Mathematical Biosciences and Engineering

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

MBE, 18(6): 8524–8534.
DOI: 10.3934/mbe.2021421
Received: 05 July 2021
Accepted: 22 September 2021
Published: 29 September 2021

Research article

Neural network approach to data-driven estimation of chemotactic sensitivity in the Keller-Segel model

Sunwoo Hwang[1], Seongwon Lee[2] and Hyung Ju Hwang[1,*]

[1] Department of Mathematics Pohang University of Science and Technology Pohang, Republic of Korea

[2] Innovation Center for Industrial Mathematics, National Institute for Mathematical Sciences, Daejeon, Republic of Korea

* **Correspondence:** Email: hjhwang@postech.ac.kr.

Abstract: We consider the mathematical model of chemotaxis introduced by Patlak, Keller, and Segel. Aggregation and progression waves are present everywhere in the population dynamics of chemotactic cells. Aggregation originates from the chemotaxis of mobile cells, where cells are attracted to migrate to higher concentrations of the chemical signal region produced by themselves. The neural net can be used to find the approximate solution of the PDE. We proved that the error, the difference between the actual value and the predicted value, is bound to a constant multiple of the loss we are learning. Also, the Neural Net approximation can be easily applied to the inverse problem. It was confirmed that even when the coefficient of the PDE equation was unknown, prediction with high accuracy was achieved.

Keywords: differential equation; approximated solution; artificial neural networks; Patlak-Keller-Segel equation

1. Introduction

Chemotaxis is the process by which cells or organisms change the state of migration by reacting to the presence of chemicals to access chemically favorable and adverse environments. It plays an important role in many cases, for example, that bacteria move toward the hight concentration of foods and eukaryotic cells migrate towards or away from the chemical source [1, 2]. Several aspects of chemotactic motility have been studied in great detail, especially for the model organism E. coli . Keller and Segel propsed a mathematical model with chemotaxis in 1970 [3]. The model described the behavior of chemotaxis well and has been widely used for the modeling chemotaxis [4, 5].

This article considers the mathematical model of coinage introduced by Patlak, Keller, and Segel [3, 6–8]. Aggregation and propagation exist everywhere in the population dynamics of chemo-

tactic cells. Chemical chemotactic bacteria are well known for expanding their habitats by patterning them into localized populations. Aggregation occurs in the chemotaxis of a migrating cell that is attracted to move the cell to a higher concentration of the region of the chemical signal it produces. The reaction-diffusion-progression equation for describing the population dynamics of chemotactic cells was proposed by Keller and Segel, where the term of reaction refers to local population dynamics such as cell proliferation and the term of diffusion refers to random movement and convection of cells. A variant of the classic Keller-Segel model was developed to obtain a model that could explain more realistic behavior of chemotactic cells.

Focus on the following equation, the Patlak-Keller-Segel equation.

$$\begin{aligned}
\partial_t u &= D_u \partial_{xx} u - \partial_x(\chi u \partial_x v) \\
\partial_t v &= D_v \partial_{xx} v + \alpha u - \beta v \\
u(0,x) &= u_0(x), \quad v(0,x) = v(x) \\
u(t,x) &= g_u(t,x), \quad v(t,x) = g_v(t,x), \quad \text{for } x \in \{-1,1\}
\end{aligned}$$

We study on deep learning architectures to solve Patlak-Keller-Segel equation. Using the initial, boundary condition, and pde equations as loss functions, the Neural Net will be trained as an approximation solution. This approach has two major advantages. First, a function learned in our method can make predictions on any gird. A function trained with a neural net gives an output for any input value. This is a difference from the existing numerical approach that predicted only the value of a specific gird. Second, the unknown coefficient can be predicted by learning. It is sometimes necessary to make predictions when the coefficient is unknown. In this Forward-Inverse Problem, the Neural Net approach is easily applied [9]. Each coefficients parameterizes as weights that can be learned.

There have been attempts to solve Differential Equation using Neural Nets [10,11]. Neural Nets can fit any function [12]. In the case of PDEs with complex behaviors, it is difficult to find a solution. While studies to find an approximation solution using a Neural Net in such a PDE has been started recently, there are not many works with the Neural Net approaches in the field of mathematical biology. In this paper, we applied the Neural Net approach to the Patlak-Keller-Segel equation. It is shown in the paper that the Neural Net method can have a good accuracy compared to the existing method, Finite Volume Methods (FVM).

2. Materials and methods

2.1. Approximation solution

Keller-Segel equations deal with several numerical approaches. Numerical methods have been implemented using methods such as finite volume methods and finite element methods. We prove existence and uniqueness of a numerical solution to the proposed scheme. Then, we give a priori estimates and establish a threshold on the initial mass, for which we show that the numerical approximation converges to the solution to the PKS system when the initial mass is lower than this threshold. Numerical simulations are performed to verify accuracy and the properties of the scheme. Finally, in the last section we investigate blow-up of the solution for large mass.

In [11], a mass transport steepest descent scheme was proposed to solve the modified 1D Keller-Segel system for the log interaction kernel proposed in [9]. This method is designed based on the

transformation method of the Fokker-Planck type equation introduced in [13, 14] and applied to the Keller-Segel type model in to satisfy the discrete free energy dissipation principle. By taking into account the problem of transformed variables, this method can accurately solve areas of high concentration without fine-tuning the mesh. This approach has been extended to several dimensions for nonlinear aggregate-diffusion equations and from discretization in [15, 16] references therein to other approaches.

In these methods, the χ is entered as a condition. Looking at the CFL condition, it is necessary to hold the grid more closely according to the χ. This is why it is difficult to analyze the Inverse Problem numerically. It is impossible to satisfy the CFL condition with an unknown χ value. A new method is needed to deal with the inverse problem.

The neural net can be used to find the approximate solution of the PDE. We define the loss function according to equation, boundary and initial condition. Neural net structure is built and learning is performed using the loss function. It has been proved that this approximate solution can always be obtained using the universal approximation theorem. The Neural Net approach is useful for solving inverse problems. We define unknown coefficients as variables.

The neural net approach is useful for solving inverse problems. We define unknown coefficients as variables and proceed with learning. When learning through the optimizer, a coefficient close to the actual value can be obtained. This method does not depend on the value of χ.

Neural network approaches rarely discuss errors between actual and predicted values. It cannot be easily discussed because the actual value is unknown. For this reason, existing studies often measure only the loss used for learning. We compensate for this weakness by bounding the error using loss. The next subsection is about this topic.

2.2. Deep Learning algorithm

In this section, we introduce our DNN structure and Numerical Method. The proof process in this section was applied by modifying the proof applied to the Fokker-Planck equation. [17]

Let $f^{NN}(t, x; m, w, b)$ be a deep learning function with smooth activation function. f will be approximation solution of Keller-Segel equation. The structure of f is as follows.

$$z_j^{(l+1)} = \sum_{i=1}^{m_l} w_{ji}^{(l+1)} \bar{\sigma}_l(z_i^{(l)}) + b_j^{(l+1)}$$

2.3. Input points

We need the data of grid points for each variable domain. We use random sampling to pick grid points within the domain. The gird points of initial condition and the boundary condition are the same as $\{(t = 0, x_j)\}_j$ and $\{(t_i, x = 1 \text{ or } -1)\}_i$, respectively.

2.4. Loss functions

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

The governing equation loss is defined as follows:

$$\begin{aligned}\mathrm{Loss}_{\mathrm{GE}}^{1} = & \int_{(0,T)} dt \int_{(-1,1)} dx \Big| \partial_t f_1^{\mathrm{NN}}(t,x;m,w,b) \\ & - D_u \partial_{xx} f_1^{\mathrm{NN}}(t,x;m,w,b) + \chi \partial_x f_1^{\mathrm{NN}} \partial_x f_2^{\mathrm{NN}} + \chi f_1^{\mathrm{NN}} \partial_{xx} f_2^{\mathrm{NN}}, \\ \mathrm{Loss}_{\mathrm{GE}}^{2} = & \int_{(0,T)} dt \int_{(-1,1)} dx \Big| \partial_t f_2^{\mathrm{NN}}(t,x;m,w,b) \\ & - D_v \partial_{xx} f_1^{\mathrm{NN}}(t,x;m,w,b) - \alpha f_1^{\mathrm{NN}}(t,x;m,w,b) + \beta f_2^{\mathrm{NN}}(t,x;m,w,b).\end{aligned}$$

Then we define $\mathrm{Loss}_{\mathrm{GE}}$ as

$$\mathrm{Loss}_{\mathrm{GE}} = \mathrm{Loss}_{\mathrm{GE}}^{1} + \mathrm{Loss}_{\mathrm{GE}}^{2}.$$

The initial grid points as follows:

$$\mathrm{Loss}_{\mathrm{IC}} = \int_{(-1,1)} dx |f^{\mathrm{NN}}(0,x) - f_0(x)|^2 \approx \frac{1}{N_j} \sum_j |f^{\mathrm{NN}}(0,x_j) - f_0(x_j)|^2.$$

The boundary condition is defined as follows:

$$\begin{aligned}\mathrm{Loss}_{\mathrm{BC}} = & \int_{(0,T)} |f^{\mathrm{NN}}(t,-1;m,w,b) - g(t,-1;m,w,b)|^2 dt \\ & + \int_{(0,T)} |f^{\mathrm{NN}}(t,1;m,w,b) - g(t,1;m,w,b)|^2 dt \\ & + \int_{(0,T)} |f_t^{\mathrm{NN}}(t,-1;m,w,b) - g_t(t,-1;m,w,b)|^2 dt \\ & + \int_{(0,T)} |f_t^{\mathrm{NN}}(t,1;m,w,b) - g_t(t,1;m,w,b)|^2 dt \\ \approx & \frac{1}{2N} \sum_{x\in\{-1,1\},i} |f^{\mathrm{NN}}(t_i,x;m,w,b) - g(t_i,x;m,w,b)|^2 \\ & + \frac{1}{2N} \sum_{x\in\{-1,1\},i} |f_t^{\mathrm{NN}}(t_i,x;m,w,b) - g_t(t_i,x;m,w,b)|^2.\end{aligned}$$

Finally, we define the total loss as

$$\mathrm{Loss}_{\mathrm{total}} = \mathrm{Loss}_{\mathrm{GE}} + \mathrm{Loss}_{\mathrm{IC}} + \mathrm{Loss}_{\mathrm{BC}}. \tag{1}$$

2.5. Error bound with Loss

Theorem 1. *Let*

$$\begin{aligned} l_1 &= \int_{D_{\mathrm{int}}} \|L(u,v)\| \\ l_2 &= \int_{D_{\mathrm{ini}}} \|(u,v)(x,0) - (u_0,v_0)\| \\ l_3 &= \int_{D_{\mathrm{bd}}} \left| \frac{\partial}{\partial n}(u,v) \right|. \end{aligned}$$

Then,

$$\int_D |(u, v) - (u^*, v^*)| \leq C(l_1 + l_2 + l_3)$$

for some constant C, where (u^*, v^*) *is the solution of equation.*

Proof. Consider

$$\partial_t u - D_u \partial_{xx} u + \chi \partial_x (u \partial_x v).$$

Integrate on I, then

$$\partial_t \int_I u - D_u \int_I \partial_{xx} u + \chi \int_I u \partial_x v = \partial_t \int_I u - D_u [\partial_x u]_{\partial I} + \chi [u \partial_x v]_{\partial I}.$$

Integrate on $[0, T]$, then

$$\begin{aligned} l_{1,u} &= \int_I u(,t)dx - D_u \int_I [\partial_x u]_{\partial I} u + \chi \int_I [\partial_x u]_{\partial I} \\ &= \int_I u(,t)dx - D_u \int_I [\partial_x u]_{\partial I} u - \chi \int_I [\partial_x u]_{\partial I} \end{aligned}$$

Subtract same equation of u^*.

Also consider

$$\partial_t v - D_v \partial_{xx} v + \beta v = \alpha u$$

□

Remark 2. *The assumption* $f \in \widehat{C}^{(1,1)}([0, T] \times [0, 1])$ *can be replaced by a general Sobolev space, since the functions in a Sobolev space can be approximated by the continuous functions on a compact set.*

Let u and v be solutions and u^{NN} and v^{NN} be approximation solutions. ans $u^d = u - u^{\mathrm{NN}}$, $v^d = v - v^{\mathrm{NN}}$.

$$\begin{aligned} d_{\mathrm{ge}}^{(1)} &= \partial_t u^d - \partial_{xx} v^d + \partial_x (u \partial_x v)^d \\ d_{\mathrm{ge}}^{(2)} &= \partial_t v^d - \partial_{xx} v^d - u^d + v^d \\ d_{\mathrm{bd}}^{(1)} &= \partial_x u^{\mathrm{NN}} \\ d_{\mathrm{bd}}^{(2)} &= \partial_x v^{\mathrm{NN}} \\ d_{\mathrm{ini}}^{(1)} &= u^d \\ d_{\mathrm{ini}}^{(2)} &= v^d \end{aligned}$$

Consider

$$2u^d \partial_t u^d = 2u^d \partial_{xx} u^d + 2u^d \partial_x (u \partial_x v)^d + 2u^d d_{\mathrm{ge}}^{(1)}.$$

Integrate on x, then the first term of RHS

$$\begin{aligned} 2 \int_I u^d \partial_{xx} u^d dx &= 2 \int_{\partial I} u^d \partial_x u^d n_x dS_x - 2 \int_I (\partial_x u^d)^2 dx \\ &= \int_{\partial I} (u^d)^2 n_x dS_x + \int_{\partial I} (\partial_x u^d)^2 n_x dS_x - 2 \int_I (\partial_x u^d)^2 dx \end{aligned}$$

$$= \int_I (u^d)^2 n_x dS_x + \int_{\partial I} (\partial_x u^d)^2 n_x dS_x - \int_I (\partial_x u^d)^2 dx.$$

The second term of RHS

$$2\int_I u^d \partial_x (u\partial_x v)^d n_x dS_x = 2\int_{\partial I} u^d (u\partial_x v)^d dx + 2\int_I \partial_x u^d (u\partial_x v)^d dx$$

Define $B_1 = 2\int_{\partial I} u^d (u\partial_x v)^d n_x dS_x$, and $B_2 = 2\int_I \partial_x u^d (u\partial_x v)^d dx$.

$$\begin{aligned}
|B_1| &= 2\left|\int_{\partial I} (u - u^{\mathrm{NN}})(u\partial_x v - u^{\mathrm{NN}}\partial_x v^{\mathrm{NN}}) n_x dS_x\right| \\
&\le 2\left|\int_{\partial I} (u^d)^2 \partial_x v^{\mathrm{NN}} n_x dS_x\right| + 2\left|\int_{\partial I} u u^d \partial_x v^{\mathrm{NN}} n_x dS_x\right| \\
&\le 2\|d_{\mathrm{bd}}^{(2)}\|_{L^\infty(\partial I)} \|u^d\|_{L^2(\partial I)} + M_1(\epsilon_1 \|u^d\|_{L^2(\partial I)} + \frac{1}{\epsilon_1}\|v_x^{\mathrm{NN}}\|_{L^2(\partial I)})
\end{aligned}$$

where $\|u^d\|_{L^\infty(I)} < M_1$.

We may assume $\|d_{\mathrm{bd}}^{(2)}\|_{L^\infty(\partial I)} < 1/6$.

$$\begin{aligned}
|B_1(t)| &\le \left(\frac{1}{3} + M_1\epsilon_1\right)(\|u^d\|_{L^2(\partial I)}) + \frac{M_1}{\epsilon_1}\|d_{\mathrm{bd}}^{(2)}\|_{L^2(\partial I)} \\
&\le \left(\frac{1}{3} + M_1\epsilon_1\right)(\|u^d\|_{L^2(I)} + \|\partial_x u^d\|_{L^2(I)}) + \frac{M_1}{\epsilon_1}\|d_{\mathrm{bd}}^{(2)}\|_{L^2(\partial I)}
\end{aligned}$$

Choose ϵ_1 such that $1/3 + M_1\epsilon_1 < 1/2$,

$$|B_1(t)| \le \frac{1}{2}\|u^d\|_{L^2(I)} + \frac{1}{2}\|\partial_x u^d\|_{L^2(I)} + C_1\|d_{\mathrm{bd}}^{(2)}\|_{L^2(\partial I)}.$$

Consider $B_2(t)$

$$B_2(t) = 2\int_I \partial_x u^d u^{\mathrm{NN}} \partial_x v^d dx + 2\int_I \partial_x u^d \partial_x v u^d dx.$$

First term of RHS is transformed as follows.

$$\begin{aligned}
2\left|\int_I \partial_x u^d u^{\mathrm{NN}} \partial_x v^d dx\right| &\le \|u^{\mathrm{NN}}\|_{L^\infty(I)}\left(\epsilon_2\|\partial_x u^d\|_{L^2(I)} + \frac{1}{\epsilon_2}\|\partial_x v^d\|_{L^2(\partial I)}\right) \\
&\le M_2\epsilon_2\|\partial_x u^d\|_{L^2(I)} + \frac{M_2}{\epsilon_2}\|\partial_x v^d\|_{L^2(\partial I)}
\end{aligned}$$

where $\|u^{\mathrm{NN}}\|_{L^\infty(I)} \le \|u\|_{L^\infty(I)} + \|u^d\|_{L^\infty(I)} < M_2$. Second term of RHS is transformed as follows.

$$\begin{aligned}
2\left|\int_I \partial_x u^d \partial_x v u^d dx\right| &\le \|\partial_x v\|_{L^\infty(I)}\left(\epsilon_2\|\partial_x u^d\|_{L^2(I)} + \frac{1}{\epsilon_2}\|u^d\|_{L^2(\partial I)}\right) \\
&\le M_3\epsilon_3\|\partial_x u^d\|_{L^2(I)} + \frac{M_3}{\epsilon_3}\|u^d\|_{L^2(\partial I)}
\end{aligned}$$

where $\|\partial_x v\|_{L^\infty(I)} < M_3$. Hence

$$|B_2(t)| \le (M_2\epsilon_2 + M_3\epsilon_3)\|\partial_x u^d\|_{L^2(I)} + \frac{M_2}{\epsilon_2}\|\partial_x v^d\|_{L^2(\partial I)} + \frac{M_3}{\epsilon_3}\|u^d\|_{L^2(\partial I)}$$

Choose ϵ_3 such that $M_2\epsilon_2 + M_3\epsilon_3 < 1/2$,

$$\frac{d}{dt}\|u^d\|_{L^2(I)} \le \|d_{\text{ge}}^{(1)}\|_{L^2(I)} + \|d_{\text{bd}}^{(1)}\|_{L^2(\partial I)} + C_1\|d_{\text{bd}}^{(2)}\|_{L^2(\partial I)} + C_2\|u^d\|_{L^2(\partial I)} + M\|\partial v^d\|_{L^2(\partial I)}. \tag{2}$$

Also consider,

$$2v^d\partial_t v^d = 2v^d\partial_{xx}v^d - 2v^d u^d + 2(v^d)^2 + 2v^d d_{\text{ge}}^{(2)}.$$

Silimarly,

$$\frac{d}{dt}\|v^d\|_{L^2(I)} \le \|d_{\text{ge}}^{(2)}\|_{L^2(I)} + C_3\|d_{\text{bd}}^{(2)}\|_{L^2(\partial I)} + C_4\|u^d\|_{L^2(\partial I)} + C_5\|u^d\|_{L^2(\partial I)} - \|\partial v^d\|_{L^2(\partial I)}. \tag{3}$$

Calculate (2) + $M \times$ (3),

$$\begin{aligned}\frac{d}{dt}\|u^d\|_{L^2(I)} + M\frac{d}{dt}\|v^d\|_{L^2(I)} \le{}& \|d_{\text{ge}}^{(1)}\|_{L^2(I)} + \|d_{\text{bd}}^{(1)}\|_{L^2(\partial I)} + C_1\|d_{\text{bd}}^{(2)}\|_{L^2(\partial I)} + C_2\|u^d\|_{L^2(\partial I)} \\ &+ M\|d_{\text{ge}}^{(2)}\|_{L^2(I)} + MC_3\|d_{\text{bd}}^{(2)}\|_{L^2(\partial I)} + MC_4\|u^d\|_{L^2(\partial I)} + MC_5\|u^d\|_{L^2(\partial I)}\end{aligned}$$

Hence,

$$\begin{aligned}\|u^d\|_{L^2(\Omega\times[0,T])} + M\|v^d\|_{L^2(\Omega\times[0,T])} \le{}& C(\|d_{\text{ge}}^{(1)}\|_{L^2(\Omega\times[0,T])} + \|d_{\text{ge}}^{(2)}\|_{L^2(\Omega\times[0,T])} + \|d_{\text{bd}}^{(1)}\|_{H^1(\partial\Omega\times[0,T])} \\ &+ \|d_{\text{bd}}^{(2)}\|_{H^1(\partial\Omega\times[0,T])} + \|d_{\text{ini}}^{(1)}\|_{L^2(\Omega\times\{0\})} + \|d_{\text{ini}}^{(2)}\|_{L^2(\Omega\times\{0\})}),\end{aligned}$$

for some constant C.

3. Results

3.1. Theoretical results

In most cases, L^2 loss is used in the PDE solution approximation through Neural Net. As mentioned in the previous subsection, this does not guarantee error bound. We found a loss that can bound the error through the energy estimate.

The energy estimate was calculated from the given PDE equation. By modifying this equation, we proved that the error, the difference between the actual value and the predicted value, is bound to a constant multiple of the loss we are learning. From this proof it can be seen that the approximation with a sufficiently small loss is very close to the actual solution.

The found loss uses H^1 loss instead of L^2 loss in some terms. This means that reducing the difference in function values is insufficient to reduce errors. If H^1 loss is used, not only the function value but also the difference between the derivative value is reduced. In order to make the error sufficiently small, the derivative value of some terms must be small.

Our method is directly applicable to the Inverse Problem. When there is an unknown coefficient, most of the methods cannot be applied as it is. In particular, Numerical methodologies are developed depending on coefficients. It is only possible to apply a new method of predicting parameters

through observed data points. On the other hand, our method needs no transformation. If it converges sufficiently, it becomes a good approximator for both the parameter and the solution.

Chemotaxis has been primarily studied as an average characteristic of a population, with little regard for variability among individuals. Despite the simplifying approximations involved in our derivations, especially in the extrapolation to higher spatial dimensions, the models demonstrate a satisfactory and very useful ability to quantitatively interpret population assays for bacterial and leukocyte chemotactic migration.

3.2. Simulation results

The results from the previous method are as follows. It can be seen that the Neural Net method fits well with the Keller-Segel solution like well-known numerical method. A finite volume method (FVM) was used for the comparison. The simulations were performed using Clawpack [18, 19] with a fractional step method [20] in Python 3.6.

Figure 1 and Figure 2 show the result of Neural Net approximation and FVM approximation, repectively. It can be seen that the Neural Net method is not behind the FVM. Our approximate solution has a chemotaxis phenomenon. One can see that the aggregation takes place over time. This new methodology can sufficiently replace the existing methodology.

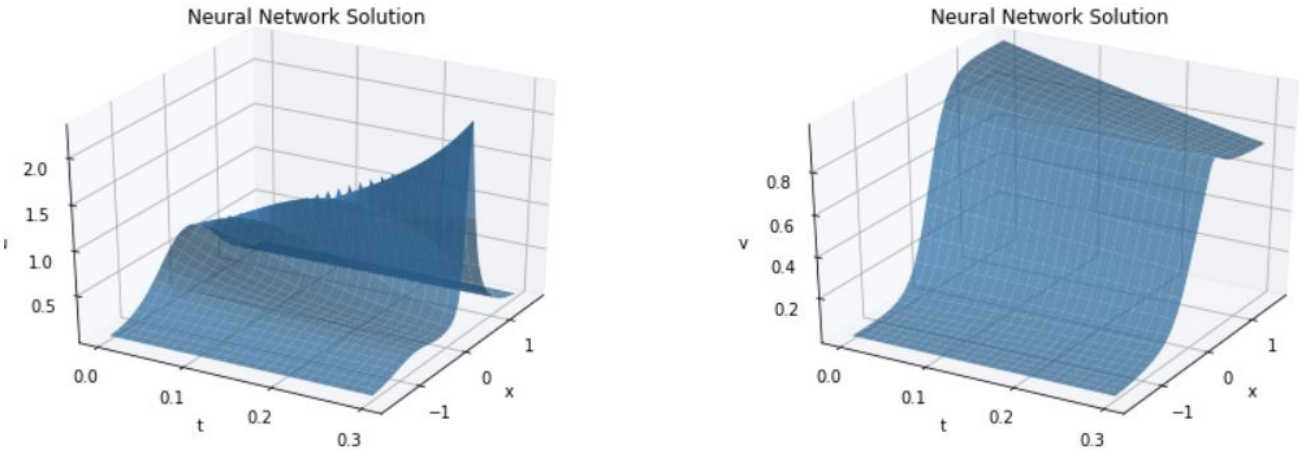

Figure 1. Neural Net Approximation solution of u, v.

We find the coefficients of the solution and the solution itself simultaneously. It couldn't be done with the Numerical method. Find the coefficient and solution simultaneously through the observed data points. It can be seen that there is very little error even when comparing the found coefficient with the actual value.

Our method has the advantage of being able to deal with the inverse problem as well as the forward problem compared to the previous studies. As for the existing numerical method, few studies of the Inverse Problem in Keller-Segel have been conducted. In the case of Neural Net, the forward-inverse problem can be easily solved through simple transformation.

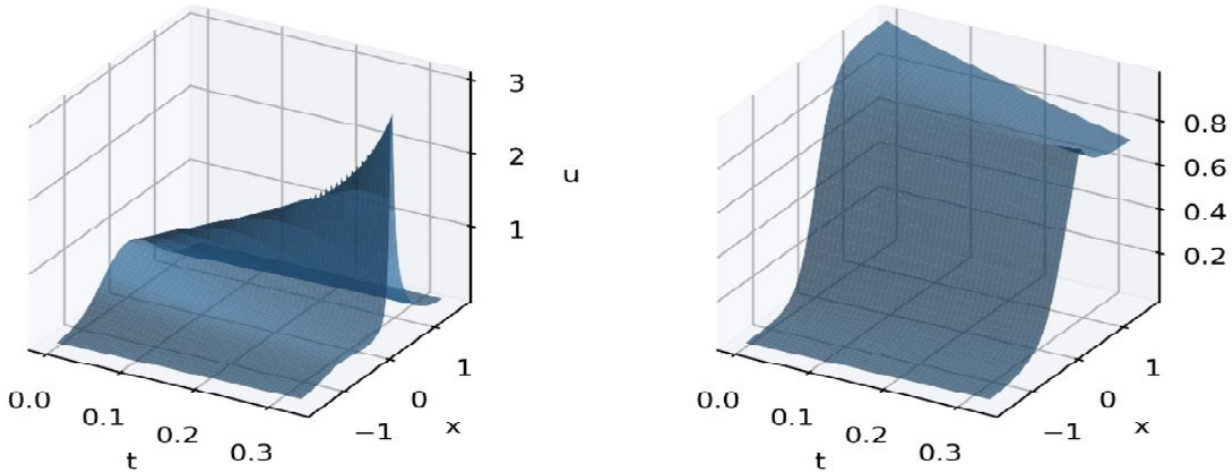

Figure 2. FVM Approximation solution of u, v.

4. Conclusion

Neural network approaches rarely discuss errors between actual and predicted values. It cannot be easily discussed because the actual value is unknown. For this reason, existing studies often measure only the loss used for learning. We compensate for this weakness by bounding the error using loss.

H^1 loss is used instead of L^2 loss in some terms. This means that reducing the difference in function values is insufficient to reduce errors. If H^1 loss is used, not only the function value but also the difference between the derivative value is reduced. In order to make the error sufficiently small, the derivative value of some terms must be small.

Acknowledgments

Seongwon Lee was supported by National Institute for Mathematical Sciences (NIMS) grant funded by the Korea government (MSIT) (No. B21810000). Hyung Ju Hwang was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2017R1E1A1A03070105 and NRF-2019R1A5A1028324), Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)), and the Information Technology Research Center (ITRC) support program (No. IITP-2018-0-01441).

Conflict of interest

The authors have no conflict of interest.

References

1. P. Devreotes, C. Janetopoulos, Eukaryotic chemotaxis: Distinctions between directional sensing and polarization, *J. Biol. Chem.*, **278** (2003), 20445–20448.

2. D. V. Zhelev, A. M. Alteraifi, D. Chodniewicz, Controlled pseudopod extension of human neutrophils stimulated with different chemoattractants, *Biophys. J.*, **87** (2004), 688–695.
3. E. F. Keller, L. A. Segel, Initiation of slime mold aggregation viewed as an instability, *J. Theor. Biol.*, **26** (1970), 399–415.
4. L. A. Segel, A. Goldbeter, P. N. Devreotes, B. E. Knox, A mechanism for exact sensory adaptation based on receptor modification, *J. Theor. Biol.*, **120** (1986), 151–179.
5. J. A. Sherratt, Chemotaxis and chemokinesis in eukaryotic cells: The Keller-Segel equations as an approximation to a detailed model, *Bull. Math. Biol.*, **56** (1994), 129–146.
6. C. S. Patlak, Random walk with persistence and external bias, *Bull. Math. Biophys.*, **15** (1953), 311–338.
7. E. F. Keller, L. A. Segel, Model for chemotaxis, *J. Theor. Biol.*, **30** (1971), 225–234.
8. E. F. Keller, L. A. Segel, Traveling bands of chemotactic bacteria: a theoretical analysis, *J. Theor. Biol.*, **30** (1971), 235–248.
9. H. Jo , H. Son, H. J. Hwang, E. H. Kim, Deep neural network approach to forward-inverse problems, *Networks Heterog. Media*, **15** (2020), 247.
10. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, `arXiv:1711.10561`.
11. 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.
12. X. Li, Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer, *Neurocomputing*, **12** (1996), 327–343.
13. J. Soler, J. A. Carrillo, L. L. Bonilla, Asymptotic behavior of an initial-boundary value problem for the Vlasov–Poisson–Fokker–Planck system, *SIAM J. Appl. Math.*, **57** (1997), 1343–1372.
14. F. Bouchut, J. Dolbeault, On long time asymptotics of the Vlasov-Fokker-Planck equation and of the Vlasov-Poisson-Fokker-Planck system with Coulombic and Newtonian potentials, *Differ. Integral Equat.*, **8** (1995), 487–514.
15. J. Han, A. Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, *Proceed. Nat. Aca. Sci.*, **115** (2018), 8505–8510.
16. J. Bourgain, Fourier transform restriction phenomena for certain lattice subsets and applications to nonlinear evolution equations, *Geometric & Functional Analysis GAFA*, **3** (1993), 209–262.
17. 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.
18. Clawpack Development Team, Clawpack software, *http://www.clawpack.org, Version 5.5.0*, (2018).
19. K. T. Mandli, A. J. Ahmadia, M. Berger, D. Calhoun, D. L. George, Y. Hadjimichael, et al., Clawpack: Building an open source ecosystem for solving hyperbolic PDEs, *PeerJ Comput. Sci.*, **2** (2016), e68.

20. R. Tyson, L. G. Stern, R. J. LeVeque, Fractional step methods applied to a chemotaxis model, *J. Math. Biol.*, **41** (2000), 455–475.

AIMS Press