MDPI

Article

Deep learning-based models numerical solutions and their theoretical stability for a parabolic-parabolic chemotaxis models with nonlocal logistic sources in bounded heterogeneous environments

Tahir Bachar Issa [1,†,‡] and Yawo Ezunkpe [2,‡]

[1] San Jose State University; tahirbachar.issa@sjsu.edu
[2] San Jose State University; yawo.ezunkpe@sjsu.edu
* Correspondence: tahirbachar.issa@sjsu.edu;
† Current address: One Washington Sq, San Jose, CA 95192
‡ These authors contributed equally to this work.

Abstract: This paper proposes a Deep Neural Network(DNN) based model to numerically solve a parabolic-parabolic chemotaxis model with Lokta-Volterra type logistic sources in heterogeneous environments and study the convergence of numerical solutions to corresponding theoretical solutions and find a priori estimates of predictor error. The advantages of deep learning-based methods on numerical methods include solutions obtained that are not restricted to the grid points and we can predict the future dynamical behavior of the system.

Keywords: Deep Learning; Chemotaxis; Numerical Solution; Convergence

1. Introduction

Neural nets, the building blocks of deep learning-based models are good approximation functions according to the Universal Approximation Theorem by X. Li in [18]. Deep learning has been used recently to approximate solutions of PDEs see [7,9–11] and references therein. We refer to [6] for recent work on the deep learning-based numerical study of chemotaxis models.

This paper introduces a Deep Neural Network (DNN) based model that can be used to solve numerically the 1-D chemotaxis model with Lokta-Volterra type logistic sources(see [1,2,12–17,19,20] and reference therein for works chemotaxis models) and study the convergence of numerical solutions to corresponding theoretical solutions and find apriori estimates of predictor error.

$$\begin{cases} u_t = u_{xx} - \chi(uv_x)_x + u\Big(a_0(t,x) - a_1(t,x)u - a_2(t,x)\int_\Omega u\Big), \quad x \in (-1,1) \\ \tau v_t = v_x - \lambda v + \mu u, \quad x \in (-1,1) \\ u_x(\cdot,0) = v_x(\cdot,0) = 0 \end{cases} \tag{1}$$

2. Materials and Methods

Our method consists of a DNN architecture, which is the composition of multiple layers of Neural Networks(NN). A single layer of NN is a composition of an affine function with a non-linear activation function. Each layer of DNN takes inputs from previous layers and progressively refines them. The layers are trained by algorithms that minimize errors and improve accuracy. Deep learning has been used recently in a variety of fields [3–5,7–11] including to approximate solutions of PDEs or to learn nonlinear operators [7,9–11].

Citation: Issa, T.B; Ezunkpe, Y., Title. *Journal Not Specified* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

Copyright: © 2023 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Baseline DNN model and architecture:

In this section, the DNN approximation function will be denoted by $f^{NN}(t,x;m,w,b)$ and we suppose that our DNN has L layers. That means our baseline DNN model has an input layer, $L-1$ hidden layers, and an output layer. The input layer takes (t,x) as input and the final layer gives $f^{NN}(t,x;m,w,b)$ as the output. We denote the relation between the $l-th$ layer and $(l+1)-th$ layer $(l=1,2,\cdots,L-1)$ as

$$z_j^{(l+1)} = \sum_{i=1}^{m_l} w_{ij}^{(l+1)} \sigma_l(z_i^l) + b_j^{l+1}$$

were $m=(m_0,m_1,m_2,\cdots,m_{L-1}), w = w_{ij}^{k\,m_{k-1},m_k,L}{}_{i,j,k=1}, b = b_j^{k\,m_k,L}{}_{j=1,k=1}$ and

- z_i^l: the $i-th$ neuron in the $l-th$ layer
- σ_l: the activation in the $l-th$ layer
- $w_{ji}^{(l+1)}$: the weight between the $i-th$ neuron in the $l-th$ layer and the $j-th$ neuron in the $(l+1)-th$ layer
- $b_j^{(l+1)}$: the bias of the $j-th$ neuron in the $(l+1)-th$ layer
- m_l: the number of neurons in the $l-th$ layer

and the relation between the input layer and the first-hidden layer is expressed as follows:

$$z_j^1 = \sum_{i=1}^{2} w_{ji} z_i^0 + b_j^1$$

where $(z_1^0, z_2^0) = (t,x)$.

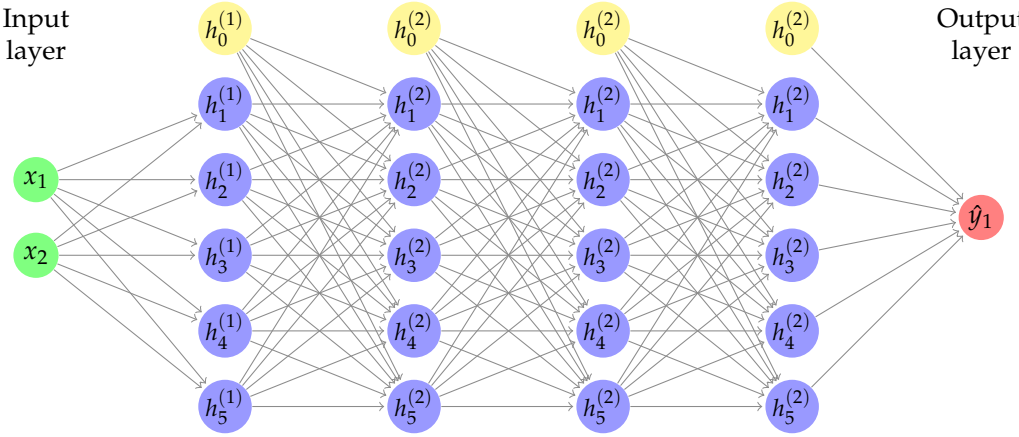

Figure 1. An Example of DNN with L=4.

Pytorch, Adam optimization algorithm, and relu activation functions can be used to build and train our DNN model.

Input data:

To approximate the solution $u(t,\cdot;0,u_0)$ of our system by the DNN numerical solution, we need the data of grid points for each variable domain. We use random sampling to pick grid points within the domain. More precisely,we choose the grid points for training uniformly as follows:

$$\{(t_i,x_j)_{i,j}) \in [0,T]\times[-1,1] \text{ with } \Delta t = 0.01, \Delta x = 0.02\}$$

For the initial condition, we use the grids

$$\{(t=0,x_j)_{i,j})\}$$

and for the boundary conditions, we use the grids

$$\{(t_i, x = 1 \text{ or } -1)_{i,j})\}$$

Loss Function:

We use the Adam optimizer to find the optimal parameteres $w_{ji}^{(l+1)}$ and $b_j^{(l+1)}$ to minimize the loss functions using the back-propagation.

The governing equation of the loss function is defined as follows:

$$Loss_{GE}^1 = \int_0^T \int_{-1}^1 |A_1(t,x) - B_1(t,x) - C_1(t,x)|^2 dxdt$$

$A_1(t,x) = f_1^{NN}(t,x;m,w,b)_t - f_1^{NN}(t,x;m,w,b)_{xx}$,
$B_1(t,x) = \chi(f_1^{NN}(t,x;m,w,b) f_2^{NN}(t,x;m,w,b)_x)_x$,
and

$$C_1(t,x)$$
$$= f_1^{NN}(t,x;m,w,b)\Big(a_0(t,x) - a_1(t,x) f_1^{NN}(t,x;m,w,b) - a_2(t,x)\int_{-1}^1 f_1^{NN}(t,x;m,w,b)dx\Big)$$

$$Loss_{GE}^2 = \int_0^T \int_{-1}^1 |A_2(t,x) - B_2(t,x)|^2 dxdt$$

$A_2(t,x) = f_2^{NN}(t,x;m,w,b)_t - f_2^{NN}(t,x;m,w,b)_{xx}$ and $B_2(t,x) = \lambda f_2^{NN}(t,x;m,w,b) + \nu f_1^{NN}(t,x;m,w,b)$

We define the general equations $Loss_{GE}$ as

$$Loss_{GE} = Loss_{GE}^1 + Loss_{GE}^2$$

We define the loss of the initial conditions as:

$$\begin{aligned} Loss_{IC} &= \int_{-1}^1 \left(|f_1^{NN}(0,x;m,w,b) - u_0(x)|^2 + |f_2^{NN}(0,x;m,w,b) - v_0(x)|^2\right)dx \\ &\approx \frac{1}{N}\sum_{j=1}^N \left(|f_1^{NN}(0,x_j;m,w,b) - u_0(x_j)|^2 + |f_2^{NN}(0,x_j;m,w,b) - v_0(x_j)|^2\right) \end{aligned}$$

We define the boundary condition loss as :

$$\begin{aligned} Loss_{BC} &= \int_0^T \left(|f_1^{NN}(t,-1;m,w,b)_t|^2 + |f_1^{NN}(t,1;m,w,b)_t|^2\right)dt \\ &+ \int_0^T \left(|f_2^{NN}(0,-1;m,w,b)_t|^2 + |f_2^{NN}(0,1;m,w,b)_t|^2\right)dt \\ &\approx \frac{1}{N}\sum_{i=1}^N \left(|f_1^{NN}(t_i,-1;m,w,b)_t|^2 + |f_1^{NN}(t_i,1;m,w,b)_t|^2\right) \\ &+ \frac{1}{N}\sum_{i=1}^N \left(|f_2^{NN}(t_i,-1;m,w,b)_t|^2 + |f_2^{NN}(t_i,1;m,w,b)_t|^2|\right) \end{aligned}$$

Finally, we define the total loss as:

$$Loss_{Total} = Loss_{GE} + Loss_{IC} + Loss_{BC}.$$

3. Results

First, we address the question of the convergence of the proposed numerical scheme in the sense that there exists a sequence of solutions that makes the total loss term converge to zero if a classical solution of (1) exists. Note that $\inf_{t,x} a(t,x) > 0$ implies the existence of global solutions for system (1).

Theorem 1. *Assume* $\inf_{t,x} a_1(t,x) > 0$, *given* $(u_0, v_0) \in C(\bar{\Omega}) \times W^{1,\infty}(\Omega)$ *with* $u_0, v_0 \geq 0$ *and let* $(u(\cdot,0;u_0,v_0), v(\cdot,0;u_0,v_0))$ *be the classical solution of system* (1) *with initial* (u_0, v_0).*Then, there exists* $\{m[j], w[j], b[j]\}$ *such that a sequence of DNN solutions with* $m[j]$ *nodes, denoted by* $\{f_j(t,x) = (f_1^{NN}(t,x;m[j],w[j],b[j]), f_2^{NN}(t,x;m[j],w[j],b[j]))\}$ *satisfies*

$$Loss_{Total}(f_j) \to 0 \text{ as } j \to \infty. \tag{2}$$

Comment: Theorem 2 states that there exist weights of the neural network that reduces the error function as much as we want. However, this does not guarantee that the neural network could converge to the solution of the original equation when the loss function converges to zero. We next would like to address the stability of the scheme in the sense that the neural network architecture converges to an analytic solution of (1) in a suitable function space (to be determined) when the weights of the neural networks minimize the loss $Loss_{Total}$.

Theorem 2. *Let* $\epsilon > 0$, $\{m[j], w[j], b[j]\}_{j=1}^{\infty}$ *be a sequence miminizing* $Loss_{Total}(f_j)$. *Under suitable non-explicit conditions on the model coefficients, the proposed scheme is stable in the sense the* $Loss_{Total}(f_j) \to 0$ *as* $j \to \infty$ *implies*

$$\|f_j(\cdot,\cdot;m[j],w[j],b[j]) - (u(\cdot,0;u_0,v_0), v(\cdot,0;u_0,v_0))\|_{L^\infty([0,T]:L^2([-1,1]))} \leq C(u_0,v_0,a_i,\sigma,\lambda,\nu)\epsilon^2. \tag{3}$$

4. Discussion

In this work, we propose a deep-learning-based model to approximate the numerical solution of 1-D chemotaxis models and study the convergence and stability of the numerical scheme. Our next step is to solve numerically our model using pytorch and compare the results with recent numerical methods. The advantages of deep learning-based methods on numerical methods include solutions obtained that are not restricted to the grid points and we can predict the future dynamical behavior of the system. A natural extension is to use Convolution Neural networks or Sequential Deep Learning models.

Author Contributions: "Conceptualization, Issa.T.B and Ezunkpe.Y.; methodology, Issa.T.B and Ezunkpe.Y; formal analysis, Issa.T.B and Ezunkpe.Y; writing—original draft preparation, Issa.T.B and Ezunkpe.Y; writing—review and editing, Issa.T.B and Ezunkpe.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Negreanu M, Vargas AM. Dynamics in a Chemotaxis Model with Periodic Source. *Mathematics*. 2022; 10(3):312. https://doi.org/10.3390/math10030312
2. T.B. Issa, R.B. Salako, W. Shen,Traveling wave solutions for two species competitive chemotaxis systems, *Nonlinear Analysis*, **212**,2021,112480,ISSN 0362-546X,https://doi.org/10.1016/j.na.2021.112480.
3. T. B. Issa, David Umnisky et al, Toward Automatic Mammography Auditing via Universal Language Model Fine Tuning, *IEEE 22nd International Conference on Information Reuse and Integration for Data Science* (2021)
4. C. Cronister, T.B. Issa , D. Uminsky , and R.W. Filice, Toward an Automated Radiology Exam Protocol Selection System, 2021 (submitted)
5. Riley P. Buley, Hannah E. Correia, Ash Abebe, Tahir B. Issa, Alan E. Wilson, Predicting microcystin occurrence in freshwater lakes and reservoirs: assessing environmental variables,*Inland Waters*, (2021), 1:3, 430-444, DOI: 10.1080/20442041.2021.1938491

6. Sunwoo Hwang, Seongwon Lee and Hyung Ju Hwang, Neural network approach to data-driven estimation of chemotactic sensitivity in the Keller-Segel model, *MBE* , 18(6) (2021): 8524–8534. DOI: 10.3934/mbe.2021421
7. Lu, L., Jin, P., Pang, G. et al., Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators *Nat Mach Intell* 3, 218–229 (2021). https://doi.org/10.1038/s42256-021-00302-5
8. T. B. Issa, C. Vinegoni, A. Shaw, P. F. Feruglio, R. Weissleder and D. Uminsky, Video-rate acquisition fluorescence microscopy via generative adversarial networks, *IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), Cincinnati, OH, USA* (2020), pp. 569-576, doi: 10.1109/BIBE50027.2020.00098.
9. Raissi, Maziar and Perdikaris, Paris and Karniadakis, George E, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, (2019), vol. 378, 686–707, Elsevier
10. Raissi, Maziar and Perdikaris, Paris and Karniadakis, George E, Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, (2017), arXiv preprint arXiv:1711.10561
11. Raissi, Maziar and Perdikaris, Paris and Karniadakis, George E, Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations, (2017),arXiv preprint arXiv:1711.10566
12. T. B. Issa and R. Salako, Asymptotic dynamics in a two-species chemotaxis model with non-local terms, *Discrete Contin. Dyn. Syst. Ser. B*, **22** (2017), no. 10, 3839-3874.
13. T. B. Issa and W. Shen, Dynamics in chemotaxis models of parabolic-elliptic type on bounded domain with time and space dependent logistic sources, *SIAM J. Appl. Dyn. Syst.*, **16** (2017), no. 2, 926-973.
14. T. B. Issa and W. Shen, Persistence, coexistence and extinction in two species chemotaxis models on bounded heterogeneous environments, *J. Dyn. Diff. Equat.*, (2018). https://doi.org/10.1007/s10884-018-9686-7
15. T. B. Issa and W. Shen, Uniqueness and stability of coexistence states in two species models with/without chemotaxis on bounded heterogeneous environments,*J. Dyn. Diff. Equat.*, **31** 2305–2338 (2019) https://doi.org/10.1007/s10884-018-9706-7.
16. T. B. Issa and W. Shen, Pointwise persistence in full chemotaxis models with logistic source on bounded heterogeneous environments *Journal of Mathematical Analysis and Applications* **490**(1) 124204 (2020) https://doi.org/10.1016/j.jmaa.2020.124204
17. M. Winkler, How far can chemotactic cross-diffusion enforce exceeding carrying capacities? *J. Nonlinear Sci.*, **24** (2014), 809-855.
18. X. Li, Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer, *Neurocomputing*, **12** (1996), 327–343.
19. E.F. Keller and L.A. Segel, Initiation of slime mold aggregation viewed as an instability, *J. Theoret. Biol.*, **26** (1970), 399-415.
20. E.F. Keller and L.A. Segel, A model for chemotaxis, *J.Theoret. Biol.*, **30** (1971), 225-234.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.