Regression Model to predict ophys data

Our goal is to build a GLM to predict neural activity during visual behavior. We'd like a model that applies to all sessions (active, passive, trained, novel), all cell types (SlC, VIP, SST, and others), all brain regions and depths (V1, AM, LM, etc), and all behavioral states (engaged, disengaged, timing or task motivated).

Basic Design

Consistent with recent results in the literature, we will predict the df/f signal from each cell using a gaussian noise model, meaning a simple mean-squared error loss function between the model's prediction and the data, and a linear link function. The df/f signal is generated at a 31Hz rate, so we will use $1/31 = 0.032$ s time bins for scientifica data. The model prediction is the linear sum of kernels that are convolved with external regressors. Some of these regressors are discrete task events, others are continuous. Each kernel is defined as a vector of weights. For neuron i with m regressors and df/f signal y_i, the model is:

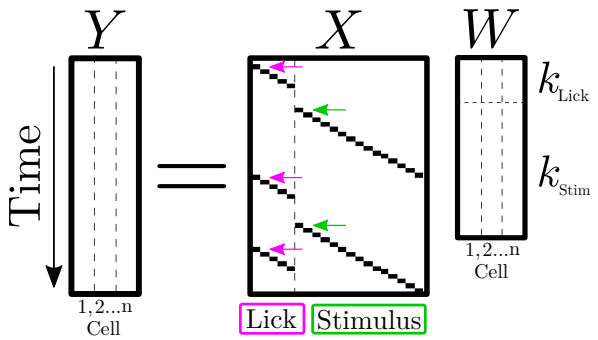
$$y_i = \sum_j^m k_{i,j} * x_j(t). \tag{1}$$

Since the discrete time convolution is a linear time-invariant operator, we can write the model as an ordinary least squares regression problem with a toeplitz design matrix. The key step is the construction of the design matrix X which has a banded structure to define the time lags of each regressor. Let w_i be a vector containing the concatenation of all the individual kernels $k_{i,j}$ for neuron i, then the model is equivalently written as:

$$y_i = X * w_i. \tag{2}$$

In general we can think about fitting a session with n neurons, t timebins, and m kernels. Each kernel can have a different length, but lets say their combined length is z. We can then define: $Y \in (t \times n)$ a matrix containing the df/f signal for all the neurons, $X \in (t \times z)$ a toeplitz matrix containing the external events with the time-lags creating the banded structure, and $W \in (z \times n)$ a matrix containing all the of kernel weights for all the neurons. The advantage of this model is that we can write the analytical solution.

$$Y = XW \tag{3}$$
$$X^T Y = X^T X W \tag{4}$$
$$(X^T X)^{-1} X^T Y = (X^T X)^{-1} X^T X W \tag{5}$$
$$W = (X^T X)^{-1} X^T Y \tag{6}$$



Basic data

We'll start with just using full trials so we dont need to think about premature licks.

Future Improvements

- How to model active/passive, trained/Novel images
- Fit one model per session, or fit all sessions jointly for tracked cells?
- Omission Kernel
- Image specific change kernel
- Specific kernels for 2nd, 3rd, etc repetition of each image
- Behavioral response kernels for licking, rewards, running, or pupil diameter
- Using the time varying model weights as regressors

1 2019-12-19 Initial model fit to SFN manifest

We fit a basic model to a bunch of sessions in the SFN manifest to get an idea for how things are working. This model included a kernel for each image presentation (including omissions) and a separate kernel aligned to each change.

1.1 Model at this time

- One param per ophys frame (31Hz for scientifica)
- One kernel for each image (9 total), 30 params long
- One kernel aligned to changes, 100 params long
- No cross validation, no regularization, no cell scaling
- Fitting each session separately

1.2 Results

We find that we can predict up to 80% of the dF/F variance for some cells. On average, this model predicts 14.4% of the variance for cells that are classified as 'responsive' according to the visual coding definition (>25% of flashes have p<0.05 when compared to shuffled spontaneous distribution). We predict on average just 2.5% of variance in cells classified as 'non-responsive'. However, there are clearly a number of 'non-responsive' cells for which we can explain a sizeable portion of the variance with this model, suggesting that the visual coding responsiveness metric may be biased towards selecting for cells with specific response profiles.

1.3 Initial Cross Validation

1.4 Ridge Regression

We next added L2 ridge regression, and evaluated the fit on cross validation. We can find the solution analytically with:

$$w = (X^T X + \lambda I)^{-1} X^T Y \tag{7}$$

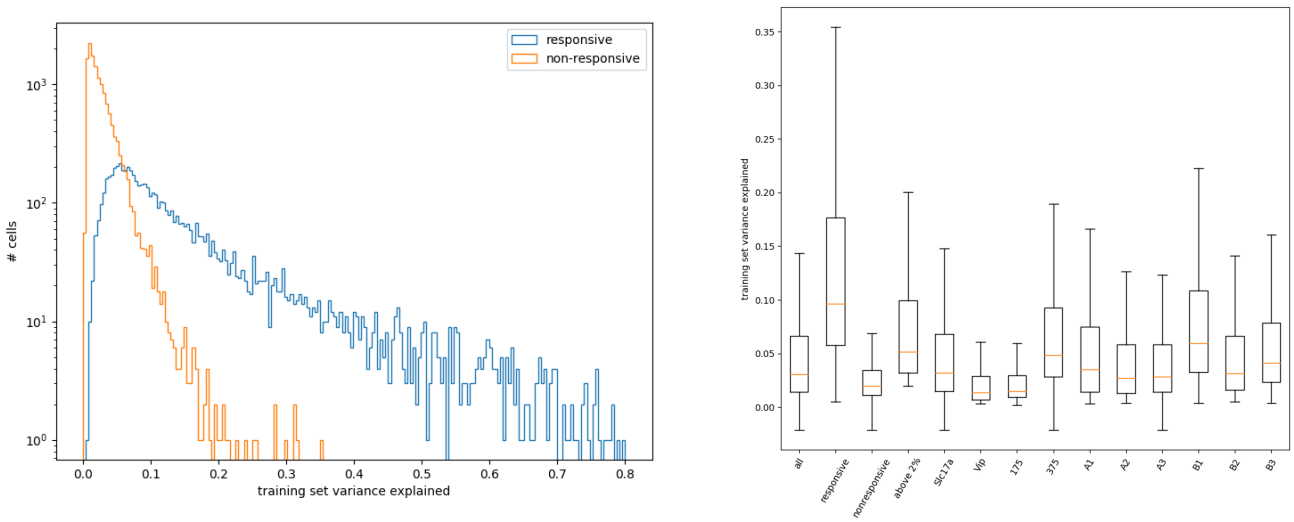


Figure 1: (Left) The model explains more variance on average for neurons that are classified as responsive using the visual coding definition. (Right) The model fits neurons from some conditions (such as neurons from 375um) better than others.

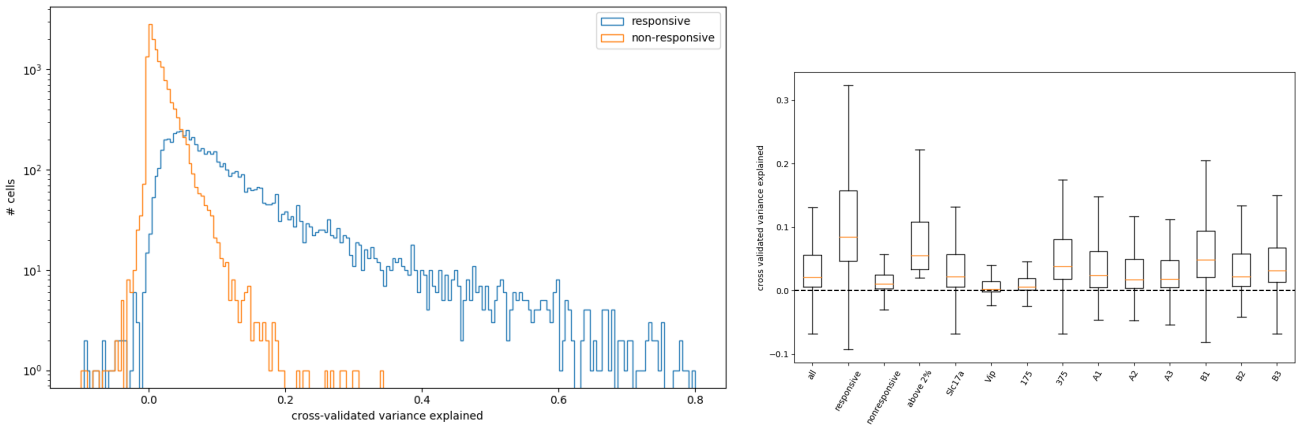


Figure 2: Same as above, but using 6 fold cross validation. Metrics shown were computed on the test folds

Where $\lambda = 1/\theta$, where θ is the variance of the prior. We can find λ by using cross validation.

One design question is whether we want to use the same λ for each session, or pick the best lambda for each session separately. Right now, we aren't normalizing each cell's responses, so sessions with more active cells are going to be impacted by regularization more. We did the cross-validation grid search for all sessions, and found a λ of 70 to be reasonable for more cells. We need to revisit this parameter once we add more filters, and scale each cell.

1.5 Comparison between Regularization Types

Make a plot of variance explained distributions for several different methods

1.6 Model Evaluation

Having fit a model with regularization we are left with the question of how well does the model describe the data. So far we have been using variance explained. Following Steinmetz, 2019, we can use a shuffle analysis to determine a threshold of variance explained that is meaningful. For each cell, I shuffled the image labels, thus preserving the flash-by-flash response dynamics, but scrambling the image identity.

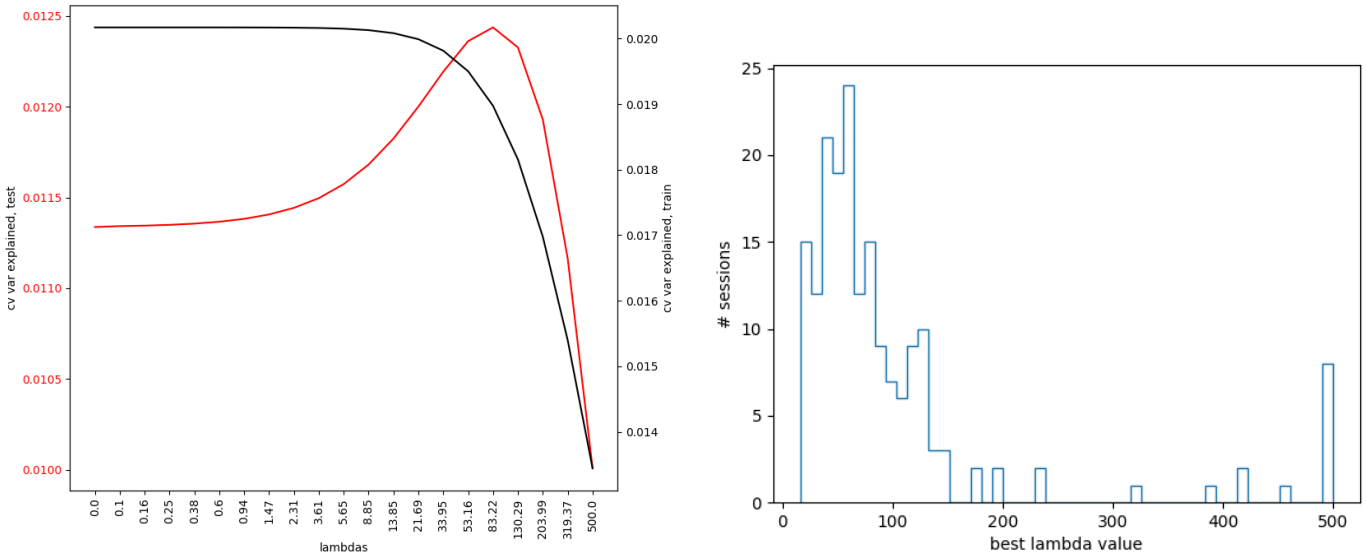


Figure 3: (Left) Cross Validation variance explained for the test and training sets (averaged across 6 folds) with increasing prior strength. This plot is averaging across all the cells in this session. The peak in the red curve is the best lambda value. (Right) Best Ridge Regression parameter for every session.

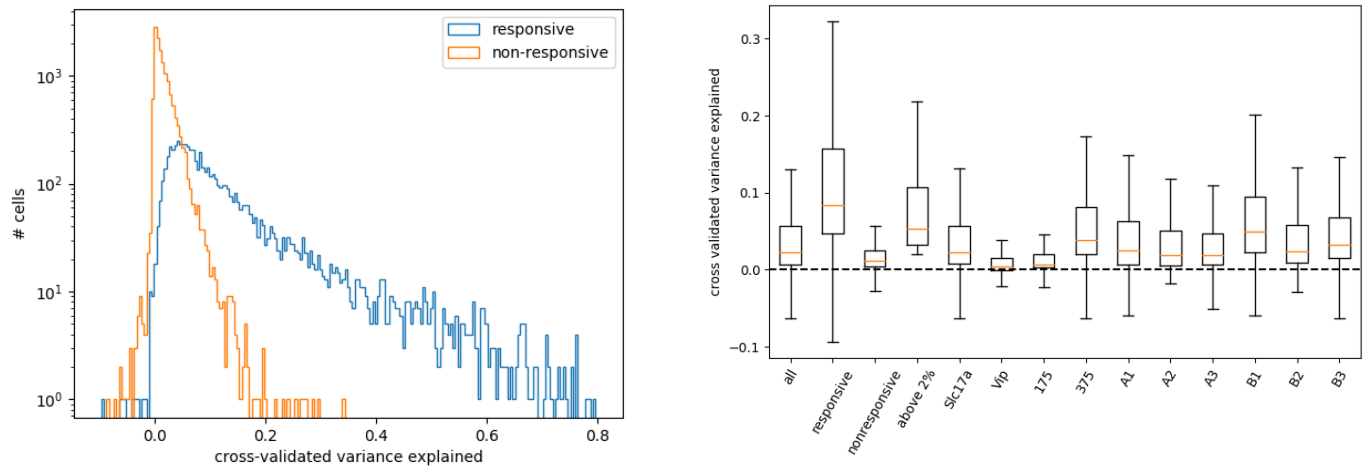


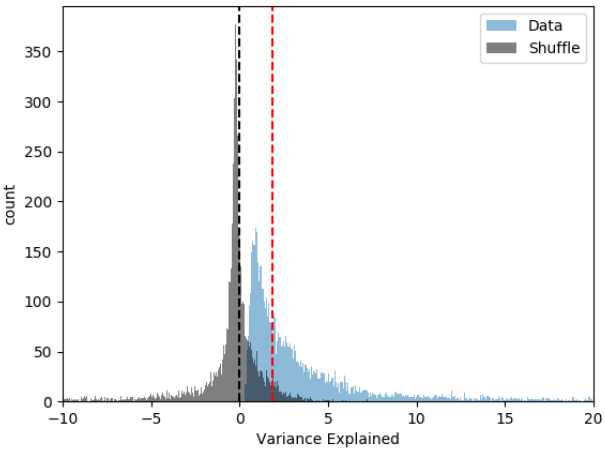
Figure 4: Same as above, but using 6 fold cross validation with ridge regression. Metrics shown were computed on the test folds

Then for each cell I compute the variance explained of this shuffled model response. We can then set a false positive rate (alpha), and find a threshold in the shuffle distribution that has this false positive rate. Then we only analyze real cells that have a variance explained about that threshold. Here, computed on a single session, we find the threshold to be 0. We will need to compute this threshold on all sessions together, but I havent done that yet.

NOTE: I shuffled the model responses with respect to the data. I didn't shuffle the image identity (preserving other regressors), re run the model, and then compute the variance explained.

1.7 Next Steps

- Sanity Check: plotting psths against cells
- Sanity Check: We should look at the 'non-responsive' cells for which we can explain a lot of variance and see what is going on.




```
>>> g.analyze_threshold(all_cv,all_shuf,threshold)
Variance Explained % threshold:       1.88 %
Percent of cells above threshold:    57.32 %
False positive if using 0% threshold: 0.32
False positive if using 2% threshold: 0.04
Threshold needed for Steinmetz level: 4.06 %
>>>
```

Figure 5: Shuffle Analysis to determine significant variance explained.

- Model Evaluation: Plot variance explained as a function of time
- Model Evaluation: Variance Explained on change vs non change images
- Try elastic net regularization and L2 smoothing prior regularization
- Decide whether we will fit regularization hyperparameter for each session together or separately.
- Add in Reduced Rank Regression to fitting
- Add in new regression kernels, and do model selection
- New kernels should address engagement, separate visually responsive from choice encoding cells or reward encoding cells. One kernel for all 64 possible image transitions? One change image for all 8 images? Urgency or expectation kernel? Kernel for image novelty? Kernel for thirst level? Kernel for running speed? Kernel for licking, or for lick bout?
- Add a threshold for including cells in analysis. For now, we are trying a threshold on the max df/f above 0.5. Future work might count the number of transients above some threshold.
- Need for scaling each cell, so regularization hits all evenly. Looks like we are under-fitting strong responses. We decided to divide each df/f by the max value for that trace. Each normalized df/f is bounded by -Inf and 1.
- might be a start of session transient that we miss.
- weird periodic structure that is not due to reward filter, design matrix looks fine. Might be result of image filters. Need to plot flash times and see how it could be happening. Might be the result of dropped frames, of aliasing between dff-timestamps and events. We think this is a result of aliasing. We confirmed that the histogram of diffs between image times was not always exact.
- Need to redo the lambda selection every time we add an kernel. Need to experiment with lambda scaling and thresholding. revisit whether lamdba should be consistent across all sessions.
- need to think about Nick Steinmetz' answers

1.8 Questions (emailed Nick Steinmetz on Jan 2nd)

- In the Steinmetz paper, does he fit multiple sessions at the same time? If so, how?
- If he does not fit sessions simultaneously, does he do any comparisons to see how fit accuracy is dependent on number of cells in each session?
- Steinmetz lets the rank for each neuron vary separately, but when comparing methods uses rank = 18. What is the distribution of ranks across cells? Are there any patterns across brain regions?
- If we can't fit multiple sessions simultaneously, do we have enough cells to use reduced rank regression?