COMPARISON OF DATA-DRIVEN AND PHYSICS-INFORMED NEURAL NETWORKS FOR SURROGATE MODELLING OF THE HUXLEY MUSCLE MODEL

Bogdan Milićević[1,2], Miloš Ivanović[2,3], Boban Stojanović[2,3], Miljan Milošević[2,4,5], Vladimir Simić[2,4], Miloš Kojić[2,6,7] and Nenad Filipović[1,2]

[1] Faculty of Engineering, University of Kragujevac, Sestre Janjić 6, 34000 Kragujevac
e-mail: bogdan.milicevic@uni.kg.ac.rs, fica@kg.ac.rs
[2] Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovića 6, 34000 Kragujevac
[3] Faculty of Science,University of Kragujevac, Radoja Domanovića 12, 34000 Kragujevac
e-mail: mivanovic@kg.ac.rs, boban.stojanovic@pmf.kg.ac.rs
[4] Institute for Information Technologies, University of Kragujevac, Jovana Cvijića bb, 34000 Kragujevac, Serbia. e-mail: miljan.m@kg.ac.rs, vsimic@kg.ac.rs,
[5] Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade, Serbia.
[6] Serbian Academy of Sciences and Arts, Knez Mihailova 35, 11000 Belgrade, Serbia.
e-mail: mkojic42@gmail.com
[7] Houston Methodist Research Institute, The Department of Nanomedicine, 6670 Bertner Ave., R7 117, Houston, TX 77030.

Abstract:

Biophysical muscle models based on sliding filament and cross-bridge theory are called Huxley-type muscle models. The method of characteristics is typically used to solve Huxley's muscle contraction equation, which describes the distribution of attached myosin heads to the actin-binding sites, called cross-bridges. Once this equation is solved, we can determine the generated force and the stiffness of the muscle fibers, which can then be used at the macro level during finite element analysis. In our paper, we present alternative approaches to finding an approximate solution of Huxley's muscle contraction equation using neural networks. In one approach, we collect the data from simulations and train multilayer perceptron to predict probabilities of cross-bridge formation based on the available actin site positions, time, activation, current and previous stretch. In another approach, besides using the data, we also inform the neural network with Huxley's equation, thus improving the generalization of the neural network's predictions.

Keywords: Huxley muscle model, physics-informed neural networks, numerical solving of partial differential equations, multi-scale modeling.

1. Introduction

Physics-informed neural networks (PINNs) are trained to handle supervised learning tasks while respecting any given physical principle described by general nonlinear partial differential equations[1]. These neural networks represent a brand-new family of data-efficient approximators for universal functions that easily encode any underlying physical laws as prior knowledge [1]. With PINN, a key innovation is the addition of a residual network that encodes the governing physics equations, and uses the output from a deep learning network, called a surrogate, to compute a residual value [2]. The neural network is trained to reduce the differential equation's residual along with residuals of initial and boundary conditions. PINNs use automated differentiation to compute differential operators on graphs.

The fundamental PINN formulation does not require labeled data, results from other simulations, or experimental data. For PINNs, only the residual function calculation is necessary. It is also possible and sometimes required to provide simulation or experimental data for the network to be trained in a supervised way, particularly for inverse problems. The experimental or simulation data can also be used when boundary conditions or an Equation of State are missing to close a system of equations. After a PINN is trained, its inference can be used in scientific computing to replace conventional numerical solvers [2]. PINNs are a gridless technique because any point in the domain can be used as input without the need to define a mesh. Additionally, without having to be retrained, the trained PINN network can be used to predict the results on simulation grids with various resolutions [2]. Time-dependent issues can also benefit from the use of PINNs. Since time can be modeled as any other variable, it is possible to predict the output at a given moment without having to account for earlier time steps. For these reasons, unlike many conventional computational techniques, the computational cost does not scale with the number of grid points. PINN has been employed for predicting the solutions for the Burgers' equation, the Navier–Stokes equation, and the Schrodinger equation [3]. In this study, we solved Huxley's muscle equation, using PINN, to acquire the distribution of attached myosin heads to the actin-binding sites.

2. Methods

Huxley thought about the movements of the filaments within muscle and the likelihood of myosin heads connecting with actin filaments inside sarcomeres to form bridges (cross-bridges). [4]. The $n(x,t)$ function describes the rate of connections between myosin heads and actin filaments, as a function of the position of the nearest available actin-binding site relative to the equilibrium position of myosin head x:

$$\frac{\partial n(x,t)}{\partial t} - v\frac{\partial n(x,t)}{\partial t} = [1 - n(x,t)]f(x,a) - n(x,t)g(x), \forall x \in \Omega \tag{1}$$

where *f(x,a)* and *g(x)* represent the attachment and detachment rates of cross-bridges respectively, v is the velocity of filaments sliding, calculated using current and previous stretch, and a is muscle activation given as a function of time. The partial differential equation (1) can be solved using the method of characteristics with the initial condition *n(x,0)=0*. Once the $n(x,t)$ values are acquired we can calculate force F within the muscle fiber and stiffness K using the equations:

$$F(t) = k\sum_{-\infty}^{\infty} n(x,t)x\ dx \quad \text{and} \quad K(t) = k\sum_{-\infty}^{\infty} n(x,t)\ dx \tag{2}$$

where k is the stiffness of cross-bridges. Stress and stress derivative can be calculated as:

$$\sigma_m = F\frac{\sigma_{iso}}{F_{iso}} \quad \text{and} \quad \frac{\partial \sigma_m}{\partial e} = \lambda L_0 K\frac{\sigma_{iso}}{F_{iso}}, \tag{3}$$

where F_{iso} is maximal force achieved during isometric conditions, σ_{iso} maximal stress achieved during isometric conditions, L_0 the initial length of the sarcomere and λ is stretch. Calculated stresses and stress derivatives can be further used at the macro-level during finite element analysis. We used SciANN to implement PINN and integrate equation (1). This is a high-level artificial neural network API written in Python with Keras and TensorFlow backends. SciANN is designed to abstract the construction of neural networks for scientific computing and the solving and discovery of partial differential equations (PDEs) using physics-informed neural networks.

3. Results and discussion

Using the SciANN framework we constructed a neural network with 8 layers, each containing 20 neurons with a hyperbolic tangent activation function. The network is trained by minimizing the difference between actual and predicted values and also by minimizing the residuals derived from equation (1) and its initial conditions. We used Adam optimizer with a learning rate of 5×10^{-5} and batch size of 16384, during 7000 epochs. We also used the neural tangent kernel (NTK) method to get the adaptive weights, balancing the difference between the number of points, used to minimize the residual of PDE, and the number of points used to minimize the residual of the initial condition. We also trained ordinary multilayer perceptron (MLP) with the same architecture as PINN and we used the same data, but without providing the specificity of Huxley's muscle equation to the network.

Neural network:	**PINN**		**MLP**	
Identification number of numerical experiment	**Correlation coefficient (stress)**	**Correlation coefficient (stress derivative)**	**Correlation coefficient (stress)**	**Correlation coefficient (stress derivative)**
1	0.9929	0.9943	0.9852	0.9872
2	0.9860	0.9860	0.7782	0.8925
3	0.9972	0.9958	0.9902	0.9959
4	0.9343	0.9584	0.1286	0.8354
5	0.9817	0.9909	0.9964	0.9956
6	0.9962	0.9893	0.0884	0.8561
7	0.9978	0.9855	0.1559	0.7753
Average value:	**0.9837**	**0.9857**	0.5890	0.9054
Standard deviation:	**0.0209**	**0.0117**	0.4088	0.0823

Table 1. Correlation coefficients between original values, obtained by the method of characteristics, and predicted values, obtained by neural networks. Shown numerical experiments were used to acquire the data and train the neural networks.

Once the networks were trained, we integrated them into the finite element solver and used them at the micro-level instead of the method of characteristics. In Table 1, we show the correlation coefficients between original values, obtained in finite element simulations with the method of characteristics at the micro-level, and predicted values, obtained in simulations with the neural network at the micro-level. We presented acquired stresses and stress derivatives. These values were obtained in numerical experiments that were used to collect the data and train the neural networks. It can be seen that stresses and stress derivatives obtained with PINN are closer to the original values than the values obtained by MLP. In Table 2, numerical experiments that were not used in the training set are shown. It can be seen that PINN performed better in these experiments, which indicates that PINN generalizes better than the standard MLP.

Neural network:	PINN		MLP	
Identification number of numerical experiment	Correlation coefficient (stress)	Correlation coefficient (stress derivative)	Correlation coefficient (stress)	Correlation coefficient (stress derivative)
8	0.8861	0.9592	0.6968	0.8795
9	0.9704	0.9811	0.1854	0.7707
Average value:	**0.9283**	**0.9702**	0.4411	0.8251
Standard deviation:	**0.0421**	**0.0109**	0.2557	0.0544

Table 2. Correlation coefficients between original values, obtained by the method of characteristics, and predicted values, obtained by neural networks. Shown numerical experiments were used to test the neural networks.

4. Conclusions

In our article, we presented alternative methods to find approximate solutions of Huxley's muscle contraction equation using neural networks. We collected data from simulations and trained multilayer perceptron to predict cross-bridge formation probabilities. In addition to using the data, we also informed the neural network by calculating the residual of the Huxley equation, which resulted in an improvement of the neural network's ability to generalize predictions.

Acknowledgement: This research was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 952603 (http://sgabu.eu/). This article reflects only the author's view. The Commission is not responsible for any use that may be made of the information it contains. Research was also supported by the SILICOFCM project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777204. This article reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information the article contains. The research was also funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia, contract numbers [451-03-68/2022-14/200107 (Faculty of Engineering, University of Kragujevac) and 451-03-68/2022-14/200378 (Institute for Information Technologies Kragujevac, University of Kragujevac)].

References

[1] Raissi M., Perdikaris P., and Karniadakis G. E., Physics Informed Deep Learning (Part I): Data-Driven Solutions of Nonlinear Partial Differential Equations. New York City, NY: arXiv preprint arXiv:1711.10561. 2017a .

[2] Markidis Stefano, The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers, Frontiers in Big Data, Vol. 4, ISSN 2624-909X 2021. https://doi.org/10.3389/fdata.2021.669097

[3] 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. 378 686–707. 2019.

[4] Williams, W.O., Huxley's Model of Muscle Contraction with Compliance. Journal of Elasticity, 105, 365-380. 2011.