Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(4): 7154–7170.
DOI: 10.3934/mbe.2023309
Received: 31 October 2022
Revised: 19 January 2023
Accepted: 01 February 2023
Published: 10 February 2023

Research article

The monotone traveling wave solution of a bistable three-species competition system via unconstrained neural networks

Sung Woong Cho, Sunwoo Hwang and Hyung Ju Hwang*

Department of Mathematics, Pohang University of Science and Technology, Pohang 37673, Republic of Korea

* **Correspondence:** E-mail: hjhwang@postech.ac.kr; Tel: +82542792056; fax: +82542792799.

Abstract: In this paper, we approximate traveling wave solutions via artificial neural networks. Finding traveling wave solutions can be interpreted as a forward-inverse problem that solves a differential equation without knowing the exact speed. In general, we require additional restrictions to ensure the uniqueness of traveling wave solutions that satisfy boundary and initial conditions. This paper is based on the theoretical results that the bistable three-species competition system has a unique traveling wave solution on the premise of the monotonicity of the solution. Since the original monotonic neural networks are not smooth functions, they are not suitable for representing solutions of differential equations. We propose a method of approximating a monotone solution via a neural network representing a primitive function of another positive function. In the numerical integration, the operator learning-based neural network resolved the issue of differentiability by replacing the quadrature rule. We also provide theoretical results that a small training loss implies a convergence to a real solution. The set of functions neural networks can represent is dense in the solution space, so the results suggest the convergence of neural networks with appropriate training. We validate that the proposed method works successfully for the cases where the wave speed is identical to zero. Our monotonic neural network achieves a small error, suggesting that an accurate speed and solution can be estimated when the sign of wave speed is known.

Keywords: partial differential equation; traveling wave solution; forward and inverse problems; physics-informed neural networks; monotonic neural networks; convergence analysis

1. Introduction

1.1. Motivation

In this paper, we consider the following bistable three-species competition systems.

$$\begin{aligned} u_t &= d_1 u_{xx} + a_1 u(1 - u - b_2 v), \\ v_t &= d_2 v_{xx} + a_2 v(1 - v - b_1 u - b_3 w), \\ w_t &= d_3 w_{xx} + a_3 w(1 - w - b_2 v), \end{aligned} \tag{1.1}$$

$$\text{where} \quad (u, v, w)(0, x) = (u_0(x), v_0(x), w_0(x)) \to \begin{cases} (u_-, v_-, w_-) & \text{as } x \to -\infty, \\ (u_+, v_+, w_+) & \text{as } x \to \infty. \end{cases} \tag{1.2}$$

The above system describes the population distribution of predators over time when three predators compete for two prey. u, v, and w represent the nonnegative population distributions of the three predators. Three distributions usually have values less than 1. $\{a_i\}_{i=1}^{3}$ and $\{d_i\}_{i=1}^{3}$ denote each predator's diffusion coefficient and net growth rate, respectively. Finally, $\{b_i\}_{i=1}^{3}$ refers to the proportion of population decline caused by one predator competing with another. In this system, we assume that predators corresponding to u and w do not compete with each other.

We aim to approximate the monotone traveling wave solution for the above system through the definite integral of positive functions. The traveling wave solution, a particular solution of partial differential equations (PDE), may exist in several forms for a single equation because there is no constraint on the wave speed. The most common case is when traveling wave solutions always exist for speeds higher than a specific value, called minimum wave speed. The exact minimum wave speed was revealed in the case of the Keller-Segel model with logistic growth dynamics [1]. The authors in [2] investigated the presence of a minimum wave speed and a sharp bound for a non-KPP type reaction-diffusion system. The authors in [3] identified the necessary and sufficient conditions for the maximum delay time to assure the existence of the traveling wave solution to the Lotka-Volterra competition system. It turns out that only monotone solutions are possible in this system.

In contrast to the previous examples, there are several equations where a solution and wave speed are uniquely determined. A monotone traveling wave solution exists uniquely for the Keller-Segel model with a chemotactic sensitivity term in the form of a logarithmic function [4]. The same results hold for the Allen-Cahn model with relaxation terms and Lotka Volterra competition diffusion system [5, 6]. When several functions can satisfy constraints, it is challenging to predict to which solution the trained neural network will converge. Since the three-species competition systems covered in this paper ensure the uniqueness of the solution under the assumption of monotonicity, we solve the well-posed forward-inverse problem that simultaneously approximates a solution and speed.

For all positive integers n, shallow neural networks without bias are dense in $C^n(\cdot)$, representing the set of functions whose all n th partial derivatives are continuous [7]. It suggests that neural networks can approximate the PDE or ODE solution with an arbitrarily small error. Modifying a fully connected neural network can also impose boundedness or monotonicity on neural networks. Therefore, the deep neural network (DNN) approach becomes a flexible method for solving differential equations with additional constraints.

1.2. Related works

The authors in [8] presented a physics informed neural networks (PINN) in which neural networks simultaneously learn solutions and coefficients of partial differential equations. They employed automatic differentiation [9] in computing the derivatives of neural networks and optimized the mean squared loss corresponding to initial and boundary conditions. Since it can estimate a high-order differential coefficient of neural networks at all points, the method does not have a constraint to be trained on a fixed grid.

Several recent studies have shown that minimizing the loss function designed above is equivalent to finding an approximator near a solution in terms of pointwise error. The authors in [10] first proved the existence of a neural network with a sufficiently small loss for quasilinear parabolic partial differential equations. Assuming the regularity of the solution, they concluded that the neural network eventually converges to the solution when the residual loss decreases to zero. Subsequently, studies showing that the neural network approximation sequence converges to a solution of PDE through the Gronwall inequality have emerged. The authors in [11] discussed the 1-dimensional kinetic Fokker-Planck equation for inflow and specular boundary conditions. The second-order parabolic equation of the zero Dirichlet boundary problem was analyzed in [12]. The authors in [13] employed monotonicity to set the point where the solution necessarily passes and perform estimates of traveling wave solutions for various equations. Furthermore, estimates for the unique continuation problem, including additional observation of the ill-posed forward problem, have also been recently studied. The authors in [14] identified the number of training points required to reduce residual loss, considering the errors which arise from the quadrature rule.

1.3. Outline of the paper

In Section 2, we introduce the structure of a fully connected neural network model. We construct a residual loss to approximate the solution of the system and describe the monotonic neural network. In Section 3, existing theoretical results are first provided, such as the uniqueness of the traveling wave solution and the sign of wave speed. Then we prove that our model can minimize the loss function and that the sequence of DNN solutions approaches the actual solution. In section 4, corresponding experiments of several methods are addressed, that impose monotonicity. We offer the shortcomings of each method and experimental results, which prove that our model can overcome them. Finally, we conclude the article by summarizing the results and discussing how to supplement the paper and develop future research topics.

2. Preliminaries

In this section, we describe our neural network model to approximate the solutions of the system and the wave speed. The construction of the loss function is fundamentally based on PINN [8] and follows the detailed settings in [13].

Consider the system (1.1) with the traveling wave ansatz

$$(u(t,x), v(t,x), w(t,x)) = (U(z), v(z), W(z)), \quad z = x - st,$$

where s denotes the wave speed. Then, traveling wave solution $(U, V, W)(z)$ of the system (1.1) with the boundary condition (1.2) is a solution of the following system.

$$\begin{aligned} -sU_z &= d_1U_{zz} + a_1U(1 - U - b_2V), \\ -sV_z &= d_2V_{zz} + a_2V(1 - V - b_1U - b_3W), \\ -sW_z &= d_3W_{zz} + a_3W(1 - W - b_2V), \end{aligned} \tag{2.1}$$

$$\text{where} \quad (U, V, W)(z) \to (u_-, v_-, w_-) \quad \text{as } z \to -\infty, \quad (U, V, W)(z) \to (u_+, v_+, w_+) \quad \text{as } z \to \infty. \tag{2.2}$$

In this paper, we set the two constant states (u_-, v_-, w_-) and (u_+, v_+, w_+) in the asymptotic boundary conditions (2.2) to $(1, 0, 1)$ and $(0, 1, 0)$ respectively. If the sign of wave speed s is positive, the predator corresponding to V disappears, and the predators corresponding to U and W, which do not compete with each other, survive. In the opposite case, only the predator corresponding to V survives. It should be noted that in this research, we are only concerned with the study of monotone traveling wave solutions. That is, $U_z < 0, V_z > 0, W_z < 0$.

2.1. Neural network model & Loss function

The fully connected neural network U^{nn}, an approximator of the solution $U(t, x)$, is formulated as a recurrence relation with a linear affine transformation and a nonlinear activation function σ.

$$U^{nn}(t, x) = W_L \cdot \sigma(W_{L-1} \cdots \sigma(W_1 \cdot (x - s^{nn}t) + b_1) + b_{L-1}) + b_L.$$

$\left\{W_i \in \mathbb{R}^{h_i} \times \mathbb{R}^{h_{i+1}}\right\}_{i=1}^{L}$ denotes the sequence of matrices where $\{h_i\}$ denotes the number of hidden neurons in each layer. Since h_1 and h_{L+1} denote the dimension of input and output respectively, they are set to 1. s^{nn} is an approximator of the wave speed s in our method, which varies by optimizing the loss functions provided below. The other two networks, V^{nn} and W^{nn}, are defined similarly.

Now we define the following three operators $P(U, V, W; s)$, $Q(U, V, W; s)$ and $R(U, V, W; s)$.

$$\begin{aligned} P(U, V, W; s) &:= sU_z + d_1U_{zz} + a_1U(1 - U - b_2V), \\ Q(U, V, W; s) &:= sV_z + d_2V_{zz} + a_2V(1 - V - b_1U - b_3W), \\ R(U, V, W; s) &:= sW_z + d_3W_{zz} + a_3W(1 - W - b_2V). \end{aligned}$$

We use the truncated interval as the domain instead because we cannot consider all the points on a infinite real line. For a sufficiently large interval $[-L, L]$, the residual loss function for each governing equation in (2.1) is defined by

$$\begin{aligned} Loss_{GE}^{(1)} &= \int_{-L}^{L} (P(U^{nn}, V^{nn}, W^{nn}; s^{nn}))^2 dz \approx \sum_i (P(U^{nn}(z_i), V^{nn}(z_i), W^{nn}(z_i); s^{nn}))^2, \\ Loss_{GE}^{(2)} &= \int_{-L}^{L} (Q(U^{nn}, V^{nn}, W^{nn}; s^{nn}))^2 dz \approx \sum_i (Q(U^{nn}(z_i), V^{nn}(z_i), W^{nn}(z_i); s^{nn}))^2, \\ Loss_{GE}^{(3)} &= \int_{-L}^{L} (R(U^{nn}, V^{nn}, W^{nn}; s^{nn}))^2 dz \approx \sum_i (R(U^{nn}(z_i), V^{nn}(z_i), W^{nn}(z_i); s^{nn}))^2, \\ Loss_{GE} &= Loss_{GE}^{(1)} + Loss_{GE}^{(2)} + Loss_{GE}^{(3)}, \end{aligned}$$

where $\{z_i\}_i$ represents the set of sample points over the spatial domain used for computing the integral expression of the mean squared loss. At each iteration of the training process, new points are randomly selected for use in the Monte Carlo integration method.

It is also challenging to consider asymptotic boundary conditions on the real line. Therefore, we assign boundary conditions to both endpoints of the truncated domain instead. The residual boundary loss function is formulated as follows.

$$\begin{aligned}
Loss^{(1)}_{limit} &= (U^{nn}(-L) - u_-)^2 + (U^{nn}(L) - u_+)^2,\\
Loss^{(2)}_{limit} &= (V^{nn}(-L) - v_-)^2 + (V^{nn}(L) - v_+)^2,\\
Loss^{(3)}_{limit} &= (W^{nn}(-L) - w_-)^2 + (W^{nn}(L) - w_+)^2,\\
Loss_{limit} &= Loss^{(1)}_{limit} + Loss^{(2)}_{limit} + Loss^{(3)}_{limit}.
\end{aligned}$$

In this paper, we propose a Neumann boundary condition the solution inherently satisfies. Multiplying both sides of the first equation in (2.1) by U_z and integrating them over a truncated interval $[-c, b]$, we can derive the convergence of U_z through the following computation. First, the last term on the right side of the equation must be finite when b goes to infinity.

$$\int_{-c}^{b} UU_z - U^2U_z - b_2UU_zVdz = \frac{1}{2}(U^2(b) - U^2(-c)) - \frac{1}{3}(U^3(b) - U^3(-c)) - \int_{-c}^{b} b_2UU_zVdz,$$

where last integral must be finite since $V(z)$ should be less than 1 and UU_z is a negative integrable function. Therefore, the definite integral $\int_{-c}^{b} sU_z^2 + d_1U_zU_{zz}dz = \int_{-c}^{b} sU_z^2dz + d_1(U_z^2(b) - U_z^2(-c))/2$ must converge.

When we consider the asymptotic boundary condition, the only possibility is that $U_z(b)$ converges to zero regardless of the sign of s. For $s = 0$, the convergence of $U_z(b)$ permits the case only when $U_z(+\infty) = 0$. When s is positive, the increasing function $\int_{c}^{b} sU_z^2dz$ should converge so that $1/2U_z(b)^2$ also converge. For negative s, we focus on the first equation of (2.1). Since $sU'(z) + d_1U''(z)$ goes to zero, there exists an arbitrary small $\epsilon > 0$ such that for all sufficiently large z,

$$sU'(z) + d_1U''(z) < \frac{d_1}{2}\epsilon.$$

Suppose that there exists large z_1 such that $U'(z_1) < (d_1/s)\epsilon$. Then, the inequality $d_1\epsilon + d_1U''(z_1) < sU'(z_1) + d_1U''(z_1) < \frac{d_1}{2}\epsilon$ results in the negativity of $U''(z_1)$. This contradicts to the asymptotic boundary condition (2.2) since $U'(z)$ cannot maintain values indefinitely below a certain negative constant. Therefore, we put forward the following Neumann boundary condition (2.3).

$$\lim_{\xi\to\infty} U_z(\xi) = \lim_{\xi\to\infty} V_z(\xi) = \lim_{\xi\to\infty} W_z(\xi) = \lim_{\xi\to\infty} U_z(-\xi) = \lim_{\xi\to\infty} V_z(-\xi) = \lim_{\xi\to\infty} W_z(-\xi) = 0. \tag{2.3}$$

The loss function for the boundary condition of the derivative is defined as follows.

$$\begin{aligned}
Loss^{(1)}_{BC} &= (U_z^{nn}(-L))^2 + (U_z^{nn}(L))^2,\\
Loss^{(2)}_{BC} &= (V_z^{nn}(-L))^2 + (V_z^{nn}(L))^2,\\
Loss^{(3)}_{BC} &= (W_z^{nn}(-L))^2 + (W_z^{nn}(L))^2,
\end{aligned}$$

$$Loss_{BC} = Loss_{BC}^{(1)} + Loss_{BC}^{(2)} + Loss_{BC}^{(3)}.$$

We note that the translation of a traveling wave solution is still a traveling wave solution. To consider a wide interval on the real line in both directions, it is necessary to fix a point through which the solution must pass. If the monotone function's left and right limits on the real line are 0 and 1 respectively, the range of the function must contain $1/2 \in \mathbb{R}$. For the symmetric interval $[-L, L]$ about the origin, it would be natural to enforce a value of $1/2$ at $x = 0$.

$$Loss_{trans} = \left(U^{nn}(0) - \frac{u_- + u_+}{2}\right)^2 = \left(U^{nn}(0) - \frac{1}{2}\right)^2.$$

Finally, we optimize the total loss function aggregating the above losses. Changes in weights included in neural networks follow the ADAM [15] based on gradient descent.

$$Loss_{total} = Loss_{GE} + Loss_{limit} + Loss_{BC} + Loss_{trans}.$$

2.2. *Monotonic neural network*

The original monotonic network in [16] was constructed by combining positive constrained weights and min-max operation. In the case of simply forcing the positivity of the weights, a neural network with a popular activation function, such as ReLU [17] or ELU [18], can only represent a convex function [19]. In contrast, the above min-max monotonous neural network has universal properties for monotone functions in $C^1([0, 1]^D; \mathbb{R})$. Some studies propose a method of considering the regularizer for monotonicity [20]. The authors in [21] revealed that in multi-dimensional spaces, the upper envelope of the function must be a monotone function. They employed a monotone function in prediction by directly constructing the upper envelope of the counterexample.

In this paper, we transform the unconstrained monotonic neural network [22], which devised the integration of the positive function. By replacing numerical integration with DeepONet-based operator learning without using the quadrature rule, the neural network becomes a smoother function suitable for representing solutions to differential equations.

We generate m variables $\{p_i\}_{i=1}^m$ to represent the values of a positive function at m positions $\{x_i\}_{i=1}^m$. To find a fully connected neural network representing the primitive function, we perform operator learning for the integration based on DeepONet. The overall structure of our neural network model is formulated as follows.

$$U^{mon}(z) := \langle \text{Branch Net}(p_1, \cdots, p_m), \text{Trunk Net}(z) \rangle$$

It is worth noting that the structure of this method is the same as that of DeepONet, which is capable of learning general operators. In this research, it is specifically designed to learn integral operators. We construct a monotonic neural network model U^{mon} as an inner product of Branch Net and Trunk Net, which are fully connected neural networks in Section 2.1. Branch Net receives a discretization of function as an input to determine which positive function to integrate. Trunk Net takes the point of the domain as input and queries the location of interest. Due to the high expressivity for compact operators, combining the two networks will improve the approximation possibility for monotone functions. We remark that Branch Net and Trunk Net are called shallow networks when they are two-layered fully connected neural networks (i.e. $L = 2$).

3. Theory

In this section, we introduce existing theories about the system and provide results on the convergence of our model. The first part discusses the assumption that guarantees uniqueness other than the initial and boundary conditions. The following part addresses the theory obtained based on DeepONet's universality [23] and Grönwall inequality.

3.1. Existing theoretical results

In the case of an original bistable three-species competition System (1.1), we cannot guarantee the monotonicity and uniqueness of traveling wave solutions. The authors in [24] additionally assume one of the two following conditions.

$$(1) \quad U' < 0, V' > 0, W' < 0, \quad \text{or} \quad (2) \quad 0 < U, V, W < 1.$$

It may be straightforward to show that the former condition implies the latter when U, V, W are traveling wave solutions of the system (1.1). The authors in [24] remarked that the above two inequalities are indeed equivalent conditions. Under the above assumption, the following theorem states the uniqueness of a traveling wave solution.

Theorem 1. *(Thm 3.1 in [24]) The three-species competition system (1.1) has a traveling wave solution that is unique up to translation with the unique wave speed.*

In several cases, the sign of wave speed was revealed without knowing the exact value. The theorem below presents that U and W should become extinct when the predators corresponding to U and W are relatively less competitive than the predator corresponding to V.

Theorem 2. *(Thm 3.2 in [24]) Suppose that $a_1 = d_1, a_3 = d_3, b_3 = b_1$. If $0 < b_1 < 1 < 2 \leq b_2$, then the sign of wave speed s for the system (1.1) is negative.*

Similar to the above, there is a theorem describing the sign of wave speed for various parameter settings. In particular, the theorem provides the presence of a solution with a wave speed of zero. If the condition $b_2 = 2b_1$ is satisfied, it will be used as a validation example to verify the accuracy of proposed method.

Theorem 3. *(Thm 3.4 in [24]) Suppose that $a_1 = d_1 = a_3 = d_3$ and $0 < b_1 < 1 < b_2 < 2$. Then, the sign of wave speed s for the system (1.1) is the same as the sign of $(2b_1 - b_2)$. In particular, the wave speed s is precisely zero when b_2 is identical to $2b_1$.*

3.2. Convergence of neural network model

We now introduce theoretical results on the performance of our model when we optimize the loss introduced in Section 2.1. Our monotonic neural network in Section 2.2 replaces the original fully connected neural network. When approximating each traveling wave solution, we maintain the weights of Branch Net and Trunk Net, and only a sequence $\{p_i\}_{i=1}^{m}$ depends on the solution. In the above situation, our model capabilities embrace uniform convergence.

Corollary 1. *Assume that an activation function σ is continuous and non-polynomial. Let $S(U) \in C^1([-L, L])$ denote the set which contains traveling wave solutions U of the first predator in the system (2.1). Suppose the set of derivatives $D(U)$ of elements in $S(U)$ is compact in $C([-L, L])$. Then for any arbitrary small $\epsilon > 0$, there exists a large positive integer m with shallow Branch Net and Trunk Net such that for any $U \in S(U)$, our monotonic neural network $U^{mon}(z)(= \langle Branch\ Net(p_1, \cdots, p_m), Trunk\ Net(z)\rangle)$ with an appropriate sequence $\{p_i\}_{i=1}^m$ satisfies the following.*

$$||U^{mon}(z) - U(z)||_{L^\infty([-L,L])} < \epsilon.$$

Proof. The compactness of the set of derivatives implies the compactness of $S(U)$ since the integral operator is continuous when the domain of functions is a finite closed interval. Define a operator $G : D(U) \to S(U)$ as $G(f)(z) = \int_L^z f dz$ for $z \in [-L, L]$. As in Theorem 5 in [23], we can choose m grid points $\{x_i\}_{i=1}^m$ for G which guarantees

$$||\langle \text{Branch Net}(f(x_1), \cdots, f(x_m)), \text{Trunk Net}(z)\rangle - G(f)(z)||_{L^\infty([-L,L])} < \epsilon, \quad \forall f \in D(U).$$

We then achieve the desired result by setting f and $\{f(x_i)\}_{i=1}^m$ to $U'(z)$ and $\{p_i\}_{i=1}^m$, respectively. □

Remark 1. *As a representative example of $D(U)$ that satisfies the requirement, we have considered the case of a finite-dimensional space $S(U)$ with bounded properties, which can be achieved through the use of sequentially compact bounded intervals. However, it should be noted that due to the unbounded differential values of the monotone traveling wave solution, it is not possible to search the entire traveling wave solution within the solution space using a single DeepONet. The size of $D(U)$ is determined by the user of the theorem. The assumption becomes necessary from the fact that a single DeepONet structure, comprising of parameters with appropriate values, was employed to approximate the integration of multiple functions.*

Proposition 1 suggests that if DeepONet approximates the integral operator with proper performance, our monotonic neural network does not deviate significantly from the possible solutions. In other words, we can assume the boundness of the approximator when the loss for operator learning becomes sufficiently reduced. Given the optimized residual loss with the assumption, we estimated the pointwise error for the solution of the approximator. Note that the same results hold for fully connected neural networks where s^{mon} denotes our method's approximation of speed s.

Theorem 4. *Suppose that our monotonic networks $U^{mon}, V^{mon}, W^{mon}$ are bounded by a constant C. Denote $|(U(z), V(z), W(z)) - (U^{mon}(z), V^{mon}(z), W^{mon}(z))|$ by $E(z)$. Then, the following holds.*

$$E(z) \leq K_1 exp(K_2(z + L)), \forall z \in [-L, L],$$

where K_1, K_2 are constants depending on $E(-L)$, $Loss_{GE}$, $Loss_{BC}$, $Loss_{limit}$, L, s, s^{mon}, and the parameters of the system (2.1).

Proof. Integrating the first equation of the system (2.1), we derive the followings.

$$U'(z) = -\frac{s}{d_1}U(z) + \frac{s}{d_1}U'(-L) - \frac{a_1}{d_1}\int_{-L}^{z} U(1 - U - b_2V)dz := F(z, U),$$

$$V'(z) = -\frac{s}{d_2}V(z) + \frac{s}{d_2}V'(-L) - \frac{a_2}{d_2}\int_{-L}^{z} V(1 - V - b_1U - b_3W)dz := G(z, V),$$
$$W'(z) = -\frac{s}{d_3}W(z) + \frac{s}{d_3}W'(-L) - \frac{a_3}{d_3}\int_{-L}^{z} W(1 - W - b_2V)dz := H(z, W).$$

With the residual loss for governing equation, our monotonic neural network satisfies a similar system above. The detailed form of a system can be written as follows.

$$\begin{aligned}
(U^{mon})'(z) &= -\frac{s^{mon}}{d_1}U^{mon}(z) + \frac{s^{mon}}{d_1}(U^{mon})'(-L) - \frac{a_1}{d_1}\int_{-L}^{z} U^{mon}(1 - U^{mon} - b_2V^{mon})dz + \frac{1}{d_1}\int_{-L}^{z} f(y)dy \\
&:= F^{mon}(z, U^{mon}), \\
(V^{mon})'(z) &= -\frac{s^{mon}}{d_2}V^{mon}(z) + \frac{s^{mon}}{d_2}(V^{mon})'(-L) - \frac{a_2}{d_2}\int_{-L}^{z} V^{mon}(1 - V^{mon} - b_1U^{mon} - b_3W^{mon})dz \\
&+ \frac{1}{d_2}\int_{-L}^{z} g(y)dy := G^{mon}(z, V^{mon}), \\
(W^{mon})'(z) &= -\frac{s^{mon}}{d_3}W^{mon}(z) + \frac{s^{mon}}{d_3}(W^{mon})'(-L) - \frac{a_3}{d_3}\int_{-L}^{z} W^{mon}(1 - W^{mon} - b_2V^{mon})dz + \frac{1}{d_3}\int_{-L}^{z} h(y)dy \\
&:= H^{mon}(z, W^{mon}),
\end{aligned}$$

where $\int_{-L}^{L} \{f(y)\}^2 + \{g(y)\}^2 + \{h(y)\}^2 dy = Loss_{GE}$ as in Section 2.1. We now leverage the boundedness of the solutions and triangular inequalities to estimate how similarly U and U^{mon} follow the system of ordinary differential equations. First of all, the difference in the first equation of the system can be decomposed into three terms as follows.

$$\begin{aligned}
F(z, U) - F(z, U^{mon}) &= -\frac{s}{d_1}(U - U^{mon})(z) + \frac{s}{d_1}(U'(-L) - (U^{mon})'(-L)) \\
&- \frac{a_1}{d_1}\int_{-L}^{z}(U - U^{mon} - (U^2 - (U^{mon})^2) - b_2UV + b_2U^{mon}V^{mon})dz \\
&:= f_1 + f_2 + f_3
\end{aligned}$$

The last term f_3 can be bounded by a constant via the boundedness of solutions.

$$\begin{aligned}
|f_3| &\le \frac{a_1}{d_1}\int_{-L}^{z} |U - U^{mon}| + |U^2 - (U^{mon})^2| + b_2|UV - U^{mon}V^{mon}|dz \\
&\le \frac{a_1}{d_1}\int_{-L}^{z} |U - U^{mon}| + |U - U^{mon}||U + U^{mon}| + b_2|U - U^{mon}||V| + b_2|U^{mon}||V - V^{mon}|dz \\
&\le 2L\left(\frac{a_1}{d_1}\right)((C + 1) + (C + 1)^2 + b_2(C + 1)^2)
\end{aligned}$$

Therefore, we complete the estimates of $|F(z, U) - F(z, U^{mon})|$ by triangular inequalities.

$$\begin{aligned}
&|F(z, U) - F(z, U^{mon})| \\
&\le \frac{s}{d_1}|U - U^{mon}|(z) + \frac{s}{d_1}|U' - (U^{mon})'|(-L) + 2L\left(\frac{a_1}{d_1}\right)((C + 1) + (C + 1)^2 + b_2(C + 1)^2)
\end{aligned}$$

Estimates of the second and third equations can be obtained by using a similar manner. We combine all the results to obtain the following inequality,

$$\begin{aligned}&|(F(z,U),G(z,V),H(z,W))-(F(z,U^{mon}),G(z,V^{mon}),H(z,W^{mon}))|\\&\leq C_1|(U(z),V(z),W(z))-(U^{mon}(z),V^{mon}(z),W^{mon}(z))|+C_2,\end{aligned}$$

where a constant C_1 depends on s and the diffusion coefficients of system and a constant C_2 depends on the parameters of system, $|U'-(U^{mon})'|(-L)$, L and C.

Finally, we would like to complete the proof by presenting the upper bound for the growth of the error function $E(z)$.

$$\begin{aligned}&E(z)-E(-L)\\&=\int_{-L}^{z}|(F(y,U),G(y,V),H(y,W))-(F^{mon}(y,U^{mon}),G^{mon}(y,V^{mon}),H^{mon}(y,W^{mon}))|dy\\&\leq\int_{-L}^{z}|(F(y,U),G(y,V),H(y,W))-(F(y,U^{mon}),G(y,V^{mon}),H(y,W^{mon}))|dy\\&+\int_{-L}^{z}|(F(y,U^{mon}),G(y,V^{mon}),H(y,W^{mon}))-(F^{mon}(y,U^{mon}),G^{mon}(y,V^{mon}),H^{mon}(y,W^{mon}))|dy\\&\leq\int_{-L}^{z}C_1E(y)+C_2dy+\int_{-L}^{z}\frac{(f(y))^2}{d_1}+\frac{(g(y))^2}{d_2}+\frac{(h(y))^2}{d_3}+C_3|s-s^{mon}|dy\end{aligned}$$

where C_3 is a constant depending on C, $(U^{mon})'(-L)$, and the diffusion coefficients of the system. Therefore, we can get the following inequalities.

$$\begin{aligned}E(z)&\leq E(-L)+\int_{-L}^{z}C_1E(y)+C_2dy+\int_{-L}^{z}\frac{(f(y))^2}{d_1}+\frac{(g(y))^2}{d_2}+\frac{(h(y))^2}{d_3}+C_3|s-s^{mon}|dy\\&\leq E(-L)+2LC_2+\int_{-L}^{L}\frac{(f(y))^2}{d_1}+\frac{(g(y))^2}{d_2}+\frac{(h(y))^2}{d_3}+C_3|s-s^{mon}|dy+\int_{-L}^{z}C_1E(y)dy\end{aligned}$$

By the Grönwall's inequality in section 17.3 of [25] with two constants $E(-L)+2LC_2+\int_{-L}^{L}\frac{(f(y))^2}{d_1}+\frac{(g(y))^2}{d_2}+\frac{(h(y))^2}{d_3}+C_3|s-s^{mon}|dy$ and C_1,

$$E(z)\leq(E(-L)+2LC_2+\int_{-L}^{L}\frac{(f(y))^2}{d_1}+\frac{(g(y))^2}{d_2}+\frac{(h(y))^2}{d_3}+C_3|s-s^{mon}|dy)\exp(C_1(z+L))$$

□

We remark that Grönwall's inequality can be modified by considering the opposite direction. The estimate above states that the trained neural network effectively represents the solution near the boundary, assuming that s^{mon} approximates s within a small error. We conclude this section by introducing the existence of our monotonic network model that reduces the $Loss_{total}$ for the solution of the finite-dimensional space.

Theorem 5. *Assume that an activation function σ is continuous and non-polynomial. Consider a finite-dimensional space $O\subset C^2([-L,L])$ which is bounded. Then for any positive ϵ, there exists a large positive integer m with shallow Branch Net and Trunk Net such that for any traveling wave solution (U,V,W) of 2.1 where $U,V,W\in O$, our monotonic neural network $U^{mon}(z)(=$*

$\langle$Branch Net$(p_1, \cdots, p_m)$, Trunk Net$(z)\rangle$) with an appropriate sequence $\{p_i\}_{i=1}^m$ satisfy the following.

$$Loss_{total} < \epsilon.$$

Proof. For any $\epsilon > 0$, we first note that Trunk Net learns the basis of the function space, and Branch Net learns the coefficients for each basis. Denote the ornormal basis of O by $\{o_i\}_{i=1}^n$. By universal properties of neural network in [7], there exists a sequence of neural networks $\{NN_i\}_{i=1}^n$ such that

$$\sum_{i=1}^n \sum_{k=0}^2 \left\| \frac{d^k}{dz^k}(o_i) - \frac{d^k}{dz^k}(NN_i) \right\|_{L^\infty([-L,L])} \leq \epsilon.$$

Suppose that Trunk Net represents the NN_i. Then by the Theorem 5 in [23], we can choose m values $\{p_i\}_{i=1}^m$ as in the proof of proposition 1 which guarantees

$$\|\langle \text{Branch Net}(p_1, \cdots, p_m), \text{Trunk Net}(z)\rangle - U(z)\|_{L^\infty([-L,L])} < \epsilon, \quad \forall U \in O.$$

Since $\{NN_i\}_{i=1}^n$ is the approximator of $\{o_i\}_{i=1}^n$ in $C^2([-L, L])$, the above inequality implies that Branch Net estimates the appropriate coefficients, resulting the uniform convergence in $C^2([-L, L])$ for all $U \in O$.

If $(U^{mon}, V^{mon}, W^{mon})$ are efficient approximations of (U, V, W) in $C^2([-L, L])$, $Loss_{limit}$, $Loss_{BC}$ and $Loss_{trans}$ should be small (t.e. less than a constant multiple of ϵ). For the term $Loss_{GE}$, we only consider the loss function $Loss_{GE}^{(1)}$ for the first equation in the governing equation. For the traveling wave solution (U, V, W) of the system (2.1),

$$\begin{aligned}
&sU_z^{mon} + d_1 U_{zz}^{mon} + a_1 U^{mon}(1 - U^{mon} - b_2 V^{mon}) \\
&= sU_z^{mon} + d_1 U_{zz}^{mon} + a_1 U^{mon}(1 - U^{mon} - b_2 V^{mon}) - (sU_z + d_1 U_{zz} + a_1 U(1 - U - b_2 V)) \\
&= s(U_z^{mon} - U_z) + d_1(U_{zz}^{mon} - U_{zz}) + a_1(U^{mon} - U) - a_1((U^{mon})^2 - U^2) - a_1 b_2(U^{mon}V^{mon} - UV).
\end{aligned}$$

Since we may assume the boundedness of $(U^{mon}, V^{mon}, W^{mon})$ by Corollary 1, we can decompose the nonlinear product term in the system (2.1) as $|UV - U^{mon}V^{mon}| \leq |U - U^{mon}||V| + |V - V^{mon}||U^{mon}|$. By observing this type of triangular inequality, we can conclude that the convergence in $C^2([-L, L])$ indeed guarantees an arbitrary small $Loss_{GE}$. □

4. Experiments

In this section, we present experimental results that our model can be more accurately trained than previous min-max monotonic networks [16] or fully connected neural networks. Before interpreting the tables and figures, we describe experimental settings in detail. In all cases, the ADAM optimization algorithm, as outlined in the literature by Kingma and Ba [15], was utilized with a learning rate of 1e-2 to minimize the loss function. By employing the StepLR scheduler, the learning rate decreased by 0.9 per 300 epochs. All experimental results were adopted using an optimal model based on the value of the loss function in training up to 10000 epochs. The 5-layered neural network has 48 hidden neurons for each hidden layer. The initial weights were set in accordance with the default initialization method provided by PyTorch.

Training of the DeepONet in Section 2.2 utilized the 20th-order Chebyshev polynomials. The model learns the integral of Chebyshev polynomials in the closed interval [0, 1] and derives the primitive function on [-50, 50] using a linear transformation. For the discretization of the function on 300 points, a dataset was generated by odeint implemented in the module named Scipy in Python. Trunk Net and Branch Net are 5-layer neural networks with a width of 256 and a latent dimension of 64. For the integration, we used an Adam optimizer and StepLR scheduler with a learning rate of 1e-3 and a rate of 0.9.

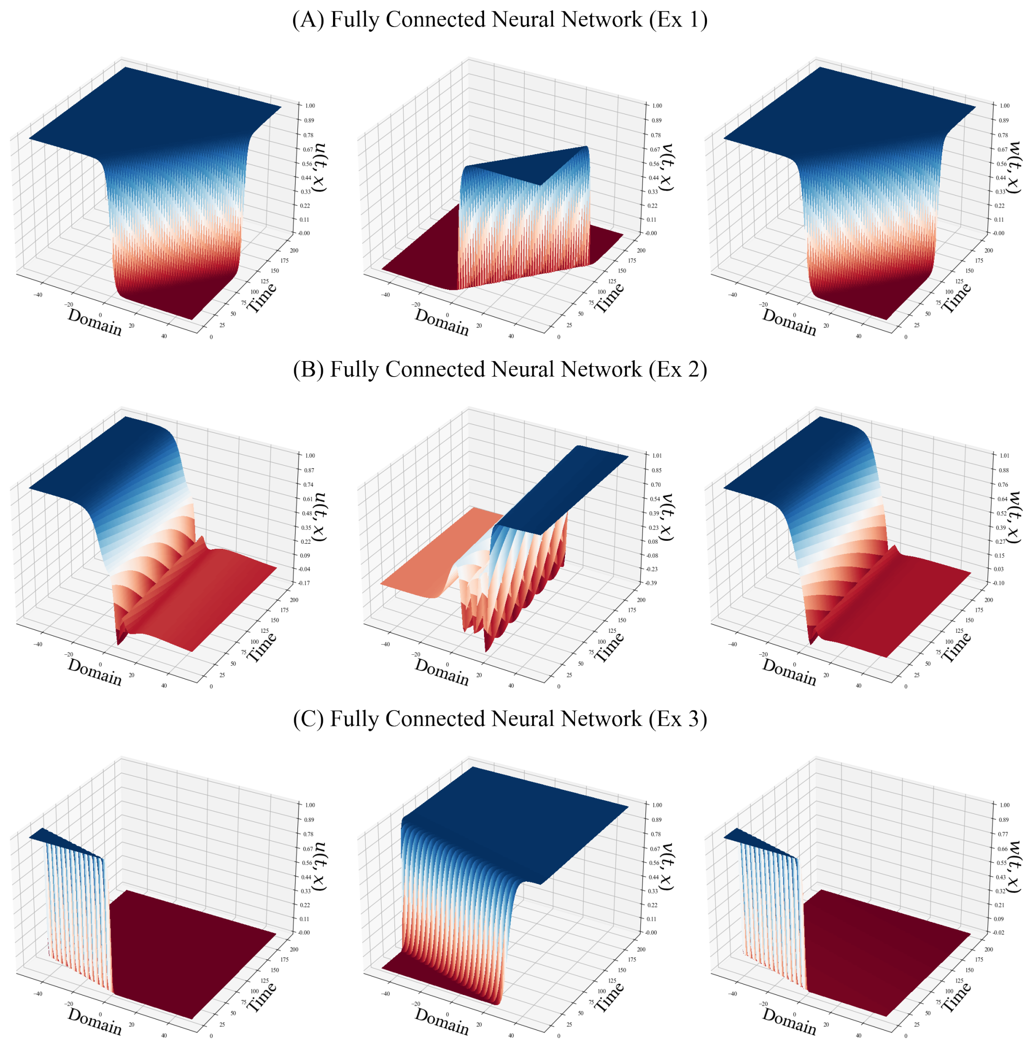

Figure 1. Approximations of the solutions to the system (2.1) via three neural networks.

If we optimize only the residual loss of Section 2.1 without any restrictions, the fully connected neural network proceeds with unstable training. Figure 1 shows the results of the convergence of the fully connected neural network for three different seeds provided by Pytorch and Numpy. Theorem 3 implies that the wave speed is precisely zero for $(a_1, d_1, a_3, d_3, b_1, b_2, b_3) = (1, 1, 1, 0.6, 1.2, 0.6)$. Although the fully connected neural network seems to extract the corresponding wave speed in the second experiment, the approximations do not have monotonicity. In the case of the first and third experiments, as seen in Table 1, the model achieves the approximations with a small loss function. This result raises the possibility that the system without monotonicity may admits another solution.

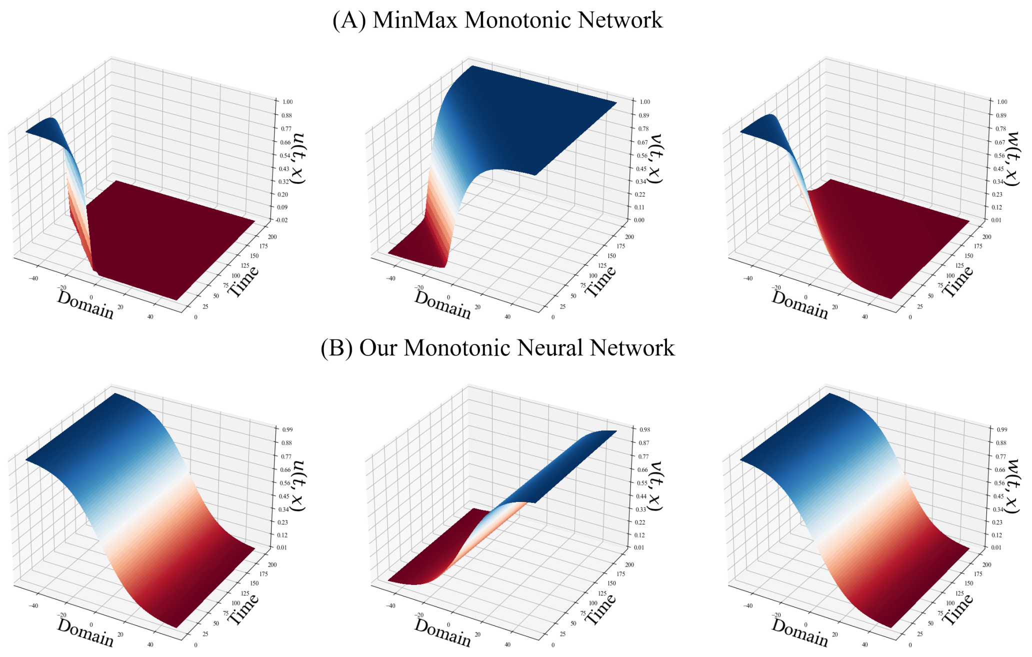

Figure 2. Approximations of the solutions to the system (2.1) via two types of monotonic networks.

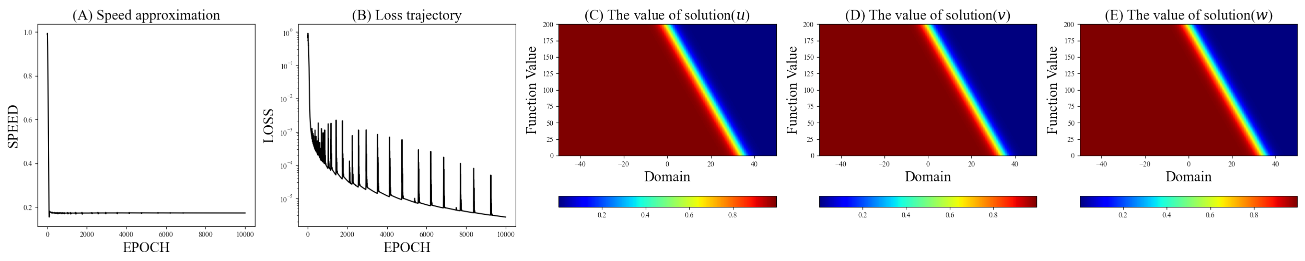

Figure 3. Training trajectory and our approximations of the solutions to the system (2.1) via three neural networks for positive wave speed.

Table 1. Experimental results of three neural network models for $(a_1, d_1, a_3, d_3, b_1, b_2, b_3) = (1, 1, 1, 1, 0.6, 1.2, 0.6)$.

Exact Speed(0)	Fully Connected Neural Network			min-max Monotonic Network	Our model
	Experiment 1	Experiment 2	Experiment 3		
Estimated Speed	4.36e-01	-5.84e-02	-1.80e-00	-9.80e-02	2.61e-03
Loss	1.66e-07	1.77e-03	3.66e-07	2.38e-01	5.60e-06

In contrast to the above, our method showed outstanding performance. Figure 2(B) shows that our model indeed approximates smooth function. By observing that our model predicts an accurate speed with a small loss in Table 1, the results of Theorem 4 suggest that our model exists near the solution. min-max monotonic networks have failed to predict the exact speed and reduce loss. We analyzed that the network cannot satisfy the governing equation because max or min operation hinders differentiability.

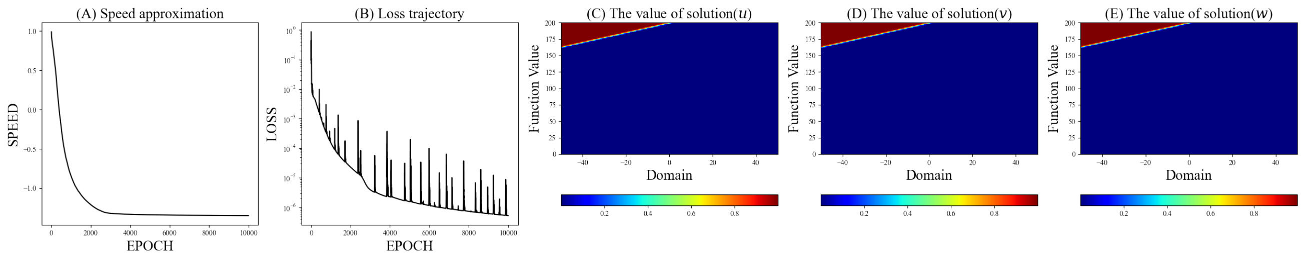

Figure 4. Training trajectory and our approximations of the solutions to the system (2.1) via three neural networks for negative wave speed.

When the sign is known instead of the exact value of wave speed, our model has performed feasible predictions. Figures 3 and 4 show the experimental results of our model when $(a_1, d_1, a_3, d_3, b_1, b_2, b_3)$ is determined to (1, 1, 1, 1, 0.75, 0.75, 1.25) and (1, 1, 1, 1, 0.75, 0.75, 1.75), respectively. Our model predicted the correct sign, and the loss function converged to zero for the above cases. However, stabilization of learning requires more hyperparameter search. The learning and decay rates used in this paper are searched by performing a grid search on $Loss_{Total}$. The segmentation and extension of the set of hyperparameters may contribute to the convergence of the model.

Finally, for analysis of negative wave speeds related to Theorem 2, we set $(a_1, d_1, a_3, d_3, b_1, b_2, b_3)$ to $(1, 1, 1, 1, 0.99, 0.99, 2.01)$. In this case, Figure 5 shows that the solution is challenging to approximate through our methods. Although the estimated speed seems to converge to -0.2, we can observe a phenomenon in which the value of the loss function increases rapidly before reaching an acceptable level of convergence. We observed that in the majority of experiments, it was necessary for the loss function values to be less than 1e-5 to ensure convergence to accurate solutions. The experimental results presented in Table 1 demonstrate that The training of neural networks with loss function values around 1e-3 can result in a failure to predict an adequate wave speed. This raised the question of whether our method indeed approximates real solutions in this negative speed case.

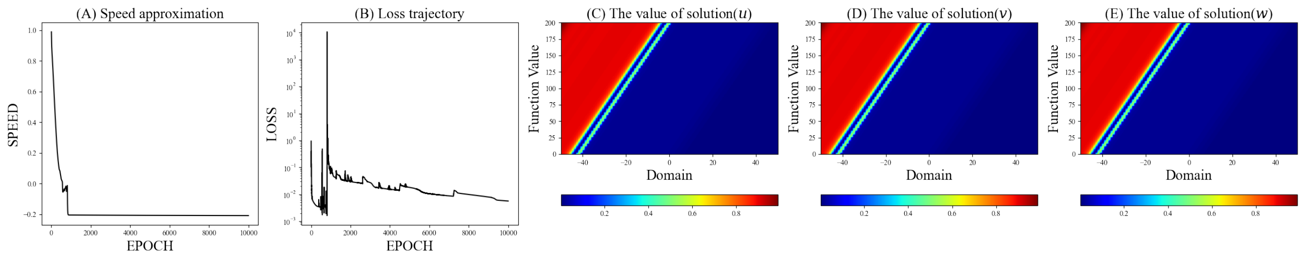

Figure 5. Training trajectory and our approximations of the solutions to the system (2.1) via three neural networks for negative wave speed in Theorem 3.

5. Conclusions

In this paper, we construct the monotonic neural network to impose constraints on the monotonicity. In all experiments, our method performed better than other methods, suggesting that it effectively represents real solutions based on a theoretical basis.

To verify that this approach is universal, we require supplementary experiments on various equations in which a unique monotone solution exists. We anticipate that the proof techniques or algorithm can be applied to several equations without significant modifications, provided that the uniqueness of the traveling wave solution is guaranteed. As a future research direction, it would be valuable to develop a neural network structure that can approximate the traveling wave solution with the minimum wave speed when multiple solutions exist.

Recently, the number of studies attempting to transform the residual loss of PINN has been increasing. The optimization problem for PDE was interpreted as a boundary constraint problem, with empirical evidence that minimizing the loss function for boundary conditions is quite challenging. Reflecting this trend, we will conduct a convergence analysis of algorithms that consider regularization rather than constraints. There exist a few traveling wave solutions which possess monotonicity as a property, rather than as an imposed constraint. In future work, we will investigate the variations that result from training with Physics-informed Neural Networks (PINNs) for this regularized problem.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-00165268) and by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government(MSIP) (No.2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. J. Bramburger, Exact minimum speed of traveling waves in a Keller–Segel model, *Appl. Math. Lett.*, **111** (2021), 106594. https://doi.org/10.1016/j.aml.2020.106594
2. X. Chen, Y. Qi, Traveling wave to non-kpp isothermal diffusion systems: Existence of minimum speed and sharp bounds, *SIAM J. Math. Anal.*, **51** (2019), 1436–1453. https://doi.org/10.1137/18M1176038
3. J. Fang, J. Wu, Monotone traveling waves for delayed Lotka-Volterra competition systems, *Discrete Cont. Dyn. Sys.*, **32** (2012), 3043–3058. http://doi.org/10.3934/dcds.2012.32.3043
4. T. Li, Z. A. Wang, Steadily propagating waves of a chemotaxis model, *Math. Biosci.*, **240** (2012), 161–168. https://doi.org/10.1016/j.mbs.2012.07.003

5. C. Lattanzio, C. Mascia, R. G. Plaza, C. Simeoni, Analytical and numerical investigation of traveling waves for the Allen–Cahn model with relaxation, *Math. Mod. Meth. Appl. Sci.*, **26** (2016), 931–985. https://doi.org/10.1142/S0218202516500226
6. Y. Kan-On, Parameter dependence of propagation speed of travelling waves for competition-diffusion equations, *SIAM J. Math. Anal.*, **26** (1995), 340–363. https://doi.org/10.1137/S0036141093244556
7. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, **2** (1989), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8
8. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
9. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey, *J. Mach. Learn. Res.*, **18** (2018), 1–43.
10. J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029
11. H. J. Hwang, J. W. Jang, H. Jo, J. Y. Lee, Trend to equilibrium for the kinetic Fokker-Planck equation via the neural network approach, *J. Comput. Phys.*, **419** (2020), 109665. https://doi.org/10.1016/j.jcp.2020.109665
12. H. Jo, H. Son, H. J. Hwang, E. Kim, Deep neural network approach to forward-inverse problems, preprint, arXiv:1907.12925.
13. S. W. Chou, H. J. Hwang, H. Son, Traveling Wave Solutions of Partial Differential Equations Via Neural Networks, *J. Sci. Comput.*, **89** (2021), 1–26. https://doi.org/10.1007/s10915-021-01621-w
14. S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, *IMA J. Numer. Anal.*, **42** (2022), 981–1022. https://doi.org/10.1093/imanum/drab032
15. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980.
16. J. Sill, Monotonic networks, *Adv. Neural Inf. Process. Syst.*, **10** (1997).
17. A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in *Proc. icml*, Citeseer, (2013), 3.
18. D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), preprint, arXiv:1511.07289.
19. X. C. liu, X. Han, N. Zhang, Q. Liu, Certified monotonic neural networks, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 15427–15438.
20. A. Gupta, N. Shukla, L. Marla, A. Kolbeinsson, K. Yellepeddi, How to incorporate monotonicity in deep networks while preserving flexibility?, preprint, arXiv:1909.10662.
21. A. Sivaraman, G. arnadi, T. Millstein, G. van den Broeck, Counterexample-guided learning of monotonic neural networks, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 11936–11948.
22. A. Wehenkel, G. Louppe, Unconstrained monotonic neural networks, *Adv. Neural Inf. Process. Syst.*, **32** (2019).

23. T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Trans. Neur. Net.*, **6** (1995), 911–917. https://doi.org/10.1038/10.1109/72.392253

24. J. S. Guo, K. Nakamura, T. Ogiwara, C. H. Wu, The sign of traveling wave speed in bistable dynamics, *Discrete Cont. Dyn. Sys.*, **40** (2020), 3451. https://doi.org/10.3934/dcds.2020047

25. M. W. Hirsch, S. Smale, R. L. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, Academic press, Waltham, 2012.