Take Home Assignment 1
Member #1: Arya Bhavesh Shah
Roll No: 25290002
arya.shah@iitgn.ac.in

Assignment Details:
Recent innovations in microscopy and computational imaging has given an unprecedented view into the neuronal structures. The MicRons project (video: Explore the connections of the brain - MICrONS neuroscience data insights) has given high resolution maps of various neurons and how they connect to each other. Continuing on realistic models of perceptron, please go through the website: https://www.microns-explorer.org/ and it's online software: https://www.microns-explorer.org/ngl-instructions.
Select at least two realistic neurons (you can take as many as you like) and model them as a perceptron.
You can feed inputs to the various dendrites through the synapses and then look at the output response on postsynaptic neuron(s).
Feel free to use simplistic or realistic models to perform these computations. There are no correct ways of doing it, just think deeply about how realistic neurons can act as perceptrons.
(Optional) Feel free to input any kind of data - image, text, audio and look at the response of the neuron.
The take home assignment has a hard deadline of August 20th, 2 PM. So please submit it before then.

Introduction

In the following exploratory assignment, I explored the MICrONS neuroscience dataset. In particular, I explored the Minnie65 Public Dataset which is available on the MICrONS Explorer. The dataset represents around 65% of a cubic millimeter-scale Electron Microscopy volume of a mouse visual cortex. In this assignment, I firstly explore the MICrONS Platform, the Minnie65 Public Dataset and then make use of their API offerings to extract two neurons and model them as

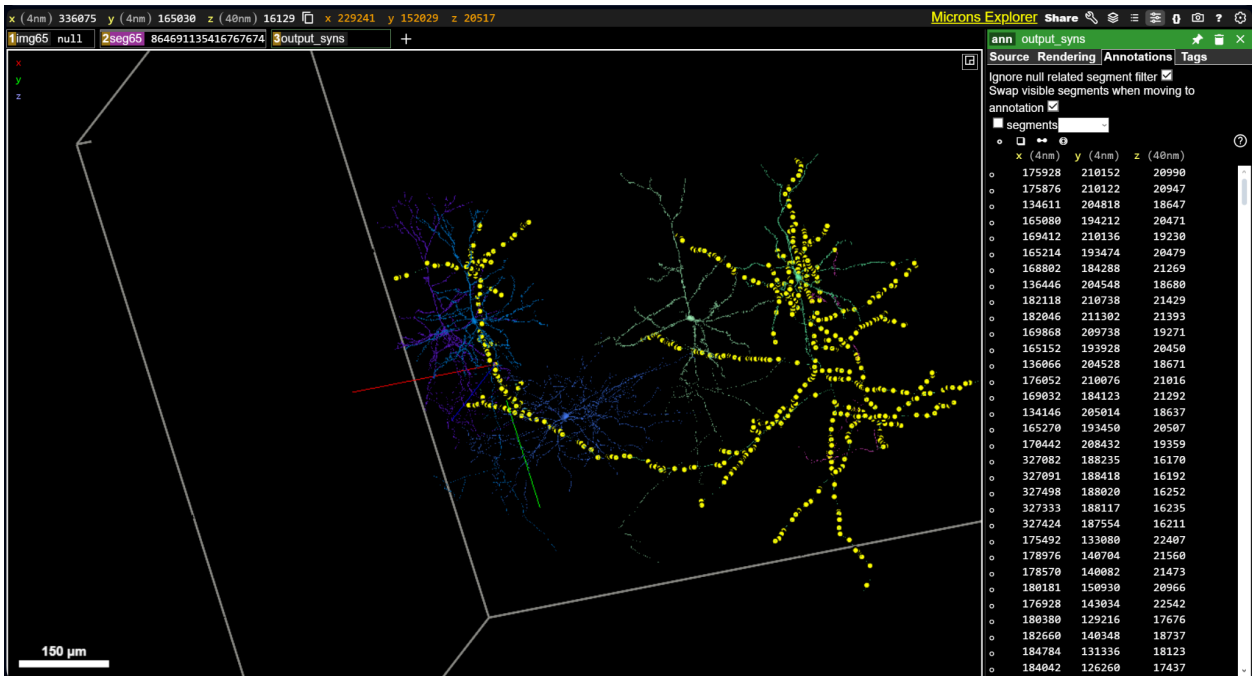
a perceptron. I also try to fit the entire MNIST Dataset in these biological neurons and report results and potential insights.

Exploring MICrONS

The starting point for exploration was the official homepage of MICrONS Explorer, from which I navigated my way to their hosted online visualizer: Neuromancer. In this visualizer I find the Minnie65 Dataset with the following specifications:

1. The platform provides Multi-resolution electron microscopy images at varied resolutions ranging from 8x8x40nm.
2. It provides voxel-based cellular segmentation with unique IDs per object that include meshes at various downsampling levels
3. These datasets are in the size range of Terabytes, the EM Imagery being around 117TB and the voxel data being 42TB
4. They also provide Synapse and Nucleus Data which includes Post Synaptic Density segmentations for synapses and nucleus segmentation with meshes along with metadata.
5. This offering by MICrONS allows querying synaptic connectivity through tools such as CAVEclient which I am using in the later part of the assignment experiments.

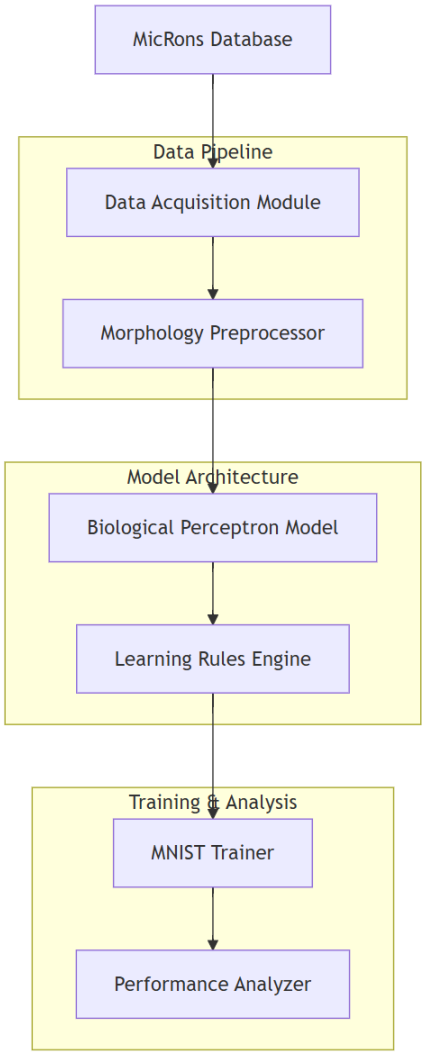
Here's how the online visualizer looks like:



Experiment

Based on the original task of this assignment, I make use of the biologically accurate neurons and model them as a perceptron using the authentic MICrONS neuronal morphology data for the task of MNIST classification. The aim is to demonstrate how we can integrate real neural anatomy with biological learning rules such as Hebbian or STDP which might provide us with insights into the computational constraints of biological neural networks.

We will use two real neurons from the MICrONS neural morphology dataset, implement dendritic processing based on the actual synaptic locations and their segment properties, then make use of learning rules like Hebbian and STDP and derive results from these neurons trained on MNSIT Dataset. Here is a high level overview of the flow of the experiment:



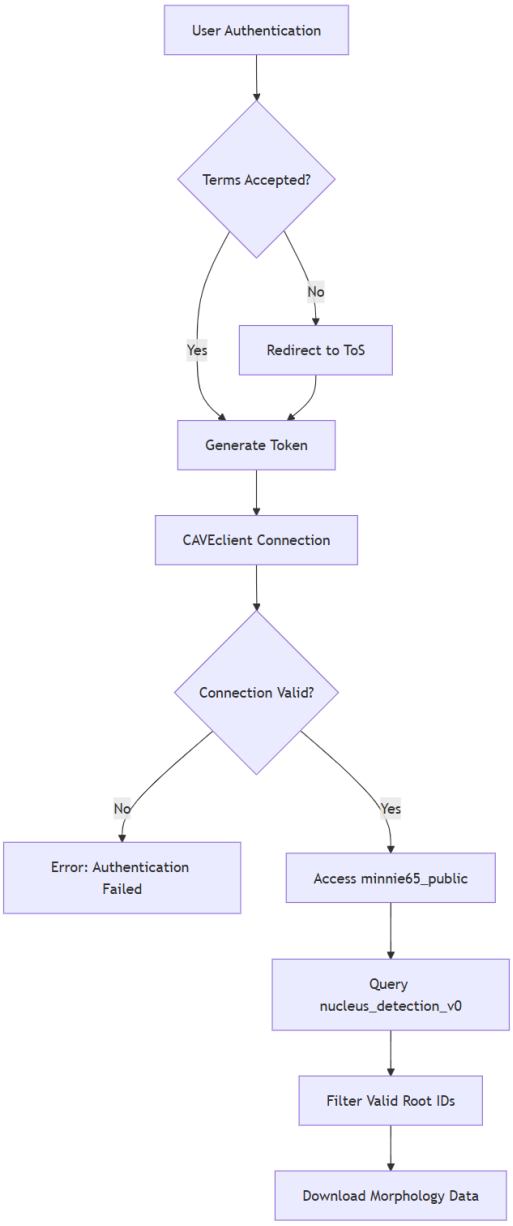
Thus our primary research question has been clear: Can real neuronal morphology data such as the one made available through MICrONS be used to create functional perceptron Models?

Data Acquisition

In order to access the data programmatically so that we can use it with python based libraries and existing MNIST Dataset I had to learn how to use their CAVEclient API which involves being registered, accepting their terms and conditions and authenticate with them in order to access our minnie65_public dataset from where we can query and extract two neurons for our study and experimentation. It was observed that the dataset had quite a few neurons and most likely not all might have been useful for our study thus a very basic neuron selection criteria was established as follows:

1. Inclusion Criteria:
    a. The root ID should be valid
    b. The skeleton Data should be present
    c. There should be a minimum number of dendritic segments available
    d. The synaptic connectivity data should be available
2. Exclusion Criteria:
    a. If any root ID is invalid or 0
    b. The skeleton information is missing
    c. The is insufficient morphological complexity

Here's how the data acquisition pipeline looks like:

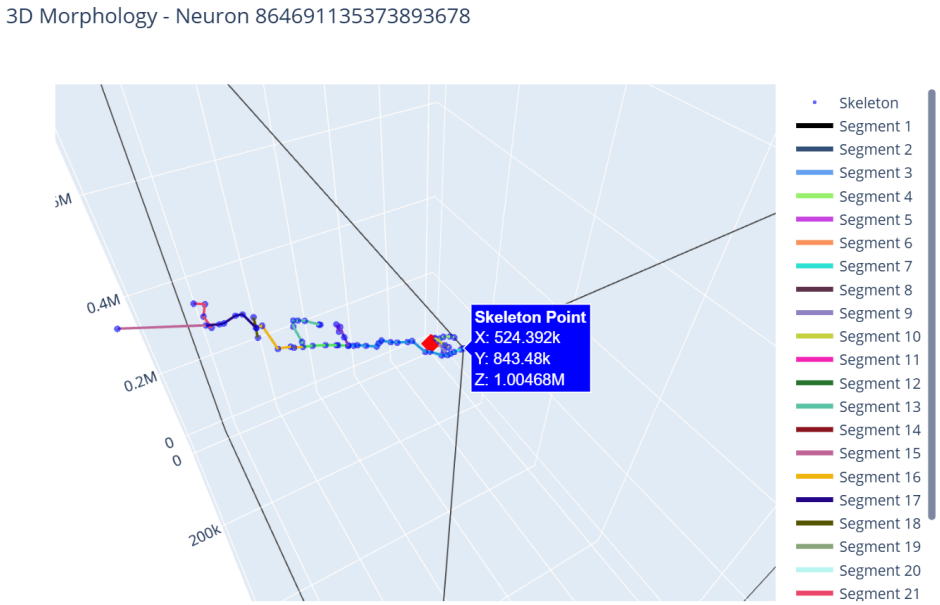


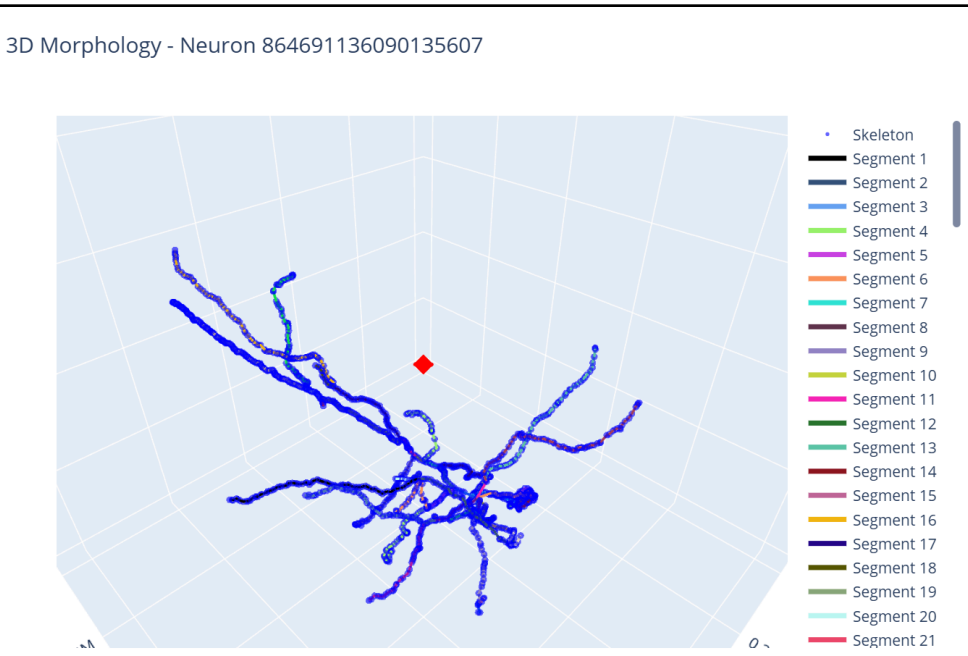
As mentioned previously, through this pipeline we extract two neurons from the minnie65_public dataset. Once the two neurons are extracted, morphological processing needs to be done as follows:

1. Extracting dendritic segments from the skeleton data
2. Map the synaptic locations to segments
3. Calculate the segment properties such as length, diameter and connectivity
4. Create the input mapping based on synaptic density

The two neurons that were selected are shown below:

3D Morphology of the Two Neurons Selected





Neuron 1 (ID: 864691136090135607):

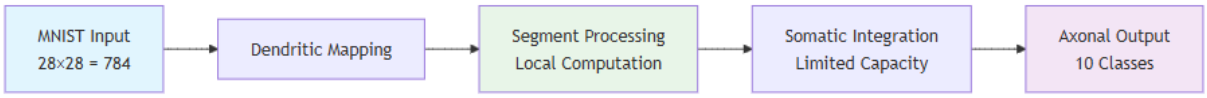
- Dendritic segments: 108
- Presynaptic connections: 40
- Postsynaptic connections: 1,994
- Morphological complexity: High

Neuron 2 (ID: 864691135373893678)

- Dendritic segments: 24
- Presynaptic connections: 0
- Postsynaptic connections: 2
- Morphological complexity: Low

Model Construction and Architecture

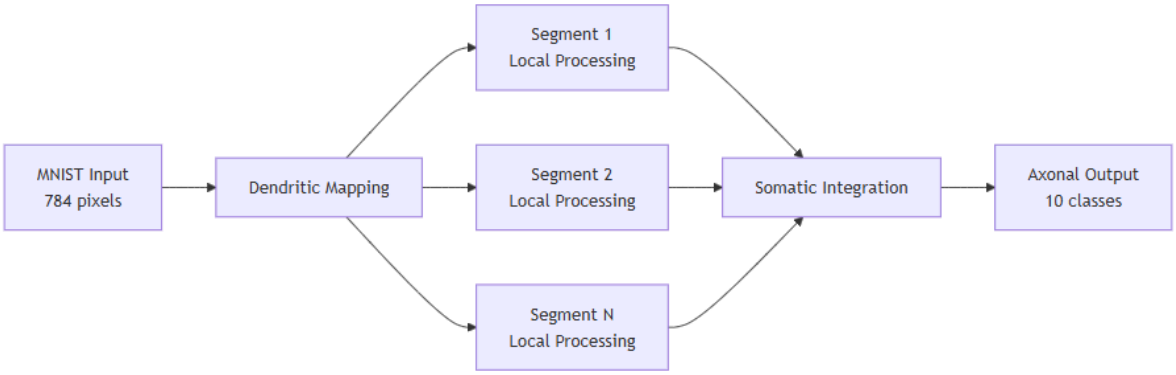
Since we have the two real neurons and we have been able to perform the morphological processing, the next step is to set up this as a biological perceptron. On a very high level this is how the architecture is setup:



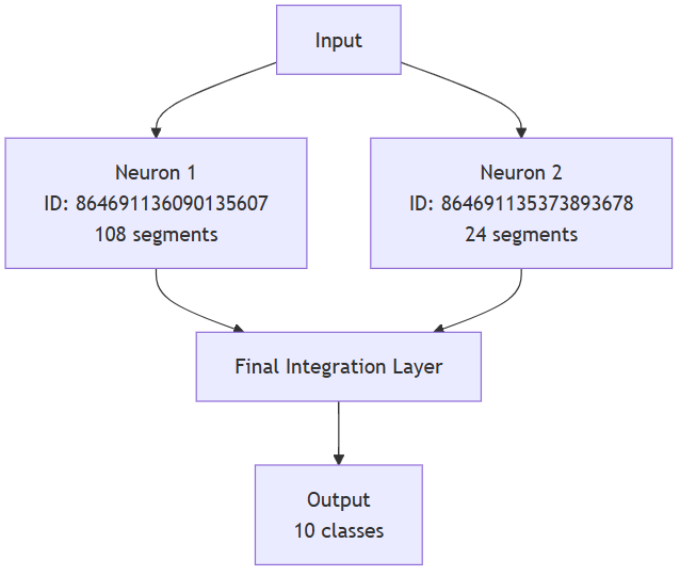
The MNIST dataset is one of the standard benchmark datasets since early Computer Vision, It consists of 60,000 Training Images and 10,000 Testing Images of English Numerical Digits from 0 to 9, thus 10 classes. We tackle this integration of fitting our data and training on this biological neuron by setting the following design principles:

1. We map the MNIST pixels to the dendritic segments based on the real synaptic density
2. Each segment performs limited computation, which sort of replicates the biological constraints
3. We maintain around 70% sparsity to mimic real neuron connections
4. We also make sure to integrate somatic layer that combines the dendritic outputs

This is how the flow can be visualized in much better detail:



Since we have two neurons, we have to include both of them in this architecture, thus the discussion done above is adapted to a multineuron architecture as depicted:



Biological Constraints applied include the following, some of which have been mentioned previously:

1. 70% Sparsity in connections
2. Local processing capacity is limited
3. Realistic weights are applied
4. Initialization is done with Xavier normal distribution

Training Pipeline and Protocols

As discussed in the previous section, now that we have the two neurons ready and the architecture established, we start training the neurons with the MNIST dataset for the task of classification. We make use of two learning rules that have been tested on these real neuron morphologies:

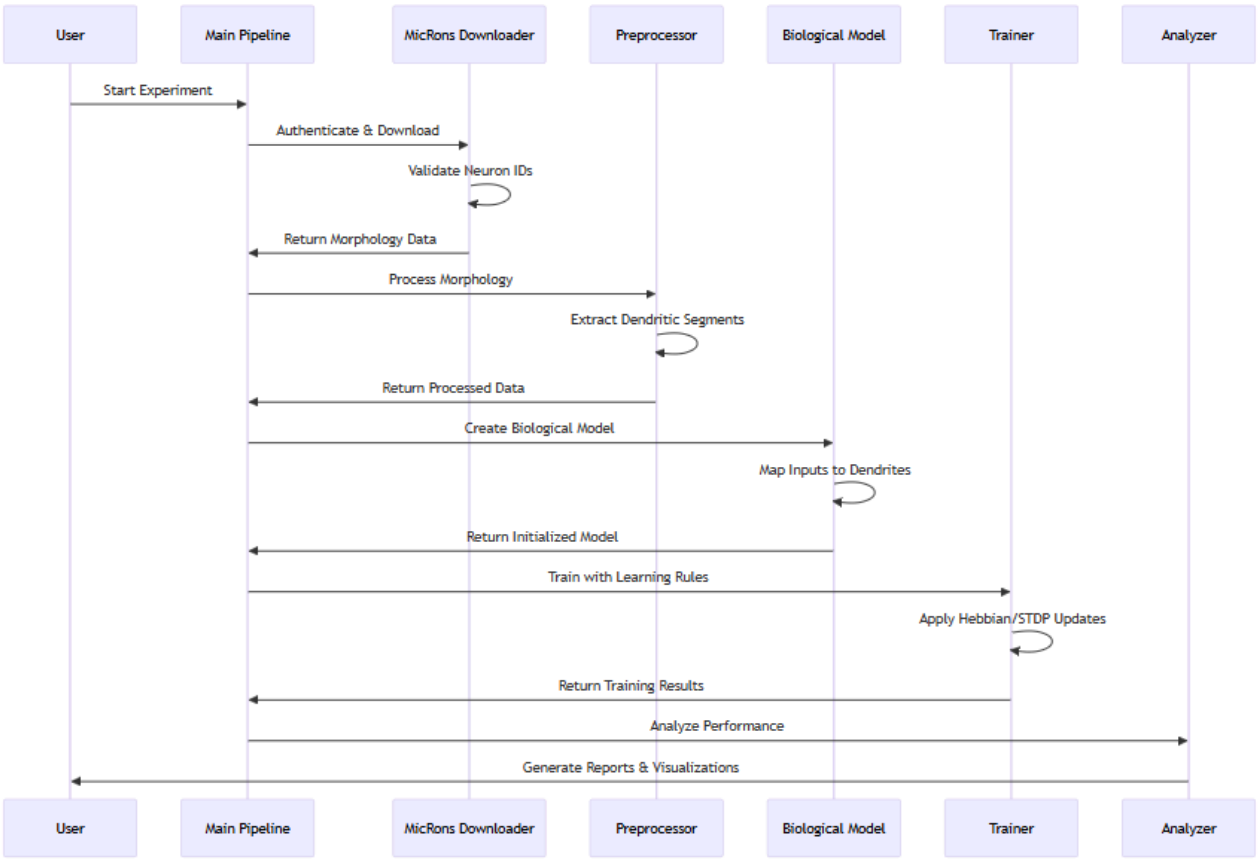
1. Hebbian Learning: Neurons that fire together, wire together; Correlation based weight updates
2. STDP Learning: Spike-timing dependent plasticity

The training configurations used in our experiment have been listed below:

Batch Size	32
Epochs	5
Learning Rates	Hebbian: 0.001
	STDP: 0.001

Weight Decay	0.01

Before we move on to the results, here is the complete data flow diagram of how we conduct the complete experiment:



Results

Before we show the training results, it is important to know the neuron utilization for each of the two neurons we used in our study:

Neuron 1 (108 Segments)	Neuron 2 (24 Segments)
• Input distribution: 784 pixels → 108 segments	• Input distribution: 784 pixels → 24 segments

• Average inputs per segment: approx. 7.3 pixels • Segment activation patterns: Uniform distribution • Synaptic density impact: High connectivity, limited local processing	• Average inputs per segment: approx. 32.7 pixels • Segment activation patterns: Sparse activation • Synaptic density impact: Low connectivity, concentrated processing

Now coming to the results of the training of MNIST Classifier on the two neurons, we train our newly created biological perceptron in two experiments:

1. Applying Biological Constraints as discussed previously
2. No Biological Constraints

No Biological Constraints Results

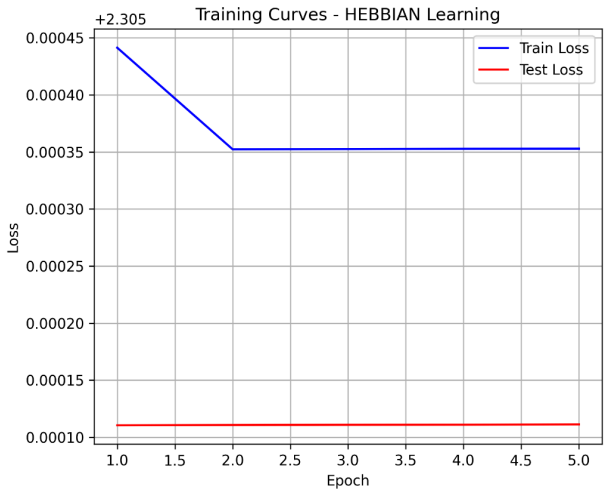
In the "No Biological Constraints", we basically disable all the biological constraints that have been discussed in the previous sections;
The results is further shown below:

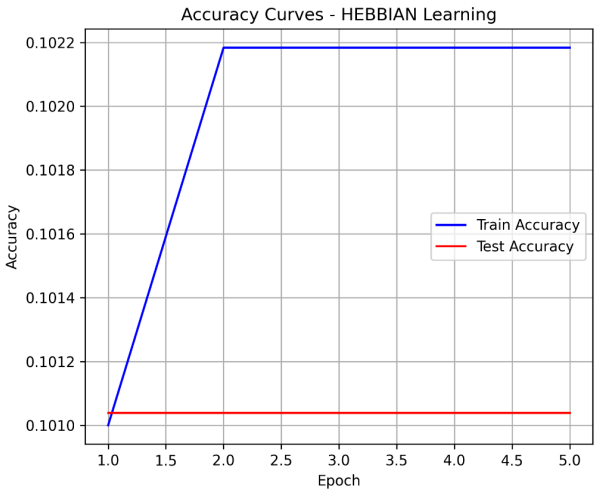
<table>
<tr><td colspan="2">

```
EXPERIMENT SUMMARY:
INFO:__main__:----------------------------------------
INFO:__main__:Best Model: BiologicalPerceptron_stdp
INFO:__main__:Best Accuracy: 0.1027
INFO:__main__:Learning Rule: stdp
INFO:__main__:
Results saved to: experiment_results_nobio
```

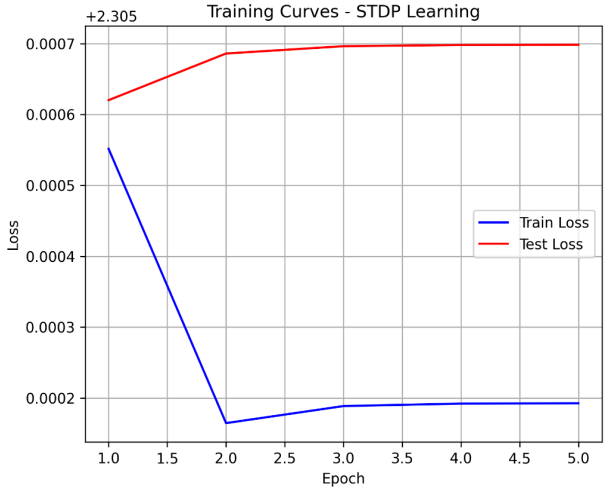
</td></tr>
<tr><td>Model Performance Rankings</td><td>1. BiologicalPerceptron_stdp: 0.1027
2. BiologicalPerceptron_hebbian: 0.1010</td></tr>
<tr><td>Learning Rule Analysis</td><td>HEBBIAN Learning:
- Mean Accuracy: 0.1010 ± 0.0000

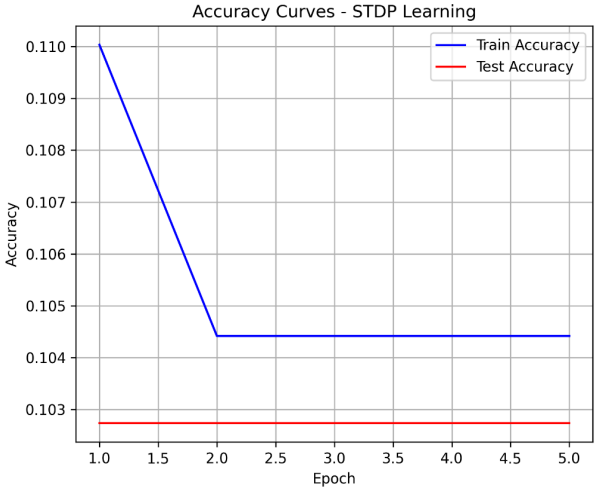
STDP Learning:
- Mean Accuracy: 0.1027 ± 0.0000</td></tr>
<tr><td colspan="2">Hebbian Training Curves</td></tr>
</table>



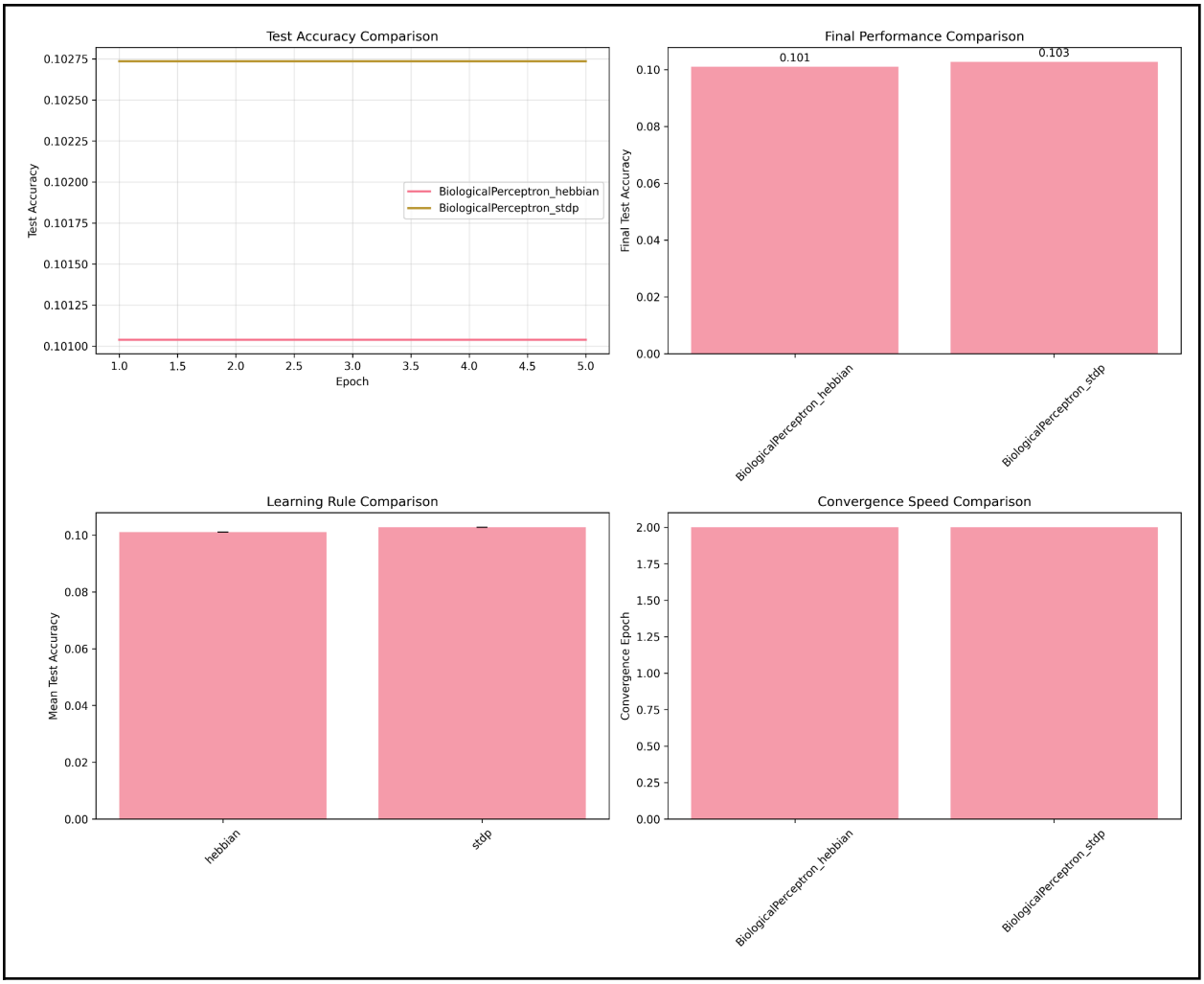


STDP Training Curves





Performance Comparison



Biological Constraints Results

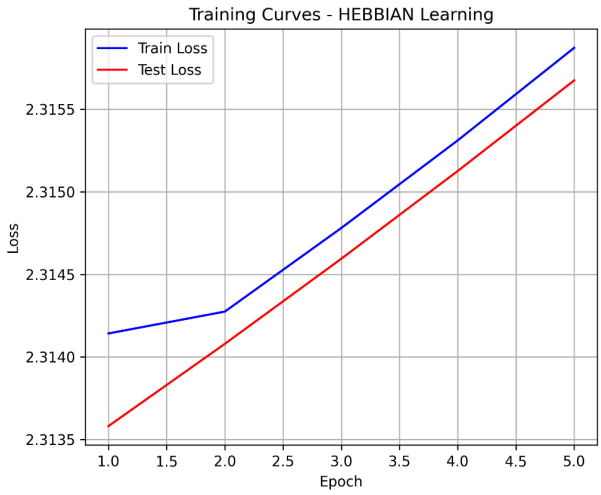
In this part of ablation, we include the biological constraints as mentioned in the previous sections ranging from sparsity, local processing, etc.

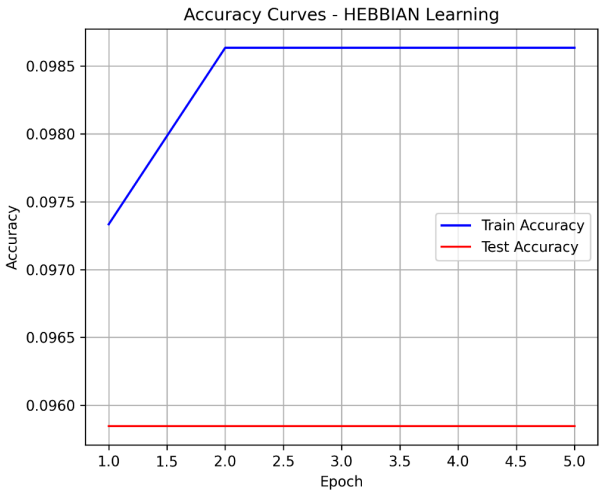
```
EXPERIMENT SUMMARY:
INFO:__main__:----------------------------------------
INFO:__main__:Best Model: BiologicalPerceptron_stdp
INFO:__main__:Best Accuracy: 0.1010
INFO:__main__:Learning Rule: stdp
```

Model Performance Rankings	1. BiologicalPerceptron_stdp: 0.1010 2. BiologicalPerceptron_hebbian: 0.0958

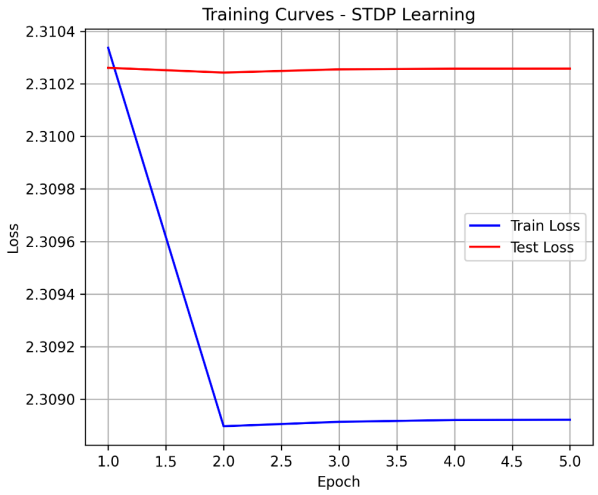
Learning Rule Analysis	HEBBIAN Learning: - Mean Accuracy: 0.0958 ± 0.0000 STDP Learning: - Mean Accuracy: 0.1010 ± 0.0000

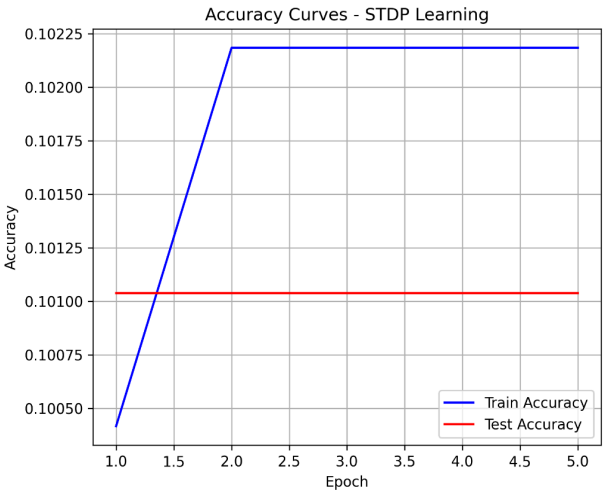
Hebbian Training Curves



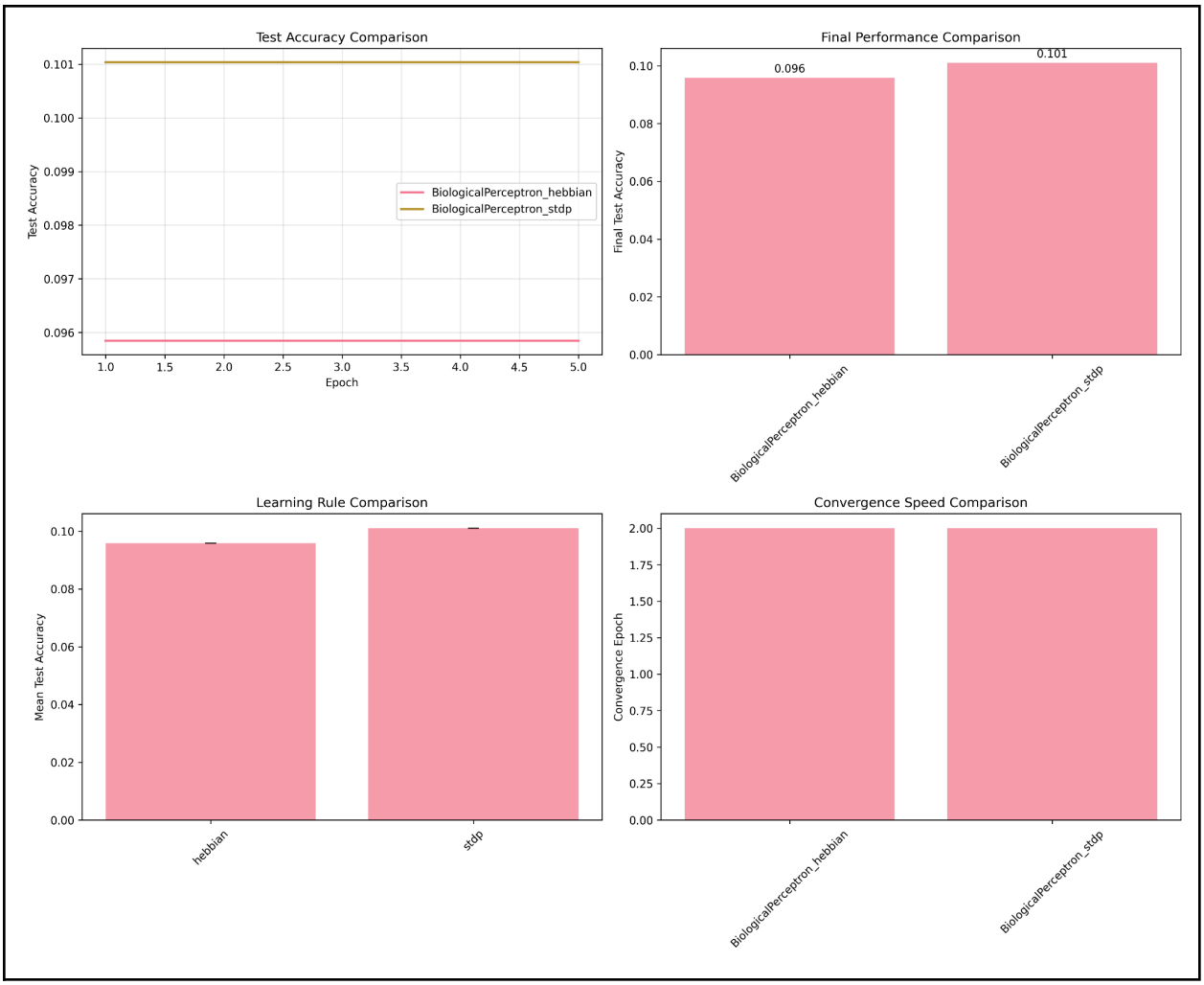


STDP Training Curves





Performance Comparison



In conclusion, we can observe and derive certain conclusions:

Firstly there are computational limits to what a single neuron can actually function. Secondly, Hebbian and STDP learning rules evolved for unsupervised pattern detection and temporal sequence learning, not supervised classification. The poor performance validates that these rules are inappropriate for classification tasks without additional mechanisms.

The meagre 10% performance, rather than being a failure, is a scientifically valuable result that validates our understanding of biological neural computation and provides important insights into the constraints that led to the improvement of neural networks.

References

- Neuroglancer Web App
- https://www.microns-explorer.org/cortical-mm3
- https://youtu.be/TxxqLk1xNmM)

- https://www.microns-explorer.org/ngl-instructions
- https://www.microns-explorer.org/
- https://tutorial.microns-explorer.org/quickstart_notebooks/01-caveclient-setup.html
- https://github.com/AllenInstitute/microns_tutorial