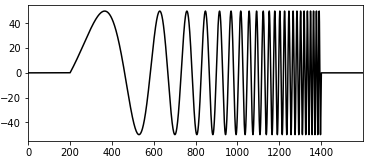
7001 Machine Learning Project for graduate students

**Background:**

Electrophysiology is the study of the electrical properties of biological cells which is widely practiced in neuroscience. When we are looking for building biophysical realistic neuron models, we have to first understand the properties of all kinds of channels on neuron membrane. One of the most important parameters for modeling a biophysical neuron model is the conductance of the ion channels. It is usually a difficult task to tune the parameters and match the activities of a neuron in real electrical recordings. Machine learning has potential in solving this problem.

**Problem:**

Let's say we have an electrical recording from a real neuron. We want to model this neuron using a typical Hodgkin-Huxley model - only 3 channels, leak, na, k. Assume that we already know the equations and all parameters but only the conductance of the 3 channels remain to be fitted. We have a protocol in the electrical recording, that is to inject a designed trace of current into a cell and record its membrane voltage response. One commonly used type of current injection is called ZAP current, which is a sinusoidally oscillating waveform, with instantaneous frequency changing in time (i.e. chirp or ZAP waveform). The advantage of ZAP current is that it can trigger response in a wide range of frequencies.



Now we know the current injection and the response of a real neuron. And we already have a model with the conductance of the 3 channels to be tuned. The task is to predict the 3 parameters given the ZAP current triggered response from a real recording.

With the help of the NEURON software and the artificial neural network, we can train a machine learning model to predict the parameters. The idea is to randomly select the 3 parameters in a given range and run lots of simulations with ZAP current injection using the random parameters in NEURON. Then we generate a large dataset which we can use to look for the mapping between the 3 parameters and the membrane voltage response. We can then train a artificial neural network model to predict.

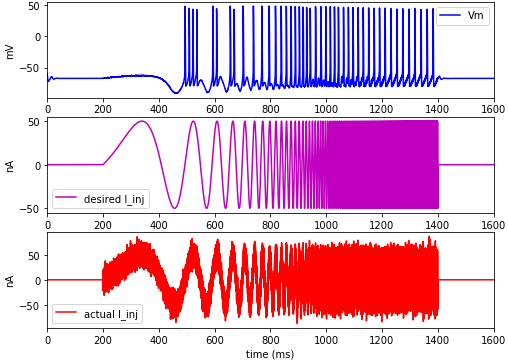
**Example codes:**

The example codes are written in python with comments that explains how it works.

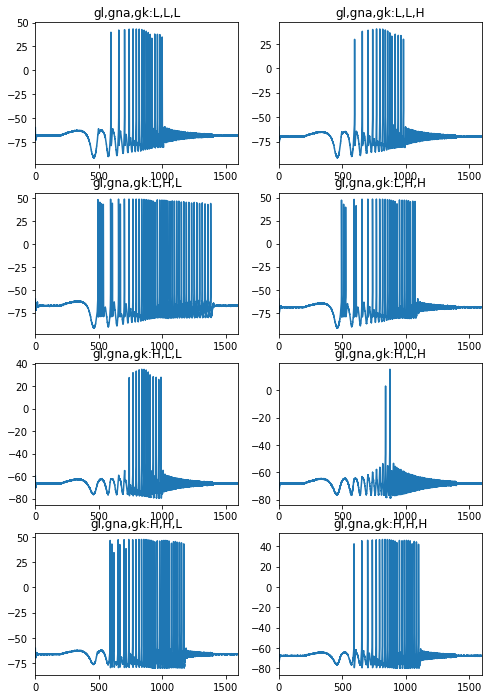
**celltemplate.py** is a module that defines the neuron model class and some methods for changing the conductance and setting the current injection. Try to understand the code with the comments and with the help of NEURON documentation.

**input\_currents.py** is a module that defines several types of current stimulation. While the ZAP current is probably the only one that will be used. But there is no harm to check other types.

**test\_injection\_response.py** is an example code that shows how to setup the model and ZAP current injection, then to display the currents injection and result membrane voltage response.



**test\_gbar\_range.py** is an example code that uses different conductance for each change and displays the resulting response for each case. The conductance value range of each channel is given here. This code runs 8 different simulations where the conductance is chosen to be either the lower bound or the upper bound of the range for each channel.



**Your Tasks:**

Your task is to develop codes that do 2 things as described below.

1. Generate random conductance for the 3 channels and run simulation for each case. Then create a dataset with the membrane voltage response and the corresponding 3 parameters.

2. Build an artificial neural network model that learns the mapping from membrane voltage response to the 3 parameters. Then test the performance using the dataset you generate. We do not have real recording to test, but we can assume that a subset of the generated data is the ground truth that can be used to test the performance.

**Hints for task 1:**

* You can use Jupyter notebook to write and run your code. For example, just copy and paste the codes from the examples to a notebook file. There is no need to change codes in **celltemplate.py** and **input\_currents.py** since they are modules to be imported to your main program.
* Use numpy Random module to generate the conductance. Uniform distribution is recommended. Generate values within the range given in **test\_gbar\_range.py**. For all other settings, also use the same as given in that example, such as model parameters, simulation parameters, current injection protocol, etc.
* The membrane voltage response during only the current stimulation duration (i.e. 200-1400 ms) is suggested to keep. But you are definitely allowed to try including the duration outside. The simulation time step size is 0.025 ms. The response data does not have to be the same resolution. Since a typical spike spread ~3 ms, using 1 ms sampling rate should be enough which will keep your dataset size small.
* You can save the data into txt file using numpy.savetxt.

**Hints for task 2:**

* You can first try a simple linear regression model on this dataset. Then compare its performance with the neural network model later.
* You are free to choose the size of your dataset, design your own network architecture, training parameters ,etc.
* You need to split the training and testing data. An appropriate metric to judge the performance should be used, e.g. mean squared error, mean absolute percentage error ,etc.
* You can test the effect of standardizing output features (check out sklearn.preprocessing.StandardScaler).
* When generating the data, we introduced some noise in the current injection which is closer to reality than perfectly clean data. You can try to generate another dataset where all the input is clean without noise. See how that will affect the performance of the prediction.
* Feel free to explore all aspects of the artificial neural network that you are interested in in this study.