Transfer-learning Enhanced Physics Informed- Neural-Networks for Burgers' Equation with High Reynolds' Number

Nishant Kumar[1*], Manish Kr. Tiwari[2*], P.S Gurugubelli[1],
Anil Kumar[2], M. Raghavendra Rao[2]

[1]Birla Institute of Technology & Science, Pilani, Hyderabad
[2]Advanced Systems Laboratory, DRDO, Hyderabad

ABSTRACT

We have attempted to solve Burgers' equation with Physics Informed Neural Networks (PINNs) using transfer learning to address the difficulties in solving the equation with high Reynolds number, that is, low values of kinematic viscosity. The obtained solutions are compared with benchmark results from classical numerical methods, and it is found that they are in close agreement. PINNs are also shown to accurately model physical behaviour at low viscosities Transfer learning is shown to be a great enhancement, particularly for low values of viscosity.

Keywords: *Burger's Equation, PINN, Transfer Learning, Reynolds Number*

1. INTRODUCTION

Partial Differential Equations (PDEs) and Ordinary Differential Equations (ODEs) have gained large popularity, due both to their mathematical and practical significance. Some of the most important PDEs that describe physical phenomena include the Navier-Stokes, the Euler's equations, the heat equation, and Schrodinger's equation. Because it is often difficult or impossible to obtain the analytic or exact solutions for these PDEs, there has been much interest in methods for obtaining numerical solutions. Finite difference methods, finite element analysis, finite volume analysis, and spectral methods are some of the schemes employed to obtain the solution of these problems.

Another group of powerful computational methods that have gained prominence over the past three decades are machine learning techniques. The field of deep learning has particularly taken off following the Deep Learning Revolution. However, these methods are data-driven, generally rendering them unsuitable for applications where obtaining data might be difficult or expensive.

There has only recently been a surge of interest at the intersections of PDEs and machine learning, especially with the advent of frameworks such as "Physics Informed Neural Networks". What is known as Scientific Machine Learning has gained prominence, and continued to expand developing frameworks such as Universal Differential Equations [1]. Many libraries for the same have also been developed, such as TensorDiffEq [2], DeepXDE [3], SciANN [4], and more. Apart from these, Julia has emerged as a prominent programming language for the implementations of these models as well as development of interfaces to facilitate such frameworks [5]. Physics Informed Neural Networks (or PINNs) were formally introduced by [6]. Although the idea of solving differential equations utilizing neural networks had originated much earlier [7], the interest in the subject was spurred by this work.

Despite significant interest in PINNs, and Burger's equation being a central benchmark in its research, there have not yet been any studies evaluating PINNs on some common set of

* Corresponding author: E-mail: nishantkumar490@gmail.com

problems widely used in the study of numerical methods. This would also entail evaluating the performance of PINNs at very low viscosities, which most numerical methods find difficult to capture and remains an active area of research. The following study hopes to contribute to that. In addition, it also provides great evidence of the usefulness of transfer learning in the context of PINNs.

The study is organised as follows. Section 2 details the methodology, including the architecture of PINNs, the structure of Burger's equation, and transfer learning. Section 3 summarises our results with a brief discussion. Section 4 concludes our paper, while Section 5 describes the challenges and future work.

2. Methodology

The fundamental idea of PINNs can be encapsulated by what is known as the "Universal Approximation Theorem". In various formulations, the theorem states that neural networks, of enough breadth and/or depth, are capable of arbitrarily approximating any real valued function [8] [9]. PINN utilizes the power of automatic differentiation, through the same technique that backpropagation is carried out, and approximates the hidden solution of a differential equation.

This is done through formulating the loss equation to reduce the residual of the Partial Differential Equation, along with the given Initial and Boundary Conditions of the problem. In formal terms, given the PDE:

$$f(x,t,u,u_x,u_t,\ldots,\lambda)=0 \quad x \in \Omega,\ t \in [0,T]$$

with the Initial Condition:

$$u(x,t_0)=g_0(x) \quad x \in \Omega$$

and the Boundary Conditions:

$$u(x,t) = g_\Gamma(t) \quad x \in \Gamma,\ t \in [0,T]$$

The loss function can be formulated as

$$Loss = Loss_f + Loss_{BC} + Loss_{IC}$$

With the availability of data, an additional loss term for data can also be added.

PINNs have some distinct advantages over classical numerical methods. First, PINNs is a meshless method, and helps avoid the complex mesh generation involved in the classical computational methods. In addition to this, any available data from experimental or other computational sources can seamlessly be integrated into this framework. PINNs also provide a greater opportunity for exploring inverse or "ill posed" problems, as they can also be investigated using essentially the exact same framework.

Despite these perks, computational costs remain a major obstacle in the implementation of PINNs [10] [11]. One of the techniques whose power can be leveraged against this is transfer learning: transferring the knowledge of one task to other similar tasks. In this case, tackling various kinds of boundary conditions and parameters can be done through this technique. Despite many of the works highlighting the importance of transfer learning in this context [11], this area has been relatively underexplored in literature. We attempt to investigate the effects of transfer learning on the convergence of models with differing parameters and

ICs/BCs, taking into consideration a well-known benchmark of the PINNs: Burger's equation.

Burger's equation is a simplified form of the Navier-Stokes, without the pressure gradient term. The equation is as follows:

$$u_t + uu_x = \nu u_{xx} \tag{1}$$

where v is the kinematic viscosity.

The Burgers' equation has a wide range of applications, from modeling of turbulence to cosmology. It is especially useful for testing of numerical schemes [12]. Indeed, Burgers' equation has widely been used as a benchmark for PINN studies [6]. The analytic solution to this equation is available, and is given by the Hopf-Cole [13] transformation as follows:

$$u \ = \ -2\nu \frac{\theta_x}{\theta}$$

This transformation converts the original Burger's equation to a form of the heat equation. By separation of variables method, an analytic solution can be reached. For the boundary conditions,

$$\mathrm{u}(0,\mathrm{t}) = \mathrm{u}(1,\mathrm{t}) = 0$$

And the initial conditions,

$$\mathrm{u}(\mathrm{x},0) = \mathrm{f}(\mathrm{x})$$

The solution is given by,

$$u(x,t) \ = \ \frac{2\pi\nu \sum_{m=1}^{\infty} mA_m \sin(m\pi x)\exp(-m^2\pi^2 t\nu)}{A_0 + \sum_{m=1}^{\infty} A_m \cos(m\pi x)\exp(-m^2\pi^2 t\nu)} \tag{2}$$

Where,

$$A_m = 2\int_0^1 \cos(\mathrm{m}\pi\mathrm{x})\exp\{-\frac{1}{2\nu}\int_0^x f(x')\,\mathrm{d}x')\}\,dx$$

$$A_0 \ = \int_0^1 exp\left\{-\frac{1}{2\nu}\int_0^x f(x')dx'\right\}dx \tag{3}$$

Despite the availability of analytic solutions, for many problems especially for lower values of viscosity (i.e., higher Reynolds number) this infinite series does not converge well [14]. Hence, numerical methods are required to compute solutions. Many such schemes have been developed over the years. [15] present an explicit and exact-explicit finite difference method. A finite element method is provided by [16]. Galerkin methods remain popular for solving these kinds of problems [17]. Recently, wavelet-based solutions have also shown success in this regard [18] [19].

We propose PINN as another method for solving particularly challenging versions of the Burgers' equation. In this approach, we employ transfer learning as a critical technique to enhance the capability of neural networks. Additionally, this helps evaluate the effectiveness of transfer learning methods in PINNs more generally, since Burgers' equation is an essential benchmark for any numerical computational technique.

Transfer learning is characterized by transferring knowledge across domains. Various techniques are used such as parameter sharing, feature-based strategy, model-based regularization, etc. [20]. Parameter sharing and fine tuning are well-regarded strategies in image processing and natural language processing tasks [21] [22]. In this method, parameters are initialized based on a benchmark network (eg. ImageNet) and the last few layers of the network are "fine-tuned" for a specific task, rather than building a network from scratch. This is based on the idea that the early layers of the network capture generalities, while the later layers get more specific.

In PINNs, these methods can be used for enhancing the learning of related sets of equations. The use of transfer learning can significantly boost the ability to solve problems with differing parameters, initial and boundary conditions.

There has been some research into the aspect of transfer learning in the context of PINNs. A key study was conducted by Desai et al, who trained networks on six families of differential equations: First order ODEs, Second-order ODEs, Coupled linear oscillators, Nonlinear oscillators, Poisson and Schrodinger's. They concluded that PINNs have the ability to learn from as few as three sets of parameters, and can then perform one-shot approximations of closely related sets of parameters with high accuracy.

Burgers' equation poses some unique problems. Modeling flows with lower levels of viscosity, or higher Reynolds number remains a challenge as this often results in formation of shock waves near boundary layers. There is also a well-established set of boundary conditions that are not as closely related as those considered by [23]. Transfer learning can thus provide a deeper advantage for these less viscous flows, whose solutions it is otherwise difficult to converge.

3. RESULTS AND DISCUSSION

We consider two problems with distinct boundary conditions for a few values of viscosity, and we train PINNs with a depth of 5 hidden layers, and a width of 20 neurons per layer based on the governing equations of these problems. We use Adam optimizer with a training rate of 5xE-4 for all the networks, but we may adjust the learning rate during training depending on the specific requirements. We compare the values with some classical numerical methods reported in various studies.

3.1. PROBLEM 1

We consider Equation (1) with the following boundary and initial conditions:

$$\begin{aligned} &u(0,t) = u(1,t) = 0, \qquad t > 0 \\ &u(x,0) = \sin(\pi x), \qquad 0 \le x \le 1 \end{aligned} \tag{4}$$

The solution from equation (3) becomes:

$$A_m = 2\int_0^1 \cos(m\pi x)\exp\left(\frac{1}{-2\pi\nu}(1-cos(m\pi x))\right)\mathrm{d}x$$

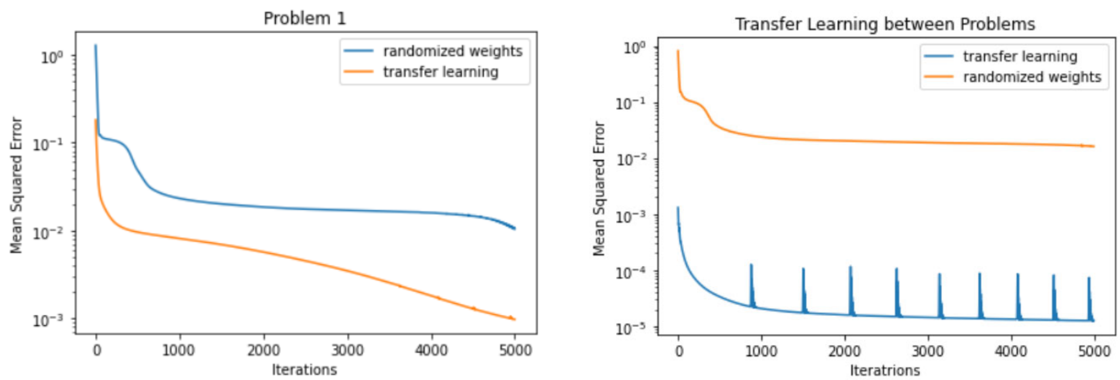
$$A_0 = \int_0^1 \exp\left(\frac{1}{-2\pi\nu}(1-cos(\pi x))\right)\mathrm{d}x \qquad (5)$$

Table 1 Solution for Problem 1 at v=0.1 and v=0.01.

x	t	v=0.1				v=0.01			
		Ref. [16]	**Ref. [18]**	**Present**	**Exact**	**Ref. [16]**	**Ref. [18]**	**Present**	**Exact**
0.25	0.4	0.31429	0.30889	0.30866	0.30889	0.34244	0.34191	0.34191	0.34191
	0.6	0.24373	0.24075	0.24076	0.24074	0.26905	0.26896	0.26887	0.26896
	0.8	0.19758	0.19569	0.19586	0.19568	0.22145	0.22148	0.22141	0.22148
	1.0	0.16391	0.16258	0.16260	0.16256	0.18813	0.18820	0.18815	0.18819
0.5	0.4	0.57636	0.56963	0.56955	0.56963	0.67152	0.66069	0.66083	0.66071
	0.6	0.45169	0.44724	0.44711	0.44721	0.53406	0.52942	0.52948	0.52942
	0.8	0.36245	0.35927	0.35924	0.35924	0.44143	0.43914	0.43917	0.43194
	1.0	0.29437	0.29195	0.29191	0.29192	0.37568	0.37443	0.37447	0.27442
0.75	0.4	0.62592	0.62537	0.62552	0.62544	0.94675	0.91023	0.91019	0.91026
	0.6	0.49034	0.48718	0.48706	0.48721	0.78474	0.76723	0.76714	0.76724
	0.8	0.37713	0.37391	0.37389	0.37392	0.65659	0.64740	0.64751	0.64740
	1.0	0.29016	0.28747	0.28758	0.28747	0.56135	0.55606	0.55609	0.55605

The collocation points are taken from an equispaced 100x100 grid on the domain $0 \le x \le 1$, $0 \le t \le 1$. PINNs are trained for values v=0.1 and v=0.01.

The network trained with v=0.01 is then used for transfer learning of v=0.004. Only the last two layers are fine-tuned and the rest of the network is frozen. In the last few iterations, the model is “opened”, i.e., all the weights are updated. This provides a heavy advantage in the convergence of the network as can be seen in Fig 1. Further, networks for viscosity values 0.001 and 0.0001 are also trained through this transfer-learning enhanced method.



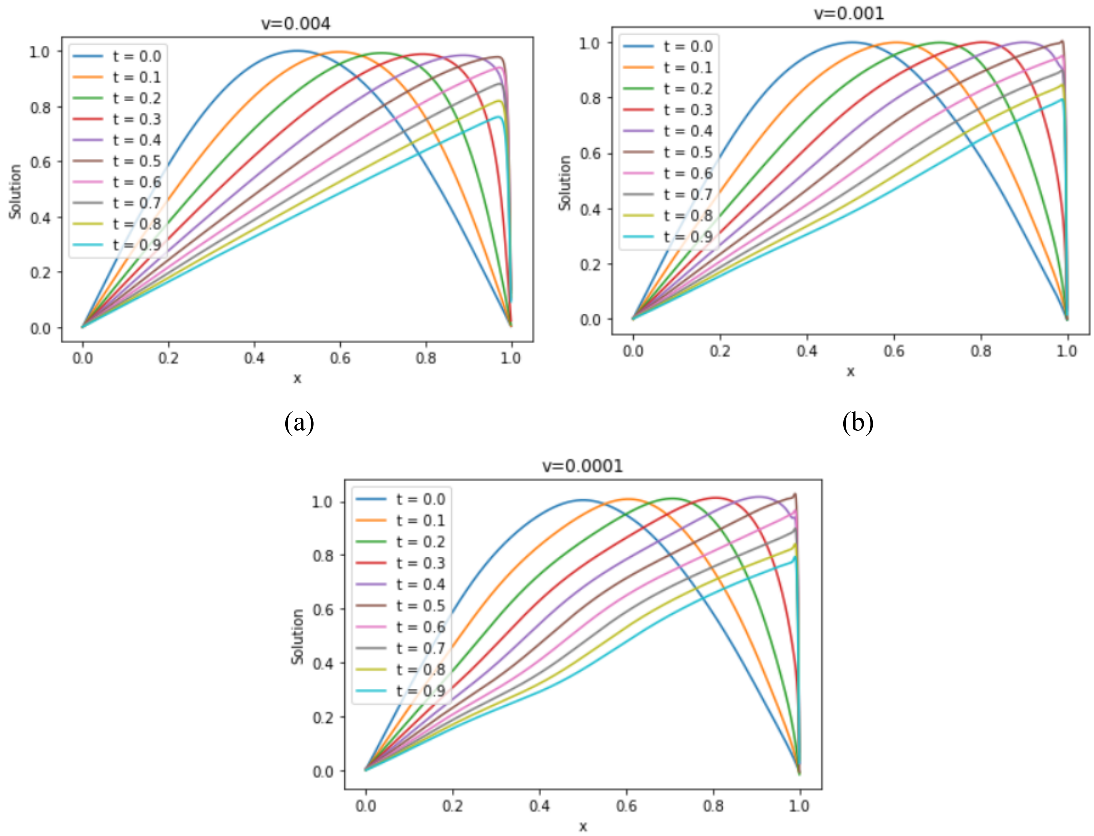
(a) (b)

Figure 1: Transfer learning. (a) shows transfer learning between two values of viscosity. Comparison of Problem 1 at v=0.004 between fine-tuning a pre-trained model on v=0.01 vs. training from randomly initialized weights. (b) shows transfer learning between Problem 1 and Problem 2 at v=0.004.

Table 2 Solution for Problem 1 and Problem 2 at v=0.004

x	**t**	**PROBLEM 1**			**PROBLEM 2**		
		Ref. [18]	**Present**	**Exact**	**Ref. [18]**	**Present**	**Exact**
0.25	1	0.18891	0.18936	0.18891	0.19640	0.19682	0.19641
	5	0.04696	0.04625	0.04697	0.04744	0.04671	0.04747
	10	0.02422	0.02414	0.02421	0.02434	0.02417	0.02434
	15	0.01632	0.01575	0.01631	0.01637	0.015503	0.01637
0.5	1	0.37598	0.37588	0.37596	0.38850	0.38849	0.038846
	5	0.09394	0.09369	0.09393	0.09491	0.09451	0.09491
	10	0.04843	0.04819	0.04843	0.04868	0.04822	0.04870
	15	0.03259	0.03234	0.03259	0.03271	0.03236	0.03270
0.75	1	0.55883	0.55848	0.55881	0.57320	0.57271	0.57315
	5	0.14089	0.14076	0.14089	0.14225	0.14205	0.14225
	10	0.07221	0.07161	0.07220	0.07258	0.071830	0.07257
	15	0.04678	0.04702	0.04677	0.04697	0.04719	0.04695

ISBN - 978-93-95038-50-8



(a) (b)

(c)

Figure 2: Solutions for Problem 1 at different values of viscosity. PINNs have the ability to quite accurately model physical behaviour at high Reynolds numbers.

3.2. Problem 2

We consider Equation (1) with the following boundary and initial conditions:

$$u(0,t) = u(1,t) = 0, \qquad t > 0$$
$$u(x,0) = 4x(1-x), \qquad 0 \le x \le 1 \tag{6}$$

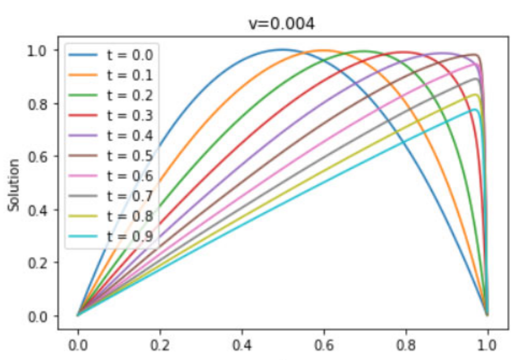
The solution from equation (3) becomes:

$$A_m = 2\int_0^1 \cos(m\pi x)\exp\left(\frac{1}{-3\nu}\left(3x^2 - 2x^3\right)\right)\mathrm{d}x$$
$$A_0 = \int_0^1 \exp\left(\frac{1}{-3\nu}\left(3x^2 - 2x^3\right)\right)\mathrm{d}x \tag{7}$$

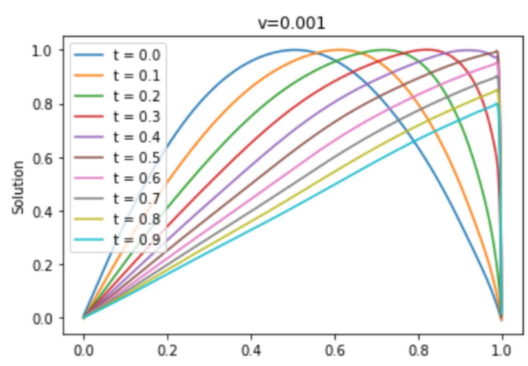
The collocation points are taken from an equispaced 100x100 grid on the domain 0≤x≤1, 0≤t≤1. PINNs from Problem 1 of equal viscosity are taken and trained with the governing equations. The comparison of the present method with exact solutions are as follows:

Table 3 Solution for Problem 2 at v=0.1 and v=0.01.

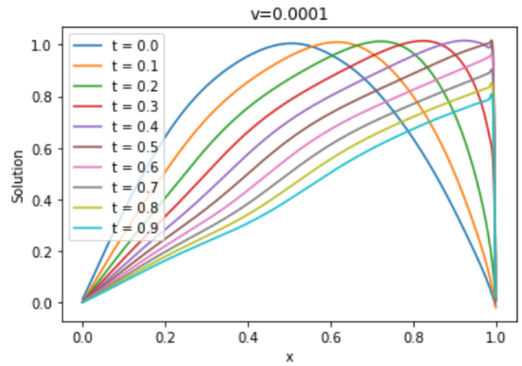
x	t	v=0.1				v=0.01		
		Ref. [16]	**Ref. [18]**	**Present**	**Exact**	**Ref. [18]**	**Present**	**Exact**
0.25	0.4	0.32679	0.31752	0.31740	0.31752	0.36225	0.36219	0.36226
	0.6	0.25117	0.24615	0.24620	0.24614	0.28204	0.28189	0.28204
	0.8	0.20270	0.19957	0.19963	0.19956	0.23045	0.23031	0.23045
	1.0	0.16780	0.16561	0.16552	0.16560	0.19469	0.19452	0.19469
0.5	0.4	0.59661	0.58454	0.58444	0.58454	0.68364	0.68380	0.68368
	0.6	0.46581	0.45800	0.45799	0.45798	0.54831	0.54835	0.54832
	0.8	0.37293	0.36744	0.36733	0.36740	0.45371	0.45372	0.45371
	1.0	0.30253	0.29838	0.29838	0.29834	0.38568	0.38570	0.38568
0.75	0.4	0.64680	0.64556	0.64564	0.64562	0.92044	0.92037	0.92050
	0.6	0.50852	0.50265	0.50263	0.50268	0.78297	0.78276	0.78299
	0.8	0.39117	0.38532	0.38533	0.38534	0.66272	0.66279	0.66272
	1.0	0.30066	0.29585	0.29583	0.29586	0.56932	0.56934	0.56932



(a)



(b)



(c)

Figure 3 Solution for Problem 2 at different values of viscosity.

4. CONCLUSION

We investigate the ability of Physics Informed Neural Networks to approximate solution to Burgers' equation at lower values of viscosity, i.e., high Reynolds numbers. While PINNs can model higher viscosities with great accuracy, it encounters difficulties at lower viscosities. We attempt to overcome this problem by using transfer learning enhanced techniques. We have found that transfer learning and fine tuning are greatly advantageous to these problems. The PINNs are made to solve two problems and the results are compared to exact solutions and some benchmark data from classical numerical methods. The solution is found to be in agreement with them.

5. CHALLENGES AND FUTURE WORK

While PINNs have shown the capability of approximating solution for Burgers' equation for a large range of values of kinematic viscosity within the same framework, there are many challenges in this area. The accuracy, as well as computational cost, worsens for much lower values of viscosity. Some possible methods of mitigation can include weighing the loss functions. These could either be hyper parameters or self-adaptive weights. Sampling of collocation points is also an extremely important aspect and deserves attention on its own. The usual hyper parameter tuning can also greatly improve the performance of PINNs.

REFERENCES

1. Rackauckas, Christopher, et al. "Universal differential equations for scientific machine learning." arXiv preprint arXiv:2001.04385 (2020).
2. McClenny, Levi D., Mulugeta A. Haile, and Ulisses M. Braga-Neto. "TensorDiffEq: scalable multi-GPU forward and inverse solvers for physics informed neural networks." arXiv preprint arXiv:2103.16034 (2021).
3. Lu, Lu, et al. "DeepXDE: A deep learning library for solving differential equations." SIAM review 63.1 (2021): 208-228.
4. Haghighat, Ehsan, and Ruben Juanes. "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks." Computer Methods in Applied Mechanics and Engineering 373 (2021): 113552.

5. Ranocha, Hendrik, et al. "Adaptive numerical simulations with Trixi. jl: A case study of Julia for scientific computing." arXiv preprint arXiv:2108.06476 (2021).
6. Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational physics 378 (2019): 686-707.
7. Lagaris, Isaac E., Aristidis Likas, and Dimitrios I. Fotiadis. "Artificial neural networks for solving ordinary and partial differential equations." IEEE transactions on neural networks 9.5 (1998): 987-1000.
8. Hornik, Kurt. "Approximation capabilities of multilayer feedforward networks." Neural networks 4.2 (1991): 251-257.
9. Kidger, Patrick, and Terry Lyons. "Universal approximation with deep narrow networks." Conference on learning theory. PMLR, 2020.
10. Chuang, Pi-Yueh, and Lorena A. Barba. "Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration." arXiv preprint arXiv:2205.14249 (2022).
11. Rezaei, Shahed, et al. "A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method." Computer Methods in Applied Mechanics and Engineering 401 (2022): 115616.
12. Bonkile, Mayur P., et al. "A systematic literature review of Burgers' equation with recent advances." Pramana 90 (2018): 1-21.
13. Cole, Julian D. "On a quasi-linear parabolic equation occurring in aerodynamics." Quarterly of applied mathematics 9.3 (1951): 225-236.
14. Zhang, D. S., et al. "Burgers' equation with high Reynolds number." Physics of Fluids 9.6 (1997): 1853-1855.
15. Kutluay, S. E. L. Ç. U. K., A. R. Bahadir, and A. Özdeş. "Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods." Journal of computational and applied mathematics 103.2 (1999): 251-261.
16. Öziş, T., E. N. Aksan, and A. Özdeş. "A finite element approach for solution of Burgers' equation." *Applied Mathematics and Computation* 139.2-3 (2003): 417-428.
17. Soliman, A. A. "A Galerkin solution for Burgers' equation using cubic B-spline finite elements." *Abstract and Applied Analysis*. Vol. 2012. Hindawi, 2012.
18. Jiwari, Ram. "A hybrid numerical scheme for the numerical solution of the Burgers' equation." *Computer Physics Communications* 188 (2015): 59-67.
19. Liu, XiaoJing, et al. "Wavelet solutions of Burgers' equation with high Reynolds numbers." *Science China Technological Sciences* 57 (2014): 1285-1292.
20. Zhuang, Fuzhen, et al. "A comprehensive survey on transfer learning." *Proceedings of the IEEE* 109.1 (2020): 43-76.
21. Guo, Yunhui, et al. "Spottune: transfer learning through adaptive fine-tuning." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
22. Houlsby, Neil, et al. "Parameter-efficient transfer learning for NLP." *International Conference on Machine Learning*. PMLR, 2019.
23. Desai, Shaan, et al. "One-shot transfer learning of physics-informed neural networks." *arXiv preprint arXiv:2110.11286* (2021).