EPFL

BIO-ENG456: Controlling behavior in animals and robots

Group: 8

Lorenzo GERMINI (SV)
Mathias NURIS-SOUQUET (SV)
Hila VARDI (MT_RO)

Professor:
Pavan RAMDYA

Teaching Assistants:
Gizem Ozdil, Jonas Braun and Chin-Lin Chen

25 May 2021 — École Polytechnique Fédérale de Lausanne

Contents

1 Abstract

This project aims at exploring neural activity and neurons correlations with behaviour in Drosophila flies. Data collected in the UPRAMDYA laboratory at EPFL provided neural activity, joint angles and joint 3D poses of the Drosophila fly for 12 trials. This data was analyzed to better understand how neural signals control limb movement. Groups of neurons with similar signals were found, suggesting that the motor control in Drosophila uses multiple neurons in parallel possibly for better robustness. Correlations between neurons and joint angles were stronger than those with behavior, suggesting that the neurons recorded are Descending Neurons. Behavior classification from neuronal activity was found to be possible. In particular, with the implementation of a more advanced neural network classifier remarkably good performance metrics were achieved for the prediction of multiple behaviors. A logistic regression method may be enough, depending on the level of accuracy needed, for a reasonable enough quality prediction in the case of a binary classification. These results constitute an important step towards a more comprehensive understanding of the function of different neurons.

2 Introduction

2.1 State of the art

Over the past years, numerous methods have emerged to automate the quantification of animal behavior at a resolution not previously imaginable. This has opened up a new field of computational ethology which enables to quantify in near completeness what an animal is doing. These advances position behavioral quantification as a core instrument in the neuroscience toolbox, essential to the quest of understanding the brain.

Looking at neuronal control for Drosophila, a study by Bidaye et al.[1] identified descending neurons (MDN) for backward walking in Drosophila as well as ascending neurons (MAN). MDN activity is required for flies to walk backward when they encounter an impassable barrier and is sufficient to trigger backward walking under conditions in which flies would otherwise walk forward. MAN on the other hand promote persistent backward walking, possibly by inhibiting forward walking. These findings provide an initial glimpse into the circuits and logic that control walking direction in Drosophila.

The first step towards studying neural activity and behavior is to choose a method for recording neural activity. One of them is depicted in a study done by Chen et al.[2], where the authors are using 2-photon functional imaging from the ventral nerve cord (VNC) of behaving adult Drosophila melanogaster to reveal patterns of activity across nerve cord populations during grooming and walking and to uncover the functional encoding of moonwalker ascending neurons (MANs) and moonwalker descending neurons (MDNs). Finally, combining 2-photon functional imaging with facilitation experimental access to VNC, the authors were able to investigate circuits associated with complex limb movements.

The following project thus aims at investigating behavior prediction adapted to Drosophila through 2-photon functional recording over 123 neurons in the cervical connective of fly and behavioral data recording (using 7 cameras over 5 markers per leg of the fly.)

2.2 Data presentation

The data used in this project was recorded from one fly (R57C10-Gal4 > UAS-GCaMP6f, UAS-tdtom) across 12 trials of around 250s each. The neural data was imaged with a 2-photon microscope at a sampling rate of around 16 Hz and consists of fluorescence traces of 123 neurons in the cervical connective expressing the genetically encoded Calcium indicator GCaMP6f[3]. The pose of the Drosophila was recorded with 7 cameras surrounding the fly, at 100 Hz, capturing its 6 legs. The fly's 5 leg joint positions were extracted from the videos using the marker-less pose-estimation algorithm DeepFly3D[4] and the joint angles were then computed (7 joint angles detailed in Figure 4.13). Finally, the data was manually labelled, indicating the fly behavior for each time frame. The possible behavioral labels used are: resting, walking, anterior grooming, abdominal pushing and posterior grooming.

3 Methods

3.1 Data Pre-processing

3.1.1 Noise reduction

For the noise reduction phase we implemented an ad-hoc tuned Butterworth low-pass filter to the neural and behavioral data. The filter tuning was performed accounting for the differences in sampling rate between the 2-photon microscope for the fluorescence traces of the 123 neurons and 7 cameras for the behavioral variables recording and computing, as well as the intrinsically different possible sources of noise present in the two different classes of signals. For the choice of the parameters, we visually inspected the performance of the filter over a wide range of cut-off frequencies, shown in Figure 3.1 with the other filter parameters fixed and specifically chosen for the data type. The aim of the parameters selection was to reach the best compromise between significant noise reduction and minimal loss of meaningful signal features.

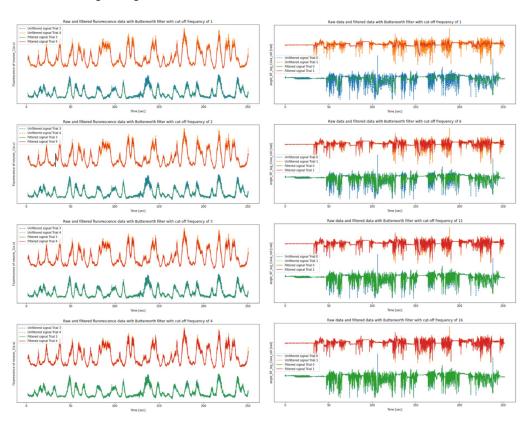***Figure 3.1:*** *Raw and filtered data with different cut-off frequencies for signals from two different trials in order to draw more robust conclusions. Respectively on the left fluorescence data and on the right behavioral data.*

Following this approach, second order low-pass filters with cut-off frequencies of 2.5 and 100 Hz were selected to be applied respectively on the neural and behavioral data.

3.1.2 $\Delta F/F$: baseline removal

In order to account for the observation of different baseline fluorescence values, one can calculate the so called "Delta F over F" :

$$\Delta F/F = \frac{F - F_o}{F_o}$$

where F_o is the baseline fluorescence of each neuron. This allows to remove any baseline from the neural data which preserves all the information from neural recordings and allows fair comparison between neurons. In order to provide stability to this step, the minimum of the moving average of each neuron signal is computed and removed from the neural signal (thus, this step is done independently of the trials).

3.2 Principal Component Analysis (PCA)

PCA is a processing method used to find the principal components and projecting the data on it. This change of basis allows to reduce the dimension of the data and bring to light differences within the data. Differences in signal ranges between neurons may arise from the data extraction methods. Therefore, the neural data was first standardized with a standard scaler transformation. This step is necessary for preventing one signal dominating the analysis.

When investigating similarities between neural activity (subsection 4.1), the PCA was done with time points as features and each neuron as a sample. On the other hand, when classifying behavior from joint angles (subsection 4.4), the joint angles are used as features and the time points as samples. This difference comes from the different goals of the analysis. The former aims at grouping neurons with similar activities whereas the latter aims at grouping time points that have similar joint dynamics.

3.3 Unsupervised clustering

Unsupervised clustering is used for grouping datapoints that are close to each other in the given dimensions. In this project, two methods are explored, k-means clustering and Gaussian mixture models (GMMs). K-means clustering consists of separating the data into clusters where each datapoint is attributed to the cluster with the nearest mean. A hard assignment is performed, the points are put into a cluster without a probability value. Clustering with GMMs computes a Gaussian distribution model for each cluster. The datapoints are assigned probabilities of belonging to each cluster, indicating how sure we are in the clustering. GMMs with 'full' covariance matrix can have elliptic decision boundaries as opposed to k-means which has circular boundaries. When working with many datapoints and dimensions, k-means is faster than GMMs but is less robust.

3.4 Correlation

As part of exploratory analysis, correlating neuronal activity and behavior helps to understand which information the neurons of the dataset convey. One can assume that a correlation between the activity of one neuron and an external variable (for example a joint angle - lead by several muscles) means that this neuron encodes information about this external variable. Two types of correlation are used in this project. Firstly Spearman rank-order correlation coefficient will be used to evaluate

relationships involving ordinal variables. Secondly, the correlation between neuronal activity during behaviors is performed using the average neuron's activity separately during the different behaviors. This is different from using a correlation coefficient because it does not provide a coefficient of whether or not an increase/decrease in A results in a change of B, but allows to highlight the difference between the average activity of neuron X occurring between each different behaviors.

3.5   Down-sampling data

Down-sampling is required to account for the difference in sampling periods of neural and behavioral data. The down-sampling function for the behavioral data was tuned to handle the case where two behaviors are equally likely and to eventually discard the data if a certain threshold of equally labelled behavioral frame was not met. After having explored different methods, given that the down-sampling from 100 Hz to 16 Hz was always resulting in the decision to be made between 6 or 7 behavior classes, a threshold of 4 was set in order to increase the robustness. This way at least 4 frames in the same moving window have to be labelled the same otherwise the all set of frames is discarded. The features associated with the window of labels that were deemed excluded were then removed from the data frame.

3.6   Behavior Classification from Neural Data

For the logistic regression classifier prior to training, the dataset was split with a 0.2 test size in a "stratified" way in order to account for eventual imbalances in the proportions between the labels. Then the two sets underwent a standardization performed by a standard scaler transform which was fitted on the training dataset. The model with the default parameters given by sklearn[5] is then fitted on the training data and a prediction is made based on the test dataset. A brief sample of the prediction compared to the true labels, a detailed classification report and the confusion matrix with heat map are printed in order to visualize the quality of the prediction as well as more insights into the classifier performance on the single labels. In the case when the model does not converge, the max iterations are increased up to 1500 if necessary. After further analysis no other action is taken as the logistic regression shows quite a stable behavior even when not reaching convergence. No grid search was run given the substantially less performing nature of Logistic Regression in complex binary classification situations especially when dealing with high number of features. Further related aspects will be discussed at the end of the results section. Additionally, the prediction's quality was already very poor.

For the neural network classifier the same preprocessing, splitting and visualization was implemented as in the case of logistic regression. The multi layer perceptron model from sklearn[6] with default parameters is used. The standard scaling during the preprocessing is particularly important for the latter training of the MLP as neural network are very sensible to unscaled data. If the optimization does not converge during training, the evolution of the loss function during gradient descent is printed to investigate which steps could be envisioned to improve the loss function behavior. If changing the initial learning rate does not improve the situation a quick grid search on a number of parameters is run in order to fine tune the hyper parameters choice for the best prediction performance. Each time when a classifier model was built the same initial random state were given in order to minimize variances in prediction performance

3.7 Wavelet transform

The wavelet transform is used to add dynamic information of the data, by adding frequency features. The normalised Morlet wavelet was used here to analyse short-time oscillations. It was applied to the joint angles before the PCA, with 25 frequencies in the range of $1 - 50\ Hz$.

4 Results

4.1 Investigating clustering of neurons' activity

With the aim of understanding how neural activity encodes behavior, we first need to understand if there are neuron groups that work together. The neurons were clustered into groups of similar activity using Principal Component Analysis (PCA) followed by unsupervised clustering. The PCA was performed, as described in subsection 3.2. The dimension-reduction performed by the PCA projects neurons with similar signals close to each other.

In order to explain 90% of the variance between neurons, we need 27 PCA components. This high number could be explained by the fact that the fly is doing repetitive tasks and there isn't one particular time point in which the neural activity varies drastically. The explained variances of the first 10 components are: $[0.273, 0.264, 0.068, 0.036, 0.033, 0.025, 0.023, 0.017, 0.017, 0.015]$

The first two principal components (PC1, PC2) explain more variance than the rest of them and may possibly convey information about the changes in the Drosophila behavior during the experiments. The loadings of the first two components are shown in Figure 4.1. The second PCA component presents a decreasing step pattern over time, this could possibly be a result of the different tasks the Drosophila performed during the later trials.

Figure 4.1 also presents the neurons projected onto the first two principal components of the PCA. Each point on this plot represents a neuron projected onto the two components and the spatial density of the points corresponds to similarities between neurons' activities.
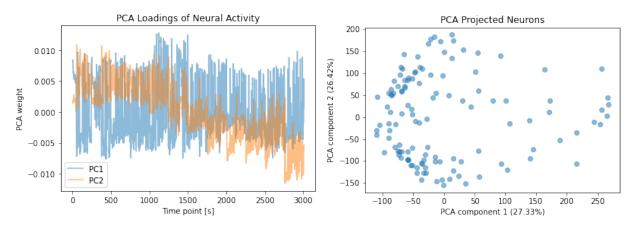
Figure 4.1: *PCA results. Left: PCA loadings for each time point, of the first two principal components (PC1, PC2). Right: Neurons projected onto the first two principal components. Each point corresponds to a neuron, with closer neurons having more similar activities.*

The clustering of neurons based on their distribution on the PCA components can be done with different unsupervised clustering algorithms. Figure 4.2 presents the results of K-Means and GMM clustering. The number of clusters was chosen manually and there are visible differences between the clusters found by these two algorithms. Performing a grid search trough the hyper-parameters of each algorithm with subsequent inter-comparison may constitute a possible marginal improvement.

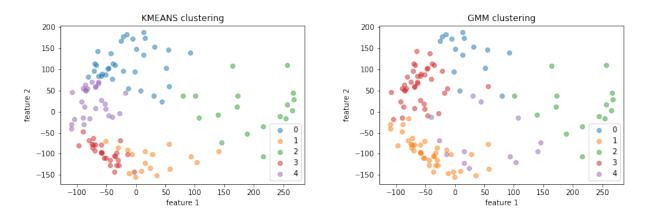***Figure 4.2:*** *Clustering of neurons according to their distribution on the first 27 PCA components. Left: k-means clustering with 5 clusters. Right: GMM clustering with 'full' covariance matrices and 5 components.*

The similarities between the activities of neurons from the same cluster can be visualized in Figure 4.3. For better visualization, only half a trial (number 8) is shown. In contrast, the signals of neurons from different clusters are also shown. It is seen that there are still some similarities between neurons of different clusters. This may be due to limitations of the clustering algorithms.
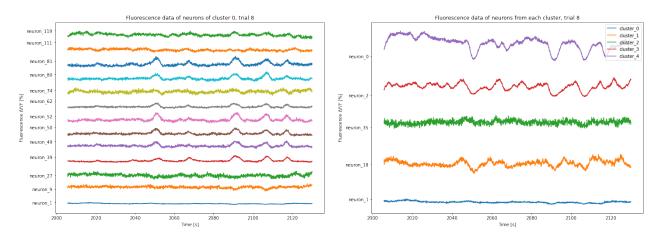
Figure 4.3: *Neural signals of neurons from the same (left) and different (right) neurons. Signals of the first half of trial 8, using the clusters found by K-Means.*

A more quantitative method of comparing between neurons is to calculate the inter-neuron correlation coefficients, giving a measure of correlation between two neurons. Figure 4.4 shows the correlation coefficients between the neurons in cluster 0 found by k-means. All of the neurons in this cluster are positively correlated with each other. Appendix A presents the correlation coefficients of neurons in each cluster, for both clustering methods. For both k-means and GMMs, the clustering works relatively well, with most neurons having positive correlation. These correlation coefficients can be further used to quantitatively compare between different clustering algorithms.

correlation coefficients of cluster 0

Figure 4.4: *Inter-neuron correlation coefficients for cluster 0, found by k-means.*

4.2 Correlating neuronal activity and behavior
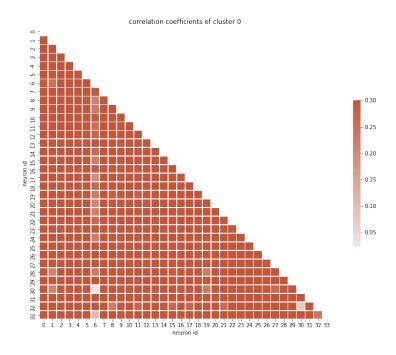
4.2.1 Identifying correlations of individual neurons

As said previously in the Methods (subsection 3.4), we used correlation in order to put into relief differences between individual neural activities as shown on Figure 4.5. In order to be easily readable, we opted for a colored matrix with individual neurons on one axis and behaviors on the other axis ; moreover : standardising the neural $\Delta F/F$ in order to have mean zero and unit variance (which allows better comparison).

First thing to say is that along the 12 trials, students of the course labeled each frame with behaviors among resting, walking, anterior grooming, abdominal pushing and posterior grooming. Thus, Figure 4.5 provides only data for these five behaviors. We can observe disparities in between individual neurons where some are more active for one single behavior, and others have low neural activity for certain behaviors. It seems that specific neurons are much more active for behaviors such as anterior grooming, abdominal pushing, posterior grooming.
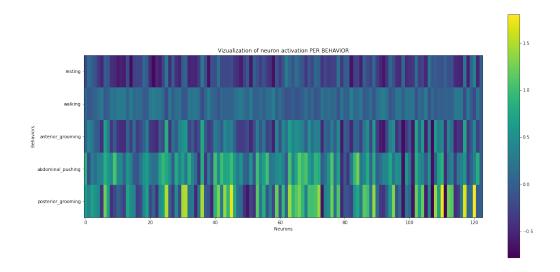
Figure 4.5: *Mean activity of a neuron during different behaviors.*

Behavior	resting	walking	anterior grooming	abdominal pushing	posterior grooming
mean	-0,231369333	0,090474215	0,006007774	0,398909232	0,406034199
std	0,251354289	0,137628048	0,382312003	0,42109597	0,668431132

Figure 4.6: *Statistics on mean neural activity among individual behaviors. std stand for Standard Deviation.*

From Figure 4.5 it seems that anterior grooming, abdominal pushing and posterior grooming are behaviors where neural activity of certain neurons is much more diverse. Indeed, we can see that for resting and walking, the standard deviations of average neural activity after standardization are respectively 0.25 and 0.13 which is quite low compared to the other three behaviors. This may show that there are no specific neurons that could lead to the prediction of those two behaviors.

More specifically, the neurons possessing a standard deviation over their behavioral mean superior than 0.5 are counted 21, 98% of them are neurons possessing a neural activity superior than 1.0 during posterior grooming and are colored in yellow on Figure 4.5 (which is now considered as a "diverse neural behavior"). These results will be taken into account for regression improvements because it could mean that these 21 neurons may be significantly involved in posterior grooming. We will particularly add these neurons for the multivariate regression for posterior grooming.

We could improve again our statistical results by performing several investigations such as: looking at the neurons with specifically high neural activity for a behavior, are they low activated/inhibited during other behaviors? We could improve this also by performing statistical significance analysis of our results, such as power analysis with statistical hypothesis testing. This could assess whether or not with a set of 123 neurons and 5 behaviors, these results are "strong enough".

In order to assess the relation between the activity of the neurons and joint angles, we computed the correlation coefficient between individual neurons and individual joint angles using the Spearman rank-order correlation coefficient (scipy.stats.spearmanr()) instead of the Pearson correlation coef-

ficient which seemed inappropriate because we don't know whether the neural data are Gaussian distributed or not.
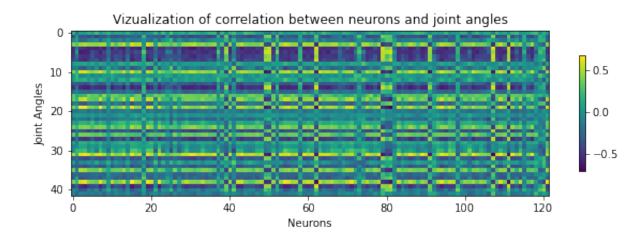

Figure 4.7: *Mean activity of a neuron during different behaviors.*

As we can see, a various range of coefficients are computed in between the minimum: -0.719 which suggests a high negative correlation (an increase in A implies a decrease in B or opposite) and the maximum: 0.689 which suggests a high positive correlation (an increase in A implies a increase in B or opposite). In particular, we can identify two particular groups, presented in Figure B.2, which shows two sets of neurons which possess the same correlation coefficients with joint angles. This analysis again shows different clusters of neurons which may be involved in different parts of the behaviors. In general, it seems that neural activity is more correlated to joint angles than to behavioral categories, which may suggest that these are Descending Neurons. We could improve our analysis by trying to investigate more on the role of this two sets of neurons identified in Figure B.2.

4.2.2 Identifying correlations of low-dimensional signals

In order to assess whether clusters identified in neuronal activity correspond to clusters in behavior data, what we did is to perform the same correlation coefficients analysis than from last part, applied to sets of neurons provided by the clustering analysis ; results can be shown in Figure B.2. We can say that clusters 1 to 4 group neurons with same correlation with individual joint angles. Moreover, we can add that clusters 1 to 3 are really similar to group A found in Figure B.2 and that cluster 4 is similar to group B. This confirms what we saw earlier in the clustering part : we identified groups of neurons with same implication on behavioral data which we will provide for regression improvements in further parts.

4.2.3   Investigating feature correlation for further regression

In order to be sure these results are useful for regression, we need to make sure that within each cluster found, individual neural activities do not provide the same information. In order to assess this claim, we performed an autocorrelation matrix between pairs of individual neurons which can be found in Figure 4.8 which allows to say that individual neurons are not much correlated because the max correlation is not above 0.25. Thus, we can conclude that each different neuron brings information which for some of them are really correlated to certain behaviors/joint angles.
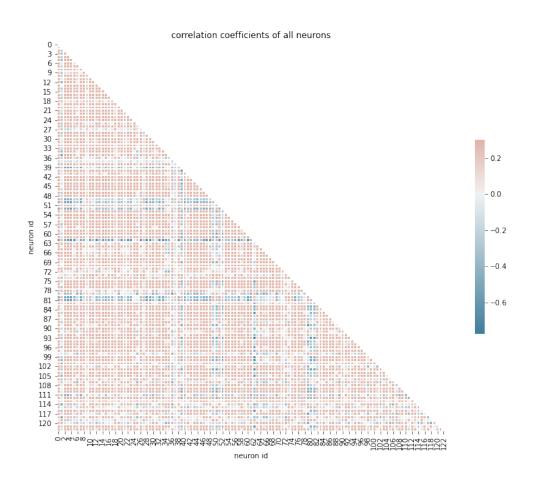

Figure 4.8: *Autocorrelation matrix between each individual neurons.*

4.3 Classifying behavior from neural activity

For the final part of the neural data analysis several methods have been tested in order to try to predict the behavior of the fly based on its neuronal activity.

4.3.1 Predicting one behavior

We implemented a logistic regression classifier to predict, using the neurons' activity as features and a binary "walking" variable as labels, whether the fly was walking or not. The number of regressors is increased with the neurons being added based on their correlation coefficient which was computed in subsubsection 4.2.1, Appendix C. In Figure 4.9 we can observe the performance metrics and confusion matrices of the prediction with respectively 1, 2, 25, 50, 100 neurons and all of them.
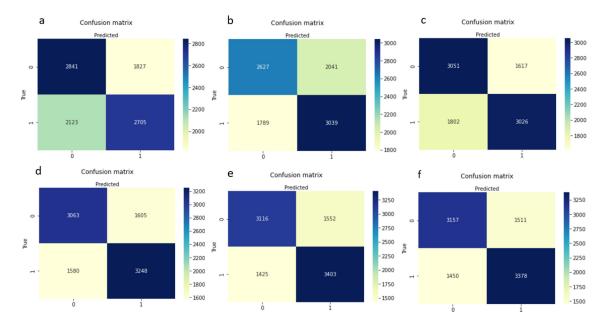

Figure 4.9: *Confusion matrix for the predictions with 1 (a), 2 (b), 25 (c), 50 (d), 100 (e) and all of the neurons (f) as regressors. The neurons were previously sorted based on the correlation coefficient with the "walking" variable. Binary class prediction with logistic regression. Detailed performance metrics in Appendix C*

With one neuron as input the prediction is very poor with very low accuracy despite using the highest correlated one. As we increase the number of neurons' features the performance metrics all gradually increase even if not by much and start to plateau when at 40-50 neurons with no to little change between the 50,100 and full 123 neurons. We can further investigate looking at the weights of the classifier fitted on all the neurons which can be more easily interpreted thanks to the standardization during the preprocessing. The interpretation of the weights in this case differ from linear regression since here the outcome is a probability between 1 and 0. The weights do not influence the probability with a linear relationship making it more difficult to interpret the values. Nevertheless we can observe that just 38 neurons out of 123 have an absolute weight higher than a threshold of 0.2 suggesting, in line with the previous results, that just about 40 neurons are used by the model to make a reasonable enough prediction.

The same binary classifier is then implemented with the features from all the neurons' activities to predict the other behavioral binary variable. In Figure 4.10 we can see the results with resting

achieving similar metrics performance, abdominal pushing remarkably higher metrics performance then the walking prediction, respectively. Both anterior and posterior grooming performance are heavily affected by the significantly lower number of binary 1 labels in the two sets compared to the 0 and the classifier almost never predicts the 1 class. This was partially expected as especially the posterior grooming behavior is very rare. A similar case may happen with abdominal pushing as we see that in general we have very few 1 labels. As a result we would say that in those cases our logistic regression classifier does not show much robustness.
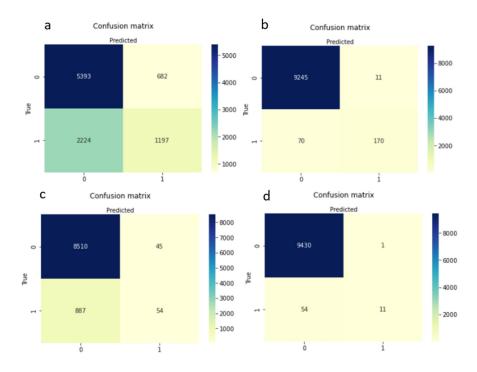
Figure 4.10: *Confusion matrix for the predictions with all of the neurons as inputs and respectively the resting (a), abdominal pushing (b), anterior grooming (c) and posterior grooming (d). Binary class prediction with logistic regression. Detailed performance metrics in* Appendix C.

4.3.2 Predicting multiple behaviors

A multi-class logistic regression classifier was then implemented to predict, based on all the neurons' activities, all the behaviors at once. As we can observe in Figure 4.11 the resulting prediction quality is very poor and worse than the case of the binary prediction for the "walking" variable.
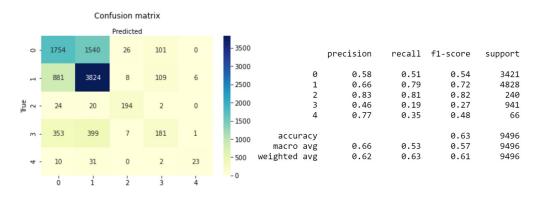
Figure 4.11: *Performance metrics and confusion matrix for the multi-class prediction with logistic regression.*

Concerning the prediction in this case, when filtering the weights with the same 0.2 absolute threshold, we observe higher number of relevant weights in order of about 70-100 depending on the class. This may suggest that several individual neurons contribute to the classification of multiple behavior but in general more individual neurons are used for the prediction.

A more advanced neural network classifier is used to classify the multiple behaviors. A multi-layer perceptron (MLP) neural network was trained with all the neurons' data and achieved substantially higher performance than the logistic regression case in the prediction's quality as observed in Figure 4.12. This is expected as the logistic regression is a rather simple model more suitable for classification of a low number of categorical features while the MLP is a more complex model that is able to better account for the complexity of the data provided and to learn linear models. Despite not requiring high computational power, having weights with straightforward interpretability and being less sensitive to feature scaling, logistic regression is vulnerable to overfitting and cannot solve non-linear problems outside of the additional problem previously mentioned.
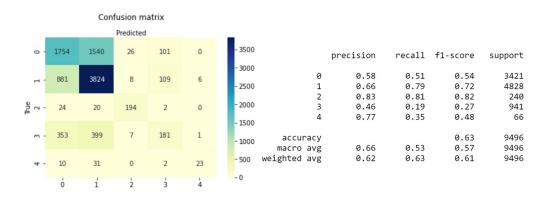
Figure 4.12: Performance metrics and confusion matrix for the multi-class prediction with MLP.

To be precise, logistic regression is actually a particular sub-case of the neural network classifier with one hidden node with the identity activation function, and one output node with zero bias and logistic sigmoid activation. Therefore MLP cannot achieve in principle worst results than logistic regression. However MLP, as other neural networks, requires a high number of hyper-parameters to be tuned for optimal performance based on non-convex loss function with multiple local minima and is very sensitive to feature scaling. Finally compared to regression techniques, it is quite difficult to interpret the weights of the MLP model, as for example understanding which neurons were primarily used for the prediction.

4.4 Classifying behavior from behavioral data

The behavior of the fly was found manually, labelling each frame with the corresponding action. This is a long and tedious work that is prone to errors. An automated behavior classification was attempted based on the fly's joint angles. The joint angles were chosen over joint 3D pose because joint angles are defined in the fly's frame of reference and are independent of the position of the fly. For example, two leg movements that only differ in the Coxa-yaw angle, will result in different pose of the Tarsus but the Tarsus-pitch angle will be the same in both movements (see Figure 4.13).
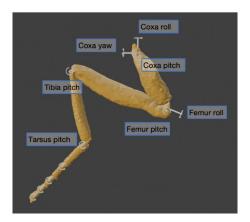
Figure 4.13: Joint angle names. Taken from the project guidelines.

Temporal information is needed in order to analyse behavior and can be added with the wavelet transform, as described in subsection 3.7. This transform adds frequency dimensions to the data and gives information about the frequency of each signal. The joint angle frequencies depend on the behavior of the fly hence features added by the wavelet transform will differ for time samples with different behaviors. The figures in Appendix D show the frequency components of each joint angle of the right front leg, separated to when the fly is resting and walking. For some joint angles, such as the Femur, the signal frequencies are different when the fly is resting and walking, as seen in Figure 4.14.
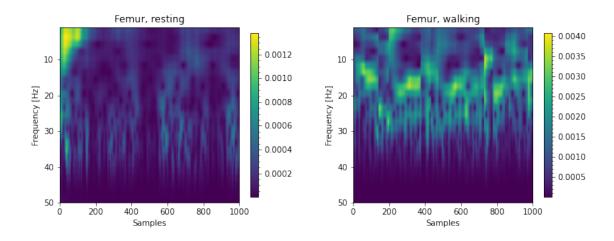
Figure 4.14: Wavelet transform results for the Femur of the right front leg. The plots present time samples for which the fly rest and walk, respectively.

On the basis of this observation, a PCA was performed in the hope of classifying behavior from the wavelet augmented data. Figure 4.15 presents the PCA projection onto the first two principal components. This PCA projection does not separate datapoints into different behaviors. Even after many trials with different parameters, the method described here did not result in the possibility to classify behavior from joint angles. Another approach should be taken to explore this classification.
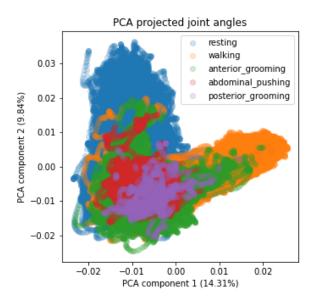
Figure 4.15: *Joint-angle augmented datapoints projected onto the first two principal components.*

5 Discussion

Through the data analysis done in this project, groups of neurons were found to have similar activities. Having multiple signals conveying similar information could be crucial for an organism, increasing robustness to noise. If an action was controlled by only one neuron, even a small signal perturbation could result in an unwanted action. Controlling an action with multiple neurons ensures that perturbation in one neuron's signal will not affect the final action done. The clustering of the neurons depends on the clustering method used and its hyper-parameters. More analysis could be done to improve the clustering by optimizing each method's parameters and more rigorously comparing the two methods.

Looking at correlation between individual neurons and binary behavior signals, we could identify different sets of neurons more likely to conserve information encoding for a particular behavior (either positively or negatively). This exploratory data analysis part was then used to improve the regression, providing a sorted neurons list with deceasing importance related to each behaviors. Indeed, sometimes regression is sensitive to the number of features provided as input; reducing the input data set allows to focus more on the provided information. Correlation was hence helpful to target desired behavior with the neurons that were considered important for each behavior.

Concerning the behavior classification from neuronal activity different methods were explored, highlighting their advantages and disadvantages. Particularly when predicting with a more advanced classifier as a neural network, it is possible to reliably classify multiple behaviors from the neural activities. In the multi-class task, implementing the logistic regression generated relatively poor results with about 70-100 neurons per class contributing to the classification. On the other hand, when dealing with a single behavior prediction, a simple logistic classifier may reach acceptable performance relying on about 40-50 neurons.

A Clustered Neurons correlation

The correlation coefficients between neurons of the same cluster are presented in the figures below. Larger figure can be seen in the code provided with this report.
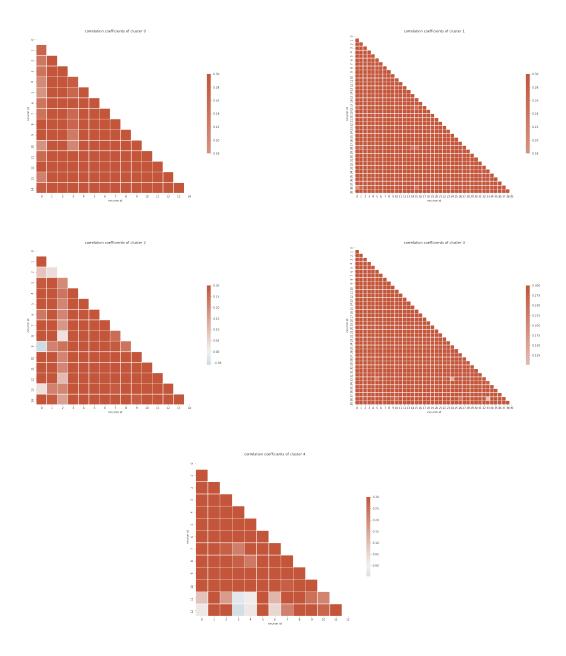
Figure A.1: *Clustering coefficients for each neuron cluster, darker color corresponds to higher positive correlation. Clustered with GMMs.*
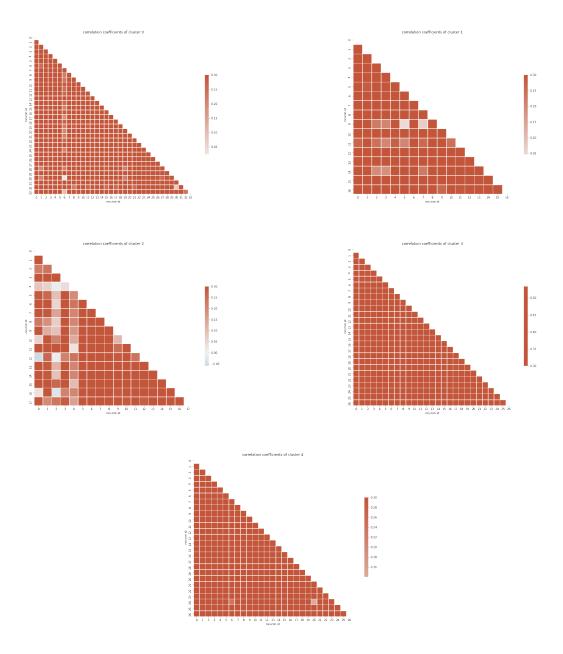
Figure A.2: *Clustering coefficients for each neuron cluster, darker color corresponds to higher positive correlation. Clustered with k-means clustering.*

B   Identifying correlations of individual neurons
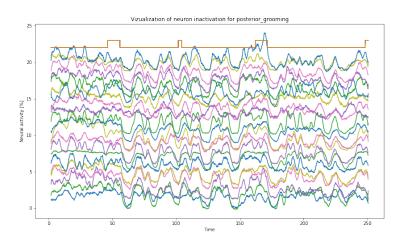
Figure B.1: *Most activated neurons for posterior grooming, with binary signal indicating behavioral label (posterior grooming is 1)*

Figure B.2: *Identification of neurons groups based on individual correlation between neurons and joint angles.*
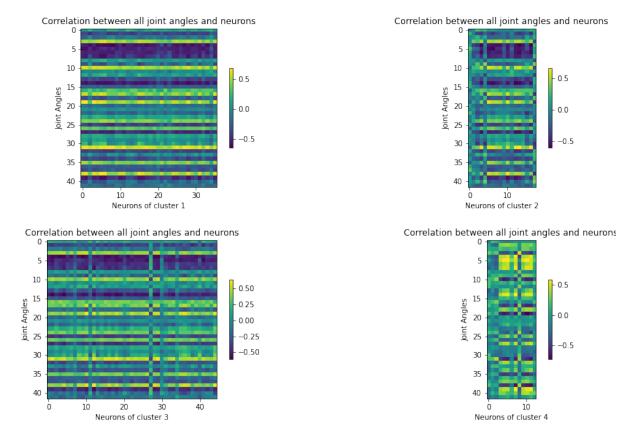
Figure B.3: *Correlation coefficients between neurons and joint angles for each clusters found based on PCA analysis.*

C   Classifying behavior from neuronal activity

The following is the list of the indexes of the neurons sorted in decreasing order of their correlation coefficient with the "walking" binary variable computed in subsubsection 4.2.1: [21, 58, 82, 92, 14, 38, 51, 12, 10, 48, 118, 105, 11, 116, 91, 34, 85, 114, 115, 22, 23, 5, 32, 20, 121, 26, 120, 79, 99, 101, 96, 8, 97, 73, 33, 47, 25, 9, 36, 57, 29, 100, 93, 28, 117, 35, 60, 56, 90, 68, 49, 75, 4, 62, 46, 102, 16, 88, 65, 54, 106, 78, 17, 103, 89, 63, 41, 24, 77, 107, 15, 83, 84, 71, 31, 1, 42, 39, 94, 0, 74, 76, 70, 37, 6, 66, 30, 72, 13, 18, 108, 44, 3, 50, 104, 69, 86, 111, 43, 113, 40, 2, 19, 81, 55, 98, 87, 53, 112, 80, 52, 110, 45, 119, 109, 7, 27, 61, 59, 95, 67, 64, 0]

The following figures present the confusion matrixes together with the performance metrics of the prediction made with the different classifiers ordered as presented in the report.

a	precision	recall	f1-score	support
0	0.57	0.61	0.59	4668
1	0.60	0.56	0.58	4828
accuracy			0.58	9496
macro avg	0.58	0.58	0.58	9496
weighted avg	0.58	0.58	0.58	9496

b	precision	recall	f1-score	support
0	0.59	0.56	0.58	4668
1	0.60	0.63	0.61	4828
accuracy			0.60	9496
macro avg	0.60	0.60	0.60	9496
weighted avg	0.60	0.60	0.60	9496

c	precision	recall	f1-score	support
0	0.63	0.65	0.64	4668
1	0.65	0.63	0.64	4828
accuracy			0.64	9496
macro avg	0.64	0.64	0.64	9496
weighted avg	0.64	0.64	0.64	9496

Confusion matrix (a): True 0 → Predicted 0: 2841, Predicted 1: 1827; True 1 → Predicted 0: 2123, Predicted 1: 2705

Confusion matrix (b): True 0 → Predicted 0: 2627, Predicted 1: 2041; True 1 → Predicted 0: 1789, Predicted 1: 3039

Confusion matrix (c): True 0 → Predicted 0: 3051, Predicted 1: 1617; True 1 → Predicted 0: 1802, Predicted 1: 3026

d	precision	recall	f1-score	support
0	0.66	0.66	0.66	4668
1	0.67	0.67	0.67	4828
accuracy			0.66	9496
macro avg	0.66	0.66	0.66	9496
weighted avg	0.66	0.66	0.66	9496

e	precision	recall	f1-score	support
0	0.69	0.67	0.68	4668
1	0.69	0.70	0.70	4828
accuracy			0.69	9496
macro avg	0.69	0.69	0.69	9496
weighted avg	0.69	0.69	0.69	9496

f	precision	recall	f1-score	support
0	0.69	0.68	0.68	4668
1	0.69	0.70	0.70	4828
accuracy			0.69	9496
macro avg	0.69	0.69	0.69	9496
weighted avg	0.69	0.69	0.69	9496

Confusion matrix (d): True 0 → Predicted 0: 3063, Predicted 1: 1605; True 1 → Predicted 0: 1580, Predicted 1: 3248

Confusion matrix (e): True 0 → Predicted 0: 3116, Predicted 1: 1552; True 1 → Predicted 0: 1425, Predicted 1: 3403

Confusion matrix (f): True 0 → Predicted 0: 3157, Predicted 1: 1511; True 1 → Predicted 0: 1450, Predicted 1: 3378
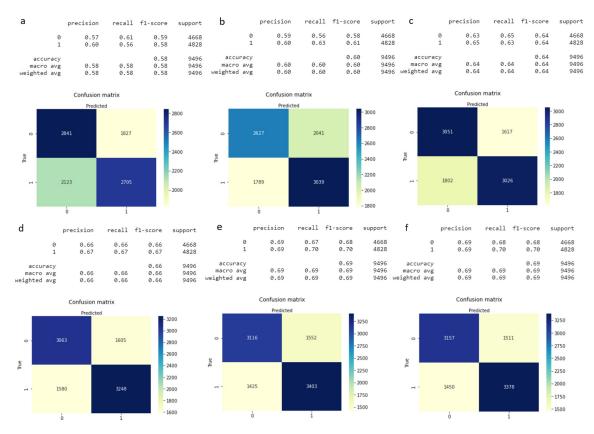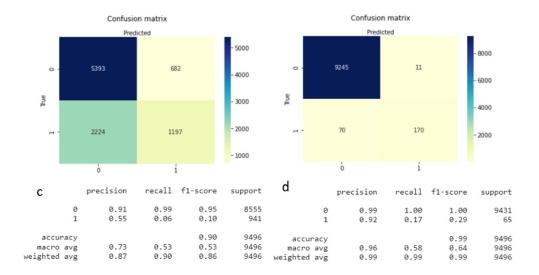
Figure C.1: *Performance metrics and confusion matrix for the predictions with 1 (a), 2 (b), 25 (c), 50 (d), 100 (e) and all of the neurons (f) as regressors. Same prediction type as Figure 4.9.*
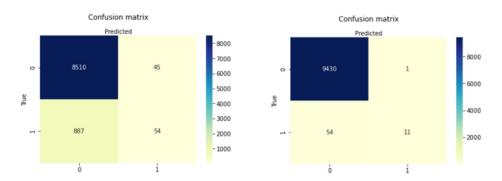
a	precision	recall	f1-score	support	b		precision	recall	f1-score	support
0	0.71	0.89	0.79	6075		0	0.99	1.00	1.00	9256
1	0.64	0.35	0.45	3421		1	0.94	0.71	0.81	240
accuracy			0.69	9496		accuracy			0.99	9496
macro avg	0.67	0.62	0.62	9496		macro avg	0.97	0.85	0.90	9496
weighted avg	0.68	0.69	0.67	9496		weighted avg	0.99	0.99	0.99	9496

Confusion matrix

Confusion matrix

c	precision	recall	f1-score	support	d		precision	recall	f1-score	support
0	0.91	0.99	0.95	8555		0	0.99	1.00	1.00	9431
1	0.55	0.06	0.10	941		1	0.92	0.17	0.29	65
accuracy			0.90	9496		accuracy			0.99	9496
macro avg	0.73	0.53	0.53	9496		macro avg	0.96	0.58	0.64	9496
weighted avg	0.87	0.90	0.86	9496		weighted avg	0.99	0.99	0.99	9496

Confusion matrix

Confusion matrix

Figure C.2: *Performance metrics and confusion matrix for the predictions with all of the neurons as inputs and respectively the resting (a), abdominal pushing (b), anterior grooming (c) and posterior grooming (d). Same prediction type as Figure 4.10.*

D   Frequencies of behavioral data

The following figures present the features added by the wavelet transform to each joint angle of the right front leg, separated to when the fly is resting and walking. The wavelet results for some joint angles clearly differ between the two behavior types.
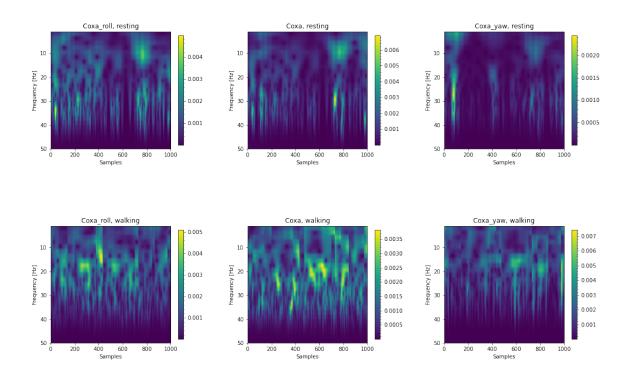
Figure D.1: *Wavelet frequency features, for the first three joint angles of the right front leg, when the fly has different behaviors. First row: resting. Second row: walking.*
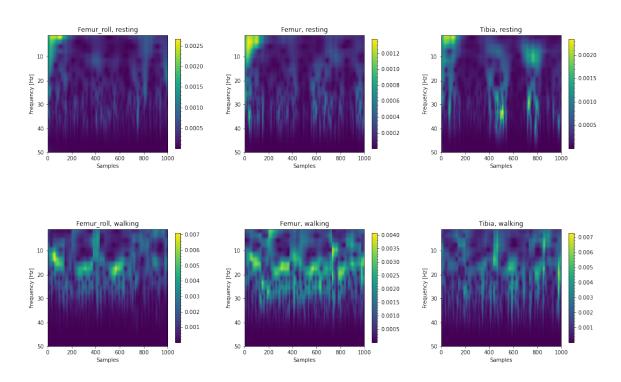

Figure D.2: *Wavelet frequency features, for the last three joint angles of the right front leg, when the fly has different behaviors. First row: resting. Second row: walking.*

References

[1] Salil S. Bidaye, Christian Machacek, Yang Wu, and Barry J. Dickson. Neuronal control of drosophila walking direction. *Science*, 344(6179):97–101, 2014.

[2] Viswanathan M.C. et al. Chen CL., Hermans L. Imaging neural activity in the ventral nerve cord of behaving adult drosophila. 9(4390), 2018.

[3] Sun Y. et al. Chen TW., Wardill T. Ultrasensitive fluorescent proteins for imaging neuronal activity. 499:295–300, 2013.

[4] Morales D. Campagnolo J. Ramdya P. Fua P. Günel S., Rhodin H. Deepfly3d, a deep learning-based approach for 3d limb and appendage tracking in tethered, adult drosophila. 8(e48571), 2019.

[5] Sklearn. sklearn linear model logistic regression.

[6] Sklearn. sklearn neural network mlp classifier.