Automatic unflooding for salt base using U-net in full-waveform inversion framework

Introduction

Full-waveform inversion (FWI) aims to reconstruct the physical properties of the subsurface, such as the velocity and density [Tarantola, 1984]. The inversion process is normally based on the adjoint state method, which utilizes the single scattering assumption, also known as the Born approximation. To comply with the Born assumption, FWI needs to start with a good initial model. The absence of a good initial model and low frequencies in the data cause severe non-linearity in the problem, leading FWI to fall into a local minimum, an issue often referred to as the "cycle skipping problem."

In models containing salt bodies, the cycle skipping problem is more severe. The salt bodies include large structures with relatively much higher velocities than the surrounding sediments. Besides, it is often hard and cumbersome to build an initial model for FWI that contains salt information, and seismic data often lack the low frequencies and large offsets needed to construct the salt body. The traditional industry practice for building the salt body is a migration-picking-flooding workflow. In this approach, the top of the salt (TOS) is picked from the migrated image, and then a salt velocity is flooded from the TOS to the bottom of the model. After that, another migration is applied to pick the base of the salt (BOS) and unflood the velocity to the correct depth. The TOS is often interpreted correctly, unlike the BOS [Hu et al., 2015; Wu et al., 2018]. This is due to the high scattering energy at the TOS and the erroneous velocity at the base. Not to mention, migration often treats reflection corresponding to single scattering.

Identifying the salt boundaries has been a topic of research in velocity model building for years. With respect to the BOS, [Hu et al., 2015] suggested to test different interpretations of BOS using fast beam migration and choose the best corresponding model. [Wu et al., 2018] developed a semi-automatic picking to extract the salt boundaries. In all of these approaches, human intervention is still needed, especially for picking the BOS.

Recently, there have been a wide interest in utilizing machine learning (ML) algorithms in many geophysical applications such as seismic modeling, processing, interpretation and inversion (e.g. [Song et al., 2021; Alali et al., 2020; Zeng et al., 2019]. In salt body inversion and reconstruction, [Waldeland et al., 2018] used a convolutional neural network (CNN) to classify the salt in seismic images. [Lewis and Vigh, 2017] used CNN to generate prior salt information from seismic images and use them in velocity inversion, and [Zeng et al., 2019] identified the salt with good precision from seismic images using a U-net architecture based on CNN. Generally, most of the machine learning applications related to salt models are in the interpretation aspects where the final image is already obtained and very little within the velocity inversion process.

Here, we apply FWI to a flooded salt model and train a network to unflood the inverted result to the BOS. We use a U-net architecture for our neural network as it has been proven to be robust in salt detection. Our implementation is 1D, which allows us to generate an abundant amount of data and ease the training of the network. We train the network using 1D layered models with flooded salt layers and evaluate the performance of the trained network on the west part of the BP 2004 salt model. Additional applications of the network will be shared at the meeting.

Method

For salt inversion, conventional FWI is often combined with TV regularization [Kalita et al., 2019] to preserve the high-contrast feature of the salt edge. The objective function for FWI with TV regularization is,

\[ J(m) = \|d_{obs} - d_{syn}\|^2_2 + \lambda \|\nabla m\|, \]

(1)

where \(m\) is the model, \(d_{obs}\) and \(d_{syn}\) are the observed and synthetic data, respectively. \(\lambda\) is a regularization coefficient representing the trade off between the two norms. \(\|\nabla m\|\) is the TV regularization.
Inverting a salt-flooded model using equation \( \text{II} \) should capture some features of the BOS. Here, we use a network to detect the BOS and unflood the velocity. In mathematical terms, this can be formulated as,

\[
m_{\text{corrected}} = \theta(m_{\text{FWI}}),
\]

where \( \theta \) represents the neural network. The network is trained in a supervised manner by minimizing the mean square error (MSE) loss,

\[
\theta_{\text{loss}} = \frac{1}{N} \sum_{i=1}^{N} (m_{\text{corrected}} - m_{\text{target}})^2,
\]

where \( N \) is the batch size and \( m_{\text{target}} \) is the target unflooded model. Using MSE loss leads to a regression problem, which aims to capture the BOS and also approximates the subsalt velocity, unlike the often used cross-entropy loss for classification that will only be useful to capture the salt boundaries.

Figure [I] shows a summary of the workflow. We start with a flooded model and apply FWI to it. The network then takes in the result of FWI and attempts to unflood the salt. Finally, a fine-tuning of the velocity by another FWI is applied.

**Network Architecture and Training**

We adopt a U-net architecture for our salt correction neural network. U-net is commonly used in detecting the salt in previous studies as a classifier. In U-net, the input and the output should have the same shape, which suites our problem. Figure [II] shows the details of the network used in this work. It consists of 4 contracting and expanding blocks (encoder and decoder parts). In each contracting block, we have two convolutional layers and a rectified linear unit (Relu) activation function. Each convolutional layer is followed by a batch normalization operator. A max-pooling is implemented to shrink the size between the blocks. The decoder is similar to the encoder except that it contains transposed convolutional layers to increase the size of the input. In a typical U-net, the number of channels increases as we go deep in the encoder and decreases in the decoder by a factor of 2 between blocks. The number of channels used in this work are 16, 32, 64, 128 and 256 in the bottleneck. All the layers in the network are 1D unlike the typical U-net, which uses 2D convolutional layers. A Sigmoid activation function is applied to the last layer to produce the output. By choice, we design the network to take an input of length 200.

We need to generate a large number of data to train the network. The input for the network is the FWI inversion results, which are relatively expensive to obtain in 2D models. The main reason for choosing a 1D network is to reduce the cost and to be able to perform fast 1D FWI inversions. We use a random layers generator containing randomly 5 to a total of 50 layers. The velocity of the models starts from 1.5 km/s to mimic marine setup and increases with depth to up to 4.3 km/s. We then randomly pick some samples from the generated models to place a salt layer of 4.5 km/s velocity with random thicknesses. Some of the training models do not contain any salt, which teaches the network to distinguish between profiles containing salt and those free of salt. We generate a total of 10000 models. The FWI initial models are the true models except beneath the salt where we flood the model. We use a Ricker wavelet with an 8 Hz dominant frequency and 5 Hz minimum frequency to generate the training samples. The maximum offset of the generated data is 5 km. The relatively short offset and the missing frequencies below 5 Hz makes it very difficult for FWI to correct for the over-flooded salt. To enhance the uniqueness of the input to the network, we add the flooded model as an additional input to the network. The target outputs for the network are the true models. We use 80% of the generated models (data) for training and
20% for validation. Figure 3 shows six random samples from the training (first row) and the validation (second row) sets. We can see that the network manages to keep the validation samples without salt layers unchanged. For the salt models, the network correctly identifies the salt body and unfloods at the correct depth. Below the salt, we can see that the network approximates the true models’ velocity fairly well.

**Figure 2** The U-net Architecture.

**Figure 3** Samples from the training data set (first row) and the validation data set (second row).

**Numerical Example**

We test the trained network on the left part of the benchmark BP 2004 salt model (Figure 4(a)). We use a Ricker wavelet with a dominant frequency of 8 Hz and synthesize 200 shots on the surface placed 0.07 km apart. The data are recorded with a streamer of 5 km length and 0.02 km receiver spacing. The starting model is the true model after flooding the salt as shown in Figure 4(b).

The inversion result, shown in Figure 4(c), fails to correct for the subsalt velocity. It also slightly degrades the velocity in the right part where there is no salt, courtesy of applying TV regularization. We apply the trained network on the inversion result and obtain Figure 4(d). The network manages to unflood the salt to the correct depth and provide a good approximation to the subsalt velocity. At around 8 km, where the BP model has two salt layers, the network only corrects for the first layer. This is because we only include one salt layer in our training dataset. Due to having the initial flooded model as
Figure 4 a) is the true model, b) is the flooded model, c) is the FWI result starting with b, d) the unflooding by the network and e) is the FWI final tuning.

an input feature, the network also corrects for the "no salt" region in the right part. This is a useful feature when we trust the starting model and suspect that the initial FWI may degrade the results. Otherwise, we can remove the flooded model feature from the input, which will keep the "no salt" part of the model unchanged. Figure 4(e) shows the final result after fine-tuning the model by running another FWI. This result demonstrates the ability of the network to unflood the salt and to assist FWI in updating the subsalt velocity, even with limited offsets and missing low frequencies. Though we propose to train the network on 1D models for efficiency, the bandwidth and aperture of the seismic data in the training, including the source wavelet, should be consistent with that of the seismic data we plan to apply the network.

Conclusion

We proposed a method to intelligently unflood the salt in the FWI framework using a U-net network. We trained the network using inverted velocities of randomly generated 1D layered models that contain flooded salt layers with a velocity of 4.5 km/s. Then, we tested the network on the BP 2004 model. The results show the potential of the method to correctly unflood the salt in spite of the short offsets used and the missing low frequencies.

References


