

Neural Models Optimization Toolkit

Ahmed Adel Mohamed
SBME, Faculty of Engineering
Cairo University
a.muhamed98@gmail.com

Adel Refat Ali
SBME, Faculty of Engineering
Cairo University
adel.elmala2020@gmail.com

Eslam Khaled Korany
SBME, Faculty of Engineering
Cairo University
eslamkhaled64@gmail.com

Mahmoud Abd Elmonem Abd Elgawwad
SBME, Faculty of Engineering
Cairo University
mahmoud.monem.elgohary@gmail.com

Abstract—Neural computer models are a great resource in studying the nervous system, and muscle control. They allow studying cellular mechanisms and simulating neural signals that are impossible to acquire from living animals. Neuron computer 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 hand tuning is the main method used in developing these models and calibrating them to experimental data. The goal of this project is to build a semi/full automated optimization tool to help save time and resources in calibrating neural computer models to experimental measurements. This optimization tool should be able to learn the system parameters that reproduce experimental data.

Keywords—Optimization, Neuron, Model, Parameters, Automated Tuning, Error Function, Fitness Function, Genetic Algorithms, Multi Objective Optimization

I. INTRODUCTION

One of the most difficult challenges for a neuro-scientist developing a detailed computational model is determining how to tune model parameters that cannot be derived directly from experimental results. In particular for neuroscientists, neuronal models consisting of many compartments incorporating multiple types of voltage-gated ion channels all of which require separate parameters are developed in complex models. As scientists in the field have seen the sharp increase of computational power and a growing knowledge of the neuronal mechanisms behind the neuronal function, These models have become increasingly complex, resulting in an increase in the number of parameters that must be fitted. In the most extreme case, each compartment or neuron may have its own set of parameters.

Until recently, the traditional approach was to manually tune neuronal model parameters. This necessitates a significant amount of effort and knowledge on the part of the scientist and can be extremely difficult due to the highly nonlinear and difficult-to-understand underlying mechanisms. It can also introduce bias because the scientist has a natural tendency to assign different roles to each parameter in advance. Complex models, on the other hand, have been successfully developed in this manner.

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.

Single neurons and neuronal networks intended to reproduce an

experimentally observed electrical behavior are modeled with systems of differential equations that contain parameters such as (but not limited to):

- Membrane capacitance, maximal conductances, half activation and inactivation voltages and time constants of individual ionic currents, axial resistance, and morphological parameters such as cell size and axon or dendrite branch structure, length(s) and diameter(s).

It is almost never possible to measure all parameters required to fully constrain the model in a single experimental preparation in the biological neurons and networks that inspire these models. Furthermore, the properties of neurons and networks differ even within the same species or within the same animal, and strategies such as

- Combining a subset of parameters measured in animal A with another subset measured in animal B.
- Obtaining model parameter values by averaging over measurements of the same parameter in different animals.

Usually, the desired model behavior is not produced. Starting with a set of differential equations that represents a neuron or network model, it is frequently necessary to find sets of model parameters that approximate the desired behavior using methods other than experimental measurement.

II. BACKGROUND

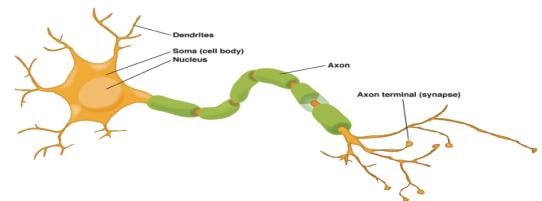
2.1. Nerve Cells

2.1.1. Nerve Cells Topology

The Nervous System Has Two Classes of Cells: nerve cells, or neurons, and glial cells, or glia. Nerve Cells Are the Signaling Units of the Nervous System.

A typical neuron has four morphologically defined regions:

- Cell body (soma)
- Dendrites
- Axon
- Presynaptic terminals.

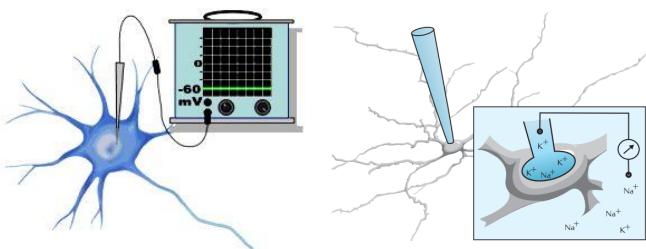


The cell body (soma) is the metabolic center of the cell. It contains the nucleus, which contains the genes of the cell.

the nucleus, which contains the genes of the cell. The cell body usually gives rise to two kinds of processes: several short dendrites and one long, tubular axon. Dendrites branch out in tree-like fashion and are the main apparatus for receiving incoming signals from other nerve cells. The axon typically extends some distance from the cell body and carries signals to other neurons. These electrical signals, called action potentials, are initiated at a specialized trigger region near the origin of the axon called the initial segment from which they propagate down the axon.

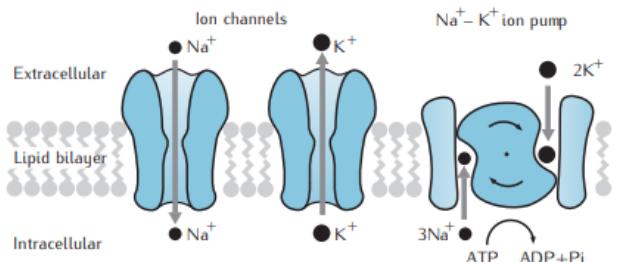
1. Membrane Potential

In neurons, as in other cells, a measurement of the voltage across the membrane using an intracellular electrode the figure below shows that there is an electrical potential difference across the cell membrane, called the membrane potential. The resting membrane potential is



typically around -65mV , meaning that the potential inside the cell is more negative than that outside.

The electrical properties which underlie the membrane potential arise from the separation of intracellular and extracellular space by a cell membrane. The intracellular medium, cytoplasm, and the extracellular medium contain differing concentrations of various ions. Some key inorganic ions in nerve cells are positively charged cations, including sodium (Na^+), potassium (K^+), calcium (Ca^{2+}) and magnesium (Mg^{2+}), and negatively charged anions such as chloride (Cl^-).



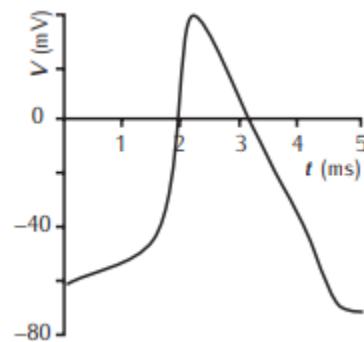
Typically, there is a greater concentration of extracellular sodium than intracellular sodium, and conversely for potassium.

2. Ion Channels

Ion channels are pores in the lipid bilayer, which can allow certain ions to flow through the membrane. Types of ion channels, referred to as active channels, can exist in open states, where it is possible for ions to pass through the channel, and closed states, in which ions cannot permeate through the channel. Whether an active channel is in an open or closed state may depend on the membrane potential, ionic concentrations or the presence of bound ligands, such as neurotransmitters. In contrast, passive channels do not change their permeability in response to changes in the membrane potential. Both passive channels and active channels in the open state exhibit selective permeability to different types of ions. For example, potassium channels primarily allow potassium ions to pass through.

3. Action Potential

Intracellular recordings demonstrate that action potentials are characterised by a sharp increase in the membrane potential (depolarisation of the membrane) followed by a somewhat less sharp decrease towards the resting potential (repolarisation). This

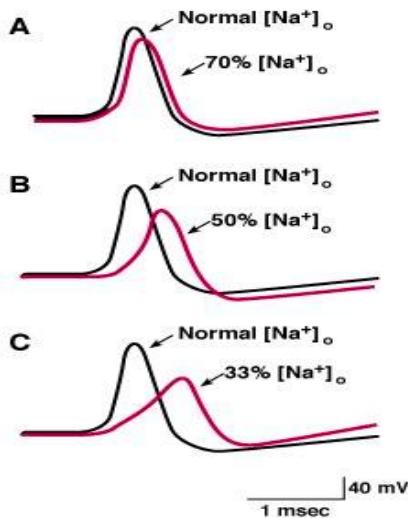


may be followed by an after hyperpolarization phase in which the membrane potential falls below the resting potential before recovering gradually to the resting potential.

4. Voltage-Dependent Conductances

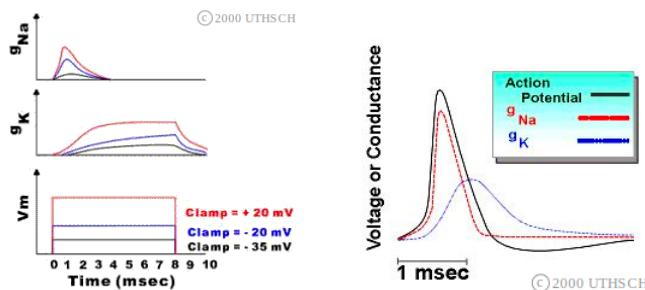
Na^+ is critical for the action potential in nerve cells. As shown in Figure [], As the concentration of sodium in the extracellular solution is reduced, the action potentials become smaller.

An important property of the voltage-dependent Na^+ channels. Note that the permeability increases rapidly and then, the



permeability decays back to its initial level. This phenomenon is called inactivation. The Na^+ channels begin to close, even in the continued presence of the depolarization. Inactivation contributes to the repolarization of the action potential. However, inactivation is not enough by itself to account fully for the repolarization.

In addition to voltage-dependent changes in Na^+ permeability, there are voltage-dependent changes in K^+ permeability. These K^+ channels are normally closed, but open in response to depolarization.



B. Computational Modeling

Computational modeling is the use of computers to simulate and study complex systems using mathematics, physics and computer science. A computational model contains numerous variables that characterize the system being studied. Simulation is done by adjusting the variables alone or in combination and observing the outcomes. Computer modeling allows scientists to conduct thousands of simulated experiments by computer. The thousands of computer experiments identify the handful of laboratory experiments that are most likely to solve the problem being studied.

1. Simulation

Simulation involves running the model with particular parameter settings to generate ‘fake’ behavioral data. These simulated data can then be analyzed in much the same way as one would analyze real data, to make precise, falsifiable predictions about qualitative

and quantitative patterns in the data. Simulation is a way to make theoretical predictions more precise and testable.

2. Parameter estimation

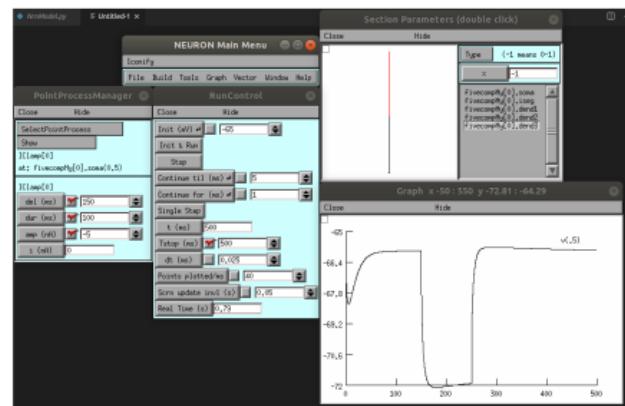
Parameter estimation involves finding the set of parameter values that best account for real behavioral data for a given model.

3. Model comparison

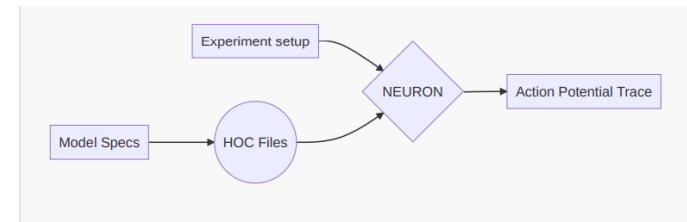
Model comparison involves trying to compute which of a set of possible models best describes the behavioral data, as a way to understand which mechanisms are more likely to underlie behavior. This is especially useful when the different models make similar qualitative predictions but differ quantitatively.

III. NEURON software

NEURON, developed at Duke University, is a simulation environment for modeling individual neurons and networks of neurons. NEURON environment is a self-contained environment allowing interface through its GUI, or via scripting with hoc or python programming language



We are 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.



By writing instructions in NEURON’s programming language ‘hoc’, we can specify a model that describes the desired model topology and different channels and their respective parameters as a simple Example :

```

/* model specification*/
/////////////// topology ///////////
create soma, apical, basilar, axon
connect apical(0), soma(1)
connect basilar(0), soma(0)
connect axon(0), soma(0)

////////// geometry ///////////
soma { L = 30 diam = 30 nseg = 1 }
axon { L = 1000 diam = 1 nseg = 37 }

////////// biophysics ///////////
forall { Ra = 100 cm = 1 }
soma { insert pas g_pas = 0.0002 e_pas = -65 }
axon { insert pas g_pas = 0.0002 e_pas = -65 }
...

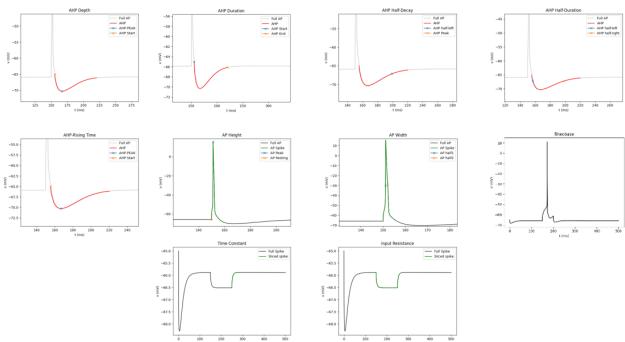
```

IV. METHODS

4.1. Feature Extraction

Instead of using the recorded trace of the model action potential in the optimization process, We are Extracting Characterizing Features such as:

- Time constant (tau)
- Internal resistance (resistance of the axoplasm) (Ri)
- Rheobase
- Action potential (AP) Height
- Action potential (AP) Width
- After-hyperpolarization (AHP) depth
- After-hyperpolarization (AHP) Duration
- After-hyperpolarization (AHP) Half-Duration
- After-hyperpolarization (AHP) Half-Decay
- After-hyperpolarization (AHP) Rising-Time



4.2. Genetic Algorithms

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitations of random search provided with historical data to direct the search into the region of better performance in solution space. They are

commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to the next generation. In simple words, they simulate “survival of the fittest” among individuals of consecutive generations for solving a problem. Each generation consists of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the chromosome.

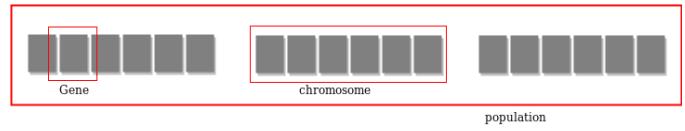
1. Foundation of Genetic Algorithms

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosomes of the population. Following is the foundation of GAs based on this analogy –

1. Individuals in the population compete for resources and mate.
2. Those individuals who are successful (fittest) then mate to create more offspring than others.
3. Genes from “fittest” parents propagate throughout the generation, that is sometimes parents create offspring which are better than either parent.
4. Thus each successive generation is more suited for their environment.

2.2 Search space

The population of individuals is maintained within search space. Each individual represents a solution in search space for a given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).



2.2 Fitness score

A Fitness Score is given to each individual which shows the ability of an individual to “compete”. The individual having optimal fitness score (or near optimal) are sought.

The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores. The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce better offspring by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals, eventually creating a new generation when all the mating opportunities of the old population are exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

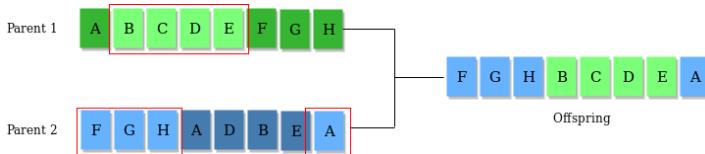
Each new generation has on average more “better genes” than the individual (solution) of previous generations. Thus each new generation has better “partial solutions” than previous generations. Once the offspring produced have no significant difference than offspring produced by previous populations, the population

converges. The algorithm is said to be converged to a set of solutions for the problem.

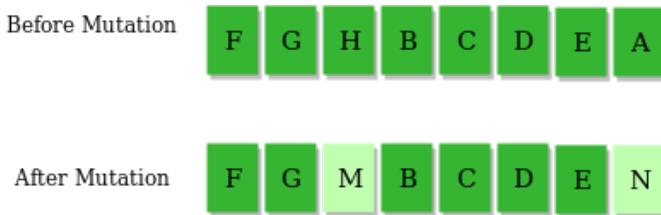
2.3 Operators of Genetic Algorithms

Once the initial generation is created, the algorithm evolves the generation using following operators -

- I. **Selection Operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to the successive generations.
- II. **Crossover Operator:** This represents mating between individuals. Two individuals are selected using a selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example:



- III. **Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in population to avoid premature convergence. For example –



3. Multi-objective Optimization

Multi-objective Optimization is an area of multiple criteria decision-making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics, where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing cost while maximizing comfort while buying a car, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle are examples of multi-objective optimization problems involving two and three objectives, respectively. In practical problems, there can be more than three objectives.

Our goal in this problem is to tune model parameters which are:

- gna_NafSmb1
- gkdr_KdrSmb1
- gkca_CaSmb1
- gcan_CaSmb1
- Ghbar_hb1
- gcal_Casmb1

So we can get the features of action potential (AP) of neuron model as neuroscientists require for example

- Time constant (τ)
- Internal resistance (resistance of the axoplasm) (R_i)

- Average internal resistance
- Action potential (AP) Height
- Action potential (AP) Width
- After-hyperpolarization (AHP) Depth
- After-hyperpolarization (AHP) Duration
- After-hyperpolarization (AHP) Half-Duration
- After-hyperpolarization (AHP) Half-Decay
- After-hyperpolarization (AHP) Rising-Time
- Rheobase

Using Non-dominated sorting genetic algorithm II (NSGA-II) we can find multiple solutions, but the goal of the algorithm is to find the pareto-optimal solutions for our defined target feature of the neuron model inside this big possible solution space.

3.1 Structure

As the name genetic algorithm already implies, the principle of Darwin survival of the fittest is used to find the fittest solution for this problem whereas the fitness value of an individual is calculated by taking the absolute difference between the calculated value for a feature and the mean of some experimental data for this feature divided by the standard deviation of that experimental data for the feature

$$F = \frac{abs(x_{model} - \mu_{exp})}{\sigma_{exp}}$$

We do that for all features to get the Pareto optimal.

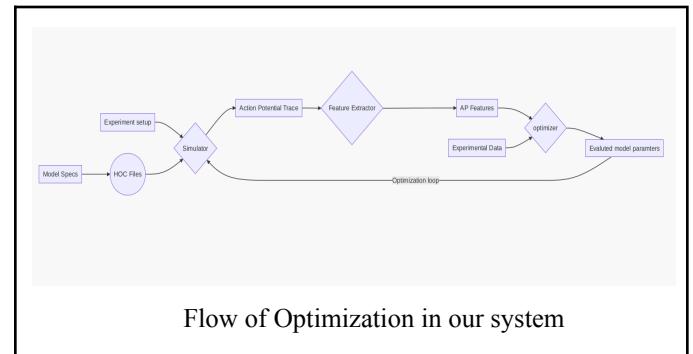
The genes in the algorithm indicate the parameters of the neuron model.

3.2 Main Loop

Firstly, there is a population containing a set number of individuals. Each individual differs in its genes and for each of them the fitness value is calculated and out of this existing population, an offspring population is created doing crossover of better-performing individuals and causing mutations.

This results in a new population which is twice the size of the original population, then every individual is ranked based on its fitness regarding the target indicators. The better its fitness, the better it is ranked. If some individuals have the same rank, then they are put together in a front.

So the new population is chosen by picking all individuals of the best fitness until the size of the original population is reached. While picking the fronts, if the size of the front to be picked is bigger than the remaining space in the resulting population then, an approach called crowding distance is used.



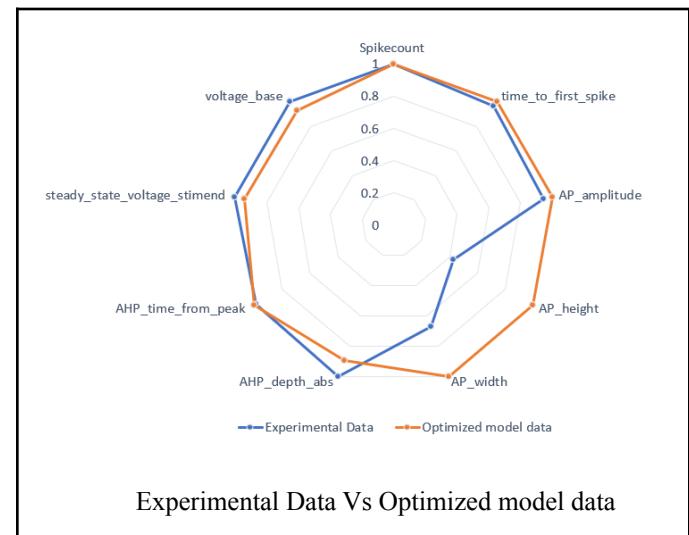
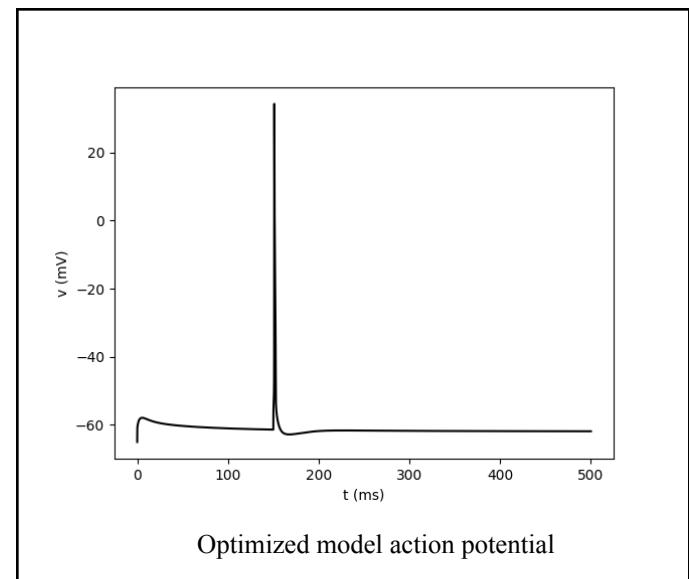
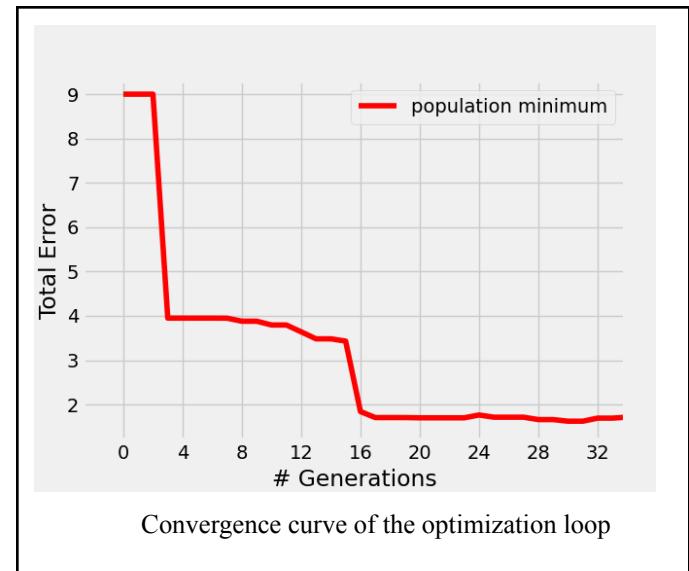
IV. RESULTS

1. Optimization

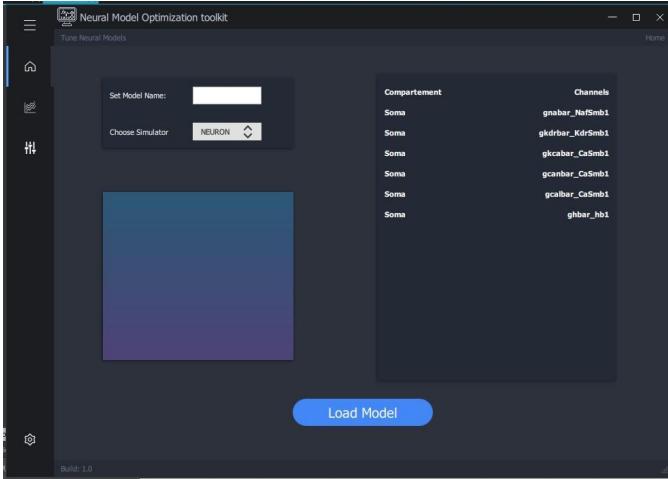
We evaluated our optimization approach on a 5-compartment model, with the aim to optimize 6 free maximal conductance parameters, table [1] shows each parameter name and its range

Parameter	Low(mho/c^2)	High(mho/c^2)
gna_NafSmb1	0	1
gkdr_KdrSmb1	0	1
gkca_CaSmb1	0	1
gcan_CaSmb1	0	1
ghbar_hb1	0	1
gcal_CaSmb1	0	1

Table [1]

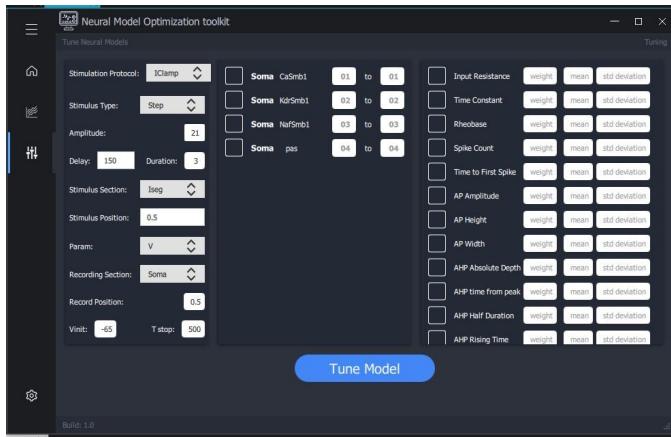


2. Tuning Toolkit UI



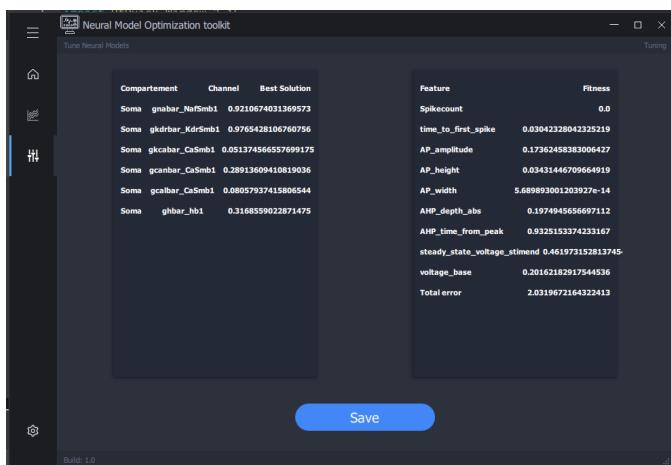
Home Page

- Set custom model name.
- Upload model.
- View model parameters and compartment



Tuning Page

- Set experiment configurations
- Choose parameters to tune.
 - Add boundaries.
- Choose desired features.
 - Add weight, mean and standard deviation.



Tuning Result Page

- View best solutions for the new tuned parameters.
- View the fitness (error) of each feature.
- Save the tuned model.

V. ACKNOWLEDGMENT

This work was funded by the Academy of Scientific Research and Technology.



أكاديمية البحث العلمي والتكنولوجيا
ACADEMY OF SCIENTIFIC RESEARCH
AND TECHNOLOGY

VI. References

- Carnevale, T., 2006. *Neuron book*. Cambridge: Cambridge Univ Press.
- De Rainville, F., Fortin, F., Gardner, M., Parizeau, M. and Gagné, C., 2014. DEAP. *ACM SIGEVOlution*, 6(2), pp.17-26.
- Druckmann, S., Berger, T., Schürmann, F., Hill, S., Markram, H. and Segev, I., 2011. Effective Stimuli for Constructing Reliable Neuron Models. *PLoS Computational Biology*, 7(8), p.e1002133.
- Kandel, E., Koester, J., Mack, S. and Siegelbaum, S., 2013. *Principles of neural science*. 5th ed.
- Prinz, A., 2007. Neuronal parameter optimization. *Scholarpedia*, 2(1), p.1903.
- Sterratt, D., 2011. *Principles of computational modelling in neuroscience*. Cambridge: Cambridge University Press.
- Van Geit, W., Achard, P. and De Schutter, E., 2007. Neurofitter: a parameter tuning package for a wide range of electrophysiological neuron models. *BMC Neuroscience*, 8(Suppl 2), p.P5.
- Van Geit, W., De Schutter, E. and Achard, P., 2008. Automated neuron model optimization techs.niques: a review. *Biological Cybernetics*, 99(4-5), pp.241-251.
- K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002. Available: 10.1109/4235.996017.