CS230

Application of Deep Learning in Subsurface Faults Detection With Seismic Data

Anatoly Aseev
aaseev
aaseev@stanford.edu

Andrew Deng
andrewde
dsyandrewdeng@gmail.com

Di Fan
difan
difan@usc.edu

Abstract

Automated fault detection is an intriguing topic in the field of subsurface geoscience. Unlike other researches we propose not only fault detection but also its classification by using Convolutional Neural Network(CNN) to classify 3 types faults. Our dataset consists of synthetic and real data obtained from publicly available seismic data. Achieved high accuracy ($f_1 = 0.95$) for the synthetic set is nearly to the domain expert level. A class activation map model based on CNN is utilized to show the location of the fault on full seismic sections. To see the faults detection performance for reverse or strike slip settings, neural style transfer technique has been applied.

1 Introduction

Faults interpretation is a significant part of modeling of the subsurface processes of any scale. At regional scale, faults model defines the geological processes associated with faulting such as fluid migration, heatflow, erosion estimation, oil and gas trap preservation. In turn, in reservoir or local scale, faults model governs the oil and gas trap geometry (reserves volumetrics), pressure compartmentalization for CO_2 storage etc.

Seismic reflection data is the main source of data for fault interpretation. Faults in a seismic image (Figure 1) are often recognized as laterally high discontinuity or low continuity of seismic reflections. Currently, a great deal of seismic interpreters utilizes manual or semi-automated faults detection and extraction. It is entirely guided by domain experts and often looks “geologically correct”, but is biased, extremely time-consuming and not repeatable.

In this work, we aim to employ the same mindset for automatic faults interpretation by means of Convolutional Neural Networks (CNN), which have been extensively utilized in identifying faces, objects or animals. Modern CNNs perform better than humans in recognizing different types of images. The main advantage of this method is that there is no need to use prior seismic pre-processing and attributes extraction as well as further post-processing. Moreover, this method can work directly from seismic amplitudes and is completely un-biased.

2 Related Work

Recently in the geophysics community, there is a growing interest in applying deep learning and particular CNN in seismic faults interpretation [5][2][3][6]. All of them use CNNs with different architectures. Today, all automatic faults interpretation techniques are being used for interpretation of only one type of faults, as a most common type of fault on seismic data. Most researchers have used

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

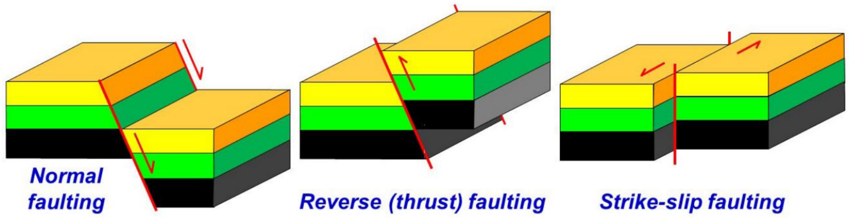

Figure 1: Simplified classification of the faults

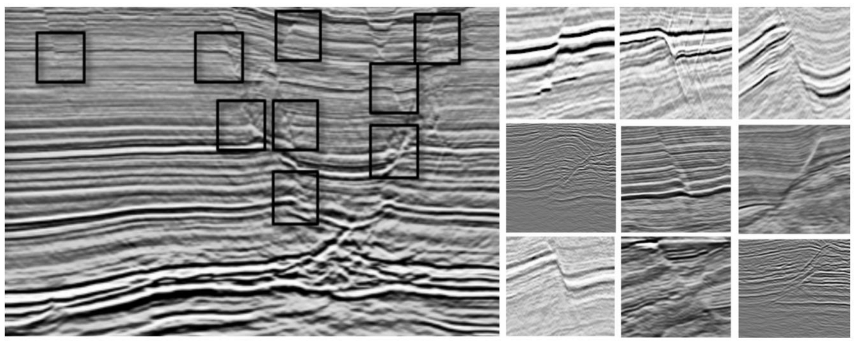

Figure 2: Collected 100x100 pixels images from public seismic datasets

synthetic seismic data [1][3][4] as a train set for machine learning using CNN for faults interpretation in general. The results achieved by research in automatic faults interpretation using CNN are good and in general may indicate a proof of concept that CNN can be utilized for automatic fault interpretation.

3 Dataset

3.1 Preparation

We used publicly available 2D and 3D seismic datasets from New Zealand and Norway (Figure 2). We collected about 1500 images of normal fault images and about 2000 images without any faults on seismic. The image size is 100x100 pixels on average, but vary in general.The raw seismic images were also supplemented with synthetic seismic data generated using Dave Hale's image processing tools for faults[7]. The tools were used to create synthetic images falling into 4 categories, no fault, normal fault, reverse fault, and strike slip fault. Normal fault category includes faults with the downward displacement from low to high magnitudes; fault plane dipping angles from 40 to 75 degrees. Reverse fault category includes faults with the upward displacement from low to high magnitudes; fault plane dipping angles from 20 to 75 degrees and random superimposing of geological folds. The latter is a typical deformation style during formation of reverse faults. Strike slip fault category includes faults with both downward and upward displacements of only low magnitudes, and sub-vertical fault plane dipping angles from 80 to 90 degrees. In order to simulate seismic image looking like real data, seismic horizons were distorted: sheared, folded and inclined. Each type of image was generated by first generating a random image of layered densities, then applying image transformations with random parameters for throw distance (displacement) and dipping angle to match the specified type of fault. In addition to random transformations, Gaussian noise was applied with random variance. Using this method, 50000 synthetic images were generated.

3.2 Split and Preprocessing

To prepare the images for input to the CNN, they were first split into training, validation, and test sets. Since we have a large amount of data, we chose the train/validation/test split to be $90/5/5$. Then, using the Keras ImageDataGenerator API, each image was set to be converted to grayscale, resized to a resolution of (256×256), and randomly shifted, scaled, and flipped horizontally. Rotation and vertical flipping were omitted because they would cause confusion between the classes; a reverse fault flipped vertically could become a normal fault, or a strike slip fault rotated slightly could become a normal fault.

4 Method

To analyze the seismic fault well, we first would like to classify the four types of the seismic images. Our approach uses two models. The prediction model which is based on convolutional neural network and is used to classify the four types of the seismic images with three types of faults. Then, a class activation map model could be utilized to help in the understanding of performance in identification of faults with the output for the VGG-19 transfer style neural network for reverse and strike slip faults, and real seismic sections for normal faults.

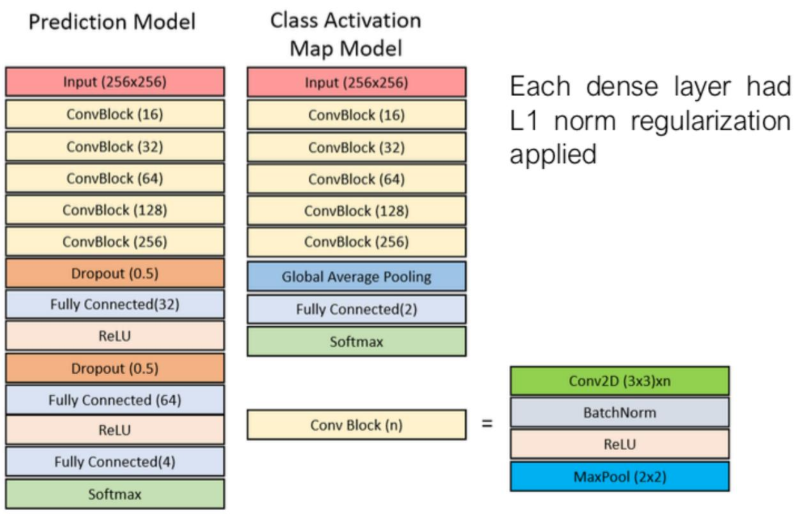

Figure 3: CNN models used for the project

We used CNNs as our model of choice to perform classification because of how well they generally perform on images. Early in the design process we replicated models in existing literature [2][4]. However, we found that these networks only achieved accuracies of around $70-80\%$ on our training data, so we made them deeper and added more dense layers to the end of the model. After adding these our training accuracies rose to 95% but our validation accuracies trailed this at 80%. To combat this we added dropout layers and added ℓ_1 regularization to the dense layers.

We primarily use the CNN model on the left of the above figure to perform classification into each of the 4 categories of faults. Then to form class activation maps, we freeze the trained weights in the convolutional layers and modify the end of the network.

The main prediction architecture consists of several convolutional blocks, where each block consists of a convolutional layer, a batch normalization layer, a ReLU layer, and a max pooling layer. We place the batch normalization layer immediately after the convolutional layer so that the input to the ReLU is centered around 0. After cascading several of these blocks, the activations are flattened into a single layer and passed through 2 fully connected layers with ReLU activation functions, then a final fully connected layer of 4 units with a softmax activation. This allows the output of the network to be treated as a probability distribution across 4 classes, no fault, normal fault, reverse fault, and strike slip.

For the class activation map, we add a global average pooling layer after the convlutional blocks and use a dense layer with 2 outputs to predict fault/no fault. Then, for each input passed to the network the output was taken at the final convolutional layer and weighted using the dense layer's weights for the fault hidden unit to produce the activations across the image for faults.

For the loss function, we use the categorical crossentropy:

$$C = -\frac{1}{n} \sum_x [y \log a + (1 - y) \log (1 - a)] \tag{1}$$

where *y* is the expect output, *a* is the actual output by the neuron.

5 Results and discussion

5.1 Classification

Table 1 shows classification statistics for the trained network on two different test sets, one consisting of synthetic data and one consisting of real data. The training accuracy was found to be 95% and the validation accuracy was 88% with a validation set drawn from real data.

Table 1: Classification Statistics on Synthetic and Real Test Data

Synthetic/real set Classification Statistics				
Metrics	NoFault	Normal	Reverse	Strike Slip
Accuracy	1.0/0.935	0.972/0.890	0.911/0.0	0.997/0.0
Precision	1/0.93	0.92/0.87	0.97/0	1/0
Recall	1/0.93	0.97/0.89	0.91/0	0.1/0
F_1	1/0.93	0.95/0.88	0.94/0	1/0

Table 2: Prediction Vectors for Misclassified Normal Faults

Sample	No Fault	Normal	Reverse	Strike Slip
1	0.6937	0.2962	0.0101	6.11E-06
2	0.7201	0.2706	0.0093	1.17E-05
3	0.66	0.3256	0.0145	1.89E-05
4	0.0033	0.414	0.58	3.00E-03

From the table, synthetic set accuracy (f_1=0.95) is close to domain expert level. A lack of real reverse and strike slip data did not allow to test real data accuracy for these classes. Strike slip accuracy metric is characterized by low recall, i.e. by abundance of false negatives. Looking at the subset of misclassified examples in Table 2 along with their images in Figure 5, we can see that the first 3 images were misclassified as no fault. The corresponding images show very weak fault features, so this mistake is understandable.

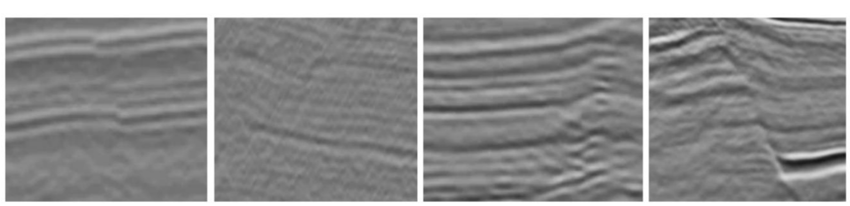

Figure 4: Misclassified samples, numbered 1-4 from left to right

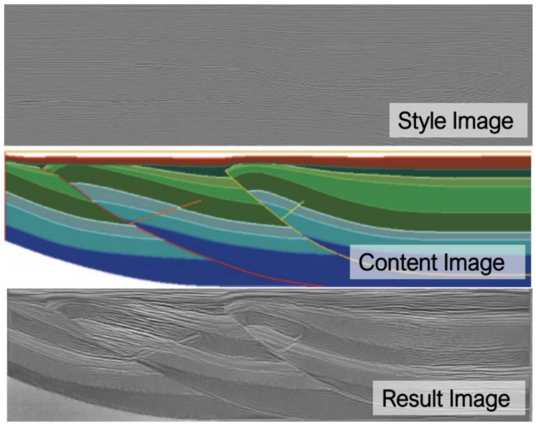

Figure 5: Neural style transfer with pre-trained VGG-19 network

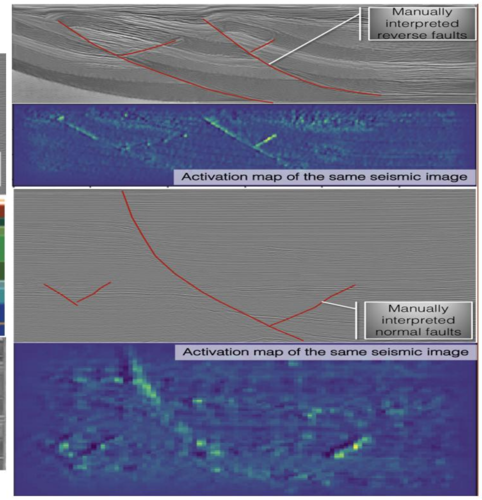

Figure 6: Full seismic images with activation maps

5.2 Style Transfer and implement the activation map

We use the VGG-19 neural network to train the style for faking reverse faulted seismic images(Figure 6). Neural Style Transfer can be utilized for faking reverse or strike slip seismic sections.

Additionally, as mentioned above the trained classification network was modified to form class activation maps on larger seismic images with multiple faults present. Using the modified network, a small window of 128x128 was passed over the larger image with a stride of 8 and the class activations were accumulated in a rectangular array. Finally, the accumulated activations were normalized to have a range between 0 and 1. The results for two different seismic images are shown in Figure 6. The activations highlight the manually identified faults as high values, indicating that the network would accurately predict them.

6 Future Work

In the future, we should gather more real reverse fault and strike slip data to validate model over those classes, since we had very little real data of those classes during this project. This would also allow us to inject more real data of these classes into the training dataset, which should improve its performance on real data.

Another approach we could take is to form an implementation on full 3D seismic sections. In this project we worked with 2D slices of 3D data, so adapting our network to take 3D data directly as an input could allow the network to exploit 3D locality in the seismic data and further improve its performance.

7 Contributions

Andrew simulated synthetic data, designed the prediction network & tuned hyperparameters, and built & generated the class activation maps for the larger seismic slices. Di tuned hyperparameters for the VGG-19 neural network for style transferring and wrote the final report. Anatoly provided domain expertise, got real datasets, did data labelling and prepared a final poster. Di and Anatoly designed initial CNN, which was significantly improved by Andrew during the course of the work.

8 Code

The GitHub link to our current repository is `https://github.com/AndrewD1996/fault-detection`.

References

[1] Araya-Polo et. al., 2017, Automated fault detection without seismic processing, The Leading Edge 36:3, 208-214

[2] Di, H., Z. Wang, and G. AlRegib, 2018, Seismic Fault Detection from Post-Stack Amplitude by Convolutional Neural Networks, in 80th EAGE Conference and Exhibition 2018: doi:10.3997/2214-4609.201800733

[3] Guitton, A., 2018, 3D Convolutional Neural Networks for Fault Interpretation, in 80th EAGE Conference and Exhibition 2018: doi:10.3997/2214-4609.201800732

[4] Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018, Convolutional neural networks for fault interpretation in seismic images: SEG, p. 1946-1950

[5] Wu, X., and Z. Zhu, 2017, Methods to enhance seismic faults and construct fault surfaces: Computers and Geosciences, 107,37–48.

[6] Xiong, W., X. Ji, Y. Ma, Y. Wang, N. M. BenHassan, M. N. Ali, and Y. Luo, 2018, Seismic fault detection with convolutional neural network: Geophysics, doi:10.1190/geo2017-0666.1

[7] https://github.com/dhale/ipf