Weakly Supervised Learning Technique for Solving Partial Differential Equations; Case Study of 1-D Reaction-Diffusion Equation

Behzad Zakeri[1(✉)], Amin Karimi Monsefi[2], Sanaz Samsam[3], and Bahareh Karimi Monsefi[4]

[1] School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran
Behzad.Zakeri@ut.ac.ir

[2] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran
A_karimimonsefi@sbu.ac.ir

[3] Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran
samsam.sanaz@ae.sharif.ir

[4] Faculty of Mechanical Engineering, Islamic Azad University Central Tehran Branch, Tehran, Iran

Abstract. Deep learning as a valuable intelligence tool to deal with complicated problems plays a crucial role in the 21st century. The utility of deep learning in solving partial differential equations (PDEs) is an interesting application of AI, which has been considered in recent years. However, supervision of learning procedure needs to have considerable labeled data to train the network, and this method could not be a beneficial technique to deal with unknown PDEs which we do not have any labeled data. To tackle this issue, in this paper a new method will be presented to solve PDEs only by using boundary and initial condition. Weakly supervision as an efficient method can provide an ideal bed to tackle boundary and initial value problems. To have better judgment about this method we chose Reaction-Diffusion equation as a versatile equation in engineering and science to be solved as a case study. By using the weakly supervised method and the finite difference method reaction-diffusion equation have solved, and the results of these methods have been compared. It has been shown that the results of deep learning have high consistency with finite difference results, and weakly supervised learning can be introduced as an efficient method to solve different types of differential equations.

Keywords: Deep learning · Weakly supervised learning · PDEs · Dynamical systems

L. Grandinetti et al. (Eds.): TopHPC 2019, CCIS 891, pp. 367–377, 2019.
https://doi.org/10.1007/978-3-030-33495-6_28

1 Introduction

Machine learning can be mentioned as one of the most influential elements in the contemporary century [2]. This powerful tool plays a crucial role in large number of engineering purposes. For instance, smile recognition in cameras, intelligent assistants in mobile phones, and attack detection in the network were no longer possible, since machine learning made them possible [1,15,24]. Supervised learning as a subset of machine learning, needs a labeled database for learning procedure [10]. Although, providing a proper database considerably improve the performance of the learning process, but this database is not always available. To deal with this problem, weakly supervised learning has been introduced [14,18]. Typically, there are three types of weak supervision which are, incomplete supervision, inaccurate supervision and inexact supervision [30]. All of these methods are designed to make the learning process possible without the large labeled database. This feature of weakly supervised learning techniques makes it a comfortable bed for defining physical problems which we do not have sufficient data about. Among all of the techniques that we use for studying physical problems, one of the important ones is solving the partial differential equations.

Partial differential equations (PDEs) are one of the cornerstones in mathematical modelling. Most of the physical description of natural phenomena are being simulated by taking advantage of PDEs [13]. For example, fluid motion (Navier-Stokes equation), electromagnetic field (Gauss's law) and electrodynamics (Schrödinger equation) have been modelled using partial differential equations [3,5,22]. Solving these equations based on analytical methods in most cases is impossible, and numerical methods should be implemented to represent the approximate solution. Common numerical methods like finite element method (FEM) and finite volume method (FVM) have some inevitable problems, such as mesh dependency and long computational time, which motivate scientists to find an alternative to these methods. Considering recent advances in machine learning and specifically Deep learning, ML methods could be an efficient alternatives for conventional numerical methods for solving PDEs in near future [23].

One-dimensional Reaction-Diffusion equation is known as one of the versatile differential equations in science and engineering. This equation can be used for modelling several phenomena, such as Turbulent flows, diffusion of ions in a reactive medium and financial progress in competitive environment [9,12,26]. As a more sensible example, the case of sulfate attack to concrete is one of the famous cases that reaction-diffusion equation can successfully simulate. In this case, we are interested in finding the concentration of sulfate ions in a known time and position, and since sulfate and concrete react with each other we have to solve reaction-diffusion for finding the correct concentration distribution [6,20,27].

In this paper, we focus on the one-dimensional reaction-diffusion with Dirichlet boundary condition. We tend to encode the behavior of the equation into a loss function, in a way that deep learning algorithm can learn and generate correct solutions for any time frame without having any labeled data. For this purpose, a convolutional kernel has been designed which encode the constraints

that must be satisfied in each time step and position, and this kernel was used to determine the loss function. By minimizing the loss function, the deep neural network learns to satisfy the constraints during solving the equation and learn the physics of the reaction-diffusion equation effectively. The reaction-diffusion equation with specified boundary conditions has been solved by weakly supervised method, also the numerical solution of this equation based on finite difference method (FDM) presented and the results of the deep learning algorithm have compared with the FDM method.

2 Related Work

This work is an interdisciplinary study between artificial intelligence and dynamical systems, and each one has been pursued by a large number of researchers. The former subject is studying by data scientists and AI developers. The main aim in weakly supervised learning is providing a general platform technique which learning algorithms will be able to learn with limited initial labeled data for training stage. The latter subject is searching to find an accurate method for solving partial differential equations. The utility of different numerical techniques for the approximation of the solution of PDEs is one of the important part of these researches.

Exchanging conventional numerical methods with alternative meshless techniques like machine learning in recent become increasingly popular. Especially in case of problems with the complex mathematical formulation machine learning schemes are replacing with classical models. Oquab *et al.* have used weakly supervised convolutional neural network for object classification in image processing to reduce the number of labeled input images [18]. This technique was a general concept, and has used in different applications, such as automatic classification, medical image analysis and solving differential equations [4,11,25]. Sharma *et al.* trained an encoder-decoder U-Net architecture, which was a fully convolutional neural network to solve the steady-state two-dimensional heat equation on a regular square. For this purpose, they used weakly supervised learning techniques in defining a proper convolutional kernel and loss function to train the network only by using the boundary conditions of the PDE rather than providing a large number of labeled data-sets [21]. Han *et al.* have introduced a new method for solving high-dimensional PDEs with the utilities of deep learning. They reformulate the PDEs in the form of backward stochastic differential equations (BSDEs) and then by using deep learning approximate the gradient of the solution. Although their method is accurate for dealing with high-dimensional cases, complexities of this method justify the searching for the comprehensive approach to tackle linear and low-dimensional PDEs [8,28].

On the other hand, conventional numerical methods such as FDM and FVM, have been widely developed to tackle different types of mathematical problems which representation of analytical solution for them is not available [17]. For instance, for the above example of the application of the reaction-diffusion equation in sulfate attack, Guo *et al.* have used the finite difference method to find

the concentration distribution of sulfate ions in concrete [31]. Also, extended researches have conducted on the application of machine learning in different engineering fields, such as tackling with turbulent flows and control theory [7,16,29].

3 Physics

3.1 Reaction-Diffusion Equation

For solving 1-D reaction-diffusion equation, a simple line has been considered as domain with Dirichlet boundary condition at the ends of the line. By assigning the arbitrary constant to the diffusion coefficient, we can control the role of the material on the transport phenomena. Also, the reaction coefficient specified the effect of interaction between the diffusive substance and medium. In this simulation, a high concentration applied to the boundaries, and the aim is modelling the propagation of that substance among the domain. The boundary conditions are given by $C(0,t) = C(L,t) = C_0$, and we want to determine $C(x,t)$, the concentration field in arbitrary time.

The general form of the reaction-diffusion equation in one-dimensional space is shown in Eq. 1:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - RC \tag{1}$$

Where $D, R > 0$ are the diffusion coefficient and reaction rate between specified material and domain respectively.

The analytical techniques for solving the reaction-diffusion equation are not as simple as pure diffusion. For this reason, we utilize numerical methods to obtain an accurate solution.

3.2 Finite Difference Method

The finite difference is a simple numerical method which is used to compute the accurate solution of the partial differential equations in regular domains. In this method governing equation and the domain both discretized, and the equations solved iteratively on the discrete domain. Considering the discretization of the domain 3 and time 4, the discretized form of reaction-diffusion equation for position m and time n would be:

$$\frac{C_m^{n+1} - C_m^n}{\Delta t} = D\frac{C_{m-1}^n - 2C_m^n + C_{m+1}^n}{\Delta x^2} - RC_m^n \tag{2}$$

$$\frac{x_L - x_0}{m} = \Delta x \tag{3}$$

$$\frac{t_\infty - t_0}{n} = \Delta t \tag{4}$$

Where C_m^n is the concentration in time n and position m, also indices 0, L and ∞ represent the initial step in time and position, end of the domain and last time step respectively.

With solving the Eq. 2 iteratively, the value of the C in each time and position converge to the real value.

4 Deep Learning Solver

The aim of this work is using the deep neural network to solve the reaction-diffusion equation with only using boundary and initial conditions, without knowing the numerical or analytical solution or even having any labeled data. For this purpose, the differential equation has been decoded into a *physical-informed loss function*. This technique helps us to find the solution of the PDE without using supervision in the form of data.

To import the initial and boundary conditions of the problem into the deep neural network, we used a $n \times m$ matrix which its columns and rows represent the positions and time steps respectively. All of the matrix elements for the input matrix are zero except the first and last columns which their values represent the boundary condition values (which in this study is C_0). Also, in this matrix each row demonstrates the concentration distribution in a specified time-frame.

4.1 Deep Learning Architecture

A fully convolutional encoder-decoder network in the form of U-Net architecture has been utilized in this study as Ronneberger *et al.* have used this architecture for biomedical image segmentation [19]. The main reason for choosing a fully convolutional architecture among other architectures is the flexibility of this structure to solve problems at multiple scales. The network contains several encoding convolutional and decoding pooling layers which save the input matrix size during the learning process. Finally, the output matrix represents the solution of the PDE in the discretized space-time domain. The schematic structure of the network is shown in Fig. 1:

As it is shown in Fig. 1, each encoding layer has been connected to the corresponding decoder layer using Fusion connection. The reason for the utility of fusion connections is to pass the boundary values of the input to the output layers, and by this technique, the network is not forced to memorize the structure of the input in its bottleneck layers. The number of layers in our architecture is arbitrary, and it is simply possible to add layers into the network as much as necessary.

4.2 Kernel

To make an intelligent network that can solve the equation in any time and position, it is necessary to define the governing rule in that equation in a simple way for the neural network. It is similar to the method that FDM use for solving the discretized equation. In fact, by discretization of a continues equation and transferring that equation into the algebraic form we can observe the governing rule for every point in space and time.

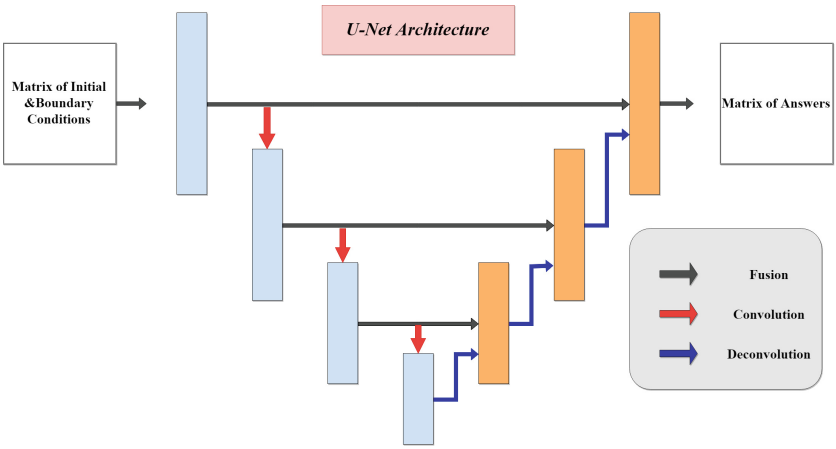


Fig. 1. Deep neural network diagram

By reforming the Eq. 2, we can find the state of an arbitrary point in the space-time domain based on its neighbours as shown in Eq. 5:

$$C_m^{n+1} = C_m^n + B\left(C_{m-1}^n - 2C_m^n + C_{m+1}^n\right) - RC_m^n \Delta t \tag{5}$$

And B defined as follow:

$$B = \frac{D\Delta t}{\Delta x^2} \tag{6}$$

For transferring the relation among variables into the neural network, Eq. 5 have been decoded into the 3×3 convolutional kernel as follow:

$$\begin{pmatrix} -a & -b & -c \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{7}$$

Where:

$$a \to B \qquad c \to B \qquad b \to (1 - 2B - R\Delta t) \tag{8}$$

Discussed Kernel has been convolved into the across the input matrix, and the output matrix after normalization was used to calculate the Loss function:

$$\sum_{i,j} \left(Conv2D(Kernel, Output)_{i,j}\right)^2 \tag{9}$$

By minimizing the Eq. 9, the deep neural network tries to make its' solution closer to the real values which can be found in Eq. 5 and changing in boundary and initial conditions train the network for solving any type of problems governed by reaction-diffusion physics.

5 Results

In this section, we will discuss the results of the deep learning solution, and compare the answers to finite difference results. In the presented study to have more realistic results, boundary conditions and all coefficients were chosen from the work of Zuo *et al.* on the sulfate attack to concrete [31]. As it was raised in the deep learning section, the output of the U-Net network for specific input is a matrix which its columns and rows represent the position and time, and the value of each element demonstrates the concentration in ith time and jth position. In Fig. 2, a sample output matrix of the deep learning solver with constant boundary condition in both sides of the domain is shown:

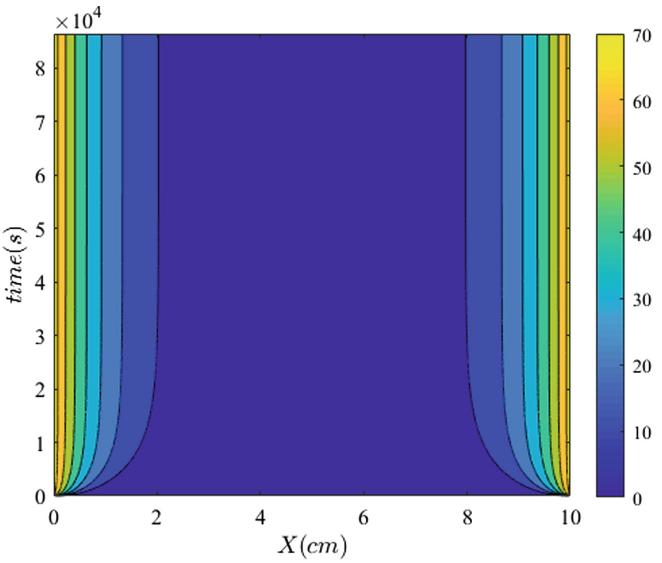


Fig. 2. Sample output matrix from neural network

Looking at Fig. 2 in more detail, the progress of concentration diffusion along the time axis is obviously visible. The value of all elements on the first and last column are same and equal to the boundary value. Also, it is clear that by passing the time, the gradient along time and position decrease, and the answer converge to the steady-state solution.

Figure 3 compares the deep learning solution with finite difference method results in terms of concentration distribution along domain in different times. With looking more precisely to Fig. 3, we observe that deep learning results have high consistency with FDM results. The only part that deep learning could not predict correctly was in the first and last part of the 1 s timeline. The reason for this disability of deep learning in these regions is the high gradient in these areas.

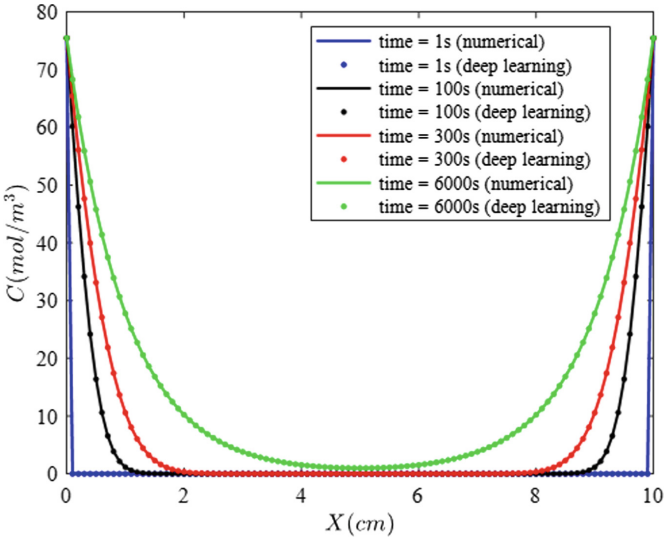


Fig. 3. 2D comparison of deep learning with FDM

In Fig. 4, solutions of the reaction-diffusion equation in three different sets of coefficients have been shown based on deep learning and numerical method. We have set the proportion of reaction and diffusion coefficients in a way that by solving the equation with these coefficients observe the physic of pure diffusion, pure reaction and reaction-diffusion.

For this purpose, a dimensionless coefficient has been defined which help us to calculate the correct proportion of reaction and diffusion coefficients to have all three state of the solution in our computation.

Damköhler number is an important dimensionless parameter in chemical engineering which clarifies the role of diffusion, reaction or simultaneous reaction-diffusion phenomena in transport phenomena and define as follow:

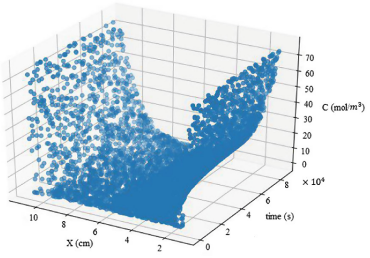
$$D_a = \frac{Rate\ of\ reaction}{Diffusion\ rate} \tag{10}$$

In our model Eq. 1, Damköhler number is defined as:

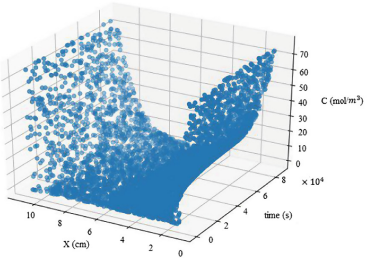
$$D_a = \frac{RL^2}{D} \tag{11}$$

This number represents the states of reaction-diffusion in different states where $D_a \cong 1$, $D_a \gg 1$, and $D_a \ll 1$ mean the physics of *Reaction-Diffusion*, *pure Reaction*, and *pure Diffusion* respectively.

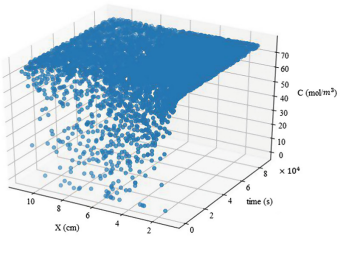
To have a quantitative assessment of deep learning solution, we assumed one of the coefficients constant, and by changing the other coefficient MSE value has been computed, and the result of this analysis is reported in Table 1:



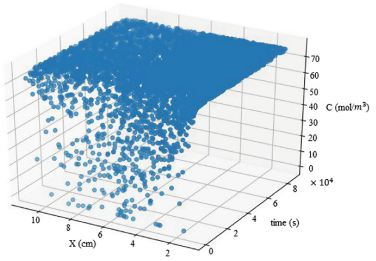
(a) Deep Learning, $D_a = 1$



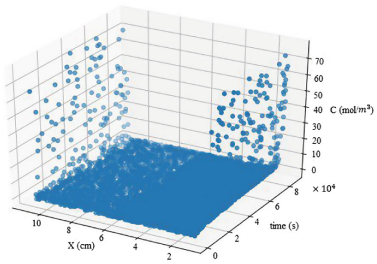
(b) FDM, $D_a = 1$



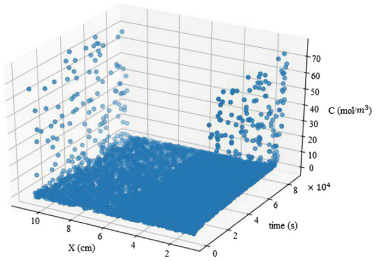
(c) Deep Learning, $D_a = 10^4$



(d) FDM, $D_a = 10^4$



(e) Deep Learning, $D_a = 10^{-4}$



(f) FDM, $D_a = 10^{-4}$

Fig. 4. 3D comparison of deep learning with FDM

Table 1. Accuracy analyze based on changing coefficients

(a) $D = 2.7 \times 10^{-9}$

R coefficient	MSE value
2.25×10^{-2}	0.587
2.25×10^{-5}	0.495
2.25×10^{-7}	0.623
2.25×10^{-10}	0.341

(b) $R = 2.25 \times 10^{-7}$

D coefficient	MSE value
2.7×10^{-2}	0.305
2.7×10^{-7}	0.576
2.7×10^{-8}	0.588
2.7×10^{-10}	0.534

The Mean Square Error index has been utilized to calculate the deep learning error. We can see that the accuracy of deep learning results is dependent on the coefficients of the equation; however, this dependency does not influence the final results quality.

6 Conclusion

In this paper, the ability of weakly supervised learning in solving the transient one-dimensional partial differential equation has shown. We saw that the results of deep learning method perfectly were similar to the FDM results. Also, we have observed that the value of the equation's coefficients can influence the deep learning accuracy. Although in this study, this effect did not distract our solution, it could manipulate the results in critical cases such as stochastic differential equations.

This technique is a functional method to approach problems that we do not have sufficient labeled data about, and authors believe that by decoding the physics of the problems into the kernel and using a proper network architecture, a wide spectrum of problems can be solvable.

References

1. Bai, Y., Guo, L., Jin, L., Huang, Q.: A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 3305–3308. IEEE (2009)
2. Bell, S.: Project-based learning for the 21st century: skills for the future. Clearing House **83**(2), 39–43 (2010)
3. Chorin, A.J.: Numerical solution of the Navier-Stokes equations. Math. Comput. **22**(104), 745–762 (1968)
4. Ciompi, F., et al.: Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Med. Image Anal. **26**(1), 195–202 (2015)
5. Feit, M., Fleck Jr., J., Steiger, A.: Solution of the Schrödinger equation by a spectral method. J. Comput. Phys. **47**(3), 412–433 (1982)
6. Gospodinov, P.N., Kazandjiev, R.F., Partalin, T.A., Mironova, M.K.: Diffusion of sulfate ions into cement stone regarding simultaneous chemical reactions and resulting effects. Cem. Concr. Res. **29**(10), 1591–1596 (1999)
7. Guo, X., Yan, W., Cui, R.: Integral reinforcement learning-based adaptive NN control for continuous-time nonlinear MIMO systems with unknown control directions. IEEE Trans. Syst. Man Cybern.: Syst. (2019). https://doi.org/10.1109/TSMC.2019.2897221
8. Han, J., Jentzen, A., Weinan, E.: Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. **115**(34), 8505–8510 (2018)
9. Kramers, H.A.: Brownian motion in a field of force and the diffusion model of chemical reactions. Physica **7**(4), 284–304 (1940)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
11. Liu, P., Gan, J., Chakrabarty, R.K.: Variational autoencoding the Lagrangian trajectories of particles in a combustion system. arXiv preprint arXiv:1811.11896 (2018)

12. Majda, A.J., Souganidis, P.E.: Large scale front dynamics for turbulent reaction-diffusion equations with separated velocity scales. Nonlinearity **7**(1), 1 (1994)
13. Mattheij, R.M., Rienstra, S.W., ten Thije Boonkkamp, J.H.: Partial Differential Equations: Modeling, Analysis, Computation, vol. 10. SIAM, Philadelphia (2005)
14. Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 992–999 (2007)
15. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Springer, Heidelberg (2013)
16. Mohan, A.T., Gaitonde, D.V.: A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. arXiv preprint arXiv:1804.09269 (2018)
17. Morton, K.W., Mayers, D.F.: Numerical Solution of Partial Differential Equations: An Introduction. Cambridge University Press, Cambridge (2005)
18. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 685–694 (2015)
19. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
20. Sarkar, S., Mahadevan, S., Meeussen, J., Van der Sloot, H., Kosson, D.: Numerical simulation of cementitious materials degradation under external sulfate attack. Cem. Concr. Compos. **32**(3), 241–252 (2010)
21. Sharma, R., Farimani, A.B., Gomes, J., Eastman, P., Pande, V.: Weakly-supervised deep learning of heat transport via physics informed loss. arXiv preprint arXiv:1807.11374 (2018)
22. Singh, C.: Student understanding of symmetry and Gauss's law of electricity. Am. J. Phys. **74**(10), 923–936 (2006)
23. Sirignano, J., Spiliopoulos, K.: DGM: a deep learning algorithm for solving partial differential equations. J. Comput. Phys. **375**, 1339–1364 (2018)
24. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 305–316. IEEE (2010)
25. Tajbakhsh, N., et al.: Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans. Med. Imaging **35**(5), 1299–1312 (2016)
26. Tang, L.H., Tian, G.S.: Reaction-diffusion-branching models of stock price fluctuations. Phys. A **264**(3–4), 543–550 (1999)
27. Tixier, R., Mobasher, B.: Modeling of damage in cement-based materials subjected to external sulfate attack. I: formulation. J. Mater. Civ. Eng. **15**(4), 305–313 (2003)
28. Weinan, E., Han, J., Jentzen, A.: Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat. **5**(4), 349–380 (2017)
29. Wu, J.L., Xiao, H., Paterson, E.: Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. Phys. Rev. Fluids **3**(7), 074602 (2018)
30. Zhou, Z.H.: A brief introduction to weakly supervised learning. Natl. Sci. Rev. **5**(1), 44–53 (2017)
31. Zuo, X.B., Sun, W., Yu, C.: Numerical investigation on expansive volume strain in concrete subjected to sulfate attack. Constr. Build. Mater. **36**, 404–410 (2012)