Seismic Monitoring of Hydrocarbon Deposits Using a Viscoelastic Medium Model Based on Deep Learning

Denis Bratchikov and Kirill Gadylshin(✉)

Institute of Petroleum Geology and Geophysics SB RAS, 3 Koptug ave., Novosibirsk 630090, Russia
gadylshin@gmail.com

Abstract. This paper presents a novel approach to solving the inverse dynamic seismic problem in seismic monitoring for the viscoelastic medium. The proposed method offers a cost-effective alternative to Full Waveform Inversion by using a deep convolutional neural network Unet-type architecture with residual blocks to approximate an inverse problem operator that translates the change in seismic data into the change in velocity models. The operability of the proposed approach is demonstrated through a model example under the assumption that the distribution of the velocity model is known at the initial moment. Furthermore, the results of neural network prediction on a realistic sample with Gullfaks deposit indicate the practical applications of the proposed approach in seismic monitoring. The proposed approach shows significant potential for advancing the state-of-the-art in solving the inverse dynamic seismic problem for the viscoelastic medium, with potential implications for improving seismic monitoring techniques in industry and academia.

Keywords: Viscoelastic medium · full waveform inversion · deep learning · seismic monitoring

1 Introduction

Building a reliable model of the Earth is a prerequisite for the search for oil and gas deposits, mining, and other geophysical tasks. The solution of the inverse dynamical problem is the primary tool in seismic exploration to obtain a high-resolution model of the Earth's internal geological structure. One of the most critical recoverable parameters in the study of coal reservoir hydrocarbons is the velocity and the corresponding attenuation of seismic wave energy since it characterizes the rock's fluid saturation. Thus, knowledge of the spatial distribution of attenuation noticeably increases the reliability of the interpretation of the results of seismic observations

Recently, the Full waveform Inversion (FWI) [9] method has become a reliable and high-resolution method for constructing geological parameters with complex

The research was done under the support of RSF grant no. 22-21-00738.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14106, pp. 59–75, 2023.
https://doi.org/10.1007/978-3-031-37111-0_5

structures from the results of measuring wave fields on the surface. It was first proposed in a series of papers by Guy Chavent and Albert Tarantola [2,8] in the time domain based on the generalized least squares method. Frequency domain FWI [6] has been developed to speed up the inversion. However, FWI has several significant disadvantages:

- the ill-posedness of the inverse problem manifests itself in the non-uniqueness of the solution and the instability when using noisy data in the inversion.
- strong non-linearity of the objective functional, which leads to the convergence of the method to a local minimum.
- sensitivity of the solution to the presence of low-frequency information in seismic data.
- the presence of elastic parameters coupling when solving a multi-parameter inverse problem, which leads to the incorrect reconstruction of the parameters.
- high computational cost mainly manifested in the restoration of the elastic parameters of the medium in a three-dimensional formulation. Substantial computational resources are required to calculate the wave fields in solving the inverse problem at each iteration of the nonlinear least squares method.

Thus, an efficient and robust FWI technique is needed for optimal oil field development. Motivated by the remarkable achievements of neural networks as non-linear approximators in computer vision, deep learning presents itself as a promising methodology for FWI as an alternative to conventional modeling approaches. The learning-based strategy obviates the need for explicit problem modeling and offers the significant benefit of economizing on computational resources expended on seismic data processing. In the proposed method, the computational expenditures are primarily incurred during the initial stages, comprising the generation of a representative training dataset and training the neural network. At subsequent stages, the trained neural network replaces the processing, enabling it to forecast the petrophysical characteristics of the reservoir by leveraging seismic data at various developmental stages.

This research paper presents a novel approach for resolving a complex multi-parameter inverse dynamic seismic problem for a viscoelastic medium model. The proposed solution involves utilizing deep learning elements to reconstruct the model of velocities (V_p) and quality factors (Q_p) of longitudinal waves. Specifically, the method involves training a deep convolutional neural network (CNN) of the ResUnet architecture to approximate the nonlinear inverse problem operator F^{-1}. This operator is responsible for establishing the link between the computed waveforms and the velocity model of the medium. By leveraging the power of deep learning techniques, our approach demonstrates significant potential for tackling this challenging problem.

2 Theory and Method

2.1 Seismic Wave Propagation Modeling for a Viscoelastic Medium Model

Let us examine a two-dimensional model of a viscoelastic medium, wherein seismic waves manifest as mechanical vibrations of the rock resulting from an excitation source. The absorption of seismic energy, which is an inherent feature of all natural geological settings, is quantified by a quality factor:

$$Q^{-1} = \frac{1}{2\pi}\frac{\Delta E}{E}, \tag{1}$$

here, we denote E as the seismic energy per unit volume per unit time, and ΔE as its corresponding loss per unit volume per unit time. In this context, the propagation of seismic waves within a viscoelastic medium can be modeled by the following set of equations in the frequency domain:

$$\begin{cases} i\omega\rho\mathbf{u} = \mathrm{div}\sigma + \mathbf{f}, \\ i\omega\varepsilon = \dfrac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^*), \\ \sigma_{ij} = \left(\lambda\left(1 + \dfrac{i}{Q_p}\right) + 2i\mu\left(\dfrac{1}{Q_p} - \dfrac{1}{Q_s}\right)\right)\mathrm{tr}\,\varepsilon\delta_{ij} + 2\mu\left(1 + \dfrac{i}{Q_s}\right)\varepsilon_{ij}, \end{cases} \tag{2}$$

where the displacement velocity is represented by $\mathbf{u} = (u, v)^T$, while the strain and stress tensors are denoted by ε and σ, respectively. Additionally, ρ stands for density, λ and μ denote the Lamme coefficients, and Q_p and Q_s correspond to the quality factors of longitudinal and transverse waves, respectively. These quality factors are used to describe the absorption of longitudinal and transverse waves. Furthermore, ω signifies frequency, and the source of seismic waves is given by $\mathbf{f} = (f, g)^T$.

The first equation in the aforementioned set is responsible for capturing the dynamics of a continuous medium and expresses the law of conservation of momentum. The second equation introduces the linear Cauchy-Green strain tensor, which quantifies the relationship between the strain tensor and the corresponding displacement velocities. Finally, the third equation is Hooke's law for an isotropic medium, which specifies the relationship between the various components of the stress tensor and the corresponding strain tensor.

Having expressed the components of the stress tensor in terms of the displacement velocities and substituting them into the equation of motion, we obtain the relation for the displacement velocities:

$$\begin{cases} \omega^2\rho u + \dfrac{\partial}{\partial x}\left[\left(\lambda\left(1+\dfrac{i}{Q_p}\right)+2i\mu\left(\dfrac{1}{Q_p}-\dfrac{1}{Q_s}\right)\right)\left(\dfrac{\partial u}{\partial x}+\dfrac{\partial v}{\partial z}\right)+2\mu\left(1+\dfrac{i}{Q_s}\right)\dfrac{\partial u}{\partial x}\right]+ \\ \qquad +\dfrac{\partial}{\partial z}\left[\mu\left(1+\dfrac{i}{Q_s}\right)\left(\dfrac{\partial u}{\partial z}+\dfrac{\partial v}{\partial x}\right)\right]=f, \\ \omega^2\rho v + \dfrac{\partial}{\partial z}\left[\left(\lambda\left(1+\dfrac{i}{Q_p}\right)+2i\mu\left(\dfrac{1}{Q_p}-\dfrac{1}{Q_s}\right)\right)\left(\dfrac{\partial u}{\partial x}+\dfrac{\partial v}{\partial z}\right)+2\mu\left(1+\dfrac{i}{Q_s}\right)\dfrac{\partial v}{\partial z}\right]+ \\ \qquad +\dfrac{\partial}{\partial x}\left[\mu\left(1+\dfrac{i}{Q_s}\right)\left(\dfrac{\partial u}{\partial z}+\dfrac{\partial v}{\partial x}\right)\right]=g. \end{cases} \tag{3}$$

Absorbing boundary conditions of the PML (Perfectly Matched Layers) type [10] are used as boundary conditions. Then the partial derivative operators $\dfrac{\partial}{\partial x}$, $\dfrac{\partial}{\partial z}$ of the (3) system inside the absorbing layer are replaced by the operators $\dfrac{\partial}{\partial \tilde{x}}$, $\dfrac{\partial}{\partial \tilde{z}}$:

$$\begin{gathered} \frac{\partial}{\partial \tilde{x}} = \frac{1}{1+\frac{d_x(x)}{i\omega}}\frac{\partial}{\partial x}, \qquad \frac{\partial}{\partial \tilde{z}} = \frac{1}{1+\frac{d_z(z)}{i\omega}}\frac{\partial}{\partial z}, \\ d_x(x) = d_0\left(\frac{l(x)}{L}\right)^4, \qquad d_z(z) = d_0\left(\frac{l(z)}{L}\right)^4, \end{gathered} \tag{4}$$

where L is the width of the absorbing layer, $l(x), l(z)$ is the distance from a point in the absorbing layer to the inner boundary of the region.

To numerically solve the (3) system, we use a finite-difference approximation of partial derivatives, in conjunction with an unconditionally stable fourth-order implicit scheme. The grid for this scheme is depicted in Fig. (??). After discretizing the (3) system, we obtain a system of algebraic equations which we solve using LU decomposition. Furthermore, the Ricker wavelet serves as the source function for our calculations.

2.2 Inverse Seismic Dynamic Problem

We will now define an inverse dynamic seismic problem aimed at restoring the parameters of a viscoelastic medium model. To achieve this, we introduce the model of a viscoelastic medium, denoted by $m = (\rho, \lambda, \mu, Q_p, Q_s)^T$. Subsequently, we can define the direct problem operator $\mathcal{F}_s(m, \omega_j) : M \to U$. This operator associates the m models for a fixed time-frequency with the solution of the (3) system of equations. Here, the source takes the form of $\delta(\mathbf{x}-\mathbf{x_s})\mathbf{f}(\omega)$, with $\delta(\mathbf{x}-\mathbf{x_s})$ representing the Dirac delta function. It is noteworthy that the result of this operator actually corresponds to the product of the momentum shape $\mathbf{f}(\omega)$ and the Green's function $G(\mathbf{x_s}, \mathbf{x}, \omega, m)$ of the given system.

Let us introduce into consideration a linear operator for taking the trace of a full wave field from one of the sources at a fixed time frequency at points corresponding to the location of geophones:

$$P : U \to \mathbb{C}^{Nr}, \tag{5}$$

where Nr is the number of receivers. Then the wave field recorded by receivers on the surface for a fixed source s and a fixed frequency ω_j can be represented as a superposition of operators:

$$F_s^j = P \circ \mathcal{F}_s(m, \omega_j) : M \to \mathbb{C}^{Nr}. \tag{6}$$

If only Ns sources are used, then the data space for each of them at a fixed frequency ω_j is determined by the action of the following operator:

$$F^j(m) = \begin{pmatrix} F_1^j(m) \\ \vdots \\ F_{Ns}^j(m) \end{pmatrix} : M \to \mathbb{C}^{Nr \times Ns} \tag{7}$$

Finally, we obtain the following forward modeling operator from the model space to the data space:

$$F(m) = \begin{pmatrix} F^1(m) \\ \vdots \\ F^{Nf}(m) \end{pmatrix} : M \to D \tag{8}$$

where $D = \mathbb{C}^{Nr \times Ns \times Nf}$, Nf - number of time-frequencies used during the inversion.

Thus, the inverse problem is to determine the model of the medium from the wave field recorded in the coordinates of the receivers so that the nonlinear identity is satisfied

$$d_{obs} = F(m), \tag{9}$$

where $d_{obs} = (\dots \mathbf{u}_s(\mathbf{x}_r, \omega_j) \dots)^T \in D, r = 1 \dots Nr, s = 1 \dots Ns, j = 1 \dots Nf$, $\mathbf{x}_r$ – coordinates of the receivers, $\mathbf{u}_s$ – wave field for the source $\delta(\mathbf{x} - \mathbf{x_s})\mathbf{f}(\omega)$, $\mathbf{x}_s$ – source coordinate.

Let us introduce the misfit functional

$$E(m) = \frac{1}{2}\|d_{obs} - F(m)\|_2^2. \tag{10}$$

The standard FWI formulation is to minimize the (10) functional, i.e., finding the model $m^* \in M$

$$m^* = \underset{m}{\operatorname{argmin}}\, E(m). \tag{11}$$

The well-known formula for gradient computation looks as follows

$$\nabla E(m) = -\Re\left(\left(\frac{\delta F}{\delta m}(m)\right)^{*}(d_{obs} - F(m))\right). \tag{12}$$

Most existing methods for solving this problem rely on gradient methods such as conjugate gradients to find the minimum point of the functional. However, this approach demands a significant amount of computational resources as it requires solving the direct problem multiple times during each iteration of minimizing the functional.

2.3 Approximation of the Inverse Problem Operator by a Deep Convolutional Neural Network

Statement of the Problem. Let's examine an alternative approach to solving the inverse problem, which is data-driven. We suppose that we possess a sample that accurately depicts the correlation between the seismic data and the corresponding geological model of the environment. By minimizing the loss function, we can train on the given dataset to retrieve the F^{-1} inverse problem operator:

$$F_{*}^{-1} = \underset{F^{-1}\in U}{\operatorname{argmin}} L(m, F^{-1}(d)), \tag{13}$$

where L is the loss function expressing the discrepancy between the data and the action of the inverse problem operator on the model, U is the space of functions used to approximate the inverse problem operator.

This study will focus on utilizing deep learning methods for full waveform inversion. The inverse problem operator F^{-1} will be approximated using a deep convolutional neural network (CNN). It has been shown in [11] that a deep convolutional neural network can approximate any continuous function of several variables with any predetermined accuracy. However, it is important to note that the theorem guarantees the existence of such an architecture and weights for an artificial neural network. Still, it does not guarantee that the optimization learning process necessarily converges to this point. Thus, if the sample used for training is representative, then seismic data mapping to the environment model can be learned with fairly high accuracy. Since the operator of the direct problem is assumed to be differentiable, the continuity of the operator of the inverse problem follows automatically. After training, the reconstructed operator $\hat{F}^{-1}$ is applied to new seismic data d_{new} to predict the medium model m_{pred}:

$$m_{pred} = \hat{F}^{-1}(d_{new}). \tag{14}$$

The learning-based approach avoids direct modeling of the problem and the construction of the adjoint operator, thereby significantly reducing resource requirements during subsequent stages of applying the inverse operator $\hat{F}^{-1}$. However, the efficacy of this approach is contingent on the availability of a representative training sample and is highly sensitive to its quality.

In our work, we consider the transition to changes in models and seismic data. At t_1, the reservoir model m_1 and the corresponding seismic data d_1 are known to us. At a later time t_2, the velocity model of the reservoir changes slightly to m_2, and the corresponding seismic data changes to d_2. Instead of solving for m_2 given d_2 in the (9) setting, we aim to estimate the change in the model, denoted as $\Delta m = m_2 - m_1$, by the corresponding change in seismic data, denoted as $\Delta d = d_2 - d_1$. Therefore, the original learning task (13) will be replaced with the following task:

$$\mathcal{F}^{-1} = \underset{\mathcal{F}^{-1}}{\operatorname{argmin}} L(\Delta m, \mathcal{F}^{-1}[\Delta d]). \tag{15}$$

Such a transition makes achieving a better approximation of the inverse problem operator possible since it allows restoring more complex geological objects and structures in the environmental model.

Thus, the solution of the inverse dynamic seismic problem is replaced by the task of training a deep convolutional neural network, where the input parameters are the change in seismic data, and the output parameters are the change in the model of the environment.

DNN Architecture. We have chosen a ResUnet convolutional neural network architecture to translate data from seismograms into models. This architecture has been widely used in geophysical applications for various classes of problems, ranging from time-based data processing algorithms to inverse problems. The complete network architecture is shown in Fig. 1. The encoder-decoder structure consists of convolutional blocks that extract high-level features from the input seismic data and compress them into a lower-dimensional multidimensional tensor. The low-dimensional distribution of the data is learned, and then using a decoder consisting of convolutional blocks, the low-dimensional representation of the data is converted into output models. The architecture contains skip connections that pass the information received by the encoder to the decoder, which helps to deal with the problem of gradient decay during the training process. Residual connections (residual units) are also introduced to prevent local gradient fading in convolutional blocks.

Each (16) convolution block consists of a convolution operator Conv, a batch normalization BN [3], and an activation function ReLU [5].

$$\mathbf{x}^{l+1} = \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}\left(\mathbf{x}^{l}\right)\right)\right). \tag{16}$$

The convolution operation presented in equation (17) was initially developed for image processing, utilizing the weight distribution of convolutional kernels to reduce the number of parameters required during training. In our neural network, we utilize a 3×3 convolution kernel, a sliding step size of 1, and a padding value of 1 to preserve the feature map size after each convolution operation.

$$\mathrm{Conv}(\mathbf{x})_{(i,j)} = \sum_{m} \sum_{n} \sum_{c} k_{m,n,c} \cdot \mathbf{x}_{(s-1)\times i+m,(s-1)\times j+n,c}, \tag{17}$$

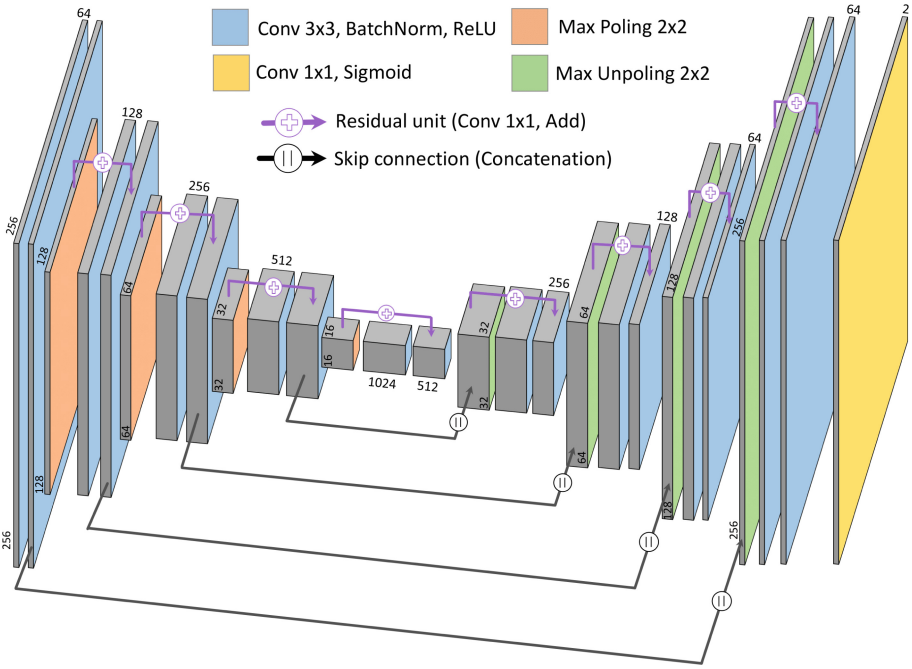

Fig. 1. Block diagram of the ResUnet deep convolutional neural network architecture. Each block represents output feature maps. The number above the block indicates the number of channels. The numbers on the side indicate the size of the feature map. Blue denotes a convolutional block consisting of a convolutional layer with a convolution kernel 3×3, batch normalization, and an activation function ReLU. Orange indicates the aggregation operator maxpooling with a window of 2×2 to reduce the dimension of the feature map. Green indicates the aggregation operator maxunpooling with a window of 2×2 to increase the dimension of the feature map. The purple arrows indicate the residual joins, consisting of a convolution with a kernel of size 1×1 and tensor addition. Black arrows represent skip connections implemented using tensor concatenation. The last layer of the neural network is highlighted in yellow, consisting of a convolution with a 1×1 kernel and an activation function Sigmoid. (Color figure online)

where the input data to the convolution block $\mathbf{x}$ and the trainable kernel of the convolution operator K are three-dimensional tensors with the first two dimensions i, j and m, n responsible for the spatial coordinate, and the third for the number of channels c, s – step with which the convolution kernel runs through the input data.

To regularize the learning process, a layer of batch normalization (2.3) is used, which brings the output of each layer to a distribution with mean β and variance γ, which are also learning parameters.

$$\mathrm{BN}_{\gamma,\beta}\left(\mathbf{x}_{i,j,c}\right) = \gamma\left(\frac{\mathbf{x}_{i,j,c} - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \varepsilon}}\right) + \beta,$$

where γ and β are two training parameters, $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ are the mean and variance, computed for each feature map $\mathbf{x}$ over data packets, ε is a negligible constant introduced for numerical stability of the method.

The activation function ReLU (2.3) is used to introduce non-linearity into the approximation of the inverse problem operator. The main advantage of this activation function is that the derivative does not vanish for positive values of the argument, which again contributes to the fight against gradient decay in the learning process.

$$\mathrm{ReLU}(\mathbf{x}) = \begin{cases} \mathbf{x}, & \text{if} \quad \mathbf{x} \geq 0 \\ 0, & \text{if} \quad \mathbf{x} < 0. \end{cases}$$

In the encoder of our neural network, we use the max pooling aggregation operator to reduce the dimension of the feature map by selecting the maximum values within a given window. Similarly, in the decoder, we use the max unpooling operator after the convolutional blocks to increase the dimension of the feature map by restoring the values in the window of a given size to the location of the maximum value. At the end of the network, we apply a convolution with a 1×1 kernel to adjust the output models to the required size. We use the Sigmoid activation function (18) to constrain the output model values to the interval $[0, 1]$.

$$\mathrm{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \tag{18}$$

DNN Training. Thus, using the convolutional neural network, the architecture of which is described above, we are looking for the inverse problem operator F^{-1} in the form of a mapping:

$$\begin{aligned} F^{-1} &\approx \hat{F} = f_n\left(\ldots f_2\left(f_1\left(d, w_1\right), w_2\right), \ldots, w_n\right), \\ x_{i+1} &= \begin{cases} f_1\left(d, w_1\right), & i = 1 \\ f_i\left(x_i, w_i\right), & i > 1 \end{cases}, \quad i = 1, \ldots, n \end{aligned} \tag{19}$$

The convolutional neural network in our study is made up of n convolutional blocks, with f_i representing one such block. The network takes as input seismic data d and uses network weights w_i, which are optimized to minimize the loss function. In order to train the network, we select a loss function L and an optimizer that minimizes the loss function (15). We choose to use the root mean square error (20) as our loss function, which is a commonly used metric for regression problems. Additionally, we employ the Adam optimizer [4], which is an adaptive learning rate optimization algorithm that combines momentum accumulation and gradient frequency maintenance principles.

$$L = \frac{1}{N} \sum_{i=1}^{N} \left(m_i^{true} - m_i^{pred} \right)^2, \tag{20}$$

where m_i^{true} is the true model, $m_i^{pred} = \hat{F}(d_i^{obs})$ is the predicted model, N is the size training dataset.

The training process involves updating the weights in (19) using the optimizer in each training epoch. Once the training of the convolutional neural network is complete, it can be used to predict the environmental model. Since the architecture mainly consists of the convolution operator and the activation function ReLU, the predictions can be made very quickly.

The coefficient of determination R^2 was employed to evaluate the training quality accuracy:

$$\begin{aligned} R^2 &= 1 - \frac{\sum_{i=1}^{N} \left(m_i^{true} - m_i^{pred} \right)^2}{\sum_{i=1}^{N} \left(m_i^{true} - \bar{m} \right)^2}, \\ \bar{m} &= \frac{1}{N} \sum_{i=1}^{N} \left(m_i^{true} - m_i^{pred} \right). \end{aligned} \tag{21}$$

Training Dataset. Creating a representative training dataset is an important task when training a CNN. For the purposes of this study, the CNN was trained on a synthetic training set of 2D data. It is important to note that during the creation of the training sample, it was assumed that only the velocity V_p and the quality factor Q_p of longitudinal waves underwent a change in the model. At the same time, the other parameters of the medium remained unchanged. As a result, only changes in the velocities and Q-factor of compressional waves were reconstructed from the corresponding changes in seismic data in this study.

To create a training sample, the Gullfaks [1] field, located in the North Sea, was chosen. The 2D elastic model of this field is shown in Fig. (2). It consists of 559×801 points with a sampling step of 5 m.

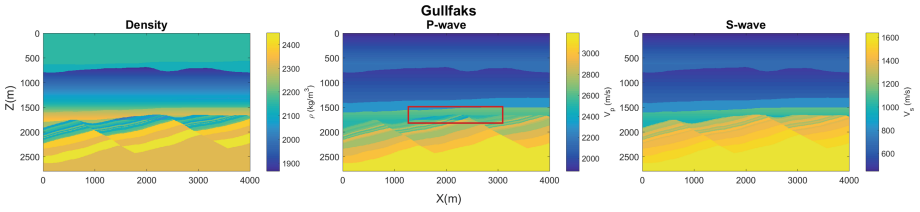

Fig. 2. The figure shows the elastic model of the Gullfaks deposit, with the density model ρ on the left, the velocity model of longitudinal waves V_p in the center, and the velocity model of shear waves V_s on the right. A red rectangle indicates the target area. (Color figure online)

Assuming a viscoelastic model of our medium, we introduced a constant quality factor of longitudinal and transverse waves with a value of 1000, which indicates that there is almost no absorption of both longitudinal and transverse waves. According to the information provided in [7], we have picked a target area containing wells that confirm the presence of oil in these areas. We filled this area with a value of 1500 m m/s for P-wave velocities and 10 for the Q-factor of P-waves, indicating fluid in the toe area. We cut out a part of the model of size $n_z \times n_x$ ($n_z = n_x = 256$) with a sampling step of 10 m so that the target area remains in the model. Thus, we got the reservoir's starting model, shown in Fig. (3).

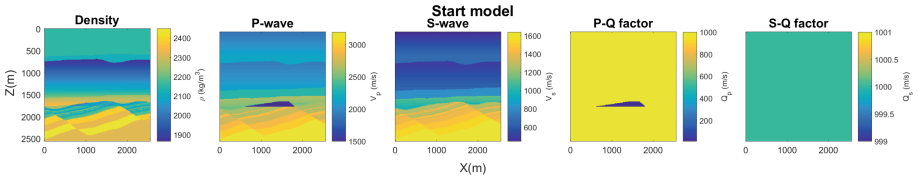

Fig. 3. The starting model of the reservoir used in the training sample. From left to right: there is a model of density ρ, longitudinal velocities V_p, transverse velocities V_s, the quality factor of longitudinal waves Q_p, the quality factor of transverse waves Q_s of the reservoir, respectively.

We assume that the shape of the reservoir changes randomly after the injection of a displacing agent. To achieve this, we vary the shape of the reservoir using a stochastic algorithm. Additionally, we set the velocity of longitudinal waves to 1500 m/s and the quality factor of longitudinal waves to 10, leaving all other parameters unchanged. This generates 1000 different true environment models whose changes we aim to restore. We obtain the changes by subtracting the true starting model and normalizing the model changes to the values of the interval [0,1]. Then, we combine the changes for the velocities and Q-factor of longitudinal waves into a common tensor of dimension $2 \times n_z \times n_x$ to serve as the neural network's output. A few examples of the generated true models and their corresponding changes are shown in Fig. (4).

After generating models, seismic data is modeled using finite difference formulas, as shown in Fig. (??). The approach described in the (??) paragraph is used, and a surface acquisition geometry is employed. At the level $z_s = 0$, $N_s = 256$ sources are evenly distributed with a step of 10 m, $x_s = 10 : 10 : 2550$ m. Similarly, at $z_r = 0$, $N_r = 256$ receivers are evenly distributed with a step of 10 m, $x_r = 10 : 10 : 2550$ m. T15 Hz Ricker wavelet is used as the source, and $N_f = 9$ frequencies with a step 2 Hz, $\omega = 4 : 2 : 20$ Hz are used for modeling seismic data. To accelerate the calculation time for creating a training dataset, the solver is parallelized by frequencies using MPI technology. Only the vertical component of the displacement velocity v is used to register a seismic signal. The data acquisition scheme is shown in Fig. (5). Seismic data is first modeled for a

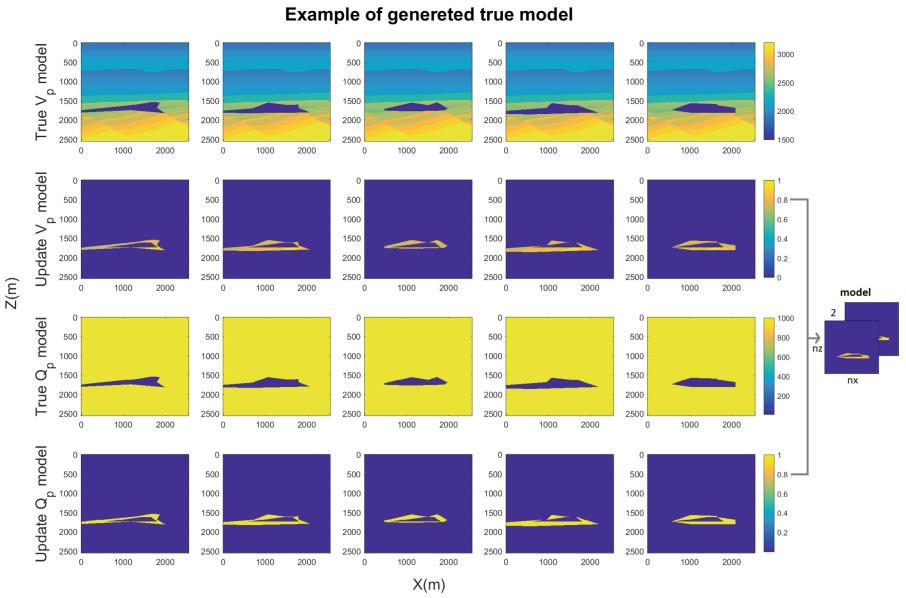

Fig. 4. Examples of generated true reservoir models and their corresponding changes that form the training sample. The first line contains examples of models of longitudinal velocities v_p, and the second line contains the corresponding normalized changes in longitudinal velocities Δv_p. The third line contains examples of models of the quality factor of longitudinal waves Q_p. The fourth line contains the corresponding normalized changes in the quality factor of longitudinal waves ΔQ_p. The arrows show how the change tensor of the $2 \times n_z \times n_x$ ($n_z = n_x = 256$) model is formed, which serves as the output of the neural network.

specific frequency ω_j, and the real and imaginary parts are extracted from them. The data is then converted into $N_s \times N_r$ feature maps for the real and imaginary parts. By subtracting the data for the starting model from the data for the true model, the data corresponding to the change in the model is obtained. The data change is normalized to the values of the interval [0,1] and combined over all frequencies into a tensor of dimension $2N_f \times N_s \times N_r$. This tensor serves as the input of the neural network for training.

In the resulting data set, 70% of the sample is used for the training part, on which we will train the neural network, and 30% for the validation part to ensure that the neural network does not overfit. The division is done so that the distributions of the training and test sets are similar. Separately, several examples are created for the test part, on which the quality of training will be assessed.

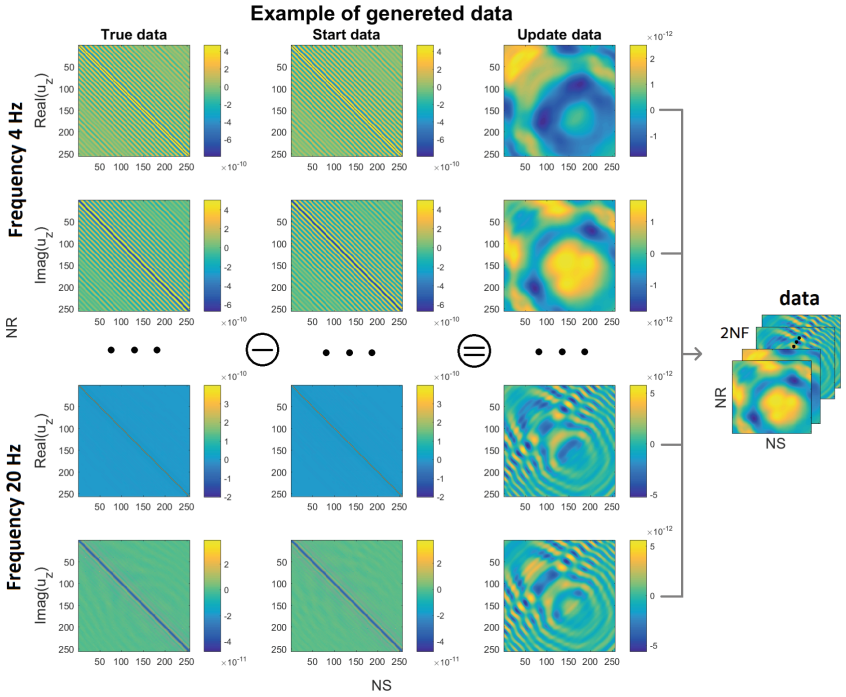

Fig. 5. Examples of simulated synthetic data forming a training set. The data is a feature map of size $N_s \times N_r$, which contains the values of the vertical component of the displacement velocity u_z, recorded by the receiver r on the Earth's surface for the excitation source s. The first column contains the data simulated for the true models, the second column contains the data for the starting model, and the third column contains the data change. The first and second lines contain the real and imaginary parts of the vertical component of the displacement velocity u_z, modeled at a frequency 4 Hz. In the next two lines, similarly for a frequency 6 Hz, and so on up to a frequency 20 Hz. The arrows show how the $2N_f \times N_s \times N_r$ ($N_f = 9, N_s = N_r = 256$) data change tensor is formed, which serves as the input of the neural network.

3 Numerical Experiments

A convolutional neural network (CNN) is typically trained in two stages: the first stage is training, during which the loss function is minimized; the second stage is validation, where we monitor the generalization error. In this case, the ResUnet was trained for 100 epochs, using an initial learning rate of 0.001 and a data batch size of 16. Figure 6 displays the variation in the loss function (20) and the metric (21) on the training and validation sets during the learning process. As the number of epochs increases, we observe a decrease in the loss function on both parts of the data. However, around epoch 60, the learning process no longer reduces the generalization error. Consequently, to assess the quality of training, we fixed a model with weights at epoch 60 on the test set (highlighted by a

dot on the graph). At epoch 60, the loss function for the training and validation parts reached $3 \cdot 10^{-4}$ and 1.72^{-3}, respectively. The prediction accuracy at epoch 60 was 94% for the training set and 89% for the validation set.

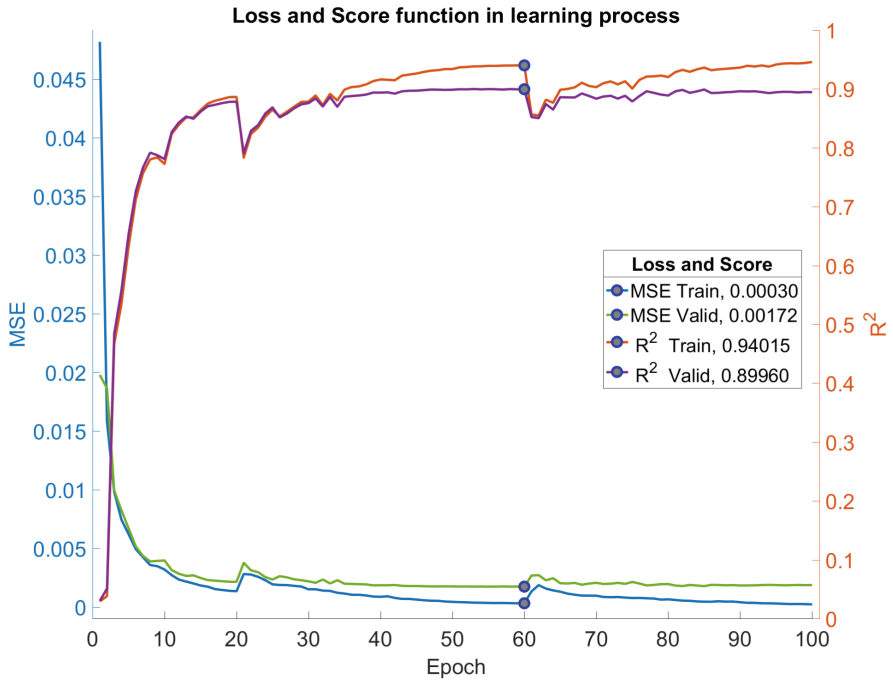

Fig. 6. Changing the MSE loss function and the R^2 metric during neural network training. The left vertical axis corresponds to the MSE values, and the blue and green lines show the change in MSE during training for the training and validation sets, respectively. The right vertical axis corresponds to the values of R^2, and the orange and purple lines show the change in R^2 during training for the training and validation sets, respectively. The point on the graph indicates the moment when the loss function on the validation part of the sample reached its minimum value. The plot legend contains the values of the functions corresponding to the point. (Color figure online)

After training, the neural network was applied to the test dataset. The results of predicting changes in the models of velocities and quality factors of longitudinal waves are shown in Fig. (7 and 8). As one can see, the convolutional neural network generally copes with restoring the environment model from seismic data.

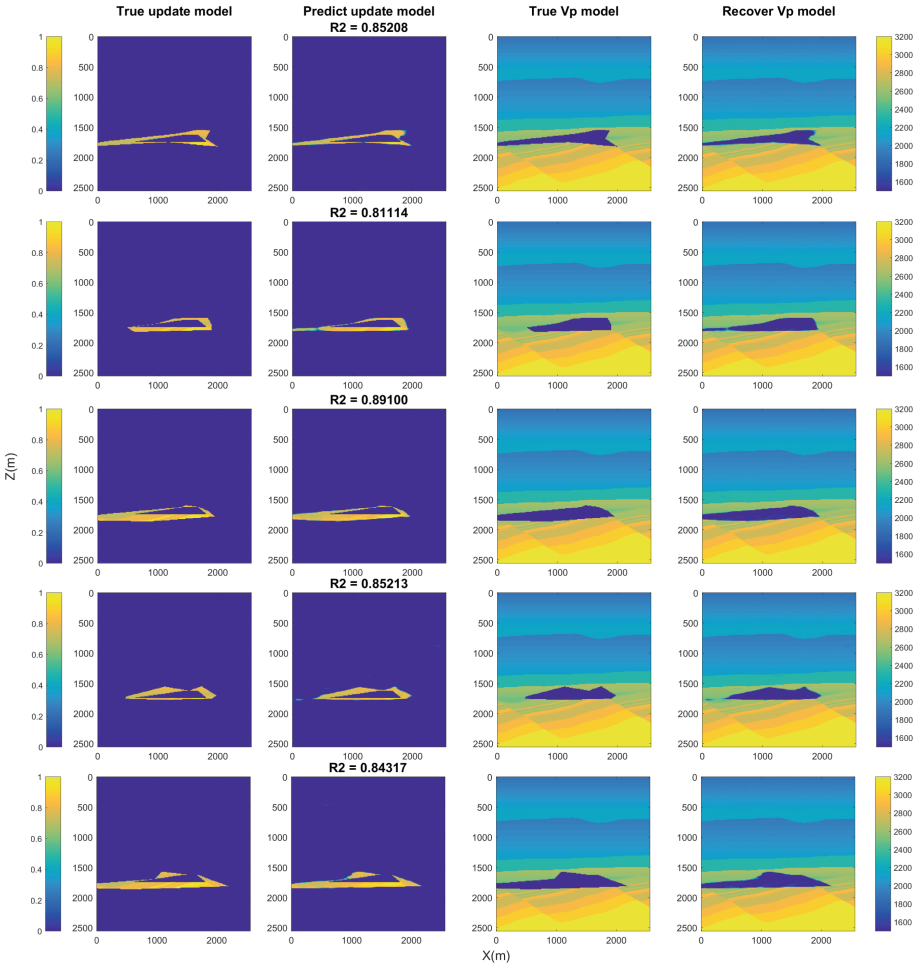

Fig. 7. Examples of predicting changes in the velocity of longitudinal waves by a neural network on a test set. The first column contains the true change in the velocity model. The second column contains the neural network's prediction of the speed model. Above the prediction is the value of the R^2 metric. The third column contains the true velocity model. In the fourth column the reconstructed velocity model for predicting velocity changes.

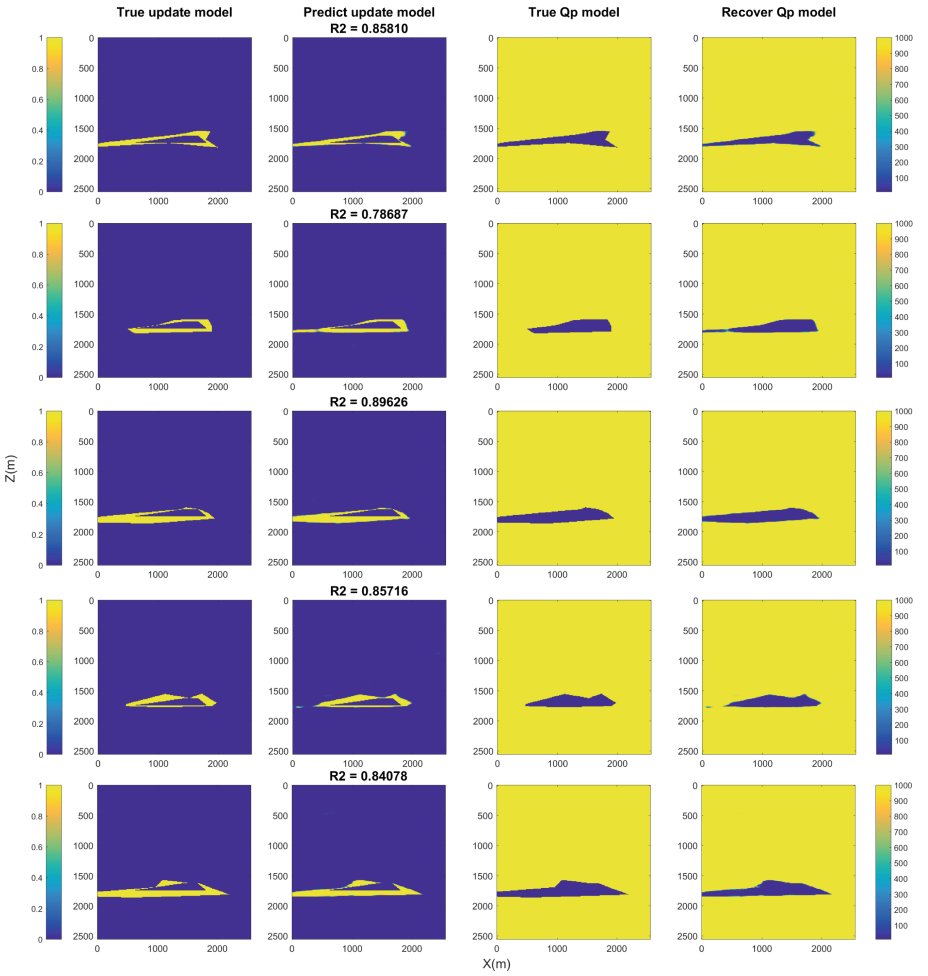

Fig. 8. Examples of predicting changes in the quality factor of longitudinal waves by a neural network on a test sample. The first column contains the true change in the quality factor model. The second column contains the prediction of the quality factor model by the neural network; above the prediction, the value of the R^2 metric is indicated. The third column contains the true quality factor model. In the fourth column, the reconstructed Q-factor model is presented.

4 Conclusions

The paper presents an innovative approach to addressing the inverse dynamic seismic problem in seismic monitoring for the viscoelastic medium. The proposed method offers a cost-effective alternative to Full Waveform Inversion by utilizing a deep convolutional neural network Unet-type architecture with the residual blocks to approximate an inverse problem operator that translates the change in

seismic data into the change in velocity models. The model example presented in this paper demonstrates the operability of this approach for solving the inverse seismic problem under the assumption that the distribution of the velocity model is known at the initial moment. The results of neural network prediction on a realistic sample with Gullfaks deposit indicate that the proposed approach has practical applications in seismic monitoring. Overall, the proposed approach holds significant promise for advancing the state-of-the-art in solving the inverse dynamic seismic problem for the viscoelastic medium, with potential implications for improving seismic monitoring techniques in industry and academia.

References

1. Structural core analysis from the gullfaks area: northern north sea. Mar. Pet. Geol. **18**(3), 411–439 (2001)
2. Bamberger, A., Chavent, G., Hemon, C., Lailly, P.: Inversion of normal incidence seismograms. Geophysics **47**(5), 757–770 (1982). https://doi.org/10.1190/1.1441345
3. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167 (2015)
4. Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2014)
5. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th International Conference on Machine Learning (2013)
6. Pratt, R.G., Shin, C., Hick, G.J.: Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. Geophys. J. Int. **133**(2), 341–362 (1998). https://doi.org/10.1046/j.1365-246X.1998.00498.x
7. Stovas, A., Landrø, M., Arntsen, B.: A sensitivity study based on 2D synthetic data from the Gullfaks field, using PP and PS time-lapse stacks for fluid-pressure discrimination. J. Geophys. Eng. **3**(4), 314–328 (2006)
8. Tarantola, A.: Inversion of seismic reflection data in the acoustic approximation. Geophysics **49**(8), 1259–1266 (1984). https://doi.org/10.1190/1.1441754
9. Virieux, J., Operto, S.: An overview of full-waveform inversion in exploration geophysics. Geophysics **74**(6), WCC1–WCC26 (2009). https://doi.org/10.1190/1.3238367
10. Yao, G., da Silva, N.V., Wu, D.: An effective absorbing layer for the boundary condition in acoustic seismic wave simulation. J. Geophys. Eng. **15**(2), 495–511 (2018)
11. Zhou, D.: Universality of deep convolutional neural networks. CoRR abs/1805.10769 (2018)