SAND2020-12489

SANDIA REPORT
SAND20XX-XXXX
Printed Click to enter a date

Convolutional Neural Networks for Signal Detection

Robert D. Forrest

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods/

National Nuclear Security Administration

ABSTRACT

Currently, traditional methods such as short-term average/long-term average (STA/LTA) are used to detect arrivals in three-component seismic waveform data. Accurately establishing the identity and arrival of these waves is helpful in detecting and locating seismic events. Convolutional Neural Networks (CNNs) have been shown to significantly improve performance at local distances. This work will expand the use of CNNs to more remote distances and lower magnitudes. Sandia National Labs (SNL) will explore the advantages and limits of a particular approach and investigate requirements for expanding this technique to different types, distances, and magnitudes of events in the future. The team will describe detailed performance results of this method tuned on a curated dataset from Utah with its expert-defined arrival picks.

Contents

This page left blank

1. INTRODUCTION

The ability to comprehensively detect and properly classify low magnitude earthquakes using ground motion recorded at remote distances is key to understanding fundamental seismological activity and a plethora of terrestrial and anthropogenic processes.

Characterizing lower magnitude events, where signal may be masked by significant noise, requires powerful and interesting analysis in a variety of fields. Data that has traditionally been gathered by high quality seismometers is now complimented with data from more unique, common, and varied sensors. The results are orders of magnitude more data, which present fantastic scientific opportunities yet fundamentally change the tools and techniques required to make use of this data. For example, template methods that use per-station historic patterns may not be practically feasible with significantly more new stations that lack a long history of labeled events.

To build seismic events from ground motion data recorded by a network of stations, we generally first classify phases in several single station seismograms. The quality of these initial phase picks determines the quality and magnitude of the event that is built from them; hence analysts often spend a lot of their time editing these picks. However, human effort is better spent on the higher-level logical tasks associated with event building and refinement rather than determining phases in particular waveforms.

Fortunately, signal detection is well suited for neural networks. There is a tremendous amount of human labeled data available whose quality is cross checked by expert visual inspection, including comparison with data at other stations. Generally, the data is well understood, fairly consistent, and for many sensor networks, a significant quantity exists even for a single station.

However, there are significant challenges in applying neural networks to seismic data for signal detection. Most event catalogs deliberately prioritize high magnitude seismic events, with far fewer low amplitude examples available, hence some potentially valuable training inputs that should have been labeled as signals are by default mislabeled as noise. Further, once neural networks have been trained, they may output real events that do not appear in catalogs and hence are scored as false positives. Therefore, determining accurate performance metrics for low magnitude event signal detection is difficult.

We find, through detailed analysis of two stations from the University of Utah Seismic Station (UUSS) network, that significantly less training data is to create an effective detector than has been shown in other recent studies applying neural networks to signal detections (e.g. Ross et al., 2018). This is an important result because in many areas of monitoring interest, large labeled training sets may not be available. For station PNSU our results are particularly impressive, with an AUC of 0.95 and a true positive rate of 69%. For the BPRU station we have a TPR of 67% and an AUC of 0.83, and we also note a very high false positive rate. We believe this is because the data available to train BPRU is inadequate for our methods.

2. HISTORIC WORK

Historically there have been significant improvements to earthquake detection from upgraded detector technologies and analysis approaches. The STA/LTA approach (Allen, 1978) is a robust standard in detection that compares signal amplitudes ratios over varying time windows. While simple and effective across a variety of scales and geographic regions, more modern techniques have been demonstrated to capture more true events, though typically they require more tuning and use significantly more computational resources. Waveform correlation approaches (template matching) exploit the similarity between seismic traces observed for an event that recurs in the same location and is of the same type; these methods are computationally expensive in that they must compare repeated waveforms point by point, but modern versions (Yann LeCun, 1998) use approximation methods to provide improved performance (Clara E. Yoon, 2015) relative to earlier methods. To implement waveform correlation as a signal detector, relevant templates of waveforms for individual stations are built up in a database, then compared to continuous incoming waveform data from that station. As these historical databases increase in size, hence sampling a greater range of source locations and types, the method becomes more powerful. However, computational requirements grow significantly as the size of the template database increases. Additionally, all waveform correlation methods are only effective for repeated events, and they perform very poorly on unfamiliar seismic activity (i.e. new source types and/or source regions).

Recent advances in identifying phases in seismic waveforms have been made by leveraging deep neural networks. The key insight is that methods developed for other fields are incredibly powerful when applied to time series data. Neural networks are trained to construct abstract representations of input data based on multiple layers of internal weights and connections. Convolutional neural networks (CNNs) (Yann LeCun, 1998) leverage the convolutional operation to essentially learn relevant filters at varying scales of the input. These convolutions act at various time scales and therefore can construct generalized representations of waveforms that do seem do have the ability to generalize where template matching cannot.

Recent advances in seismic CNNs have shown significant advances relative to previously state of the art results. ConvNetQuake (Thibaut Perol, 2017) both detects and locates earthquakes from a single waveform and finds twenty times more earthquakes than are recorded in catalogs generated with traditional methods, and with significantly fewer computational resources required. Another method (Zachary E. Ross, 2018) has shown the ability to achieve generalized phase detection (GPD) by training with large amounts of analyst-reviewed P and S phase data from Southern California. Both methods demonstrate the ability to generalize over both magnitude and region, which is important in that it significantly increases the utility of the technique and moves towards the capability to apply CNNs trained in one region to data from new regions.

As these methods move to maturity and become commonly used, it is important to understand the specific behaviors and how they generalize. As we introduce additional phases and vary epicentral distances, categorizing the behavior and capability will allow a deeper understanding of how to apply these tools and where the approach may break down. Additionally, understanding the effectiveness of neural networks when applied to smaller datasets in new geographies allows us to round out our knowledge.

In this paper we apply the ConvNetQuake method to a highly curated benchmark dataset of seismic events from Utah (Linville L. B., 2019). Using such a detailed human-labeled catalog gives us the ability to more deeply understand the training and performance of ConvNetQuake. We are able to look for recordings of different phases recorded at a range of epicentral distances for low magnitude events. Significantly, we have the ability to understand the false positive rate through the use of additional curated noise windows. We look at single stations to understand them more completely, then expand to multiple station systems to examine how our results generalize. The results presented here reinforce the impressive performance of the neural network approach shown in other studies and allow us to characterizer its generalizability, particularly with regard to different phases and different regions of seismicity. Most significantly, it allows us to more deeply understand the performance of these detectors.

3. METHOD

Recently, various approaches have emerged that classify phases in waveforms with convolutional neural networks. Fundamentally, they all map input windows to an output classification through a series of simple mathematical operations, that form a complex overall system due to the large number of operations that are involved. The exact architecture varies between approaches but consists of layers of connections in which the output of one layer is the input of the next. Commonly, the first layer takes a window of time series data and subsequent layers convolve channels of the previous layer with learned filters before passing them on. Normally the last layer is a fully connected linear layer that is reduced into a feature vector used for classification.

In this work we use ConvNetQuake as described in (Thibaut Perol, 2017) as our reference framework. We found that altering the structure of the network did not result in any significant improvements, though we did update the software framework to TensorFlow 2.0.

A detailed description of the network is available in the original ConvNetQuake paper, but we will review it briefly here. A diagram is provided in *Figure 1* The network architecture is fairly simple, with eight convolutional layers, followed by a fully connected layer that outputs class scores. At each convolutional layer except the first, there are 32 channels. Linear 1D filters (kernel size of 3) are convolved with the previous layer's channels and summed. This quantity and a bias term are input to a non-linear ReLU function, then output to the next layer. Convolutions are strided at each level with S=2, resulting in an effective down sampling at each layer. This means that at each layer, filters essentially act on raw data and draw out features at different time scales. At the earliest layers, full resolution high frequency features are learned, while deeper levels act on the down sampled results picking out lower frequency features over a wider time window.

Ten second segments of three component 100Hz seismic waveform data are introduced into the first layer as a 2D tensor "2-D tensor Z0". At the eighth layer, the network processes the final tensor of shape (4,32) into a vector and then a fully connected layer that outputs the classification scores. These scores are normalized by a SoftMax function to what is interpreted as a probability function. Our work only differs from the original paper in that we have just one output node, as we disregard location data associated with region identification.

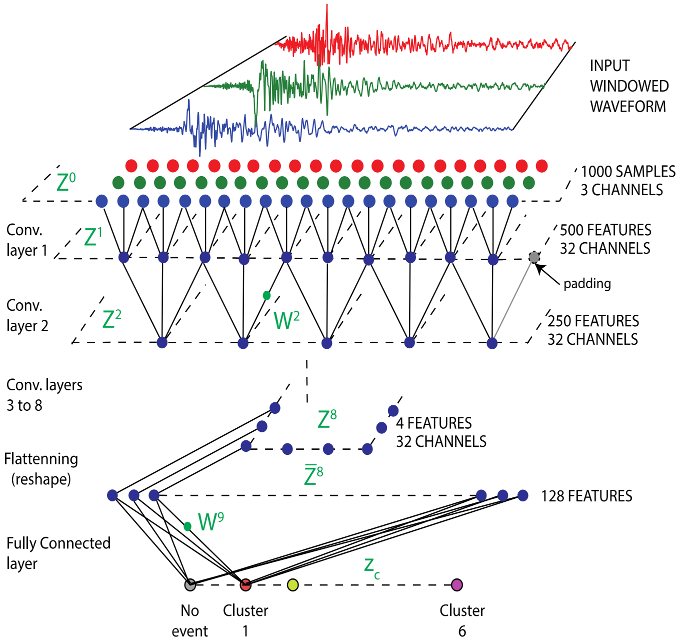

Figure 1: ConvNetQuake Architecture, from (Thibaut Perol, 2017)

4. DATA SET - GENERAL QUALITIES

Part of the challenge of developing new arrival detection methodologies is the lack of appropriate datasets to assess performance, especially at low detection thresholds that may include significant anthropogenic sources. Lowering detection thresholds introduces significant complications that must be assessed. Anthropogenic noise may dominate and look similar to event-generated seismic activity at very local scales. Additionally, event-generated seismic activity may be present at individual stations, but from events that are so small that they cannot be detected at other nearby stations, eliminating the ability to confirm a signal detection or construct the corresponding event.

In general, there are several advantages to using high quality standard datasets for training and testing CNNs. Primarily, significant effort and expertise has already been put into grooming and labeling the dataset, something that may be very costly on an ad-hoc basis. Additionally, standard datasets allow for a more equitable comparison of results between approaches and algorithms. Nominally, datasets are chosen based on availability to the researcher and compatibility with their approach, leading to results that are difficult to interpret more broadly.

Additionally, as we describe below, seismic event bulletins by their nature are downstream in the analysis process pipeline and are built on mostly automatically detected arrivals. As our method improves arrival picking, the quantity event construction is built on, we will have to address this discrepancy in arrivals (as we will describe below) to faithfully test the method. Nevertheless, starting from a high-resolution reference dataset built around confirmed smaller magnitude events is important for both algorithm training and testing.

1.1 Dataset - Our Data

We use a unique, high-quality standard dataset of events and associated arrivals called the Unconstrained Utah Event Bulletin (UUEB), (Linville L. R., 2019). The UUEB greatly extends the much less complete catalog that is routinely produced by the University of Utah using their regional University of Utah Seismograph Stations network (UUSS). The University operates this network throughout the state to better understand local earthquake behavior for the purpose of assessing seismic hazard in populated areas. The UUSS event bulletin as well as UUSS waveform data are freely available from IRIS (Utah, 1962).

The UUEB event catalog spans January 1 through Jan 14th, 2011. This timeframe was selected explicitly to include extensive low magnitude events from both anthropogenic and natural sources. The catalog contains seismic aftershock events from a large earthquake in southern Utah, as well as extensive mining events of various types from various locations. Each event in the dataset has been built and reviewed by an expert analyst, including careful examination of all associated arrivals. Accordingly, the catalog is much more complete than the UUSS catalog for the same time period. In total, 8270 events in or close to the state of Utah are included in the UUEB catalog in the two-week period, as compared to less than 200 in the UUSS catalog.

1.2 Data Set - Further Investigations by Experts

We used UUEB events in our analysis that had three confirmed arrivals. If we detected arrivals at additional stations, we examined the waveforms to determine if the detected arrivals associated with a given event were present. The advantage of using this dataset is that we were able to focus on small events that would otherwise been skipped in most operational environments.

During the final data quality pass, an expert augmented our picks to be more appropriate for assessing signal detection. Additional real arrivals (as judged by an expert manually reviewing waveforms) were added, even if they could not be associated with a 3-station event. This has the effect of identifying smaller signals that should have been detected.

Significantly smaller arrivals were suppressed to avoid contaminating our results. When an analyst is examining signals across a network of stations for a given event, some marginal quality signals may be associated primarily based on evidence provided by the quality of signal detection on adjacent stations in order to produce an event with three arrivals. That is, two obvious arrivals at nearby stations point the analyst to a specific time window in which a marginal signal may be discerned on a third station. We screened such events in our final data quality pass.

The important implication is that, although this is an excellent dataset for our study due to the detail in which it has been carefully reviewed by an expert analyst, we note that is still incomplete. Marginal three-station events we screened, as were one or two station events

1.3 Data Set - Specifics

The intent of this study primarily is to understand in detail the subtleties of training seismic neural networks. Specifically, we intend to understand the benefits, if any, to including waveforms from other local stations to supplement the data available to train on a per-station basis. As stated above, we hope this will add to our understanding of training neural networks in general, and area that is often overlooked. Equally important, we hope that by looking at supplementing training data in this way, we can understand the extent of potential generalizability of these methods and the extent to which each station may benefit from supplemental training with data from other local stations.

To this end we choose two stations the UU network that we believe are suitable to understand the effect of supplemental data in training. Station PNSU has more associated signals in the UUEB than any other UUSS station. These are dominated by Lg waves likely from nearby mining activity.

Station	Total Arrivals	Total Lg waves
PNSU	13883	4603
BRPU	3611	1726

Station BRPU is close to PNSU and hence can potentially detected a lot of the events PNSU sees. Data available used in the analysis is in Table 1.

Table 1: Arrivals available at Used UUSS Stations

5. NETWORK DESIGN

5.1. Differences from Original paper

In terms of significant differences from the original ConvNetQuake approach there are a few key points. First, we do not investigate the ability to locate the seismic event, as we are exploring more general capabilities of phase detection. Because we have an abundance of Lg waves in the two-week window, and because other work has looked at P and S wave detection, we chose to focus on the Lg phase.

The dataset is significantly smaller than some other analyses (Zachary E. Ross, 2018) and is limited to single station models deliberately, because we want to examine how such models generalize. Also, we test on real, continuous data (as opposed to windowed data around known signals) which we believe is the ultimate test of the capability of a detector.

5.2. Training the Network

For all our training we use data from Jan 3-14th and test on held out Jan 1-2 waveforms. We also trained and tested on different days of data (train on Jan 1-11 and test on 12, 13, 14th) but found that results did not differ. Because of the relatively small amount of data, it is difficult to have a training, development and test set. Therefore, we use a portion of the training set timeframe as a development set during training.

As in the original ConvNetQuake paper (Thibaut Perol, 2017), ten second fixed windows of the three-component waveform are used for training. We found, after significant investigation, that performance suffers if the training event signal onset is centered (t=4s) in the window. Centering the wave in the window for training decreases sensitivity to waves not close to the center. We therefore introduce a shift around the onset of the wave in the window at training (see section A.1). The results show that performance is significantly improved if training includes waves beginning within the first approximately 8 seconds of the ten second window, enough to capture some significant part of the onset. Unlike the original approach, we do not add gaussian noise to our training events.

The network minimizes an L2 regularized cross entropy loss function that essentially classifies as noise or signal at the output. We use the ADAM optimizer with all other parameters the same as in the original work. ConvNetQuake is implemented in TensorFlow 2.0. We train the network until the loss plateaus, and when this occurs varies for different stations based on the available training data. We use batches of 256 events to train balanced between noise and signal. Because of the plethora of noise windows available as compared to signal windows, we train in epochs defined as one full pass of the set of noise windows. Therefore, one epoch may contain the signal events more than once. Because each station has a significantly different quantity of data available (imbalance of quantity of signal and noise) and because the random nature of events and noise that are introduced at each step, we converge at varying rates per dataset when looking at plots of loss during training.

6. RESULTS

To measure performance, we run the trained CNN on streaming data from the test set time period and compare the results to our expert picked event catalog. Our network requires 10 second windows, and we employ two methods to prevent potential overlap of signal events between windows. First, we choose windows every 11 seconds. Second, we count as a successful pick any 10 second window in which there is an event from t = -1 s to t = 10s.

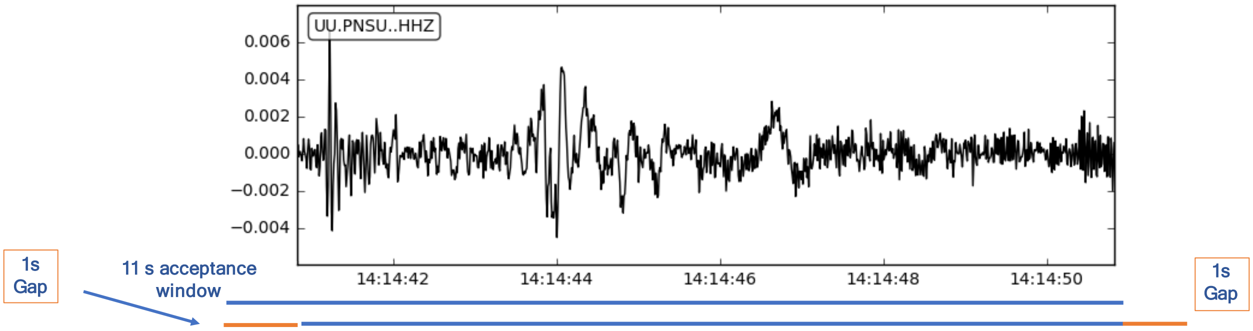

Figure 2: Acceptance window. We count as a success any test set pick that occurs from t=-1s to t=10s. We avoid overlap by testing on 10s windows, spaced by 11 seconds.

To prevent overtraining, we stop training when the loss plateaus, or does not decrease, for several epochs in a row.

6.1. Results from PNSU

We first look at the results from training only PNSU. The training curves can be found in the appendix. The receiver operator curve (ROC) of True Positive Rate versus False positive rate is shown in **Error! Reference source not found.**:

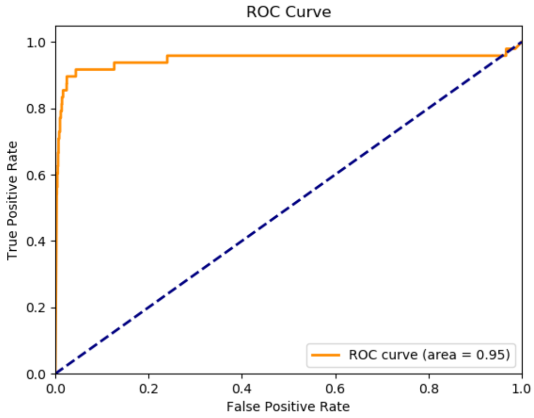

Figure 3: The receiver operator curve (ROC) of True Positive Rate versus False positive rate for PNSU. Area Under the Curve (AUC) is 0.95.

This result is at a decision on the probability threshold of 0.5, as shown in *Figure 3*, we can adjust the confidence level of the decision threshold and potentially tradeoff between FPR and TPR according to desired usage of the detector, therefore ROC is a valid measure of overall detector performance.

The confusion matrix from the test set is shown in Table 2:

PNSU Confusion Matrix		
	T	F
T	321	145
F	52	16763

Table 2: Confusion Matrix of PNSU. Predicted (Vertical) events versus real events (Horizontal).

We can see that for the 466 real signals, the network found 321 and missed 145 (false negatives). The great majority of examples in the test set (17,280) are noise and the network correctly identified

16,763 of these, while falsely classifying just 52 samples (i.e. false positives). The accuracy on the test set is 91% and the false positive rate is 0.86%.

6.2. BRPU Results

Results from BRPU were significantly worse, likely because of the lack of data available in that station, see Table 1 where true negatives dominate because of the predominance of noise. While there may be useful techniques to assist with this in the future, such as weight transfer, we wanted a valid comparison to PNSU. Results are in *Table 3*. We can see that for the 175 real signals, the network found 118 and missed 57 (false negatives). The great majority of examples in the test set are noise and the network correctly identified 17,105 of these, while falsely classifying 2260 samples (i.e. false positives), a considerably higher number than for PNSU. The AUC is 0.83. We tried to decrease the false positive rate but were unable to do so. We believe the problem is the small number of signals available for training at BRPU compared to PNSU. This sharp decrease in performance may indicate that with these two stations we have made a significant advancement in establishing the minimum number of arrivals needed to train an effective CNN signal detector (somewhere between BRPU number and PNSU number).

Table 3: Confusion matrix for BRPU Predicted (Vertical) events versus real events (Horizontal).

BRPU Confusion Matrix		
	T	F
T	118	57
F	2260	17105

6.3. False Positives

Noise waveform windows may be falsely identified as signals for a variety of reasons that are important to understand. We look at randomly selected false positive windows to understand what features the detector may be triggering on. For example, there may be a simplistic average energy component that dominates selection, or a more complex selection of frequency content within the waveform, or a more non-linear combination of many features.

Because of the nature of the noise waveforms which are automatically selected to precede first P arrivals and are not manually reviewed, sometimes false positive windows do actually include a real, legitimate signal, and we confirmed this does occur in our dataset. We cannot look at all of the FP windows to understand the breakdown completely, but based on looking at a random sampling, we estimate that at least 70% of false positive noise windows actually have genuine signal. The percentage could be even higher because for very weak signals from low magnitude events, confidently identifying a signal becomes subjective even for expert analysts.

7. CONCLUSION

We have explored the performance of a convolutional neural network Lg phase detector suing local distance events recorded by two stations from the University of Utah Seismic Stations network. Training data and test data were taken from a highly curated event catalog that included far more low magnitude events than in typical event catalog. Because of the dataset used, we are able to determine with confidence the performance on both signal and noise events with high confidence.

Ideally, additional signal data at each station would be added to the training set to achieve better performance. However, even when only a small amount of training data are available, these networks show promise in detecting phases other than P and S waves that have been the focus of many previous studies. Additionally, we discovered new real arrivals that were not present in our curated event catalog, implying this method is more sensitive than traditional methods. However, for one of our stations we also generated high numbers of false positives, which we attributed to an inadequate number of signals available for training, suggesting that there are limits to how small a training set is needed to achieve good results. Although we investigated various methods to remedy this limitation, we were unable to overcome it and believe that there may be a hard limit on the number of labelled signals needed.

REFERENCES

[1] Allen, R. (1978). Automatic earthquake recognition and timing from single traces. *Bull. Seism. Soc. Am*, vol. 68, no. 5, 1521-1532.

[2] Clara E. Yoon, O. O. (2015, 12 4). Earthquake detection through computationally efficient similarity search. *Science Advances*, Vol. 1, no. 11, e1501057.

[3] Linville, L. B. (2019). Global-and Local-Scale High-Resolution Event Catalogs for Algorithm Testing. *Seismological Research Letters*, 90(5), 1987-1993.

[4] Linville, L. R. (2019). Global to local high-resolution event catalogs for algorithm testing and source studies. *Seismol. Res. Lett.*

[5] Thibaut Perol, M.¨. (2017, 2 7). Convolutional Neural Network for Earthquake Detection and Location. *physics.geo-ph*.

[6] Utah, U. O. (1962). University of Utah Regional Seismic Network. International Federation of Digital Seismograph Networks. https://doi.org/10.7914/SN/UU.

[7] Yann LeCun, L. B. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

[8] Zachary E. Ross, M.-A. M. (2018). Generalized Seismic Phase Detection with Deep Learning. *Bulletin of the Seismological Society of America*.

APPENDIX A. TRAINING AND LEARNING CURVES

An integral part of effectively using convolutional neural networks is understanding behavior when training. Especially in the regime of smaller quantities of signal data, it is vital to discern when the network is overtraining, how to make the network converge, when to use pretrained weights, and the effect of training signal data on performance. To reiterate, we have 10 second windows of 100 Hz three component data. We have a relatively small amount of signal data windows (~1000) relative to noise (~10,000), and in each step of training we use an equal number of signal and noise windows. Therefore, in training, the network will see less of a variety of signal data relative to noise. Here we describe important training effects we have seen, the data augmentation we perform and show the learning curves of the network.

A.1. Signal Position in the 10s Window

An important variable in the ability to train ConvNetQuake is the position of the signal onset Lg wave in the 10 second window. Initially, the signal in the training data was placed in the middle of the 10s window at t=4s. The network would converge (loss decreased) after several epochs. During test time, we test in consecutive 10s windows, overlapping by However, we noticed that in running over the test set, performance would suffer. The intuition is that the network would be trained to expect the onset of the Lg wave at t=4s and would not be robust to streaming data. We then began to shift the signal onset wave around in the window from t=0s to t=8s. When we did this with window weights initialized randomly, the loss would not decrease. We found after many iterations and much effort, that we initially needed to train with the signal sitting statically at one position in the time window (t=4s). After the initial convergence with this signal data, we were able to introduce windows containing the signal onset shifted, and the network would then recognize the possibility of the signal onset at any time in the 10s window. Performance on the test set then improved dramatically. We found this effect by plotting the percentage of missed test set events against where the signal occurred in the window, as illustrated below in *Figure 4*. When trained with signal only occurring at t=1s (**Left**) we see that the network detects events that predominantly start in a window from -4s (Lg wave starts before test window, but coda is detected). When trained incrementally with signal that starts from t=0s-7s (**Right**) the percentage of TP is more evenly distributed, as it should be and the edges of the window contain most of the FP's, mostly because those events are being picked up by the subsequent window, as it should be. When tested, the network on the right detected 231 out of 298 events, while the network on the left detected only 99.

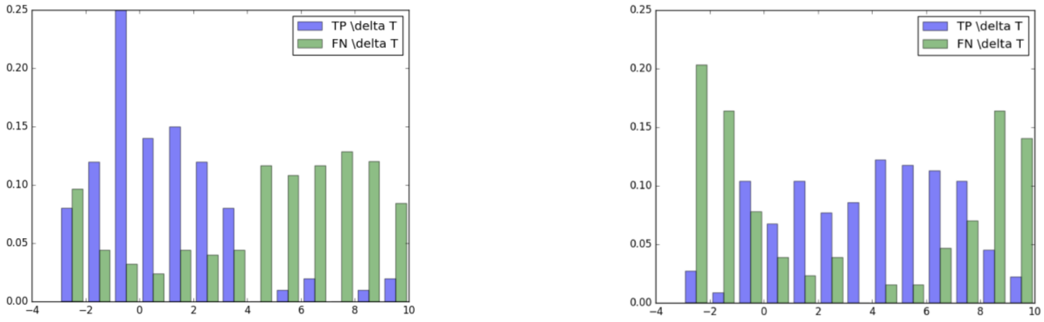

Figure 4:Percentage of True Positives (TP) and False Negatives (FN) as a function of test set PNSU Lg wave onset.

A.2. Training Curves

Here we show representative training curves for the PNSU station. As we became familiar with training, we noticed a few common effects. First, as mentioned in A.1, initial convergence was altered based on the amount of data containing events starting at t=4s. Second, for certain stations with small amounts of data, training was unstable and training curves such as loss tended to oscillate. PNSU is a representative example of training behavior that is somewhat stable. If we look at *Figure 5* we see the training set accuracy increasing significantly through epoch 20 and then leveling off to 98% in this case. The validation set oscillates and may decrease a small amount as the network overfits the training data, but largely remains stable. In *Figure 6* we see the same behavior as the loss function similarly decreases monotonically for the training set and oscillates a bit for the validation set. Unfortunately, we did observe the performance of the network does vary based on how and on which epoch one stops training. Because of the small amount of data, our validation and test sets are the same, so we pick epoch=100 a-priori to stop training to avoid bias.

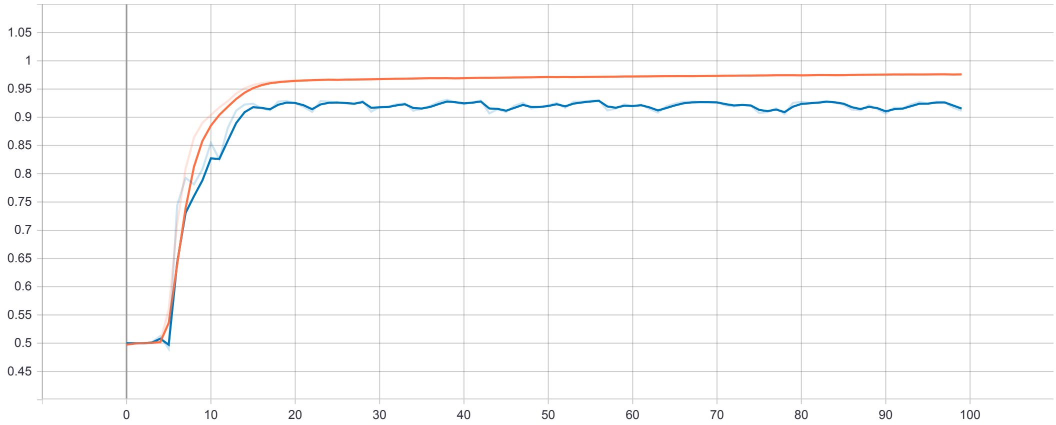

Figure 5: Accuracy of training (Orange)and test (Blue) set.

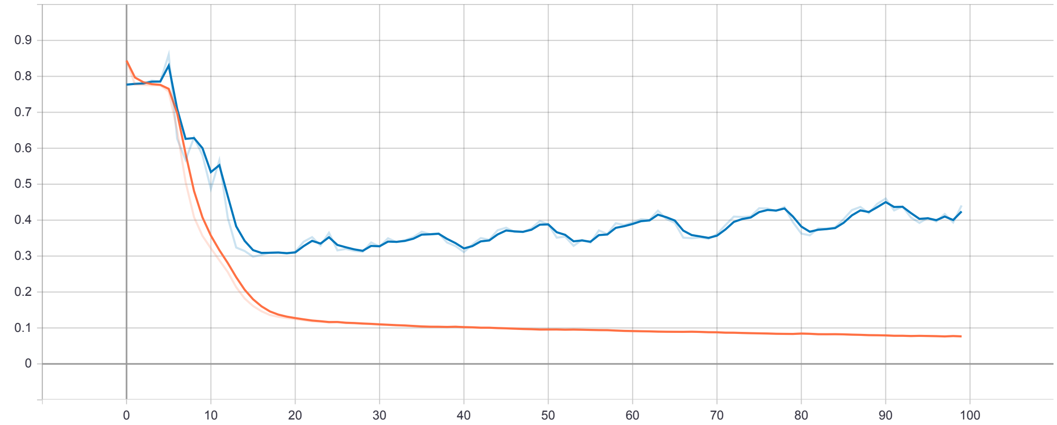

Figure 6: Loss of training (Orange)and test (Blue) set.

A.3. Overtraining

In our formulation, as described above, we train the network with a balance of signal and noise data windows. However, stations contain significantly more variety of noise than of signal. The result of this is that we are in danger of overtraining on signal while simultaneously under training on noise. As an example of overtraining, we look at a histogram of output probability (probability that event is real and not noise) of labeled test events and noise from station BRPU trained with 100 epochs versus 300 epochs in *Figure 7*. One can see that as the network overtrains, it continues to try to differentiate signal and noise with more events pushed towards the extremes, but ultimately performance degrades. After training for 100 epochs many noise events are considered uncertain, as their probabilities remain in the 0.1-0.9 range. As the network is overtrained, essentially all events are binned near 0 or 1, as the network essentially memorizes certain events or features. Ideally, we would not have a network overtrained in this way, so we could select the threshold at which to operate on the ROC curve, based on the task at hand.

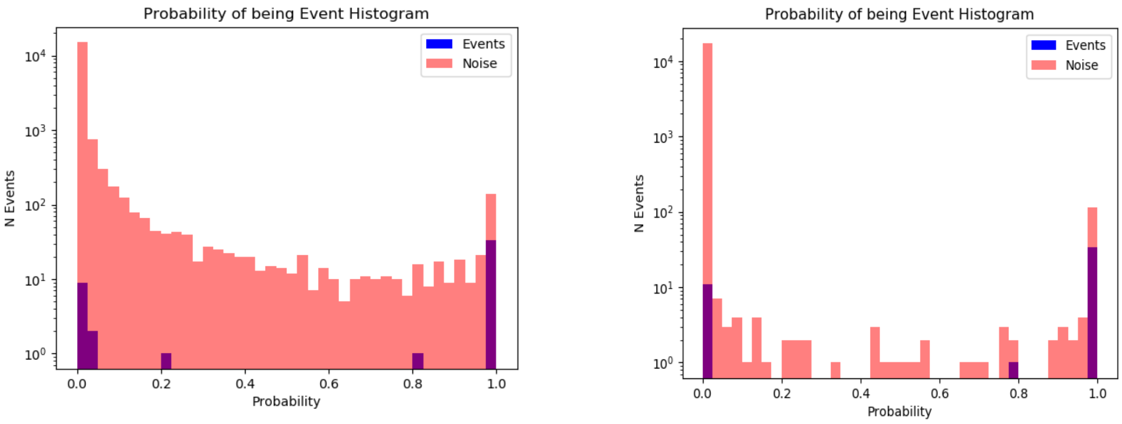

Figure 7: Probability of being an event after training for 100 epochs (Left) versus 300 epochs (right)

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Robert Forrest	08176	rforres@sandia.gov
Technical Library	01977	sanddocs@sandia.gov

This page left blank

This page left blank

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.