Neural Network-Based Rapid Predictor of Biological Nerve Fiber Activation

Justin Golabek[1], Connor Dupuis[1], Shreya Saxena[1], Matthew Schiefer[2], and Erin Patrick[1]

[1] Department of Electrical & Computer Engineering, University of Florida, USA
[2] Malcom Randall Department of Veterans Affairs Medical Center, Gainesville, FL, USA

E-mail: jgolabek1@ufl.edu

Abstract

Objective. To greatly enhance the clinical effectiveness of electrical neuromodulation therapies for various medical conditions (e.g., pain, movement disorders, depression, and sensorimotor prosthetics), efficient and patient-specific computer models are needed for the prediction of the neural response. The objective of this work is to develop a method based on deep-learning neural networks to replace computationally expensive methods for the prediction of neural fiber activation by electrical stimulation. A neural network-based, rapid predictor will decrease the computational demands and allow for implementation of real-time patient-specific models. *Approach.* We developed an artificial neural network (ANN) to predict the binary activation response of myelinated nerve fibers from deep brain stimulation. To generate training data, we used COMSOL to create an FEM model of the DBS electrode and surrounding neural tissue, and the NEURON simulation environment to simulate axons within this tissue. Training, testing, and optimization was parallelized across 200 nodes on the UF supercomputer, HiPerGator. We compared our ANN performance to NEURON results across 3 unique datasets, comparing both predicted thresholds and fiber recruitment. *Main results.* The optimized ANN could predict thresholds for both orthogonal and parallel fibers with less than 0.1 V mean error. For a more complex dataset, the ANN estimated recruitment with high accuracy, even if predicting on pulse widths not explicitly trained on. *Significance.* We demonstrate that an artificial neural network, if constructed and trained properly, can be an effective predictor of single fiber activation from electrical stimulation.

Keywords: artificial neural network, deep brain stimulation, biophysical computational modelling, fiber recruitment

1. Introduction

Computational models can serve as powerful tools to enable non-invasive experimentation with neuromodulation technologies. Multistep simulation frameworks have been used to optimize neural interface design parameters as well as deep brain stimulation (DBS) procedures [1,2]. DBS is an application of particular interest due to its effectiveness in treating a wide range of neurological disorders including Parkinson’s disease, essential tremor, and Tourette syndrome [3,4,5].

A critical component in the DBS simulation workflow is the prediction of neural activation in response to stimulation. Different types of predictor models exist, each with advantages and disadvantages. One model that is commonly used in both clinical and research settings is the volume of tissue activated (VTA) [6]. VTA models have been effectively applied in the evaluation of ideal stimulation locations [7]. These models however make several assumptions and simplifications that limit their utility in more complex parameter spaces. One such example is the assumption that fiber pathways are straight and orthogonal to the DBS electrode [8].

Multicompartment cable models are markedly different from VTA models in that they consist of circuit elements representing anatomical features of the axon. Simulation frameworks that use these cable models have been shown to accurately reproduce experimental data [9]. Predicting activation with cable models however requires solving the dynamic equations governing the electrical circuits. The computational expense required for this typically prohibits them from being of use in clinical analyses.

Driving-force models are a middle ground between VTA and cable-model predictors. These models specifically use the second spatial difference of the applied extracellular potential to predict axonal response, and have been shown to achieve higher accuracies than VTA-based predictors while remaining computationally inexpensive [10, 11]. However, like VTA, driving-force methods suffer from inaccuracies due to their simplification from cable models, particularly when predicting the activation of fibers with non-straight trajectories.

The correlation between accuracy and computational expense in simulating neural activation thus remains, calling for the development of a new type of predictor model that reduces computation cost while retaining high accuracy. In this study, we explore the use of an artificial neural network (ANN) to this end. This is not the first attempt to apply ANNs to computational DBS. McIntyre et al created an ANN to predict the VTA from the stimulation parameters and electrode configurations [12]. The inherent drawbacks of VTA however limit the usefulness of their model. For this reason, we investigate the usage of an ANN to model the dynamics of nerve fibers rather than VTA. The ideal ANN-based predictor would achieve maximum generalization across the application space, minimizing both bias and variance. We focus here on an early iteration of this predictor model within a subset of the DBS space.

2. Methods

Our methodology for generating datasets, as well as our experimental design for ANN development, is described in the following sections.

2.1 Finite Element Models

We constructed a finite element model of the DBS apparatus with the lead in a box approach described by Howell et al. [13]. Briefly, we used COMSOL to model a Medtronic 3387 DBS electrode (Medtronic, Minneapolis, MN) centered within a 30 mm cubic region of bulk tissue. The bulk tissue medium was homogenous and isotropic, and assigned a conductivity of 0.2 S/m. The lead was surrounded by 0.5 mm-thick encapsulation tissue with a conductivity of 0.07 S/m [14]. The contact second from the bottom was represented as active by assigning a value of -1 V potential to its surface, and the surfaces of the bulk tissue cube were specified as ground. The resulting potential field represented that produced by the DBS electrode at a -1 V amplitude setting, and we linearly scaled this solution to simulate stimulation of variable amplitude.

2.2 Axon Models

We created multi-compartment double-cable models of 5.7- μm diameter axons using the MRG model within the NEURON simulation environment [15, 16]. The MRG model uses explicit representations of the nodes of Ranvier, paranodal, and internodal sections of the axon, and has been shown to accurately reproduce experimental data [17]. We modified the MRG model by adding NEURON's extracellular mechanism to each compartment, allowing us to directly simulate fiber stimulation by extracellular potential rather than equivalent intracellular current injection.

Extracellular stimulation of the axon was implemented in three steps. First, we obtained the potential value at the center of each compartment using data from our FEM model, linearly interpolating as needed. Second, these values were scalar-multiplied to the time-varying stimulus waveform vector to create stimulus vectors for each compartment. Finally, these vectors were applied to the axon compartments with NEURON's *play* method. All NEURON simulations used a timestep of 5 μs.

We determined activation threshold via a binary search where we linearly scaled the stimulus amplitude until a precision of .005 V was reached. We used a single pulse-spike pair to determine whether an axon became activated due to stimulation, and our pulses were exclusively monophasic cathodic. For each simulation, we stored the nodal extracellular potentials, pulse-width, and threshold multiplier. This data allowed us to create arbitrarily large datasets as we could create any number of binary activation examples around a single threshold.

2.3 ANN-based Predictor

We used TensorFlow in python for all steps of the machine learning process, including ANN creation, training, optimization, and testing [18]. Dataset generation using NEURON simulation was performed in parallel across 200 CPUs on the UF supercomputer, HiPerGator. We optimized our ANN by running hyperparameter searches across a wide range of parameters and selecting the parameter combination that resulted in the highest accuracies.

2.3.1 Model Architecture

Our ANN is a fully connected, feedforward architecture. We constructed the ANN in TensorFlow by stacking dense *Keras* layers within the *Sequential* model. Three input features characterize the extracellular stimulation of a single fiber, and the binary output represents its predicted activation. The first of the input features is the stimulus pulse width, which we assume to belong to an ideal monophasic cathodic square pulse. The second feature is an array of nodal extracellular potentials. The last is an array of the second spatial differences of the nodal extracellular potentials, although these arrays need not be the same size. We included the size of each of these two array features as hyperparameters in our searches, which are discussed in the validation section. Both arrays are centered around the node where the second spatial difference of the extracellular potential is maximal. Our use of the second spatial difference in this way is motivated by its correlation with the membrane current induced at the node; we assume that activation is most likely to originate from this node [19]. Inputs are pre-processed by taking the feature standard-scores based on the means and standard deviations of the training dataset.

2.3.2 Training

Table 1. Model Inputs and Outputs

Inputs	Output
Pulse width (ms)	Activation
Extracellular potential array (mV)	
Second spatial difference array (mV)	

To train our ANN, we generated a regular grid population of straight fibers orthogonal to the DBS electrode, shown in Fig 2(A). Electrical stimulation of each fiber was simulated using in NEURON across a discrete range of stimulus pulse-widths. Binary-activation training examples were derived from the NEURON runs by generating a normal distribution of 400 stimulus amplitudes around the stimulus threshold and labelling each case by comparing its amplitude to the threshold. Training was performed using TensorFlow's default *compile* and *fit* methods. We selected binary cross-entropy loss and stochastic gradient descent with the Adam optimizer [20].

2.3.3 Validation and Optimization

Neural networks have several parameters not determined through training, known as hyperparameters, whose configuration can drastically alter model performance [21]. Examples of these include those that relate to network structure (e.g., number of layers, number of nodes per layer, activation function for each node) and those that relate to the training/learning process (e.g., learning rate, epochs, batch size). We explored these and several additional hyperparameters of interest, which are listed in table 2. Because the optimal configuration of these parameters cannot be known *a priori*, we used random searches across the parameter ranges of interest and tested each ANN on a validation dataset using TensorFlow's *evaluate* method [22]. Each model configuration was independently trained and tested three times and the average accuracy on the validation data was used as the comparison metric. The hyperparameter combinations that produced the highest performing ANN were then stored and used to construct our final model. These values are also shown in table 2.

The validation set was similar to the training set but with two distinctions. First, the fibers were positioned parallel to the electrode instead of orthogonal. Second, the fiber position density in the y dimension was quadrupled. This setup, shown in figure 2(B), allowed us to select the optimal hyperparameter combination based on the model's ability to accurately predict data not explicitly trained on.

2.3.4 Testing

Table 2. Summary of Hyperparameter Search

Hyperparameter	Search Range	Optimal Value
Hidden layers	1-4	1
Units per h-layer	15-500	250
Dropout per layer	0-0.5	0
Learning rate	{0.0001,0.001, 0.01, 0.1}	0.001
Epochs	1-6	4
Batch size	{32, 62, 128}	32
Activation Function for h-layers	{relu, tanh, sigmoid}	relu
Size of extracellular potential input array	{1, 3, 5, 7}	3
Size of second spatial difference input array	{1, 3, 5, 7}	7

We generated a new test set different than the training and validation sets to ensure an unbiased measure of model performance. This set was based off the orthogonal set but added random fiber angles in the x-z plane. Additionally, fibers were placed further in the y direction than in either of the other datasets.

2.4 Performance Analyses

We compared the predictions of our final optimized ANN to the NEURON results for each of the discussed datasets. For the training and validation sets, we focused primarily on the predicted thresholds for the fibers in each population. For the test set, we instead analyzed fiber recruitment by sweeping the

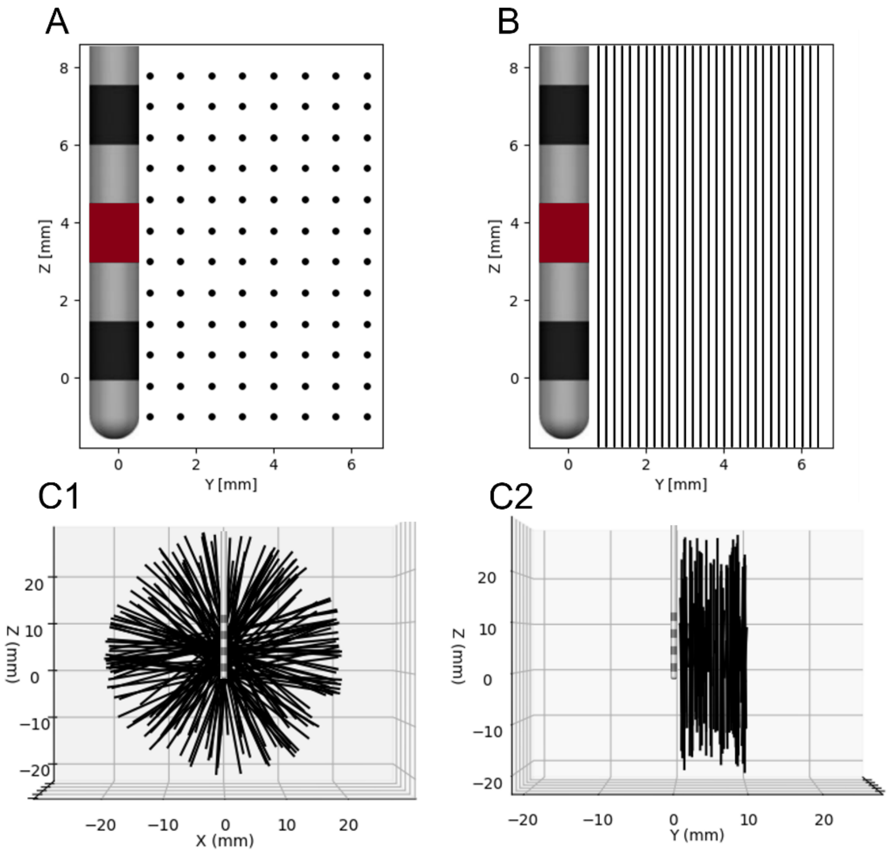

Figure 1. Fiber orientations of different datasets. (A) The training set: fibers are oriented orthogonal to the DBS electrode (i.e., coming out of the page). (B) The validation set: fibers are oriented parallel to the DBS electrode. Each fiber extends well beyond what is shown, to approximately z = ±20 mm. (C) Different views of the test set, which expands upon the training set by adding random angles in the x-z plane.

stimulus amplitude from 0–10 V and comparing the percentage of fibers that were predicted to activate.

3. Results and Discussion

3.1 Training Set Performance

We evaluated our final ANN on a dataset identical to that used in training except that fiber density was quadrupled across both spatial dimensions. This setup allowed us to concurrently assess the model's performance on the training data and its interpolation ability on untrained fiber positions. Figure 3(A) presents the absolute error distributions of the ANN's predicted thresholds as a function of the trained pulse-widths. All but the predictions at the lowest pulse width are within ±0.1 V of the expected thresholds, and the mean absolute value of the errors is 0.067 V. Outlying error points are not shown to improve viewing.

Figure 3(B) plots the absolute threshold error as a function of threshold amplitude as computed by NEURON. Predictions at all pulse-widths are included in this graph, but are limited to the realistic stimulus amplitude range of 0–10 V. No significant bias is observed on this range, and a dense collection of points is seen to be grouped around zero error across amplitude. We show the error trend across space for a pulse-width of 90 µs in Figure 3(C). No clear trends are visible, although distinct error regions are seen that vertically align with the inactive contacts although slightly separated from the electrode surface. Plotting the second spatial difference profiles at these positions (not included) showed noise not typically present in our training examples, explaining the error.

3.2 Validation Set Performance

The ANN's performance on the validation set was studied in the same way as the training set. Figure 3(D) plots the error distributions as a function of pulse width and shows absolute errors of near-zero at all pulse widths except 30 µs. The mean absolute value of the errors is in this case is 0.05 V. Prediction accuracy as a function of threshold amplitude is likewise shown in figure 3(E) and shows no significant trends in error. At pulse width = 90 µs, figure 3(F) continues in this direction by showing errors extremely close to zero, except at 0.6 mm from the electrode where 2 predictions have errors at around

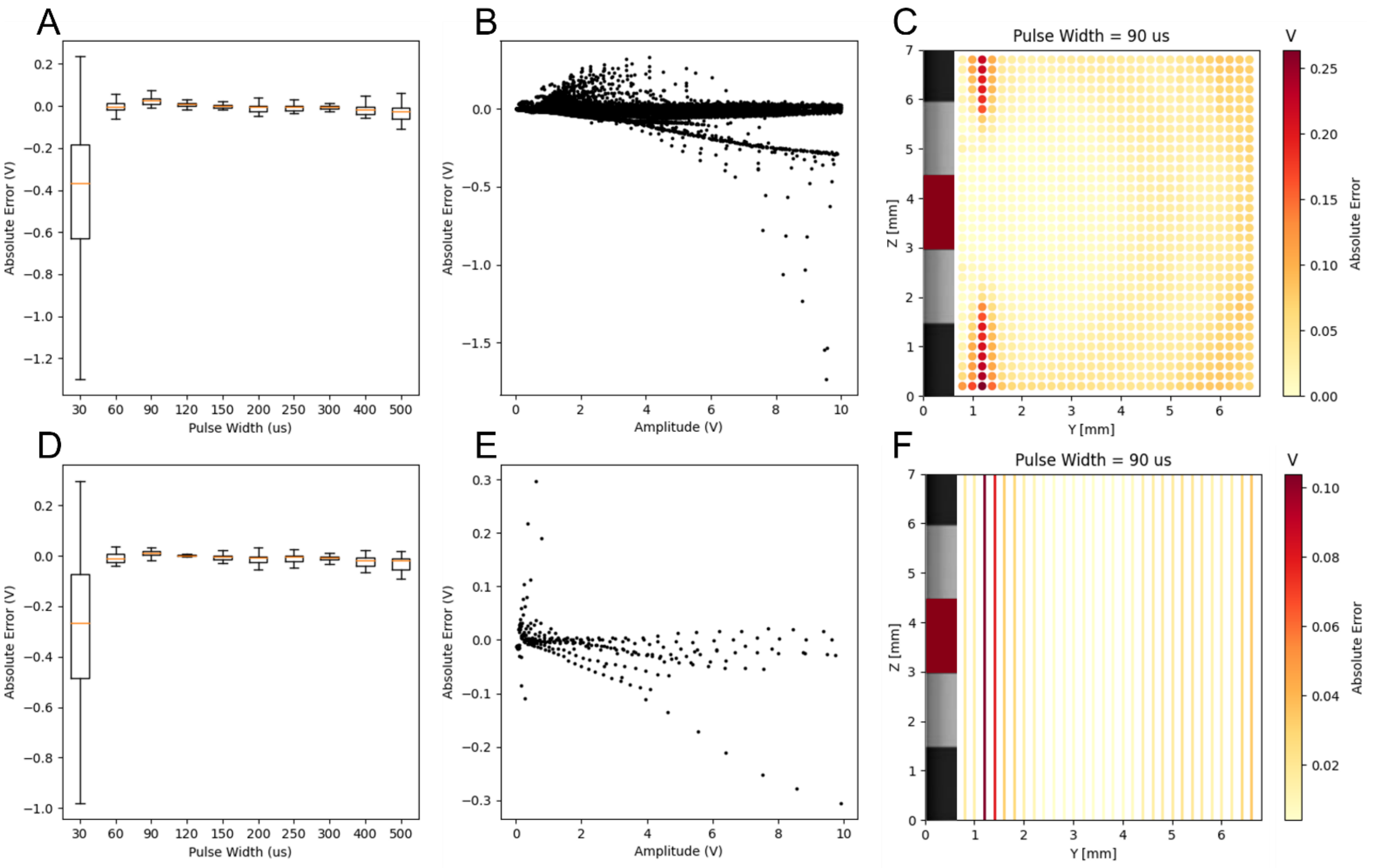

Figure 2. ANN performance results for both training and validation sets. (A) The training set performance across pulse width. (B) The training set across actual threshold amplitude. (C) The training set across space for a single pulse width case. (D) The validation set performance across pulse width. (E) The validation set across actual threshold amplitude. (F) The validation set across space for a single pulse width case.

0.1 V. The ANN's performance on this parallel dataset demonstrates its ability to effectively predict fibers orientations not explicitly trained on.

3.3 Test Set Performance

The ANN's performance on the test set was assessed differently than the training and validation sets. The recruitment of the fiber population was compared as opposed to directly comparing predicted thresholds. Here, we investigated the recruitment at 3 pulse width values (45, 90, and 450 µs), 2 of which were not trained on (45 and 450 µs). This gave a look at our model's ability to interpolate in the pulse width space. At pulse width = 45 µs, the ANN estimated recruitment with an average absolute value error of 1.05%. At pulse width = 90 µs, this average dropped to 0.62%. At pulse width = 450 µs, the average error decreased further to 0.53%. Each of these results indicates near perfect recruitment estimation by the ANN, which is especially significant given that two of the pulse widths were not included in the training set.

3.4 Limitations and Next Steps

The primary limitation of this study is the small size of the parameter space chosen to train and predict on. Factors like electrode-node alignment, non-straight fiber pathways, anisotropic tissue conductivities, and multi-contact stimulation are known to introduce error in other predictor models but have yet to be investigated within our ANN methodology. Our next steps are thus to introduce these factors one-by-one and observe if high accuracy is retained. For cases like multi-contact stimulation, it's possible that a new training dataset may need to be created for an ANN-based predictor to be successful. Also absent from this study is an analysis of the computational expense of our ANN-predictor in comparison to NEURON, VTA, and driving-force predictor models. As one of the primary motivations for this study, this will be investigated shortly. Assuming that we achieve high accuracy and low computational expense, we plan to apply our ANN to tissue conductivities and fiber pathways obtained from diffusion tensor imaging data, creating an ANN-based pathway activation model.

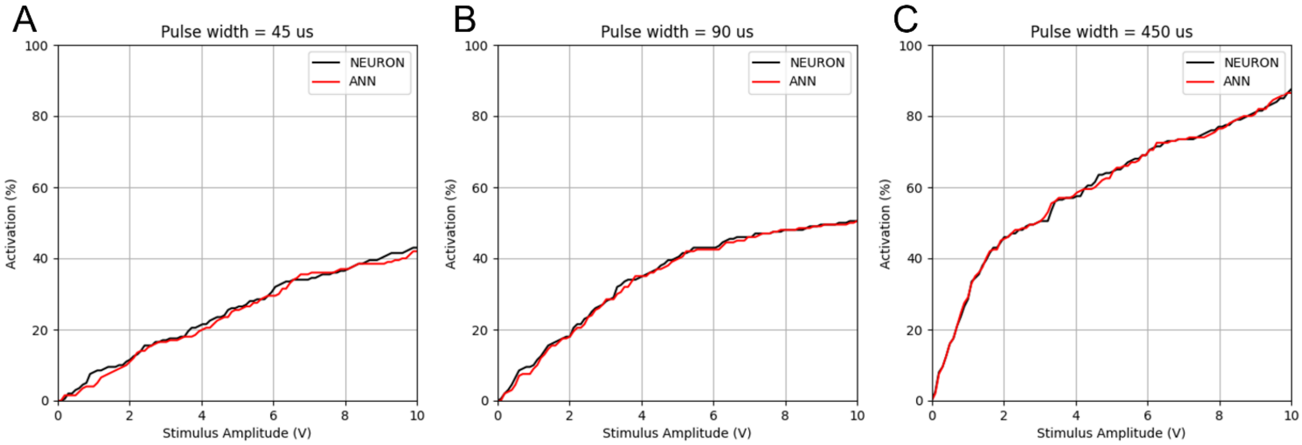

Figure 3. ANN fiber recruitment results for the test set. (A) Recruitment comparison at pulse width = 45 us. (B) Recruitment comparison at pulse width = 90 us. (C) Recruitment comparison at pulse width = 450 us

4. Conclusion

We've shown how an artificial neural network (ANN) might be constructed and optimized to accurately predict the activation of single myelinated nerve fibers exposed to electrical stimulation. While our study encompassed a relatively small parameter space, we open the door to further iterations that might supplant existing predictor models by achieving high accuracy and low computational expense.

References

1. Zelechowski M, Valle G, Raspopovic S. A computational model to design neural interfaces for lower-limb sensory neuroprostheses. Journal of NeuroEngineering and Rehabilitation. 2020 Feb 19;17(1):24.

2. Gunalan K, Chaturvedi A, Howell B, Duchin Y, Lempka SF, Patriat R, et al. Creating and parameterizing patient-specific deep brain stimulation pathway-activation models using the hyperdirect pathway as an example. PLoS One. 2017;12(4):e0176132.

3. Benabid AL. Deep brain stimulation for Parkinson's disease. Current Opinion in Neurobiology. 2003 Dec 1;13(6):696–706.

4. Lyons KE, Pahwa R. Deep brain stimulation and tremor. Neurotherapeutics. 2008 Apr;5(2):331–8.

5. Ackermans L, Temel Y, Cath D, van der Linden C, Bruggeman R, Kleijer M, et al. Deep brain stimulation in Tourette's syndrome: Two targets? Movement Disorders. 2006;21(5):709–13.

6. Butson CR, Cooper SE, Henderson JM, McIntyre CC. Patient-specific analysis of the volume of tissue activated during deep brain stimulation. NeuroImage. 2007 Jan 15;34(2):661–70.

7. Akram H, Sotiropoulos SN, Jbabdi S, Georgiev D, Mahlknecht P, Hyam J, et al. Subthalamic deep brain stimulation sweet spots and hyperdirect cortical connectivity in Parkinson's disease. NeuroImage. 2017 Sep 1;158:332–45.

8. Gunalan K, Howell B, McIntyre CC. Quantifying axonal responses in patient-specific models of subthalamic deep brain stimulation. NeuroImage. 2018 May 15;172:263–77.

9. Chaturvedi A, Butson CR, Lempka SF, Cooper SE, McIntyre CC. Patient-specific models of deep brain stimulation: Influence of field model complexity on neural activation predictions. Brain Stimulation. 2010 Apr 1;3(2):65–77.

10. Howell B, Gunalan K, McIntyre CC. A Driving-Force Predictor for Estimating Pathway Activation in Patient-Specific Models of Deep Brain Stimulation. Neuromodulation. 2019 Jun;22(4):403–15.

11. Peterson EJ, Izad O, Tyler DJ. Predicting myelinated axon activation using spatial characteristics of the extracellular field. J Neural Eng. 2011 Aug;8(4):046030.

12. Chaturvedi A, Luján JL, McIntyre CC. Artificial neural network based characterization of the volume of tissue activated during deep brain stimulation. J Neural Eng. 2013 Oct;10(5):056023.

13. Grill WM, Thomas Mortimer J. Electrical properties of implant encapsulation tissue. Ann Biomed Eng. 1994 Jan 1;22(1):23–33.

14. McIntyre CC, Richardson AG, Grill WM. Modeling the excitability of mammalian nerve fibers: influence of afterpotentials on the recovery cycle. J Neurophysiol. 2002 Feb;87(2):995–1006.

15. Hines ML, Carnevale NT. The NEURON simulation environment. Neural Comput. 1997 Aug 15;9(6):1179–209.

16. Wongsarnpigoon A, Woock JP, Grill WM. Efficiency Analysis of Waveform Shape for Electrical Excitation of Nerve Fibers. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2010 Jun;18(3):319–28.

17. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning. :21.

18. HiPerGator – Research Computing [Internet]. [cited 2021 Nov 18]. Available from: https://www.rc.ufl.edu/services/hipergator/

19. Warman EN, Grill WM, Durand D. Modeling the effects of electric fields on nerve fibers: Determination of excitation thresholds. IEEE Transactions on Biomedical Engineering. 1992 Dec;39(12):1244–54.

20. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980 [cs] [Internet]. 2017 Jan 29 [cited 2021 Dec 15]; Available from: http://arxiv.org/abs/1412.6980

21. Probst P, Bischl B, Boulesteix A-L. Tunability: Importance of Hyperparameters of Machine Learning Algorithms. arXiv:180209596 [stat] [Internet]. 2018 Oct 22 [cited 2021 Nov 17]; Available from: http://arxiv.org/abs/1802.09596

22. Bergstra J, Bengio Y. Random Search for Hyper-Parameter Optimization. :25.