

Applications of physics-informed neural networks for property characterization of complex materials

Sangmin Lee¹, John S. Popovics^{1*}

¹ Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, Illinois 61801, USA

Received: 02 November 2022 / Accepted: 14 December 2022 / Published online: 30 January 2023

© The Author(s) 2022. This article is published with open access and licensed under a Creative Commons Attribution 4.0 International License.

Abstract

The characterization of in-place material properties is important for quality control and condition assessment of the built infrastructure. Although various methods have been developed to characterize structural materials in situ, many suffer limitations and cannot provide complete or desired characterization, especially for inhomogeneous and complex materials such as concrete and rock. Recent advances in machine learning and artificial neural networks (ANN) can help address these limitations. In particular, physics-informed neural networks (PINN) portend notable advantages over traditional physics-based or purely data-driven approaches. PINN is a particular form of ANN, where physics-based equations are embedded within an ANN structure in order to regularize the outputs during the training process. This paper reviews the fundamentals of PINN, notes its differences from traditional ANN, and reviews applications of PINN for selected material characterization tasks. A specific application example is presented where mechanical wave propagation data are used to characterize in-place material properties. Ultrasonic data are obtained from experiments on long rod-shaped mortar and glass samples; PINN is applied to these data to extract inhomogeneous wave velocity data, which can indicate mechanical material property variations with respect to length.

Keywords: Artificial neural networks; Construction material characterization; Physics-informed neural networks; Porous media flow; Wave propagation

1 Introduction

Real-world engineering and science phenomena are complex, yet they can be simplified under some assumptions and represented using appropriate differential equations. The most appropriate and accurate differential equation for a given phenomenon or behavior is called its governing equation. This type of approach represents a “physics-based” model, as opposed to a “data-driven” model that does not consider any underlying physics or mathematics in its solution. Here we present several important representative differential equations that serve as governing equations to simulate processes and behavior in the construction materials field: *Fourier’s law* is used to model heat transfer problems; *Fick’s law* is used to model transport or diffusion of ions through a material; *Darcy’s law* is widely used to model fluid flow in porous media; and *Euler’s laws of motion* are used to describe mechanical responses of materials. Although physics-based models may not perfectly represent a specific real-world problem because of its complexity, they are effective and broadly used because of the flexibility provided by coefficient changes to account for varying environmental or material conditions. When modeling engineering problems, it often is important to understand or predict those coefficients within the governing equations. For example, those coefficients can represent material properties (e.g.,

density, elastic modulus, thermal expansion coefficient, specific heat, permeability, viscosity, etc.) that are important for fundamental research and engineering design, and furthermore may be used as criteria for the evaluation of material integrity. Such material characterization is more difficult with complex, inhomogeneous, or mechanically nonlinear materials such as cement, concrete and rock where properties vary as a function of space, time, stress state or ambient environment; as a result, purely physics-based models often fall short in application to construction materials.

The popularity of data-driven models has increased recently because of the development of machine learning tools, access to increased computational power, and ease of data collection [1]. A key machine learning tool is the artificial neural network (ANN) [2,3], which has universal approximation capability [4–6] and has demonstrated extraordinary results in image classification [7,8], time series regression/prediction [9–11], and natural language processing [12–14] applications. For example, the performance of general object detection has improved rapidly over time as shown in Figure 1 (a). Even for complicated board games (e.g. Go and Chess), the performance of ANNs is already superior to human ability [15]. At the same time, the size and complexity of ANN models have continuously increased: the total number of

*Corresponding author: John S. Popovics, E-mail: johnpop@illinois.edu

parameters of noted ANN models has increased dramatically over time as shown in Figure 1 (b). Despite these developments, ANNs still exhibit multiple drawbacks: (1) they represent “black box” computations with no understanding of internal processes, (2) they require tremendous amount of training data, (3) they demand high computing power requirements, and (4) they show comparatively poor performance for unseen data (i.e., generalization errors or overfitting problems) [16].

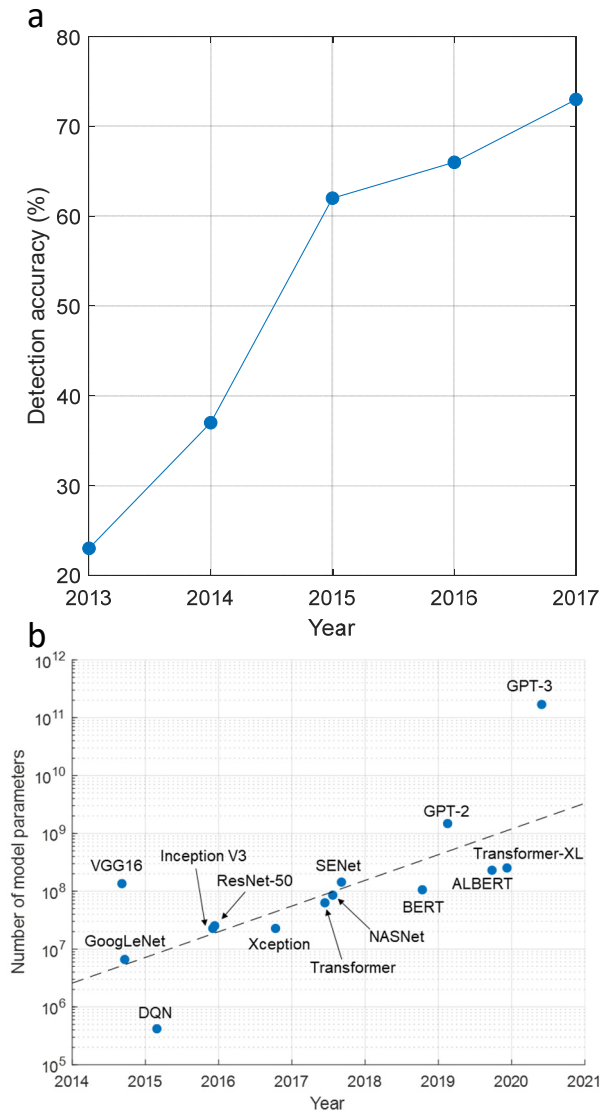


Figure 1. (a) Results from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) over a four-year period. The detection accuracy increased from around 20 % to 70 % over a relatively short period of time. (b) Trends of total number of parameters included in noted ANN models (Data taken from [17]).

In order to address increasingly complex and difficult engineering and science problems, models that combine or fuse physics-based and data driven approaches have been explored more recently. One example of this approach is a “digital twin,” which is a model that represents a real-world problem with additional virtual characteristics [18,19]. There are three main components of a digital twin: (1) physical object, (2) virtual object, and (3) connection between two

objects. A distinct characteristic of the digital twin model is a seamless connection between the physical object and the virtual object. Rather than relying on data or governing equations, it is continuously updated with additional collected data. It can thereby provide a more accurate and versatile representation of the physical object. A more direct combination of physics-based and data-driven models is represented by physics-informed neural networks (PINN). PINN is a relatively new technique among machine learning/neural networks [20], although it is not a new concept [21,22]. PINN is a type of ANN that includes physics-based equations, usually differential equations, as prior knowledge within the training/prediction process, while conventional ANNs do not use prior knowledge or laws of physics about the prediction target. Therefore, parameters of conventional ANNs (i.e., weights and bias) are learned only through training data. PINNs differ from conventional ANNs in that they contain governing equations, in the form of differential equations, and consider compatibility between the equations and the training data. A more detailed explanation of PINN will be provided in Section 2. Although PINNs are starting to be applied to solve challenging problems across a broad range of engineering fields, applications in civil engineering, and construction materials in particular, are limited [23,24].

The aim of this technical letter is to introduce the basic structure and function of artificial neural networks (ANN) and physics-informed neural networks (PINN) (Section 2) and to highlight the potential of PINN to contribute to material characterization tasks by exploring selected PINN-related research work (Section 3) and finally by applying PINN to specific wave propagation-based experimental data (Section 4). In the experimental work, inhomogeneous material property variation is characterized by predicting the wave velocity over space. Wave velocity is often used as a material characterization parameter as it serves as a proxy for material compliance because material Young’s modulus is proportional to the square of the wave velocity.

2 Artificial neural networks and physics-informed neural networks

2.1 Structure and function of artificial neural networks

The basic structure of an ANN is shown in Figure 2. ANNs consist of multiple layers of neurons, where the first layer is called the input layer and the last layer is called the output layer. The layers in between the input and the output layers are called hidden layers. ANNs predict an output(s) based on input data utilizing the hidden layer to do so. Figure 2 (b) shows detail about the function of one neuron within the hidden layer. Each neuron is represented by weights and biases, often called neural network parameters. Once input data are passed to a neuron, the neuron’s weights are multiplied by the input data and summed and the bias is added to the result. After that, the result is fed into an activation function, which is a characteristic feature of ANNs.

This process is expressed as

$$a = \mathcal{A}(\mathbf{w}^T \mathbf{p} + b) = \mathcal{A}(z) \quad (1-1)$$

$$a = \mathcal{A} \left([w_1 \ w_2] \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + b \right) = \mathcal{A}(z) \quad (1-2)$$

where \mathbf{p} is the input vector, \mathbf{w} and b the weight vector and bias of a neuron, respectively, \mathcal{A} the activation function, z the argument of an activation function, and a the output from the neuron. Usually a nonlinear activation function is used, which can take the form of a hyperbolic tangent, rectified linear unit, or sigmoid function; the choice of a specific function depends on the application. An ANN can also be expressed as

$$\hat{\mathbf{u}} = f_{\theta}(\mathbf{p}) \quad (2)$$

where $\hat{\mathbf{u}}$ represents the output values from the ANN (also known as the predicted values with respect to target values) and f_{θ} the ANN expressed as a function. Throughout this paper, the hat symbol indicates predicted values. ANN parameters (θ) are usually learned using gradient descent algorithms (e.g., stochastic gradient descent (SGD), SGD with Momentum, Adam, etc.) [30–32] in an iterative training process with the objective of minimizing the error between the outputs of the ANN and the training outputs. This is presented as

$$\theta_{n+1} = \theta_n - \eta \nabla_{\theta} J(\theta_n) \quad (3)$$

where η is the learning rate, θ_n learnable parameters such as weight and bias in neurons, J the cost function, ∇_{θ} the gradient with respect to θ , and n the iteration number. The cost function (also called the loss function) is the function that quantifies the error between model outputs and expected outputs. The choice of the cost function J depends on the application and structure of the ANN. Typically, mean square error (MSE) and cross-entropy functions are preferred for regression and classification applications, respectively. An example cost function when MSE is used is given by

$$J(\theta) = \frac{1}{N} \sum_i^N [f(\mathbf{p}_i) - f_{\theta}(\mathbf{p}_i)]^2 \quad (4)$$

where N is the total number of training data, $f(\mathbf{p}_i) = \mathbf{u}_i$ the training output data corresponding to the training input data \mathbf{p}_i or latent (or true) function, and $f_{\theta}(\mathbf{p}_i) = \hat{\mathbf{u}}_i$ the output from an ANN model corresponding to the training input data \mathbf{p}_i .

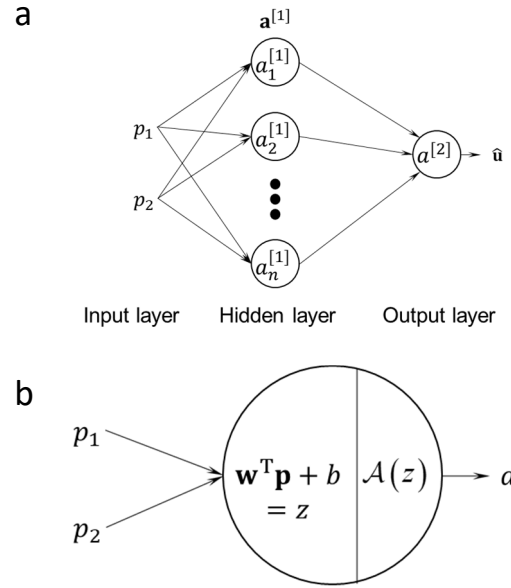


Figure 2. a) A basic ANN structure with one hidden layer and b) detailed computational process within one neuron.

2.2 Physics-informed neural networks

PINN is a form of ANN where physics-based equations (most usually in the form of differential equation) are embedded directly within ANN structure; these equations act as regularization agents during the training process. In other words, the parameters in PINN are tuned to comply with the embedded physics-based governing equations. Because PINNs are closely related to differential equations, we define a differential equation in general terms as

$$\begin{aligned} \mathcal{N}[\mathbf{u}(\mathbf{p})] &= 0, \\ \mathbf{p} &= [\mathbf{p}_x \ p_t], \\ \mathbf{p}_x &\in \Omega, \\ p_t &\in [0, T] \end{aligned} \quad (5)$$

where $\mathcal{N}[\cdot]$ is a differential operator, \mathbf{u} the dependent variable, and \mathbf{p} the independent variable. If \mathbf{p} contains spatial and temporal data then it can be rewritten as $[\mathbf{p}_x \ p_t]$ where \mathbf{p}_x is spatial coordinate data, and p_t temporal data and Ω denotes the spatial domain. Spatial and temporal data are separated intentionally here to distinguish spatial and spatiotemporal problems. Partial differential equations (PDEs) usually contain initial and boundary conditions to ensure the existence and uniqueness of the solution, for example

$$\mathbf{u}(\mathbf{p}; t = 0) = h(\mathbf{p}_x), \quad \mathbf{p}_x \in \Omega \quad (6-1)$$

$$\mathbf{u}(\mathbf{p}; \mathbf{x} = \mathbf{p}_x) = g(\mathbf{p}), \quad \mathbf{p}_x \in \partial\Omega \quad (6-2)$$

$$\mathbf{p}_x \in \Omega \subset \mathbb{R}^d, \quad p_t \in [0, T] \quad (6-3)$$

where t is time, T the upper bound of time-domain data, and d the dimension of the spatial data, and $h(\mathbf{p}_x)$ and $g(\mathbf{p})$ are arbitrary functions. Eqs. (6-1) and (6-2) represent initial condition (IC) and boundary condition (BC), respectively.

The main characteristic of PINN is that physics-based equations are embedded as prior knowledge; a conventional ANN does not use any prior knowledge about data to be trained or the target, so it is a fully data-driven model. In the case of conventional ANNs, all parameters (i.e., weights and bias of each neuron) are randomly initialized and learned (or tuned) to minimize the specific cost function that is defined. Because conventional ANNs are fully data-driven models, they usually require a great amount of training data to adapt to a generalized problem. On the other hand, PINNs require less training data when they are used for inverse problem solving [33,34], and in some cases they do not need any training data when deployed for forward problem solving. Furthermore, PINNs are more robust and generic than traditional ANNs [35]. To achieve a specific objective using an ANN, two different approaches are typically used. The first is to build or design a specific neural network architecture, for example a convolutional neural network [36,37], recurrent neural network [38], or long short-term memory network [11,39]. The second approach is to choose an appropriate loss or cost function. Most of PINN applications take the second approach. The physics-based equations are incorporated in the cost function as

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N_L} w_i \mathcal{L}_i \quad (7)$$

where \mathcal{L} represents loss terms and N_L is the total number of loss terms in the cost function. The number of loss terms varies depending on application. This cost function is different from that shown in Eq. (4) that is used in a conventional ANN. Typical loss terms used in PINN are given by

$$\mathcal{L}_{\mathcal{R}} = \frac{1}{N_{\mathcal{R}}} \sum_{i=1}^{N_{\mathcal{R}}} |\mathcal{R}(\mathbf{p}_i)|^2 \quad (8-1)$$

$$\mathcal{L}_{\mathcal{D}} = \frac{1}{N_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} |f^*(\mathbf{p}_i) - f_{\theta}(\mathbf{p}_i)|^2 \quad (8-2)$$

$$\mathcal{L}_{\mathcal{J}} = \frac{1}{N_{\mathcal{J}}} \sum_{i=1}^{N_{\mathcal{J}}} |h(\mathbf{p}_i) - f_{\theta}(\mathbf{p}_i)|^2 \quad (8-3)$$

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{N_{\mathcal{B}}} \sum_{i=1}^{N_{\mathcal{B}}} |g(\mathbf{p}_i) - f_{\theta}(\mathbf{p}_i)|^2 \quad (8-4)$$

where $\mathcal{L}_{\mathcal{R}}$ is the loss from PDE residual, $\mathcal{L}_{\mathcal{D}}$ the loss between observed data and output values from PINN, $\mathcal{L}_{\mathcal{J}}$ the loss from initial condition, $\mathcal{L}_{\mathcal{B}}$ the loss from boundary condition, and $N_{\mathcal{R}}, N_{\mathcal{D}}, N_{\mathcal{J}}, N_{\mathcal{B}}$ are the number of training data used in each respective loss term. Note that not all PINN applications use the full set of loss terms listed here. The form of the residual (\mathcal{R}) depends on the embedded equations in the PINN model.

For example, if the embedded equation is the 2-D Laplace equation, then the residual is given by

$$\mathcal{R}(\mathbf{p}_i) = \frac{\partial^2}{\partial x^2} f_{\theta}(\mathbf{p}_i) + \frac{\partial^2}{\partial y^2} f_{\theta}(\mathbf{p}_i) \quad (9)$$

The input data (\mathbf{p}_i) used in Eq. (9), also called the collocation points, must satisfy the physics-based equation. To calculate the residual the dependent variable, which is the output of PINN ($f_{\theta}(\mathbf{p})$), is differentiated with respect to independent variables, which are the input of PINN (\mathbf{p}). This differentiation is carried out using Automatic differentiation [40,41]. $f^*(\mathbf{p})$ represents the given target values. Depending on the application, the target values ($f^*(\mathbf{p})$) can be either $f(\mathbf{p})$, which is a true function without error, or $f_{observed}(\mathbf{p}) + \epsilon(\mathbf{p})$, which is the summation of measurement and unknown error.

PINNs can be used to solve a PDE, also known as the *forward problem*, or to predict coefficients within embedded governing equations, also known as the *inverse problem*; more specifically, the inverse problem finds or extracts features or model parameters from observed or measured data. When PINNs are used to solve the forward problem, we consider it to be “semi-supervised” learning because (1) the training process requires both labeled and unlabeled data and (2) a significant portion of the data is unlabeled data where labeled data are readily obtained. When PINNs are used to solve inverse problems, labeled data are required so it is considered “supervised” learning. PINNs offer advantages over conventional numerical techniques for the forward problem: it is mesh-free so there is no discretization error and the challenge of dimensionality is avoidable. Considering that solving inverse problems is not trivial using conventional methods, PINNs may provide effective solution approaches.

3 Material characterization related applications

Although the application of PINN in the construction materials field is not yet common, several application studies have been conducted. In this section, selected papers that are directly or indirectly related to civil engineering or construction materials fields are reviewed. First, fluid flow (or, diffusion) related applications are considered. Tartakovsky et al. [42] studied flow in porous media using 2-D steady-state linear and nonlinear diffusion equations

$$\nabla \cdot [K(\mathbf{x})\nabla p(\mathbf{x})] = 0 \quad (10-1)$$

and

$$\nabla \cdot [K(p)\nabla p(\mathbf{x})] = 0 \quad (10-2)$$

where $K(\mathbf{x})$ and $p(\mathbf{x})$ are hydraulic conductivity and hydraulic head or capillary pressure, respectively. The main objective of Tartakovsky’s study was to predict full K and p fields from individual sparse measurements of K and p . Essentially, Eq. (10-1) describes saturated flow in an inhomogeneous medium while Eq. (10-2) describes unsaturated flow in a homogeneous medium.

To solve and predict the linear diffusion equation (Eq. 10-1), the authors used two separate ANNs to predict K and p individually. Each ANN had 3 hidden layers and 50 neurons per layer. The number of hidden layers and neurons is usually determined empirically, which was the case with Tartakovsky's work. The training data (or reference data) were generated using the finite volume method. In addition to 1024 uniformly distributed collocation points, 250 points of hydraulic conductivity and 100 points of capillary pressure were randomly selected and used for the training process. The relative L_2 errors were 1.7 and 0.5 % for K and p , respectively. The relative L_2 error is defined as $\|\hat{m} - m\| / \|m\|_2$ where m is the quantity of interest and $\|\cdot\|_2$ the two-norm. When they solved the nonlinear diffusion equation (Eq. 10-2), the reference data were generated using a numerical solver (Subsurface Transport Over Multiple Phases (STOMP) [43]). In this case, hydraulic conductivity values were not provided and the relative L_2 error were less than 1 % for both K and p . Additionally, the authors found that the initial parameters (weights and basis) of the ANN model have only minor effects on prediction results. Fuks and Tchepeli [44] solved a forward problem concerning two-phase transport in porous media using the 1-D Buckley-Leverett model [45]. They well estimated water saturation when there was no abrupt discontinuity (i.e., shock behavior) in the data; however, they had difficulty in finding a solution in the presence of a shock. By adding a diffusion term in the model equation, the problem was alleviated. Yu et al. [46] solved the diffusion-reaction equation using gradient-enhanced PINN (gPINN). The diffusion-reaction equation describes substance changes over time and space considering reaction and diffusion. gPINN makes use of the fact that the derivatives of the PDE residual (\mathcal{L}_R) should be zero, which is considered in the cost function (Eq. (7)). The results were compared with the analytic solution and the relative L_2 error for the solution was less than 1%. In addition, an inverse problem was solved using a steady-state diffusion-reaction equation with a space-dependent reaction rate term. The reaction rate function was well predicted and gPINN showed better performance than a conventional PINN.

Now we consider studies related to heat transfer. He et al. [47] studied 1-D heat conduction using PINN. The governing equation that they used is

$$a \frac{\partial T}{\partial t} - b \frac{\partial^2 T}{\partial x^2} - F(x, t) = 0, \quad (11)$$

$$x, t \in [0, 1]$$

where T is the temperature and $F(x, t)$ the heat source function. At first, two forward problems were solved: respectively Dirichlet and Neuman boundary problems. When the coefficients of the governing equation, a and b , were set to 1, the solution from PINN well matched the analytic solution. However, when wood and steel material properties were used, the conventional PINN model produced large errors. After normalizing the time-domain data and appropriately scaling the physics-based equation, reasonably good results were obtained when compared with numerical simulation results (ABAQUS [48]). The authors

claim that data normalization alleviates the gradient vanishing problem. Next, three inverse problems were carried out. The first inverse problem predicted the constant coefficient a in Eq. (11) given b and $F(x, t)$. The second predicted the two constant coefficients a and b in Eq. (11) given $F(x, t)$. The third predicted the source term in Eq. (11), which is e^{x+2t} , given a and b . The authors showed that two techniques help to increase PINN's performance: the skip connection used in ResNet [49] and the adaptive activation function [50]. Cai et al. [51] solved convective heat transfer problems using PINN. In their model, the convection-diffusion equation and incompressible Navier-stokes equation were used. In their study the governing equations were given but only partial boundary conditions were provided, which represents an ill-posed problem. In their solution, sparse measurements of temperature and velocity were provided to a PINN to predict temperature, velocity, and pressure fields. Rad et al. [52] used PINN to solve an alloy solidification modeling problem. In their paper, the model consists of the energy conservation equation, solute conservation equation, and thermodynamic relations. The outputs of the PINN model were temperature, solid fraction, and solute concentration. The temperature predictions were compared with the results from an open-source computational fluid dynamics software platform (OpenFOAM). The solid fraction and solute concentration predictions were compared with the exact analytical solutions, where the PINN predictions matched the analytical solutions well. The authors also analyzed the optimal range of initial learning rate for the Adam optimizer for this problem.

Finally, we consider solid mechanics related materials studies. Haghghat et al. [53] predicted Lamé parameters (typically represented as λ and μ) for a 2-D homogeneous elastic plane-strain problem, which represents an inverse problem. The training data were obtained from an analytic solution where displacement, stress, and force were used as training data sets. They applied PINN to solve this nonlinear problem based on the von Mises elastoplastic constitutive model, where yield stress and Lamé parameters were predicted. Bharadwaja et al. [54] solved a 2-D inhomogeneous and linear elasticity problem. They considered an inhomogeneous material where either internal voids or high elastic modulus inclusions were included within a base solid material. The PINN framework 'Modulus,' developed by Nvidia, was used to create PINN models. As Tartakovsky et al. [42] did, two separate ANN models were used to predict displacement and stress separately. The stress predictions from the PINN model matched well with results obtained by a commercial finite element software platform.

4 Specific application related to material characterization using mechanical wave propagation

In this section, we demonstrate how PINN can be used to characterize material properties using mechanical wave propagation data.

The well-known linear wave equation, which governs mechanical wave propagation in solids, is

$$\frac{\partial^2 u}{\partial t^2} = c_0^2(x) \frac{\partial^2 u}{\partial x^2} \quad (12)$$

where u is displacement, x space, and c_0 the propagating wave velocity. Here the wave velocity is a function of space and is proportional to square root of Young's modulus of the material. Our goal is to characterize wave velocity variation along the length of inhomogeneous samples. To enable this, wave propagation data were collected over the length of two different samples. Because wave velocity is proportional to the elastic modulus of the material, prediction of the wave velocity over space can be used to evaluate spatial variations in material integrity. In this work, PINN was used as an inverse problem solver. Considering that most PINN papers published to date consider data obtained by numerical analysis or analytic solution, the results shown here serve to show the potential of PINN for handling experimental data.

4.1 Experimental setup and sample description

Two cylindrical rod samples were used in the experiments: one is composed of borosilicate glass and the other of portland cement mortar. Both samples are 25.4 mm in diameter and 147 cm in length. The glass sample is pristine and ostensibly homogeneous without any obvious or known damage. The mortar sample consists of two distinct sections along its length: a "strong" matrix design with a water-to-cement (w/c) ratio of 0.5 and a "weak" design with a w/c of 0.6, with a distinct boundary between the two mixtures that exists at about one-third length of the sample. For both mortar mixtures, the cement-to-sand ratio is 1/3 by mass. Figure 3 shows the experimental setup from which wave propagation data were collected along each sample's length. To generate mechanical (ultrasonic) waves in the samples, piezoelectric (PZT) discs are attached to the flat end of each sample. The waves generated by the PZT discs propagate along the length of the samples and are detected using an air-coupled transducer (ACT) positioned above the outer surface; ACTs offer the advantage of not requiring physical contact with the samples and the ability to collect large amounts of consistent data quickly. A 1-D linear lead screw actuator was used for the motion stage to control the position of the ACT along the sample length. For both samples, ultrasonic signal data were measured every 5 mm along the length of the samples using a sampling rate of 12.5 MS/s. The distance (lift-off) between ACT and the samples was set and maintained at 65 mm.

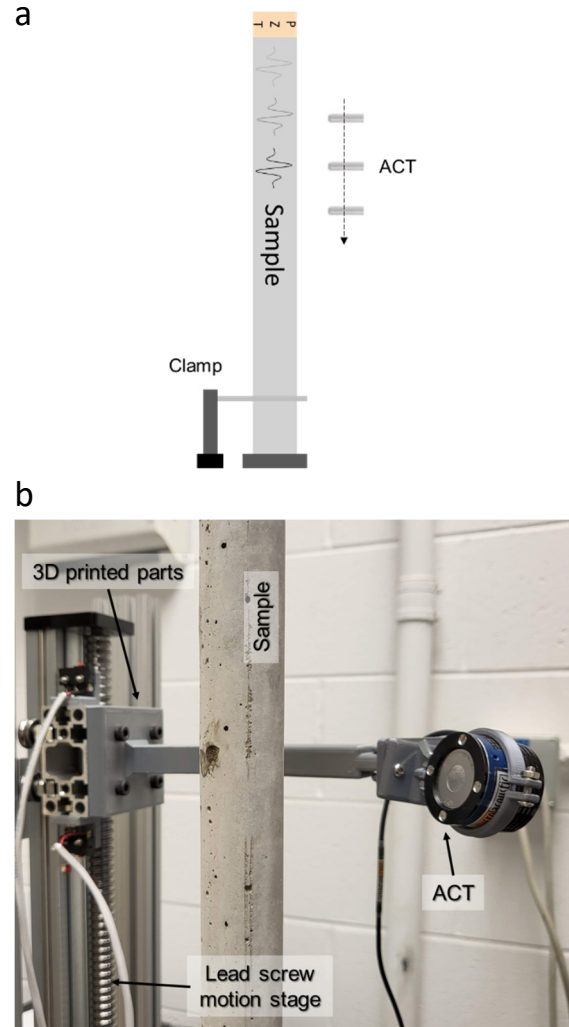


Figure 3. The mechanical wave measurement system: (a) a schematic image that shows the entire measurement system and (b) detailed view of the measurement system with the mortar sample.

4.2 Detail of PINN model

Figure 4 shows the PINN architecture used in this work. The first layer is the input layer, which accepts two signal features: space and time. It is common to normalize data to improve convergence speed when ANNs are used. One drawback, however, of normalizing data before feeding into a model is that the physical meaning of the input data is eliminated. Therefore, rather than using normalized input data, a normalization layer is added into the architecture of the ANN so that the model can retrieve original data values through the process of back-propagation. The normalization layer calculates the mean value and standard deviation with regard to the input data before the training process begins. The calculated values are stored in the normalization layer and they are used to convert the input data set to have a mean of 0 and a standard deviation of 1, which is z-score normalization. After that, two separate 1-D sets of scalar input data are concatenated into one vector in the subsequent concatenate layer. Then the normalized data are passed into the fully connected layer network. As shown in Figure 4, the network comprises a 'Main network' and other

subnetworks. The main network is used to predict output values (displacement) and other subnetworks are used to predict coefficients of the embedded differential equation; one subnetwork is used to predict wave velocity (c_0), which is the principal output value of interest. This latter subnetwork considers spatial coordinate information so that the prediction of the wave velocity is a function of the spatial coordinate. Note that the true output values of the main network (displacement) are only provided during the training process, and those of the subnetwork (wave velocity) are not provided. A total of 4 hidden layers, each with 40 neurons, comprise the main network and 3 hidden layers, each with 20 neurons, comprise the subnetwork. The output value (\hat{u}) from the main network is double-differentiated with respect to the input data (x, t). Finally, the differentiated values ($\partial^2 \hat{u} / \partial t^2, \partial^2 \hat{u} / \partial x^2$) are multiplied with corresponding coefficients to calculate the residual of the implemented physics-based equation. That residual is defined as

$$\mathcal{R}(x, t) = \frac{\partial^2 \hat{u}}{\partial t^2} - \hat{c}_0^2(x) \frac{\partial^2 \hat{u}}{\partial x^2} \quad (13)$$

and the cost function used in this work is defined as

$$J(\Theta) = \frac{1}{N} \sum_{i=1}^N |f^*(\mathbf{p}_i) - f_\theta(\mathbf{p}_i)|^2 + \frac{1}{N} \sum_{i=1}^N |\mathcal{R}(\mathbf{p}_i)|^2 \quad (14)$$

where $\mathbf{p} = [x \ t]$. The Adam optimizer is used in the training process using an initial learning rate of $5e^{-4}$. The objective of using this PINN model is to predict the wave velocity using the embedded equation and training output, which is the displacement. The predicted wave velocity can be related to Young's modulus (E) and density (ρ) using the equation $V_{1D} = \sqrt{E/\rho}$ to monitor material compliance

changes along the long rod-shaped samples where V_{1D} is the bar velocity and is considered to represent c_0 .

4.3 Experimental results and PINN predictions

Figures 5 (a) and (b) show mechanical wave propagation measurement data for the glass and mortar samples, respectively. Signal data were measured every 5 mm along the length of the samples. All of the multiple wave measurements are presented at once using the time-space domain, where the amplitude of the received signal is normalized to have values from -1 to 1 as represented by the color scale. The signal amplitude from the ACT sensor is directly proportional to the measured acoustic pressure. When absolute values of displacement are not needed, the small-strain, and harmonic wave propagation assumption enables us to consider that the signal amplitude is proportional to relative surface displacement. A 14-cycle tone burst signal with a 90 kHz center frequency was used as the excitation source for the glass sample, and 75 kHz center frequency for the mortar sample. For both cases, the excitation voltage to the PZT sender was 40 V. In both material samples, coherent ballistic propagating wave fronts can be observed without noticeable dispersion; thus, it is reasonable to apply the simple 1-D wave equation here. In order to reduce the number of training data, a time window (250~350 μ s for the glass data and 300~450 μ s for the mortar sample) was applied to the original data, and only that part of the signal within the window was used. 20 % of the data were randomly selected to be set aside and thus were not used for PINN training; these data were later used to test the performance of PINN with regard to unknown spaces. A total of 61,299 and 91,924 data points were used for the glass and mortar samples, respectively. Although the amount of data may seem large, one training data set consists of only three scalar quantities, so the absolute data set sizes are relatively small at 500 and 740 kB, respectively.

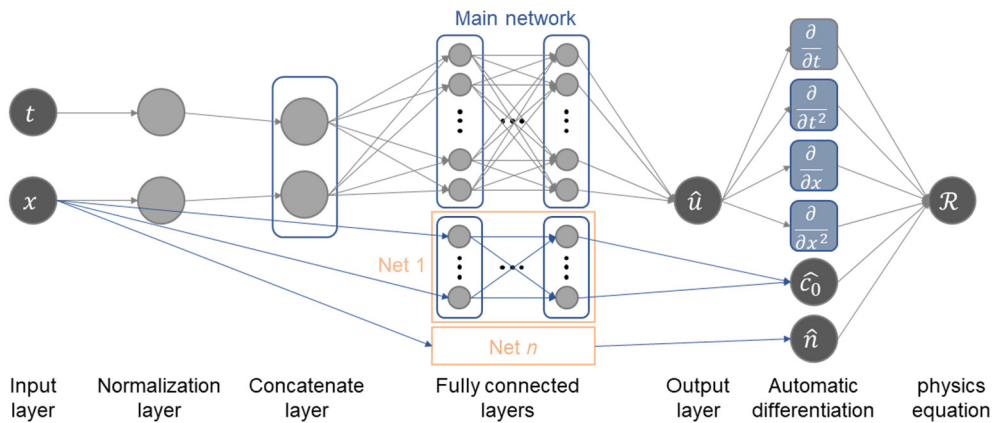


Figure 4. The architecture of the physics-informed neural network considering the inhomogeneous material case. The model consists of multiple fully connected layers (main network and subnetworks). The main network is used to predict the output values, and the subnetworks (indicated as 'Net n') are used to predict coefficients of the embedded wave equation. The number of subnetworks depends on the number of coefficients predicted.

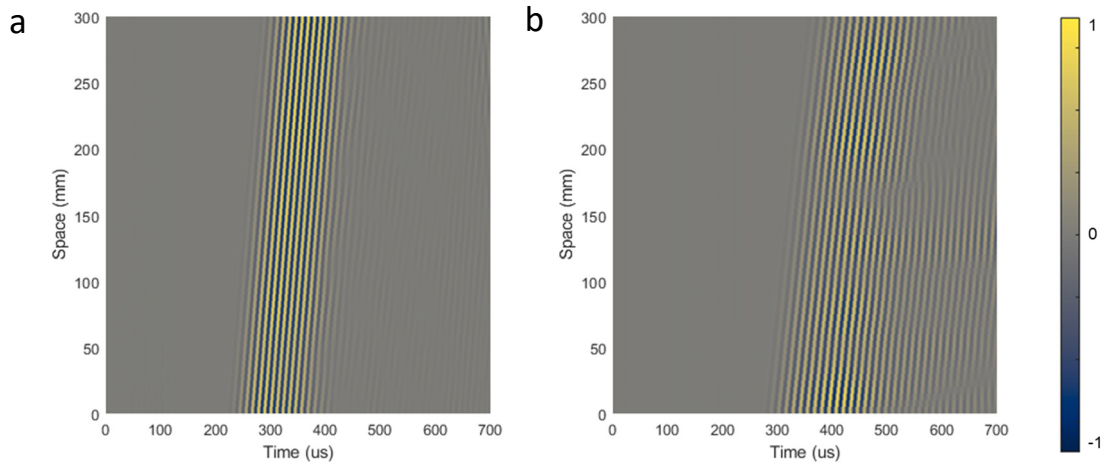


Figure 5. Mechanical wave propagation measurement results presented in the time-space domain: (a) glass sample and (b) mortar sample. The measured signals using the ACT are normalized to have values from -1 to 1 where the amplitude of the signal, which represents a relative displacement, is represented by the color scale.

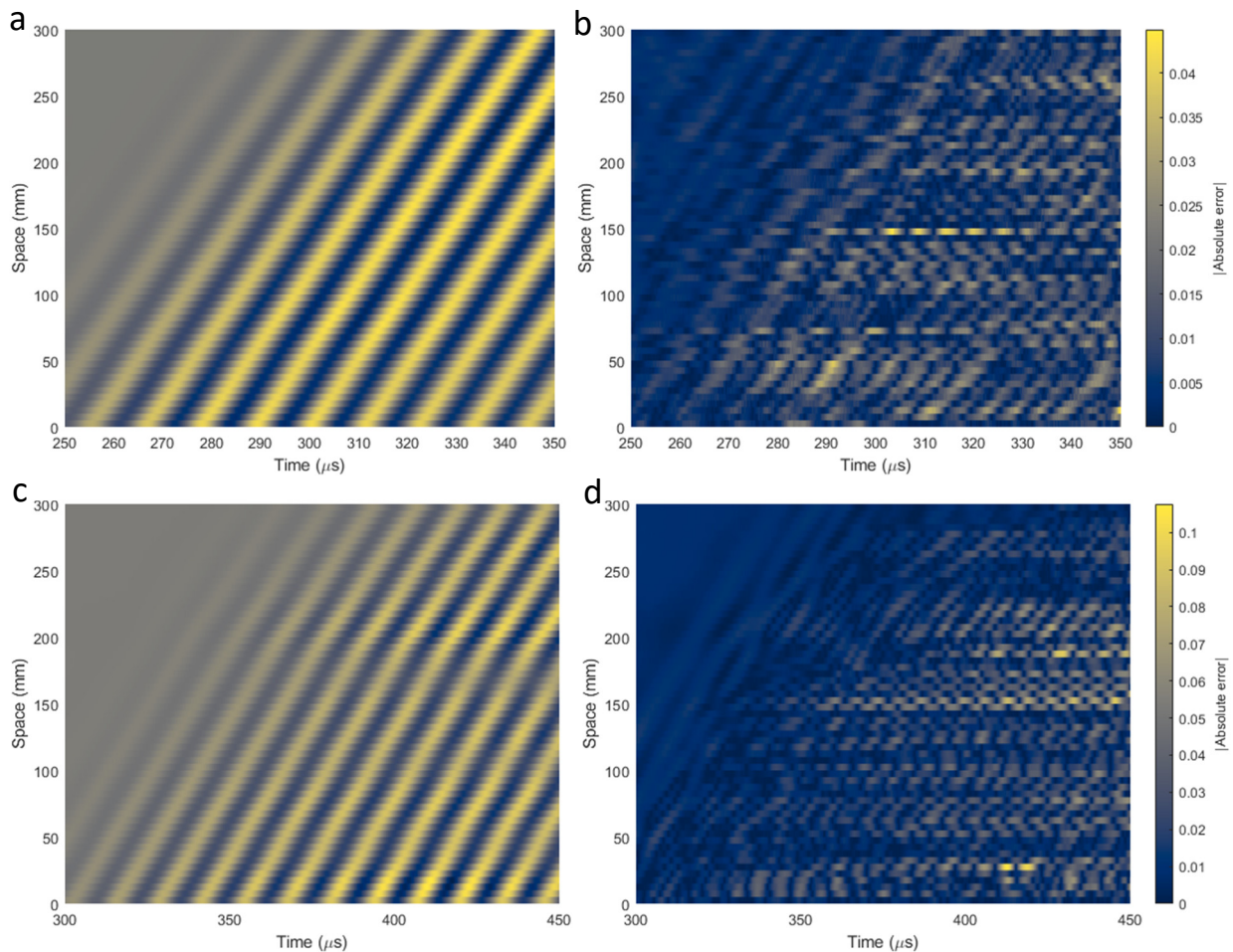


Figure 6. PINN prediction results: (a) relative displacement prediction for the glass sample, (b) absolute error between PINN results and measurement data for the glass sample, (c) relative displacement prediction for the mortar sample, and (d) absolute error between PINN results and measurement data for the mortar sample.

Figures 6 (a) and (c) show displacement prediction results in the time-space domain using the developed PINN for the glass and mortar samples, respectively. The amplitude of the time-space domain signal data are represented in terms of normalized signal amplitude, while the amplitude of the predicted displacement error for the glass and mortar samples, shown in Figures 6 (b) and (d) respectively, are represented by the absolute error of normalized signal amplitude. The PINN provides overall displacement prediction errors of 1.85 and 4.98 % for the glass and mortar samples, respectively, as measured by the relative L^2 norm.

The wave velocity prediction results over length for the glass and mortar samples are shown in Figures 7 (a) and (b), respectively. The green solid line represents the experimentally obtained wave velocity profile calculated by connecting multiple zero-crossing points of the wave fronts to determine time of flight for specific, known measurement positions over the length of the sample; we consider this to be the true wave velocity profile although this is not fully correct because of inherent measurement error and the simplifying linear wave propagation approximation. The black solid line connecting black and red points indicates wave velocity prediction results from the PINN model. As mentioned earlier, 20 % of data were retained as test data (red points) and the remaining 80 % of the data (black points) were used for training the PINN. Although no significant wave velocity prediction error was observed at the red points for both tests, improved prediction performance is expected if more training points are used.

In the case of the glass sample, the wave velocity shows variation along its length where the mean value for the “true” wave velocity profile obtained from measurement data is 5163.3 m/s, while that from PINN is 5146.4 m/s, resulting in an overall relative error of -0.33 % for the PINN model. For the mortar sample, a notable change (reduction) in the wave velocity is observed to start at around 180 mm from the first measurement point in Figure 7 (b), which matches the expected location of the boundary of the two component sections (“strong” and “weak”) within the mortar sample. Based on the measurement results, the mean value of the experimentally obtained wave velocity in the strong section is 3967.1 m/s and that in the weak section is 3564.2 m/s, which confirms expected behavior. The PINN prediction gives mean values of 3925.1 m/s and 3421.0 m/s, respectively. This results in relative PINN prediction velocity errors of -1 and -4 % for the strong and weak sections, respectively.

The developed PINN model well predicts wave velocity over space based on the measured mechanical wave data. This wave velocity information can be used to calculate variations in Young’s modulus along the length of the bar knowing $V_{1D} = \sqrt{E/\rho}$. Once the PINN model is trained, the wave velocity, and thus E , can be predicted anywhere within the range of the training data if density is known. Compared to traditional nondestructive methods, such as ultrasonic pulse velocity that requires point by point measurements to characterize spatially varying material properties, PINN provides inhomogeneous varying material properties efficiently.

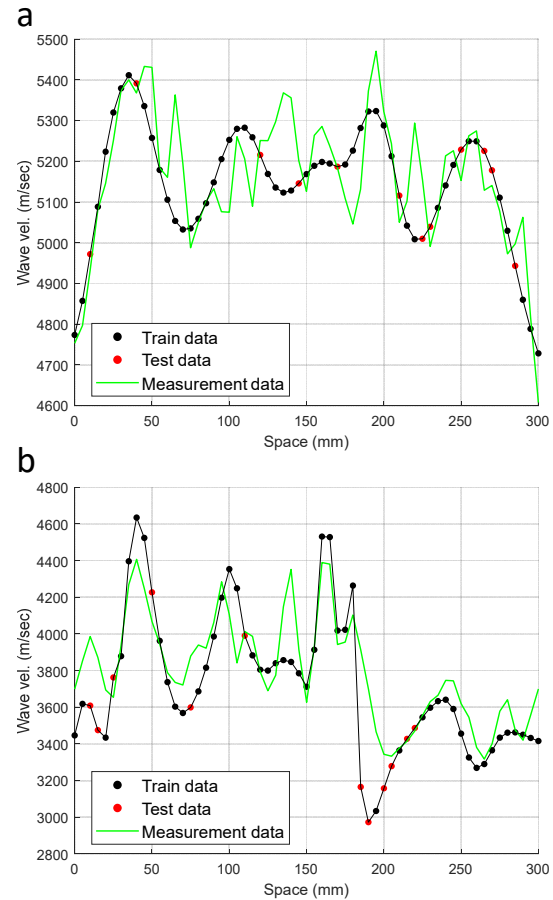


Figure 7. Wave velocity from PINN prediction (black line and points) and measurements (green line): (a) glass sample and (b) mortar sample.

5 Summary, Conclusions, and Perspectives

In this paper the basic structure and function of artificial neural networks (ANN) and physics-informed neural networks (PINN) are introduced, and the potential for PINN to contribute to material characterization tasks are considered. We suggest that PINNs show potential for effective application to a broad range of problems related to construction materials issues, in particular flow in porous media and characterization of inhomogeneous media. A specific application of PINN analysis for mechanical wave propagation in homogeneous and inhomogeneous media is presented, where material wave velocity profile as a function of space was predicted. We conclude that this particular application of PINN is useful in that it can be used, using one measurement set, to characterize inhomogeneous material properties as a function of space, and thus may be useful to evaluate material integrity variations knowing the connection between wave velocity and material properties. Although only a simple 1-D case was considered in this work, the concept can be readily expanded to 2-D or 3-D cases, which suggests that this approach can be applied to more realistic structural elements, such as beams or plates, to characterize in-place material properties. Furthermore, other meaningful wave propagation characteristics, such as wave attenuation coefficient or wave nonlinear parameters can be extracted through appropriate modifications to the PINN structure.

These additional characteristics could be used to characterize sophisticated material properties or to diagnose early damage.

It is clear that the development and deployment of PINN to engineering and science problems is a rapidly emerging field, and one that shows great potential. However, PINN is not a universal tool and one can, and should, question whether PINN can completely replace existing analytical, numerical, or purely physics-based methods. PINNs exhibit many aspects that need improvement, for example choice of hyperparameters, spectral bias, balancing between loss terms, computational cost, etc. At the same time it should be noted that PINN is in its infancy; as future technological and computing developments enable rapid collection of large data sets, increase in computing power, and rapid growth and acceptance of the ANN algorithm, we expect PINN will emerge as a helpful and commonly applied tool to solve difficult problems related to construction materials research.

Authorship statement (CRediT)

Sangmin Lee: Conceptualization, Methodology, Software, Investigation, Writing - original draft

John S. Popovics: Conceptualization, Supervision, Project administration, Writing - review and editing

References

- [1] The rise of data-driven modelling. *Nature Reviews Physics* (2021) 3 (6): 383-383. <https://doi.org/10.1038/s42254-021-00336-z>
- [2] McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* (1943) 5 (4): 115-133. <https://doi.org/10.1007/BF02478259>
- [3] Ivakhnenko, A.G. and Lapa, V.G. Cybernetic predicting devices. (1966).
- [4] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* (1989) 2 (4): 303-314. <https://doi.org/10.1007/BF02551274>
- [5] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks* (1989) 2 (5): 359-366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [6] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The Expressive Power of Neural Networks: A View from the Width. *Advances in Neural Information Processing Systems*, 2017.
- [7] Guo, T., Dong, J., Li, H., and Gao, Y. Simple convolutional neural network on image classification. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017. <https://doi.org/10.1109/ICBDA.2017.8078730>
- [8] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* (2017) 60 (6): 84-90. <https://doi.org/10.1145/3065386>
- [9] Azadeh, A., Ghaderi, S., and Sohrabkhani, S. Forecasting electrical consumption by integration of neural network, time series and ANOVA. *Applied Mathematics and Computation* (2007) 186 (2): 1753-1761. <https://doi.org/10.1016/j.amc.2006.08.094>
- [10] Yolcu, U., Egrioglu, E., and Aladag, C. H. A new linear & nonlinear artificial neural network model for time series forecasting. *Decision support systems* (2013) 54 (3): 1340-1347. <https://doi.org/10.1016/j.dss.2012.12.006>
- [11] Fischer, T. and Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research* (2018) 270 (2): 654-669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- [12] Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*, 2008. <https://doi.org/10.1145/1390156.1390177>
- [13] Hirschberg, J. and Manning, C. D. Advances in natural language processing. *Science* (2015) 349 (6245): 261-266. <https://doi.org/10.1126/science.aaa8685>
- [14] Goldberg, Y. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* (2016) 57 345-420. <https://doi.org/10.1613/jair.4992>
- [15] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* (2018) 362 (6419): 1140-1144. <https://doi.org/10.1126/science.aar6404>
- [16] Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.
- [17] Bernstein, L., Sludde, A., Hamerly, R., Sze, V., Emer, J., and Englund, D. Freely scalable and reconfigurable optical hardware for deep learning. *Scientific reports* (2021) 11 (1): 1-12. <https://doi.org/10.1038/s41598-021-82543-3>
- [18] Grieves, M. Digital twin: manufacturing excellence through virtual factory replication. White paper (2015)
- [19] Grieves, M. and Vickers, J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*; Springer, (2017) 85-113. https://doi.org/10.1007/978-3-319-38756-7_4
- [20] Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* (2019) 378 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [21] Lagaris, I. E., Likas, A., and Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* (1998) 9 (5): 987-1000. <https://doi.org/10.1109/72.712178>
- [22] Psichogios, D. C. and Ungar, L. H. A hybrid neural network-first principles approach to process modeling. *AIChE Journal* (1992) 38 (10): 1499-1511. <https://doi.org/10.1002/aic.690381003>
- [23] Bughin, J., Hazan, E., Ramaswamy, S., Chui, M., Allas, T., Dahlstrom, P., Henke, N., and Trench, M. Artificial intelligence: the next digital frontier? McKinsey Global Institute, 2017.
- [24] State of the Nation 2017: Digital transformation. Institution of Civil Engineers, 2017.
- [25] Basic theory. In *Fluid Flow in Porous Media*; (2020) 47-67. https://doi.org/10.1142/9789811219535_0002
- [26] Schweizer, B. Darcy's law and groundwater flow modelling. (2015)
- [27] Avrami, M. Kinetics of phase change. I General theory. *The Journal of chemical physics* (1939) 7 (12): 1103-1112. <https://doi.org/10.1063/1.1750380>
- [28] Avrami, M. Kinetics of phase change. II transformation-time relations for random distribution of nuclei. *The Journal of chemical physics* (1940) 8 (2): 212-224. <https://doi.org/10.1063/1.1750631>
- [29] Thomas, J. J. A New Approach to Modeling the Nucleation and Growth Kinetics of Tricalcium Silicate Hydration. *Journal of the American Ceramic Society* (2007) 90 (10): 3282-3288. <https://doi.org/10.1111/j.1551-2916.2007.01858.x>
- [30] Kiefer, J. and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* (1952) 462-466. <https://doi.org/10.1214/aoms/117729392>
- [31] Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics* (1964) 4 (5): 1-17. [https://doi.org/10.1016/0041-5553\(64\)90137-5](https://doi.org/10.1016/0041-5553(64)90137-5)
- [32] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv:1412.6980 (2017)
- [33] Zhao, J. Discovering Phase Field Models from Image Data with the Pseudo-Spectral Physics Informed Neural Networks. *Communications on Applied Mathematics and Computation* (2021) 3 (2): 357-369. <https://doi.org/10.1007/s42967-020-00105-2>
- [34] Ma, Y., Xu, X., Yan, S., and Ren, Z. A Preliminary Study on the Resolution of Electro-Thermal Multi-Physics Coupling Problem Using Physics-Informed Neural Network (PINN). *Algorithms* (2022) 15 (2): 53. <https://doi.org/10.3390/a15020053>
- [35] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics* (2021) 3 (6): 422-440. <https://doi.org/10.1038/s42254-021-00314-5>

- [36] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* (1980) 36 (4): 193-202. <https://doi.org/10.1007/BF00344251>
- [37] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* (1989) 1 (4): 541-551. <https://doi.org/10.1162/neco.1989.1.4.541>
- [38] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature* (1986) 323 (6088): 533-536. <https://doi.org/10.1038/323533a0>
- [39] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation* (1997) 9 (8): 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [40] Linnainmaa, S. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics* (1976) 16 (2): 146-160. <https://doi.org/10.1007/BF01931367>
- [41] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of machine learning research* (2018) 18 1-43.
- [42] Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., and Barajas-Solano, D. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research* (2020) 56 (5): e2019WR026731. <https://doi.org/10.1029/2019WR026731>
- [43] White, M., Oostrom, M., and Lenhard, R. Modeling fluid flow and transport in variably saturated porous media with the STOMP simulator. 1. Nonvolatile three-phase model description. *Advances in Water Resources* (1995) 18 (6): 353-364. [https://doi.org/10.1016/0309-1708\(95\)00018-E](https://doi.org/10.1016/0309-1708(95)00018-E)
- [44] Fuks, O. and Tchelepi, H. A. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing* (2020) 1 (1): <https://doi.org/10.1615/JMachLearnModelComput.2020033905>
- [45] Buckley, S. E. and Leverett, Mc. Mechanism of fluid displacement in sands. *Transactions of the AIME* (1942) 146 (01): 107-116. <https://doi.org/10.2118/942107-G>
- [46] Yu, J., Lu, L., Meng, X., and Karniadakis, G. E. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering* (2022) 393 114823. <https://doi.org/10.1016/j.cma.2022.114823>
- [47] He, Z., Ni, F., Wang, W., and Zhang, J. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. *Materials Today Communications* (2021) 28 102719. <https://doi.org/10.1016/j.mtcomm.2021.102719>
- [48] ABAQUS. Dassault Systèmes, 2014.
- [49] He, K., Zhang, X., Ren, S., and Sun, J. In Deep residual learning for image recognition, 2016. <https://doi.org/10.1109/CVPR.2016.90>
- [50] Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics* (2020) 404 109136. <https://doi.org/10.1016/j.jcp.2019.109136>
- [51] Cai, S., Wang, Z., Wang, S., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer* (2021) 143 (6): <https://doi.org/10.1115/1.4050542>
- [52] Rad, M. T., Viardin, A., Schmitz, G., and Apel, M. Theory-training deep neural networks for an alloy solidification benchmark problem. *Computational Materials Science* (2020) 180 109687. <https://doi.org/10.1016/j.commatsci.2020.109687>
- [53] Haghighat, E., Raissi, M., Moure, A., Gomez, H., and Juanes, R. A deep learning framework for solution and discovery in solid mechanics. *arXiv:2003.02751 [cs, stat]* (2020)
- [54] Bharadwaja, B., Nabian, M. A., Sharma, B., Choudhry, S., and Alankar, A. Physics-Informed Machine Learning and Uncertainty Quantification for Mechanics of Heterogeneous Materials. *arXiv preprint arXiv:2202.10423* (2022) <https://doi.org/10.1007/s40192-022-00283-2>