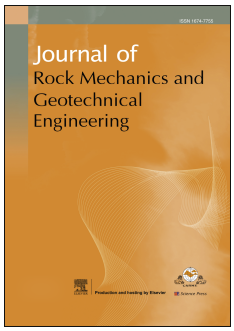
Journal Pre-proof



Physics-informed deep learning for one-dimensional consolidation

Yared W. Bekele

PII: S1674-7755(20)30138-4

DOI: https://doi.org/10.1016/j.jrmge.2020.09.005

Reference: JRMGE 722

To appear in: *Journal of Rock Mechanics and Geotechnical Engineering*

Received Date: 4 May 2020

Revised Date: 18 August 2020

Accepted Date: 9 September 2020

Please cite this article as: Bekele YW, Physics-informed deep learning for one-dimensional consolidation, *Journal of Rock Mechanics and Geotechnical Engineering*, https://doi.org/10.1016/j.jrmge.2020.09.005.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Institute of Rock and Soil Mechanics, Chinese Academy of Sciences. Production and hosting by Elsevier B.V. All rights reserved.

Journal Pre-proof

Physics-informed deep learning for one-dimensional consolidation

Yared W. Bekele

Rock and Soil Mechanics Group, SINTEF AS, Trondheim, Norway

ARTICLEINFO

Article history:
Received 4 May 2020
Received in revised form
18 August 2020
Accepted 9 September 2020
Available online

Keywords:
Physics-informed deep learning
Consolidation
Forward problems
Inverse problems

ABSTRACT

Neural networks with physical governing equations as constraints have recently created a new trend in machine learning research. In this context, a review of related research is first presented and discussed. The potential offered by such physics-informed deep learning models for computations in geomechanics is demonstrated by application to one-dimensional (1D) consolidation. The governing equation for 1D problems is applied as a constraint in the deep learning model. The deep learning model relies on automatic differentiation for applying the governing equation as a constraint, based on the mathematical approximations established by the neural network. The total loss is measured as a combination of the training loss (based on analytical and model predicted solutions) and the constraint loss (a requirement to satisfy the governing equation). Two classes of problems are considered: forward and inverse problems. The forward problems demonstrate the performance of a physically constrained neural network model in predicting solutions for 1D consolidation problems. Inverse problems show prediction of the coefficient of consolidation. Terzaghi's problem, with varying boundary conditions, is used as a numerical example and the deep learning model shows a remarkable performance in both the forward and inverse problems. While the application demonstrated here is a simple 1D consolidation problem, such a deep learning model integrated with a physical law has significant implications for use in, such as, faster real-time numerical prediction for digital twins, numerical model reproducibility and constitutive model parameter optimization.

© 2020 Institute of Rock and Soil Mechanics, Chinese Academy of Sciences. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Machine learning has been a fast-growing field of research over the last years with ever growing areas of application outside pure computer science. Deep learning, a particular subset of machine learning which uses artificial neural networks (ANNs), is being applied in various disciplines of science and engineering with usually surprising levels of success. Recently, the application of deep learning related to partial differential equations (PDEs) has been picking up pace. Several researchers are contributing to this effort where different names are given to the use of deep learning associated with physical systems governed by PDEs. Some of the commonly encountered labels include physics-informed neural networks, physics-based deep learning, theory-guided data science, and deep hidden physics models, to name a few. In general, the aims of these applications include improving the efficiency, accuracy and generalization capability of numerical methods for the solution of PDEs.

Data-driven solution of PDEs was recently presented by Raissi et al. (2019a). The authors investigated various differential equations and demonstrated how deep learning models can be applied in forward and inverse problem settings. Some of the PDEs studied include Burgers' and Navier-Stokes equations. The forward problems demonstrated how deep learning models can be trained based on sample data from exact solutions while optimizing an embedded governing PDE. In the inverse problems, deep learning models were trained to identify the coefficients of the PDEs. For practical problems in science and engineering, this is equivalent to identifying material parameters based on given exact or numerical solutions. The neural network models for both the forward and inverse problems showed astonishing levels of accuracy. Bar-Sinai et al. (2019) presented a similar study where the emphasis was on learning data-driven discretizations that are best suited to a PDE with certain boundary conditions. A related study combining deep learning and PDEs was presented by Sirignano and Spiliopoulos (2018) where the authors introduced a so-called deep Galerkin method (DGM). The proposed method was applied to different PDEs. This method was also applied to application problems such as in quantitative finance and statistical mechanics, governed by the Black-Scholes and Focker-Planck PDEs, respectively (Al-Aradi et al., 2018). The application areas are increasing rapidly with different variations in the general methodology. A deep learning-based solution of the Euler equations for modeling high-speed flows was presented by Mao et al. (2020) where physics-informed neural networks were used for forward and inverse problems. Deep learning for computational fluid dynamics, in particular for vortex-induced vibrations, was presented by Raissi et al. (2019b). A related work for predictive large-eddy-simulation wall modeling was presented by Yang et al. (2019). The solution of time-dependent stochastic PDEs using physics-informed neural networks by learning in the modal space was demonstrated by Zhang et al. (2019a). A conceptual framework for theory-guided deep learning was presented by Karpatne et al. (2017). Application of deep learning, with physics-informed recurrent neural networks, to fleet prognosis was presented by Nascimento and Viana (2019). Bending analysis of Kirchhoff plates using a deep learning approach was shown by Guo et al. (2019). Deep learning-based study of linear and nonlinear diffusion equations to learn parameters and unknown constitutive relationships was presented by Tartakovsky et al. (2018). Other recent and related studies are those by Huang et al. (2019), Tipireddy et al. (2019), Yang and Perdikaris (2019), Zhang et al. (2019b), Zheng et al. (2019), Jia et al. (2020), Meng and Karniadakis (2020), Sun et al. (2020), and Xu and Darve (2020). While the neural network architecture used in many studies is a feed-forward network with the desired number of layers and hidden units, a recent study applied convolutional neural networks for the solution of the Poisson equation with varying meshes and Dirichlet boundary conditions (Özbay et al., 2019).

Soil consolidation coupled with equilibrium equations constitutes

E-mail address: yared.bekele@sintef.no

Journal Pre-proof

the foundation of numerical modeling in geomechanics. One of the greatest challenges in numerical modeling in geotechnical engineering is the optimization of material and model parameters and faster inverse analyses for parameter determination. This is especially significant when performing large-scale simulations with complex constitutive models. Physics-informed deep learning models can be used to enhance such simulations where they can be applied in both forward and inverse problem settings. Another important application of physics-informed deep learning models is faster numerical prediction for digital twins where examples in geotechnical engineering include monitored foundations of critical infrastructure, geotechnical large-scale experiments and monitored slopes. To this end, the potential offered by physics-informed deep learning models can be demonstrated through application to one-dimensional (1D) consolidation, one of the most ubiquitous problems in geomechanics. Even though 1D consolidation involves only a single governing equation and only one material parameter (the coefficient of consolidation), the concept can be upscaled for problems in geomechanics involving coupled governing differential equations with several material and model parameters. Analysis of soil consolidation and settlement is an important problem in geotechnical engineering. The theory of consolidation describes fluid flow and excess pore water pressure dissipation in porous media, or particularly soils in geotechnics. The pioneering works of Terzaghi and Biot have contributed the most to the theory (Biot, 1941; Terzaghi et al., 1996), from first formulation for a 1D case to later generalization for three-dimensional cases. Analytical solutions exist for the solution of the governing equation for 1D cases (Verruijt, 2013). For complex materials or higher-dimensional problems, numerical methods such as the finite difference method and the finite element method are applied (Schrefler, 1987). For a soil layer or layers sustaining loads in initially undrained conditions, the process of consolidation determines how long it takes for excess pore pressures to dissipate and how much settlement will occur. Good estimates on the degree of consolidation and settlement highly depend on the accuracy of the material parameters determined from laboratory tests and the solution method used, whether analytically or numerically.

This paper presents a study on the application of physics-informed deep learning model for 1D consolidation. The governing equation for the problem is first discussed briefly. The deep learning model for the governing PDE is then described. The problem is studied both for forward and inverse problems and the results from these are presented subsequently. Resources related to this work can be found on the author's GitHub page here.

2. Governing equation

The theory of consolidation describes the dissipation of fluid from a porous medium under compressive loading, thereby causing delay of the eventual deformation of the porous medium. The governing equation for 1D consolidation is given by

$$\frac{\partial p}{\partial t} - \frac{\alpha m_v}{S + \alpha^2 m_v}\frac{\partial \sigma_{zz}}{\partial t} - c_v \frac{\partial^2 p}{\partial z^2} = 0 \tag{1}$$

where p is the pore fluid pressure, α is the Biot coefficient, S is the storativity of the pore space, m_v is the confined compressibility of the porous medium, σ_{zz} is the vertical effective stress, and c_v is the coefficient of consolidation. The classical 1D consolidation problem is that at time $t = 0$, a compressive load in the direction of fluid flow is applied and the load is maintained for $t > 0$ (Verruijt, 2013). This implies that, for $t > 0$, the stress σ_{zz} is constant, i.e. with a magnitude q. Thus, we can reduce the general governing equation in Eq. (1) as

$$\left.\begin{aligned} p = p_o = \frac{\alpha m_v}{S + \alpha^2 m_v} q \quad (t = 0) \\ \frac{\partial p}{\partial t} - c_v \frac{\partial^2 p}{\partial z^2} = 0 \quad (t > 0) \end{aligned}\right\} \tag{2}$$

The first equation in Eq. (2) establishes the initial condition for the 1D consolidation problem where at $t = 0$, the total vertical load is carried by the pore fluid and any fluid dissipation does not yet occur from the porous medium. The second equation governs the dissipation rate of the pore fluid as a function of both time and spatial dimension. This equation may be solved for various drainage boundary conditions (Fig. 1) at the top and bottom of the porous medium through either analytical or numerical methods. We consider an analytical solution here for two different drainage boundary conditions, which are described in a later section.

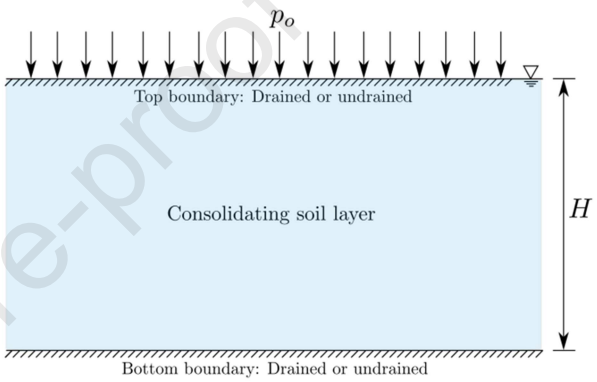


Fig. 1. One-dimensional consolidation.

3. Deep learning model

In this section, the neural network architecture is first discussed and the approach for applying a physical constraint, based on the governing 1D consolidation equation, is then presented. Automatic differentiation is briefly discussed. The model training procedure and the hyper-parameters that are controlled during training are also presented.

3.1. Neural network architecture

A fully-connected deep neural network with the desired number of hidden layers and hidden units is used as a model to be trained with the 1D consolidation problem. An illustration of the neural network architecture is shown in Fig. 2. For the 1D consolidation problem herein, the input layer provides inputs of (z, t) values from the training data, which usually include initial and boundary condition data. The details are discussed in forward and inverse numerical example sections later. The neural network with the desired number of hidden layers and hidden units predicts the excess pore pressure, which is then used to compute the loss based on the excess pore pressure training data. The neural network also includes a physical constraint based on the governing 1D consolidation equation, where the constraint is evaluated using automatic differentiation, briefly discussed in a sub-section below. The neural network is designed to optimize both the training loss and the physical constraint.

3.2. Automatic differentiation

A key part of the deep learning model for the problem here is automatic differentiation. It is important to not confuse automatic differentiation with other methods of computing derivatives in computer programs. There are four ways of computing derivatives

using computers (Baydin et al., 2017): (i) manually obtaining the derivatives and coding them; (ii) numerical differentiation using finite difference approximations; (iii) computer-based symbolic differentiation and subsequent evaluation based on the algebraic expressions; and (iv) automatic differentiation, which is used herein. Like the other methods, automatic differentiation provides numerical values of derivatives obtained using the rules of symbolic differentiation but keeping track of derivative values instead of obtaining the final expressions. This approach of tracking derivative values makes automatic differentiation superior to the two most commonly used methods of computing derivatives, namely numerical differentiation and symbolic differentiation. Automatic differentiation exploits the fact that any derivative computation, no matter how complex, is composed of a sequence of elementary arithmetic operations and elementary function evaluations. It applies the chain rule repeatedly to these operations until the desired derivative is computed. Such an approach for computation makes automatic differentiation to be accurate at machine precision and computationally less demanding than other methods. Once a deep neural network is constructed for 1D consolidation with the pore pressure as the output variable and the spatial and temporal coordinates as inputs, automatic differentiation is used to estimate the derivatives involved in the governing equation based on the universal mathematical expression established by the neural network. For this purpose, the automatic differentiation capability in TensorFlow is utilized. TensorFlow is an open-source software developed by the Google Brain team at Google and it is a symbolic math library that can be used for different tasks such as data flow, differentiable programming and machine learning (Abadi et al., 2016a, b). It provides an application programming interface (API) for automatic differentiation by recording all operations and computing the gradients of the recorded computations using reverse mode differentiation. The Python-based open-source library for neural networks called Keras, which has been integrated in TensorFlow in the latest versions, is used for implementation of the deep learning model here. The advantage of Keras with TensorFlow as the back-end is that it is user-friendly, modular, extensible, and allows for a faster experimentation for deep learning (Gulli and Pal, 2017).

3.3. Model training and hyper-parameters

The deep learning model training is performed in slightly different ways for forward and inverse problems. However, the model hyper-parameters in both cases are adjusted in a similar way. For forward problems, the training data involve initial and boundary condition data, i.e. a pair of initial and boundary (z, t) values and the corresponding excess pore pressure values $p(z, t)$. The model predicts the excess pore pressure value $\hat{p}$ for a given data point. The training loss is calculated as a mean squared error from:

$$MSE_{\mathrm{p}} = \frac{1}{N}\sum_{k=1}^{N} | p(z_k, t_k) - \hat{p}(z_k, t_k) |^2 \tag{3}$$

where N is the number of training data, and (z_k, t_k) represents the training data point. The physical constraint based on the governing PDE is applied at randomly generated collocation points (z_c, t_c). The collocation points are generated using a Latin hypercube sampling strategy where the bounds of the original training data are taken into consideration. The physical constraint is evaluated at the collocation points based on the predicted excess pore pressure using automatic differentiation:

$$f_{\mathrm{c}} = \frac{\partial \hat{p}}{\partial t_{\mathrm{c}}} - c_{\mathrm{v}} \frac{\partial^2 \hat{p}}{\partial z_{\mathrm{c}}^2} \tag{4}$$

The constraint loss is calculated as a mean squared error from the following equation:

$$MSE_{\mathrm{c}} = \frac{1}{N_{\mathrm{c}}}\sum_{k=1}^{N_{\mathrm{c}}} | f_{\mathrm{c}}(z_{\mathrm{c},k}, t_{\mathrm{c},k}) |^2 \tag{5}$$

where N_c is the number of collocation points, and ($z_{c,k}$, $t_{c,k}$) represents a data point from the collocation points. The total loss from training and the physical constraint is defined as

$$MSE = MSE_{\mathrm{p}} + MSE_{\mathrm{c}} \tag{6}$$

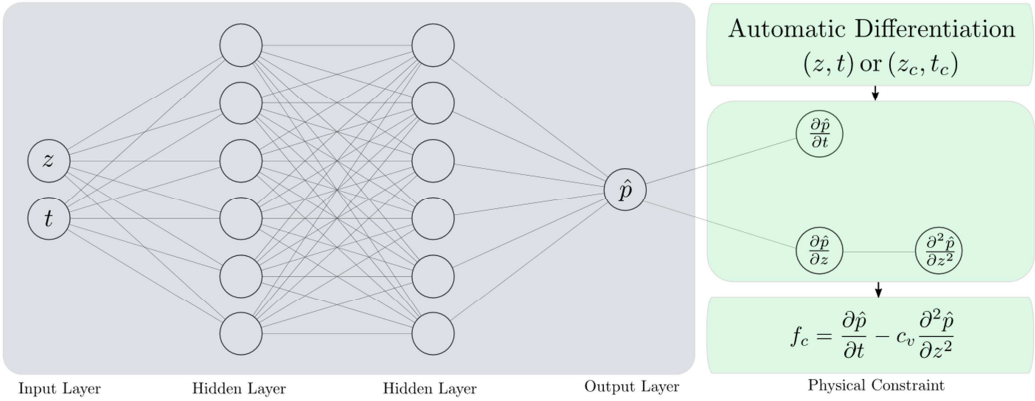


Fig. 2. Illustration of the neural network architecture with input, hidden and output layers. The activation function used at the hidden units is tanh(x), where x denotes the total weighted and bias-added input to the hidden unit. Automatic differentiation is used to determine the partial derivatives in the governing equation and is used as a physical constraint to optimize together with the prediction error based on training data. The number of hidden layers and hidden units in this figure is for illustration only, and the actual number of hidden layers and hidden units used for different cases are discussed in a later section.

which is minimized by the model optimizer. For inverse problems, the training procedure and loss evaluation are mostly similar with some

differences. A larger size of (z, t) training data is used and collocation points are not generated in this case. This implies that automatic differentiation for the physical constraint is evaluated at the original training data points and the coefficient of consolidation is defined as a trainable parameter:

$$f_c = \frac{\partial \hat{p}}{\partial t} - c_{vt} \frac{\partial^2 \hat{p}}{\partial z^2} \tag{7}$$

where c_{vt} is the trained value of the coefficient of consolidation updated during each step. An additional trainable model variable or weight w_{cv}, associated with the coefficient of consolidation, is introduced in the neural network and is used to evaluate c_{vt} based on an exponential function to always guarantee positive values during training:

$$c_{vt} = \exp(w_{cv}) \tag{8}$$

The hyper-parameters tuned during training, both for forward and inverse problems, include the number of hidden layers, number of hidden units, batch size and learning rate. The batch size is adjusted to control the number of samples from the training data that are passed into the model before updating the trainable model parameters. The total loss here is minimized using the Adam optimizer where its associated learning rate is tuned during the training process. After training, the model is used to predict the pore pressures (and the coefficient of consolidation for inverse problems) for given input data. The model is improved by making the necessary adjustments for bias-variance tradeoff. If the model is observed to have high bias, the neural network architecture (numbers of hidden layers and hidden units) is adjusted and/or the model is trained longer. In case of high variance, the amount of training data is increased and/or the neural network architecture is adjusted. Regularization techniques can also be used for decreasing variance, but for the models trained in this paper, no regularization is used.

4. Forward problems

The problem considered here is a classical 1D consolidation problem, usually referred to as Terzaghi's problem, as illustrated in Fig. 3. The problem examines the dissipation of excess pore pressure with time from the soil column due to the application of a surcharge load of magnitude p_o at the top boundary. This numerical example is studied for two drainage boundary conditions: with only the top boundary drained and with both the top and bottom boundaries drained. The training data for forward problems are selected to include initial and boundary condition data, as shown in Fig. 4. The nodes shown in the figure are for illustration only and the actual numbers of nodes and time steps depend on the spatial and temporal discretization used to obtain the analytical solution to the problem. The total number of training points is divided into batches during training by using a specified batch size. The results from this numerical example for the two different boundary conditions are presented in the following sub-sections.

4.1. Consolidation with a drained top boundary

The first case considered is a variation of Terzaghi's problem where excess pore pressure dissipation is allowed at the top boundary only, i.e. the bottom boundary is considered impermeable. These boundary conditions are expressed mathematically as

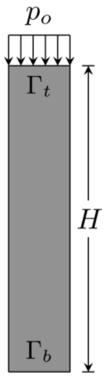


Fig. 3. Example for one-dimensional consolidation.

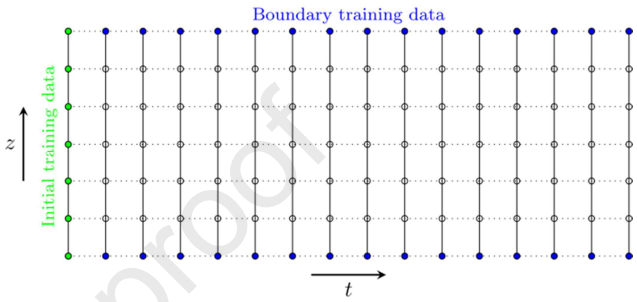


Fig. 4. Initial and boundary condition training data for forward problems. The total number N of training data points (z, t) depends on the spatial and temporal discretization used to obtain the exact solution. The training data are shuffled and divided into batches according to a specified batch size during training.

$$\left.\begin{aligned} p &= 0 \quad (\text{at } \Gamma_t,\ t > 0) \\ \frac{\partial p}{\partial z} &= 0 \quad (\text{at } \Gamma_b,\ t > 0) \end{aligned}\right\} \tag{9}$$

The initial condition is $p = p_o$ for $t = 0$. The analytical solution for the excess pore pressure as a ratio of the initial value is given by (Verruijt, 2013):

$$\frac{p}{p_o} = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{2k-1} \cos\left[(2k-1)\frac{\pi z}{2h}\right] \exp\left[-(2k-1)^2 \frac{\pi^2}{4} \frac{c_v t}{h^2}\right] \quad (t > 0) \tag{10}$$

where h is the so-called drainage path which in this case is equal to the total height of the domain, i.e. $h = H$. The spatial coordinate z can be chosen to have its origin either at the top or at the bottom with positive coordinates in the domain. For the boundary conditions in this case, we have $0 \le z \le H$.

For a numerical example, let us consider the height of the 1D domain to be $H = 1$ m and the coefficient of consolidation of the soil as $c_v = 0.6$ m^2/a. The exact solution based on the analytical solution is obtained for $t = 1$ a using a spatial discretization with $N_z = 100$ and a temporal discretization with $N_t = 100$. As illustrated in Fig. 4, the initial and boundary data, z, t, and p, are extracted from the exact solution as training data. The inputs to the neural network are the values of (z, t) where the model predicts a pore pressure value $\hat{p}$ as an output, which is then used to calculate the training loss.

A neural network with 10 hidden layers and 20 hidden units at each layer is used as the model to be trained. The spatial and temporal derivatives of the predicted pore pressure, as they appear in the constraint equation, are determined at selected collocation points using automatic differentiation in TensorFlow. The collocation points are generated using the Latin hypercube sampling strategy and for example here $N_c = 10{,}000$ collocation points are generated. After the derivatives of $\hat{p}$ with respect to the collocation points are determined, the constraint loss is calculated. A combination of training and constraint losses, as defined in Eq. (6), is minimized

using the Adam optimizer for the desired number of epochs. The learning rate for the optimizer and the batch size used for training are 0.001 and 100, respectively. The model training was performed on an NVIDIA Tesla K80 GPU and for the hyper-parameter combinations used here, the model training time for 10,000 epochs was about 12 min. Models with lower numbers of hidden layers, hidden units and epochs take lesser time for training.

The final model trained using the initial and boundary data, and constrained at the collocation points according to the governing equation, is used to predict the excess pore pressure for spatial and temporal points of the model domain. Prediction based on a trained model takes only a fraction of a second. The results obtained from the analytical solution and model prediction are shown in Fig. 5 in terms of color plots on a two-dimensional grid from the (z, t) data. The color plot is obtained by interpolation of the nearest excess pore pressure values from the actual grid points and is chosen here only for visualization convenience. As can be seen from the color plots of the analytical solution and model prediction, the deep learning model predicts the excess pore pressure values at the interior grid points reasonably well just based on initial and boundary training data. This demonstrates the remarkable accuracy of the physical constraint enforced through automatic differentiation in the deep learning model. A closer comparison of the analytical solution and model prediction is shown for selected time steps in the plot in Fig. 6a. The time steps used for comparison are shown in the top plot in Fig. 5. The results show a remarkably good agreement. The L_2 norm of the relative error is expressed as follows:

$$e = \frac{||\hat{p} - p||_{L_2}}{||p||_{L_2}} \tag{11}$$

In this case, e was found to be 4.462×10^{-3}. A plot of the mean squared errors versus number of epochs is shown in Fig. 6b. The plot shows the total mean squared error as well as the mean squared errors of the training and constraint losses. Mean squared error values in the order of 10^{-5} are obtained near the end of the training.

To show how the performance of the model varies and for comparison purposes, different combinations of number of hidden layers and number of hidden units are analyzed. The results are presented in Table 1. The results show that a shallower network with smaller number of hidden units per layer performs poorly compared to deeper networks in terms of model performance as measured by the relative L_2 error in the predicted pore pressure. The results obtained for a network with 10 hidden layers and 20 hidden units at each layer are considered satisfactory.

Table 1. Relative L_2 error in the model predicted pore pressure for different combinations of number of layers and number of hidden units.

Number of hidden layers	Number of hidden units	Relative L_2 error
2	10	2.379×10^{-2}
	20	1.212×10^{-2}
5	10	1.085×10^{-2}
	20	7.677×10^{-3}
10	10	6.295×10^{-3}
	20	4.642×10^{-3}

4.2. Consolidation with drained top and bottom boundaries

When both the top and bottom boundaries are permeable, excess pore pressure can dissipate through both boundaries. Mathematically, this boundary condition is expressed as

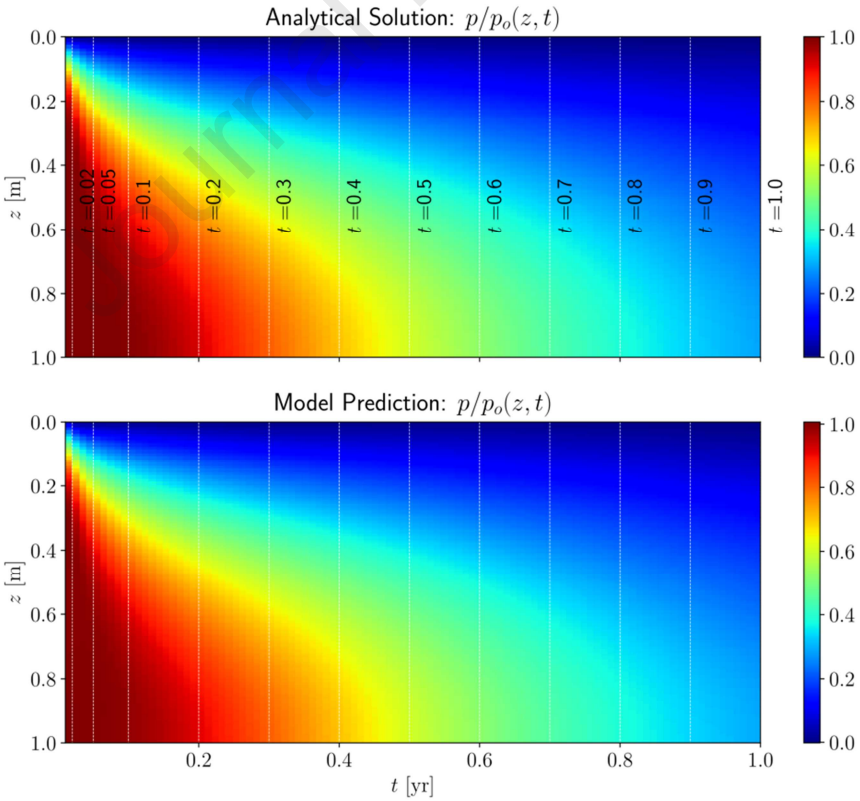


Fig. 5. Results from analytical solution and model prediction in terms of color plots on (z, t) grid for a drained top boundary. The color plots are obtained using interpolation with the nearest available values. The batch size used for training the model is 100.

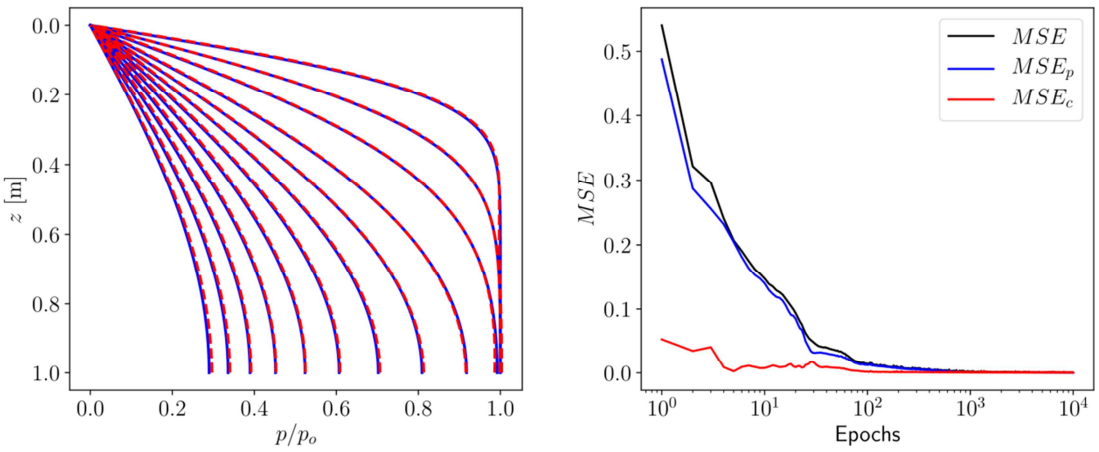


Fig. 6. (a) Comparison of the excess pore pressure ratio between analytical solution (solid blue lines) and model prediction (red dashed lines) for selected time steps, for a drained top boundary. The time steps for comparison are those shown using vertical lines in the top plot in Fig. 5. (b) Mean squared error versus number of epochs used for training.

$$p = 0 \quad (\Gamma_t \cup \Gamma_b,\ t > 0) \tag{12}$$

The analytical solution to Eq. (10) still holds in this case where the drainage path is half of the height of the sample, i.e. $h = H/2$, as the pore fluid is allowed to dissipate through both the top and bottom boundaries. This case is equivalent to stating that there is no pore fluid flow at the center of the sample. Thus, the origin of the spatial coordinate is defined at the mid-height of the domain and we have $-H/2 \le z \le H/2$.

The numerical example in the previous section is considered here again with the same model geometry but with different boundary conditions, i.e. with drained top and bottom boundaries. In addition, the coefficient of consolidation in this case is assumed to be c_v = 0.1 m^2/a for the same model height, considering the faster consolidation as a result of both boundaries being drained. The analytical solution is again obtained using N_z = 100 and N_t = 100. The initial and boundary data are extracted in a similar way as in the previous case for training the model. The hyper-parameters of the neural network model are set to be similar as well. A deep network with 10 layers and 20 hidden units at each layer is used. The number of collocation points and the learning rate for the optimizer (Adam) are 10,000 and 0.001, respectively. A batch size of 100 is used here as well.

Grid color plots comparing the analytical and model predicted solutions are shown in Fig. 7. We again observe a good performance by the deep learning model in predicting the excess pore pressure at the interior grid points. A closer comparison of the excess pore pressure for selected time steps is shown in Fig. 8a. The time steps selected for comparison are shown in the top color plot in Fig. 7. We see a very good agreement between the analytical solution and the deep learning model prediction. The L_2 norm of the relative error between the analytical and predicted solutions in this case is found to be 2.62×10^{-3}. Fig. 8b shows the evolution of the mean squared error with the number of epochs for the total mean squared error as well as the mean squared errors for the training and constraint losses.

5. Inverse problems

The second class of problems considered are inverse problems where we aim to find material/model parameters used in analysis based on a given solution to the problems. For our problem described herein, the aim is to determine the coefficient of consolidation used in an analytical solution given the excess pore pressure as a function of space and time, i.e. $p(z, t)$. The deep learning model is trained based on training data randomly selected from the whole analytical solution, as shown in Fig. 9. The size of the training data may be adjusted as desired but a limited sample size is usually sufficient for training to obtain good model prediction capabilities. Before training, the training data are shuffled and divided into batches based on a specified batch size. The main feature that differentiates inverse models from forward models, in the context of our problem here, is the fact that a trainable variable in addition to the default model variables is required to keep track of the coefficient of consolidation. Thus, a trainable weight is added to the deep learning model to update the coefficient of consolidation after initialization using a random weight. The constraint function based on the governing equation, with the trainable coefficient of consolidation, is continuously optimized together with the training loss. The following examples demonstrate this procedure numerically. It is worthwhile to mention that even though we are dealing with a simple demonstration using a 1D problem here, the approach has important implications in numerical modeling in science and engineering. Some of the potential applications of the methodology that could be mentioned are improved model reproducibility (given a certain numerical solution for a physical problem) and optimization of constitutive model parameters for complicated numerical simulations. Such important problems and applications may be addressed in a later study.

5.1. Consolidation with a drained top boundary

The first example in the forward problems in Section 4 is considered here in an inverse setting with the same geometry and material/model parameters, i.e. a 1D model with a drained top boundary, a model height of H = 1 m and a coefficient of consolidation of c_v = 0.6 m^2/a is analyzed. The analytical solution is again obtained using N_z = 100 and N_t = 100. This implies that the number of (z, t, p) points in the exact solution is equal to 10,000. The neural network architecture is set up to have 10 hidden layers with 20 hidden units at each layer. A random sample of 2000 points is selected from the analytical solution data (with 10,000 available points) for training the neural network.

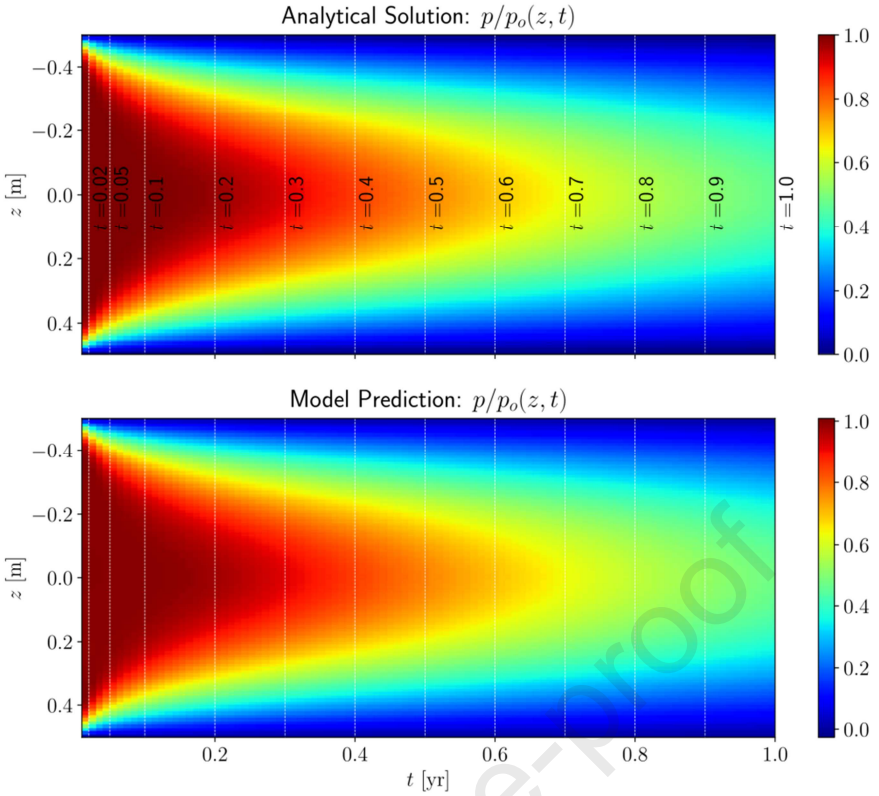


Fig. 7. Results from analytical solution and model prediction in terms of color plots on (z, t) grid for drained top and bottom boundaries. The color plots are obtained using interpolation with the nearest available values. The batch size used for training the model is 100.

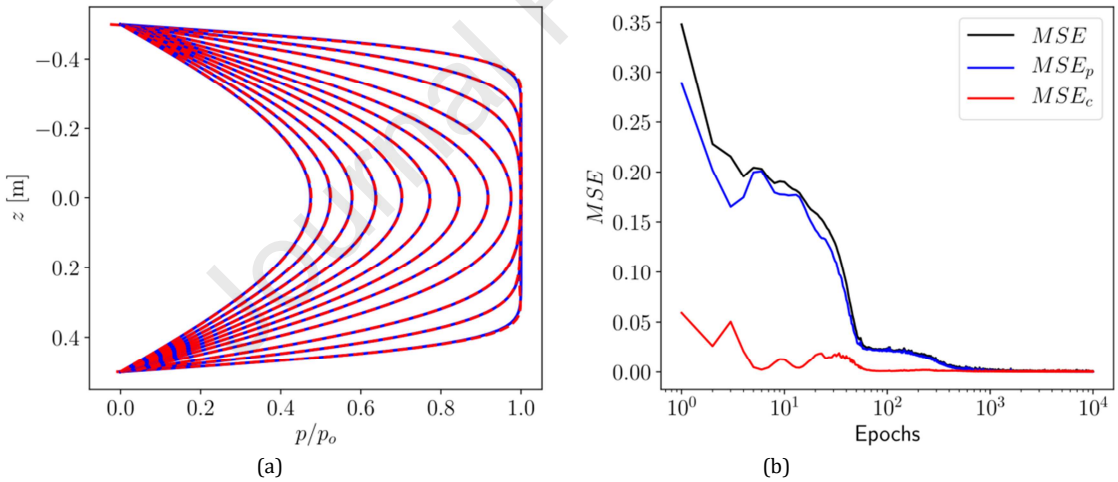


Fig. 8. (a) Comparison of the excess pore pressure ratio between analytical solution (solid blue lines) and model prediction (red dashed lines) for selected time steps, for drained top and bottom boundaries. The time steps for comparison are those shown using vertical lines in the top plot in Fig. 7. (b) Mean squared error versus number of epochs used for training.

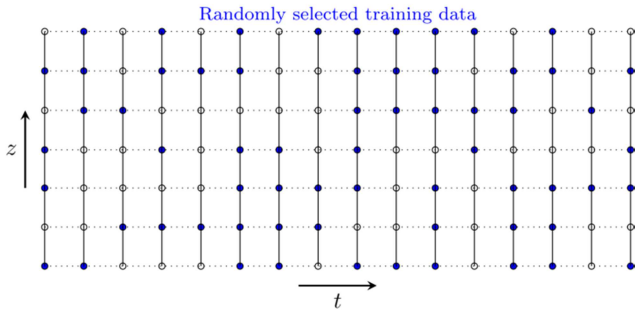


Fig. 9. Randomly selected training data for inverse problems. The total number of training data points (z, t) is chosen depending on the specific problem under consideration. The training data are shuffled and divided into batches according to the specified batch size during training.

The training data are shuffled and divided into batches using a batch size of 200. The trainable weight corresponding to the coefficient of consolidation is initialized as $w_{cv} = 0$, implying an initial coefficient of consolidation of 1 m²/a. Adam optimizer is used to minimize the training and constraint losses with a learning rate of 0.0001. The model training for 10,000 epochs and for the combination of hyper-parameters described here takes approximately 8 min. The training was performed on an NVIDIA Tesla K80 GPU.

The results for inverse analysis of the problem with a drained top boundary are shown in Fig. 10. The randomly selected training data points are shown in the top color plot as white dots. As in the case of the forward problem, the deep learning model predicts the excess pore pressure well from a limited training sample data and prediction

takes only a fraction of a second. This again shows the remarkable performance of the physical constraint obtained using automatic differentiation. Fig. 11a presents the evolution of the predicted coefficient of consolidation as the training progresses, as a function of the number of epochs. The final predicted value of the coefficient of consolidation is c_v = 0.597 m²/a, which is close to the expected value of c_v = 0.6 m²/a with an absolute error of 3 × 10⁻³. Fig. 11b shows the evolution of the mean squared errors as a function of the training epochs.

In a similar way as shown for forward problems, the model performance of different combinations of number of hidden layers and number of hidden units is investigated for inverse problems. The results are presented in Table 2. The results show that a shallower network with smaller number of hidden units per layer performs poorly compared to deeper networks in terms of model performance as measured by the absolute error in the predicted coefficient of consolidation. The result obtained for a network with 10 hidden layers with 20 hidden units at each layer is considered satisfactory.

5.2. Consolidation with drained top and bottom boundaries

We consider again the numerical example in the previous section where both the top and bottom boundaries are drained. The model geometry remains similar but the analytical solution here is obtained using a coefficient of consolidation of c_v = 0.1 m²/a, which we aim to predict using the inverse deep learning model. We have 10,000 data points from the analytical solution based on N_z = 100 and N_t = 100, from where a training sample size of 2000 is randomly selected. The neural network has a similar architecture with 10 hidden layers and 20 hidden units at each layer. The other model hyper-parameters remain similar: the batch size is 200 and Adam optimizer is used with a learning rate of 0.0001.

Table 2. Absolute error in the predicted coefficient of consolidation for different combinations of number of hidden layers and number of hidden units per layer.

Number of hidden layers	Number of hidden units	Absolute error
2	10	1.201×10^{-2}
	20	1.527×10^{-2}
5	10	1.085×10^{-2}
	20	7.72×10^{-3}
10	10	6.46×10^{-3}
	20	3×10^{-3}

The results for inverse analysis for drained top and bottom boundaries are shown in Fig. 12. With a limited training sample, the model predicts the excess pore pressures throughout the grid with a good accuracy. The coefficient of consolidation predicted by the deep learning model is c_v = 0.0994 m²/a, which compared to the expected value of c_v = 0.1 m²/a implies an absolute error of 6 × 10⁻⁴. These results are shown in Fig. 13a, which presents the evolution of the predicted coefficient of consolidation as a function of the number of training epochs. Fig. 13b shows the mean squared errors (total, training and constraint) as a function of the number of training epochs.

6. Summary and conclusions

A deep learning model for 1D consolidation is presented where the governing PDE is used as a constraint in the model. Research on physics constrained neural networks has been gaining traction recently in the machine learning research community and the work presented here adds to that effort. Various application areas where the idea has been applied were briefly reviewed, indicating the potential in diverse science and engineering disciplines.

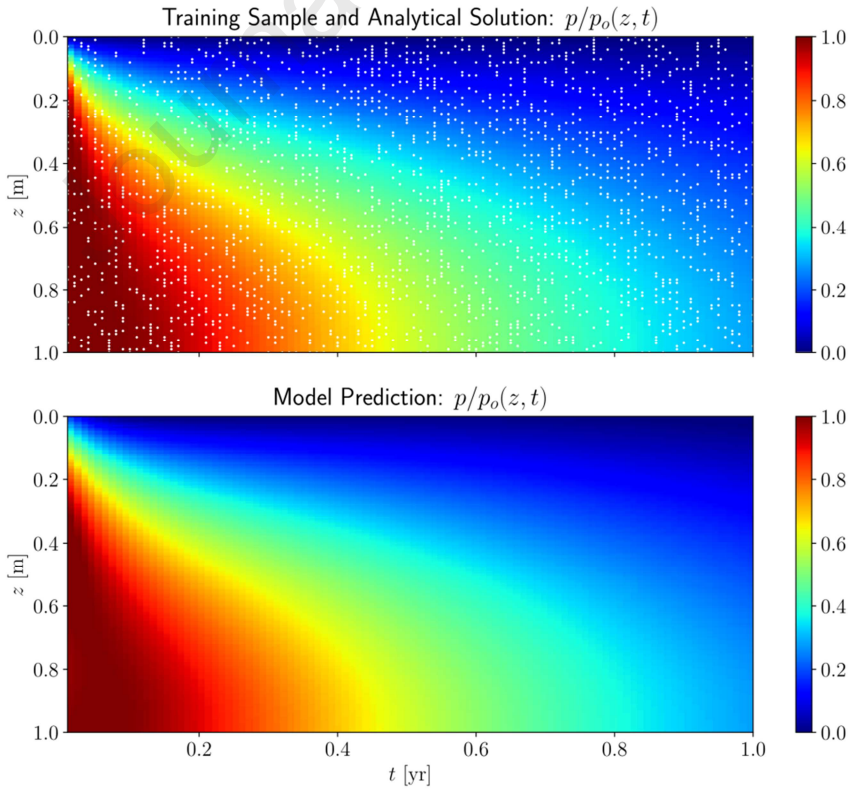
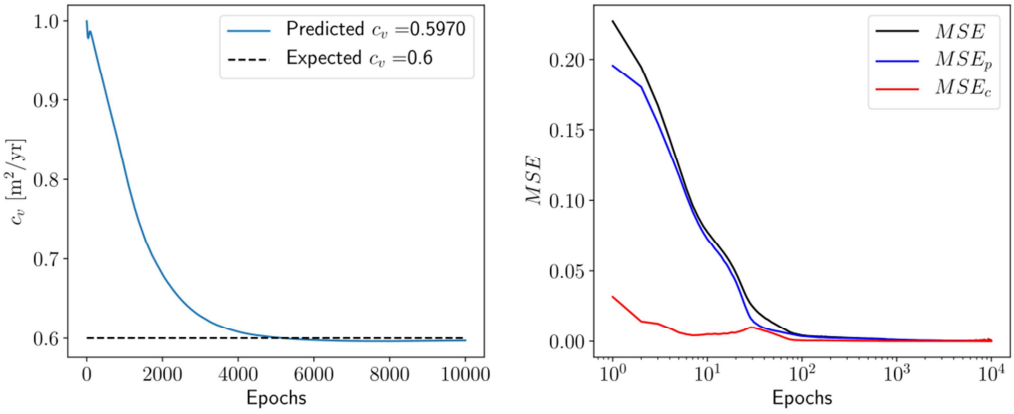


Fig. 10. Inverse analysis results from analytical solution and model prediction in terms of color plots on (z, t) grid for a drained top boundary. The color plots are obtained using interpolation with the nearest available values. The white dots represent the randomly selected training data points and the sample size used is 2000. The batch size used for training is 200.



(a) (b)

Fig. 11. (a) Evolution of the predicted coefficient of consolidation as a function of the number of training epochs, for a drained top boundary; and (b) Mean squared error versus number of epochs used for training.

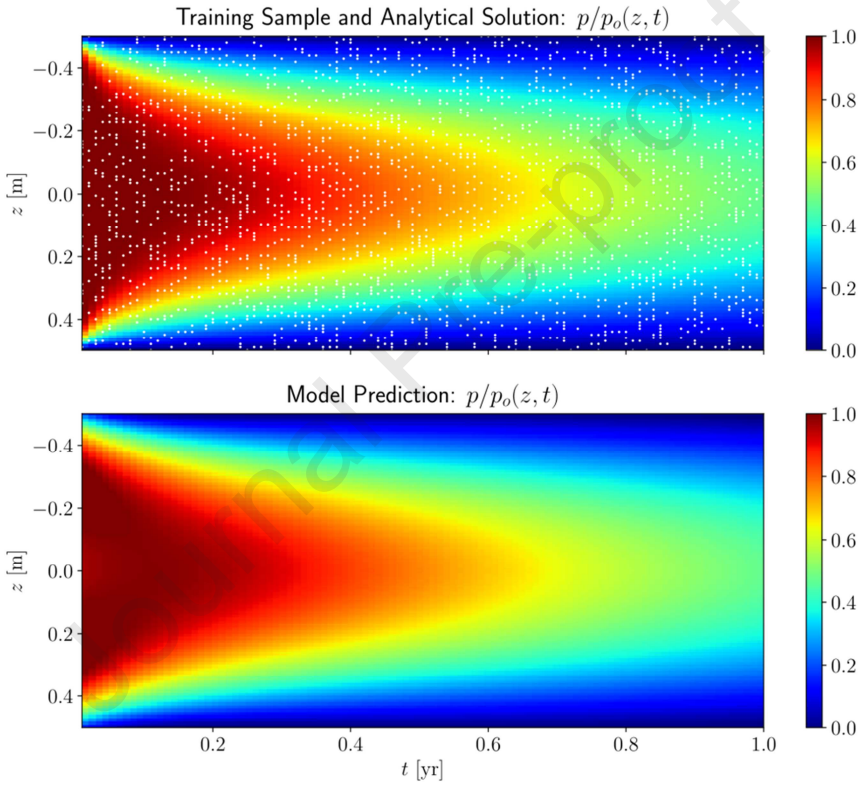
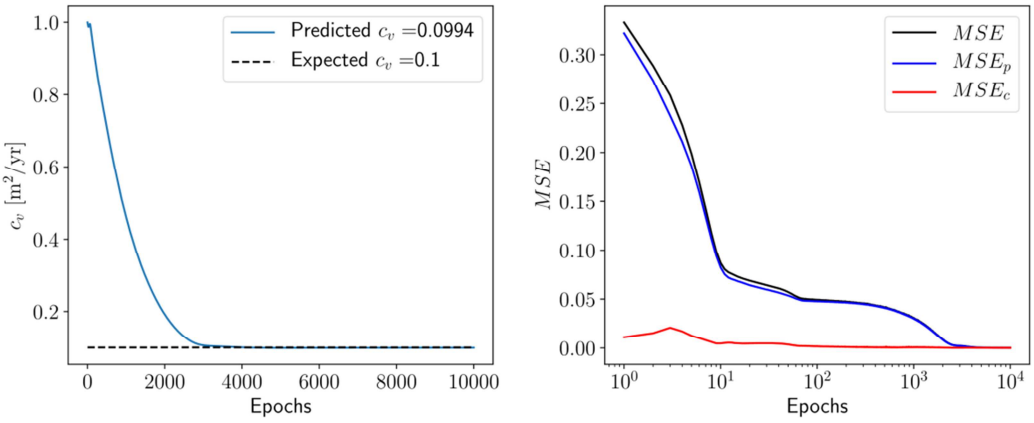


Fig. 12. Inverse analysis results from analytical solution and model prediction in terms of color plots on (z, t) grid for drained top and bottom boundaries. The color plots are obtained using interpolation with the nearest available values. The white dots represent the randomly selected training data points and the sample size used is 2000. The batch size used for training is 200.



(a) (b)

Fig. 13. (a) Evolution of the predicted coefficient of consolidation as a function of the number of training epochs, for drained top and bottom boundaries; and (b) Mean squared error versus number of epochs used for training.

The deep learning model used here is a fully-connected sequential neural network. The neural network is designed to take the spatial and temporal coordinates as inputs and predict the excess pore pressure, which is a function of these parameters. A key feature of the neural network architecture here is automatic differentiation which makes physical constraint of the system possible, using the governing equation of 1D consolidation. The spatial and temporal derivatives of the predicted excess pore pressure are evaluated using the automatic differentiation capability in TensorFlow. Two classes of problems are considered: forward and inverse problems. For forward problems, the derivatives are evaluated at a certain number of collocation points which are generated using the Latin hypercube sampling strategy, with the spatial and temporal bounds of the actual model taken into consideration. This approach made training of the neural network using just the initial and boundary condition data possible with a remarkable degree of accuracy. For inverse problems, a larger size of randomly selected data is used for training the neural network and the derivatives according to the governing equation are evaluated at the training data points. The coefficient of consolidation is used directly for forward problems while it is left as a trainable parameter for inverse problems. For both forward and inverse problems, the total training loss is defined as a combination of the training and constraint losses. The mean squared error for the training loss is evaluated based on the model predicted excess pore pressure and the corresponding exact analytical solution. The mean squared error for the constraint loss is evaluated according to the governing PDE based on the derivatives evaluated using automatic differentiation. A model optimizer (Adam) is used to minimize the total mean squared error during training. For both classes of problems, the model hyper-parameters that are tuned include the number of hidden layers, number of hidden units at each layer, the batch size for training and the learning rate for the optimizer. The potential of the model is demonstrated using Terzaghi's 1D consolidation problem with two variations: with only a drained top boundary and with drained top and bottom boundaries. The model resulted in remarkable prediction accuracy for both classes of problems; the pore pressure throughout the model's spatial and temporal bounds was predicted well for forward problems and the coefficient of consolidation was predicted with a very good accuracy for inverse problems.

While the application presented here is a simple 1D consolidation problem, the implications of such a deep learning model are far greater. Even though 1D consolidation involves only a single governing equation and only one material parameter (the coefficient of consolidation), the concept can be upscaled for problems in geomechanics involving coupled governing differential equations with several material and model parameters. The efficiency demonstrated for forward problems indicates the potential for predicting numerical solutions faster using deep learning models trained using a very limited amount of data but with a very good accuracy because of the physical constraint. This could have a huge potential for digital twins where faster real-time numerical prediction is desirable. The potential for predicting material parameters, as demonstrated using inverse problems, could be very useful in numerical model reproducibility and optimization of constitutive material and model parameters for complex numerical models. A faster and efficient design optimization can be achieved in computational geomechanics by exploiting this advantage of physics-informed deep learning models.

Declaration of competing interest

The author confirms that there are no known conflicts of interest associated with this publication, and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

The research is supported by internal funding from SINTEF through a strategic project focusing on Machine Learning and Digitalization in the infrastructure sector. The author gratefully acknowledges the support provided.

References

Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016a.

Abadi M, Barham P, Chen J, et al. Tensorflow: A system for large-scale machine learning. In: OSDI'16: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. Berkeley, USA: USENIX Association;2016b. p. 265–83.

Al-Aradi A, Correia A, Naiff D, Jardim G, Saporito Y. Solving nonlinear and high-dimensional partial differential equations via deep learning. arXiv preprint arXiv:1811.08782, 2018.

Bar-Sinai Y, Hoyer S, Hickey J, Brenner MP. Learning data-driven discretizations for partial differential equations. Proceedings of the National Academy of Sciences of the United States of America 2019;116(31):15344–9.

Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: A survey. Journal of Machine Learning Research 2017; 18: 5595–637.

Biot MA. General theory of three-dimensional consolidation. Journal of Applied Physics 1941;12(2): 155–64.

Gulli A, Pal S. Deep learning with Keras. Packt Publishing Ltd., 2017.

Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of kirchhoff plate. Computers, Materials & Continua 2019;59(2):433–56.

Huang DZ, Xu K, Farhat C, Darve E. Predictive modeling with learned constitutive laws from indirect observations. 2019. arXiv preprint arXiv:1905.12530.

Jia X, Willard J, Karpatne A, Read JS, Zwart JA, Steinbach M, Kumar V. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. arXiv preprint arXiv:2001.11086, 2020.

Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V. Theory-guided data science: A new paradigm for scientific discovery from data., IEEE Transactions on Knowledge and Data Engineering 2017;29(10):2318–31.

Mao Z, Jagtap AD, Karniadakis GE. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering 2020; 360: 112789. https://doi.org/10.1016/j.cma.2019.112789.

Meng X, Karniadakis GE. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. Journal of Computational Physics 2020;401:109020. https://doi.org/10.1016/j.jcp.2019.109020.

Nascimento RG, Viana FA. Fleet prognosis with physics-informed recurrent neural networks. arXiv preprint arXiv:1901.05512, 2019.

Özbay AG, Laizet S, Tzirakis P, Rizos G, Schuller B. Poisson CNN: Convolutional neural networks for the solution of the Poisson equation with varying meshes and dirichlet boundary conditions. arXiv preprint arXiv:1910.08613, 2019.

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

Journal Pre-proof

707.

Raissi M, Wang Z, Triantafyllou MS, Karniadakis GE. Deep learning of vortex-induced vibrations. Journal of Fluid Mechanics 2019;861:119–37.

Schrefler BA. The finite element method in the deformation and consolidation of porous media. John Wiley & Sons; 1987.

Sirignano J, Spiliopoulos K. DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics 2018;375:1339–64.

Sun L, Gao H, Pan S, Wang JX. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Computer Methods in Applied Mechanics and Engineering 2020;361:112732. https://doi.org/10.1016/j.cma.2019.112732.

Tartakovsky AM, Marrero CO, Perdikaris P, Tartakovsky GD, Barajas-Solano D. Learning parameters and constitutive relationships with physics informed deep neural networks. arXiv preprint arXiv:1808.03398, 2018.

Terzaghi K, Peck RB, Mesri G. Soil mechanics in engineering practice. John Wiley & Sons; 1996.

Tipireddy R, Perdikaris P, Stinis P, Tartakovsky A. A comparative study of physics-informed neural network models for learning unknown dynamics and constitutive relations. 2019. arXiv preprint arXiv:1904.04058.

Verruijt A. Theory and problems of poroelasticity. Delft University of Technology; 2013. p. 71.

Xu K, Darve E. Physics constrained learning for data-driven inverse modeling from sparse observations. arXiv preprint arXiv:2002.10521, 2020.

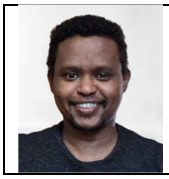
Yang X, Zafar S, Wang JX, Xiao H. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. Physical Review Fluids 2019; 4(3):034602. https://doi.org/10.1103/PhysRevFluids.4.034602.

Yang Y, Perdikaris P. Adversarial uncertainty quantification in physics- informed neural networks. Journal of Computational Physics 2019;394:136–52.

Zhang D, Guo L, Karniadakis GE. Learning in modal space: Solving time-dependent stochastic PDEs using physics-informed neural networks. arXiv preprint arXiv:1905.01205, 2019.

Zhang D, Lu L, Guo L, Karniadakis GE. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. Journal of Computational Physics 2019;397:108850. https://doi.org/10.1016/j.jcp.2019.07.048.

Zheng Q, Zeng L, Karniadakis GE. Physics-informed semantic inpainting: Application to geostatistical modeling. arXiv preprint arXiv:1909.09459, 2019.



Dr. Yared W. Bekele is currently working as a Research Scientist in the Rock and Soil Mechanics Group at SINTEF in Trondheim, Norway. He obtained his MSc and PhD degrees in Geotechnical Engineering from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. His main expertise is advanced theoretical and numerical modeling of thermo-hydro-mechanical ground processes for various applications in civil engineering. His research interests include advanced numerical modeling, visualization, machine learning and digitalization in geotechnical engineering.

Journal Pre-proof