Decoding handwritten characters from neural activity using RNNs

Tyler Singer-Clark
Tarun Tiwari
Karamjeet Gulati
Adil Bukhari

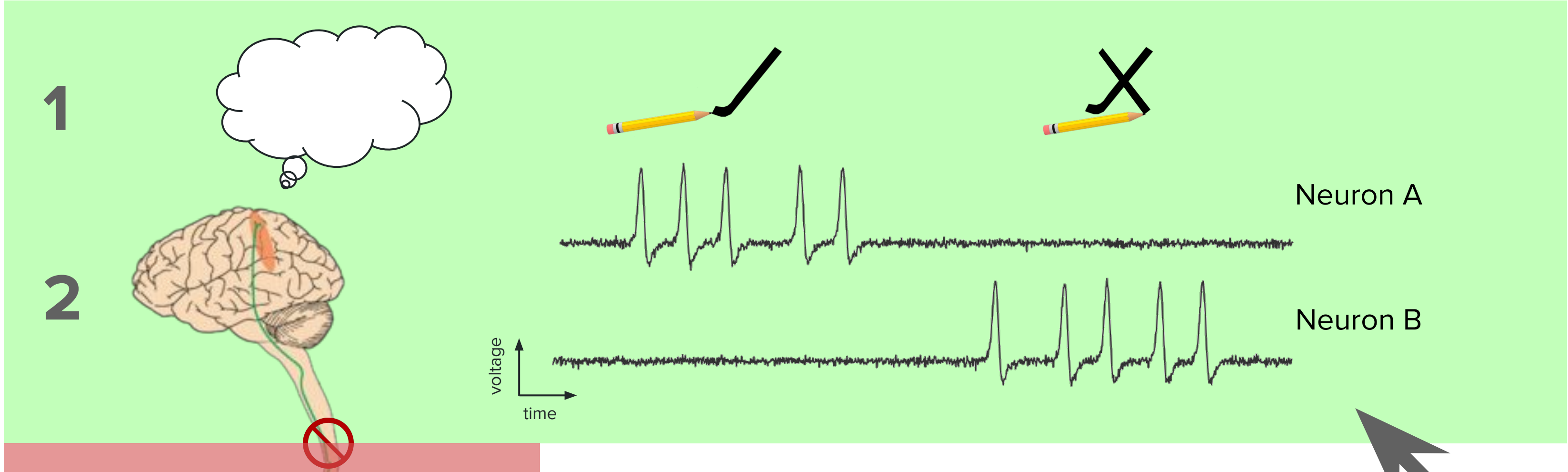
Brain-computer interfaces (BCI) help people with paralysis

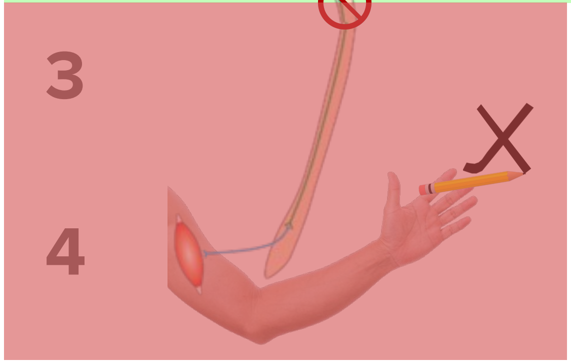
Think about moving

Control a device

Hochberg et al. 2006

Overview of the neural control of movement



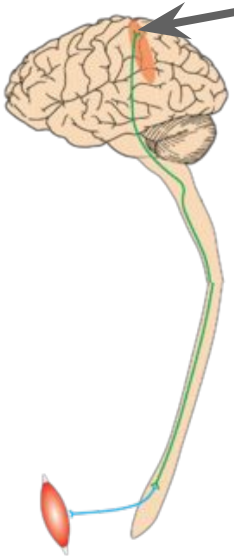


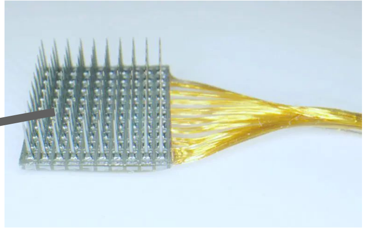
Paralysis due to:
- **ALS**
- **brainstem stroke**
- **spinal cord injury**

We can record neurons in the brain directly!

Microelectrode arrays record individual neuron activity

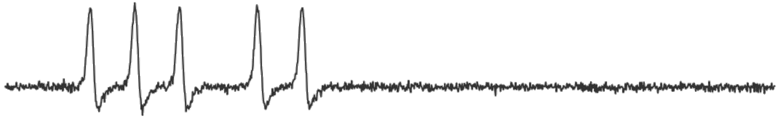
surgically
implanted



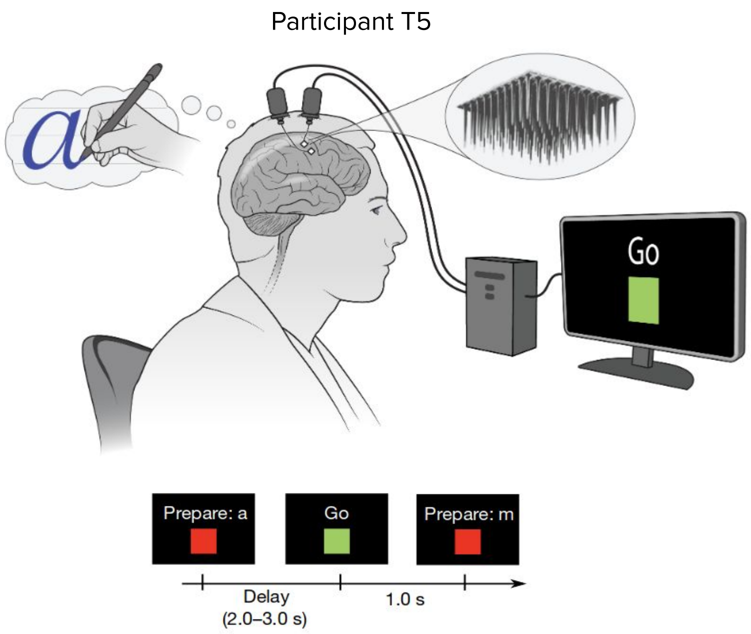


Blackrock Neurotech's Utah array

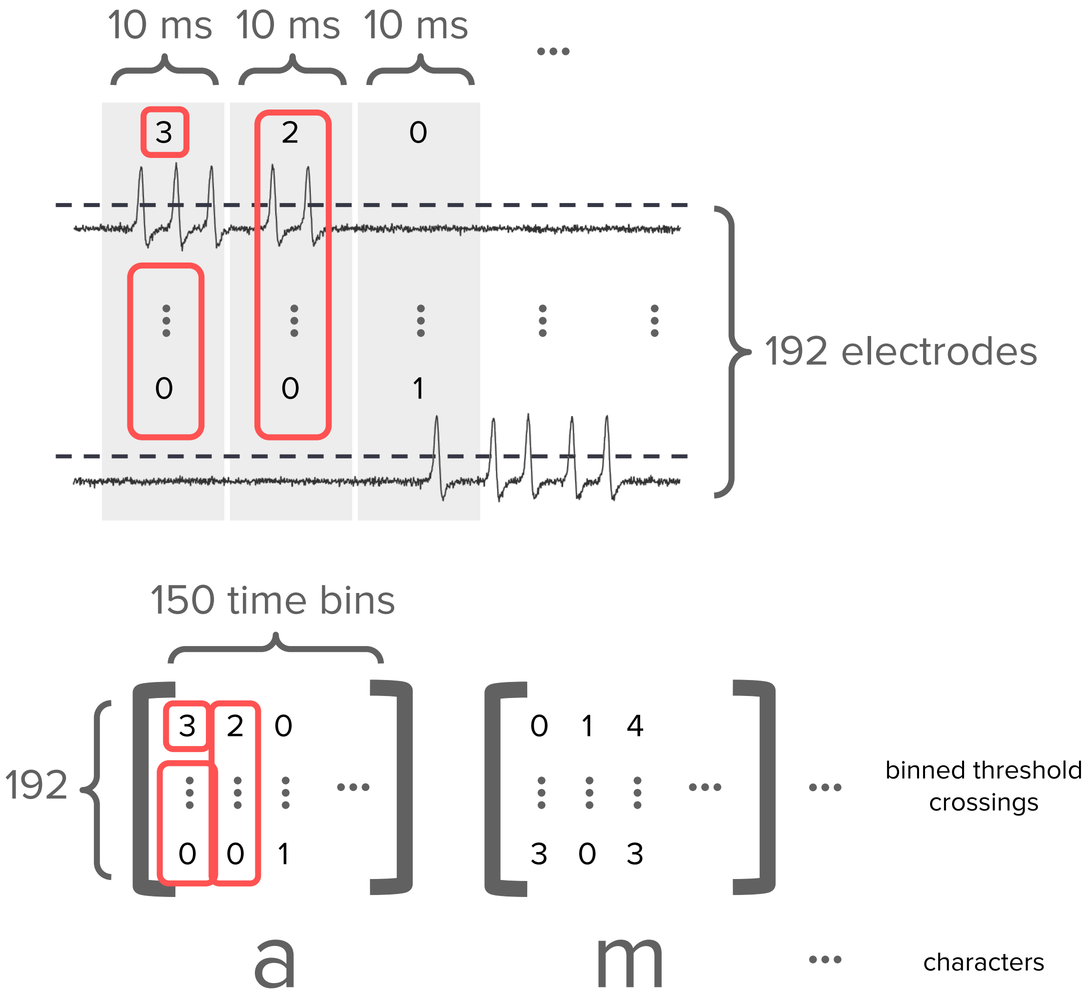
96 x



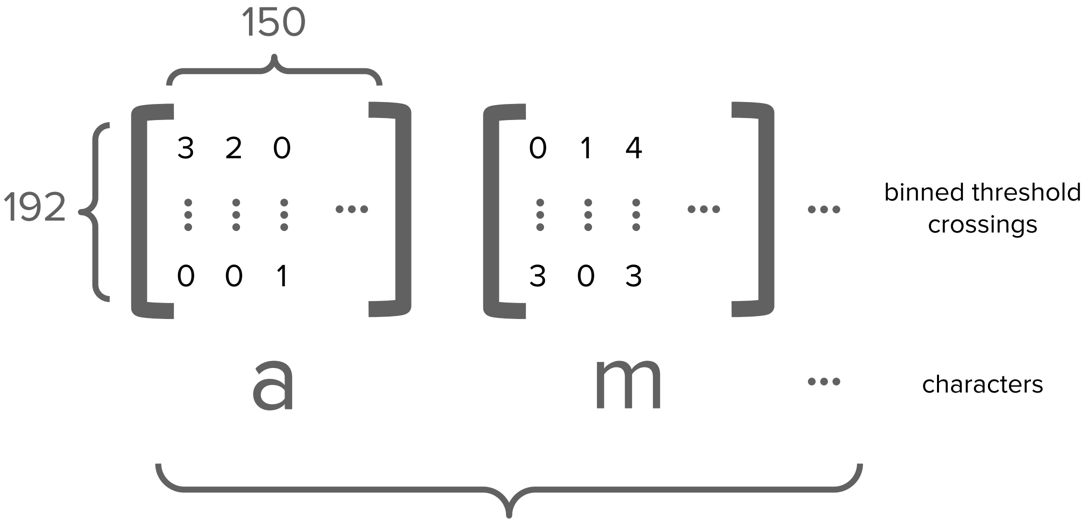
BCI handwriting dataset



Willett et al. 2021



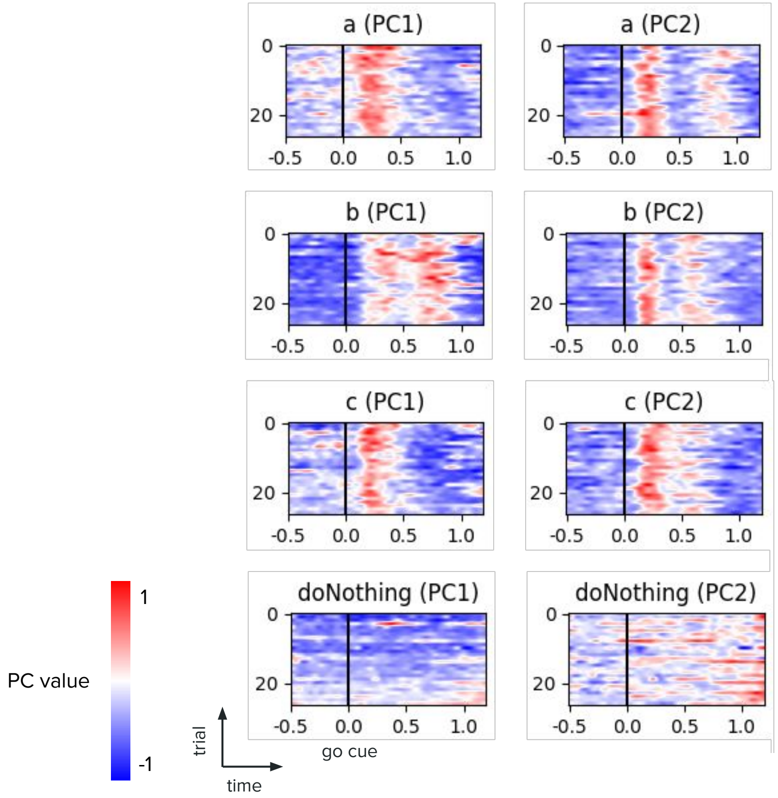
BCI handwriting dataset

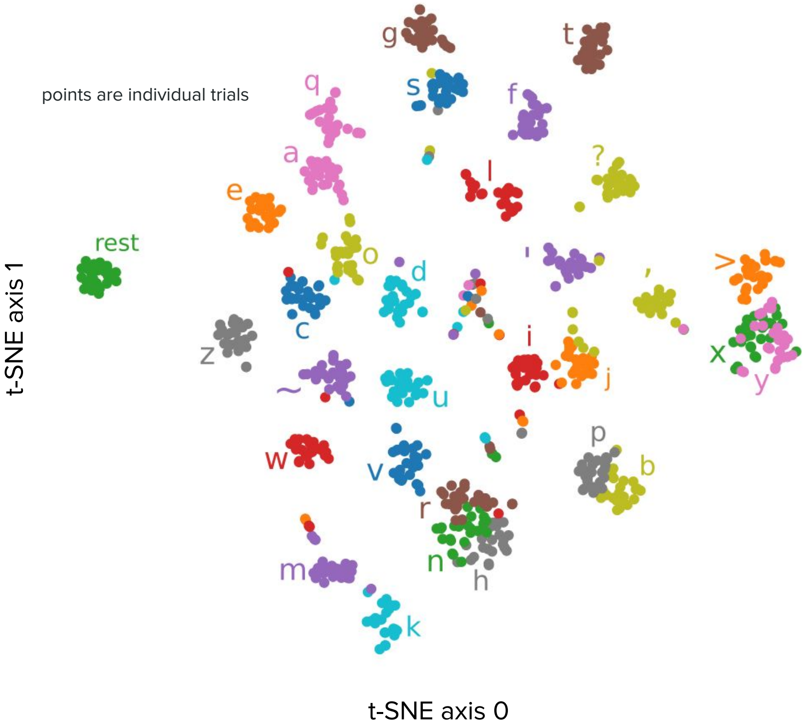


~3600 trials
collected across **10 days**

31 classes
a b c d e f g h i j k l m n o
p q r s t u v w x y z > , ‘ ? ~
balanced

The neural data contain character-specific patterns





Our question: Can deep learning techniques improve on the original study's 94.1% k-nearest neighbors performance?

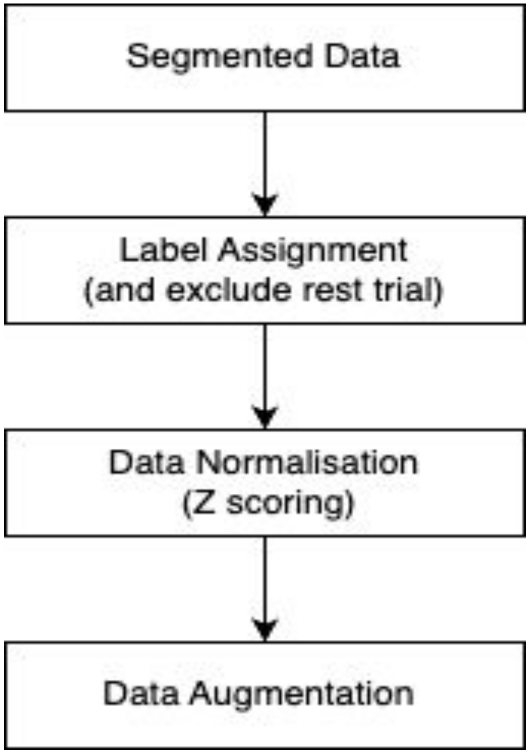
Original k-NN	Our k-NN	Chance	Logistic regression	FFNN	RNN	GRU	LSTM
94.1%	?	~3%	?	?	?	?	?
State-of-the-art	Replication	Baselines		Deep learning			

Data Pre-Processing

Data Labeling: Labelling the data with corresponding character labels (excluding rest trials)

Data Normalization: Applying Z-scoring to the neural data using block-specific means and standard deviations.

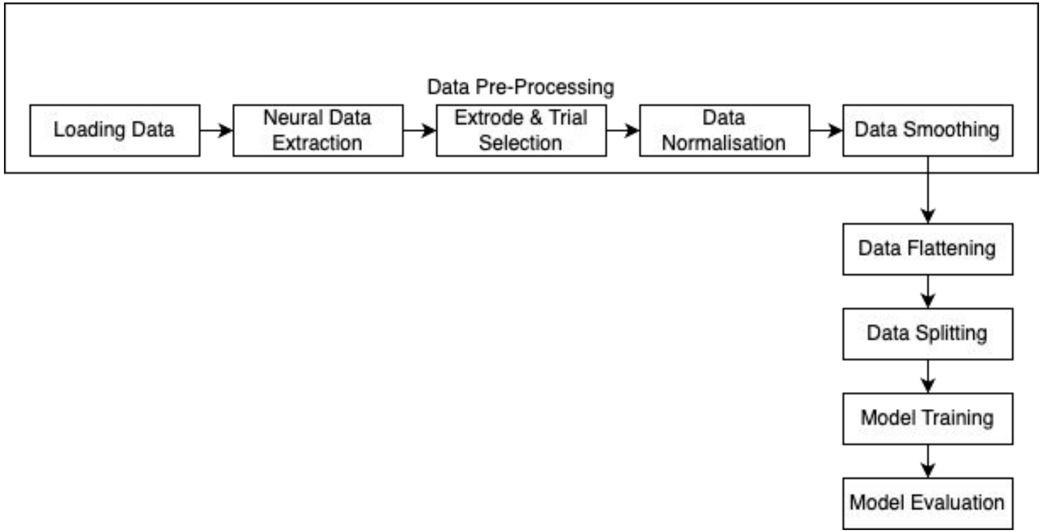
Data Smoothing: Applies Gaussian smoothing to the neural data to reduce noise.



Logistic Regression

- Trained in multiclass setting.
- Reshaped the neural data into 1D array of trial window 28800-length vector.

(192 electrodes x 150 time bins = 28800 features)

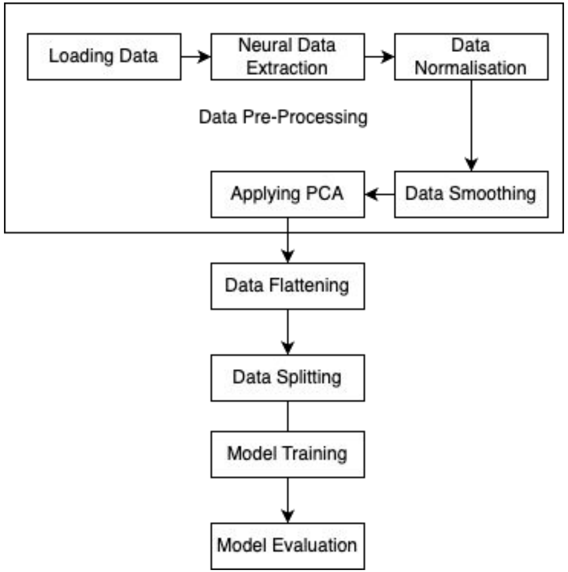


Solver Function: newton-cg solver

k-Nearest Neighbors

- Applies Principal Component Analysis (PCA) to reduce the dimensionality of the data, retaining only the most significant components
- Number of neighbors = 10
- Dimension of each time bin's z-scored threshold crossing vector reduced from 192 to 25
- Trial window flattened to a 3750-length vector

(25 principal components x 150 time bins = 3750 features)



Uses a custom distance metric (dist_with_time_warp()) which accounts for temporal distortions in the data.

Feed Forward Neural Network

Architecture & Training:

- FFNN: Multilayer Architecture
- Implemented Varying Dropout Rates Strategically
- Used SGD Optimizer, Momentum 0.9, LR=0.001
- Training: 20 Epochs
- Varying Learning rate with LR scheduler and early stopping

Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	2,211,968
BatchNorm1d-2	[-1, 128]	256
ReLU-3	[-1, 128]	0
Dropout-4	[-1, 128]	0
Linear-5	[-1, 64]	8,256
BatchNorm1d-6	[-1, 64]	128
ReLU-7	[-1, 64]	0
Dropout-8	[-1, 64]	0
Linear-9	[-1, 32]	2,080

Total params: 2,222,688
Trainable params: 2,222,688
Non-trainable params: 0

RNN

Architecture & Training:

HIDDEN_SIZE: Set to 512 for the size of the RNN hidden layers.

NUM_LAYERS: Set to 2 for the number of stacked RNN layers.

DROPOUT: Set to 0.5, indicating a high level of dropout regularization.

Optimizer: Stochastic Gradient Descent (SGD), Learning rate 0.0005, momentum 0.99

WEIGHT_DECAY: Set to 0.0001, adding a regularization, prevents overfitting.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
               RNN-1  [[-1, 150, 512], [-1, 2, 512]]
            Linear-2                   [-1, 31]       2,380,831
================================================================
Total params: 2,380,831
Trainable params: 2,380,831
Non-trainable params: 0
----------------------------------------------------------------
```

Data Augmentation: NOISE_STDDEV: Set to 0.1, defining the standard deviation of the noise added for data augmentation. **Gaussian noise** added to the training data every epoch as intended

LSTM & GRU

Architecture & Training:

HIDDEN_SIZE: Set to 512 for the size of the RNN hidden layers.

NUM_LAYERS: Set to 2 for the number of stacked RNN layers.

DROPOUT: Set to 0.5, indicating a high level of dropout regularization.

Optimizer: Stochastic Gradient Descent (SGD), Learning rate 0.0005 - 0001, momentum 0.99

```
=================================================================
Layer (type:depth-idx)              Param #
=================================================================
├─LSTM: 1-1                         3,547,136
├─Linear: 1-2                       2,380,831
=================================================================
Total params: 5,927,967
Trainable params: 5,927,967
Non-trainable params: 0
=================================================================
```

LSTM

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
             GRU-1  [[-1, 150, 512], [-1, 2, 512]]
          Linear-2                   [-1, 31]       2,380,831
================================================================
Total params: 2,380,831
Trainable params: 2,380,831
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.11
Forward/backward pass size (MB): 600.00
Params size (MB): 9.08
Estimated Total Size (MB): 609.19
----------------------------------------------------------------
```

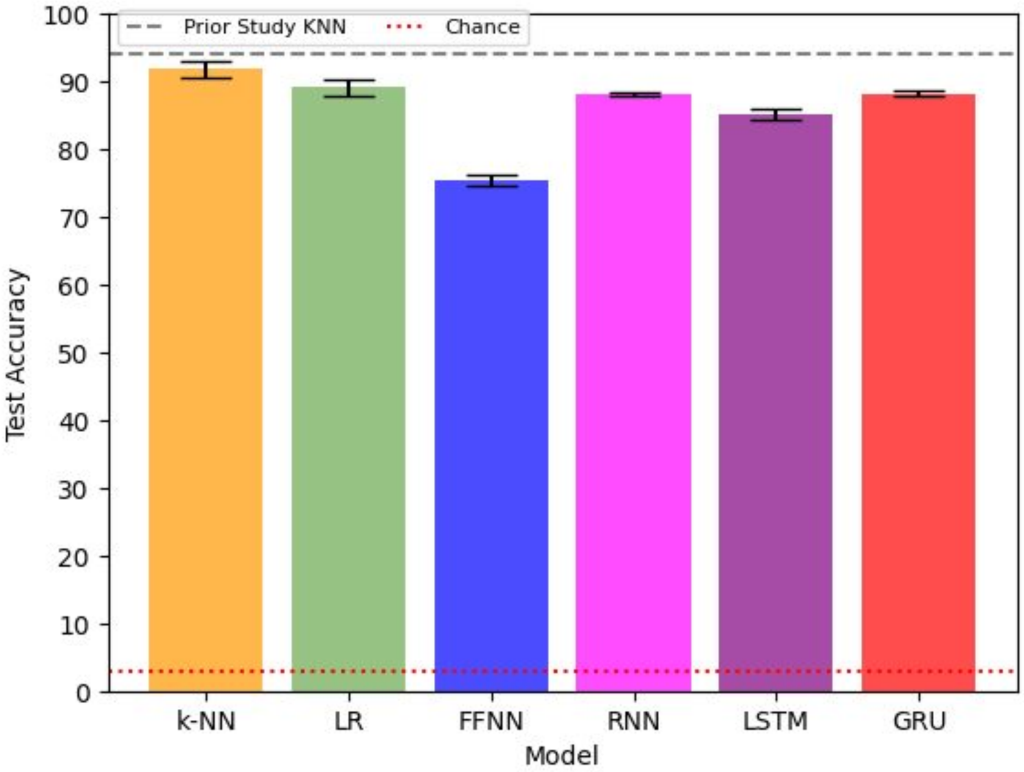
GRU

Model Evaluation

- Benchmarked against the prior study's RNN and chance

- k-NN outperformed Neural Networks

- RNNs got very close…

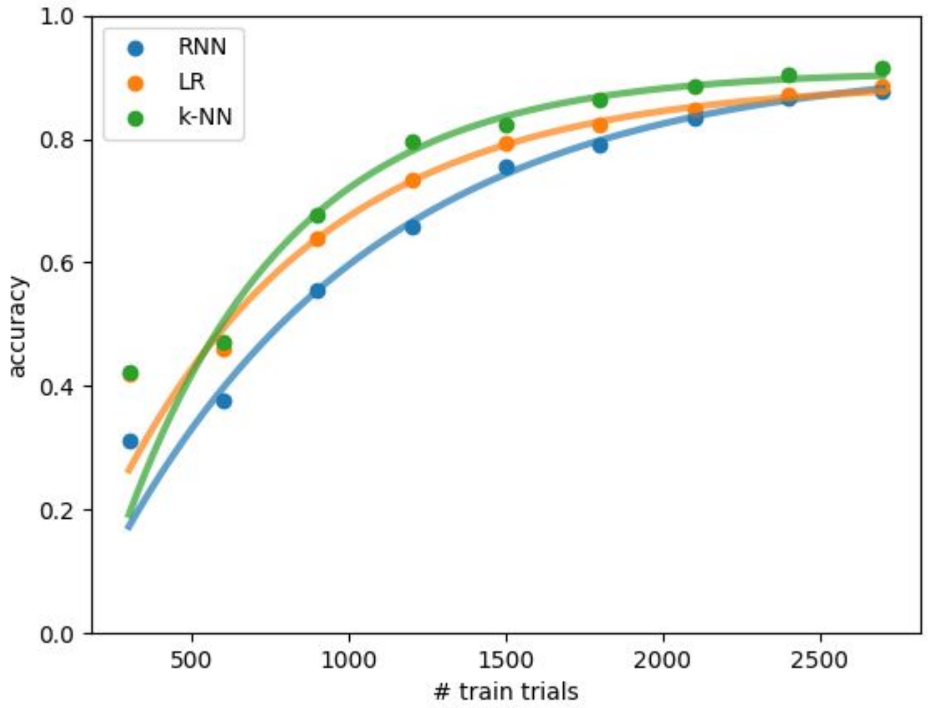


Training Trials Comparison

- Varied the number of training trials:
    - 300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700

- RNN seems to be growing at a faster rate and might plateau later

- Increasing number of training trials could lead to better results

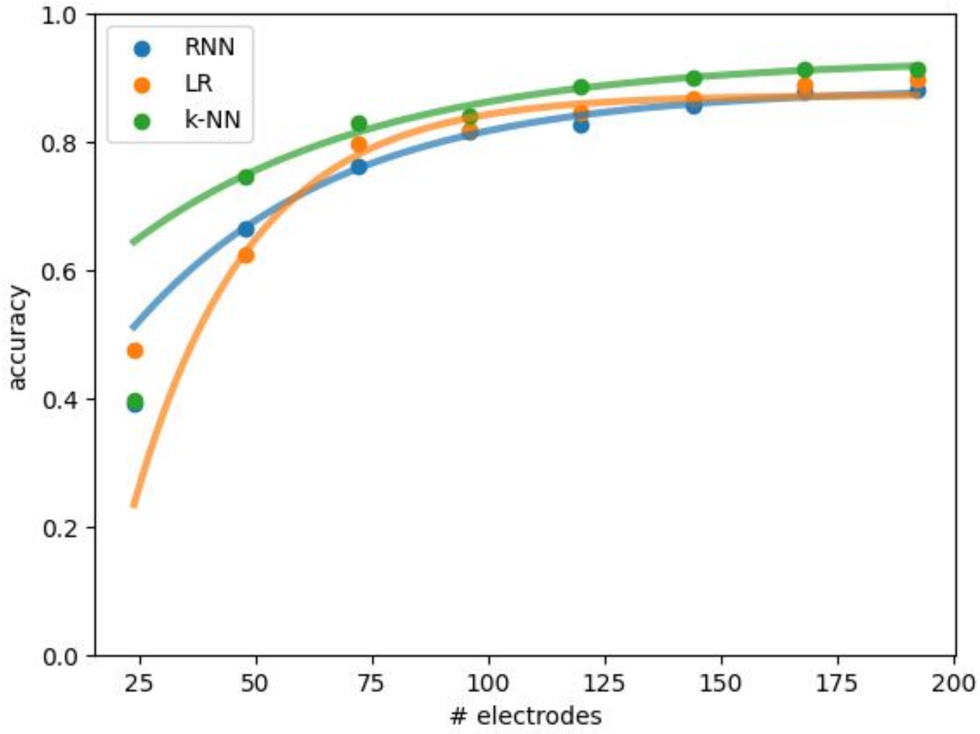


Electrodes Comparison

- Varied the number electrodes:
  - 24, 48, 72, 96, 120, 144, 168, 192

- Not super clear if RNN is taking more advantage of additional electrodes

- Increasing number of electrodes could lead to better results



Acknowledgements

- Dataset and many methods: *High-performance brain-to-text communication via handwriting*, Willett et al. 2019.
- BrainGate participant image: *Neuronal ensemble control of prosthetic devices by a human with tetraplegia,* Hochberg et al. 2006.
- Brain and motor neuron image: https://www.physio-pedia.com/Motor_Neurone
- Utah array image: https://www.medicaldesignandoutsourcing.com

Thank you for listening!

Any questions?