Introduction to Neural and Cognitive Modeling

Final Project Report

Ashutosh Rudrabhatla (2022111036)

December 2, 2024

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

Project Title: Replicating and Extending the Application of Izhikevich Neuron Model in Pattern Recognition

1 Aim of the Project

- To understand the various aspects of the Izhikevich Neuron Model.
- To prepare classification datasets as current input signals for classification using the neuron model.
- To compare the performance of a single-IZ-neuron classification model with traditional and SOTA classification models.

2 Previous Work

- **Roberto Vazquez (2012):** Application of Izhikevich Neuron Model in classification and comparison with traditional methods. (link)
- **Eugene M. Izhikevich (2003):** Original paper introducing the neuron model, explaining physical significance and simulation procedures. (link)

3 Datasets Used

- **Iris Dataset:** 150 datapoints, 3 types of flowers, features include Sepal length/width and Petal length/width.
- Used because it is relatively small in size and is a standard dataset used to train and test any classification model.
- **Penguins Dataset:** 344 datapoints, 3 penguin species, features include bill length/depth, flipper length, weight, and sex.
- Used due to its relatively small size, along with its additional complexity (such as NaN values in some cells).

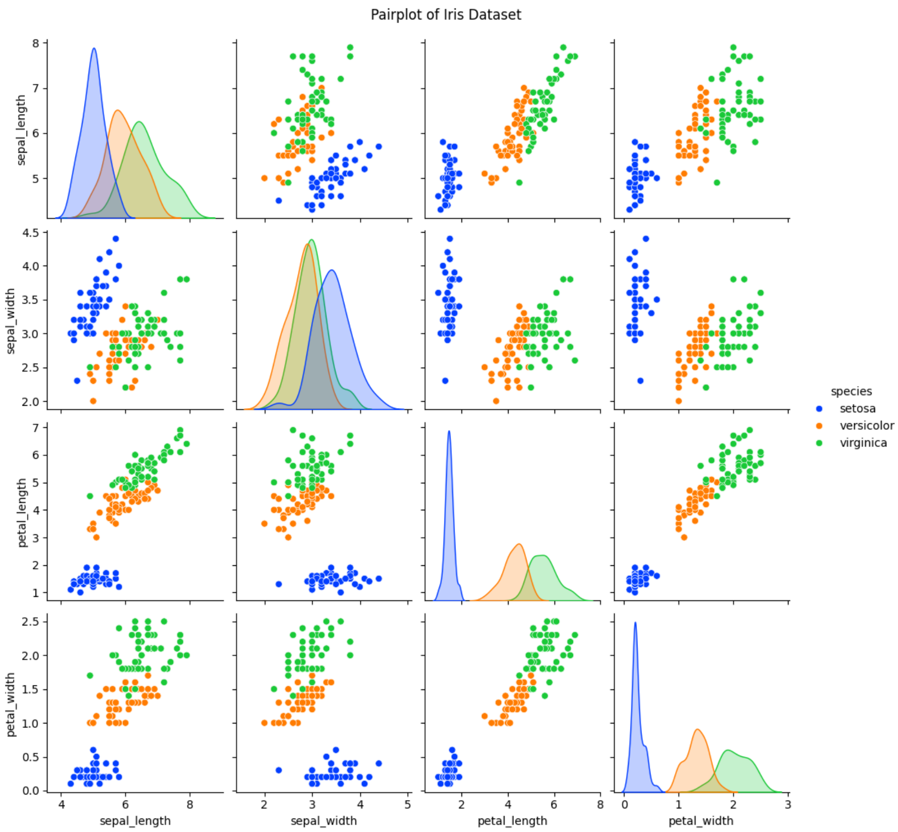

Figure 1: **A visualization of the Iris dataset**

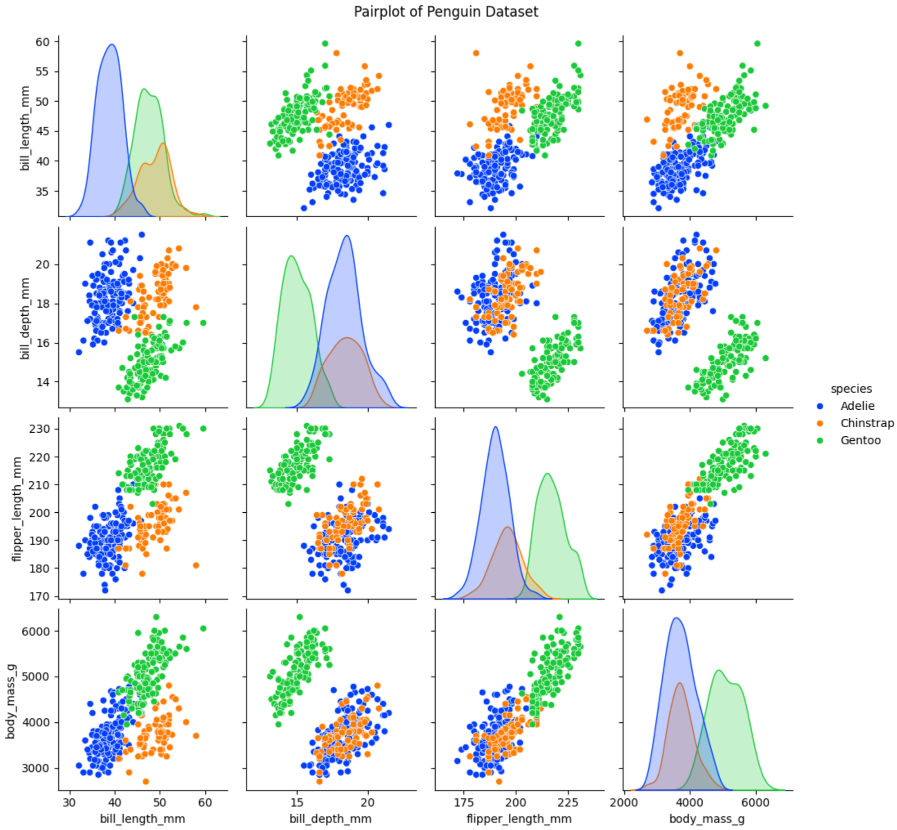

Figure 2: **A Visualization of the Penguins dataset**

4 Implementation

- **Simulation of the neuron model:** A template was referenced from here. I decided not to use inbuilt libraries since I felt that implementation would help provide a deeper understanding of the neuronal mechanics, and give me more control over the mechanisms in the subsequent steps of the implementation.
- **Data preprocessing:** The preprocessing involved excluding the text-based columns, and replacing NaN values with column means.
- **Conversion into current input signals:** Was done by splitting a current input (120 ms step current) into 4 parts, and proportionally ascribing a current value according to the values in each of the 4 columns for each row of the table. (see Fig. 5 and Fig. 6 for some samplme current inputs obtained by doing this).
- **Training and testing the model**: Activation function was **spike frequency**. Parameters were optimized using **differential evolution**.
- **Comparison of accuracy with traditional classification methods**

4.1 Conversion into Current Input

4.1.1 Preprocessing:

- Categorical variables encoded.
- Missing/NaN values replaced with column means.
- Numeric features normalized.
- Columns with non-float entries were excluded.

4.1.2 Conversion:

- portions of the step current were ascribed a proportional current value according to value of the normalized feature it represented.

- This was done for every row of the dataset, giving an equivalent data set with current signal rows that the neuron classification model can be trained on.

4.2 Training and Testing the Model

4.2.1 Training

- Optimizing parameters a, b, c, d with bounds:
  a: (0.001, 0.2), *b:* (0.1, 1), *c:* (-80, -30), *d:* (0, 10).
- Optimization using differential evolution.
- Classification feature: Spiking rate of neurons.

4.2.2 Testing

- Train-test split: 80:20.
- Five-fold cross-validation for accuracy estimation.
- Reporting optimal parameters and accuracies.

5 Results and Discussion

5.1 Neuronal Activity for a constant step current for optimized neurons

5.1.1 Iris Dataset:

Parameters for Optimized Iris Neuron:

- **a = 0.089, b = 0.124, c = -79.540, d = 7.225**

Corresponding Observed Characteristics:

- **Regular Spiking pattern** observed for constant step current input.
- **High sensitivity to input current:** Due to the combination of parameters, the neuron will exhibit a variety of spiking patterns (specifically in terms of spiking rates) for different input currents. This is to be expected, as this high sensitivity is pivotal to the neuron being able to make proper predictions for classification.

5.1.2 Penguins Dataset:

Parameters for Optimized Penguins Neuron:

- **a = 0.002, b = 0.158, c = -40.909, d = 9.927**

Corresponding Observed Characteristics:

- **Chattering or intrinsic bursting behavior:** observed due to relatively high resting potential (c value).
- **Prolonged refractory period:** this is expected due to the very low value of 'a', which delays the decay of recovery variable 'u', thereby increasing the refractory period after a set of bursts.
- **High sensitivity to input current:** Again, due to the combination of parameters, the neuron will exhibit a variety of spiking patterns (specifically in terms of spiking rates) for different input currents. This is to be expected, as this high sensitivity is pivotal to the neuron being able to make proper predictions for classification.

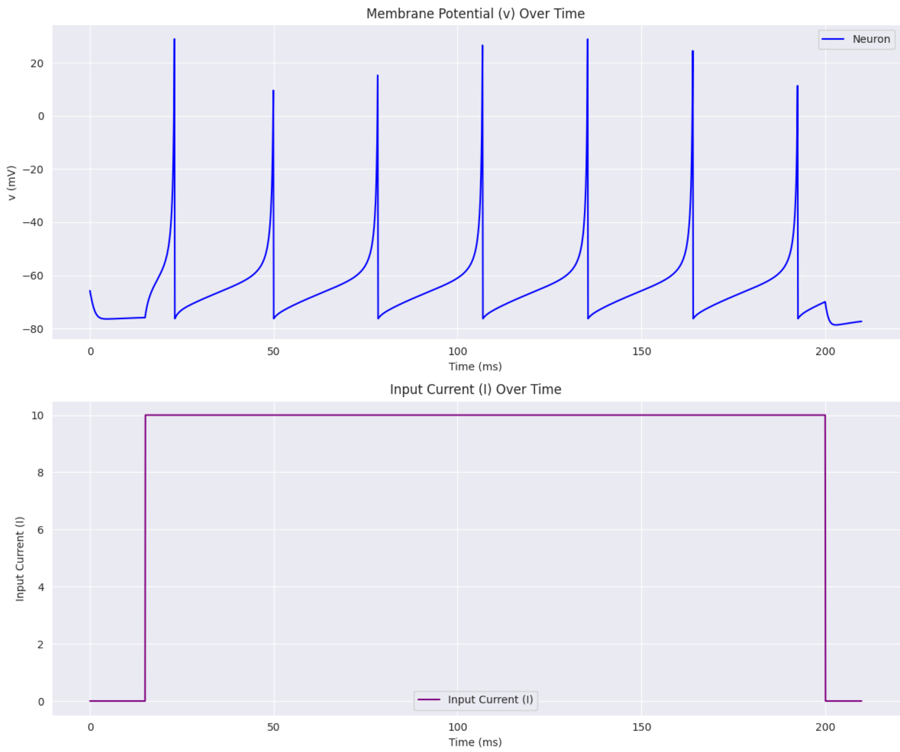

Figure 3: **Neuronal Activity of Optimized Iris neuron**: Regular spiking pattern observed

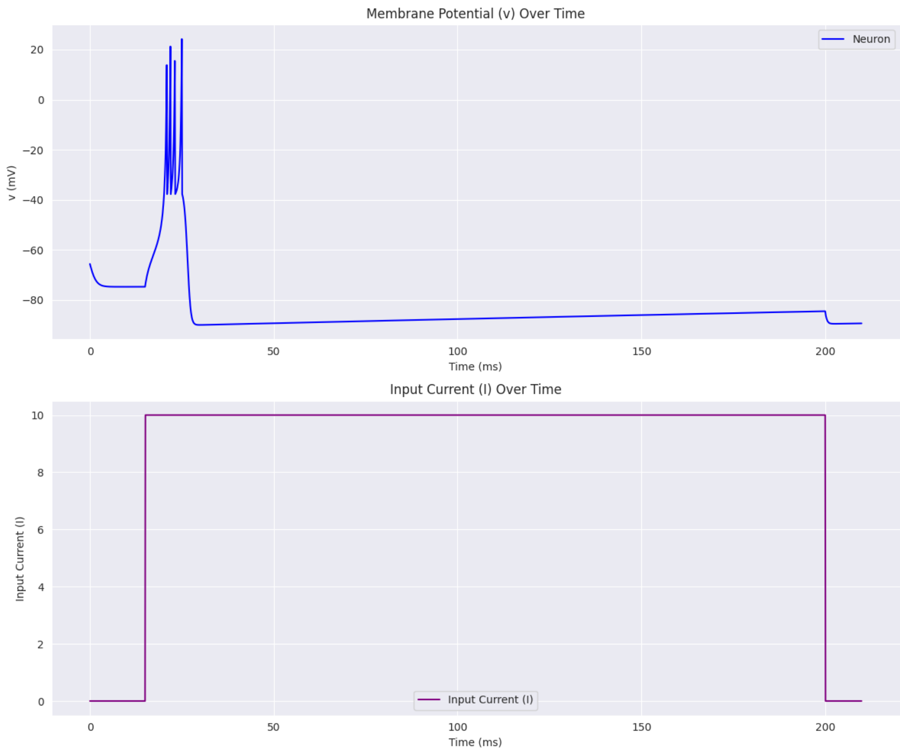

Figure 4: **Neuronal Activity of Optimized Penguins neuron**: Intrinsic bursting/chatt followed by a long refractory period observed

5.2 Neuron spiking patterns for each class in the Iris and Penguins datasets:

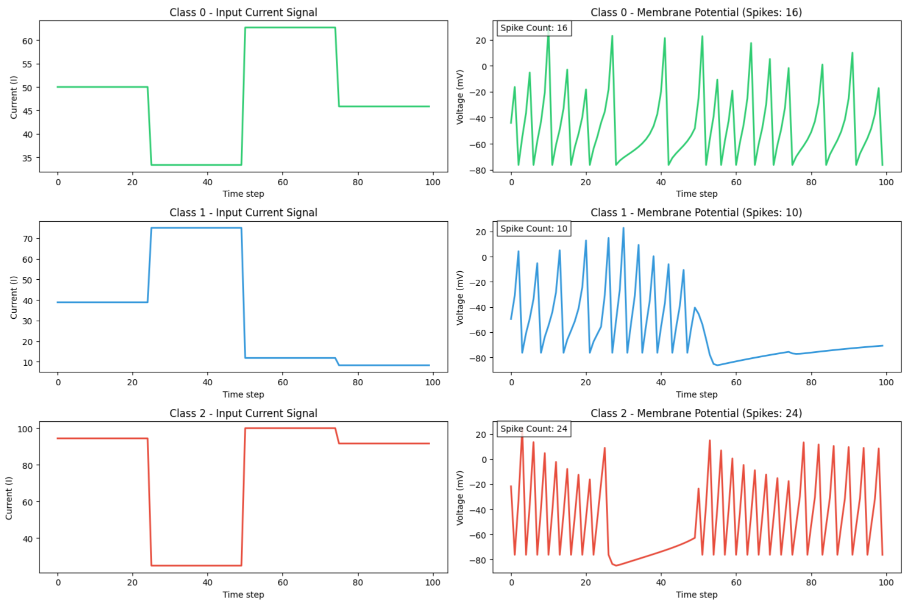

Figure 5: **Visualization of classification of the Iris dataset** - Left: Current input signals generated for different rows of data. Right: Corresponding spiking patterns for optimized neuron

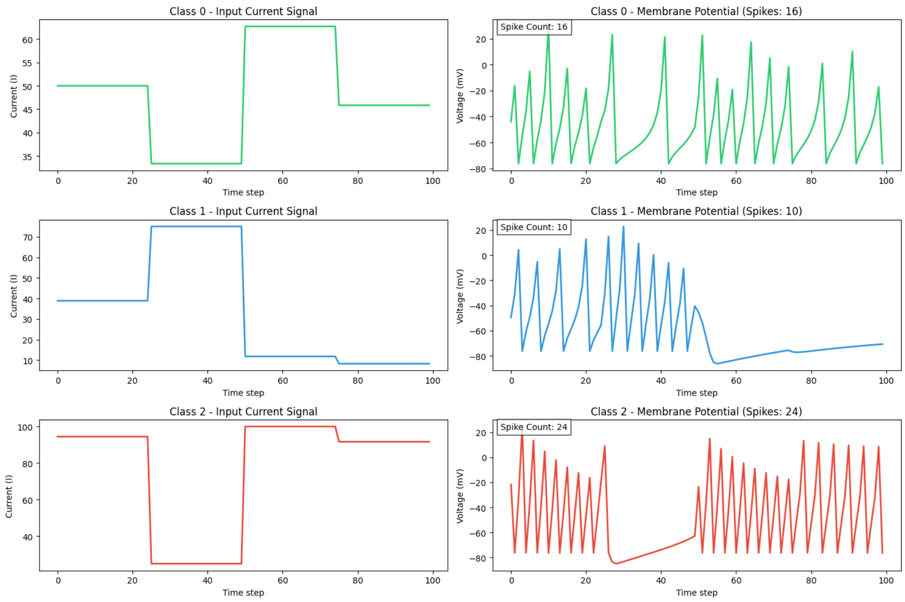

Figure 6: **Visualization of classification of the Penguins dataset** - Left: Current input signals generated for different rows of data. Right: Corresponding spiking patterns for optimized neuron

5.3 Accuracy Comparison:

Dataset	Model	Accuracy
Iris	Logistic Regression, k-NN, Decision Tree, Random Forest	1.00
	SVM	0.97
	Izhikevich Neuron Model	0.90
Penguins	Logistic Regression	0.97
	k-NN	0.68
	SVM	0.99
	Decision Tree	0.97
	Random Forest	0.96
	Izhikevich Neuron Model	0.80

6 Some Challenges Faced

- **Optimization of parameters and bounds:** The process of improving accuracy was a long and iterative one. Editing different parameters/seeds for training and testing, along with keeping appropriate bounds on a, b, c and d such that the biological integrity of the optimized neuron is not lost was a tricky task.
- **Poorly explained original study:** There was no code provided and the exact methodology such as conversion of dataset into current input signals, how the neuron models were trained etc was not explained. This meant that there had to be a lot of trial and error before a reliable strategy for training and testing the IZ neuron model was found.

7 Conclusion and Future Work

In Summary:

- The single-IZ-neuron classification model does not perform on par with the SOTA classification models. (with the exception of KNN in certain scenarios)
- The optimized IZ-neuron for the Iris dataset has parameters corresponding to **regular spiking** for constant step current input.
- The optimized IZ-neuron for the Iris dataset has parameters corresponding to **intrinsic bursting/chattering** for constant step current input.
- The biological viability of both neurons is questionable due to their high sensitivity to current input.

Some future works to expand upon what is done here:

- Exploration of features beyond firing rate for classification.
- Experiment with alternative optimization techniques.
- Utilize multi-neuron models, e.g., Izhikevich neuron as activation in neural networks.
- Apply single-neuron models to other machine learning problems.

8 Citations and Resources:

1. **Roberto Vazquez (2012):** Application of Izhikevich Neuron Model in classification and comparison with traditional methods. (link)
2. **Eugene M. Izhikevich (2003):** Original paper introducing the neuron model, explaining physical significance and simulation procedures. (link)
3. Iris Flower Dataset: (link)
4. Penguins Dataset: (link)
5. Implementation of the Izhikevich neuron model in python: (link)
6. Github link with the code: (link)