Research Progress 2/2021
An implementation of a recurrent neural network for acoustic waveform inversion

Panuwat Pukhamwong g5836046

1. Introduction

Artificial neural network (ANN) is a mathematical tool to find a useful representation of the data. For recent years deep learning [1], artificial neural network with deeper layer, is well-known and successful on processing data. Deep learning provides great results in many fields such as image recognition, speech recognition, machine translation, medical diagnosis, and etc. In geophysics, deep learning have applications on both interpretation and inversion [2].

Full waveform inversion (FWI) is a velocity model building tool. It is a method to determine physical properties of the subsurface from seismic data. Typically, FWI is mathematically formulated as a nonlinear inverse problem in which an objective function is optimized with respect to a model parameter. A review of FWI in exploration geophysics was given in [3]. A neural network was used to find a mapping forward modeling operator in FWI [4]. This network is two hidden layer feed forward network and it was trained with synthetic seismic data using perfectly matched boundary condition. An alternative formulation of FWI can be given as a neural network model such as a convolutional neural network (CNN) or recurrent network [5].

The previous work presented recurrent neural network base full waveform inversion [5]. One of the drawback of this work is that no absorbing boundary was used. Consequently, the deep-learning-based seismic modeling only models wave propagations in bounded domains. As a result, the proposed method cannot be used in the practical situation due to strong boundary reflections. In this work, we implement convolutional perfectly matched layers [6] in to recurrent neural network solving 1D acoustic full waveform inversion. Our neural network full waveform inversion is developed based on a finite difference scheme with convolutional perfectly matched layers.

2. Method

We implement 1D acoustic full waveform inversion (FWI) using Keras and TensorFlow library [7]. Acoustic wave propagation or forward modelling in FWI is developed based on second order finite difference scheme. We design recurrent neural network that able to simulate 1D acoustic wave propagation. Layers in the neural network could perform like differential operator in finite difference manner by assigning weights in this layer and set them to be untrainable parameters. 1D acoustic wave equation is shown in equation (1)

$$\frac{\partial^2 p(x,t)}{\partial t^2} + s(x,t) = v^2 \frac{\partial^2 p(x,t)}{\partial x^2} \tag{1}$$

where p is wave field, which depend on space x and time t, s is source term, and v is velocity, which is a property of the medium. Applying second order finite difference scheme to wave equation (1), wave fields at the next time step can be written in term of the current and the previous wave fields as shown in equation (2)

$$p_i^{k+1} = 2p_i^k - p_i^{k-1} + (\frac{v_i \Delta t}{\Delta x})^2 (p_{i+1}^k - 2p_i^k + p_{i-1}^k) - s_i^k \Delta t^2 \tag{2}$$

where Δx is discrete space, Δt is discrete time, p_i^k is wave field at time $k\Delta t$ and position $i\Delta x$.

The convolutional perfectly matched layer equations are obtained by introducing coordinate stretching to the PML region of wave equation in the frequency domain. The stretching parameter is given by

$$s_x = 1 + \frac{d_x}{\sigma_x + i\omega} \tag{3}$$

where d_x is a damping factor profile, $i^2 = -1$, ω is angular frequency, σ_x is a real positive parameter causing pole shifting. In time domain, the partial derivative in stretched region is giving by

$$\frac{\partial}{\partial \tilde{x}} = \mathcal{F}^{-1}(\frac{1}{s_x}) * \frac{\partial}{\partial x} \tag{4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform, tilde denotes the stretched coordinate. Equation (4) can be solved using the recursive convolution method, it can be written as

$$\frac{\partial}{\partial \tilde{x}} = \frac{\partial}{\partial x} + \phi_x \tag{5}$$

where ϕ_x is an auxiliary variable. Time evolution of ϕ_x is given by

$$\phi_x^k = a_x (\frac{\partial}{\partial x})^k + b_x \phi_x^{k-1} \tag{6}$$

where k denotes time, parameters a_x and b_x are given by

$$a_x = \frac{d_x(b_x - 1)}{d_x + \sigma_x}; \quad b_x = e^{-(d_x + \sigma_x)\Delta t} \tag{7}$$

Laplacian in PML area can be written as

$$\frac{\partial^2 p}{\partial \tilde{x}^2} = \frac{\partial}{\partial \tilde{x}} (\frac{\partial p}{\partial x} + \alpha_x) \tag{8}$$

$$= \frac{\partial}{\partial x} (\frac{\partial p}{\partial x} + \alpha_x) + \beta_x \tag{9}$$

where α_x and β_x are auxiliary variables from equation (6)

$$\alpha_{i+\frac{1}{2}}^k = a_{i+\frac{1}{2}} (\frac{p_{i+1} - p_i}{\Delta x}) + b_{i+\frac{1}{2}} \alpha_{i+\frac{1}{2}}^{k-1} \tag{10}$$

$$\alpha_{i-\frac{1}{2}}^k = a_{i-\frac{1}{2}} (\frac{p_i - p_{i-1}}{\Delta x}) + b_{i-\frac{1}{2}} \alpha_{i-\frac{1}{2}}^{k-1} \tag{11}$$

$$\beta_i^k = \frac{a_i}{\Delta x} [(\frac{p_{i+1} - p_i}{\Delta x} + \alpha_{i+\frac{1}{2}}) - (\frac{p_i - p_{i-1}}{\Delta x} + \alpha_{i-\frac{1}{2}})] + b_i \beta_i^{k-1} \tag{12}$$

$$\frac{\partial^2 p}{\partial \tilde{x}^2} \approx \frac{1}{\Delta x}[(\frac{p_{i+1}-p_i}{\Delta x}+\alpha_{i+\frac{1}{2}})-(\frac{p_i-p_{i-1}}{\Delta x}+\alpha_{i-\frac{1}{2}})]+\beta_i^k \tag{13}$$

Finite difference scheme of 1D acoustic wave equation with convolutional perfectly matched layers (CPML) can be written as follow

$$p_i^{k+1} = 2p_i^k - p_i^{k-1} + (v_i\Delta t)^2 \cdot \psi_i^k - s_i^k \Delta t^2 \tag{14}$$

ψ in equation (14) can be calculated as an algorithm in equations (15). The CPML algorithm requires memory parameters α and β. The algorithm is shown below

$$\begin{aligned}
\alpha_i^k &= a_{i_{half}} \cdot DR(p_i^k) + b_{i_{half}} \cdot \alpha_i^{k-1} \\
\beta_i^k &= a_i \cdot DL(DR(p_i^k) + \alpha_i^k) + b_i \cdot \beta_i^{k-1} \\
\psi_i^k &= DL(DR(p_i^k) + \alpha_i^k) + \beta_i^k
\end{aligned} \tag{15}$$

Operator $DR()$ is first order forward finite difference and operator $DL()$ is first order backward finite difference. $a_i, b_i, a_{i_{half}}, b_{i_{half}}$ are decay parameters of CPML. There decay parameters are zeros everywhere except PML regions. Consequently, in the physical regions, equation (14) become equation (2) which is the second order finite difference of wave equation without PML. See [6, 8] for more details on the decay parameter.

2.1. RNN Cell

The recurrent neural network cell is shown in figure 1. Inputs and outputs of the network are source signals and seismogram respectively. Parameters of CMPL, receiver position, time sampling, discrete space and discrete time are parameters that used to create a network. Layer DR and DL are fixed weight convolutional layers with kernel size three. v is an output form fully connected layer. The weights of the fully connected layer correspond to velocity profile. We only allow this layer to be trainable parameters. Other layers are set to be untrainable. To impose stability of forward modeling in FWI, we have to assign the activation function of the fully connected layer to be a custom rectifier function which has minimum equal to zero and maximum equal to one. The stability is from the 1D wave equation on finite difference scheme. Padding of this layer is custom padding which assigns velocity in PML region equal to velocities at the edges of the physical region. Layer Re is custom layer that picks seismic data at the receiver position in every sampling time. The last layer, Dc, imposes Dirichlet condition to the top of PML region.

3. Experimental setting

Physical domain is 1000 m. Source and receiver are located at 100 m and 80 m respectively. The pulse source is Ricker function with 14 Hz peak frequency. We use a finite difference method for synthesis seismic data. Discrete space and discrete time, time step, are 12.5 m and 2.38 ms respectively. Perfectly matched layers area are 75 m extended from both sides of the physical domain. The data length is 1 s with time sampling 11.9 ms, 5 times larger than the time step in the finite difference process. We test the network with two models of velocity profiles. There are two velocity profiles, model I and model II, as show in figure 2. Seismic traces that correspond to model I and model II are shown in figure 3. The synthesis seismic data are created using second order finite difference scheme with perfectly matched layer. Input and output of the network are source signal and seismic traces respectively. The objective function is a misfit between synthesis seismic trace and the output of the network in the form of mean square error. We work on Keras and TensorFlow library [7]. The optimization algorithm is adaptive moment (Adam) [9].

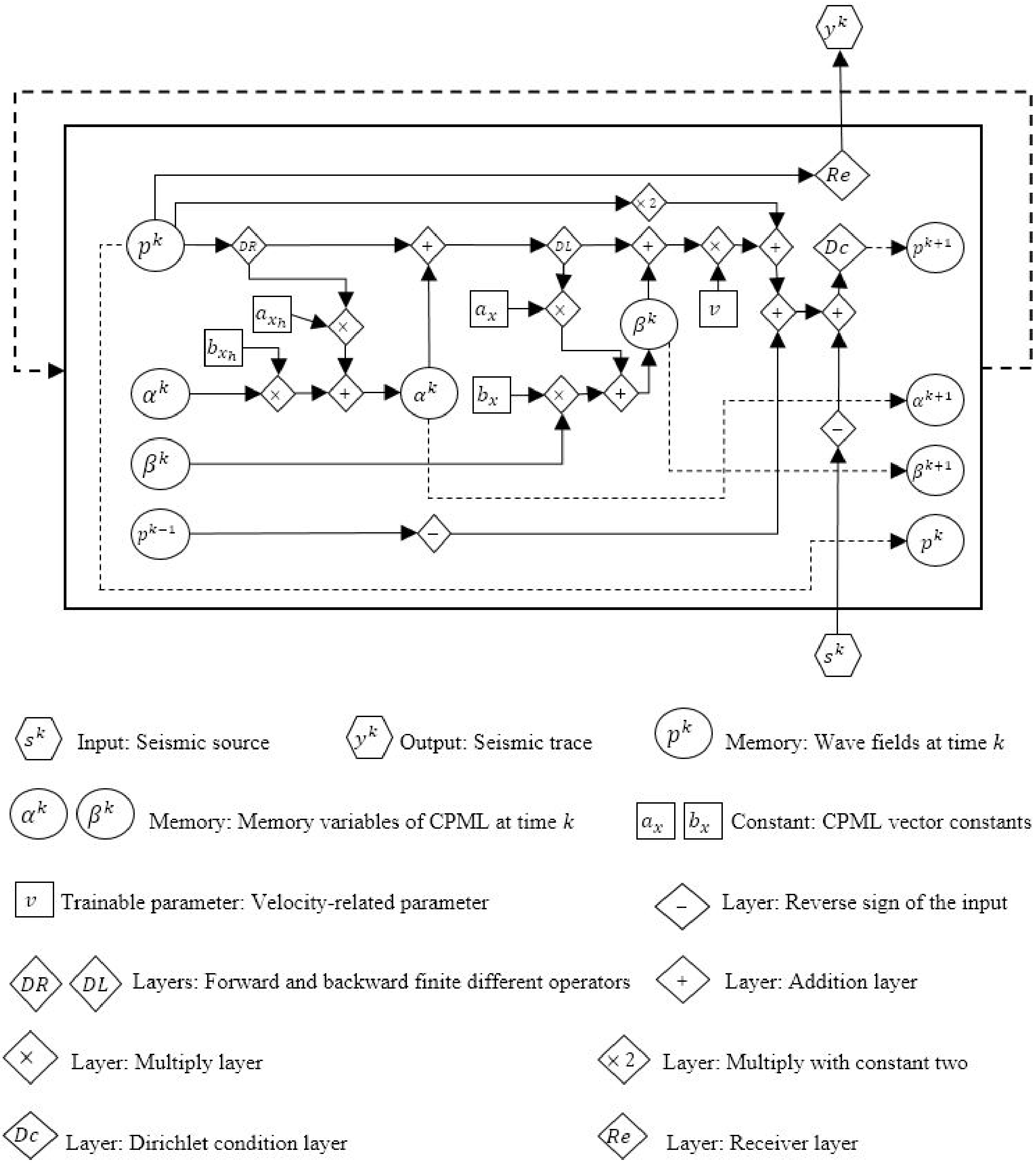

Figure 1. Recurrent neural network cell

4. Results and discussion

The network is trained with seismic trace generated from finite difference with PML method. Initial velocity profiles are smooth true velocity profiles. The trained results with learning rate 40 m/s at epoch 500 are shown in figure 4. These results from both models show that the networks provide good predictions. At epoch 500, it can extract main characteristic of true velocity profiles. There is no unstable in training process. We test the network more about convergence and learning rate by training the network up to 5000 epochs and changing the learning rate to 1, 5, 40 and 100. Loss, or misfit, and predicted velocity error between training process are shown in figure 5 and figure 10. These results agree with the previous work[5] that

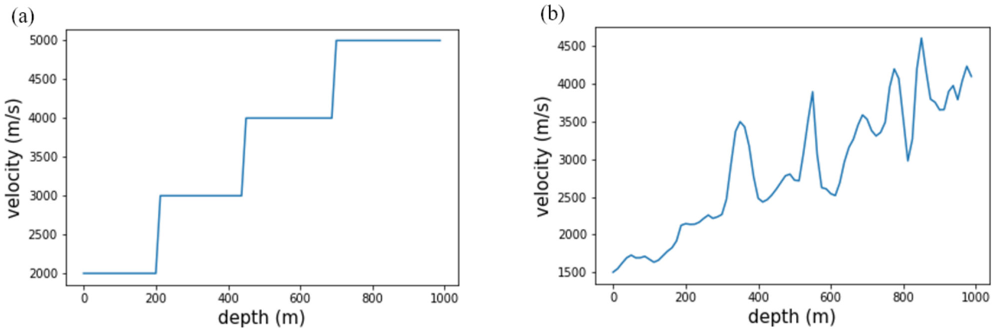

Figure 2. (a) True velocity profile model I. (b) True velocity profile model II.

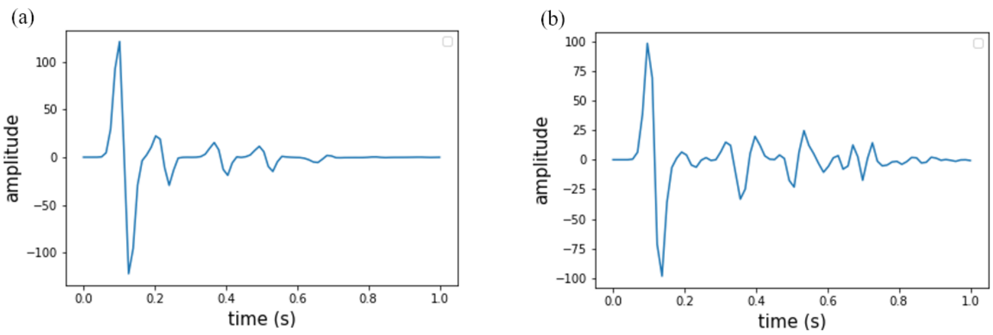

Figure 3. (a) Seismic trace from model I. (b) Seismic trace from model II.

training with learning rate 40 m/s perform a little better than other learning rates in our choice for the model I. Note that we have tested the network for other velocity profiles and other initial profiles, but we did not include it in this work, we found that learning rate 40 m/s is not the best choice in every case. It depends on true velocity profiles and initial profiles.

This recurrent neural network FWI approach dose not need adjoint base method to calculate gradient of objective function instead it use automatic differentiation in machine learning. Auto differentiation process splits gradient calculate into many mini task. It make back propagation fast, the trade off is that it require large memory to store whole hidden states. Another advantage of this approach is that it base on TensorFlow platform which is easy to change and apply optimization algorithms to a network. This method does not need a large number of data to train the network compare to other approach that base on neural network such as [4]. In future work, genetic algorithm could help this approach to find the appropriate optimizer, learning rate, and initial velocity profile for any true velocity profile.

5. preliminary results in 2D case

We have study and modify the network for 2d case. The true velocity and the initial velocity are shown in figure 7. The predicted result at epoch 25 and 100 are shown in figure 8. Dimension of the velocity model is $1000m \times 3000m$. Discrete space is $12.5m$. There are 12 source locations from the left to the right of the model. Another 2D velocity model is vertical 2D velocity. The results are shown in figure 9 and figure 10.

6. Conclusions

We extend recurrent neural network (RNN) base full waveform inversion (FWI) in order to perform on unbounded domains by implementing convolutional perfectly matched layers into a

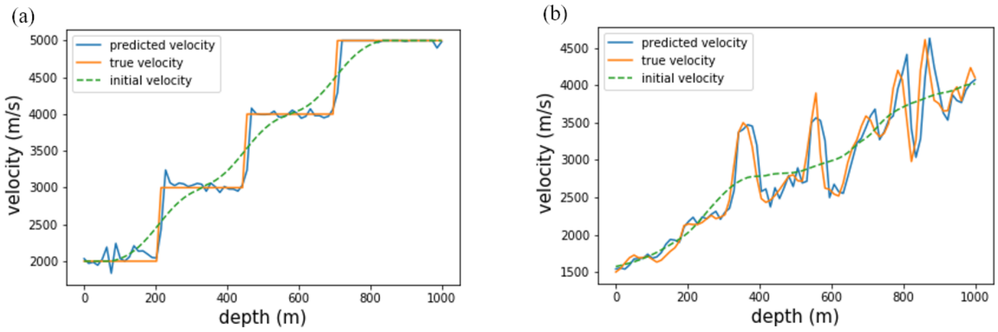

Figure 4. Predicted velocity profile at epoch 500. (a) Model I. (b) Model II.

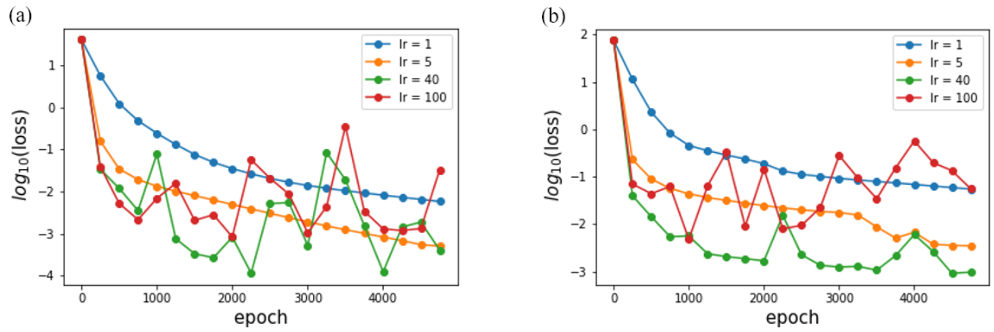

Figure 5. $log_{10}(loss)$ versus epoch. (a) Model I. (b) Model II.

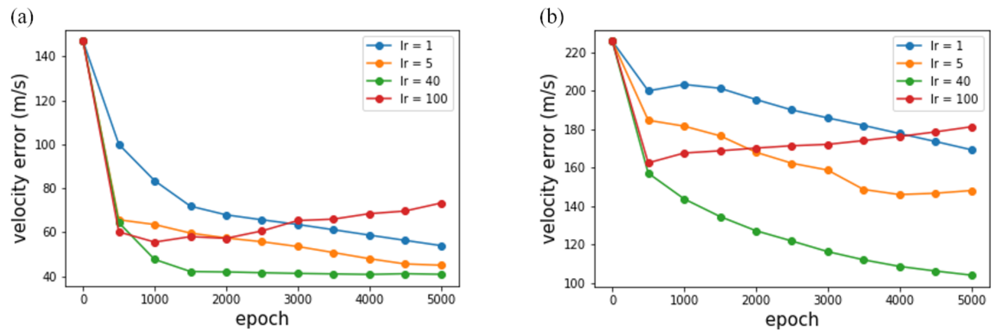

Figure 6. Velocity error versus epoch. (a) Model I. (b) Model II.

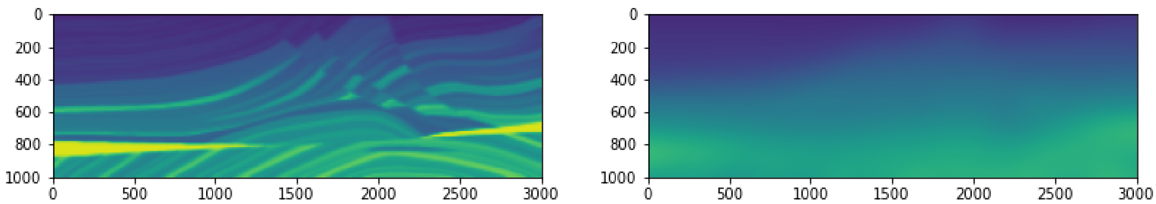

Figure 7. (left) 2D true velocity. (right) 2D initial velocity

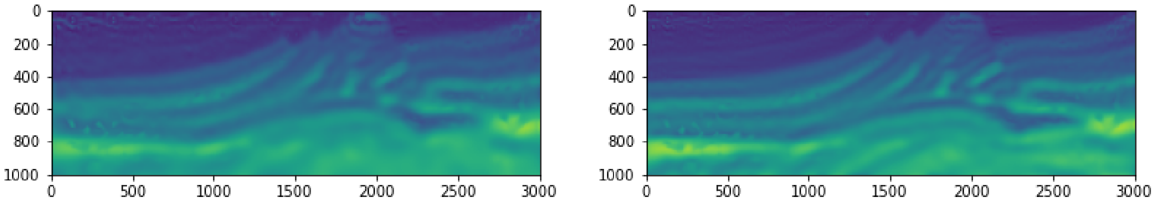

Figure 8. (left) predicted velocity at epoch 25. (right) predicted velocity at epoch 100

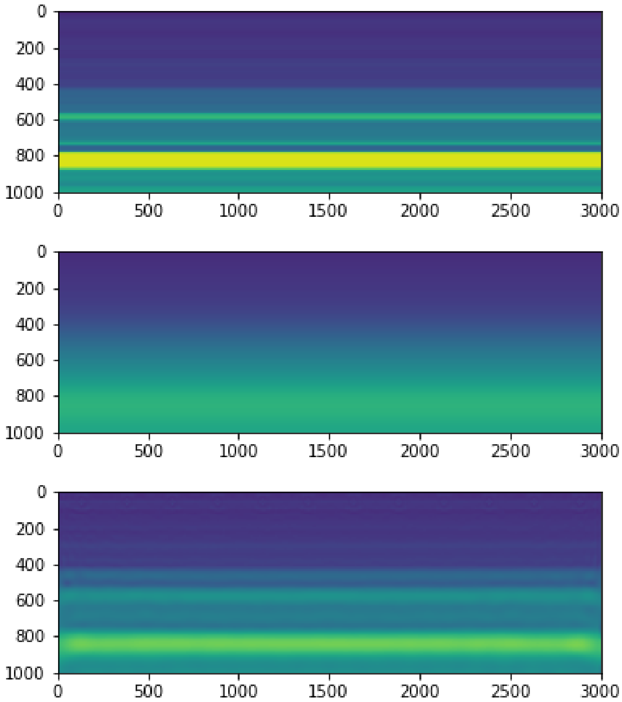

Figure 9. (top) true velocity. (middle) initial velocity. (bottom) predicted velocity at epoch 100

recurrent neural network on Keras and TensorFlow platform. The networks are trained with synthetic seismic data with unbounded domain. The results show that our method is work. The predicted velocity profiles can extract main structure from the true velocity profiles. The result of the stair velocity profile agrees with the previous work that learning rate 40 m/s perform better than others in our choices. This work could be applied to real data.

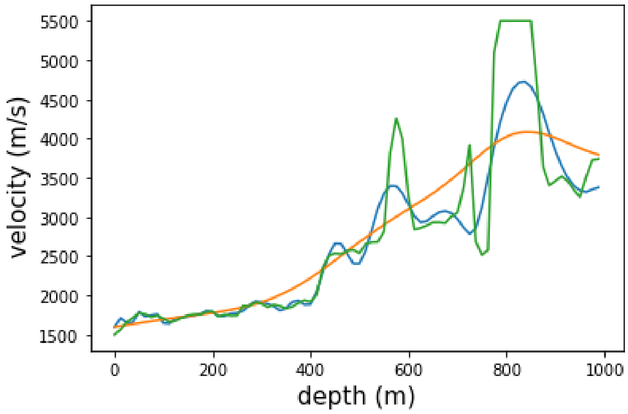

Figure 10. vertical predicted velocity at position 1500 meter

References

[1] Lecun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[2] Zheng Y, Zhang Q, Yusifov A and Shi Y 2019 Applications of supervised deep learning for seismic interpretation and inversion *Lead. Edge* **38** 526–33

[3] Virieux J and Operto S 2009 An overview of full-waveform inversion in exploration geophysics *Geophysics* **74** WCC1–26

[4] Fu H, Zhang Y and Ma M 2019 Seismic waveform inversion using a neural network-based forward *J. Phys.: Conf. Ser.* **1324** 012043

[5] Sun J, Niu Z, Innanen A K, Li J and Trad O D 2020 A theory-guided deep learning formulation and optimization of seismic waveform inversion *Geophysics* **85** R87–99

[6] Dimitri K and Roland M 2007 An unsplit convolutional Perfectly Matched Layer improved at grazing incidence for the seismic wave equation *Geophysics* **72** SM155–67

[7] Abadi M *et al* 2016 TensorFlow: A system for large-scale machine learning *12th {USENIX} Symposium on Operating Systems Design and Implementation OSDI* **16** 265–83

[8] Pasalic D and McGarry R 2010 Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations *SEG Technical Program Expanded Abstracts 2010*

[9] Kingma P D and Ba J 2014 Adam: A Method for Stochastic Optimization *arXiv preprint* arXiv:1412.6980