Building a Reinforcement learning Tool to Optimize computational models of Biological neurons

Adel R. Refat, Eslam K. Korany, Ahmed A. Mohamed , and Mahmoud Monem

***Abstract*— Objective: Creating automated tool to replace manual hand tuning of neural model parameters using reinforcement learning. Methods: Using NEURON's python package as the simulation tool and reinforcement learning to find the optimal neural channels parameters, The tool should be able to load a neural model that may include at least 5 compartments , allow the user to enter the experimental measurements set that would be used as a criterion to validate the neural model parameters,then provide a list of the neural channels parameters that were successfully +able to reproduce experimental measurements provided by the user.**

***Index Terms*— Neuron model,Neuron optimization, Reinforcement learning**

I. Introduction

Neural computational models are a great resource in studying the nervous system, and muscle control, they allow studying the cellular mechanisms and simulating neural signals that are impossible in acquisition from living animals.

Neuron computational models are complex non-linear systems that require high computational resources in operation and development. To bring these models into their optimum use, They require optimization to certain electrical behaviors and measurements to match and mimic experimental recordings from living cells, so that simulated data from these models would be related to the living systems. Due to the high complexity of these models, manual neuronal parameter hand tuning is the main method used in developing these models and calibrating them to experimental data. Other methods used are parameter space exploration and gradient descent.

Neuronal parameter optimization is the process of identifying sets of parameters that lead to a desired electrical activity pattern in a neuron or neuronal network model that is not fully constrained by experimental data. This process requires a lot of effort and knowledge from the scientist and can be very challenging since the underlying mechanisms are highly non-linear and difficult to grasp.

The goal of this project is to create an automated and efficient pipeline for model optimization, By using reinforcement learning techniques that will replace hand-tuning processes,This will help save time and resources in calibrating neural computational models to experimental measurements.

II. Methods

A. NEURON

The NEURON software, developed at Duke University, is a simulation environment for modeling individual neurons and networks of neurons. The NEURON environment is a self-contained environment allowing interface through its GUI as in Fig. 1, or via scripting with hoc or python as shown in Fig. 2.

We will be using NEURON – VERSION 7.8.1 as our primary simulation tool, NEURON is primarily used to simulate the full experiments on the models, starting from stimulating the model, recording the model's behavior, and tuning the model's parameters according to the desired behavior.

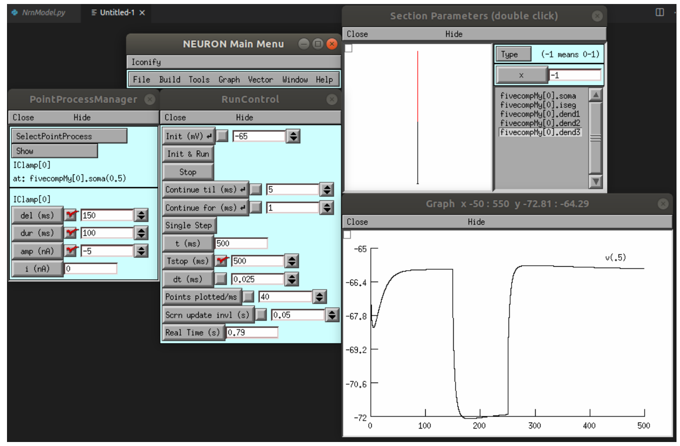

Fig. 1. NEURON's GUI, Stimulating a five-compartment model with -5 (nA) current at the center of the soma, for a duration of 100 (nSec) after 150 (nSec) from the start of the simulation, The graph is the voltage (mV) versus time (mSec) behavior of the model.

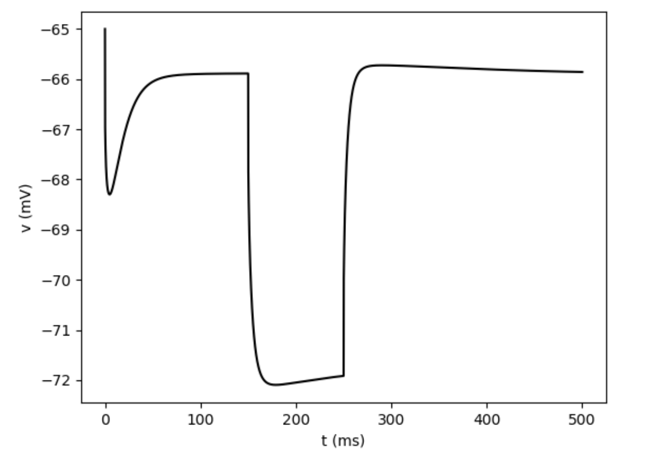

Fig. 2. Same settings and output as in Fig. 1, But using NEURON's python package.

B. Five-Compartment model

Our starting model is a five-compartment model consisting of a soma, axon hillock, and three segment Dendrites, we won't go into the details of the models as it would be provided to us and we will work on them without change.