Hierarchical network analysis of behavior and neuronal population activity

Kevin Luxem (kevin.luxem@dzne.de)
German Center for Neurodegenerative Diseases, Venusberg-Campus 1,
53127 Bonn, Germany

Falko Fuhrmann (falko.fuhrmann@dzne.de)
German Center for Neurodegenerative Diseases, Venusberg-Campus 1,
53127 Bonn, Germany

Stefan Remy[1] (stefan.remy@dzne.de)
German Center for Neurodegenerative Diseases, Venusberg-Campus 1,
53127 Bonn, Germany

Pavol Bauer[1] (pavol.bauer@dzne.de)
German Center for Neurodegenerative Diseases, Venusberg-Campus 1,
53127 Bonn, Germany

Abstract

Recording of neuronal population activity in behaving animals is becoming increasingly popular. Computational markerless annotation tools allow for tracking of animal body-parts throughout the experiment. However, the question remains of how to cross-correlate the extracted behavioral data with the simultaneously acquired neuronal population activity, when both datasets are of high dimensionality. Here we propose a combined analysis, where the behavioral data is clustered into discrete states using a deep learning model and the occurrence of each state is correlated to clusters of neuronal activity. We then model the relationship between behavioral states as a network, where related states are hierarchically grouped while the similarity between their neuronal correlates is maximized. This type of analysis allows for hierarchical exploration of the bidirectional relationship between behavior and its neuronal correlates at different temporal scales.

Keywords: behavior quantification; neuronal population activity; deep learning; clustering; network analysis

Introduction

Understanding the relationship between neuronal activity and behavior is a major challenge in neuroscience. Precise correlation analysis requires deep quantification of both, behavioral features and neuronal activity patterns. Recently, pose-estimation tools such as *DeepLabCut* enabled exact tracking of animal body-parts during experiments (Mathis et al., 2018). Such tools provide a continuous representation of the animal body movement, for which neuronal correlates could be directly detected within the population activity (Gallego, Perich, Miller, & Solla, 2017). However, it is hypothesized that behavior may be represented in form of discrete states that organize hierarchically (Tinbergen, 1951; Wiltschko et al., 2015). Still, the challenge remains at which spatio-temporal scale the discrete states should be resolved (Berman, 2018).

In this work we first show how continuous signals obtained from behavior tracking tools can be grouped into discrete states via clustering of the latent vector obtained from a recurrent neural network autoencoder. We then demonstrate how the resulting behavioral states can be correlated to neuronal activity on different hierarchical levels by considering their similarity in behavior, neuronal activity, or both. Finally, we demonstrate our analysis on a dataset obtained from a mouse running on a linear treadmill with simultaneous imaging of the hippocampal CA1 region.

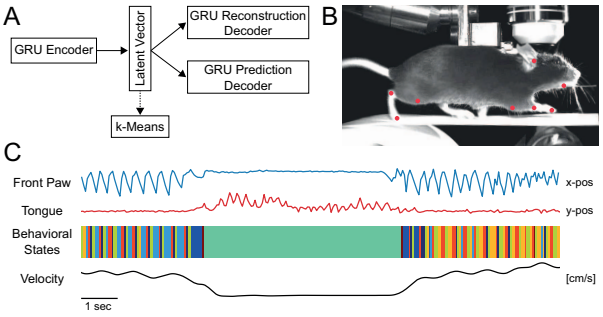

Figure 1: A: Gated Recurrent Units (GRU) sequence autoencoder model. B: Exemplary video frame showing the behaving animal with 8 virtual markers C: (Top) Two exemplary input sequences of tracked joint movements. (Middle) Behavioral states obtained from the latent vector. (Bottom) Measured velocity, which was not part of the autoencoder input.

Methods

Experimental setup

A food deprived mouse was trained to run head-fixed on a textured linear treadmill of 3.6m length. The mouse learned to re-

[1] equal contribution

ceive a liquid reward upon licking at a predefined location once per lap. This led to a repetitive behavior where the mouse was running approximately 3 rounds per minute. Simultaneously, the population activity (GCaMP 7s) of hippocampal CA1 pyramidal neurons was imaged at 15 Hz through an implanted hippocampal window using a two-photon resonant scanning system (Fuhrmann et al., 2015). Synchronized behavioral video monitoring was performed with a camera capturing the side-view of the animal at a frame rate of 25 Hz. An exemplary video frame is shown in Figure 1 (B).

Data preprocessing

The imaging stack was down-sampled to 5Hz and corrected for motion artifacts. Active temporal components were extracted using constrained non-negative matrix factorization (Pnevmatikakis et al., 2016). The factorization resulted in a deconvoluted $\Delta F/F$ time series for each detected component. We then extracted the onset of each peak from the time series via threshold-crossing and assigned a weight according to the peak maximum. If multiple peaks occurred within a transient, the weight of each onset was set to the difference of the peak to the decay of the preceding peak, which was extrapolated by an exponential function.

For behavioral pose extraction, virtual markers were placed on eight body-parts in 150 uniformly picked video frames and a residual neural network was trained to assign the virtual markers for the entire video sequence (Mathis et al., 2018).

Sequence autoencoder

In order to learn the structure of the temporal representation of a behaving animal we built a recurrent neural network autoencoder consisting of Gated Recurrent Units (GRU) (Cho et al., 2014). The goal of the autoencoder was to learn a d-dimensional latent vector $\Lambda_t \in \mathbb{R}^d$. The input sequence $\mathcal{X}_t \in \mathbb{R}^{2n \times T}$ carries the (x,y) coordinates of n marker positions for the video sequence $[t \dots t+T]$. The sequence autoencoder then learns the mapping,

$$f_{enc} : \mathcal{X}_t \rightarrow \Lambda_t. \tag{1}$$

Our approach is motivated by (Srivastava, Mansimov, & Salakhudinov, 2015), where a long short-term memory (LSTM) composite encoder-decoder model was proposed for unsupervised video representation learning. However, in order to make training more efficient we chose to use GRUs instead of LSTMs in every layer of our autoencoder model (Chung, Gulcehre, Cho, & Bengio, 2014). The encoder (1) is then trained to generate a latent vector Λ_t that is passed to two one layer GRU decoder. The first one reconstructs the sequence $\mathcal{X}_t$ and the second predicts the sequence $\mathcal{X}_{t+T}$.

The autoencoder architecture is depicted in Figure 1 (A) and was trained using the Adam optimizer (Kingma & Ba, 2015) with a fixed learning rate of 0.001 and with the mean squared error as the objective function for both, reconstruction and prediction. We used Rectified Linear Units (ReLU) activation functions in every layer. Backpropagation is performed on the combined loss and the whole network is trained on a single Nvidia 1080ti GPU. All computing was done with Pytorch (Paszke et al., 2017).

Behavioral states

To identify the behavioral state space $B = \{b_1, \dots, b_K\}$ in our dataset we computed the latent vector Λ_t for every data point t. As the full experiment contains N frames, the resulting feature matrix $\mathcal{F}$ is of dimensionality $d \times (N-T)$. We performed k-means clustering on $\mathcal{F}$ to identify K behavioral states. Figure 1 (C, Middle) shows an exemplary state sequence.

We modeled the transitions between behavioral states as a discrete-time Markov chain where the transition probability into a future state is only dependent on the present state. This results in a $K \times K$ transition probability matrix $\mathcal{T}$, with the elements

$$\mathcal{T}_{lk} = P(b_k | b_l) \tag{2}$$

being the transition probabilities from one state $b_l \in B$ to another state $b_k \in B$.

The Markov chain (2) can be represented as a directed graph $\mathbb{G}$ consisting of nodes $v_1 \dots v_K$ connected by edges with the transition probability $\mathcal{T}_{lk}$. Additionally, the size of each node corresponds to the total occurrence of the behavior state throughout all N video frames. The graph $\mathbb{G}$ is visualized in Figure 3 (left).

Clustering of neuronal data

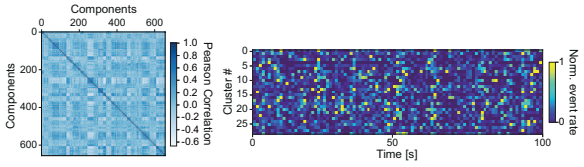

Figure 2: Left: Correlation matrix for 640 components sorted into 30 clusters. Right: An exemplary sequence of activity for each cluster.

We computed the pairwise correlations between activity traces of all active components resulting in a correlation matrix R. We then clustered R into a block-diagonal matrix Z using spectral co-clustering (Dhillon, 2001), which effectively grouped all components into M biclusters with similar values in corresponding rows and columns of R. The choice of M must be made individually based on the structure of the particular neuronal recording. We have then reduced the dimensionality of each cluster using factor analysis into a shared component (Everett, 1984), that has been shown to be particularly consequential for behavior (Kohn, Coen-Cagli, Kanitscheider, & Pouget, 2016). An exemplary sequence of activity for each cluster is shown in Figure 2.

Results

During the experiment we imaged 640 active components from the hippocampal CA1 region, that were grouped into

$M = 30$ neuronal clusters. We have set the sliding window size $T = 25$ (1s of video data), and the dimensionality of the latent vector Λ_t to $d = 20$. Hence, the model achieved a compression ratio of $\frac{16 \times 25}{20} = 20$. To identify behavioral states we chose $k = 8$ in our k-means clustering assignment, based on the Elbow method. To qualitatively validate the outcome of the clustering we inspected the original video frames for every obtained cluster. Quantitative validation was made based on the original velocity signal from the treadmill experiment, which was not used to train the sequence autoencoder. Figure 1 (C) shows that a subset of behavioral states is only active during running while a different subset of states is active during resting or reward taking periods.

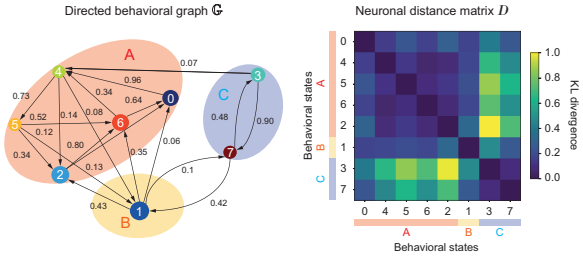

Figure 3: Left: Directed behavioral graph created from the state space B. For visualization purposes only the edges with transition probability larger than 0.05 are displayed. Right: Neuronal distance matrix D which shows the Kullback-Leibler divergence for all combinations of behavioral states. Communities A, B, C are obtained from clustering of the hierarchical representation of $\mathbb{G}$.

Combining neural and behavioral representation

To understand which clusters of neuronal activity are active during a behavioral state and vice versa, we performed a combined analysis. First, we aligned the behavioral state sequence to match the temporal resolution of the neuronal data recording. We then created a $M \times K$ combined states matrix S containing the averaged normalized event rate for each neuronal cluster and every behavioral state b_k. Next, we computed the dissimilarities between rows of S based on the Kullback-Leibler divergence,

$$KL(p||q) = \sum_{m=1}^{M} p_m \log \frac{p_m}{q_m}, \tag{3}$$

where $p, q \in \{1 \dots K\}$ are modeled as the probability distribution of the neuronal clusters from two behavioral states. Applying (3) for all combinations of behavioral states results in a $K \times K$ neuronal distance matrix D.

Graph to tree mapping

In order to inspect the relationship between behavior and neuronal activity at different hierarchical levels we transformed $\mathbb{G}$ into a binary tree $\mathbb{T}$. Thus, we have iteratively merged two

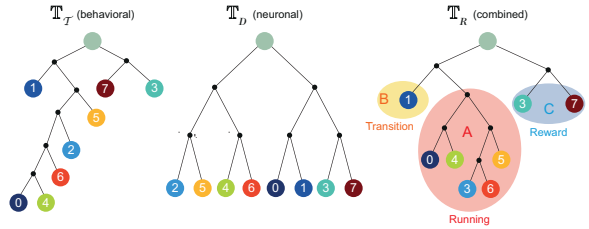

Figure 4: Hierarchical representation of the behavioral graph $\mathbb{G}$ based on three different cost functions. For the tree $\mathbb{T}_R$ communities A, B, C are assigned that have been manually labeled as *running*, *reward* and *transition* phases.

nodes (v_i, v_j) until only the root node v_R is left. To select i and j in each reduction step, we computed a cost function for all combinations of the remaining nodes. Here, we propose three different cost functions which imply different trade-offs,

$$C_{\mathcal{T}} = \max_{i,j} \mathcal{T}_{ij}, \tag{4}$$

$$C_D = \min_{i,j} D_{ij}, \tag{5}$$

$$C_R = \max_{i,j} \left(\sum_{i,j} \frac{\mathcal{T}_{ij}}{D_{ij}} \right). \tag{6}$$

The first cost function (4) merges nodes with the highest transition probability in $\mathbb{G}$, therefore considering their behavioral similarity. The second cost function (5) merges two nodes with the smallest Kullback-Leibler dissimilarity of the neuronal representation. The third cost function (6) considers a ratio of the transition probability and Kullback-Leibler dissimilarity for every node pair. After each reduction step the matrices $\mathcal{T}$ and D are recomputed for rows and columns corresponding to the merged nodes.

Figure 4 shows the resulting trees $\mathbb{T}_{\mathcal{T}}$, $\mathbb{T}_D$ and $\mathbb{T}_R$ obtained via the cost functions (4)-(6). We then used the hierarchical tree representation for community structure detection, as suggested in (Newman, 2010). We could identify three communities in the tree $\mathbb{T}_R$, based on a cut at the second hierarchical level. We then inspected the original video for every community and identified stereotyped behaviors as *running*, *reward* or *transition* phases. For visualization, we have also annotated the obtained communities in Figure 3.

Conclusion

In this paper we proposed an approach for combined analysis of behavior and neuronal population data. We have shown how continuous signals obtained from behavior tracking tools can be converted into discrete behavioral states using clustering of latent vectors obtained from a sequence autoencoder. We have then demonstrated how the resulting behavioral states can be correlated to clustered neuronal population activity via a hierarchical approach. Depending on the cost-function for aggregation of states, our analysis could extract

the organization of behavioral states, as well as structure of the underlying neuronal correlates.

The analysis has been demonstrated on a dataset where a mouse was running on a linear treadmill while receiving liquid reward at a fixed location. The behavioral clustering yielded a total of 8 behavioral states, which we grouped into three communities. In these communities, five states were active during running phases, two during resting or reward taking phases, and one during transitions between the aforementioned phases. Further investigation of the communities could lead to discovery of other sub-communities, e.g. different running patterns.

Explicit graph to tree mapping can be furthermore useful to compare and quantify behavior between different experimental conditions, trials and animals. Given for example two graphs ($\mathbb{G}_1$, $\mathbb{G}_2$) and their respective tree representation ($\mathbb{T}_1$, $\mathbb{T}_2$), it is possible to apply the Tree Edit Distance (Zhang & Shasha, 1989) to compute the dissimilarity between two trees.

The proposed behavioral clustering method is sensitive to the choice of the time window T and the clustering parameter k. Furthermore, we believe that the application of dynamic time-warping could improve the correlation between behavioral states and neuronal activity, as suggested by (Lawlor, Perich, Miller, & Kording, 2018).

For future work, we aim to generalize the analysis to a wider range of behavioral experiments, including experiments in freely-moving animals.

References

Berman, G. J. (2018). Measuring behavior across scales. *BMC biology*, *16*(1), 23.

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1724–1734).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv*. (1412.3555)

Dhillon, I. S. (2001). Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 269–274). ACM.

Everett, B. (1984). *An Introduction to Latent Variable Models*. Springer Netherlands.

Fuhrmann, F., Justus, D., Sosulina, L., Kaneko, H., Beutel, T., Friedrichs, D., ... Remy, S. (2015). Locomotion, Theta Oscillations, and the Speed-Correlated Firing of Hippocampal Neurons Are Controlled by a Medial Septal Glutamatergic Circuit. *Neuron*, *86*(5), 1253–1264.

Gallego, J. A., Perich, M. G., Miller, L. E., & Solla, S. A. (2017). Neural Manifolds for the Control of Movement. *Neuron*, *94*(5), 978 – 984.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv*. (1412.6980)

Kohn, A., Coen-Cagli, R., Kanitscheider, I., & Pouget, A. (2016). Correlations and Neuronal Population Information. *Annual Review of Neuroscience*, *39*, 237–256.

Lawlor, P. N., Perich, M. G., Miller, L. E., & Kording, K. P. (2018). Linear-nonlinear-time-warp-poisson models of neural activity. *Journal of Computational Neuroscience*, *45*(3), 173–191.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, *21*(9), 1281–1289.

Newman, M. (2010). *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in PyTorch. In *Nips autodiff workshop.*

Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., ... Paninski, L. (2016). Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron*, *89*(2), 285–299.

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using LSTMs. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (Vol. 37, pp. 843–852). PMLR.

Tinbergen, N. (1951). *The study of instinct*. Clarendon Press.

Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., ... Datta, S. R. (2015). Mapping Sub-Second Structure in Mouse Behavior. *Neuron*, *88*(6), 1121–1135.

Zhang, K., & Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, *18*(6).