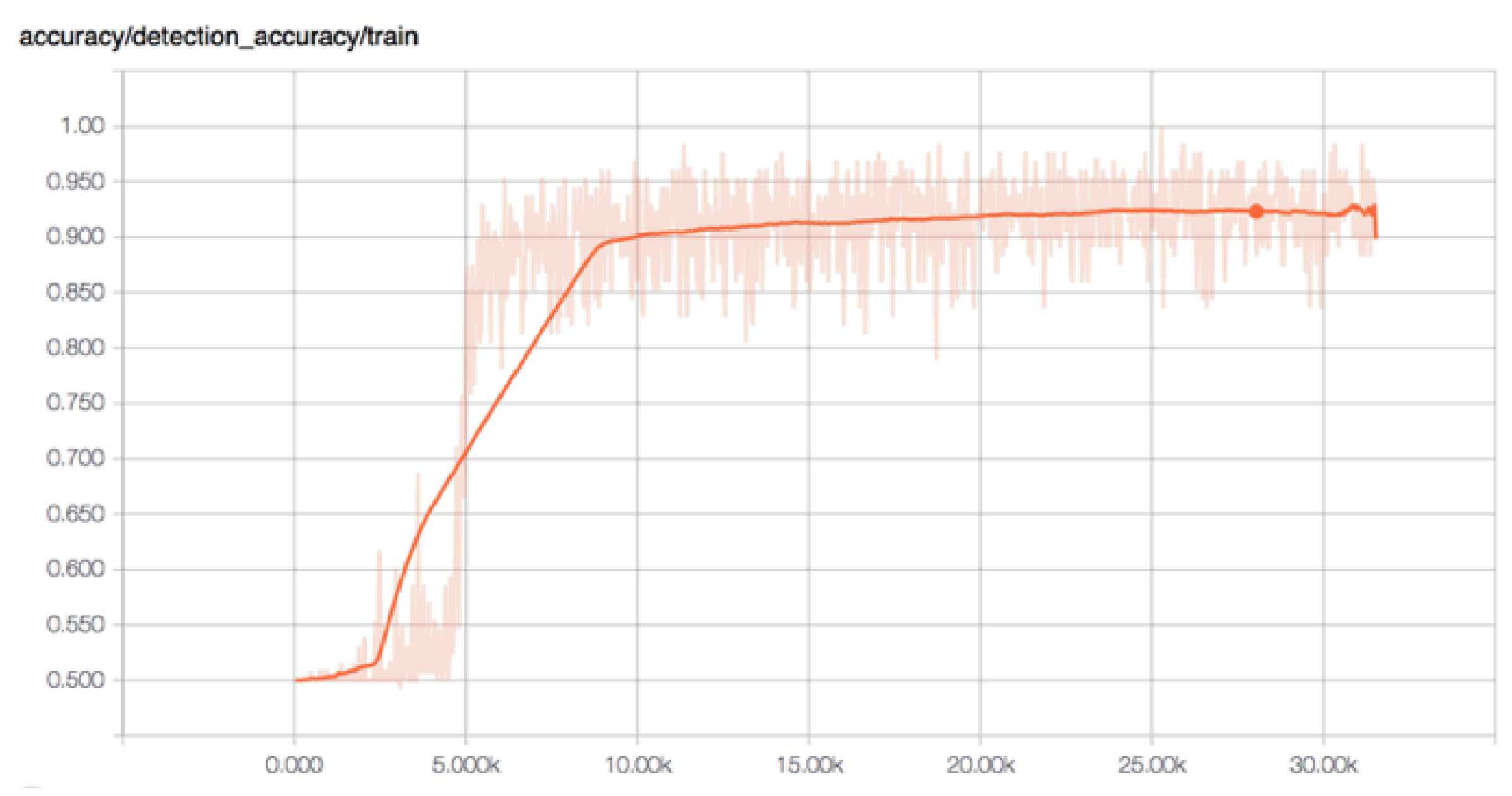


Figure 3: Training curve of the CNN, accuracy vs. steps.

Validation:
We separate the dataset into a primary training set, used above, and a distinct validation dataset that we do not use to train, but use to validate performance. Generally, for most stations we use data from 01/13/11 00:00 to 01/15/11 00:00 for validation. This gives sufficient signal events to accurately measure performance on a distinct set of data. We have both a signal and a noise validation set. Detection accuracy is the metric we use to gauge performance. We track this metric as the neural network is trained.

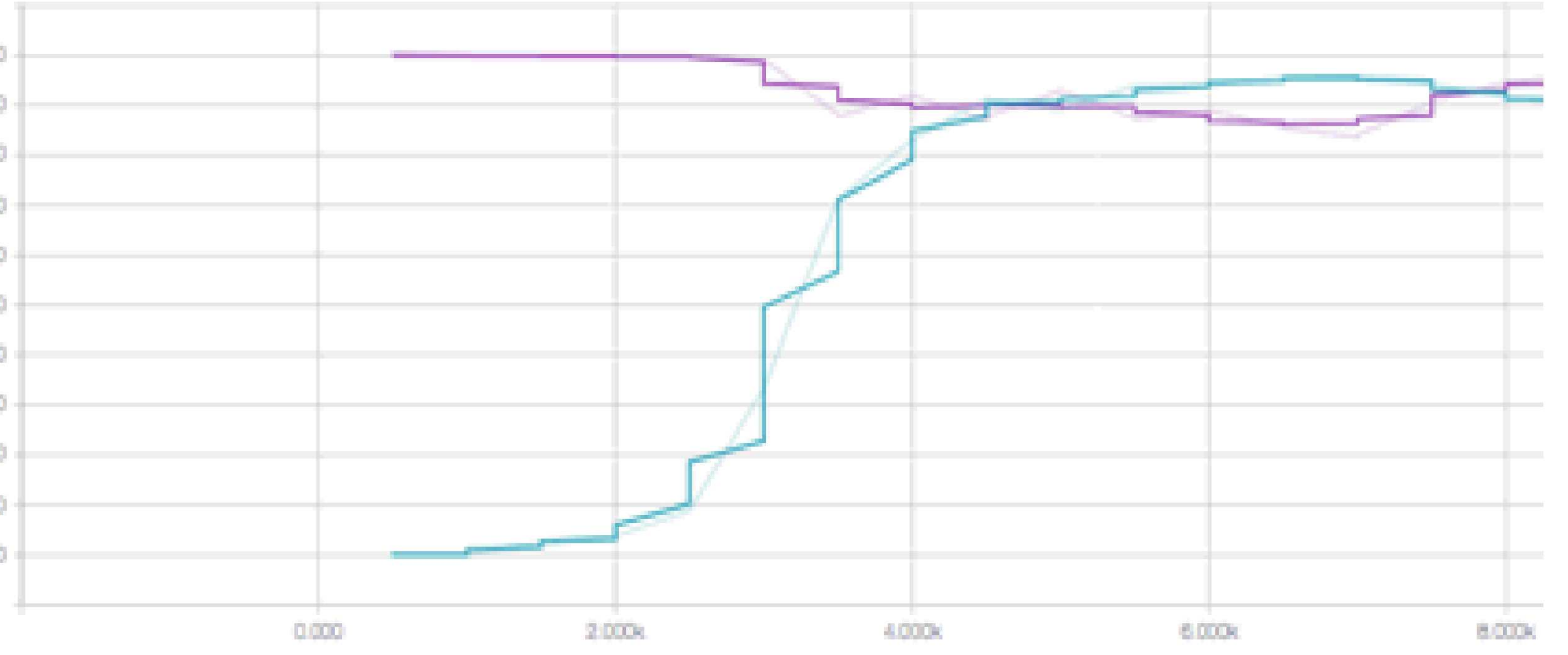
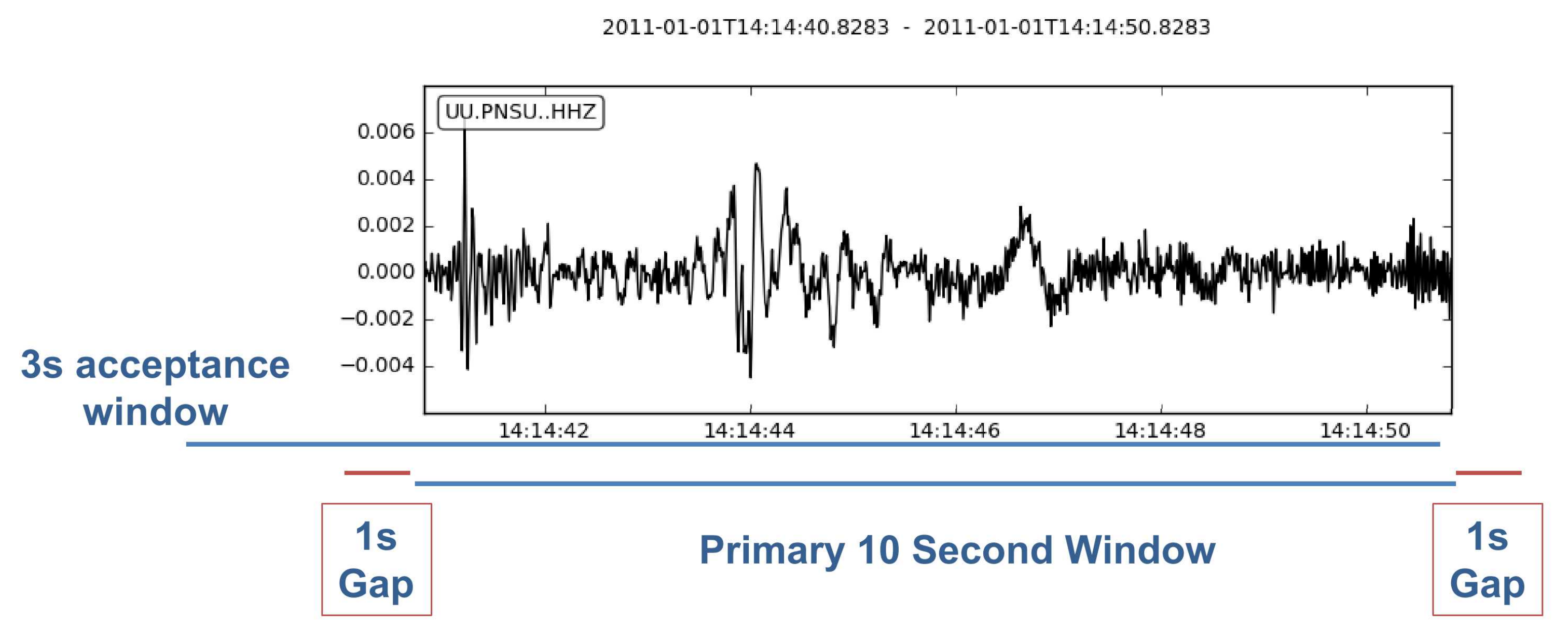


Figure 4: Detection accuracy of noise (purple) and signal events (blue).

Initially the untrained network makes blanket predictions of all noise, then performance for signal and noise validation events stabilizes.

5. Results

To measure performance, we run the trained CNN on streaming data and compare the results to expert picked event catalog. Our network requires 10 second windows, and we employ two methods to prevent potential overlap of signal events between consecutive windows. First, we choose windows every 11 seconds, creating a one second gap between consecutive windows the network sees. Second, we count as a successful picks any 10 second windows in which there is an event from t = -3 s to t = 10s. This scheme is illustrated below.



Although we do have this curated event catalog with expert picked arrivals, it still may be challenging to understand results other than true positive arrivals labeled by the network. To address this in the future, we plan to compare with other established detection methods. Additionally we plan on examining potential false positive and false negative events to expert analysts to verify a lack of signal in the data.

As mentioned above, the computational performance of neural networks is generally superior to traditional methods of detection. Our network runs on two weeks of streaming data in under three minutes for one station.

Confusion matrix for NN detections for various stations compared to expert picked arrivals

Detections		PNSU		MTPU		TCRU		BRPU	
		True	Missed	True	Missed	True	Missed	True	Missed
Expert-picked	True	307	366	TBD		TBD		TBD	
	False	137	109,823						

$$Precision = \frac{True\ Detections}{True\ Detections + False\ Detections}$$

$$Recall = \frac{True\ Detections}{True\ Detections + Missed\ Detections}$$

6. Conclusions

1. CNN provide a fast and scalable way to detect earthquakes from raw 3 component waveform data.
2. When applied to an expert picked dataset, the ConvNetQuake architecture has comparable performance.
3. Further training on large datasets is likely to improve performance, especially at lower magnitudes.

7. Future Work

1. Validate arrival picks with various other arrival detection techniques to better understand performance.
2. Examine false positive and false negative events to understand CNN outputs.
3. Move to spectrogram image inputs.
4. Explore various neural network architectures and training methods to understand if increased performance is possible.

9. Acknowledgements

We would like to especially acknowledge the original work done by Thibaut Perol,, Michael Gharbi and Marine A. Denolle who created the original ConvNetQuake on which this CNN is based.

8. References

1. Convolutional Neural Network for Earthquake Detection and Location, Science Advances 14 Feb 2018: Vol. 4, no. 2, e1700578
2. The 2018 Dynamic Networks Data Processing and Analysis Experiment (DNE18), AGU 2018 S53E-0469

This project was funded by the Department of Energy Office of Defense Nuclear Nonproliferation.

For more information, please contact:
Rob Forrest, Sandia National Laboratories, rforres@sandia.gov