Explanation of Variance in Neural Data Using a Convolution Neural Net (CNN)-based, Non-Linear Feature Extractor for Linear Model

Nakul Yadav[*], **Gregory Ghahramani**[†], **Grace Le**[‡]
Department of Computer Science
Cornell Tech
New York, NY 10044
nay2010@med.cornell.edu, gcg4001@med.cornell.edu, dl2228@cornell.edu

Abstract

With the advent of technology to record thousands of neurons in the brain of an organism performing a cognitive task, scientists are able to determine how neurons in the brain process information and drive behavior. While some neurons consistently fire for a single event, most neurons in cognitive processing give rise to a multiplexed high dimensional response. Hence, these neurons are often referred to as exhibiting "mixed selective" responses, encoding multiple aspects of the environment. Simple regression models are powerful tools to analyse but fail to explain variance in a high dimensional dataset. This paper aims to perform a comparative analysis of multiple regression models and how their performance changes by preprocessing the inputs before learning. Furthermore, to enhance the performance of a regression model, we introduce an automated feature extractor using a convolutional neural network (CNN) to provide a non-linear combination of features which feeds into the linear model, with an aim to maximise the explained variance of neuronal activity in response to task events. This paper shows that non-linear combinations of feature inputs often better explain the variance of neurons than linear features.

1 Introduction

The breakthrough of single neuron recordings using two photon microscopy has developed into the capability to simultaneously record thousands of individual neurons in awake animals. Multiple models have been used to explain information processing in brain. However, with increasing complexity of cognitive processing, we need advanced models that can explain how different factors are encoded in the single neurons. Here, we aim to elucidate different models can improve our understanding of this processing. We also implement an automated feature extractor using a convolutional neural network (CNN) to provide a non-linear combination of different "task features" to feed into a generalized linear model, which can maximize the explained variance of neuronal activity in response to task events. This will help us understand what information is encoded in the area where we record neuron activity.

2 Methods

2.1 Virtual Reality (VR) based cognitive task

We employed a VR based cognitive behavioural task where mice associated four different stimuli (visual, odor, olfactory and texture) with a reinforcement, also known as contextual learning. We also track speed and acceleration of mice on the ball while it's performing the task. All the stimuli, reinforcement deliveries and kinematics are depicted in the feature matrix for regression models.

2.2 Polynomial Expansion of Feature Matrix

To include higher order features for regression modelling, we perform a 2 dimensional expansion of the feature matrix, which would include interaction terms between each of the feature vectors to include non-linear features in the linear regression model.

2.3 Convolution Neural Net based Linear Model (CNN-GLM)

We aim to develop a Convolution Neural Net to make a feature extractor that feeds into a General Linear Model (GLM) to explain variance of neurons explained (Minderer et al., 2015). Further details of the method are described under "CNN-GLM Architecture"

2.4 Explanation of Variance by Features

We cluster features into common task events, such as all stimulus provided in reward context are clustered into "Reward Context," similarly for neutral and fear contexts. Speed and acceleration are grouped together as kinematics. To analyze how much these groups of features influence each neuron, we proceeded to drop these clustered features and retrain our model. We then use this model to predict the neuronal output, and analyze this against the actual neuronal output, producing a Pearson r correlation value. Drop/gain in r value indicate if and how much these tasks correlate with each neuron.

[*]Weill Cornell Graduate School of Medical Sciences
[†]Weill Cornell Graduate School of Medical Sciences
[‡]Cornell Tech

Applied Machine Learning Project Report. Fall 2020.

3 Dataset

The dataset collected includes 940 neuron recordings for 5845 time points (frame rate 3.2Hz) and a corresponding matrix of 15 tasks with coinciding time points, which serve as an input to model the response of the 940 neurons, as shown in **Figure 1**. These features include binary features (ex. Licking activity, 1 for a lick, 0 for no lick) and continuous features (ex. velocity and acceleration of the mice). The features are all scaled to be between 0 and 1.

4 Feature convolution with genetically encoded calcium indicator (GECI) response function

GECIs display an exponential decay response to a single action potential, so to mimic the response of neurons in response to task events we use convolve the features as inputs to the regression model in an attempt to increase the variance explained by the model. Figure **Figure 2** depicts the convolution functions (exponential decay with tau=0.6 seconds (2 frames)

5 CNN-GLM Architecture

We created a non-linear stage (CNN) which extracts convolves the feature matrix from the inputs to be fed into a linear model for each neuron. The idea behind this process is to create a non-linear interaction of task features which will mimic the non-linearity of neuronal excitation to stimuli.

The input to the model is a 40 frame long snippet (7 sec), and training the model to predict the activity at the 30^{th} frame. The goal is to use timepoints 1 - 40 to predict the neuronal reading at time point 30. Then use points 2-41 to predict the neuronal reading at time point 31, and so on, until reaching the end of the data set. This is done by stacking the data in 40 frame sets, so each x input is of shape (1, 15, 40, 1), so that at start time 1, there are 15 features and 40 time points of these feature values stacked on top of each other. The data is then split into training and validation sets. For each block of 200 time points, the first 160 points go to the training set and the last 40 go to the validation set. This is done to ensure that time does not affect the distribution of the training and validation sets, by equally representing them across all 5000 time frames. This results in an x-train shape of (4674, 15, 40, 1) for 4674 different start time points, 15 features, 40 consecutive timepoints stacked and a y-train shape of (4674, 1), for each of the 940 neurons.

The architecture of the CNN is:

Convolution-32, kernel size = (15,1)
MaxPooling - (1,2)
Convolution-32, kernel size = (15,3)
MaxPooling - (1,2)
Convolution-32, kernel size = (15,5)
MaxPooling - (1,2)
Batch Normalizaton
Dropout-.3
Flatten
Dense-128, sigmoid
Dense-64, sigmoid
Dense-1, sigmoid

We then compile the model using keras Adam optimizer with a learning rate of 0.015, using a Poisson loss.

6 Results

We used three linear regression models initially, referring to them as:

1. **Linear Regression** - scale features between 0-1 and input into a linear regression model.
2. **Conv-Linear Regression** - convolved features with decay input into a linear regression model.
3. **Poly-Regression** - input the polynomial expansion of the convolved features into a linear regression model.
4. **CNN–>GLM** - Convolved the feature matrix using a CNN, to input into a 2 hidden layer neural network

To quantify the success of the fit, the Pearson correlation was calculated for each model between predicted and validation set for each neuron. Below in Figure **Figure 3** is a representative example of a neuron's predicted and actual response across the previously mentioned three models and CNN-GLM output.

6.0.1 CNN-GLM predicts activity of more neurons than other models

Using an r value threshold of 0.05, we identified the number of neurons whose activity was reliably explained by the models. Neurons with a r < 0.05 were considered "not significantly" explained by the model and discarded. In **Figure 4**, we see linear regression showed activity of 561 neurons can be significantly explained by the model, compared to 556 neurons from Conv-Linear Regression, 493 neurons from Poly-Regression model. CNN-GLM outperformed other models and explained the significant activity in 675 neurons, capturing more high dimensional neurons previously unexplained by other simpler models.

To analyze the performance of each model across all neurons, we calculated the correlation between each model's prediction and the actual recording. Then, compared these r values between each model across all neurons with (r>0.05) . These values were then regressed with zero-intercept. Slope larger than 1 indicates that the model described on the y-axis performed better than the model described on the x-axis, shown in **Figure 5**. We show that CNN-GLM generally performed the best, followed by the linear-regression, then the Conv-Linear Regression and the worst was the Poly-Regression model.

6.1 Leave one out model explains variance explained by each feature

We used a "leave one out" method to determine which groups of feature the neurons exhibited the strongest response to. The stimuli were grouped into Reinforcement, Reward, Neutral or Fear stimuli, Kinematics, and Room (learned response). Using the best performing neurons of the CNN-GLM model, we determined how much variance each feature group contributed to the overall variance explained by the model using the equation $(X - X_i)/X$, where X is the Pearson r correlation using all feature inputs, and X_i indicates the Pearson r correlation using a feature set with the dropped groups of features. Therefore, if $X_i = X$, all of the variance is explained by group i. If $X_i = 0$, none of the variance is explained by that group of stimuli.

Figure 6 indicates the sample response and corresponding r value of the resulting model. The drop in r value is due to the removal of the input groups.The heatmap below indicates the fraction of variance explained by each stimuli group of the top 98 performing neurons.

Figure **Figure 7** shows how different features contribute to the variance explained by the CNN-GLM model as a fraction of total variance explained. Some neurons respond mostly to reinforcements such as neuron 24 and 63, whereas some respond to kinematics such as neuron 19,20 and 84.

7 Discussion

We have shown how taking into account nonlinear interactions between features can monumentally change our understanding of how the brain processes and encodes information. Neurons during cognitive processing encode information in a high-dimensional state space which can be decoded using more advanced regression models. However, we also show that just increasing the dimensionality does not increase variance explained and can also introduce noise in the system, "Poly-regression" being an example. The CNN-GLM provides a targeted approach to weigh only the most important non-linear combinations and reduce the effect of other interactions.

8 References

M. Minderer, K.D. Brown, C.D. Harvey The spatial structure of neural encoding in mouse posterior cortex during navigation Neuron, 102 (2019), pp. 232-248

9 Figures

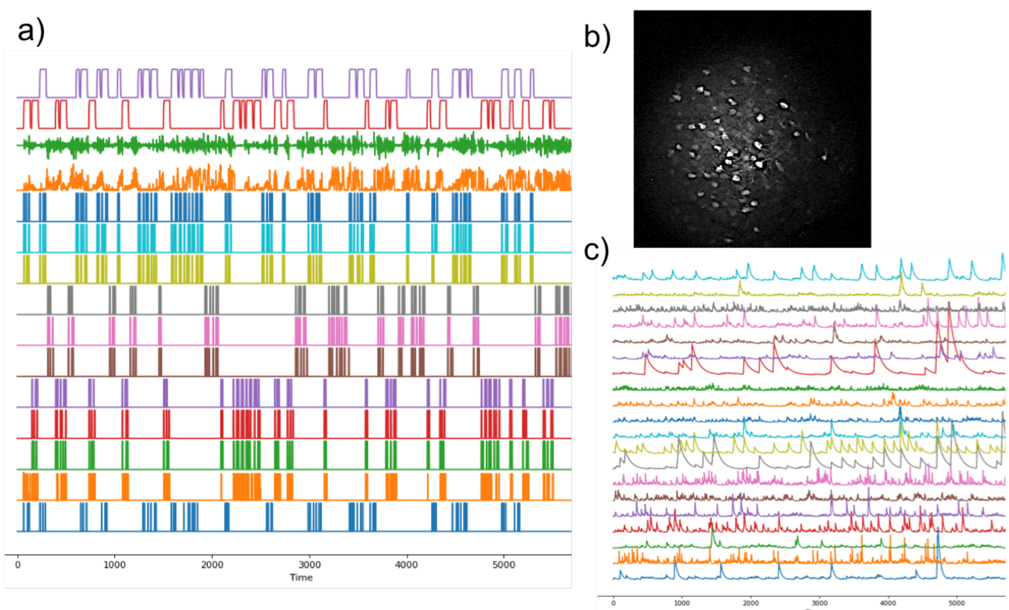

Figure 1: **Task features and neuronal recordings used for the models.** (a) the features used for the model. The recordings of cells from (b) are depicted in representative responses in (c) for the first 20 neurons.

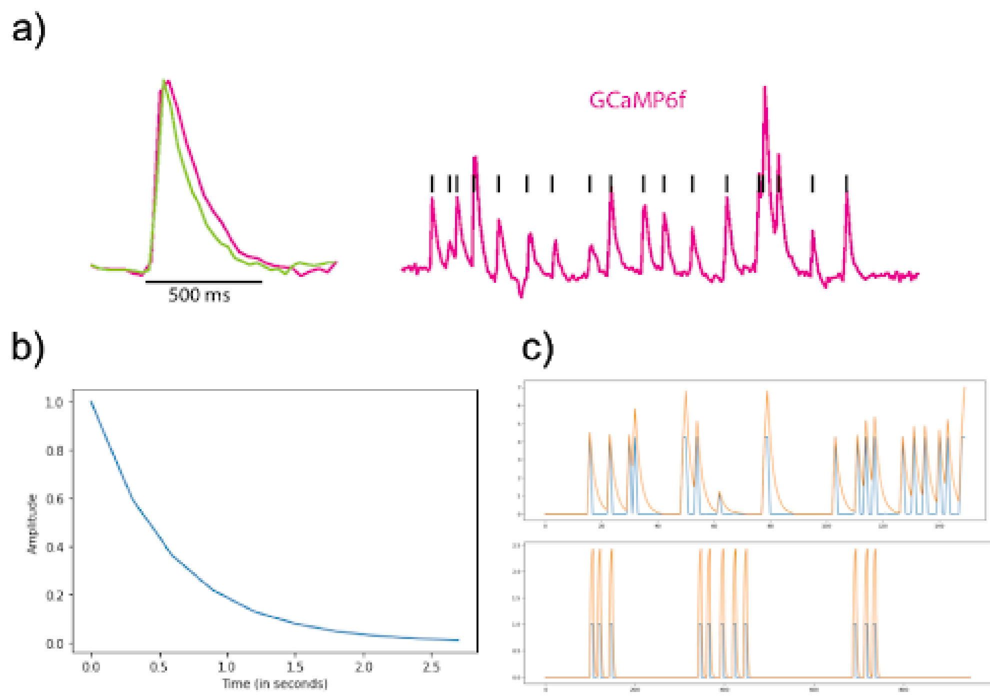

Figure 2: **Feature convolution with response function.** (a) is a representative figure of an action potential and corresponding GECI signal recorded. (b) is the exponential decay window used to convolve feature vectors which are displayed in (c) as representative examples of two feature vectors, airpuff delivery and odor in reward delivery.

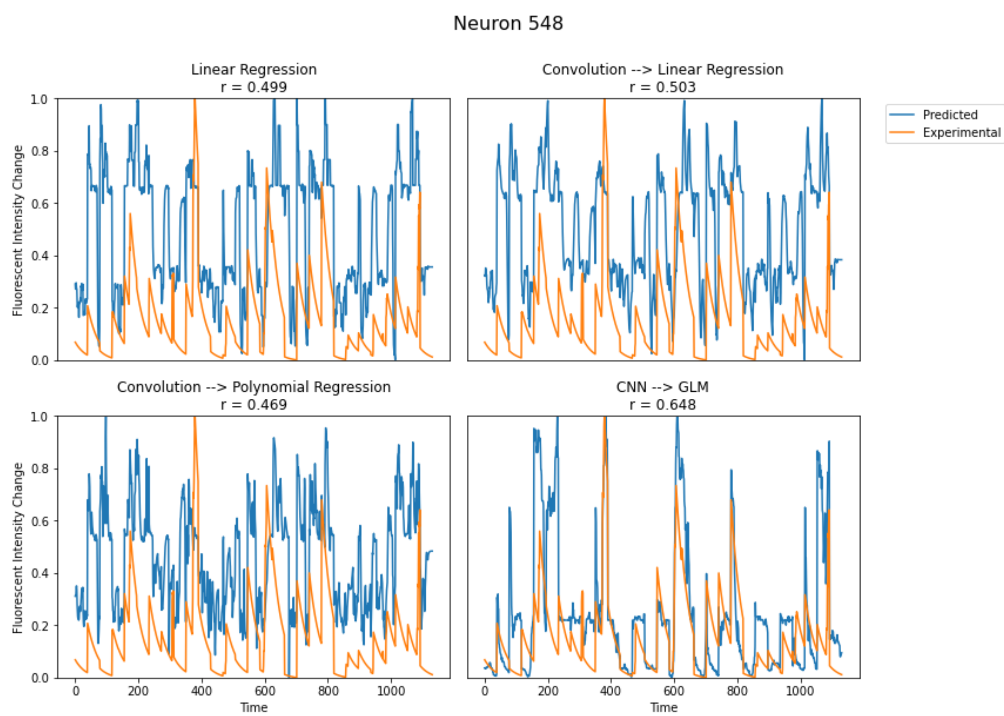

Figure 3: **Example of neuron's predicted and actual response in four models.** A representative example of a neuron's predicted and actual response across linear regression, conv-linear regression, poly-regression, and CNN-GLM output.

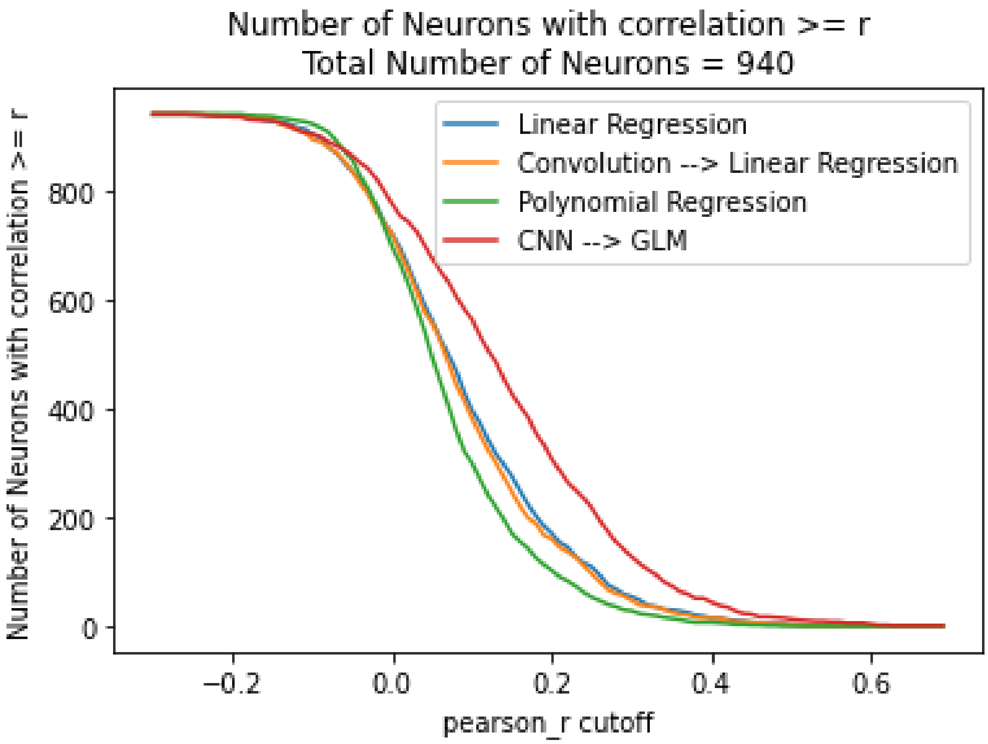

Figure 4: **Number of neurons with activity reliably explained by the models.** Using r value of 0.05 as a threshold, we identified the number of neurons whose activity was reliably explained by the models

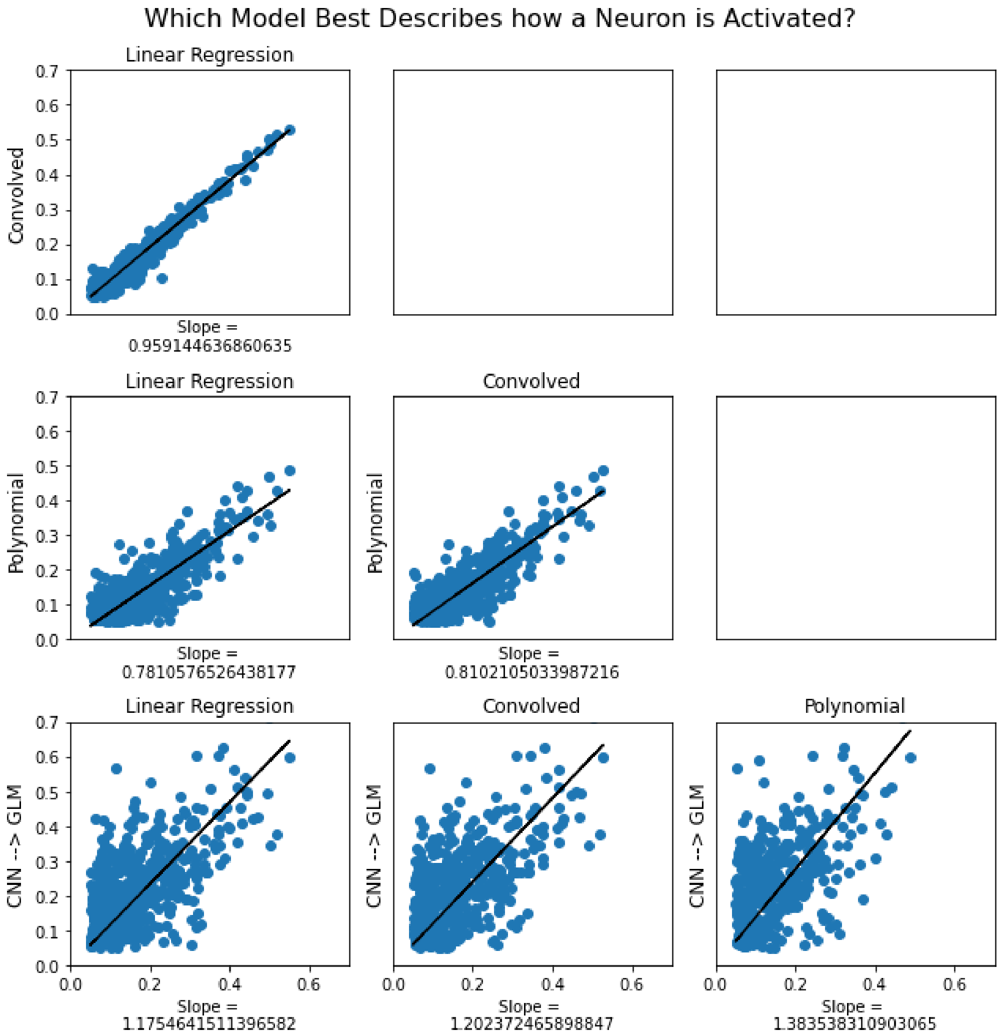

Figure 5: **Performance of four models looking at r values.** Graph comparing performance of linear regression, conv-linear regression, poly-regression, and CNN-GLM models looking at number of neurons with correlation larger or equal to r.

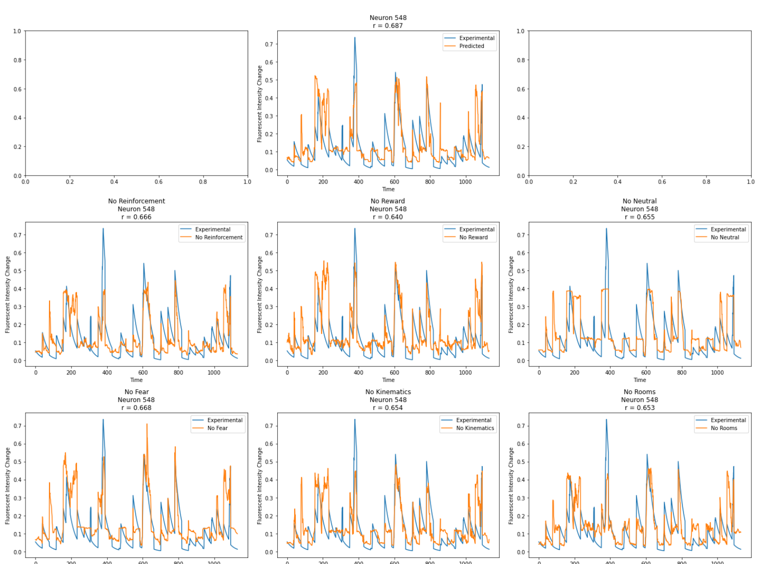

Figure 6: **Explainability of variance using leave-one-out model on a single neuron.** Using the "leave one out" method, the CNN–>GLM fit and predicted the neuronal output using all of the features except the group indicated in the subplot titles.

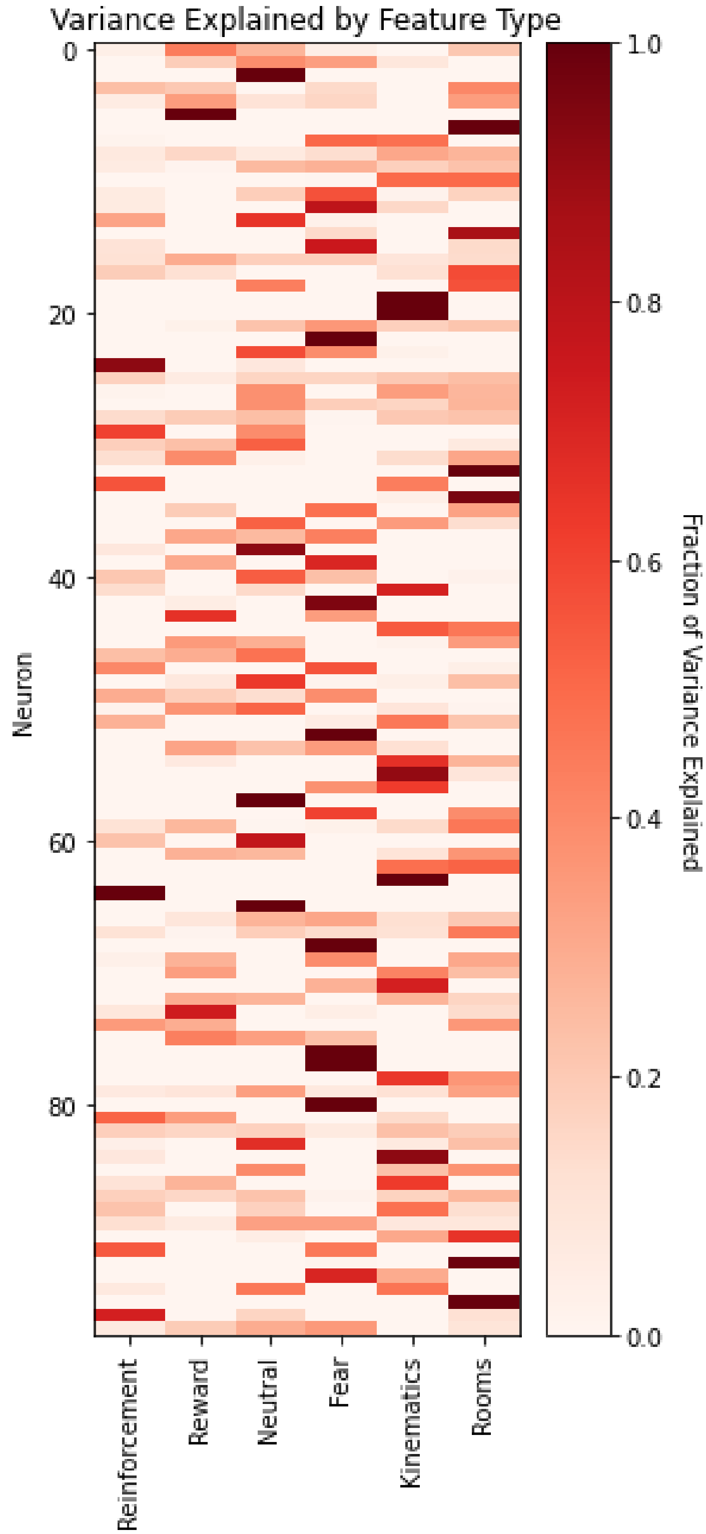

Figure 7: **Explainability of variance using leave-one-out model on the top performing neurons.** This heatmap indicates how much each neuron is influenced by the different stimuli groups, where read shows high dependence and white indicated no dependence.