Seismic Signal and Noise Separation Using Deep Neural Network on Downhole Distributed Acoustic Sensing Array Records at SAFOD

Siyuan Yuan[1], **Xiaoxi Zeng**[2], **Ruiming Zhu**[3]
Geophysics Department[1] CEE Department[2] MSE Department[3]
Stanford University
`syyuan, dawnzxx, zhurm@stanford.edu`

Abstract

Distributed Acoustic Sensing (DAS) is an emerging technology that is promising in monitoring earthquakes with low cost per sensor. We implemented neural networks to denoise the 2D DAS earthquake recordings. We experimented two architectures: U-Net based models and Xception. The neural networks were trained on synthetic data and evaluated on field data. And we chose Unet1 as our best model. Using signal-to-noise ratio (SNR) as a denoising metric, we found that Unet1 performs better than the wavelet baseline model in all of the five randomly chosen validation and field data.

1 Introduction

Earthquake signal denoising is of great concern to seismologists to improve the the data quality and benefits subsequent analyses. Traditionally, 3D geophones are widely used to measure earthquakes. Zhu et al. 2019 showed that signals of geophones can be denoised using deep learning models. Recently, a new technology of distributed acoustic sensing (DAS) has been emerging as a possible alternative to the traditional geophone sensing arrays. DAS utilities a fiber-optic cable with an interrogator to measure the strain along the fiber, which enables us to perform ground motion monitoring with a dense channel distribution. Moreover, the cost per sensor for DAS is much lower than that of traditional geophone arrays. However, we have seen that DAS recordings can be contaminated with strong ambient noise and instrument noise. Herein, we aim to apply the deep learning approach to denoise DAS recordings and hence improve the data quality for subsequent analysis such as earthquake location and earthquake detection.

A DAS recording has two dimensions, one corresponding to the time lapse and the other to the channel number (spatial location). In other words, a recording can be viewed as a 2D grey scale image containing both signal and noise. The DAS fiber we use here locates in the San Andreas Fault. In total we have one month of recordings, during which more than one hundred earthquakes occurred in the surrounding area. Figure 1 shows one example of our DAS recording. The input to our neural network will be the Short Time Fourier Transform (STFT) of the original time and space image. The STFT results are in three dimensions (time, frequency and sensor index) and have two channels (real and imaginary part of the STFT). The outputs would be two masks in the STFT domain corresponding to the recovered signal and noise respectively. Each pixel of the masks have value of a float number between zero to one, with zero to be pure noise and one to be pure signal. The loss function is a cross-entropy loss.

2 Related work

The U-Net architecture has shown to to be fast and precise in bio-medical image segmentation (Ronneberger et al., 2015). Zhu et al., 2019 used the U-Net architecture to develop a deep learning denoisier for 1D time series of the traditional geophone data. The authors trained their neural networks on various types of noise and earthquake signals and demonstrated that with their deep U-Net neural network, better signal-to-noise (SNR) ratio has been achieved compared to traditional denoising approaches (normal Spectral filtering and GCV denoising), while minimizes changes in the waveform shape of interest. Both of our work aim to denoise earthquake recordings. However, in our case, since we have 800 sensors along the fiber to record signal at the same time, we have one more dimension than their data.

CS230: Deep Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Moreover, they have an advantage of having an easy access to millions of clean signal as the ground truth, while in our case, we have only one hundreds of earthquake recordings and non of them are clean. This is our biggest challenge.

The DeepLab v3+, similar to the U-Net, also employs an encoder-decoder based structure and showed the state-of-the-art performance in image segmentation problems [3]. DeepLab uses multiple encoders including MobileNet[4], Xception[5], and ResNet-101 [6]. When implementing with Xception as encoder, DeepLab model has a better performance for image segmentation. This architecture may be promising for our earthquake denoising problem as well.

3 Dataset and Features

The SAFOD DAS data are collected with a 1D fiber down to a depth of 864m in the subsurface. Due to a failure of the loop at the end of the fiber, we limit our analysis to a depth of 800m. Along this depth, we have 800 channels with 1m spacings continuously monitoring ground motion during three weeks from 21 June, 2017 to 10 July, 2017. During this period, around 100 earthquakes occurred within a radius of 63 km away from the fiber. Figure 1 shows an example of a SAFOD earthquake recording. The vertical axis is channel number (No.0 is the channel on the ground and No.800 is the channel at the very bottom of the fiber.) The horizontal axis is the time axis. The sampling rate of the data is 100 Hz.

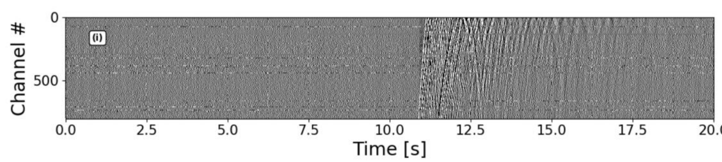


Figure 1: An example of SAFOD earthquake recordings

Since we do not have clean DAS signal and DAS signals share similarities with the geophone data. We synthesized clean DAS signal by shifting geophone signal along the depth according to the P and S wave velocity model along the fiber given by Ariel at el., 2019. To mimic real DAS signal, we added randomness on the the synthetic data (we added random small time shifts along the depth, randomly kills signal of some randomly chosen channels and add random amplitude factors for each channel). Figure 3 shows an example of the synthetic DAS data using one of the geophone data. Then we added randomly chosen noise data from our field dataset to the synthetic clean signal to get synthetic noisy data. We preprocessed the recordings by performing the STFT and we normalized our data for each of the 800 traces by their corresponding L-2 norms. In total, we have 1500 synthetic samples in the training set each with 50 seconds. In order to let neural networks see different shifted versions of signal and different noise, for each of the 100 epochs, we randomly windowed 20 seconds of those 50 second recordings and randomly select 20 seconds noise from our one month recordings to make synthetic data. And we select the hyperparameters based on the performance on the 140 validation data. And we use 140 test data and field data to evaluate the performance of our model.

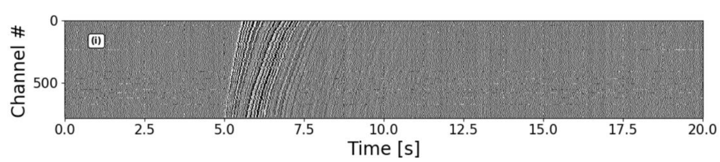


Figure 2: An example of the synthetic DAS data using one of the geophone data

4 Methods

Convolution Neural Networks (CNN) encoder-decoder architecture is able to learn efficient representation in computer vision tasks. Our encoder network maps the preprocessed 3D STFT spectrum, which contains two channels (real and imaginary parts of the STFT), to low-dimension features. The decoder network maps these features to generate output with the original dimension. Since the outputs are signal and noise masks with elements between zero and one, our task is equivalent to predict the probability of being signal for each pixel of the inputs. Thus, we adopt image segmentation approaches. We use the cross-entropy loss function to optimize our model:

$$L(p, y) = -[ylog(p) + (1 - y)log(1 - p)]$$

We developed our training framework in TensorFlow and compared the performance of two classic image segmentation architectures: one is U-Net [2], the other is DeepLab with Xception as backbone [3].

4.1 U-Net based architecture

The U-Net based architecture consists of a series of fully convolutional layers with 10 descending encoding and 10 ascending decoding layers. Skip connections was implemented to improve the convergence of training and prediction performance by passing low-level features into decoder. Figure 3 shows the U-Net architecture.

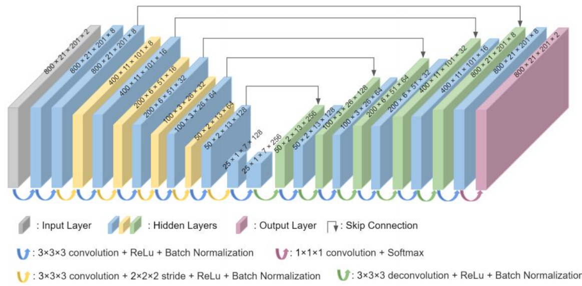


Figure 3: U-Net Based Network Architecture

4.2 Xception based architecture

We also explored DeepLab v3+, a state-of-the-art image sementation model, with Xception as a backbone model. Its encoder-decoder architecture uses Xception as an encoder module to extract multi-scale information and a simple decoder module to refine the results.

5 Experiments/Results/Discussion

5.1 Hyperparameter tuning

For U-Net based model, we experimented three different learning rates, 1e-2, 1e-3 and 1e-4, with the same decay rate of 0.95. Our GPU memory can only fit a maximum batch size of two. As a result, We tried both batch sizes of two and stochastic gradient descent. To avoid overfitting, we added dropout for Unet1 and Unet5. When training them, we added pure noise in the training set to let the networks learn noise patterns. Table 1 shows both training and validation loss for all combinations of hyperparameters we tested. Among the six, Unet1 performs the best with the lowest training and validation loss. We found that a higher learning rate tends to fasten convergence and results in a lower training error. And a batch size of two helps convergence because of the batch norm technique we implemented in our U-Net. Comparisons of Unet1 and Unet4 show that dropout decreases variance. Similarly, dropout brings down the variance of Unet5 compared to Unet6. The validation and training loss for Unet1 is the close, which indicates that Unet1 does not overfit the training data.

For Xception based model, we experimented three different learning rates, 1e-2, 1e-3 and 1e-5, with the two decay rate of 0.95 and 0.98. Our GPU memory can only fit the maximum batch size of one. As a result, We used stochastic gradient descent. And to avoid overfitting, we added weight decay for Xception 1, 2 and 3. Table 2 shows training loss for all combinations of hyperparameters we tested. Among the four, Xception1 performs the best with the lowest training loss. We found that a lower learning rate tends to help gradient decent to achieve lower training error. In order to overfit one training example, We babysat it by adjusting network dimension and implementing new output layer. However, because of the complexity of the Xception, the model is hard to converge. We failed to overfit one training example and train a effective model.

In conlusion, U-Net is a better network structure for our project compared to Xception. we chose Unet1 as our best model, which has a test loss of 0.347.

5.2 Model evaluation

We chose the signal-to-noise ratio (SNR) to evaluate the performance of our model. The SNR is calculated as:

$$SNR = 10log_{10}(\sigma_{signal}/\sigma_{noise})$$

where σ_{signal} and σ_{noise} are the standard deviations of waveforms before and after the first arrival. We chose wavelet denoising filter in python skimage package as our baseline model [7], which is commonly used for denoising images. Figure 5 shows two field earthquake recordings, FieldData1 and FieldData2, used to test the performance of our model in real world. Comparisons of (a) and (b) show that for the FieldData1, U-Net has less signal leaking problem than the baseline model. And U-Net did better on removing the noise before earthquake arrival (around 10.9s). Figure 6 (c) and (d) show respectively in spectrum and time domains signal is well recovered with Unet1 for sensor #1 of FieldData1. Figure 5 (c) and (d) show that for FieldData2, Unet also performs better than baseline with less signal leak problem and more noise energy are removed from recovered signal. In table 3, we show and compare the SNRs before and after denoising with the two models on two field data and three randomly selected validation data. The recovered signals by U-Net model have higher SNRs of the three validation data than those by the baseline model by a factor of 5.416 on average. While for the three field data, the factor is 2.016. In all five cases, U-Net achieves better SNRs of the recovered signal.

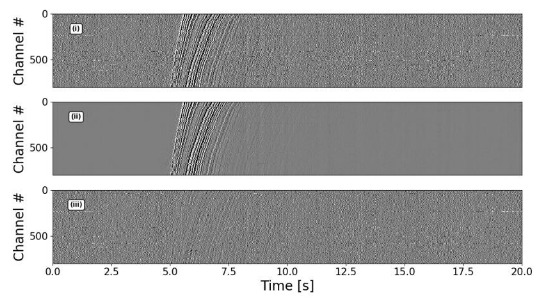


Figure 4: Denoising performance on an unseen synthetic seismograms. Real noisy signal are plotted in panels (i). Panels (ii) shows the denoised signal. The recovered noise is shown in panels (iii).

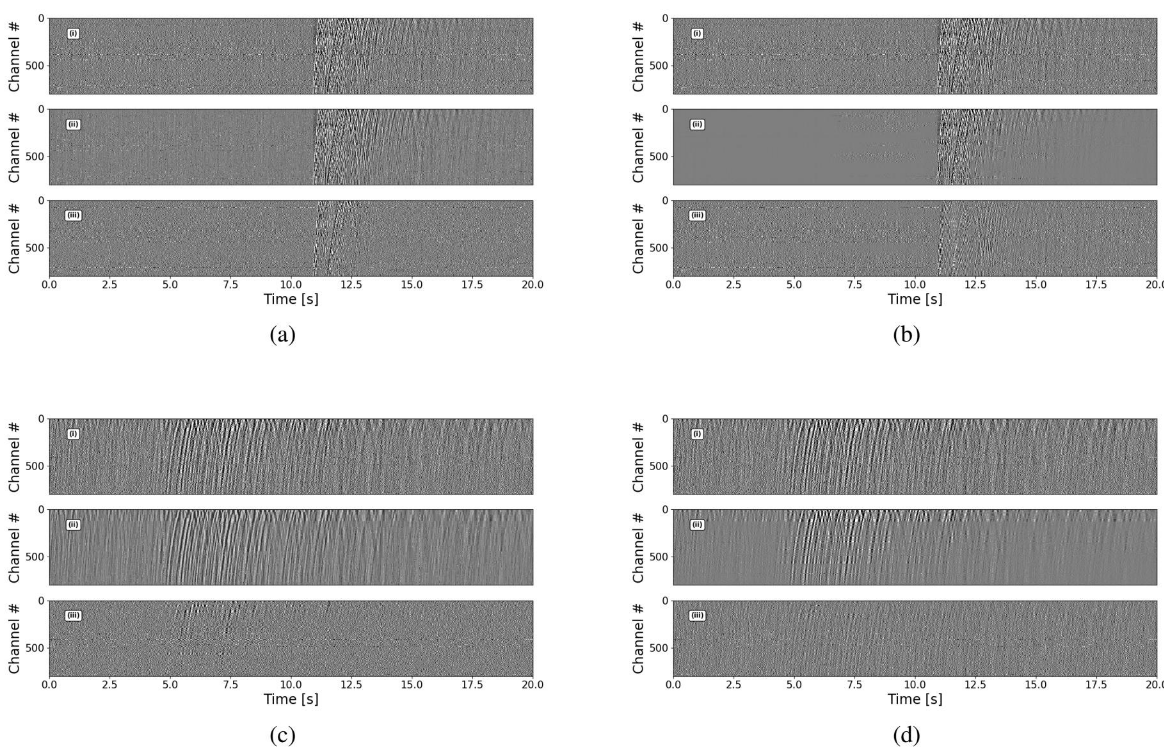
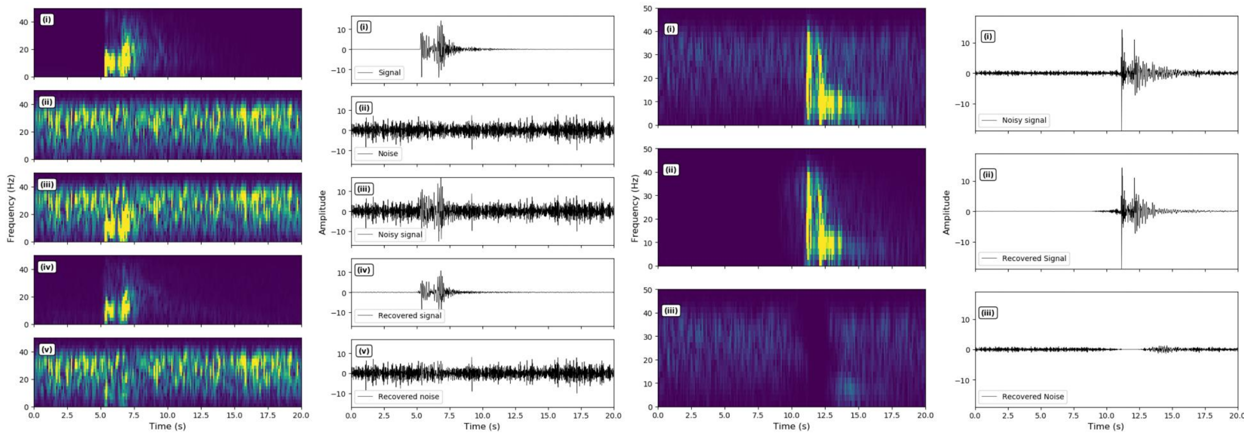


Figure 5: Denoising performance on two SAFOD seismograms, (a, b) FieldData1; (c, d) FieldData2: (a, c) Baseline model; (b, d) U-Net based model. Real noisy signal are plotted in panels (i). Panels (ii) shows the denoised signal. The recovered noise is shown in panels (iii).

6 Conclusion/Future Work

We implemented both U-Net and Xception architectures on the 2D synthetic DAS recordings. We found that U-Net is much easier to train and converge than Xception because of the simplicity of the model. After tuning hyperparameters,



(a) (b) (c) (d)

Figure 6: Denoising performance on (a, b) unseen synthetic seismograms (trace 1 of seismograph shown in Figure 4); (c, d) real SAFOD seismograms(trace 1 of FieldData1 shown in Figure 5): (a, c) time-frequency domain; (b, d) time domain. clean signal, real noise, and noisy signal are plotted in panels (i) (ii) (iii). Panels (iv) show the denoised signal. The recovered noise is shown in (v).

Table 1: U-Net Based Model Performance

Model	**Hyper-Parameters**				**Cross-entropy loss**	
	Learning Rate	**Batch Size**	**Dropout**	**Include Zero Signal**	**Training**	**Valid**
Unet1	1e-2	2	0.2	Yes	0.318	0.336
Unet2	1e-4	2	0	No	0.342	0.441
Unet3	1e-3	2	0	No	0.320	0.661
Unet4	1e-2	2	0	No	0.321	0.723
Unet5	1e-2	1	0.2	Yes	0.337	3.077
Unet6	1e-2	1	0	No	0.340	174.427

Table 2: Xception Based Model Performance

Model	**Hyper-Parameters**				**Cross-entropy loss**
	Learning Rate	**Batch Size**	**LR Decay Rate**	**Weight Decay**	**Training**
Xception1	1e-5	1	0.98	1e-8	0.529
Xception2	1e-3	1	0.99	0	0.644
Xception3	1e-3	1	0.95	1e-4	0.777
Xception4	1e-2	1	0.98	1e-8	0.785

Table 3: SNR Performance With Baseline and Best Model

Data set	**Noisy Signal SNR (dB)**	**Denoised Signal SNR (dB)**	**Baseline Model SNR (dB)**
Validation Data Sample 1	3.78	18.18	3.72
Validation Data Sample 2	1.87	18.46	3.31
Validation Data Sample 3	0.84	14.46	2.50
Field Data Sample 1	0.77	3.69	2.21
Field Data Sample 2	6.52	13.00	5.50

we chose a U-Net model with the least training and validation loss as our best model. Wavelet denoising algorithm is chosen as the baseline model. We compared the performances of the two models on five randomly selected synthetic and field data. The recovered signals by U-Net model have higher SNRs of the three validation data than those by the baseline model on average by a factor of 5.416. The factor for the two field data is 2.016. In all five cases, U-Net achieves better SNRs of the recovered signal. We found that in all cases, the U-Net model achieved higher SNRs. For future work, we could use more synthetic data to train our models, so that new models learn more general cases to improve their performance on field data. Moreover, with larger GPU memory, we would experiment larger batch sizes with batch norm.

7 Contributions

All three authors contributed equally in architecture selection, training and testing the model. Source code specific to this project can be found at: `https://github.com/syyuan93/deep_learning_denoise_SAFOD.git`.

8 Acknowlegments

We thank our TA Cristian Bartolome Aramburu and all teaching staff in CS 230 for delivering a such great class and helpful discussion. We thank our friend Weiqiang Zhu for sharing us with experience in U-Net implementation and great insights to keep us going. We thank Center for Computational Earth and Enviromental Science (CEES) for computational resources. We thank Stanford Center for Induced and Triggered Seismicity (SCITS) and OptoSense for providing us with the DAS recordings. We thank Stanford Exploration Project (SEP) for funding.

References

[1] Zhu, Weiqiang, S. Mostafa Mousavi, and Gregory C. Beroza. "Seismic signal denoising and decomposition using deep neural networks." arXiv preprint arXiv:1811.02695 (2018).

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3] Chen, Liang-Chieh, et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[4] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[5] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[6] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[7] Van der Walt, Stefan, et al. "scikit-image: image processing in Python." PeerJ 2 (2014): e453.