Author: Bryan Howell

Created: 03/26/2024

A first draft of the code stack for MetaStim is complete with a Jupyter notebook for demoing its functionality. The next step is to review and stress test the code before proceeding to publishing the library on PyPi. Code review will be done to reorganize the code if needed, and to add or remove any functionality. The stress test will identify bugs and confirm that the code is ready for production. Work on final visualizations and documentation will be halted until the code review and stress test are completed.

**Code Reorganization**

Background

* FieldANN takes in the electrode config list and xyz coordinates as independent variables. FieldANN also takes in a stimulation amplitude (independent) but this is used only for scaling the FieldANN output. See Slide 5 in MetaStimIntro-Howell.pptx. The xyz coordinates do not need to be the axonal coordinates. The user can work with the FieldANN separately and define one or many points that don’t constitute axons.
* AxonANN takes in a stimulus pulse width (PW), axon diameter (D), and axonal xyz coordinates as independent variable inputs. AxonANN also takes in axonal distances (r) and electrode case (EC) parameters as dependent inputs. There is one r parameter per axon. There is one EC parameter per axon. AxonANN calls the FieldANN to calculate the field (Φ) across the axons, which is transformed into the second difference of Φ (SD). SD is a dependent input into the AxonANN. See Slide 6 in MetaStimIntro-Howell.pptx.

Modifications

Split AxonANNModel into two classes with respective methods: FieldANN and AxonANN.

Have the output be either raw or classified. AxonANN also takes in the stimulation amplitude to classify the AxonANN output as inactive (output > amp) or inactive (output < amp). The output is returned to the user as raw (no classification) or classified (1 = active or 0 = inactive, as described above).

Return a 2nd vector, or…

[A, B] = my\_function(...)

A => [0, 1, 0, 0, 1, 1…]

B => [5.1, 7.2, 3.1, …]

Return either A or B based on input parameter

Output = my\_function(..., ‘output’=raw)

A => [0, 1, 0, 0, 1, 1…]

B => [5.1, 7.2, 3.1, …]

Keep the current utility function for calculating axonal coordinates (axons parallel to lead) separate from ANN classes. The axonal data will be provided as an atlas of precalculated brain pathways in Phase 2. For now, the utility function is a stopgap to provide axonal coordinates to use for AxonANN.

**Stress Test & Organization**

**Provisional Parameters**

Min\_distance, max\_distance, num\_axons are provisional parameters; they are used to create some ideal axons parallel to the lead to run the code. In Phase 2, we will introduce real axon bundles in human brains. For now, this is only a stopgap parameter set for getting the v1 code launched. I think we can keep the parallel axon generator as a separate utility for doing some gut checks for research users.

**Parameter Validation**

Negative pulse width

Update error message: “Negative pulse width! Pulse width must be > 0.”

Pulse width (0 < pw < 30 OR pw > 500um).

The warning for degradation of accuracy should only be thrown for positive values outside the range.

Negative fiber diameter

Update warning message: “Negative axon fiber diameter! Diameter must be > 0.”

Return to default value = 6 um and proceed.

* Show units of micrometers (um) when saying defaulting to x.

Separate warning also thrown for fiber diameters outside the range, 1.5-15um. See below.

Small and large fiber diameters (0 < D < 1.5 OR D > 15um)

The warning for degradation of accuracy should only be thrown for positive values outside the range.

Separate warning also thrown for fiber diameters outside the range, 1.5-15um.

Electrode list

Only for context: Values must be -1, 0, or 1. -1 = cathode or negative electrode. 0 = off or passive electrode. 1 = anode or positive electrode.

If values outside the above 3, throw a warning:

Update warning message:

“Warning! Electrodes can be set as a cathode (-1), anode (1), or off (0). All non-zero values have been adjusted to -1 or 1.”

Lead radius

The user selects the lead model (<xxxx> number). This variable is set from a csv file of lead parameters. The user will not modify this.

**Re-organization**

* Split AxonANNModel into two classes: FieldANN and AxonANN. See Code Review above.
* Only recalculate fieldANN when the electrode\_list is adjusted.
* Do not recalculate fieldANN when the stimulation amplitude (A) is adjusted. The fieldANN calculated the solution at A = 1V. Therefore, for a set electrode list, you can simply scale the output of the 1V fieldANN by A to get the fieldANN at any A.
* Only recalculate axonANN when pulse width or fiber diameter are adjusted. Same rules apply to fieldANN for when axonANN calls it.
* Do not recalculate axonANN when A is adjusted. A is only used to check if the output of axonANN is below A (axon activated) or above A (axon not activated).