The impact of the synthetic seismic data generation method on automated AI-based horizon interpretation

Vizeu, F., Zambrini, J.* and Canning, A.; AspenTech

Summary

This paper discusses the automatic interpretation of seismic data by using a CNN (Convolutional Neural Network) to predict RGT (Relative Geologic Time). RGT is used as a high-resolution geometric framework and is defined on a regular 3D grid, with the same dimensions as the seismic data. In this abstract, we focus on the role of training data in providing quality interpretation results. We describe a method for generating realistic synthetic 3D seismic data for training neural networks with great variability and show its performance for training a CNN.

Introduction

In recent years, the interest in applying machine learning (ML) technologies to seismic processing and interpretation is surging. Traditional methods and algorithms used for various seismic procedures are being effectively replaced by neural networks and other artificial intelligence (AI) methodologies. This transition offers such benefits as enhanced simplicity, efficiency and automation of many geoscience tasks. The application of deep neural networks covers a wide range within the exploratory geophysical scene, acting in salt classification, horizon interpretation, fault interpretation and more (Huang et al., 2017; Di et al., 2018; Guo et al., 2018; Wu and Zhang, 2018; Zhao and Mukhopadhyay, 2018; Wu et al., 2018, 2019; Bi et al., 2021 and Yang et al., 2023).

The conventional approach of formulating deterministic mathematical solutions for specific data problems is being replaced by a supervised learning approach. Instead of devising theoretical descriptions, the system is taught using numerous data examples to understand the nature of the problem and find solutions. The advantages of this approach are significant, as it allows the resolution of complex problems that are challenging to address theoretically. Moreover, it enables the combination of several steps into a single process and facilitates the automation of procedures that otherwise would require substantial human resources. Seismic interpretation, a task that demands considerable skilled labor, can be largely automated by ML technologies using this data training approach.

However, ML technologies face a major challenge related to the availability of data examples, particularly seismic data. The data sets are often large and unwieldy, and many are proprietary, limiting access for most researchers. Additionally, substantial amounts of data are required for many ML solutions, covering a wide range of geophysical problems, with ground truth information for each data set. Unfortunately, the true answers for seismic data are mostly unknown. To mitigate this data scarcity issue, synthetic data has become a common alternative in the geophysical community (Wu and Hale, 2016; Geng et al., 2019; Wu et al., 2019, 2020; Bi et al., 2021), as well as in other disciplines that apply ML technologies.

The use of synthetic data not only addresses data scarcity but also introduces physics into the solution. By generating synthetic data through simulation processes that mimic physical or mathematical ideas, the underlying physics of the problem is incorporated. As a result, the algorithm used to generate synthetic data imparts knowledge of physics to the AI system indirectly. Furthermore, using synthetic data with known ground truth removes the human bias often associated with interpretations created from real data.

Developing an automated high-resolution interpretation technology for a structural framework requires realistic synthetic seismic data for neural network training (Vizeu et al., 2021). In this work we discuss a method for generating such data, which has applications not only for seismic interpretation, but can also solve a wide range of seismic-related problems related to data processing, inversion and others. It is important to note that the synthetic data generation method described here relies on 1D convolution modeling and does not involve 3D wave equation simulations. This approach is suitable for postmigration challenges where seismic events and amplitude corresponds to the reflection coefficient. Seismic interpretation is only one example of a postmigration problem. The same rationale can be applied to generate synthetic pseudomigrated gathers using Zoeppritz equations for prestack amplitudes. These synthetic gathers can be applied to train an AI to address several local gather processing tasks.

Methodology

The workflow proposed is based on a sequence of 2D stochastic steps: select a real RGT section, simulate an acoustic impedance (IP) section using real well logs, build a subsurface model, perform augmentation, generate seismic data by using the convolutional model with full control of the synthetic wavelet, and add noise to it. To convert the 2D data into 3D we use a technique that relies on a moving window that travels in the 2D data and stacks it to the additional dimension.

The RGT is a property that carries the structural information on it and typically increases its value monotonically from zero to 1. These values are related to the geochronology of the subsurface rocks. This type of data is built from a geological model which is a time-consuming workflow. For this methodology, we have a database of real RGT volumes from several geological models. The first step is to draw an
RGT volume, select a 2D section from it and crop it in a random place. The majority of the distortion on the final data comes from this step, and since we can select thousands of examples it is possible to get a good variety of patterns and shapes providing huge variability to the synthetic data. The second step is the generation of acoustic impedance via stochastic Gaussian simulation. Using real well logs and plan-parallel horizons to act as top and base we build countless IP sections with great internal variability. This can be controlled by grid mesh size and variogram parameters. Since this property is generated with only horizontal features, it needs to be distorted to match the RGT geometry. This process is done trace by trace correlating a pair of 2D IP and RGT data (Figure 1). Once both data are paired, all the modifications done in one are automatically applied to the other. The augmentation step is made by a sequence of small internal deformations that can be: rotation and flip, stretch and squeeze, and the addition of undulations and faults. This is done to add more variability into the structural framework simulating different structural contexts such as canyons, compressive or distensive features, domes, etc. Generating the seismic section is just a matter of convolving the reflectivity obtained from the distorted impedance previously generated, and a modeled wavelet (Figure 2). This will determine the frequency content and the pattern of the seismic response. By controlling the parameters of the simulated wavelet, it is possible to emulate propagation effects such as attenuation and phase variations, enhancing the variability of the resulting synthetic data. The last step of the 2D phase is the addition of some noise to the synthetic seismic volume. Randomization of all parameters is one main principle of this methodology. This enables the generation of realistic seismic data with variability in seismic parameters such as frequency, noise, etc.

Finally, we developed a new method to randomly augment 2D data into 3D. The third axis is built section by section using a slow deformation and shifts from the previous section. These deformed sections become the third axis of the cube (Vizeu et al., 2022). Random rotations are applied to the 3D data to mitigate the directionality of this approach. Once both cubes of RGT and synthetic seismic are synced we have pairs of inputs and labels to train the CNN. We compared our synthetic data generation approach to the one proposed by Wu et al. (2020). We used for this comparison a trained network model made available by Bi et al. (2021) and used the architecture of the CNN that was also made available by Bi et al. (2020) in order to have a 1-1 comparison. We applied both trained networks to a real dataset and compared the resulting RGT.

Results

To test our approach we applied both trained network models to a real dataset and compared them with a “true” label. The true label dataset was composed of four horizons picked by an interpreter on a real dataset. We made sure to select the same vertical patch to avoid patching merging artifacts. For each horizon of the true label dataset, one contour of each predicted RGT was extracted for comparison. We selected the contours that exhibited minimum error from our base interpreted horizon. The comparison metric used was the average distance, in samples, between each interpreted horizon and the closest RGT contour obtained from the CNNs. We used 25 sub-volumes, uniformly placed throughout the dataset. Table 1 displays the results of the tests in terms of the global error and the error for each horizon, in samples. The global error is the mean of the errors in each horizon. According to Table 1, the training with our synthetic data performed better than the other approach, showing less than half of the global error. A review of all horizons shows that our approach...
outperformed the previous model, achieving the maximum error of 1.6 samples.

<table>
<thead>
<tr>
<th></th>
<th>Horizon 1</th>
<th>Horizon 2</th>
<th>Horizon 3</th>
<th>Horizon 4</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our model</td>
<td>1.042</td>
<td>0.754</td>
<td>0.842</td>
<td>1.628</td>
<td>1.066</td>
</tr>
<tr>
<td>Bi et al. (2021) model</td>
<td>1.781</td>
<td>1.616</td>
<td>2.391</td>
<td>3.576</td>
<td>2.341</td>
</tr>
</tbody>
</table>

Table 1: Quantitative results of the predictions using the two training approaches. The values are average distance errors in samples.

A more in-depth analysis of the second horizon is displayed in Figure 3, which presents both qualitative and quantitative results. The columns from left to right are: the True label set, the results from training with our data (magenta line) and the Bi et al. (2021) model result (cyan line). The two upper rows are inline and crossline sections, the third row is the depth map of the extracted surface for the event marker with an orange triangle (horizon 2), and the last row is the error associated with each result. Inspecting the results obtained by the CNN trained with our data, we noted a significant match between the RGT contours and the interpretation (dashed lines), following the trend of the seismic events. Bi’s model shows a good trend as well and a smoother characteristic, but it fails mainly in the crossline direction where it predicted a fault in a non-faulted zone. We can observe that some contours cross the event (horizon 3). Examining the topography maps extracted from the surfaces and the error associated with them demonstrates the impact a good training dataset can have on a CNN.

![Figure 3](image)

Figure 3: The results of both networks applied to real data. The columns are, from left to right, the True label set, Current network result (purple line) and Bi’s result (cyan line). The two upper lines are Inline and Crossline position, the third line is the extracted surface for the event marker with an orange triangle (horizon 2) and the last line is the error associated with each result. The error bar is the error in samples.

**Conclusions**

In this paper, we refer to Bi et al. (2021) which used synthetic data to train a CNN to predict RGT. We showed that that method to generate synthetic data impacts the results, and one potential solution to improve a CNN. Our method can produce an infinite number of seismic data examples. We build the synthetic data on the fly in parallel to the CNN training procedure, randomly selecting patches of data from the larger original set of RGT data. With this process, there is no need to store or manage training data. Another important advantage is the flexibility to add new types of geologic scenarios to the process without needing to start over. We continue to train the network with new examples and continuously see performance improvements. We compared the results of the network trained with our synthetic data, based on the architecture from Bi et al. (2020), to results showed by Bi, by applying both networks to a real dataset. The result showed the value of our synthetic data generation mechanism in a real application and the power of the AI approach to predict RGT volumes.