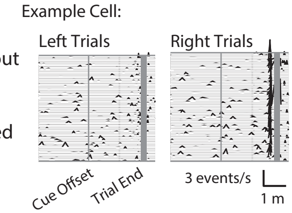
We want to predict neuronal activity, z, given an observed stimulus, x, by learning a model with parameters, β. For modelling neuronal activity we use an exponential link function to enforce nonnegativity.

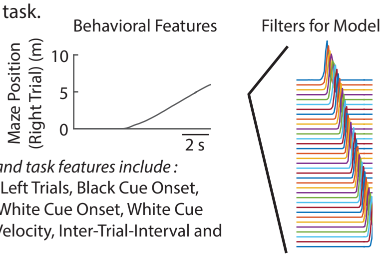
$$z_{predict} = \exp(\beta_o + \beta^T \vec{x})$$

Measure relevant behavior and task features. These data will provide observed stimulus information in our model.

Examine Single Trial Activity to get an idea about the types of responses that should be captured in our model.

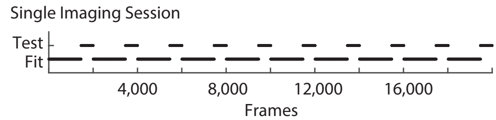


Design a Set of Filters from behavioral and task features that can be used to describe neuronal activity. For example, position in the maze is expanded into a set of filters that consist of evenly spaced Gaussian bumps that span the length of the maze to model neuronal activity in PPC during our task.



Other behavioral and task features include:
Maze Position on Left Trials, Black Cue Onset, Black Cue Offset, White Cue Onset, White Cue Offset, Treadmill Velocity, Inter-Trial-Interval and Reward

Split Data into Fitting and Testing Sets such that model parameters are learned from data in the fitting set and then model predictions are evaluated using test set data. We divide the data into fitting and testing sets such that both groups contain data from evenly distributed segments of an entire imaging session.



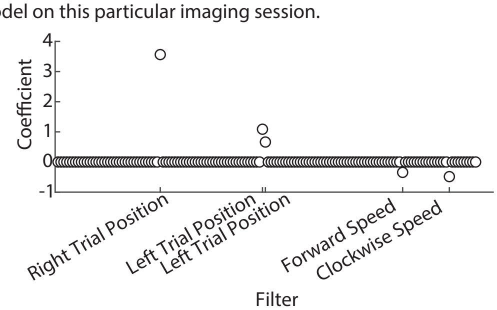
In some cases we fit on data from one imaging day and then test on data from another imaging day.

Fit Model Coefficients to Filters by regression using coordinate descent. We learn the most likely model coefficients, β, given the behavioral training data, X, and the neuronal response training data, Z, for all i = 1:N filters. Our cost function combines this likelihood term with regularization to prevent overfitting. We employ elastic net regularization, a combination of L1 and L2 regularization which penalizes the cost function to limit the number of filters and the weight assigned to each filter.

$$\min_{\beta_o,\beta} -\frac{1}{N}\ell(Z|X,\beta) + \lambda((1-\alpha)\sum_{i=1}^{N}\frac{1}{2}\beta_i^2 + \alpha\sum_{i=1}^{N}|\beta_i|)$$

Where α scales the balance between L2 (a =0) and L1(α = 1) penalties and λ scales the combined regularization penalty.

The learned model coefficients, β, relate how much a given filter contributes to activity in the model. Weights assigned to each filter for the example cell above are shown below. The last filter (at the end of the maze) related to 'Right Trial Position' has the largest coefficient and therefore, the strongest contribution to the cell's model on this particular imaging session.



Behavioral Feature Contributions are calculated by the standard deviation of the kernelized timecourse of behavioral data using a vector of the relevant model coefficients. For example, to calculate position-related contribution to modulations in the model prediction, we cross a vector of all position related filter coefficients with the timecourse of position data.

$$contribution_{position} = standard\ deviation(\beta_{position}^T x)$$

Model Prediction of Single Trial Activity can then be calculated using behavioral data from a separate testing set, either on the same day or from a different day, using the set of model coefficients that best decribed fitting set activity.

Test Model Fit on a Separate Set of Data by calculating the deviance explained by the model. This calculation quantifies how closely modulations in our model's predicted trace correspond to modulations in the real activity compared to a null model. Deviance of our model is calculated as,

$$D_{model} = 2\sum z_{test}\log(\frac{z_{test}}{\hat{z}_{test}}) - (z_{test} - \hat{z}_{test})$$

where z_{test} is the real activity of the neuron in the test set data and z-hat$_{test}$ is the prediction of the model. When predictions closely match real activity the deviance approaches 0. In the null model, we predict the neuronal activity with the most naive model, the mean activity of the neuron. The deviance of the null model is then,

$$D_{null} = 2\sum z_{test}\log(\frac{z_{test}}{\dot{z}_{fit}}) - (z_{test} - \dot{z}_{test})$$

Here z_{test} is again the real activity of the neuron in the test set data and z-dot$_{fit}$ is the mean of the neuron's activity in the fitting set. In our case the mean of the test and fitting set are both normalized to 1. The deviance explained is given by the ratio of deviance in our model and in the null model,

$$deviance\ explained = 1 - \frac{D_{model}}{D_{null}}$$

When D_{model} is low and D_{null} is high, our model does a better job at predicting the neuronal activity than the null model and the resulting deviance explained is between 0 and 1. When D_{model} is similar to D_{null} our model doesn't do much better than the null model and deviance explained is close to 0. When D_{model} is higher than D_{null} the deviance explained is below zero. In this case the model is predicting modulations in neuronal activity that are not present.