Day Schedule

~~8:30 AM - 9:30 AM: Breakfast~~

9:45 AM - 10:45 AM: Introductory Lecture

10:45 AM - 11:00 AM: Bio Break

11:00 AM - 12:00 PM: Philosophy and General Questions

12:15 PM - 1:15 PM: Lunch

1:30 PM - 2:30 PM: Methods Lecture

2:30 PM - 6:00 PM: Hands On Session

Encoding and decoding models of neural activity

Max Melin and Gabriel Rojas-Bowe

July 12, 2023

Outline

1. Motivation:
    a. What are the purpose of encoding and decoding models?
    b. How can they be used to understand neural activity?
2. Background:
    a. History of linear regression and GLMs
    b. Current applications

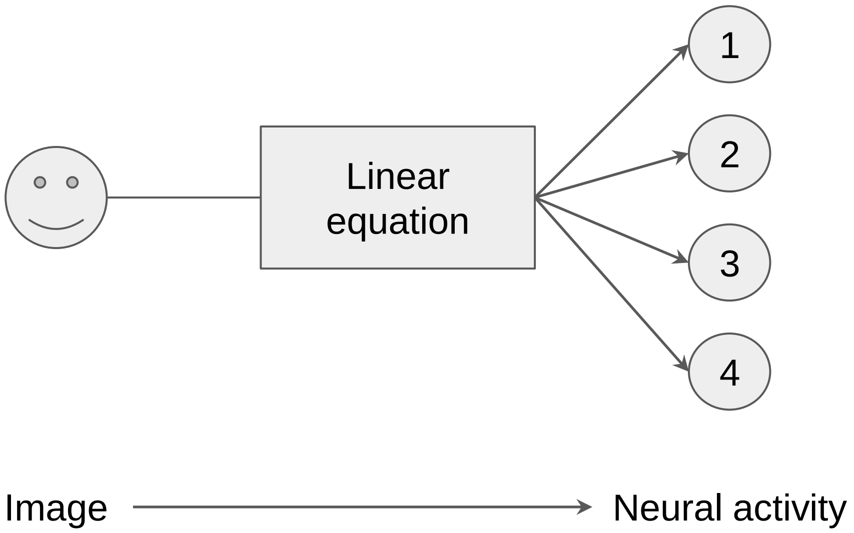
What are the purpose of encoding and decoding models?

- Encoding models are used to understand how behavioral and task variables influence neural activity

- Decoding models are used to understand how behavioral and task variables can be extracted from neural activity

How do they work?

Encoding models

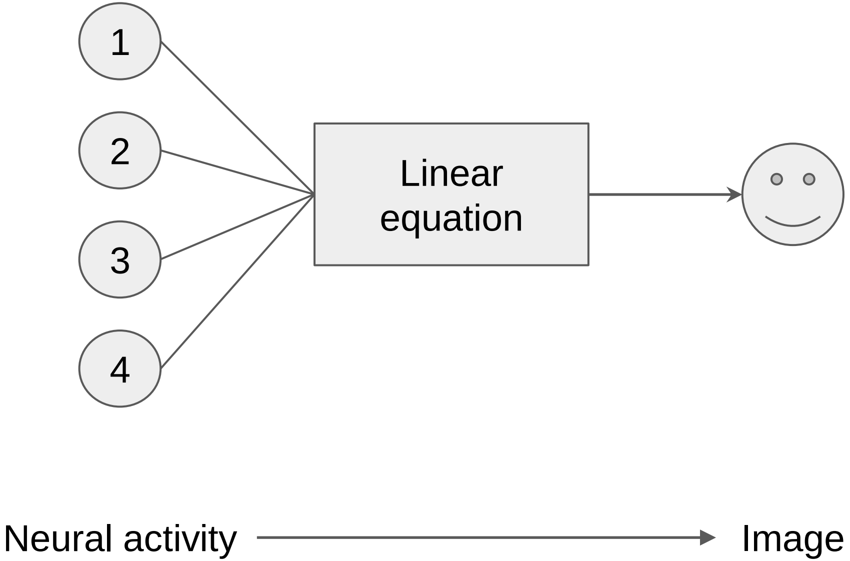


Can we predict the neural activity with just the smiley face?

Encoding model examples:

- Linear encoding models
  - Simple linear regression
  - Polynomial regression
  - Regression with regularization: **ridge regression** and lasso regression
- Nonlinear encoding models (out of scope for this workshop)
  - Generalized linear models (GLMs)
  - Neural networks

Decoding models



Can we predict if the image is a smiley face with just neural activity?

Decoding model examples:

- Support vector machines (SVMs)
- Logistic regression
- Random forest classifiers
- Hidden Markov models (HMMs)

How do you decide which model to use when analyzing your data?

depends on the question…

Encoding models

Pros:

- Allow you to separate the contributions of different variables to neural activity
- Can identify which variables are most important for driving neural activity
- Can use more complex datasets with many features

Cons:

- Struggle with interpreting variables that are related to each other (collinearity)
- Rely on simplifying assumptions such as assuming each response variable behaves independently of one another

Decoding models

Pros:

- Can be used to determine whether information is present in a brain area or not
- Use multivariate noise to their advantage (noise present in many neurons/voxels/etc.)
- Enable the development of brain-machine interfaces which are becoming increasingly important in healthcare
- Can be used for clinical diagnosis

Cons:

- Cannot be used as models of brain processing (specially in sensory systems since they operate in the inverse direction)
- Do not generally provide information about computational mechanisms
    - You can see if information is present, but not necessarily understand how it was transformed from a neural representation to a behavioral outcome

For this workshop, we are interested in understanding how behavioral variables and task events shape neural activity, so we will focus on **encoding models**

Some history on linear regression

- Based on the method of least squares, or minimum squared error
  - Measures how well a line fits to data by calculating the mean squared distance between the observed value and the value predicted by the linear model
  - First published by Legendre (1805) to predict the orbits of comets
- Generalized linear models, or GLMs, function similarly to linear regression models, but incorporate nonlinearities to capture more complex relationships between input and output variables
  - For example, a linear filter, coupled to an exponential nonlinearity, followed by a Poisson process (i.e. an LNP model) can allow you to model instantaneous firing rates for neurons (see Pillow et al., 2008, https://doi.org/10.1038/nature07140)

Current applications for linear encoding models: Musall and Kaufman et al., 2019

Article | Published: 24 September 2019

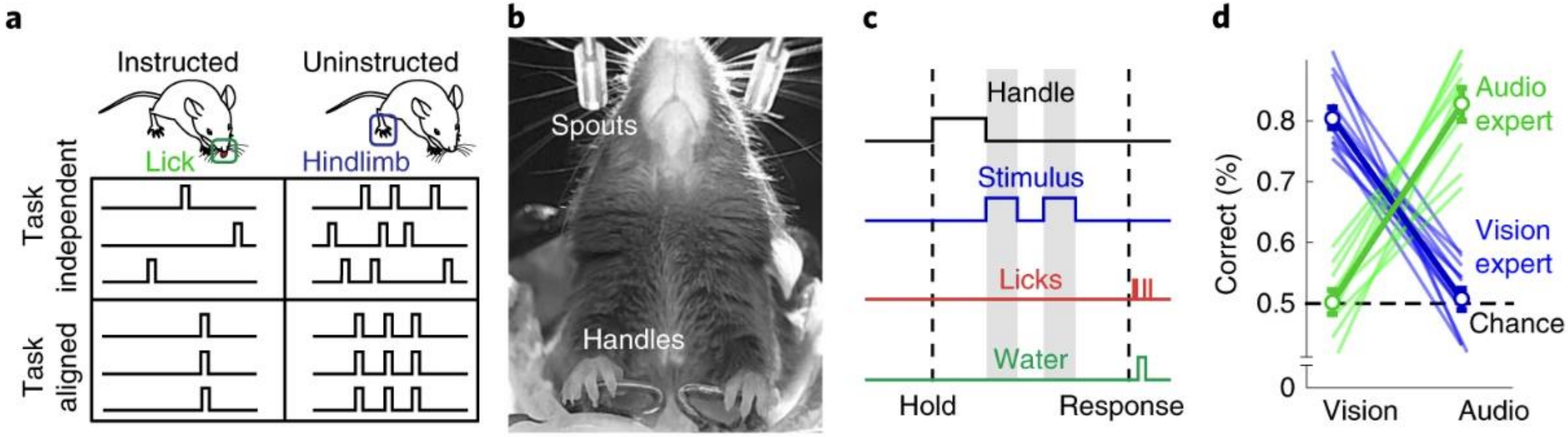
Single-trial neural dynamics are dominated by richly varied movements

Simon Musall, Matthew T. Kaufman, Ashley L. Juavinett, Steven Gluf & Anne K. Churchland

Nature Neuroscience **22**, 1677–1686 (2019) | Cite this article

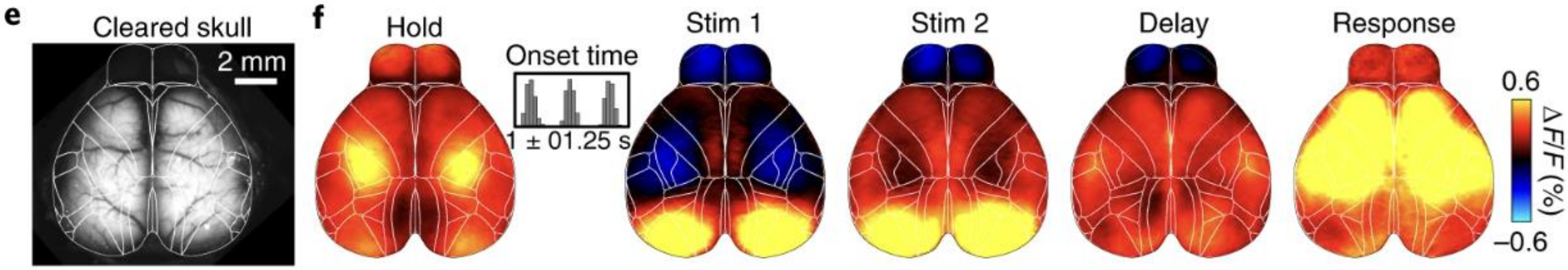
Single-trial neural dynamics are dominated by richly varied movements

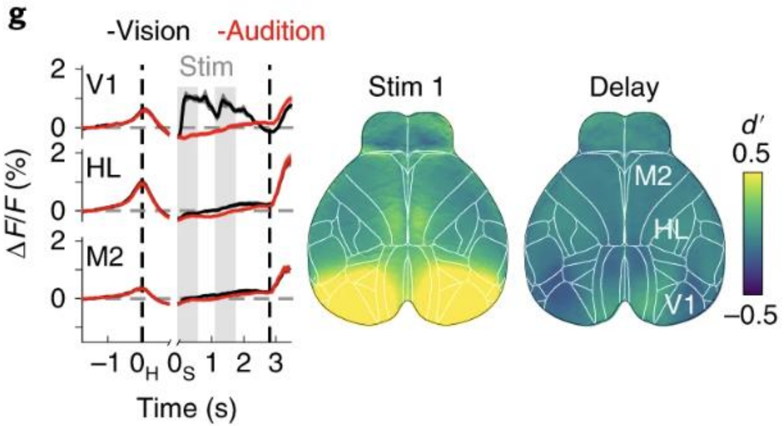
Question and task structure



Single-trial neural dynamics are dominated by richly varied movements

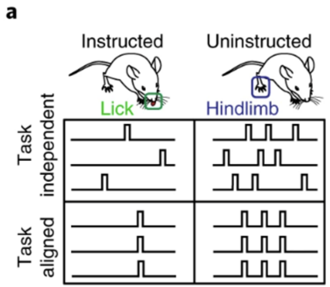
<u>Wide-field imaging for recording cortex-wide neural dynamics</u>





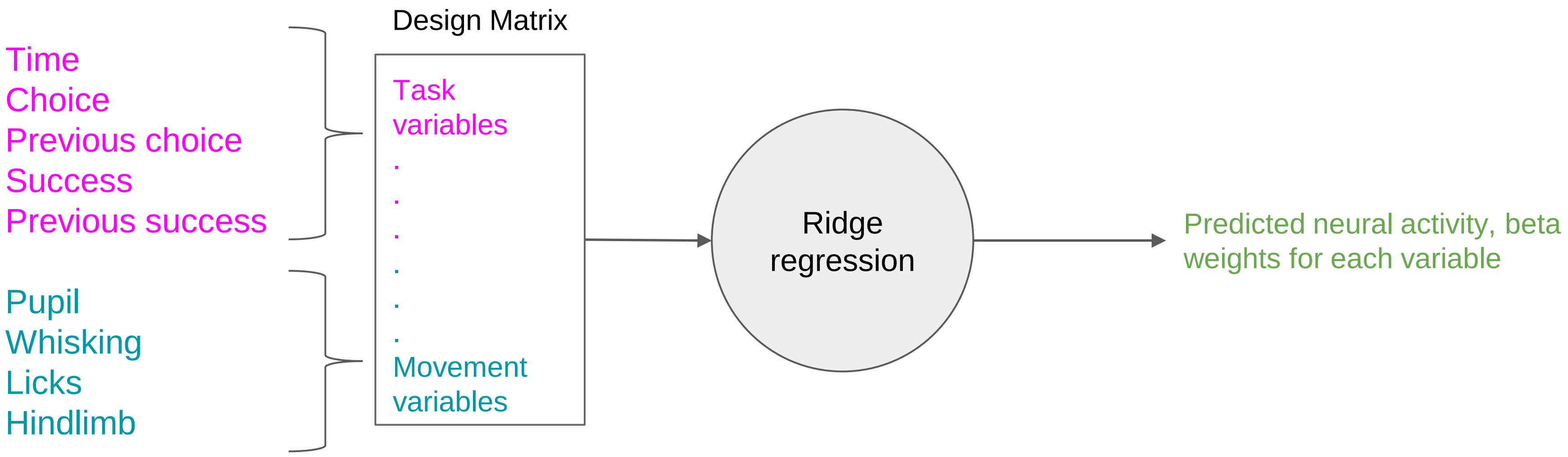
How do we determine the impact of movements on neural activity?

A linear encoding model!



Single-trial neural dynamics are dominated by richly varied movements

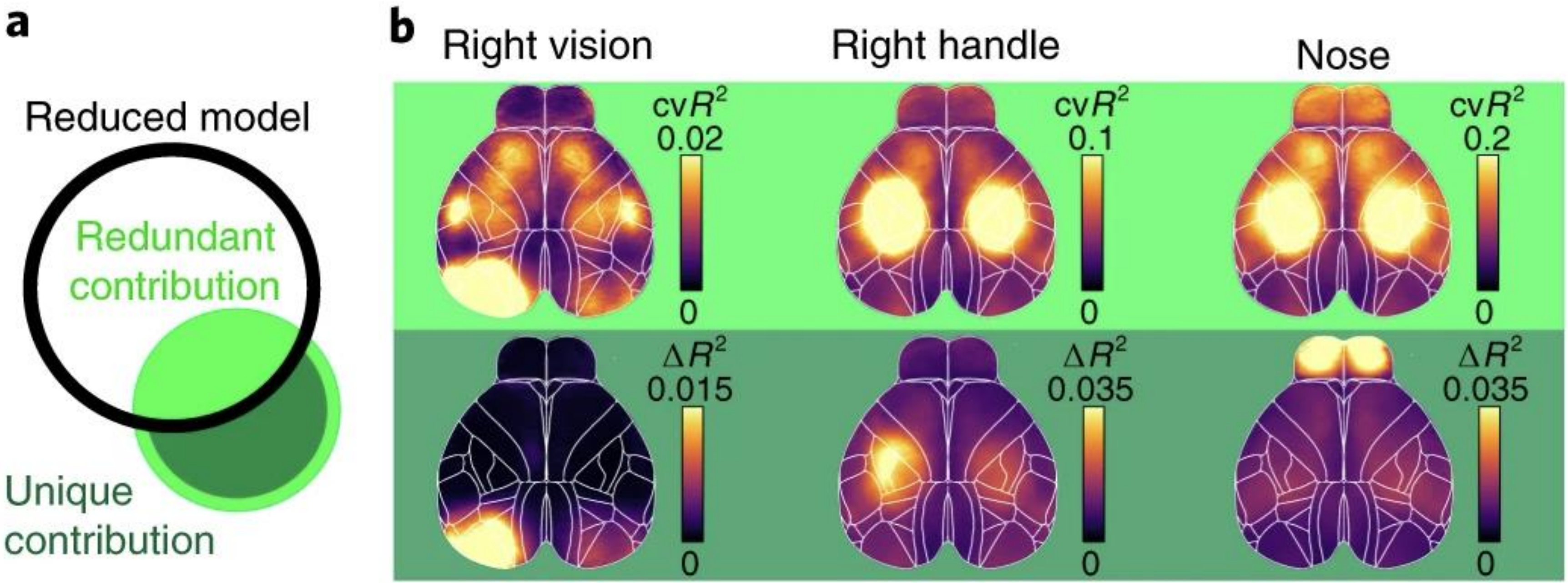
A Ridge regression model



(we'll go into the math in the Methods section later today)

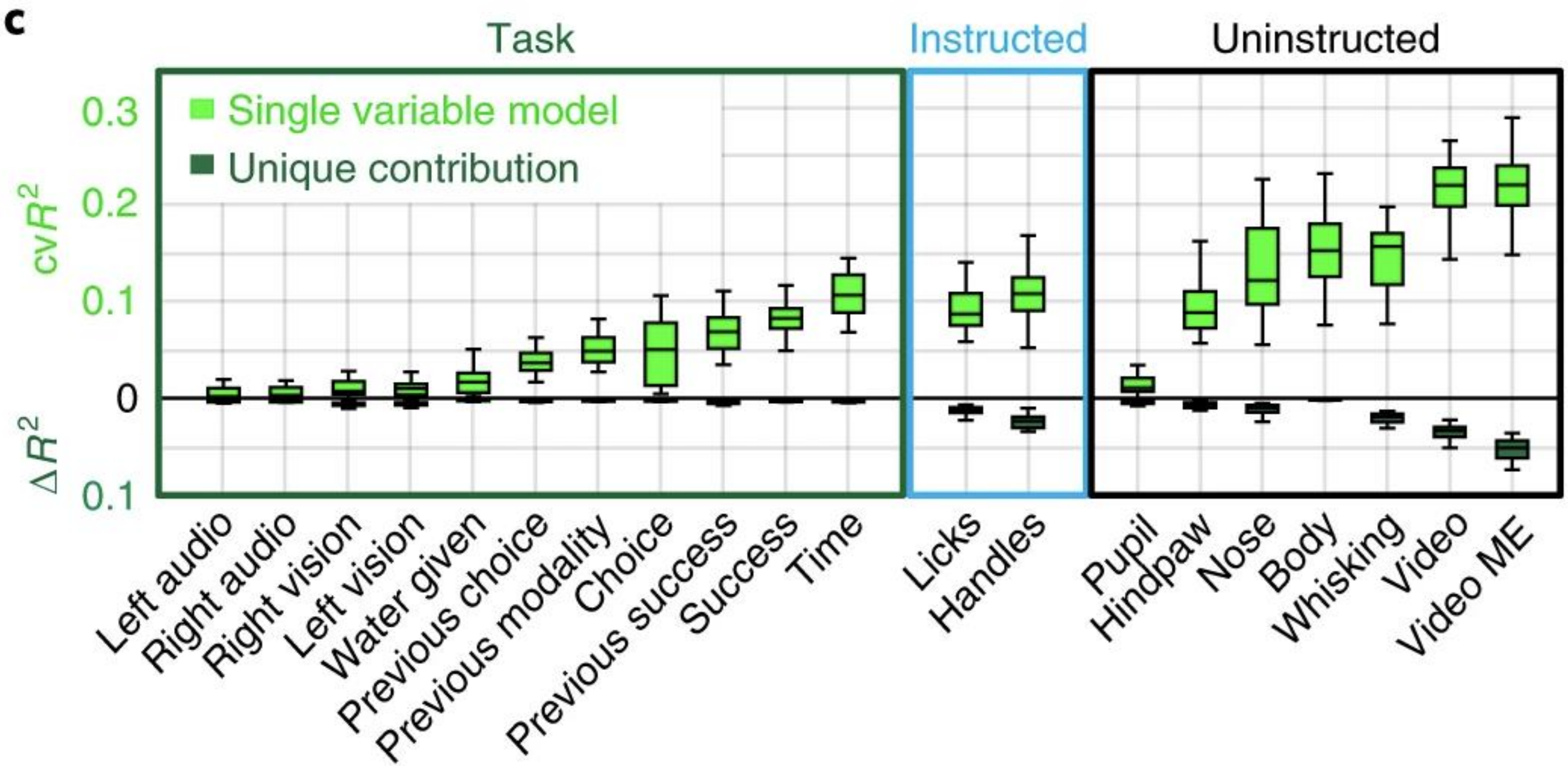
Single-trial neural dynamics are dominated by richly varied movements

<u>Which variables are most important for the model's success?</u>



Single-trial neural dynamics are dominated by richly varied movements

Which variables are most important for the model's success?



That's it for the introduction!

In the Methods lecture we will go over the following topics:

- Basics of regression
- Fitting and cross-validation
- Model outputs and visualization

Day Schedule

~~8:30 AM - 9:30 AM: Breakfast~~

~~9:45 AM - 10:45 AM: Introductory Lecture~~

10:45 AM - 11:00 AM: Bio Break

11:00 AM - 12:00 PM: Philosophy and General Questions

12:15 PM - 1:15 PM: Lunch

1:30 PM - 2:30 PM: Methods Lecture

2:30 PM - 6:00 PM: Hands On Session

Day Schedule

~~8:30 AM - 9:30 AM: Breakfast~~

~~9:45 AM - 10:45 AM: Introductory Lecture~~

~~10:45 AM - 11:00 AM: Bio Break~~

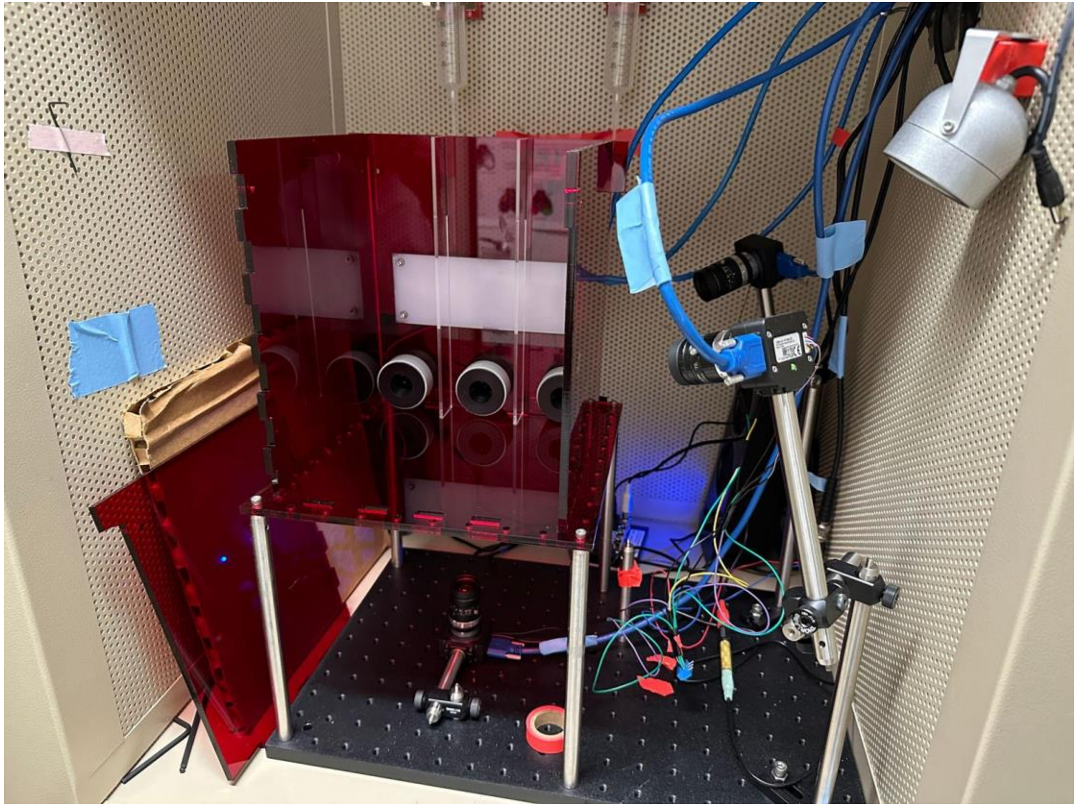
11:00 AM - 12:00 PM: Philosophy and General Questions

12:15 PM - 1:15 PM: Lunch

1:30 PM - 2:30 PM: Methods Lecture

2:30 PM - 6:00 PM: Hands On Session

The Dataset



Data collected by Gabriel Rojas-Bowe and Lukas Oesch in the Churchland Lab

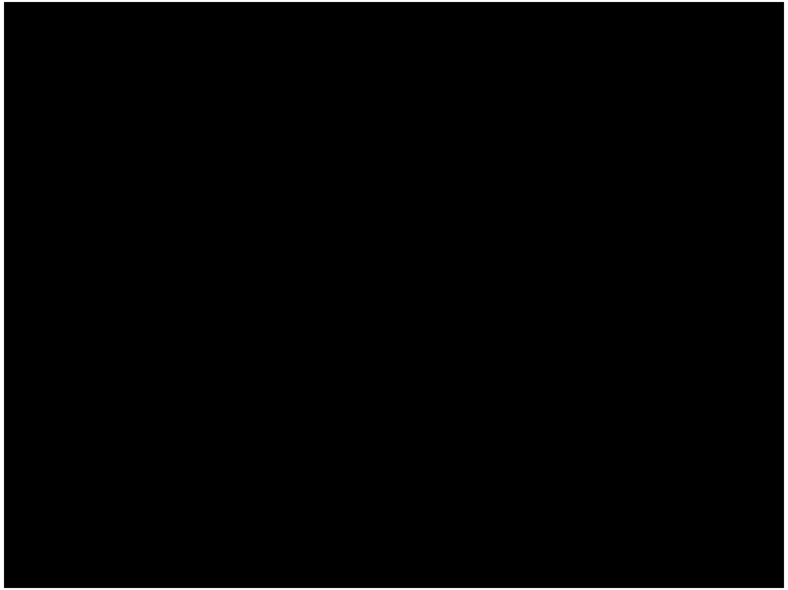
The mouse must learn to do the following

1. Poke nose into center port to initiate trial

1. Visual or auditory stimulus is delivered

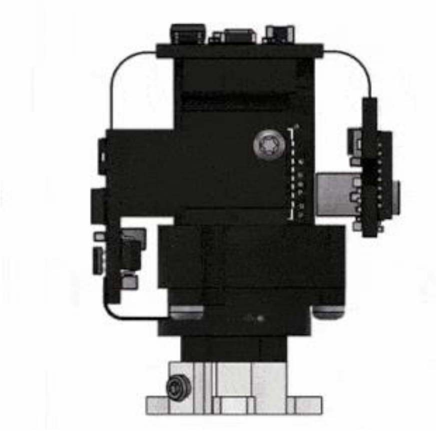
1. Depending on the rate of the stimulus (whether it is above or below the category boundary of 12 Hz), the mouse must poke at the left or right spout to receive a water reward

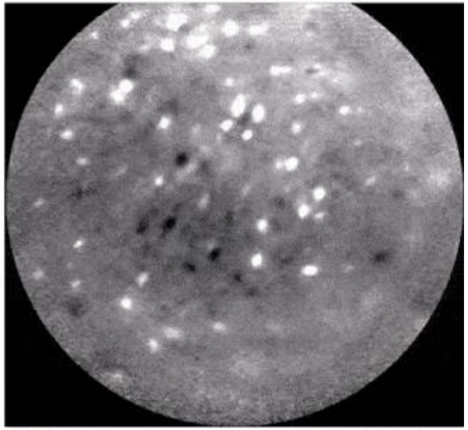
The Dataset



The Dataset

1-photon calcium imaging was performed with the UCLA miniscope V4





Courtesy of miniscope.org

Day Schedule

~~8:30 AM - 9:30 AM: Breakfast~~

~~9:45 AM - 10:45 AM: Introductory Lecture~~

~~10:45 AM - 11:00 AM: Bio Break~~

~~11:00 AM - 12:00 PM: Philosophy and General Questions~~

12:15 PM - 1:15 PM: Lunch

1:30 PM - 2:30 PM: Methods Lecture

2:30 PM - 6:00 PM: Hands On Session

Day Schedule

~~8:30 AM - 9:30 AM: Breakfast~~

~~9:45 AM - 10:45 AM: Introductory Lecture~~

~~10:45 AM - 11:00 AM: Bio Break~~

~~11:00 AM - 12:00 PM: Philosophy and General Questions~~

~~12:15 PM - 1:15 PM: Lunch~~

1:30 PM - 2:30 PM: Methods Lecture

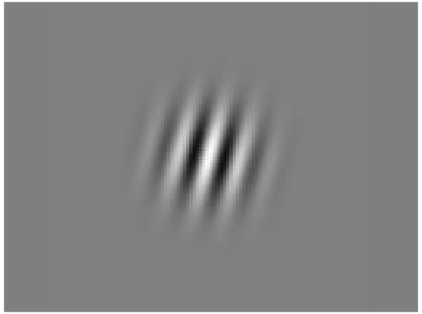
2:30 PM - 6:00 PM: Hands On Session

Methods for neural encoding models

Contents

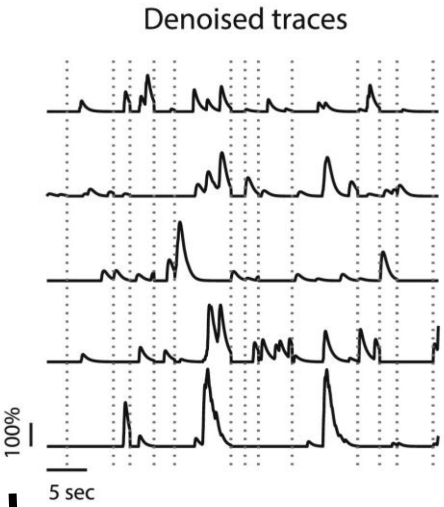
1. Preparing neural data
2. Building a design matrix
3. Basics of regression
4. Regularization and ridge regression
5. Preprocessing for regression
6. Fitting and cross-validation
7. Model outputs and visualization
8. Implementation in Python

A brief reminder of our goal… to identify how variables of interest modulate neural activity



X

$* \ \beta \ =$



Y

Y can be calcium data, ephys, fMRI, etc.

$$X\beta = \boxed{Y}$$

First, decide on the alignment scheme

This impacts all elements of your design matrix, because predictors need to be properly aligned to the neural data

- Do any of the necessary smoothing, binning, etc.
- “Trialize” the data or no?
- If you are going to split the data by trials, what do you plan to align it to?

- Ultimately the matrix Y needs to be reshaped to the size

  [n_samples, n_neurons/voxels/dimensions/etc.]

$$\boxed{X}\beta = Y$$

Building the design matrix

- Often the most time consuming part of building these models
- Highly specific to your dataset/task
- **Will have a larger impact on your results and interpretation than any minor variations in fitting procedure**

Building the design matrix

There are multiple regressor types we can include in the design matrix

Building the design matrix

There are multiple regressor types we can include in the design matrix

1. "Analog" regressors
   a. A continuous time series (eg. pupil diameter, DLC label position)

Building the design matrix

There are multiple regressor types we can include in the design matrix

1. “Analog” regressors
   a. A continuous time series (eg. pupil diameter, DLC label position)
2. Binary regressors
   a. Usually one value per trial (choice, stimulus side, etc.)
   b. How do we expand these from trials to time?

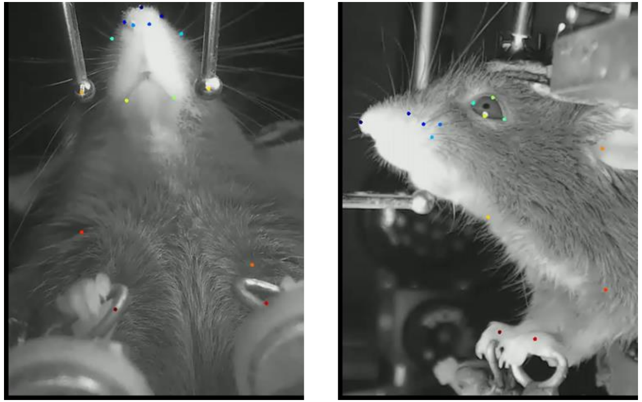
Building the design matrix

There are multiple regressor types we can include in the design matrix

1. “Analog” regressors
    a. A continuous time series (eg. pupil diameter, DLC label position)
2. Binary regressors
    a. Usually one value per trial (choice, stimulus side, etc.)
    b. How do we expand these from trials to time?
3. Trial events
    a. Occur at specific times over the duration of a trial
    b. Can be aligned across trials (stimulus onset) or unaligned (spout lick or blinking)

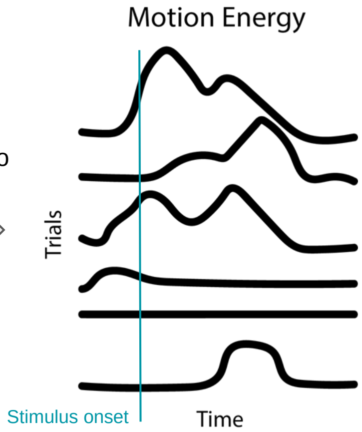
1. Analog regressors

DLC motion energy of various body parts



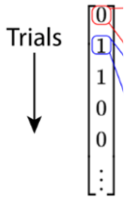
Musall *et al.* 2019

Resample and align to neural data



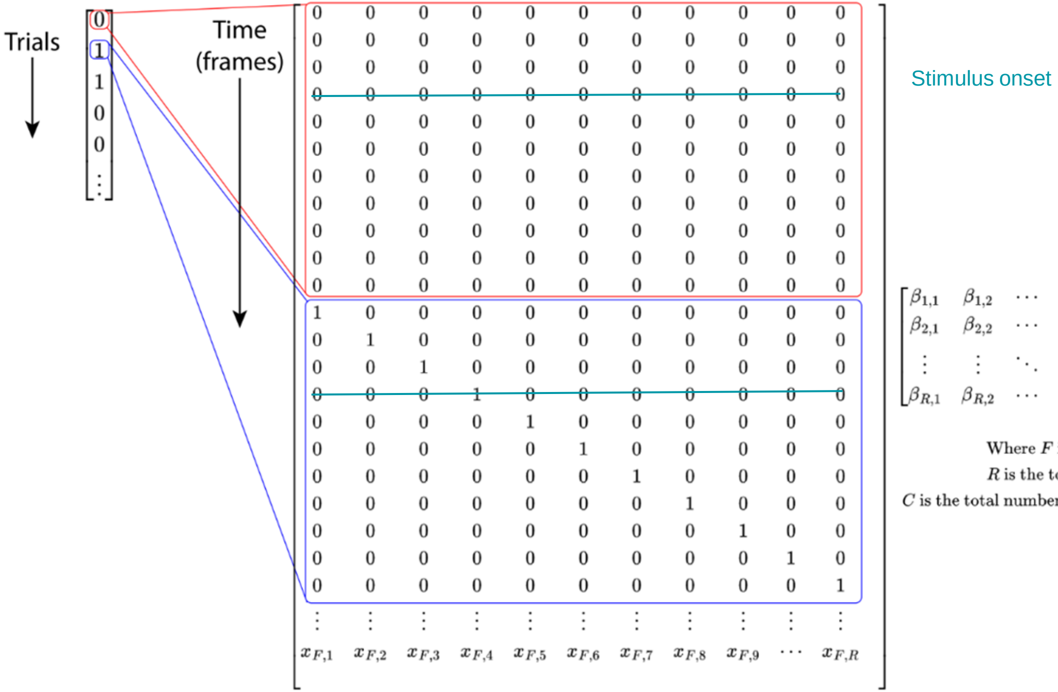
Reshape

2. Binary Regressors - left vs right choice



3. Trial events

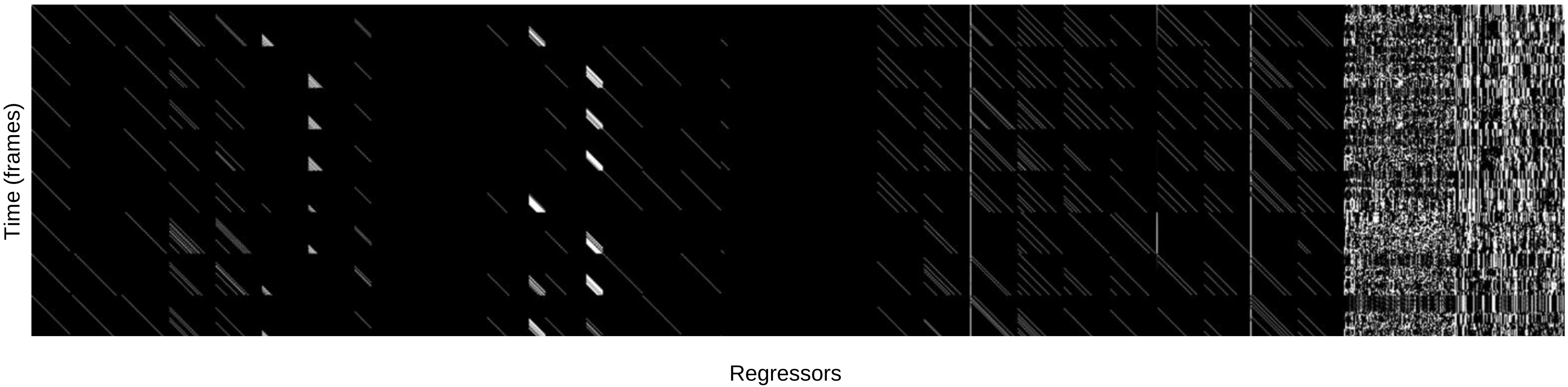
- Eg. licking or whisking
- Similar to binary regressors, but these will be shifted in time depending on when they occurred
- Define a pre_time and a post_time
    - This is the maximum time before and after the event that can be used to predict neural activity
    - Pre_time for stimuli should be 0, if a stimulus wasn’t yet delivered, how can it explain neural activity?



$$\begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,C} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{R,1} & \beta_{R,2} & \cdots & \beta_{R,C} \end{bmatrix} = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,C} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ y_{F,1} & y_{F,2} & \cdots & y_{F,C} \end{bmatrix}$$

Where F is number of frames in a session
R is the total number of linear predictors
C is the total number temporal widefield components to predict

Assemble the analog, binary, and event regressors into one large matrix



$$X\boxed{\beta} = Y$$

Fitting the model

- **1_Ridge_Regression.ipynb**
  - Will cover the basics of ordinary least squares and ridge regression. We will also motivate why ridge regression is necessary.
- **2_Fitting_Neural_Data.ipynb**
  - Use scikit-learn to fit an encoding model to calcium data
- **3_Design_Matrix.ipynb**
  - Manipulate the design matrix to get the neural variance explained by different variables