CS 229

Real-time microseismic P- and S-wave arrival picking for hydraulic fracturing

Wei Chu
Department of Geophysics
Stanford University
weichu@stanford.edu

Iris Yang
Department of Geophysics
Stanford University
iyyang@stanford.edu

1 Introduction

Microseismic event monitoring during the hydraulic fracturing operation is an important technology for evaluating the fracture network effectiveness and for preventing operation hazards. When the rock is fractured, it generates microseismic events comprised of P-, S-, and other wave modes, which are detected by a series of geophones located a few hundreds of meters away from the event locations. A real-time algorithm for picking the P- and S-wave arrivals, in particular, is instrumental to making real-time drilling and stimulation decisions: it allows the operators to estimate the microseismic event locations, the extent of the fracture growth, and any possible drilling hazards on the fly.

In the application of locating natural earthquakes, the P- and S-wave arrivals are either manually identified by experienced analysts, or automatically picked by phase picking algorithms, such as the STA/LTA method proposed by Allen (1978), and PhaseNet by Zhu and Beroza (2018). The arrival picking in the microseismic survey, compared to locating earthquakes, demands much higher accuracy: while it takes tens of seconds for the earthquake signal to travel from the source to geophones, the hydraulic fracturing induced microseismic event only takes hundreds of milliseconds to reach the geophones. A tens of milliseconds off in the arrival picking would lead to tens of meters off in estimating the extent of the fractures, a scenario certainly undesirable for the purpose of making real-time decisions. Therefore, a high-fidelity automatic arrival picking algorithm is highly sought after for microseismic surveys.

In this project, we built multiple models leveraging different machine learning algorithms, including the random forest, two-dimensional convolutional neural network (2D-CNN) and 3D-CNN to detect P- and S-wave arrivals in real-time. The two CNN models outperform the random forest model, and achieve satisfactory prediction accuracy. We also performed sensitivity analysis to test the robustness of the models. Results demonstrate high reliability in the CNN architectures.

2 Dataset and Features

2.1 Data Acquisition and Representation

We used the microseismic dataset described in Zhang et al. (2018). The data were collected during a five-stage hydraulic fracturing stimulation operation performed in the Vaca Muerta Formation at Neuquén, Argentina. The left plot of Figure 1 shows the stimulation and microseismic survey setup. The data were acquired by a 12-level double-stacked geophone (receiver) array placed in a vertical well, approximately 100 m shallower from the perforation locations. Each geophone records

CS229: Machine Learning, Autumn 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 waveform traces simultaneously along the 3 polarization directions (north-south, east-west and vertical). The sampling rate of the recording is 0.25 ms. All the waveforms have 2000 time steps, equivalent to a time duration of 500 ms. The data were recorded from simulation stage 1A, stage 1B and stage 5. The numbers of triggered microseismic events are 404, 884, and 377, respectively. Additionally, the P- and S-wave arrivals for each microseismic event had been manually picked by a vendor company, which serve as our labels.

We define our data as follows. In 2D-CNN models, each sample is a collection of waveforms (time series) from all the 12 geophones. As a result, each sample is a three-dimensional tensor of size (2000, 12, 3); the label is an integer vector of length 12, indicating the indices of either the P- or S-wave arrival picks from all the 12 geophones. The right plot of Figure 1 shows one of our samples for the 2D-CNN model. The purple and red dots represent the P- and S-wave arrivals, respectively. In the random forest models, instead of using waveforms from all the 12 geophones altogether, we treat waveforms from each individual geophone as a single sample with a size of (2000, 3). We further flattened it into a vector of size (6000, 1). The label for each sample is a single integer value reflecting the index of the first arrival. The data representation for the 3D-CNN is described in data preprocessing section.

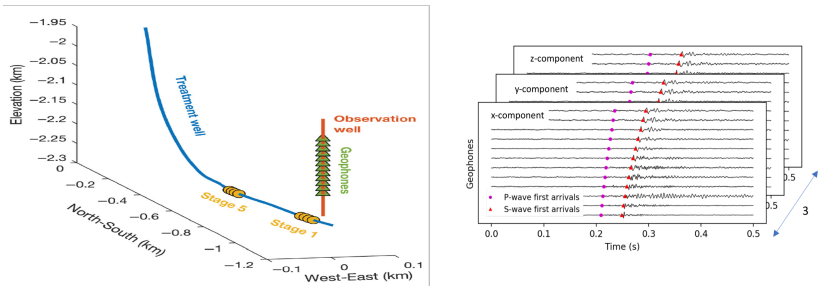

Figure 1: Acquisition setup (left) and the representation of a single sample in the time domain (right).

2.2 Data Preprocessing

We scaled the amplitudes of each waveform to -1 and 1 by dividing by the maximum and minimum amplitudes. In addition to analyzing data directly in the time domain, we also converted them into the time-frequency domain, which was shown to be beneficial for picking P- and S-wave arrivals (Zhu et al., 2018). The data are represented as spectrograms in the time-frequency domain. After hyperparameter tuning, we chose a Hamming filter with a window size of 200, and a number of overlap of 195. Those design choices are primarily based on the fact that we would like to maintain sufficient frequency resolution in 0-500 Hz, which is essentially the signal bandwidth, while keeping enough temporal information. As a result, the dimension of each sample is changed from $(2000, 12, 3)$ in the time domain to $(12, 361, 25, 3)$ in the time-frequency domain. We further scaled the frequencies of each spectrogram to -1 and 1. Figure 2 shows the data transformation of a single geophone in each polarization direction.

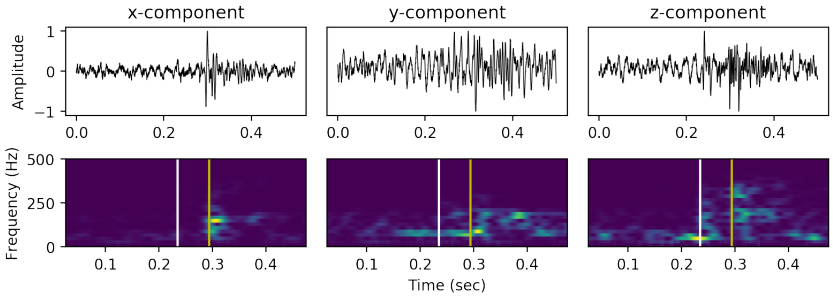

Figure 2: Time series(upper) and spectrograms(lower)

3 Methods

We experimented with three different models for picking P- and S-wave arrivals: a random forest regression model, a 2D-CNN model and a 3D-CNN model. We further performed sensitivity analysis for the CNN models.

3.1 Random Forest Regression

Random forest is an ensemble tree model. It constructs multiple decision trees, where each tree is trained on bootstrapped samples from the raw data. In addition, each training step is restricted to a random sub-feature space when splitting a branch. When making predictions, the model uses the average of each tree's prediction as the final result.

We chose random forest as our baseline model because of its high flexibility - it does not require any assumptions from the data. In addition, the overfitting problem of tree models is mitigated by applying random sub-feature space splitting and data bootstrapping. Another reason is that random forest runs fast. After hyperparameter tuning, we chose $n\_estimators = 100$, $max\_features = 80$, $min\_samples\_split = 30$, and $min\_samples\_leaf = 10$ as the optimized setup. We used this model as the baseline to compare with other models.

3.2 2D- and 3D-CNN

We chose the 2D-CNN as our model deign for two reasons. First, the arrival picking is strongly related to both the time and the spatial dimensions, suggesting a two-dimensional scan over the inputs. In addition, the three-component waveforms are analogous to the RGB channels of images, where the CNN proves to perform very well. We further designed a 3D-CNN architecture to accommodate the additional features along the frequency dimension in spectrograms. To better capture the patterns in the dataset, we applied the same idea of the architecture design for both 2D- and 3D-CNN: the number of filters to gradually increase, and the filter size to gradually decease along the forward direction. A rectified linear unit (ReLu) activation function is used after each convolutional layer, followed by batch normalization, and sometimes also the maxpooling and dropout layers. We used the Adam optimizer to fit the models with the initial learning rate of 0.001 (0.001 gives the desirable loss decreasing), followed by linear decays after each epoch. Note that less convolutional layers were used, while more maxpooling and dropout layers were added in the 3D-CNN model, to reduce the architecture's high variance. Figure 3 shows the architecture design of the two models.

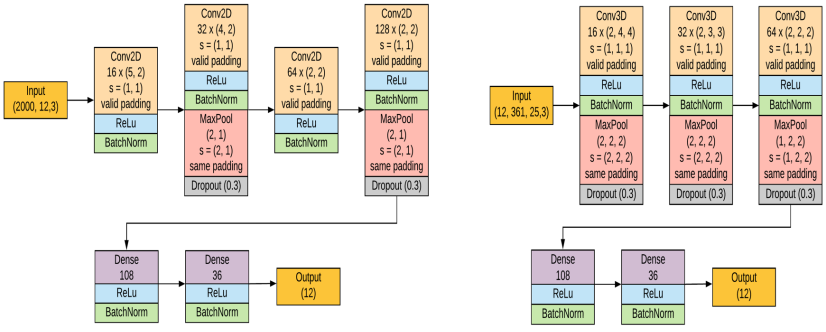

Figure 3: 2D-CNN design(left) and 3D-CNN design(right).

3.3 Sensitivity Analysis

We observed that the P- and S-wave arrivals follow a "hyperbolic pattern" in the time domain along the 12 geophones. This is also proved to be true for uniformly distributed geophones by (Dunkin and Levin, 1973). We managed to take the advantage from it and apply it to more application scenarios. It is not uncommon to see one of the 12 geophones breaks down during the hydraulic fracturing operations due to the downhole harsh working environment. A broken geophone will

block the corresponding three polarization component waveforms and make arrival picking more difficult. Since our CNN model take the full trace (12 geophones) as its input, we hope it can learn the hyperbolic pattern from the data even when one of the geophones does not work. Hence we designed an experiment to test the robustness of our model to see if it can make predictions appropriately when one or more geophones are dead.

During the training stage, we train the model with the full trace data as the same as we mentioned above. When it comes to the test stage, we arbitrarily block one geophone (3 traces) of the test sample by setting all the waveform along those traces to be zero (this mimics the scenario that a geophone is completely dead), and test if the model can predict the missing traces based on its adjacent traces. We further run the experiment with 3 geophones blocked for each sample to see how robust the model is on more missing data.

4 Results and Discussion

We trained the random forest, 2D-CNN and 3D-CNN on P- and S-waves separately. For the random forest, the train/test split is 0.8/0.2. For the two CNNs, the train/dev/test split is 0.7/0.15/0.15 with 120 epochs and a minibatch size of 128. We chose the mean square error $MSE = \frac{1}{m}\sum_i^m (y_i - \hat{y}_i)^2$ as the optimizing metric, since it is a regression problem that predicts the time of arrivals (although we defined the index of arrival locations as the direct estimator). We further defined A_5, A_{10} and A_{20}, the accuracy under different satisfying levels (5 ms, 10 ms and 20 ms) as our satisfying metrics to better interpret predictions. For instance, $A_{10} = \frac{\sum_i^m \mathbb{1}\{|y_i - \hat{y}_i| < 10\}}{m}$ indicates the proportion of the predictions with error less than 10 ms.

The results are shown in Table 1. The letter “P" represents models predicting P-wave arrivals, while the “S" represents models predicting S-wave arrivals. The letter “B" indicates the number of blocked geophones for the sensitivity analysis. For instance, “3D-CNN_S_B3" refers to the 3D-CNN model for picking S-wave arrivals where 3 geophones were blocked. By comparing training and test MSEs, we observe that our models have encountered overfitting problems. We put efforts into addressing this problem by tuning $max\_features$, $min\_samples\_split$ and $min\_samples\_leaf$ for the random forest model, and by changing the architecture and adding more maxpooling and dropout layers for the CNN models. The remaining MSE differences we saw in the table may come from the lack of the samples - we have only 1665 microseismic events in total. We note that there is no need to compare the absolute MSE values between the 2D- and 3D-CNN models: when we transformed time series to spectrograms, we changed the spacing in the time domain, so that the total time steps was reduced from 2000 to 361. The MSEs, which correspond to differences between the true and predicted indices, have different scales for 2D- and 3D-CNN cases. Therefore, we focused primarily on the satisfying metrics A_5, A_{10} and A_{20} to evaluate models.

Table 1: P- and S-wave arrival picking results

Model	A_5	A_{10}	A_{20}	TrainingMSE	TestMSE
RF_P	26%	46%	65%	6223	16004
2D-CNN_P	45%	70%	89%	533	2082
2D-CNN_P_B1	42%	67%	87%	437	2153
2D-CNN_P_B3	24%	51%	81%	563	8282
3D-CNN_P	60%	82%	91%	27	141
3D-CNN_P_B1	52%	79%	89%	22	139
3D-CNN_P_B3	25%	52%	88%	41	146
RF_S	27%	48%	70%	5027	15360
2D-CNN_S	59%	86%	97%	398	1179
2D-CNN_S_B1	56%	87%	96%	826	1854
2D-CNN_S_B3	40%	68%	90%	618	4699
3D-CNN_S	55%	86%	95%	52	410
3D-CNN_S_B1	52%	82%	95%	71	365
3D-CNN_S_B3	34%	65%	89%	47	215

The random forest model achieves between 65% and 70% for A_{20}, indicating that around 65-70% of the predictions have error less than 20 ms. This is not bad compared to natural earthquake arrival picking, but apparently worse than the CNN models, which have A_{20} higher than 90%, and A_{10} higher than 70%. One possible reason for the CNN to outperform is the spatial arrangement of the geophones: the CNNs can readily learn the “hyperbolic pattern", because the they are fed by a full suite of time series. We can infer that the CNN is more powerful than random forest for time series problems.

For the S-wave picking, both CNN models perform equally well with A_5 around 60%. When it comes to P-wave, A_5 for 2D-CNN dropped to 45% while that for 3D-CNN remained at the same level. By examining the data, we found that there are much stronger perturbation contrast in S-wave arrivals than that P-waves. The 2D-CNN, which is trained purely based on the time domain, may have failed to pick up the weak P-wave signals. In contrast, the 3D-CNN may have captured more P-wave signals from the frequency components, which results in better performance.

When one geophone was blocked, the average reduction of A_5, A_{10} and A_{20} for the 2D-CNN is 2.7% for the P-wave arrival estimates, and 1.7% for S-waves. The average reduction for the 3D-CNN is 4.3% for P-waves, and 2.3% for S-waves. This indicates that the CNN models are robust enough to maintain a high-level performance when one geophone is shut down. When the number of blocked geophones is increased to three, the average reduction of accuracy for the 2D-CNN becomes 14.7%-16%, and 16%-22.7% for the 3D-CNN. We can see the that CNN models are no longer reliable when missing three geophones. Due to the limitation of the space, here we only present one of our prediction results. Figure 4 shows the S-arrival prediction from the 2D-CNN model. The blue dots represent the true arrival of the 12 geophones, the red dots represent the prediction with made with the full trace data while the green dots represent the prediction made with one geophone broken. We clearly see that all predictions aliened very nicely with the true labels, demonstrating both high performance and high robustness of the model.

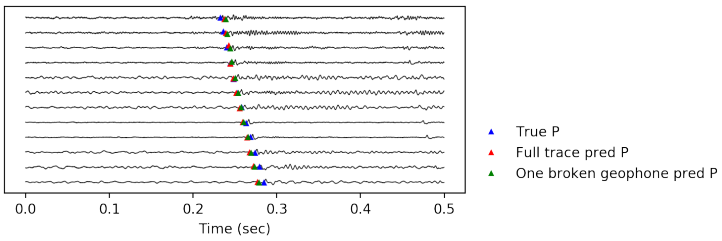

Figure 4: S-arrival prediction from 2D-CNN

5 Conclusions

- We explored three models for microseismic P- and S-wave arrival picking. Both 2D- and 3D-CNN models outperformed the random forest model, and achieved very high prediction accuracy with A_{20} higher than 90%.
- The 3D-CNN outperforms the 2D-CNN in P-wave arrival picking, because the P-wave perturbation is much less than that of the S-wave in the time domain.
- Both CNN models are robust even with one geophone completely missing. This is because the models take the advantage of the spatial arrangement of the 12 geophones. Both models become less satisfactory when missing three or more geophones.

In the future, more work could be done including training and testing the models on larger datasets, combining the time series and spectrogram CNN models into a single model, conducting further sensitivity analysis to see which geophone affects prediction the most and so on.

6 Contributions

Wei Chu and Iris Yang share equal contribution in processing data, formulating machine learning models, and drafting writeups.

7 Code

Our source code is located at: `https://github.com/ffzmm/CS229_project.git`

References

Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces: Bulletin of the Seismological Society of America, **68**, 1521–1532.

Dunkin, J., and F. Levin, 1973, Effect of normal moveout on a seismic pulse: Geophysics, **38**, 635–642.

Zhang, Z., J. Du, and F. Gao, 2018, Simultaneous inversion for microseismic event location and velocity model in vaca muerta formation: GEOPHYSICS, **83**, KS23–KS34.

Zhu, W., and G. C. Beroza, 2018, Phasenet: A deep-neural-network-based seismic arrival time picking method: arXiv preprint arXiv:1803.03211.

Zhu, W., S. M. Mousavi, and G. C. Beroza, 2018, Seismic signal denoising and decomposition using deep neural networks: arXiv preprint arXiv:1811.02695.