Physics-Constrained Deep Learning for Solving the Eikonal Equation

Conference Paper · June 2020

DOI: 10.3997/2214-4609.202011764

3 authors:

Serafim Grubas
University of Alberta

Georgy Loginov
A.A. Trofimuk Institute of Petroleum Geology and Geophysics SB RAS

Anton A. Duchkov
A.A. Trofimuk Institute of Petroleum Geology and Geophysics SB RAS

EAGE ANNUAL 2020
82ND CONFERENCE + EXHIBITION
AMSTERDAM | THE NETHERLANDS
NOW IN WINTER
8-11 DECEMBER

Fr_Forum_13

Physics-Constrained Deep Learning for Solving the Eikonal Equation

S. Grubas[1,2]*, G. Loginov[1,2], A. Duchkov[1,2]

[1] Novosibirsk State University; [2] IPGG SB RAS

Summary

The Eikonal equation is a non-linear PDE that is used for modeling seismic traveltimes. Here we test the idea of using neural networks for solving the 2D Eikonal equation. The concept of the physics-informed neural networks implies including the PDE and boundary conditions into the loss functions. Then no labeled data are required for training the network. While testing this approach we show that it is not sufficient to include only the equation and the boundary condition into the loss function as the training procedure may converge to solutions corresponding to various source terms. We propose supplementing the loss function with additional physics constraint promoting monotonic behavior (time increasing away from the source location). We were testing various neural-network architectures for several inhomogeneous velocity models: with linear vertical gradient, with a smooth high-velocity anomaly, the two-layered models. In the tests, the physics-informed neural network was able to reproduce the behavior of propagating fronts with the mean absolute relative error of about 5 % for all the considered tests. Further development of the training strategy is necessary for further accuracy improvement.

Introduction

Traveltime modeling plays a significant role in seismic processing and inversion, e.g. ray modeling (Abgrall and Benamou, 1999), seismic tomography (Li et al., 2013), migration (Gray and May, 1994), inverse kinematic problems (Loginov et al., 2015) etc. In order to compute seismic-wave traveltimes one has to solve the Eikonal equation which is a non-linear partial differential equation (PDE). There are several methods to address this problem including solving the ray-tracing system (Červený et al., 1984) and finite-difference solvers (Zhao, 2005; Sethian, 1996). These numerical algorithms approximate derivatives in the differential equation via finite-difference schemes.

Nowadays the machine-learning methods are successfully used in various applications (Goodfellow et al., 2016). Lichtenstein et al. (2019) used a deep learning approach to numerically approximate the solution to the Eikonal equation locally (without solving the PDE). One can also use artificial neural networks to solve PDEs (Raissi et al., 2019). The main idea is to train a neural network to produce a function satisfying corresponding equation and boundary conditions. This approach is called "physics informed neural networks" (PINNs). The authors demonstrated that PINNs can be used both for forward and inverse problems in the homogeneous case. Here we address the problem of using PINN for solving the Eikonal equation for inhomogeneous velocity models.

Method and Theory

The Eikonal equation is a nonlinear PDE that is written in 2D as follows:

$$\begin{gathered} |\nabla\tau(x,z)|^2 = v(x,z)^{-2}, \\ \tau(x_s,z_s) = 0, \end{gathered} \tag{1}$$

where $v(x,z)$ – velocity of the seismic wave, $\tau(x,z)$ – seismic-wave traveltime for the source point (x_s,z_s) (as follows from the boundary condition $\tau(x_s,z_s)=0$), $\nabla = (\partial/\partial x, \partial/\partial z)$ – gradient operator, $|\cdot|$ - L_2 norm of vector. The problem is to find a solution $\tau(x,z)$ for an arbitrary velocity distribution $v(x,z)$. In case of using neural network, the solution represents as $\tilde{\tau}(x,z,\Theta)$, where Θ is a set of parameters of the neural network. A neural network can be considered as a set of nonlinear transformations over the input data. We introduce solving the Eikonal equation using a fully connected neural network which can be represented by:

$$\begin{gathered} h_0 = \mathbf{X}, \\ h_{l+1} = f_l(h_l * \theta_l), \\ \tilde{\tau}(x,z,\Theta) = h_L, \end{gathered} \tag{2}$$

where $\mathbf{X}$ - matrix of input data, which is the set of points (x_i,z_j) of the computational grid, h_l - hidden state on the l-th layer, f_l - activation function, $*$ - matrix multiplication, θ_l - vector of the weights, L - total number of hidden layers, Θ - set of all θ_l, $\tilde{\tau}(x,z,\Theta)$ - the result of all transformations of the neural network and the assumed solution to the equation.

The training process is performed using the backpropagation algorithm and a loss function is minimized using the adaptive gradient descent (Adam) method. At the zero iteration of training, weights of the neural network are randomly initialized, then the weights are updated at each subsequent iteration. To simulate the wave propagation a training schedule is used. The wavefront, over time, expands and moves away from the source. Thus, the domain of definition of the function increases gradually and this imitation of expanding fronts is used. In the beginning, the domain of definition of the function $\tilde{\tau}$ is a small set of points near the source. The neural network is trained at this small area, then adjacent points are added at each new iteration. As a result, training is taking place in a growing field.

Input data are a set of points (x_i,z_j) of the computational grid X,Z, where we define the solution. The grid is shifted to the source location (x_s,z_s) and normalized by the maximum absolute value of the grid $M = max(|X,Z|)$. Thus all values are within [-1, 1]. Velocity model V is used only in the loss function, but is also normalized by the maximum value in the model V_{max}:

$$X' = \frac{X-x_s}{M},\ Z' = \frac{Z-z_s}{M},\ V' = \frac{V}{V_{max}}. \tag{3}$$

Since the input data and velocity model are normalized, the output value of the neural network $\tilde{\tau}'(x',z',\Theta)$ is normalized as well. These values $\tilde{\tau}'$ are used in the loss function. To scale the values $\tilde{\tau}'$ back to the computational grid X, Z, and V we use inversion of the normalization:

$$\tilde{\tau}(x,z,\Theta) = \frac{M}{V_{max}} \tilde{\tau}'(x',z',\Theta). \tag{4}$$

Loss function development

To train the PINN we assume that during minimization of the loss function a PINN solution is converging to the true solution. Therefore, all constraints to the solution must eliminate any possible wrong solutions. To construct the loss function following constraints to the solution are considered: the equation, the boundary condition, the positivity, and the monotonicity relative to the seismic ray. The positivity constraint is obvious since the solution where $\tilde{\tau}' < 0$ isn't physical. The monotonicity is more complicated because traveltime increases strictly along a seismic ray. But to calculate seismic rays, the Eikonal equation must be resolved in advance. Therefore, we propose to consider monotonicity relative to the distance from the seismic source location to each node of the computational grid. Therefore, the zero point $(0,0)$ of the grid is shifted to the source location (x_s,z_s) (see formula 3). The monotonicity can be taken into account by using a positivity of first partial derivatives $sgn(x')(\partial\tilde{\tau}'/\partial x') > 0$ and $sgn(z')(\partial\tilde{\tau}'/\partial z') > 0$ at each point (x',z'). Such an approach assumes that the velocity model doesn't have sharp local anomalies. Therefore, the rays which curve for more than 90 degrees with respect to the vector from the source aren't considered. Each constraint is a term in the loss function $J(\Theta)$ and represent mean squared error:

$$\begin{aligned} J(\Theta) = {} & \frac{1}{N}\sum_{x'_i,z'_j} \left| \left|\nabla\tilde{\tau}'\right|^2 - v'^{-2} \right|^2 + \lambda_1 \left|\tilde{\tau}'_0 - \tau_0\right|^2 + \lambda_2 \frac{1}{N}\sum_{x'_i,z'_j} H\left(-\tilde{\tau}'\right)\left|\tilde{\tau}'\right|^2 + \\ & + \lambda_3 \frac{1}{N}\sum_{x'_i,z'_j} \left(H\left(-\frac{\partial\tilde{\tau}'}{\partial x'} sgn(x')\right) \left|\frac{\partial\tilde{\tau}'}{\partial x'}\right|^2 + H\left(-\frac{\partial\tilde{\tau}'}{\partial z'} sgn(z')\right) \left|\frac{\partial\tilde{\tau}'}{\partial z'}\right|^2 \right), \end{aligned} \tag{5}$$

where $\tilde{\tau}' = \tilde{\tau}'(x',z',\Theta)$ - the neural network, $\tilde{\tau}'_0 = \tilde{\tau}'(0,0,\Theta)$ - value of the neural network at the source location, $\tau_0 = \tau(x_s,z_s) = 0$ - true solution at the source location, (x'_i,z'_j) - the normalized computational grid, H - Heaviside function (indicator for the constraints), $\lambda_{1,2,3}$ - regularization parameters, N - the number of the points for training. The first term is the Eikonal equation (E), the second one is the boundary condition (B), the third one is the positivity (P), the fourth one is the monotonicity (M). The influence of each term is defined by regularization parameters.

Experiments

A fully-connected neural network with 3 hidden layers and 50 units on each layer is used. Activation on each layer is a hyperbolic tangent function. For training, we used the Adam adaptive gradient descent algorithm with a learning rate equal to 0.0005. The computational grid has a size of 51x51 nodes and limits X, $Z \in [0,1000]$, $[0,1000]$ meters for all tests.

The first example is a model with a vertical gradient of velocity, where an analytical solution is known (Alkhalifah and Fomel, 2010). Figure 1 shows an example of solving the eikonal equation for a seismic source located in the center $x_s = 500$, $z_s = 500$ and demonstrates the influence of the constraints on the PINN solution. One can see that taking only E and B into account (fig. 1.a) doesn't allow PINN to converge to the desired solution from the seismic source. This solution is suitable for the Eikonal equation, but it is not what we want. Adding the P constraint (fig. 1.b) makes the solution positive, but it is not the desired solution as well. The gradient descent algorithm gets to a local minimum of the loss function and stays there. P is the necessary condition but isn't sufficient for the problem. M constraint is stronger and moreover simultaneous taking B and M into account makes the solution positive automatically (fig. 1.c). Whole training time took about 3 minutes, and the mean absolute error relative to the analytical solution is about 3.5 ms (3.0 %). The neural network gives various results each time because initialization of the weights is random and 10 independent tests for one location of the source give errors from 3 to 6 %. For different seismic locations, the mean error is about 5 %. One can see that the PINN solution (fig. 1.c) has convexity along the horizontal line across the source. There is a possibility that the neural network creates complex sources with different times of explosions and it satisfies all the constraints. This behaviour of the PINN solution must be limited, but it's required additional constraint. These examples showed that P constraint is not necessary when M is included. Finally, the loss function includes 3 terms E, B ($\lambda_1 = 1$), and M ($\lambda_3 = 1$) without P ($\lambda_2 = 0$).

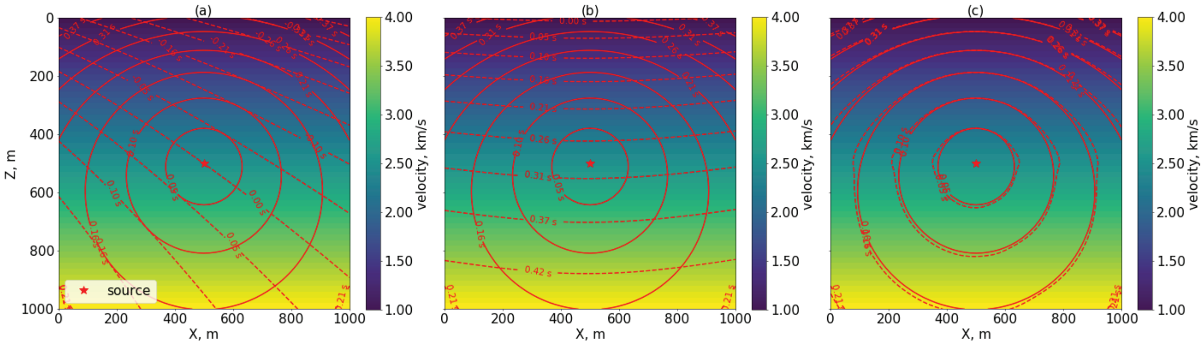

Figure 1 *Examples of solving the Eikonal equation by PINN for the model with the vertical gradient. Three types of constraints in the loss functions are considered. The loss function includes: (a) - E and B; (b) - E, B, and P; (c) - E, B, and M. The red contours are wavefronts for the analytical solution (solid lines) and PINN solution (dashed). The red star is the source location.*

The next test is for different types of velocity models: with local high-velocity anomaly, two layers with high- and low-velocity lower layer (see figure 2). The solutions for the models with local anomalies were calculated by FSM (Nikitin et al., 2018) to compare with the PINN solution. The time fields for the layered models were calculated analytically. The relative absolute errors are within 3-6 % for all models. The PINN solution for the high-velocity anomaly is close to the FSM solution and the wavefronts propagate through the anomaly correctly (fig. 2.a). Though the PINN can manage with the transition into the high-velocity and low-velocity layers (fig. 2.b and 2.c), the headwave is difficult to be represented.

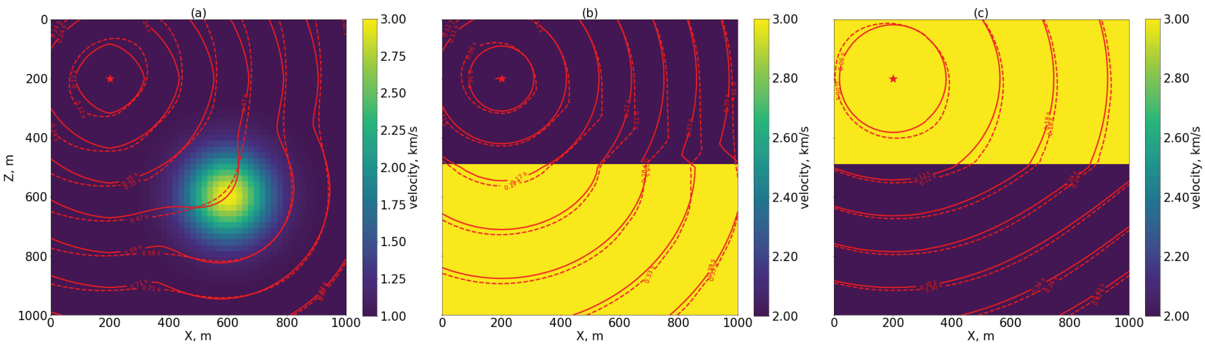

Figure 2 *Examples for different velocity models with: (a) - local high-velocity anomaly; (b) - two layers with a high-velocity lower layer; (c) - two layers with a low-velocity lower layer. The red contours are wavefronts for the analytical and FSM solutions (solid lines), and PINN solutions (dashed). The red star is the source location.*

Conclusions

The tests above show that PINNs can manage with solving the Eikonal equation without a training dataset if we take into account the monotonicity, but the problem is still challenging. The tests showed that PINN is able to reproduce the behavior of propagating fronts with the mean absolute relative error of about 5 % for all the considered tests. But sometimes the solution still has wrong features like convexity due to the additional sources. The next steps to increase the accuracy are to improve the physics constraints. The monotonicity may be implemented by conjugating with a ray-tracing procedure and the existence of the additional sources may be constrained by the laplacian of the traveltime field. Further development of the training strategy is necessary to enhance performance and accuracy.

Discussion

In this paper forward problem of the Eikonal equation is considered. The next problem is inverse problem. Since a velocity model is included in the loss function, consequently, it can be weights of the neural network as well. Such an approach assumes that we will find both a solution and a velocity model for the solution. In this case, there are a lot of uncertainties, therefore the inverse problem can be considered separately. For example, we have a solution on surface or in wells and know the location of the seismic source, then we can continue the known solution into the media and restore the velocity model. If velocity mode is known, the source location can be restored by PINN as well. An additional advantage of the neural network approach is an opportunity to use any non-regular computational grid.

Acknowledgements

The authors are grateful to Alexander Chapovsky for discussions and consultations on this work. His advice allowed significant progress in solving this problem.

The research was carried out with the financial support of the RFBR in the boundaries of the research project No. 18-35-00412.

References

Abgrall, R. and Benamou, J.D. [1999] Big ray-tracing and eikonal solver on unstructured grids: Application to the computation of a multivalued traveltime field in the Marmousi model. *Geophysics*, **64**(1), 230–239.

Alkhalifah, T. and Fomel, S. [2010] An eikonal-based formulation for traveltime perturbation with respect to the source location. *Geophysics*, **75**(6), T175–T183.

Červenỳ, V., Klimeš, L. and Pšenčík, I. [1984] Paraxial ray approximations in the computation of seismic wavefields in inhomogeneous media. *Geophysical Journal International*, **79**(1), 89–104.

Goodfellow, I., Bengio, Y. and Courville, A. [2016] *Deep learning*. MIT press.

Gray, S.H. and May, W.P. [1994] Kirchhoff migration using eikonal equation traveltimes. *Geophysics*, **59**(5), 810–817.

Li, S., Vladimirsky, A. and Fomel, S. [2013] First-break traveltime tomography with the double-square-root eikonal equation. *Geophysics*, **78**(6), U89–U101.

Lichtenstein, M., Pai, G. and Kimmel, R. [2019] Deep Eikonal Solvers. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 38–50.

Loginov, G., Yaskevich, S., Duchkov, A. and Serdyukov, A. [2015] Joint processing of surface and underground microseismic monitoring data in hard mineral mining. *Journal of Mining Science*, **51**(5), 944–950.

Nikitin, A.A., Serdyukov, A.S. and Duchkov, A.A. [2018] Cache-efficient parallel eikonal solver for multicore CPUs. *Computational Geosciences*, **22**(3), 775–787.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. [2019] Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, **378**, 686–707.

Sethian, J.A. [1996] A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, **93**(4), 1591–1595.

Zhao, H. [2005] A fast sweeping method for eikonal equations. *Mathematics of computation*, **74**(250), 603–627.