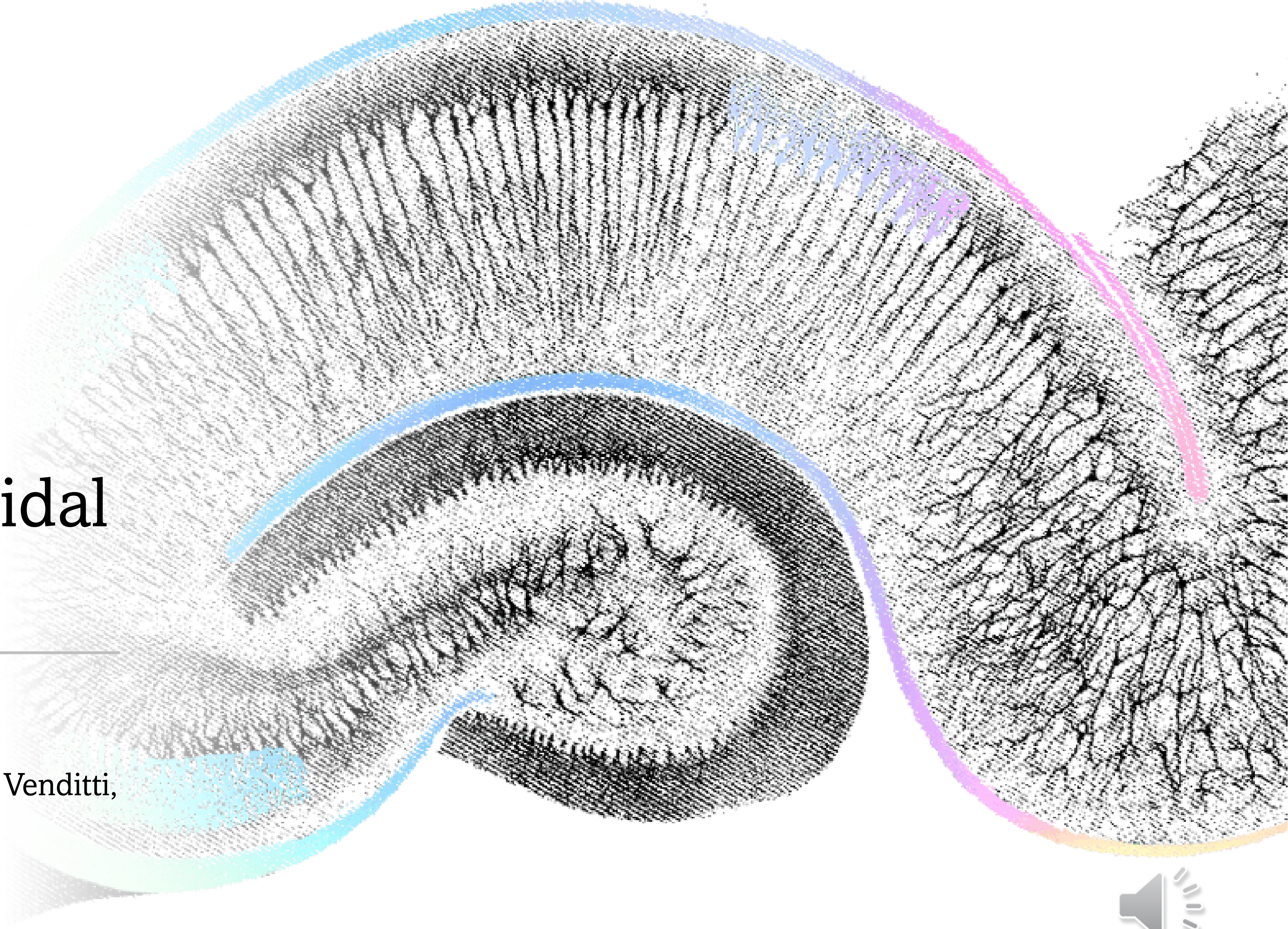
Modelling the Dendritic Tree Function of Cortical Pyramidal Neurons

MTHE 493 – Final Presentation

Amean Asad, Caroline Kim, Eric Venditti,
Jade Watson, Julia Wiercigroch,
Kai Ferrall, Paul Majewski



Agenda

1 **Background Information**
- Neuron Communication
- Neural Networks
- Information Theory

2 **Problem Definition**

3 **Design Solution**
- Neural Network Framework
- Model Training
- Information Theoretic Metrics

4 **Results**
- Weighted Sum Model
- Beniaguev's TCN Model
- FCN vs TCN

5 **Limitations & Trade Offs**

6 **Application**
- Rehabilitation for Neuromuscular Disease
- Brain-Computer Interface

7 **Triple Bottom Line (TBL)**

Neuron Communication

1. Input: Neurotransmitters to synapse
2. Non-linear processing of neurotransmitters at dendrites [2]
3. Integration of inputs at soma (different for each neuron type)
4. Output: Action potential – yes/no

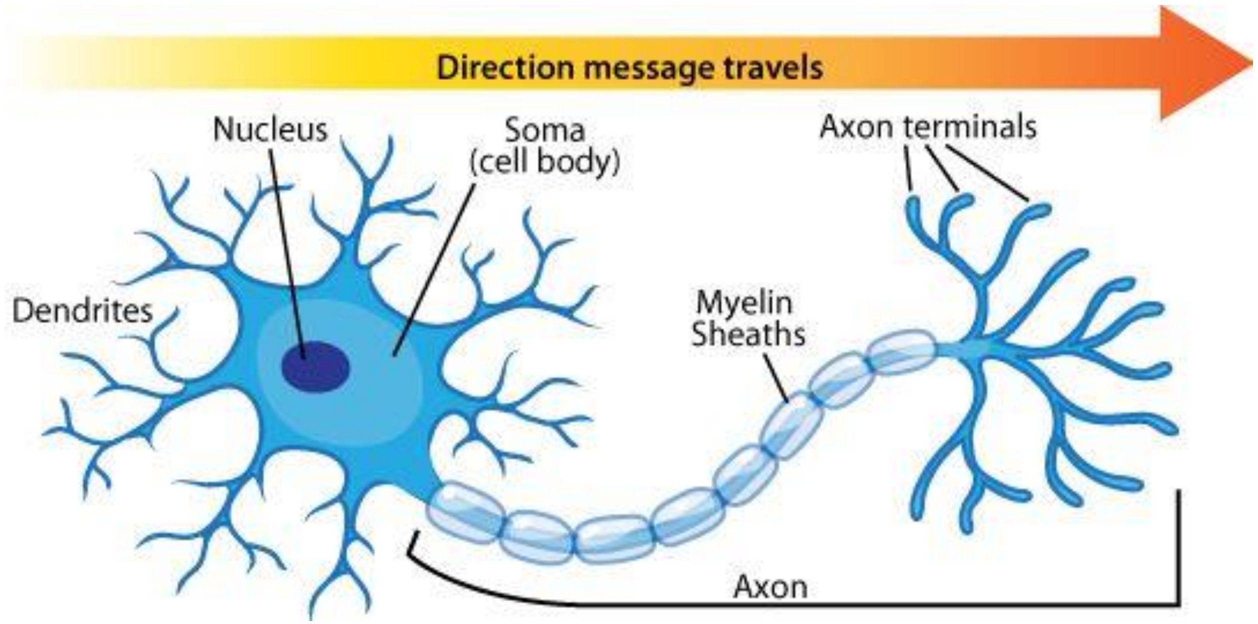


Figure 1. Anatomical structure of a simplified neuron [1].

[1] "Neuron Diagram & Types | Ask A Biologist." [Online]. Available: https://askabiologist.asu.edu/neuron-anatomy. [Accessed: 04-Dec-2021].
[2] David Beniaguev, Idan Segev, and Michael London. "Single cortical neurons as deep artificial neural networks". In: *Neuron* 109.17 (Sept. 2021), 2727{2739.e3. issn: 0896-6273. doi: 10.1016/J.NEURON.2021.07.002.

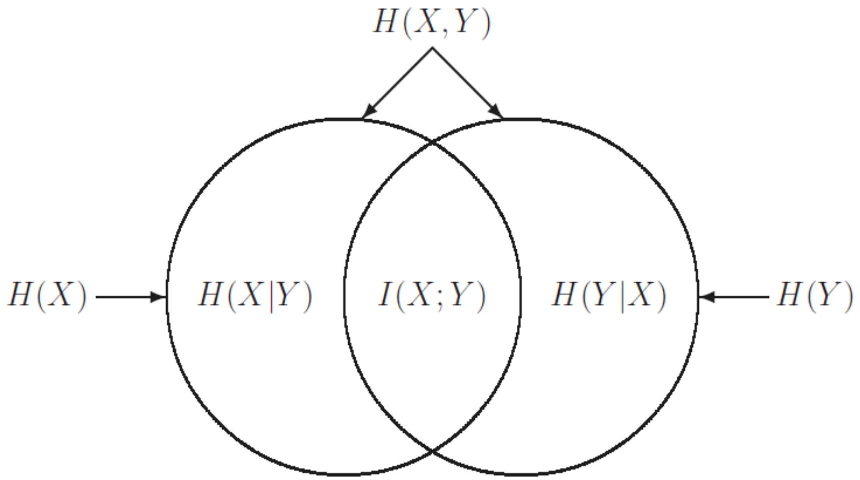
Information Theory

Overview
- The notion of computational complexity here is not equivalent to the widely understood computer science sense

Goal
- A method of quantifying the amount of information the artificial neuron can process

Approach
- Information theoretic approach

High-level prerequisites:
- Probability distributions for the excitatory and inhibitory inputs
- Conditional distributions between the input and output
- A method of transforming the inputs such that they can be modelled as a distribution



Research Question

Current gap:

A simple, accurate method of modelling neuron output does not exist

Research questions:

1. By altering the parameters of a Neural Network, how are accuracy and computational complexity affected?
2. How can the computational complexity be quantified?

Problem Definition

Aim:	Create a **Neural Network** to model the nonlinear function of an L5 pyramidal cortical neuron's dendritic tree and summarize its computational complexity

Constraints

- Current research is not well-documented
- Number of dendrite branches, E-I balance, neurotransmitters involved
- Spikes in dataset are very infrequent

Functional Requirements

- Seamlessly integrable within a larger model of the brain
- Cannot be computationally expensive
- Must be done in PyTorch
- Have a metric for evaluating computational complexity

Risks

- Applicable standards: Ethics in Animal Research
- Economic, environmental, cultural considerations

Design Solution

Data Pre-Processing

- Explore aggregate statistics of the simulation used to assess the variability of the data
- Identify redundancies

Neural Network Designs

- Fully Connected Networks
- Convolutional Networks
- Temporal Networks

Information Theoretic Analysis

- Create probability distributions
- Measuring mutual information and entropy
- Comparison between models for perspective

Neural Network Framework

1. Data Pre-Processing
Parse raw simulation data to extract synaptic inputs and spike train outputs. As well as, set up PyTorch classes for model design.

2. Model Design
Using Fully Connected Networks, Convolutional Networks, and Temporal Convolutional Neural Networks to perform the prediction

3. Model Training
Training on a dedicated GPU or cloud computing using a learning algorithm

4. Evaluation
Evaluate the model on a dedicated test data set using Mean Square Error (MSE) loss.

5. Deployment
Publish the model as a reusable module using Python and PyTorch

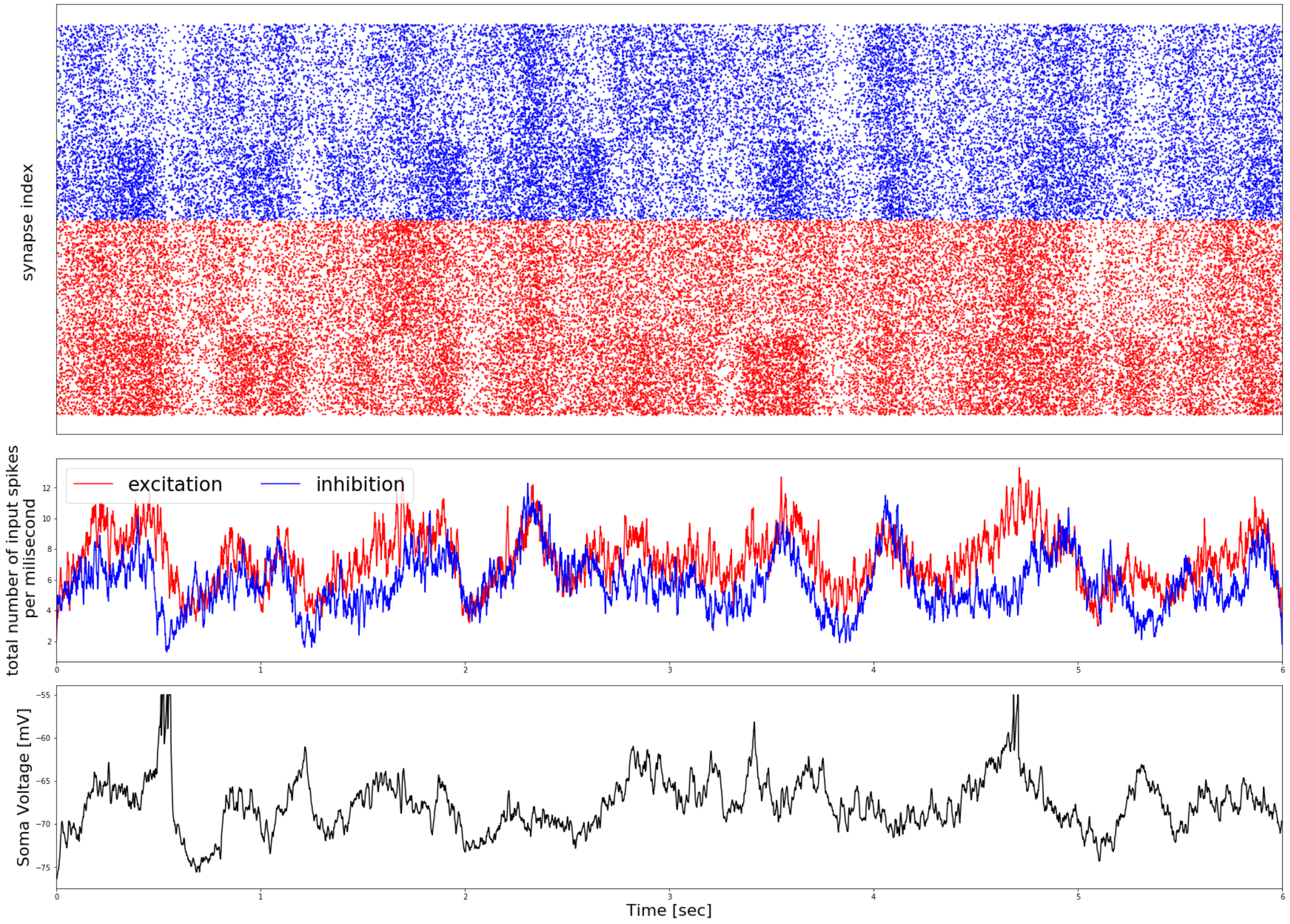
Data

Dataset: 10-hour simulation time of cortical layer 5 pyramidal neurons

- **Inputs:** a vector of binary values, each representing a spike of an inhibitory or excitatory input to the dendritic tree
- **Output:** a real-valued vector representing the somatic voltage
- **Constraints:** type of neuron used, the distribution of synapses, and the number of synapses in the dendritic tree

---

The dataset has been successfully been used in research to develop accurate DNNs to represent neuronal activity [2].

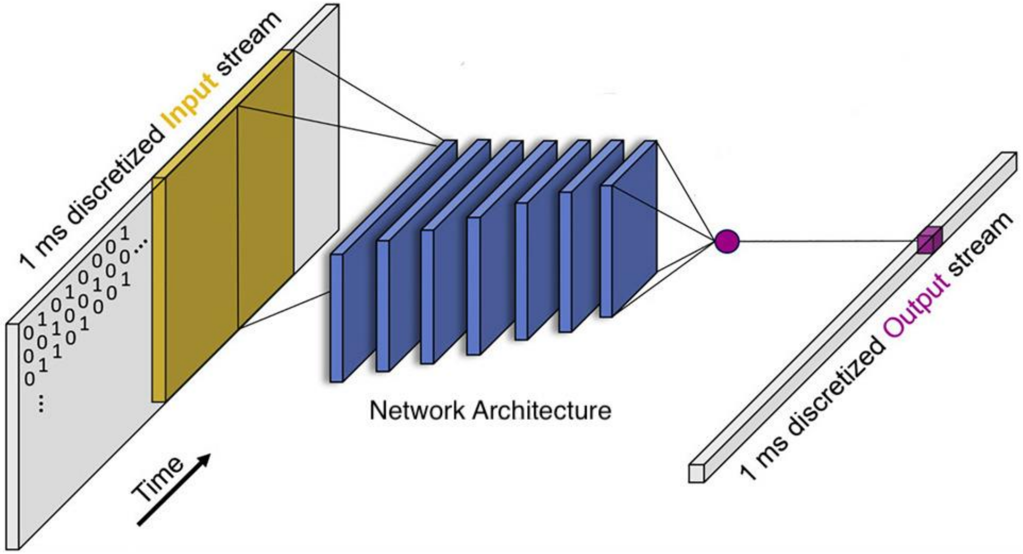


[2] David Beniaguev, Idan Segev, and Michael London. "Single cortical neurons as deep artificial neural networks". In: *Neuron* 109.17 (Sept. 2021), 2727{2739.e3. issn: 0896-6273. doi: 10.1016/J.NEURON.2021.07.002.

Data

Overview

- The input data is discretized by 1ms time steps
- Spike train: a sequence of binary values $S(t)$ such that $S(t) \in [0,1]$, t is an index of 1ms time intervals
- The spike trains have a corresponding output $y_{voltage}(t)$ representing the somatic voltage of the neuron



Model input/output

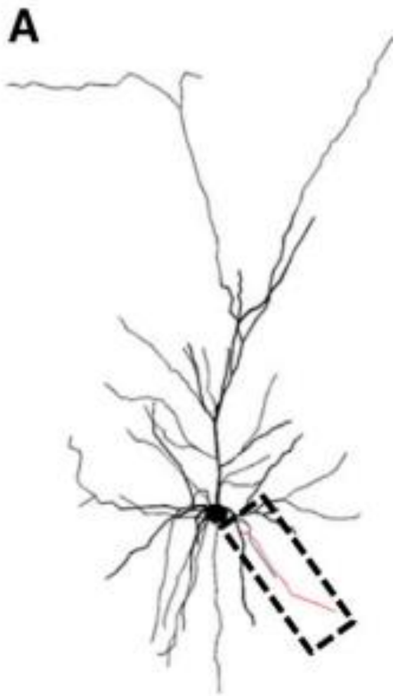
- Input (spike train): $X(s)$ where $s \in \{1, \cdots, N_{syn}\}$ denotes the synapse index in the dendritic tree and$t \in \{1, \cdots, T\}$ denotes the time index
- Goal: to predict $y_{voltage}(t)$ based on a time input window of size K, where vector

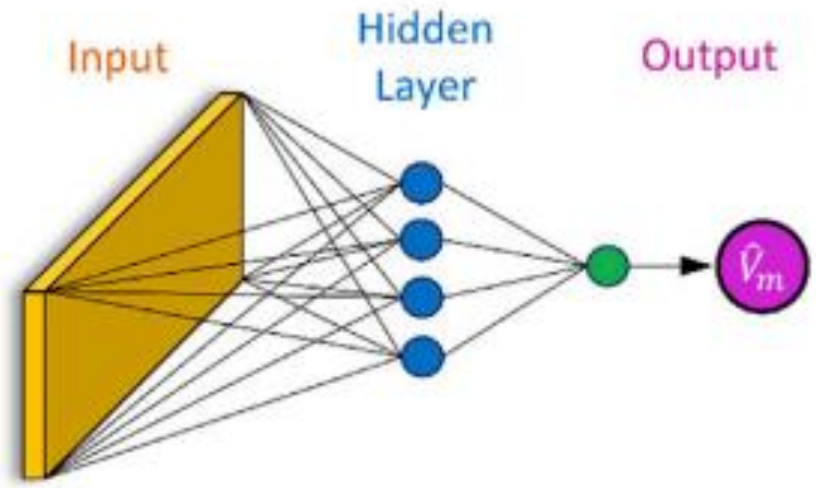
$$\hat{x}_{t_i} = [X(s,t)] \, , s \in \{1, \cdots, N_{syn}\} \, , \quad t \in \{t_i, t_i - 1, \cdots, t_i - K\}$$

- Output: $\hat{y}_{voltage}(t) = ANN(\hat{x}(t), \theta)$, where ANN represents the architecture and θ represents model parameters. الختيارc

Model Design

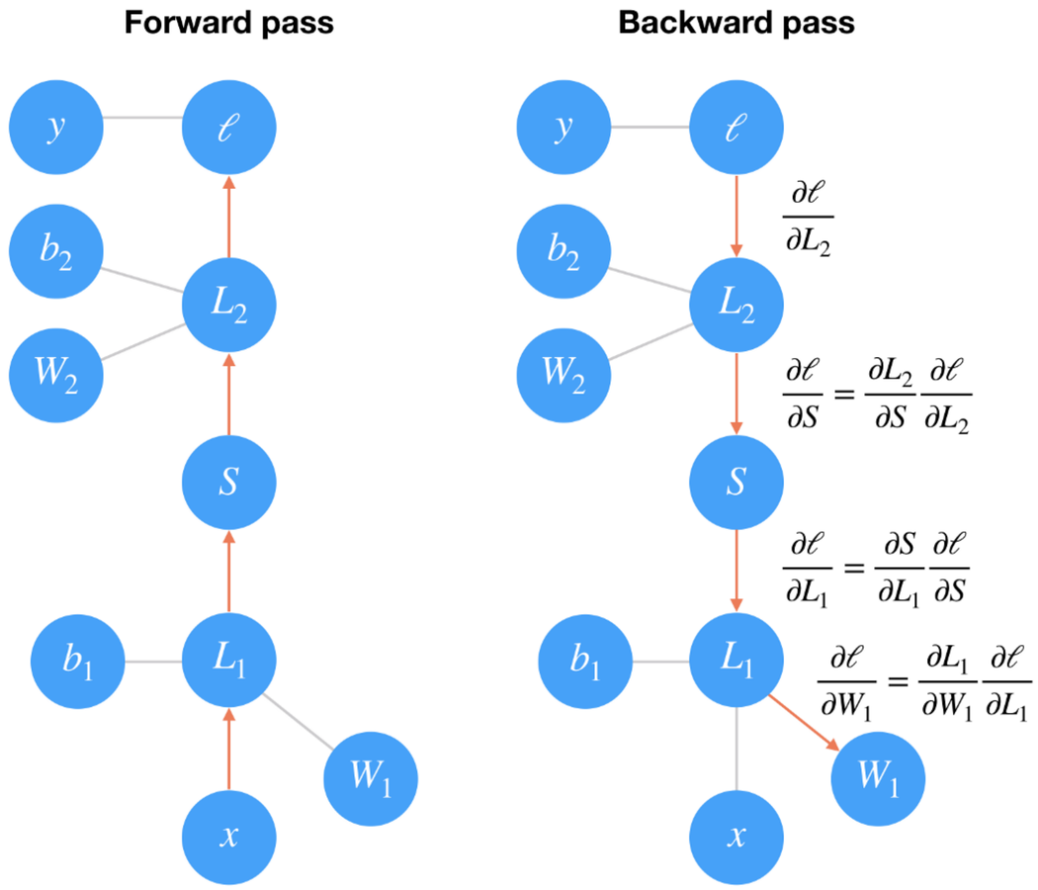
Goal: Fit the I/O relationship of a dendritic tree model using an analogous neural network





[2] David Beniaguev, Idan Segev, and Michael London. "Single cortical neurons as deep artificial neural networks". In: *Neuron* 109.17 (Sept. 2021), 2727{2739.e3. issn: 0896-6273. doi: 10.1016/J.NEURON.2021.07.002.

Model Training



$$\boxed{\frac{\partial \ell}{\partial W_1} = \frac{\partial L_1}{\partial W_1}\frac{\partial S}{\partial L_1}\frac{\partial L_2}{\partial S}\frac{\partial \ell}{\partial L_2}}$$

$$\boxed{W_1' = W_1 - \alpha\frac{\partial \ell}{\partial W_1}}$$

[3] T.Zaman, “Training Nerual Networks using PyTorch”, Towards Data Science, 2019.

Temporal Convolutional Nets

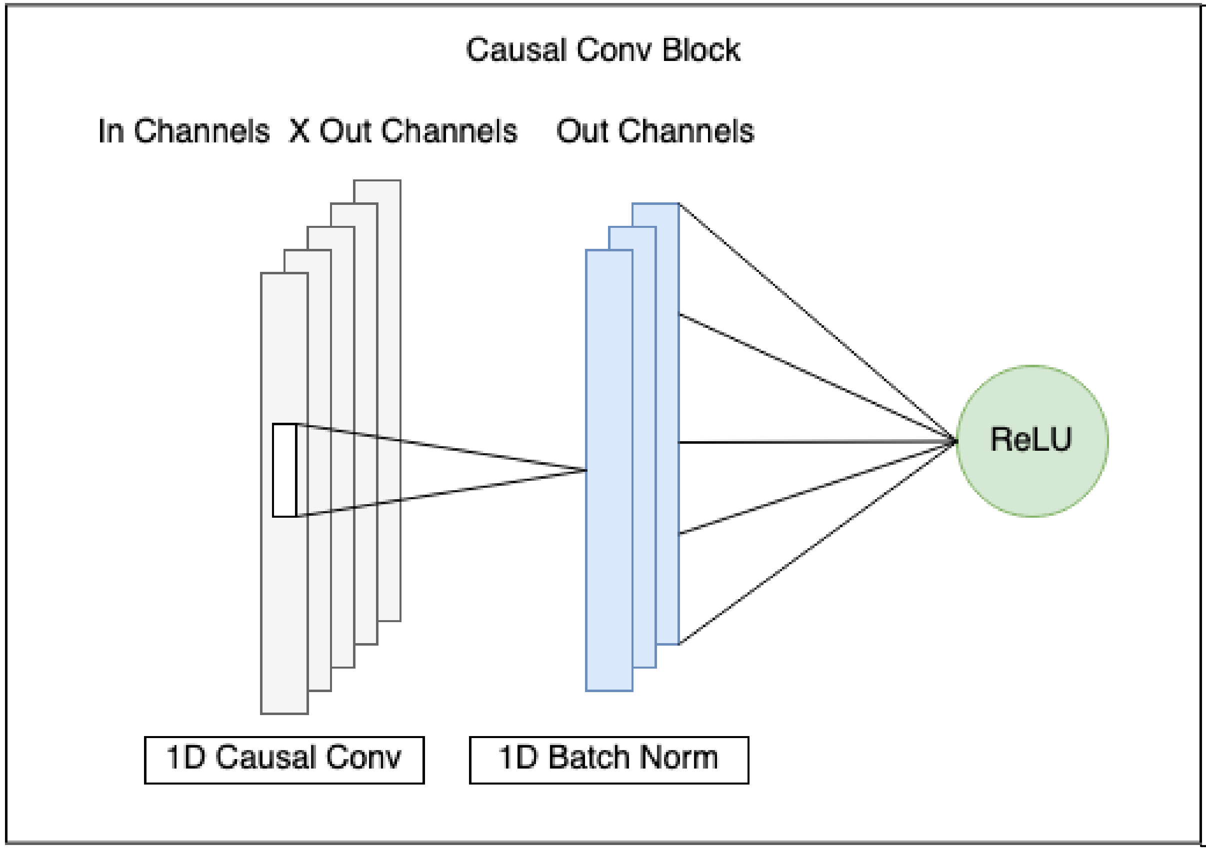
- An architecture that is commonly used for sequence modelling.
- **Temporal Modelling:** Captures the temporal aspect of dendritic tree modelling.
- **Causality:** The convolutions used are **causal,** ensuring no information leakage from future sequences [4].
- **Variable Length Input:** Can take sequences of any length and map it to output sequences with the same length, allowing testing for multiple window sizes for spike train sequences.
- **Receptive Fields:** Uses dilation to give flexibility for the size of receptive fields used.

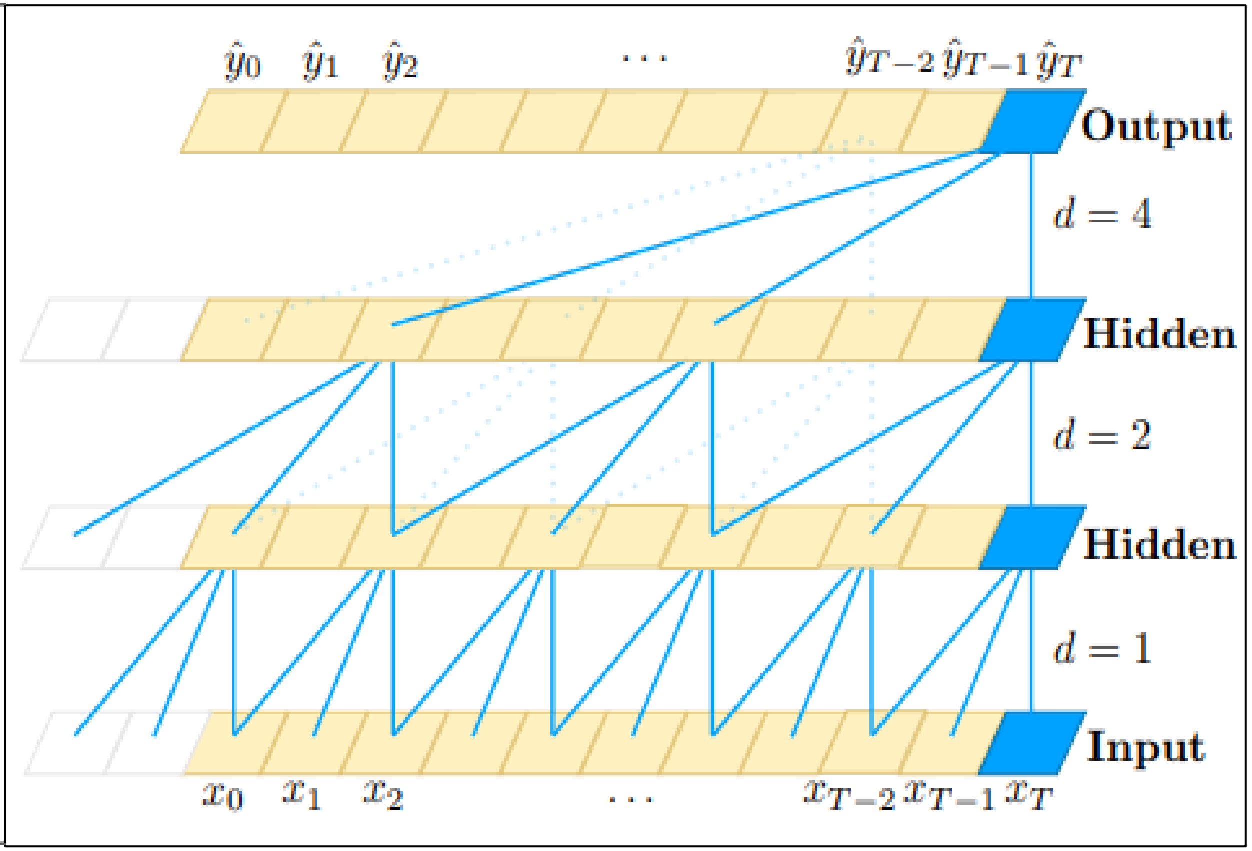
[4]Oord, Aaron van den, et al. 'WaveNet: A Generative Model for Raw Audio'. *ArXiv:1609.03499 [Cs]*, 2, Sept. 2016. *arXiv.org*, http://arxiv.org/abs/1609.03499.

TCN Machinery

The main driving mechanism of TCN's is dilated 1D convolutions:

$$F(s) = (x \ *_d \ f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d_i}$$





Training Process

Training process was standardized for all our network types

For each batch:

- Forward propagate the batch inputs in the network
- Compute the loss using the MSE loss function.
- Uses an ADAM optimizer to compute the gradients for backpropagation
- Update the weights for each layer.

Compute the loss on test and training datasets as well as the 1% accuracy for each epoch.

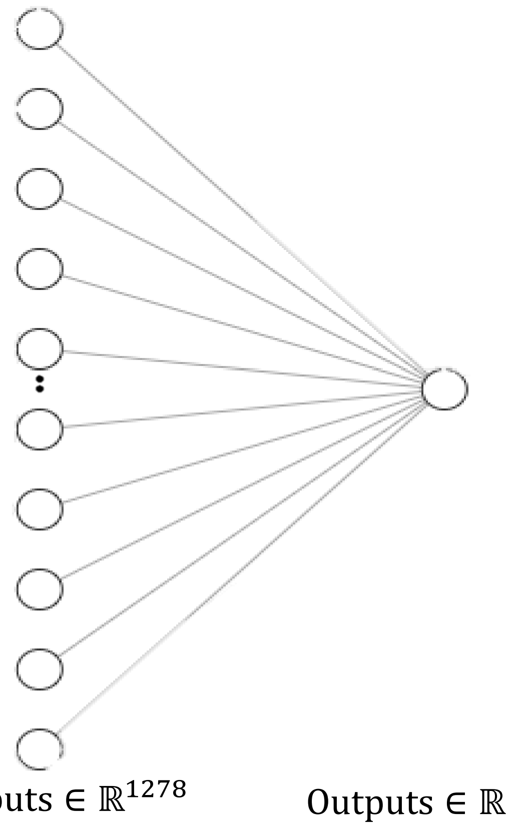
Adam Optimizer Update Rule:

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

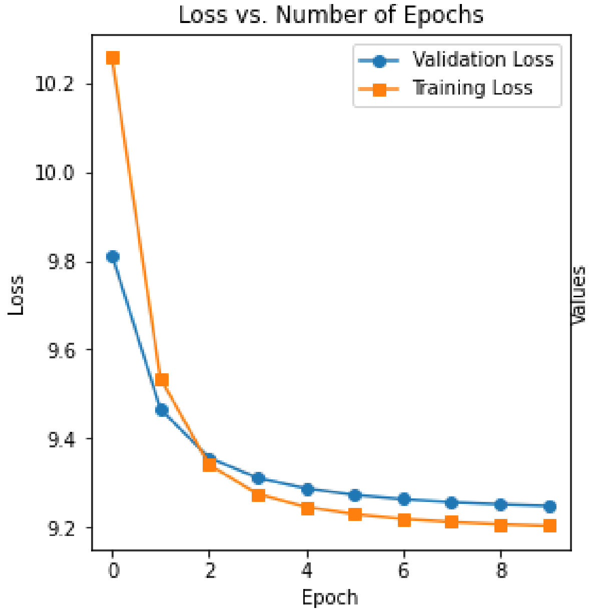
$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

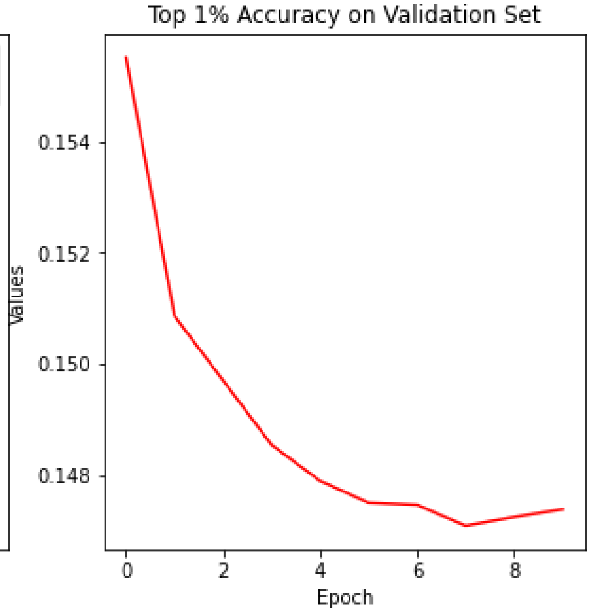
Results

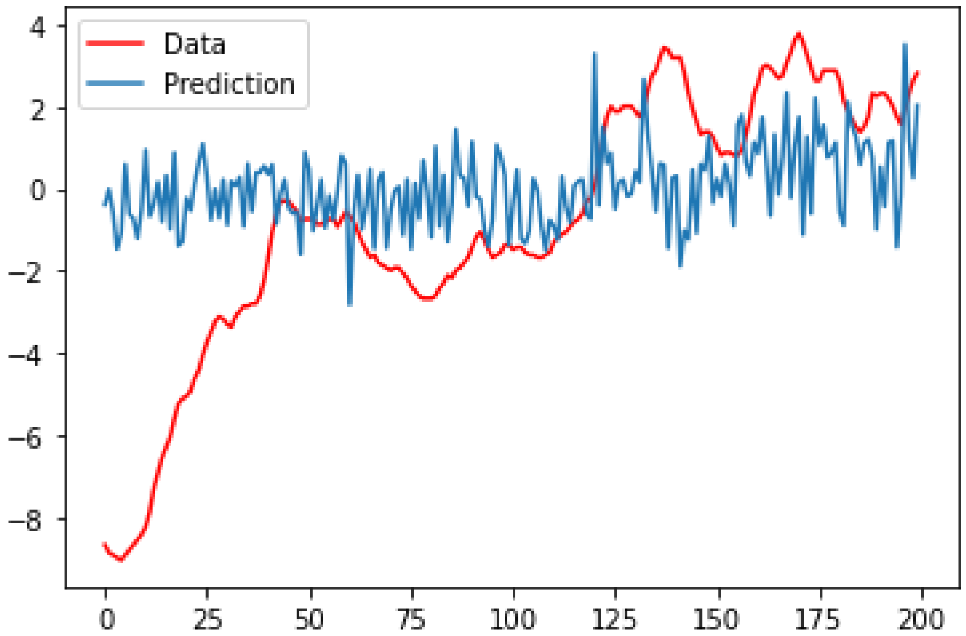
Current Research Practice: **Weighted Sum Model**



Architecture Sketch

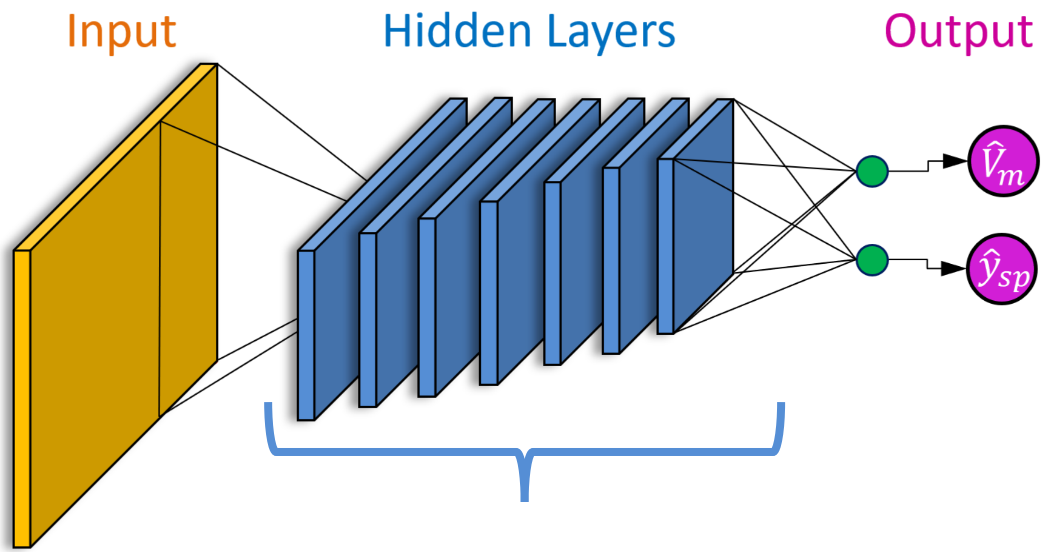






Results

Beniaguev's Temporal Convolutional Network Model



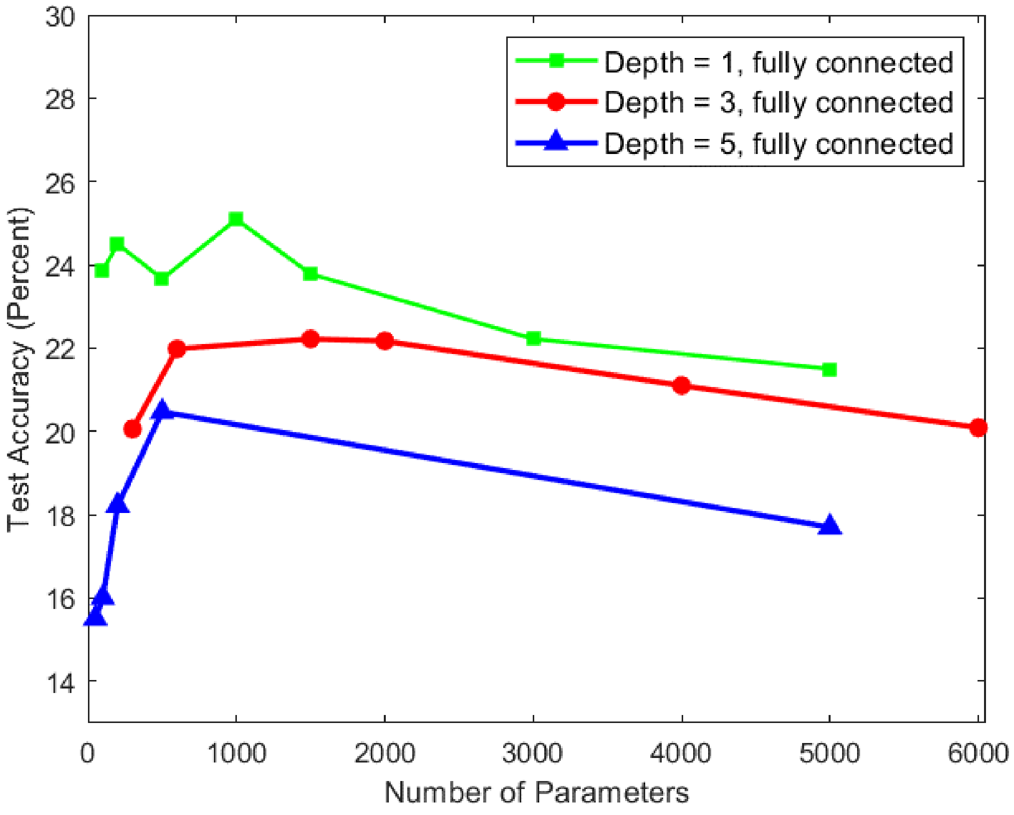
One Hidden Layer consists of:

- Causal Convolution (1 Dimension)
- Rectified Linear Unit (ReLU) Activation Function
- Batch Normalization

[2] David Beniaguev, Idan Segev, and Michael London. "Single cortical neurons as deep artificial neural networks". In: *Neuron* 109.17 (Sept. 2021), 2727{2739.e3. issn: 0896-6273. doi: 10.1016/J.NEURON.2021.07.002.

Results – Fully-Connected Neural Network

Increasing the complexity of the neural network (number of layers or parameters) does not necessarily increase accuracy.



Results – 1-D Convolutional Neural Networks

Layers	Structure	Validation Accuracy
1	1278 channels → 1 output → Rectified Linear Unit Function (ReLU)	15.98%
2	1278 channels → 128 outputs → 1 output → ReLU	16.63%
	1278 channels → 150 outputs → 1 output → ReLU	14.47%
	1278 channels → 64 outputs → 1 output → ReLU	14.69%
3	1278 channels → 128 outputs → 64 outputs → 1 output → ReLU	15.80%
	1278 channels → 128 outputs → 32 outputs → 1 output → ReLU	16.06%
	1278 channels → 64 outputs → 32 outputs → 1 output → ReLU	15.54%
4	1278 channels → 128 outputs → 64 outputs → 32 outputs → 1 output → ReLU	16.45%

Results: Temporal Convolutional Networks

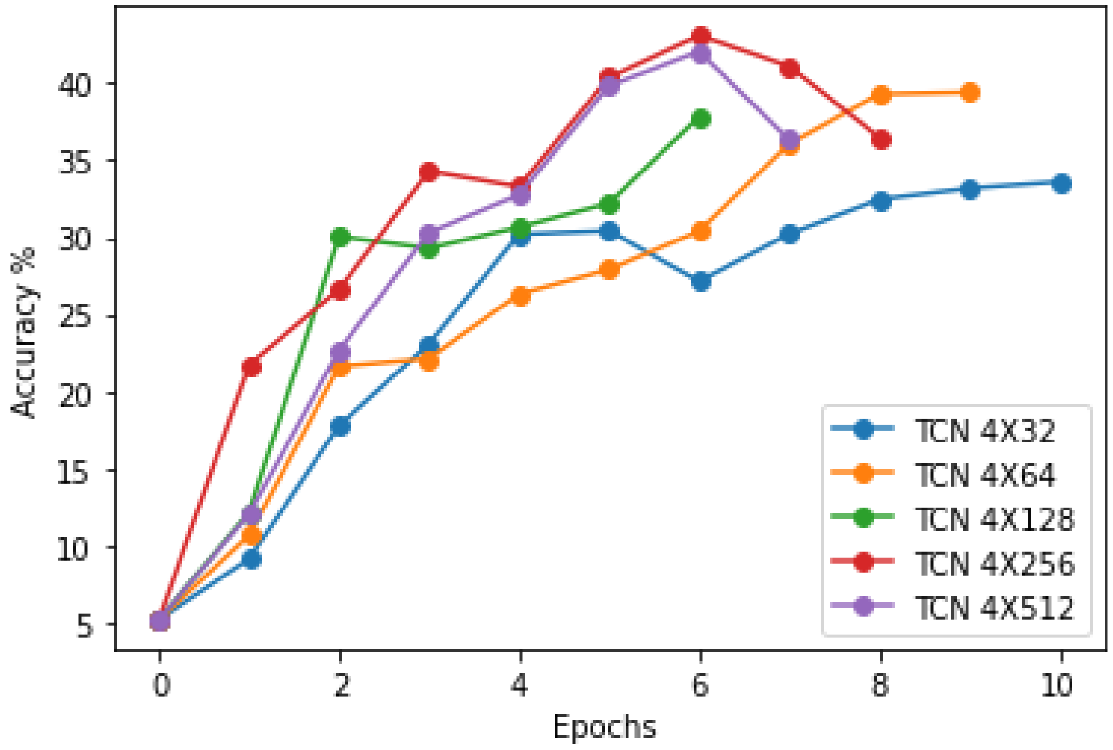
- A strong dependence was established between network width and accuracy.
- No dependence was found between network depth and accuracy for TCN's.

Model	No of Filters	1% Accuracy
TCN 4X32	32	30.52%
TCN 4X64	64	35.27%
TCN 4X128	128	39.88%
TCN 4X256	256	42.04%
TCN 4X512	512	43.79%

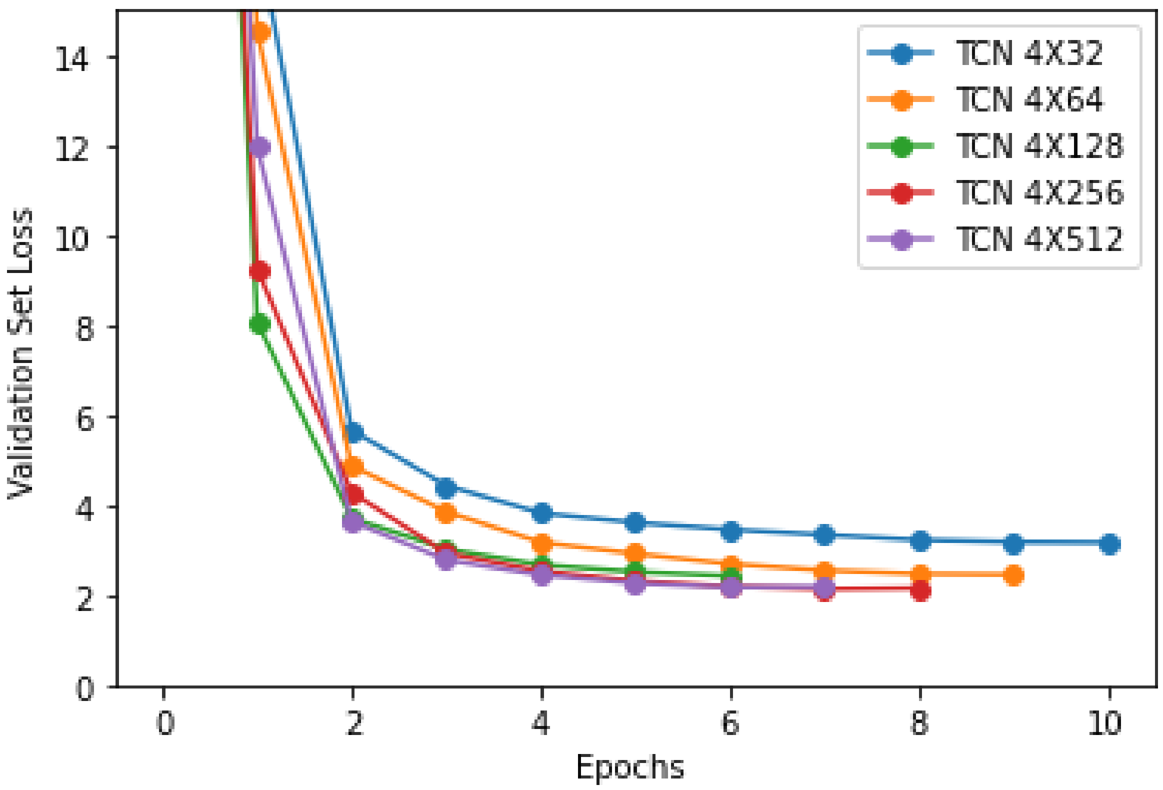
Results: Temporal Convolutional Networks

Both accuracy values and loss values appear to converge as network width increases.

Accuracy vs Epoch Number



Validation Loss vs Epoch Number

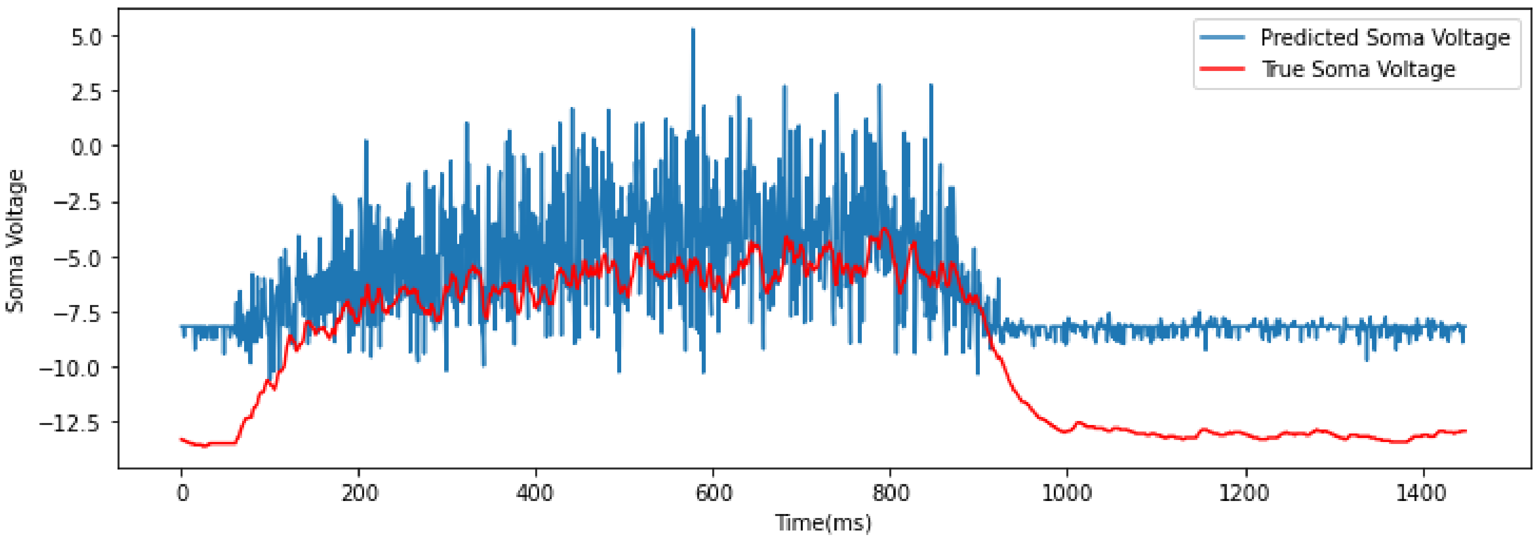


Results

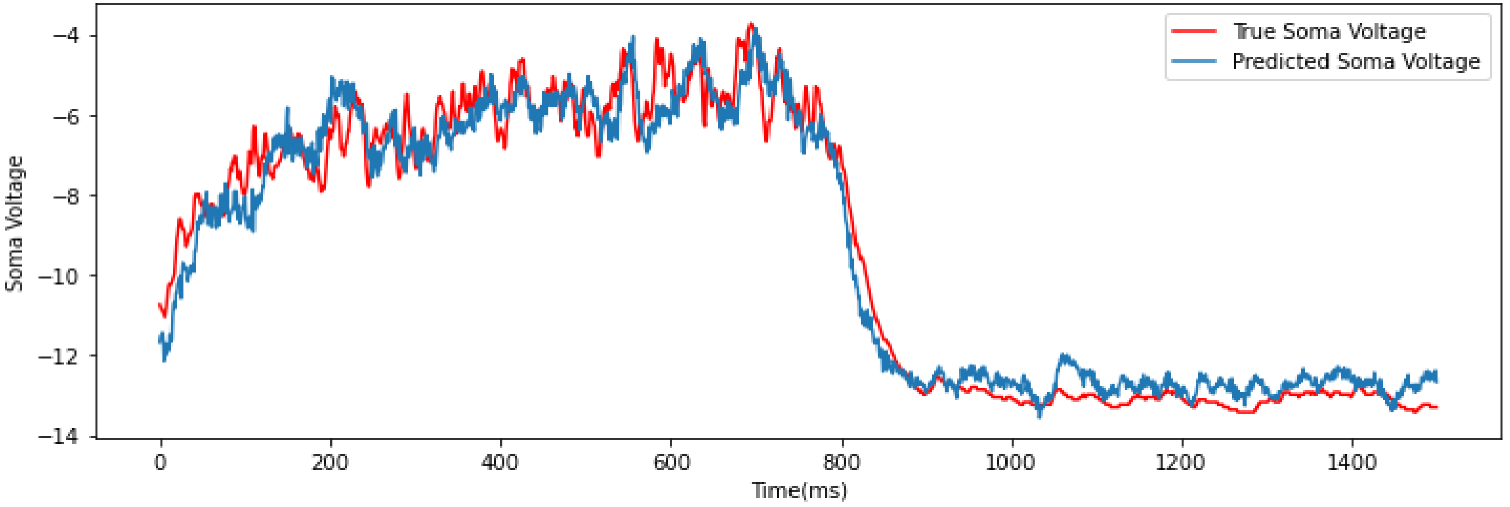
Results show significant improvement on the currently used models.

Category	Weighted Sum Model	TCN Model
1% Accuracy	10.3%	43%
20% Accuracy	15.4%	73.4%

Weighted Sum Prediction



TCN 4X256 Prediction



Quick Background

Aim: To encapsulate the information transfer from inputs to outputs

- Comparing information transfer of simulation data to that of models

Input Entropy (1)

$$H_x = -\sum_{x \in X} P(x) \log(P(x))$$

Mutual Information (2)

$$I(X;Y) = \sum_{x \in X, y \in Y} P(x) \mathrm{P(y|x)} \log \frac{P(y|x)}{P(y)}$$

Input Encoding (3)

$$e_i = \sum_{i=1}^{1278} x_i$$

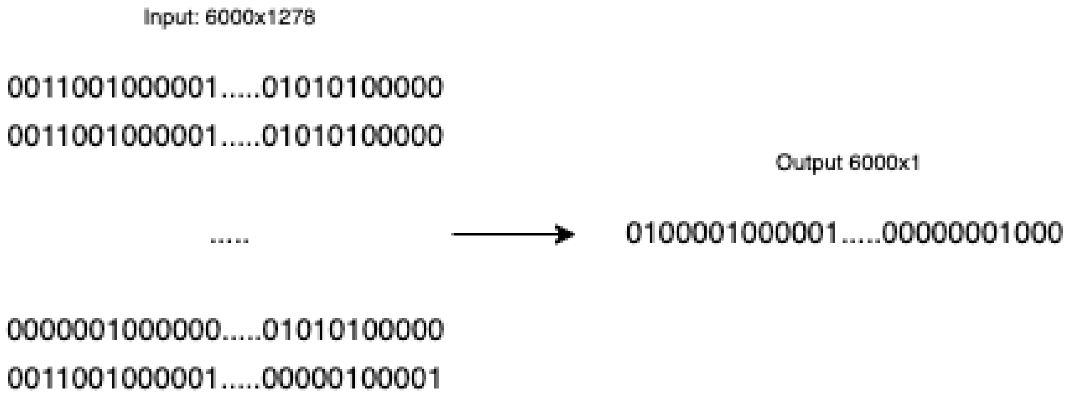
Average Transferred Entropy [4] (4)

$$F = < \frac{I}{H_x} > samples$$

[5] Fleur Zeldenrust, Sicco de Knecht, Wytse J. Wadman, Sophie Denève and Boris Gutkin. "Estimating the Information Extracted by a Single Spiking Neuron from Continuous Input Time Series". In: frontiers in Computation NeuroScience (Jun. 2017).

Information Theoretic Analysis

1. First approach using inhibitory and excitatory input mapping to spike train. [6]



Due to the sparsity of spikes in the output this approach did not work and gave mutual information of 0.

2. Second approach using inhibitory/excitatory inputs mapping to soma voltages
- Quantization of soma voltages
- Developed distributions and measured mutual information

[6] Gianni Pola, Simon R. Schultz, Rasmus S. Petersen and Stefano Panzeri. "A Practical Guide to Information Analysis of Spike Trains". [Online]. Available: http://www.schultzlab.org/resources/Papers/Pola_chapter.pdf. [Accessed: 07-Mar-2022].

Input Distribution

The input distribution was modelled as a Gaussian distribution with mean 13.3039 and variance 4.98.

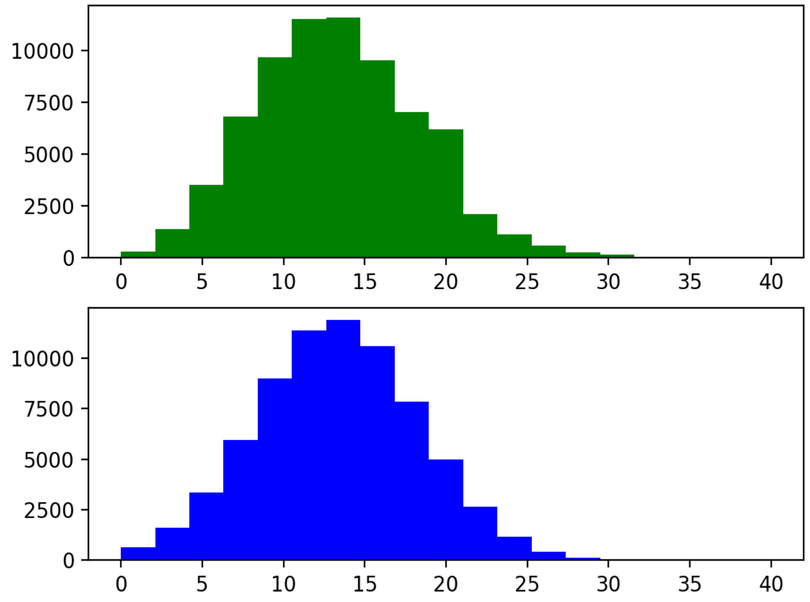
- This was done by getting the mean and standard deviation of the encoded inputs
- Input distribution (Gaussian pdf):

$$P(x) = \frac{1}{4.98\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-13.3039}{4.98}\right)^2}$$

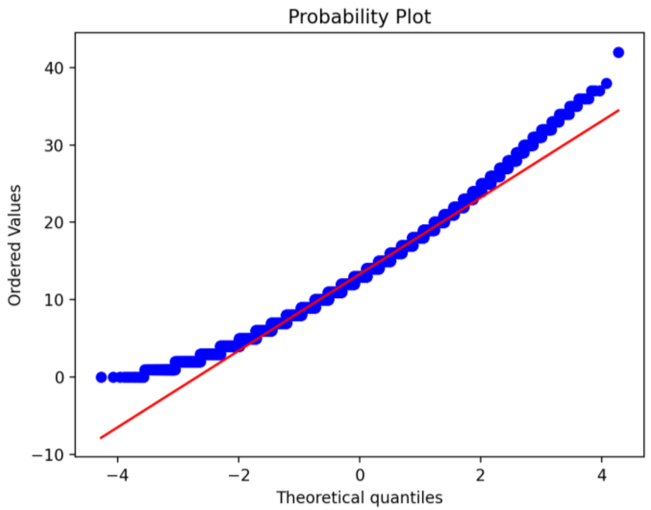
- This Gaussian distribution gives a simple calculation for the input entropy:

$$H_x = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2} = \frac{1}{2}\log(2\pi * 4.98^2) + \frac{1}{2}$$

$$= 4.14189\ bits$$



Input (green), Sample Gaussian (Blue)



Quantile-Quantile Plot of Input vs Theoretical Values

Developing the Conditional Distribution

Quantize soma voltages otherwise we will just get a uniform distribution.

As expected, the mutual information decreased and MSE increased with larger quantizer cells.

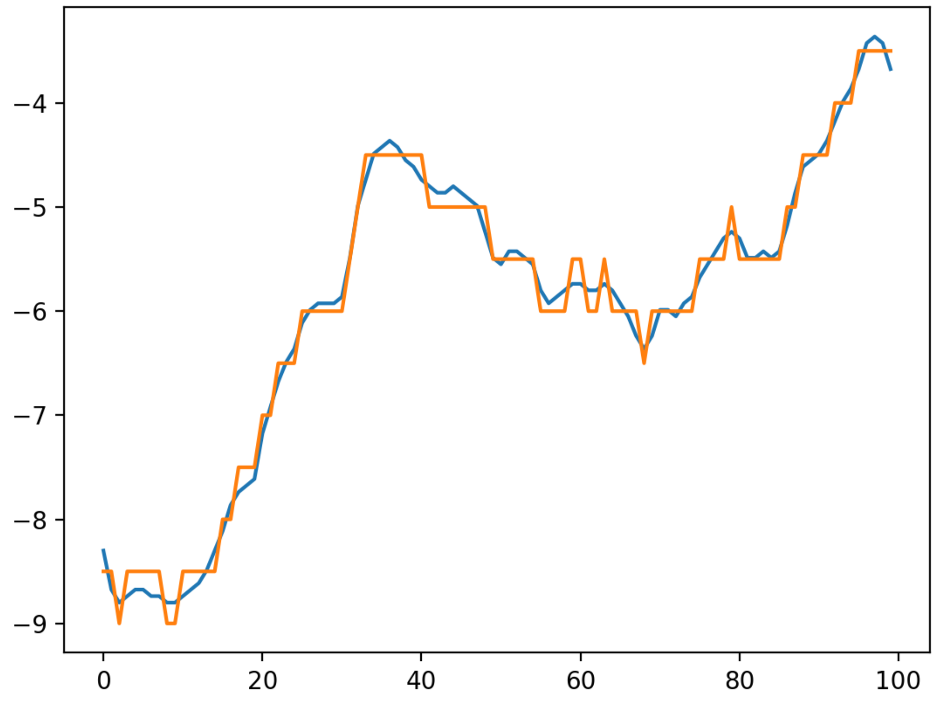
- Choice of quantizer will affect our conditional distribution.

Conditional Distribution: $P(y|x) = \frac{n_{ij}}{\sum_{k=1} n_{ik}}$

Mean Squared Error vs Quantizer:

MSE	**Quantizer Cell Size**
0.0052	0.25
0.0130	0.5
0.0243	0.75

Sample of 100 quantized soma voltage points:



Results

Simulation Data

Mutual Information (bits)	Transferred Entropy (bits)
0.331	0.079

Limitations of Results

- Difficult to contextualize values in absolute terms
- Large trade-off between quantizing and change in distributions
- Summation encoding is not granular

Model TCN 4x64 Data

Mutual Information (bits)	Transferred Entropy (bits)
0.253	0.0611

Model TCN 4x128 Data

Mutual Information (bits)	Transferred Entropy (bits)
0.284	0.068

Model TCN 4x256 Data

Mutual Information (bits)	Transferred Entropy (bits)
0.293	0.071

Mutual Information (bits)	Transferred Entropy (bits)
0.331	0.079

Model Name	Mutual Information (bits)	Transferred Entropy (bits)
4x64	0.253	0.0611
4x128	0.284	0.068
4x256	0.293	0.071

DNN Limitations and Trade-offs

- Model:
  - The model architectures did not capture any information on the respective location of the synapses on the dendritic tree.
  - Model sizes were restricted by available GPU memory.
- Training:
  - Only trained on 3% (3GB) of the data due to resource limitations in compute power and time limitations to produce results.
  - Empirical nature of DNN modeling makes it difficult to determine optimal training parameters and extensibility of the shown results.
  - Batch sizes were restricted by memory capacity.

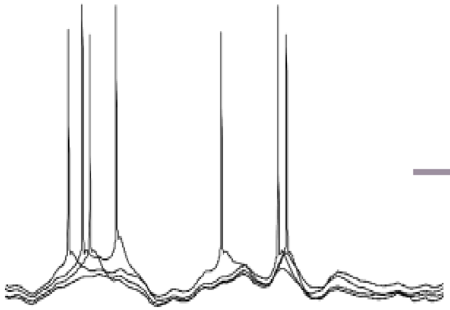
Rehabilitation for Neuromuscular Disease

- ***Neuromuscular diseases:*** dysfunction of muscles due to problems with nerves and muscles [7], incurable [8]
- Affects over **50,000** Canadians [9]
- Current rehabilitation: adaptive devices [10]

Stephen Hawking, assistive speech device that relies on small cheek movements [4]

[7] Mayo Clinic, "Neuromuscular Disease - Overview," *Mayo Clinic*. https://www.mayoclinic.org/departments-centers/neuromuscular-disease-group/overview/ovc-20443670 (accessed Nov. 26, 2021).
[8] G. Carter, "Rehabilitation Management of Neuromuscular Disease: Overview, Clinical Characteristics of Neuromuscular Disease, Management of Neuromuscular Disease," Jun. 26, 2021. https://emedicine.medscape.com/article/321397-overview (accessed Nov. 26, 2021).
[9] J. Medeiros, "How Intel Gave Stephen Hawking a Voice | WIRED," *Wired*. https://www.wired.com/2015/01/intel-gave-stephen-hawking-voice/ (accessed Nov. 26, 2021).
[10] HealthPartners, "Muscular Dystrophy | HealthPartners." https://healthpartners.ca/healthy-canadians/life-saving-research/muscular-dystrophy (accessed Nov. 26, 2021).

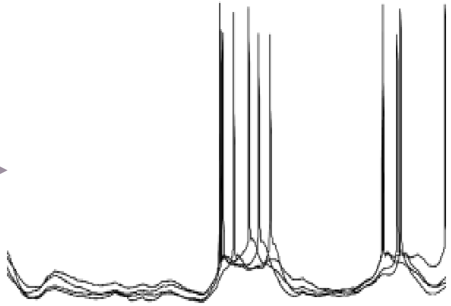
Application: Brain-Computer Interface (BCI)



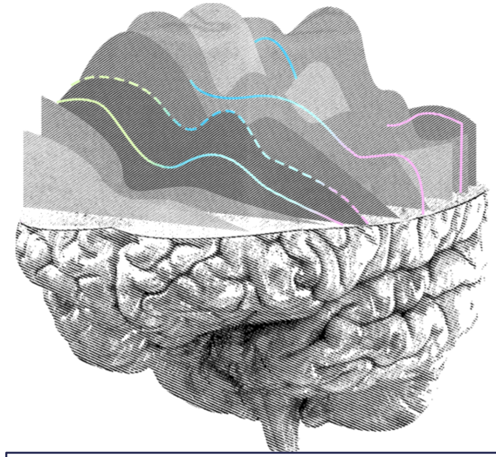
Input signal [1]

BCI [2]

?



Output signal [11]



Rehabilitated brain [12]

- BCI analyzes brain signals that encodes intention from a person, decodes the intended output, then translates the intention to a device to complete an action [3]

[2] David Beniaguev, Idan Segev, and Michael London. "Single cortical neurons as deep artificial neural networks". In: *Neuron* 109.17 (Sept. 2021), 2727{2739.e3. issn: 0896-6273. doi: 10.1016/J.NEURON.2021.07.002. .
[11] P. Studio, “Approach,” Neuralink. https://neuralink.com/approach/ (accessed Nov. 26, 2021).
[12] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, “Brain-Computer Interfaces in Medicine,” Mayo Clin Proc, vol. 87, no. 3, pp. 268–279, Mar. 2012, doi: 10.1016/j.mayocp.2011.12.008.

Triple Bottom Line: BCI

People

- Patients with neuromuscular disease benefit from a higher quality of life
- Decreases chances of injury
- Relieves burden on caregivers

Planet

- Reduce need to change adaptive devices based on changing function
- Reduce reliance on primate and rodent subjects for scientific testing

Profit

- Cost of devices decreases for patients
- Medical costs for patients decreases

Acknowledgements

- Dr. Gunnar Blohm
- Dr. Yanglei Song
- David Beniaguev, Idan Segev, and Michael London (Single cortical neurons as deep artificial neural networks)
- Fleur Zeldenrust, Sicco de Knecht , Wytse J. Wadman, Sophie Denève, and Boris Gutkin (Estimating the Information Extracted by a Single Spiking Neuron from a Continuous Input Time Series)