Multilayer perceptron and Bayesian neural network based implicit elastic full waveform inversion

Tianze Zhang, Jian Sun,†, Daniel O. Trad*, Kristopher A. Innanen,**

**University of Calgary †Ocean University of China*

SUMMARY

We introduce and analyze implicit full waveform inversion (IFWI), which uses a neural network to generate velocity models and perform full waveform inversion. IFWI carries out inversion by linking two main networks: a neural network that generates velocity models, and a recurrent neural network to perform the modeling. The approach is distinct from conventional waveform inversion in two key ways. First, it reduces reliance on accurate initial models, relative to conventional FWI. Instead, it invokes general information about the target area, for instance estimates of means and standard deviations of medium properties in the target area, or alternatively well-log information in the target area. Second, iterative updating affects the weights in the neural network, rather than the velocity model directly. Velocity models can be generated in the first part of the IFWI process in either of two ways: through use of a multilayer perceptron (MLP) network, or a Bayesian neural network (BNN). Numerical testing is suggestive that the MLP based IFWI approach in principle build accurate models in the absence of an explicit initial model, and the BNN based IFWI could give the uncertainty analysis for the prediction results.

INTRODUCTION

Full waveform inversion (FWI) is an optimization based inverse procedure with the capacity to produce high-resolution velocity models and subsurface images (Virieux and Operto, 2009). Practical FWI faces several key challenges. On land, adequate near-surface models typically require elastic or viscoelastic propagation models to be considered, and within these, accurately-determined shallow heterogeneities (Teodor et al., 2021). However in multi-parameter inversion, different sensitivities across parameter classes cause crosstalk in reconstructed models (e.g., Keating and Innanen, 2020, 2019). Even in single-parameter problems challenges arise. The commonly-used ℓ_2 norm objective function typically contains many local minima, requiring either very accurate initial models or high-fidelity broadband data (Lailly and Bednar, 1983; Tarantola, 1984; Tromp et al., 2005; Plessix, 2006; Virieux and Operto, 2009); without sufficiently accurate initial models, converge to the right solution is unlikely. Multiparameter elastic FWI is even more vulnerable to the problem of local minima, with different wave modes in the data and larger numbers of elastic parameter classes involved. Several strategies have been introduced for the selection of initial models. First-arrival traveltime tomography is often used to retrieve smooth P-wave velocity (Teodor et al., 2021); surface-wave (SW) analysis and inversion techniques can be adopted to retrieve the S-wave velocity (V_s) variations, using processing workflows based on windowing and wavefield transform to extract and invert local dispersion curves (DCs). Ren et al. (2021) developed a joint diving/direct and reflected wave method to build P- and S-wave velocity macromodels.

Neural networks are rapidly developing as avenues to support and formulate FWI algorithms; here we develop a network focused specifically on addressing the interrelated challenges of EFWI updating and initial model selection. We consider a procedure linking a model selection network and a simulation network. We examine two promising types of selection network. The first is the coordinate-based multilayer perceptron (MLP). Coordinate-based MLPs take low-dimensional coordinates as input (usually in $\mathbb{R}^2$ or $\mathbb{R}^3$) and are trained to output a representation of some generalized target (Tancik et al., 2020). These networks (e.g., Rahaman et al., 2019; Stanley, 2007; Genova et al., 2020) have been developed and applied in image representation problems, shape representation in texture synthesis, shape inference from images, and novel view synthesis, and in these applications have achieved state-of-the-art results (Rahaman et al., 2019; Stanley, 2007; Genova et al., 2020; Park et al., 2019; Liu et al., 2019; Sitzmann et al., 2019). Analysis of such networks is suggestive that they tend to have a very strong capacity to resolve low frequencies in the loss function; whereas difficulties are often reported in their ability to resolve high-frequency information (Basri et al., 2020; Rahaman et al., 2019). The second type is the Bayesian neural network (BNN), first introduced by MacKay (1992). The BNN generates as output uncertainty estimates for the neural network; it is characterized by weights whose values follow a probability density function (pdf) over the weight space. The pdf is initialized and then updated (via Bayes' theorem) as training progresses; the network when specified includes best choices for weights, and an output distribution gauging their uncertainty.

With these ingredients, we formulate what we call implicit full waveform inversion (IFWI). This involves combining one of either a coordinate based MLP network, equipped with sinusoidal activation functions, or a Bayesian neural network (BNN), with a theory guided recurrent neural network (RNN) to generate seismic records. The IFWI proceeds in the manner of a waveform inversion, but does not require an explicit initial model to begin. The network takes, as input, the spatial positions of the grid cells associated with the model we wish to reconstruct. The network (either MLP or BNN) generates velocity models with expected properties. These velocity model realizations are then sent to the RNN and through it seismic records are simulated. The residual between the observed and simulated data are returned, and updating occurs. The final output of the IFWI procedure is a set of network weights (as opposed to model parameters), which when used in the final network generates a suite of elastic models which honour the data, as opposed to a single, assumed optimal, model. It therefore has within it a capacity for uncertainty estimation, and does not rely on a single initial model. We numerically analyze it to determine whether it exhibits stability in the absence of a well-resolved initial model.

THEORY

In the FWI problem we seek to minimize the following equation, normally using an l_2-norm misfit function

$$\phi(\mathbf{m}) = \sum_{\mathbf{x_r}} \int_0^T ||\mathbf{d}_{\text{syn}}(\mathbf{x_r}, t, \mathbf{m}) - \mathbf{d}_{\text{obs}}(\mathbf{x_r}, t)||^2 dt, \quad (1)$$

where $\mathbf{d}_{\text{obs}}$ represents the observed data, $\mathbf{d}_{\text{syn}}$ represents the synthetic data (obtained through $\mathbf{d}_{\text{syn}} = \mathbf{Ru}$), $\mathbf{R}$ is the receiver operator, $\mathbf{x_r}$ represents the receiver location vector, T is the maximum receiving time, and $\mathbf{m}$ is the model vector. The field $\mathbf{u}$ is obtained with $F(\mathbf{u}, \mathbf{m}) = \mathbf{f}$, where F is the forward modeling operator and $\mathbf{f}$ is the source term. In conventional FWI we calculate the gradient, $\frac{\partial \phi}{\partial \mathbf{m}}$, via the adjoint state method, and update the model unknowns $\mathbf{m}$ directly, through, e.g., a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) or conjugate gradient optimization method. This approach faces several well-known issues. First, because FWI is a highly nonlinear optimization problem, it depends on an accurate initial model, without which it will tend to converge to one of many possible local minima. Second, the quantification of the uncertainties in FWI is difficult and remains an open question.

With these issues in mind, we formulate the multilayer perceptron (MLP) and Bayesian neural network (BNN) based IFWI. In the MLP based IFWI we did not use an explicit initial model as a start point of the inversion, we use only the well log information to perform the inversion. In the BNN based IFWI, we treat the weights in the neural network as trainable probability distributions (pdf). If the network is well trained, it could be used to give the uncertainly analysis for the inversion.

MLP based Implicit elastic FWI

IFWI can be implemented by generating test elastic models with either an MLP or a BNN; here we will formulate the problem with both in turn, allowing them to be compared in numerical tests. We start by formulating the full IFWI algorithm with an MLP and then form a second algorithm by switching out the MLP for the BNN.

IFWI involves solving for network weights, as training progresses, to more and more effectively produce realizations of elastic models. For the IFWI problem, the effective objective function is:

$$\phi(\mathbf{w})_{\text{IFWI}} = \sum_{\mathbf{x_r}} \int_0^T ||\hat{\mathbf{d}}_{\text{syn}}(\mathbf{x_r}, t, N_{\text{mlp}}(\mathbf{c}; \mathbf{w})) - \mathbf{d}_{\text{obs}}(\mathbf{x_r}, t)||^2 dt, \quad (2)$$

where $N_{\text{mlp}}(\mathbf{c}; \mathbf{w})$ is the MLP network which generates elastic models. It in turn is formulated as

$$N_{\text{mlp}}(\mathbf{c}; \mathbf{w}) = (h_1 \circ h_2 \circ \cdots \circ h_n)\text{m}_{\text{std}} + \text{m}_{\text{mean}}, \quad (3)$$

where $\circ$ represents the fact that the hidden layers of the neural network are linked in sequence, h_l stands for the l^{th} hidden layer (where $l \leq n$), n is the maximum number of network layers, $\mathbf{c}$ is the coordinate vector, defining the size of the model to be generated, $\mathbf{w}$ are the trainable weights of the neural network, and m_{std} and m_{mean} are the standard derivation and the mean values of the elastic models obtained from prior information (e.g., a single or several well logs). These two values are used to scale the output of the MLP making sure that elastic property values in the output are within the reasonable range for the exploration area. The calculation in each hidden layer is expressed as:

$$h_l(\mathbf{o}_l) = \sin(\mathbf{w}_l \mathbf{o}_l + \mathbf{b}_l), \quad (4)$$

$\mathbf{o}_l$, $\mathbf{w}_l$ and $\mathbf{b}_l$ are the input, weight and bias vectors in the l^{th} layer. In the MLP based IFWI we use the *sin* activation function. We can see that in IFWI, the elastic model is parameterized with the weights and bias in the neural network. The detailed data dimension in the network has been illustrated in Table 1.

We train the network as follows. First, we initialize the weights as fixed values in the MLP. Second, we use the MLP network to predict the velocity models, which are V_p,V_s, and ρ. In this study, we use the velocity parameterization. Third, we solve $F(\hat{\mathbf{u}}, N_{\text{mlp}}(\mathbf{c}; \mathbf{w})) = \mathbf{f}$ for $\hat{\mathbf{u}}$, and the synthetic data is obtained with $\hat{\mathbf{d}}_{\text{syn}} = \mathbf{R}\hat{\mathbf{u}}$. In this study, we solve $\hat{\mathbf{u}}$ with a recurrent neural network (RNN) (Sun et al., 2020; Zhang et al., 2020). Fourth, we calculate the residual with equation (2), and then calculate the gradients by using the automatic differential method. We update the weights $\mathbf{w}$ in $N_{\text{mlp}}(\mathbf{c}; \mathbf{w})$ with the gradient based optimization method: $\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \frac{\partial \phi(\mathbf{w})_{\text{IFWI}}}{\partial \mathbf{w}_{t-1}}$, where t is the iteration time and α is the step length. If biases are included in the calculation, then the bias vector should be updated in the same way as the weights. The general work flow of the IFWI has also been illustrated in Figure 1. This training process distinguishes it from conventional FWI, in which the elastic model parameters are solved for explicitly, and are based on updates of an explicitly selected initial model.

Table 1: **The MLP network used in IFWI**

Layer name	Description	dimension
Input	Coordinate in x and z direction	$\mathbb{R}^{2 \times n_z \times n_x}$
Permute	Dimension permute	$\mathbb{R}^{n_z \times n_x \times 2}$
layer 1	FC layer	$\mathbb{R}^{n_z \times n_x \times 256}$
activation 1	sin	$\mathbb{R}^{n_z \times n_x \times 256}$
layer 2	FC layer	$\mathbb{R}^{n_z \times n_x \times 256}$
activation	sin	$\mathbb{R}^{n_z \times n_x \times 256}$
layer 3	FC layer	$\mathbb{R}^{n_z \times n_x \times 3}$
Permute	Dimension permute	$\mathbb{R}^{3 \times n_z \times n_x}$
Output	Scale with well log information	$\mathbb{R}^{3 \times n_z \times n_x}$

BNN based Implicit elastic FWI

In this section, we replace the MLP network with a probabilistic interpretation of neural network learning by implementing a Bayesian neural network (BNN), denoted as $N_{\text{bnn}}(\mathbf{c}; \theta)$. The difference between the BNN and the MLP is that the weights in the BNN are regarded as probability distribution functions (pdfs) rather than fixed values.

In the BNN, we seek to estimate the posterior distribution $p(\mathbf{w}|\mathbf{D})$, referred as the probability of $\mathbf{w}$ when data vector $\mathbf{D}$ is observed.

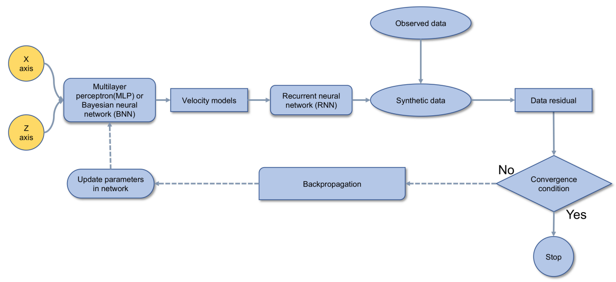

Figure 1: IFWI work flow.

This is approximated by a learned distribution weights $p(\mathbf{w}|\theta)$, with learnable parameter θ. This process is usually enforced by minimizing a Kullback-Leibler divergence (KL):

$$\begin{aligned} &\arg\min_{\theta} \mathrm{KL}[p(\mathbf{w}|\theta)||p(\mathbf{w}|\mathbf{D})] \\ &=\arg\min_{\theta} \mathbb{E}_{p(\mathbf{w}|\theta)}\left[\log\frac{p(\mathbf{w}|\theta)}{p(\mathbf{w}|\mathbf{D})}\right] \\ &=\arg\min_{\theta} \mathbb{E}_{p(\mathbf{w}|\theta)}\left[\log\frac{p(\mathbf{w}|\theta)p(\mathbf{D})}{p(\mathbf{D}|\mathbf{w})p(\mathbf{w})}\right] \\ &=\arg\min_{\theta} \mathbb{E}_{p(\mathbf{w}|\theta)}\left[\log\frac{p(\mathbf{w}|\theta)}{p(\mathbf{D}|\mathbf{w})p(\mathbf{w})}\right], \end{aligned} \tag{5}$$

using Bayes' theorem. Minimizing the above loss is equivalent to minimizing

$$\phi(\theta) = \log p(\mathbf{w}|\theta) - \log p(\mathbf{w}) - \log p(\mathbf{D}|\mathbf{w}). \tag{6}$$

For simplicity in this study, we assume each element of the weight vector to follow a Gaussian distribution, and be characterized with a mean value and the standard deviation. During the forward calculation in the neural network, realizations of the weights are drawn randomly from their current probability distribution. For example, assume that w_i is the i^{th} value in vector $\mathbf{w}$, then it can be realized by using the following equation:

$$w_i = w_{\mu_i} + \log(1+e^{\rho_{w_i}})w_{\varepsilon_i}, \tag{7}$$

where w_{μ_i} is the mean value of w_i, and w_{ε_i} is a random value generated with the Gaussian distribution, i.e., $w_{\varepsilon_i} \sim N(0,1)$. w_{μ_i} is the value which determines the mean of the probability distribution for parameter w_i. ρ_{w_i} influences the standard deviation for the parameter w_i. Each value in the $\mathbf{w}$ is obtained through the above process. We regard the w_{μ_i} and ρ_{w_i}as the training parameters. For simplicity, the training parameters are all noted as θ. After the weights are realized, the output of the BNN is obtained with equation (3).

The BNN formulation involves the following training process. First, we obtain the weights as described above. Second. we use the weights to calculate the outputs of the network, which are the elastic property models V_p,V_s, and ρ. Third, use the same recurrent neural network (RNN) as discussed previously to carry out forward modeling via $F(\hat{\mathbf{u}}, N_{\text{bnn}}(\mathbf{c};\theta)) = \mathbf{f}$ and obtain the synthetic data $\hat{\mathbf{d}}_{\text{syn}}$. Fourth, we calculate the $\log p(\mathbf{w}|\theta)$, $\log p(\mathbf{w})$, $\log p(\mathbf{d}_{\text{obs}}|\mathbf{w})$, for the objective function (5). For FWI problem, using $\log p(\mathbf{d}_{\text{obs}}|\hat{\mathbf{d}}_{\text{syn}})$ to replace $\mathrm{logp}(\mathbf{d}_{\text{obs}}|\mathbf{w})$. Thus, the loss function we use for BNN IFWI is defined as:

$$\phi(\theta)_{\text{IFWI}} = (\log p(\mathbf{w}|\theta) - \log p(\mathbf{w}))\,\lambda_c - \log p(\mathbf{d}_{\text{obs}}|\hat{\mathbf{d}}_{\text{syn}}). \tag{8}$$

Fifth, we use a gradient based method to update the weights: $\theta_t = \theta_{t-1} - \alpha\frac{\partial\phi(\theta)_{\text{IFWI}}}{\partial\theta_{t-1}}$, where the t is the iteration time, and $\frac{\partial\phi(\theta)}{\partial\theta}$ is the partial derivative of the loss with respect to the parameter θ. λ_c is the value that controls the contribution of the $(\log p(\mathbf{w}|\theta) - \log p(\mathbf{w}))$ term in the loss function. α is the step length for the gradient based optimization method. The gradient calculations are performed with an automatic differential method (autodiff).

NUMERICAL TESTS

MLP IFWI numerical test

In this section, we will use the IFWI to perform the elastic full waveform inversion. The 2D V_p model is obtained through the 3D Overthrust model, and the V_s and density models are calculated through scaling the true V_p model. The size of elastic models is 100×125, and we use $dx = dz = 20m$ as the grad length of the model. The source wavelet is Ricker's directional wavelet with a main frequency of $15Hz$. The maximum receiving time is 2.6 seconds, with $dt = 0.002sec$. To calculale the synthetic training data we use a staggered grid stress velocity finite difference method with 10 cells-width PML boundaries. The maximum number of iterations is set to 2000.

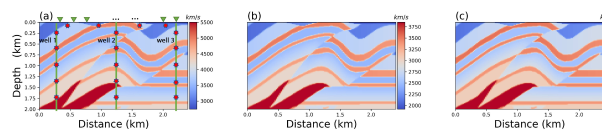

Figure 2: True models. (a) True V_p models. (b)True V_s model. (c) True density ρ model. The green triangles are the receiver positions. The green lines are the well log positions.

Figure 2 (a) shows the acquisition geometries that we use in this study. In these tests, we locate shots on the surface and the well logs, and the receivers are located on the surface of the model. As mentioned above, the MLP network needs the well log information for scaling the output of the network. The well log positions are also illustrated in Figure 2 (a). Figure 3 shows inversion results using well log 1. The subfigures in Figure 3 from left to right, in columns, are the inversion results at 1, 100, 300, 500, and 2000 iterations, and from top to bottom, in rows, are V_p, V_s, and density. We can see that after 300 iterations, the network could generate the general background of the model. After 500 iterations, we can see some geological-meaningful structures at $1km$ depth in the results. After 1000 iterations, more details are added to the velocity models. After 2000 iterations, the structure of the deeper part of the model has been recovered. We can see that in IFWI we can recover the general structure of the investigation area with only one well log information with the absence of an explicit initial model.

BNN IFWI numerical test

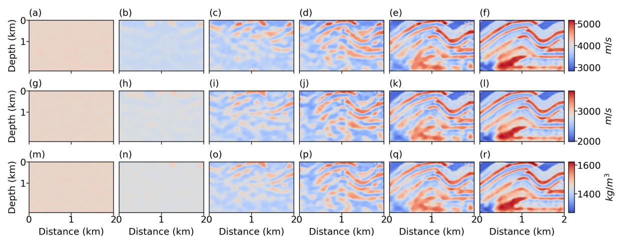

Figure 3: Elastic IFW inversion results using well log 1. Figure (a)-(e) shows the inversion results of the V_p model at 1, 100, 300, 500, 1000 and 2000 epochs. The second and the third row shows the inversion results for V_s and density.

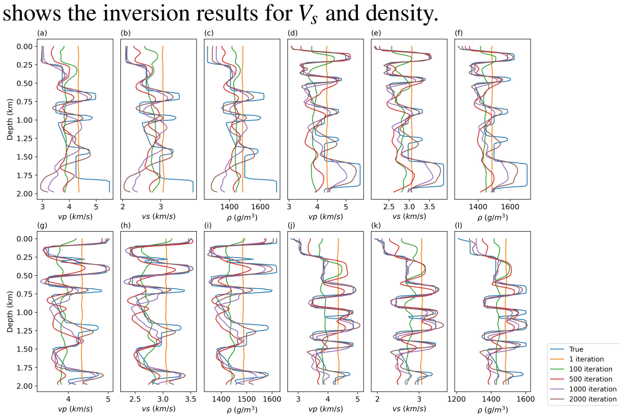

Figure 4: Vertical profile of the IFWI for V_p, V_s and density at 1, 100, 500, 1000, and 2000 iterations.

In this section, we will use the Bayesian neural network to generate velocity models. The size of the model we use in this test is 100×130 and the grid length we use in this test is $dx = dz = 20m$. The main frequency of the source wavelet is 10Hz, and the maximum receiving time is $3s$. The inversion takes 1000 iterations.

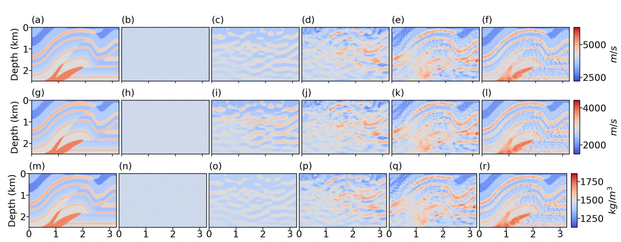

Figure 5: BNN elastic IFWI Overthrust model V_p, V_s and ρ prediction results. The first row is the prediction for V_p. The second row are the predictions for V_s. The third row are the predictions for ρ at different iterations of training.

Figure 5 shows the prediction of the BNN at 10, 50, 100, 500 iterations. In Figure 5, from top to bottom in rows are the V_p, V_s, and ρ predictions respectively. From left to right in columns, in Figure 5 are the true, and BNN IFWI prediction results at 1, 10, 50, 100, 500 iterations. We can also clearly see how the BNN IFWI can correctly recover the velocity models progressively. The final output of the network aligns well with the true models.

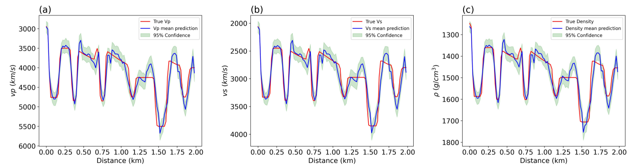

Figure 6: Overthrust model Vertical profile of the BNN elastic IFWI for V_p, V_s and density.

In the BNN, the weights are regarded as probability distributions functions. The models in Figure 5 are just one realization of the models generated with BNN, but different velocity models will be generated for each forward calculation. We produce 100 forward predictions using the trained BNN and then perform the statistic analysis of the results to obtain the mean and the standard deviation of the model. The prediction results for the BNN IFWI are plotted in Figure 6: from left to right, we see the predictions for V_p, V_s and ρ located at 100m of the models. We plot the mean and the 95% confidence for the prediction results, which gives the uncertainty analysis of FWI results, demonstrating that the BNN IFWI could give uncertainty analysis for prediction results.

CONCLUSIONS

In this study, we introduce the MLP based IFWI to perform inversion without using an explicit initial model for mitigating the initial mode-dependent problem for FWI. We also introduce the BNN based IFWI to give the uncertainty analysis for FWI. IFWI consists of two parts. The first part is a coordinated-based network, and the second part is an RNN based neural network forward modeling method. The first part of the network takes the coordinate information as the input and predict the velocity models. In the second part, the velocity models are sending into the RNN for obtaining the synthetic data. The weights in network are updated according the the residual between the synthetic data and observed data. For MLP based IFWI, the weights are fixed values. In BNN IFWI, the weights are regarded as probability distribution functions. The numerical tests demonstrate how the velocity model generated with the MLP based IFWI evolves through iterations. The predictions results of the MLP based IFW align well with the true models. We also show that the BNN based IFWI could give the uncertainty analysis for the prediction results.

ACKNOWLEDGMENTS

We thank the sponsors of CREWES for their continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19. The first author is also supported by the China Scholarship Council (CSC).