Detecting encoding-relevant cells in a neural ensemble by weight pruning

Tim Herfurth, Simon Kolb, Raunak Basu, Hiroshi Ito, Tatjana Tchumatchenko
Max Planck Institute for Brain Research, Frankfurt, Germany

INTRODUCTION

Modern technology allows to record from increasingly large numbers of neurons in in-vivo situations. Often, for the analysis of the obtained spiking data one is only interested in those cells whose activity is related to a certain external variable, e.g. sensory stimuli or behavioral observables. In such cases, unrelated cells add noise to the data and increase the data dimensionality without contributing positively to the analysis.
Here, we introduce a method to select only those cells of an ensemble that have the highest predictive power for a given external variable - a type of feature selection with automated discard and without ranking.

THE IDEA

Our starting point is a selection layer that receives one-to-one connections from the input features/neurons. We then make use of classic L_1 regularization and the fact that, for non-negative inputs, a relu activation effectively silences all features with a negative weight to the selection layer. The loss function is designed such that the selection weights are drawn towards a negative value and, at the same time, a cost is added to non-optimal prediction of the chosen variable **y**. A regularization factor determines the balance of both loss terms. Neurons (**X**) that are not important for predicting the variable are assigned negative selection weights during training and are hence pruned and removed.

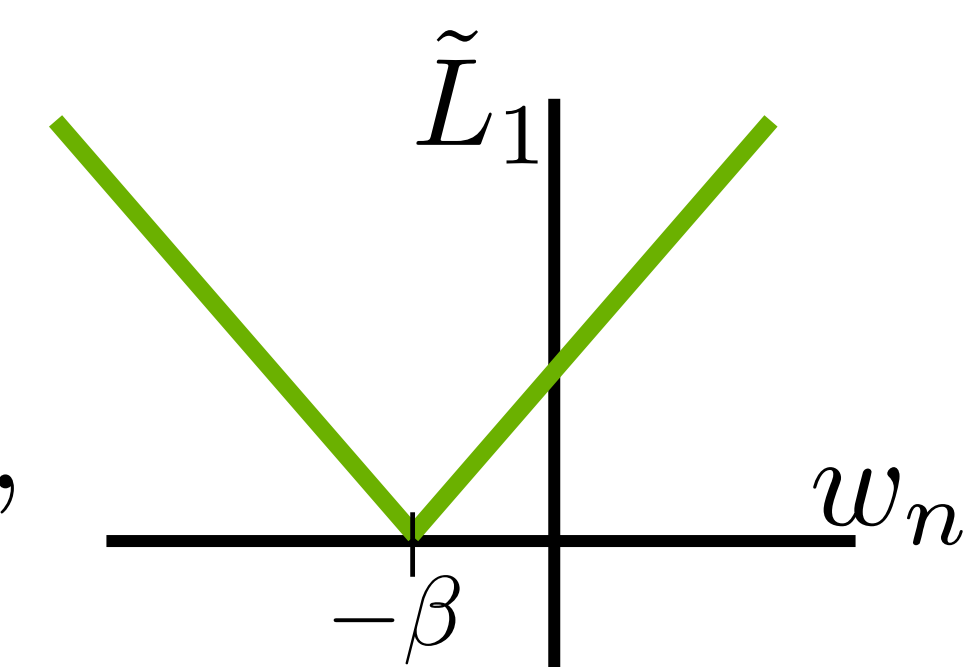all $X_n \geq 0$

$\mathrm{relu}(w_n \cdot X_n) =$

Neural network

$\mathbf{y}(\mathbf{X})$

$L = \tilde{L}_1 + L_{error}$
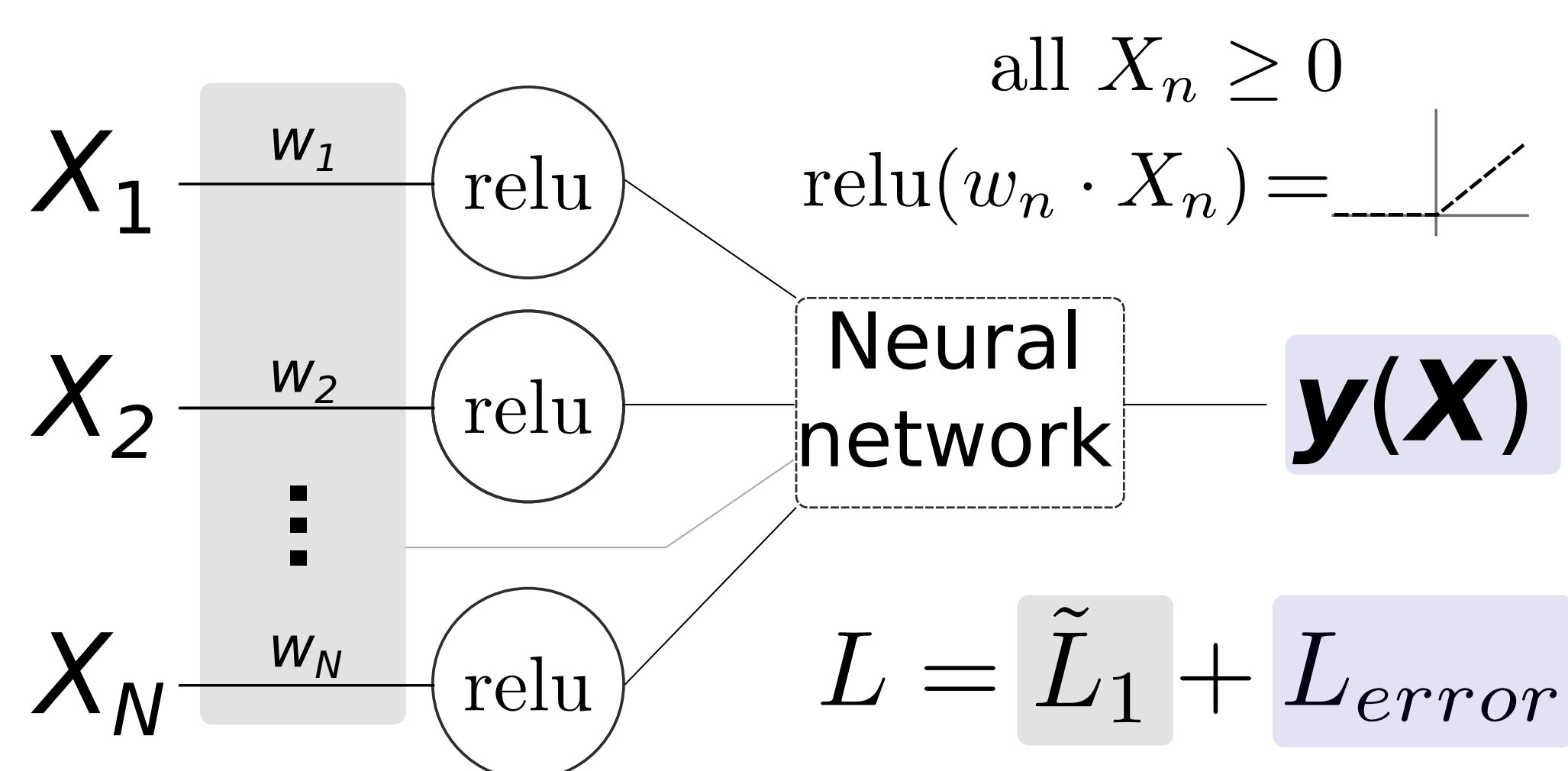
diagonal weight matrix to selection layer:
$w_{nk} = w_n$ if n=k and 0 otherwise

$L_{error}(y(\mathbf{X}), y_{true})$: prediction error

$\tilde{L}_1 = \alpha \sum_n |w_n + \beta|$, with $\beta = 0.05$ here.

SELECTING HIPPOCAMPAL PLACE CELLS

We analyzed 84 preselected neurons of hippocampus CA1 that were recorded during random foranging of a rat (10 min). Neurons are selected as for their predictive power for the current position of the animal (*x,y*-coordinates). Thereby, firing rate time series are fed into a decoding network of LSTM and dense layers. The error function is RSS.
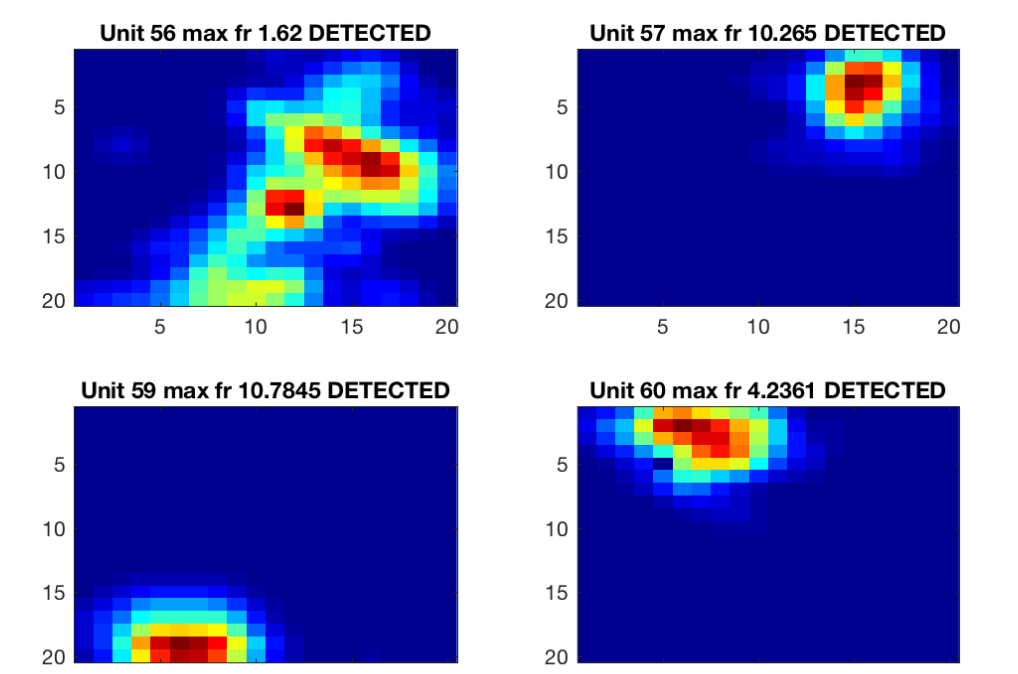
Performance and feature selection as function of α
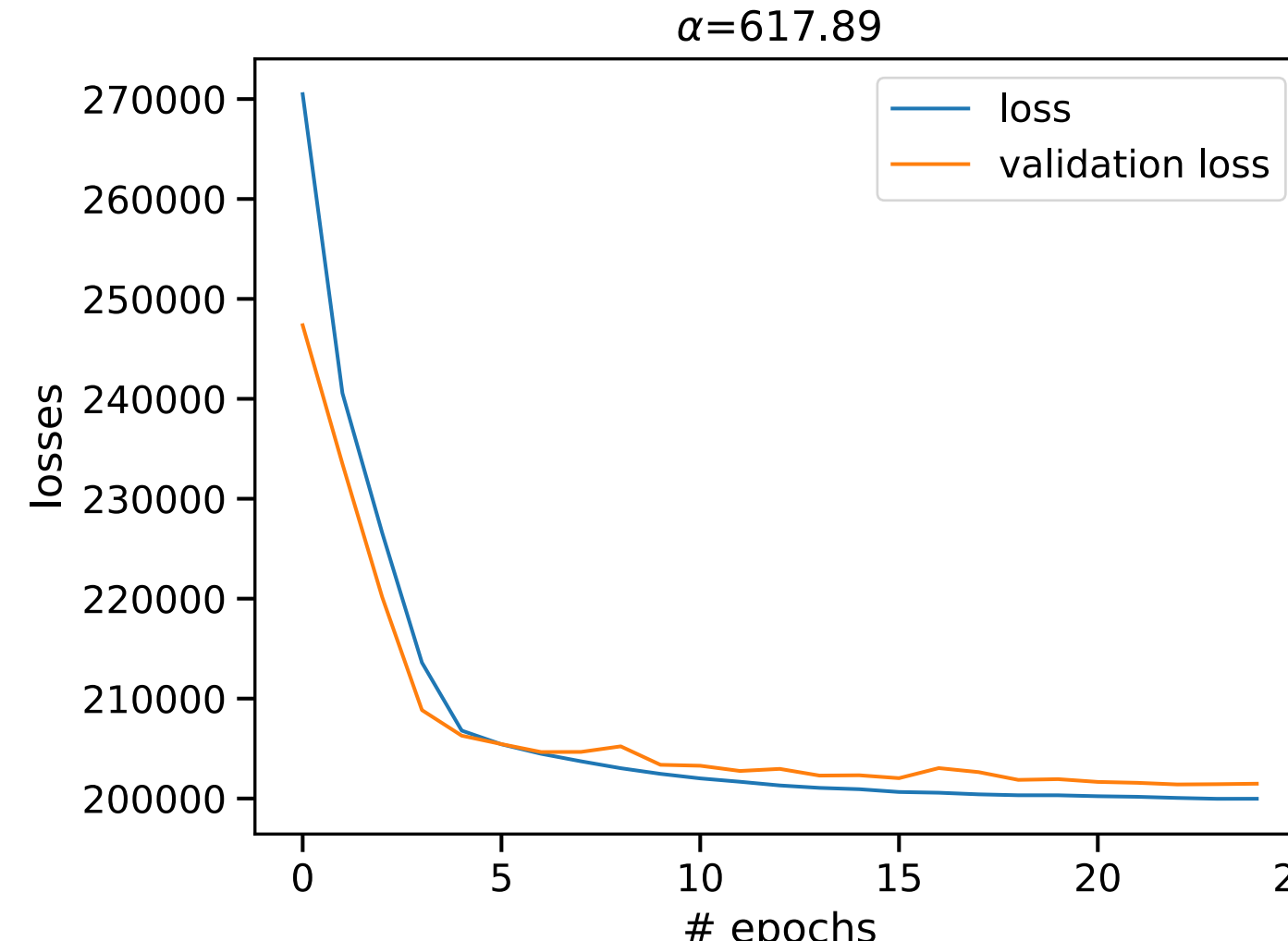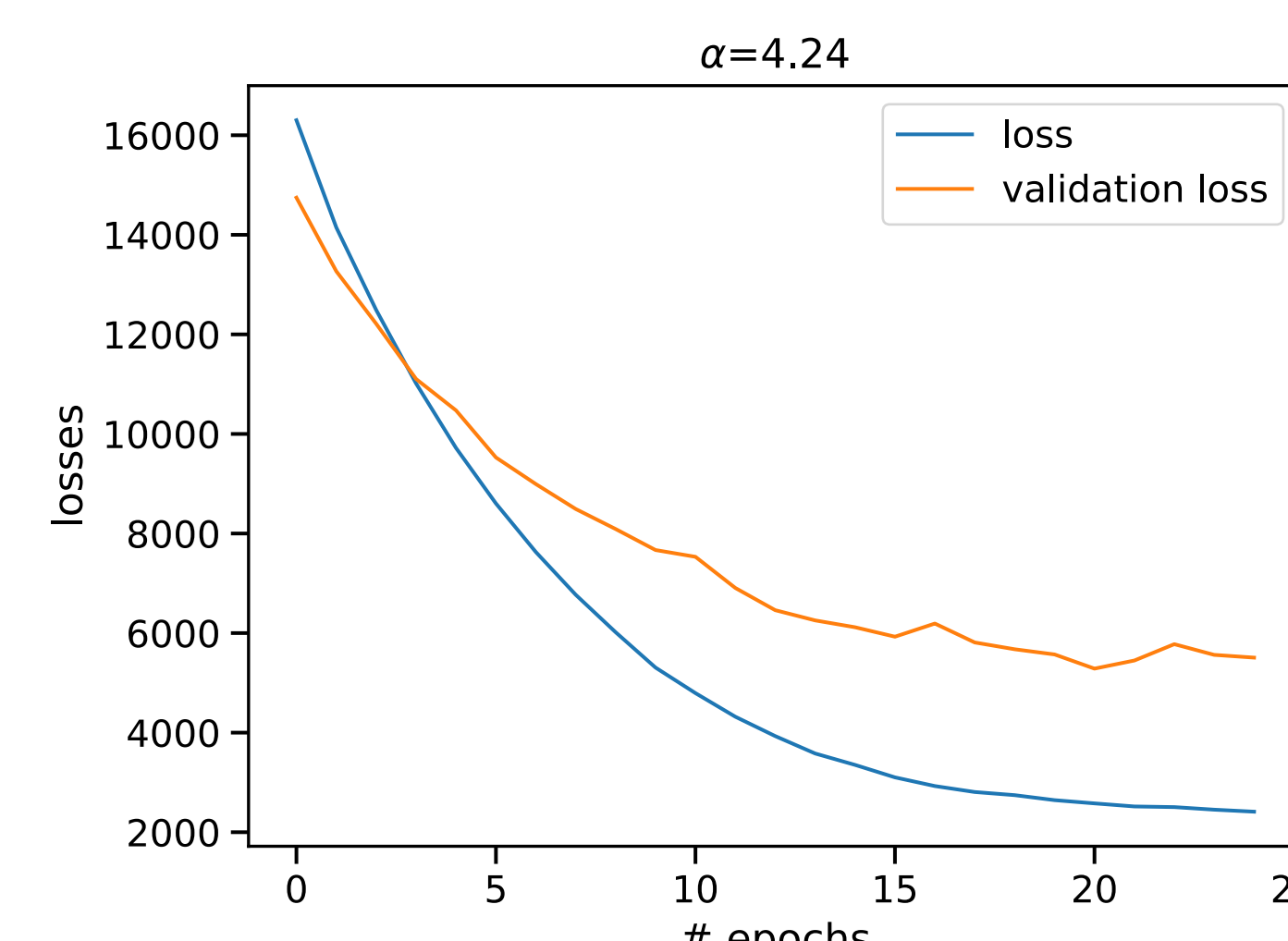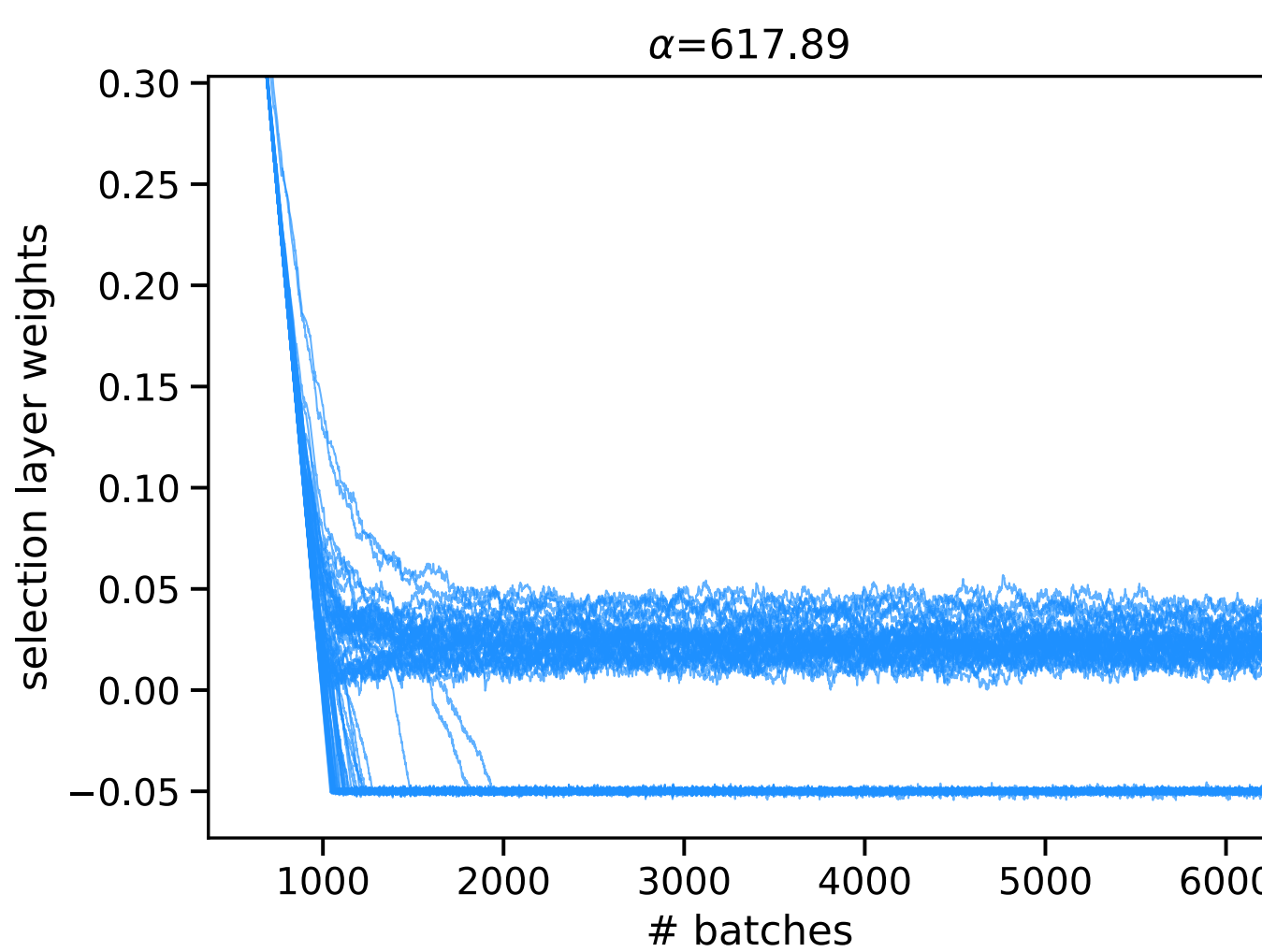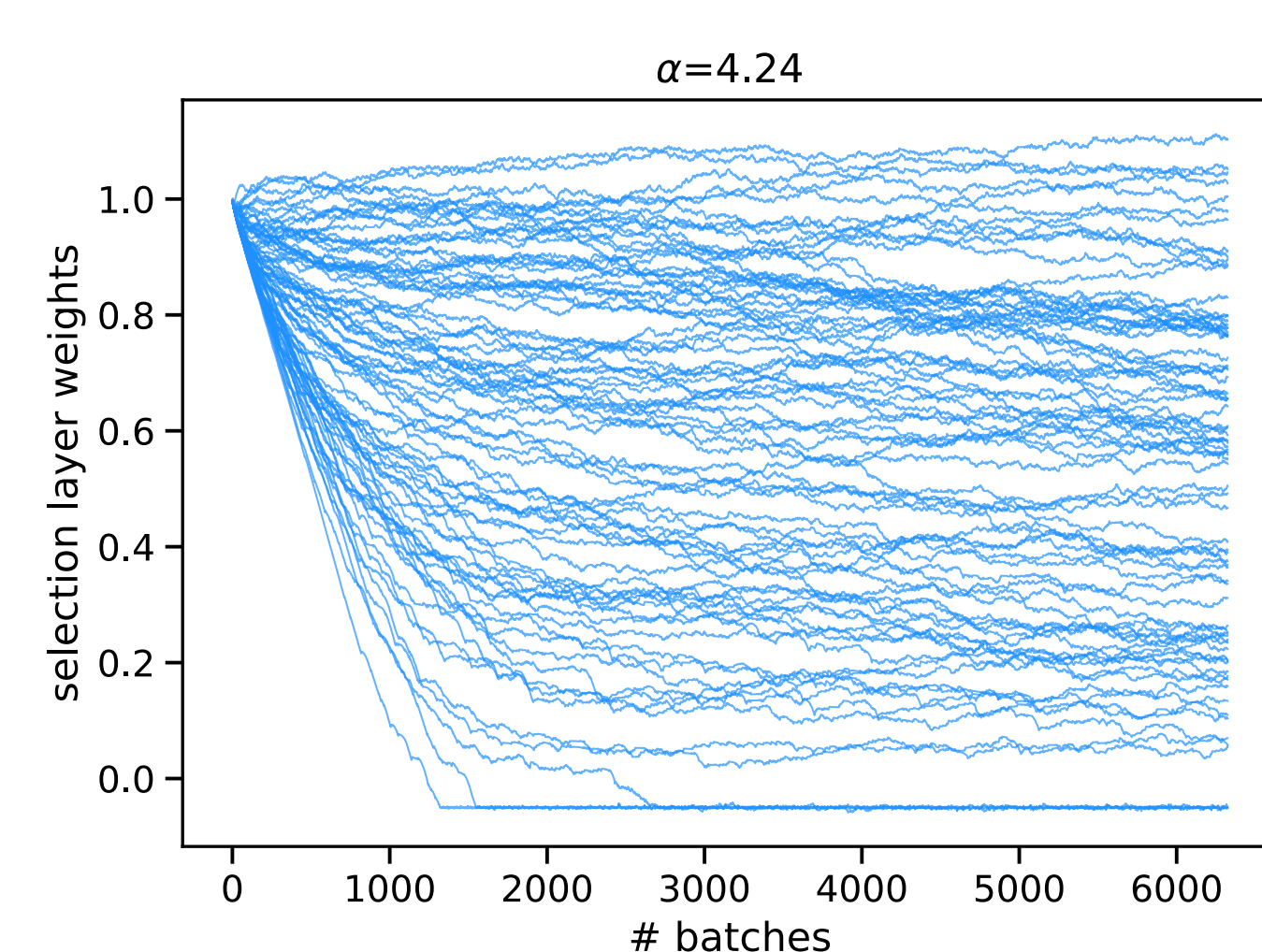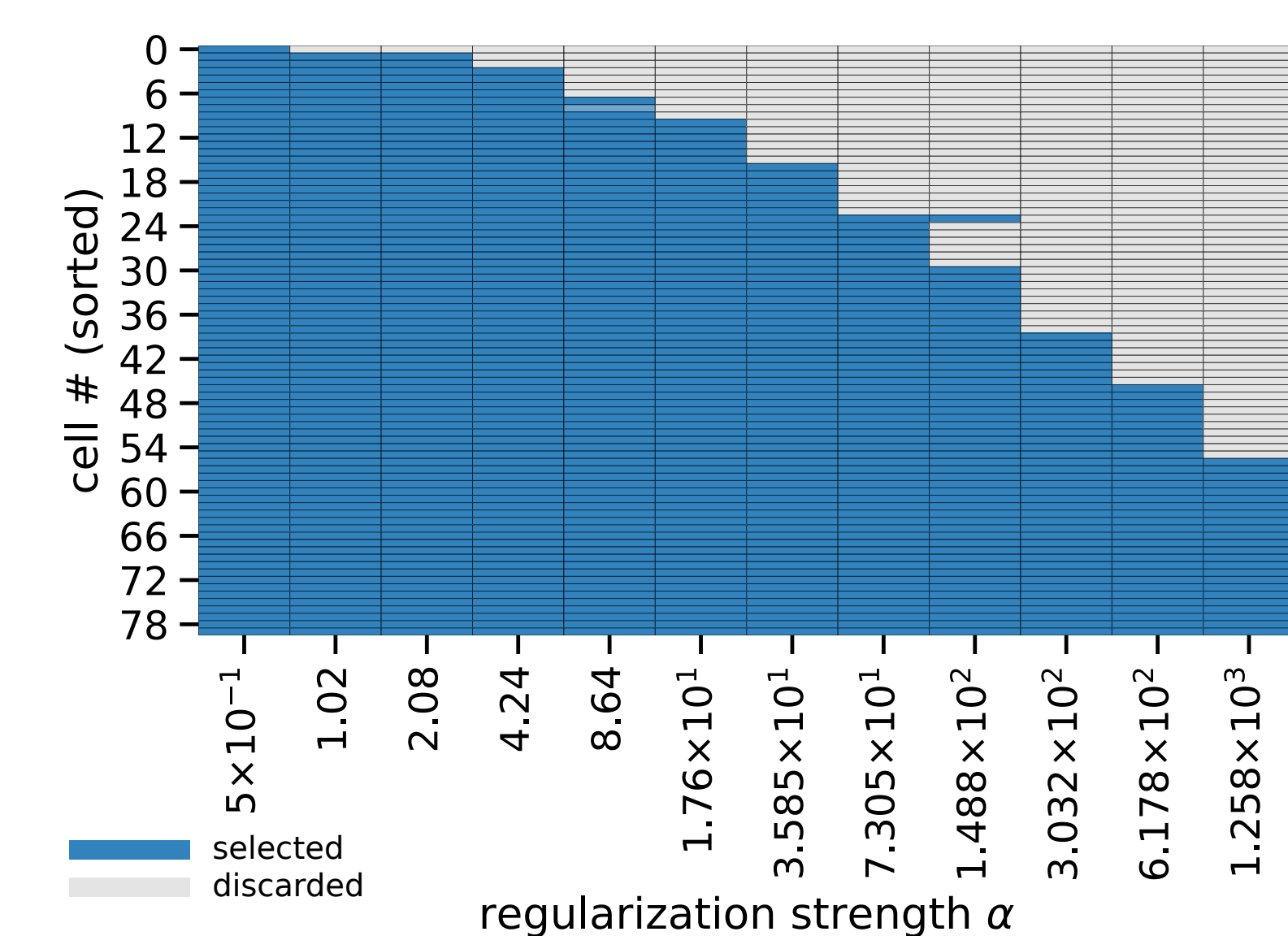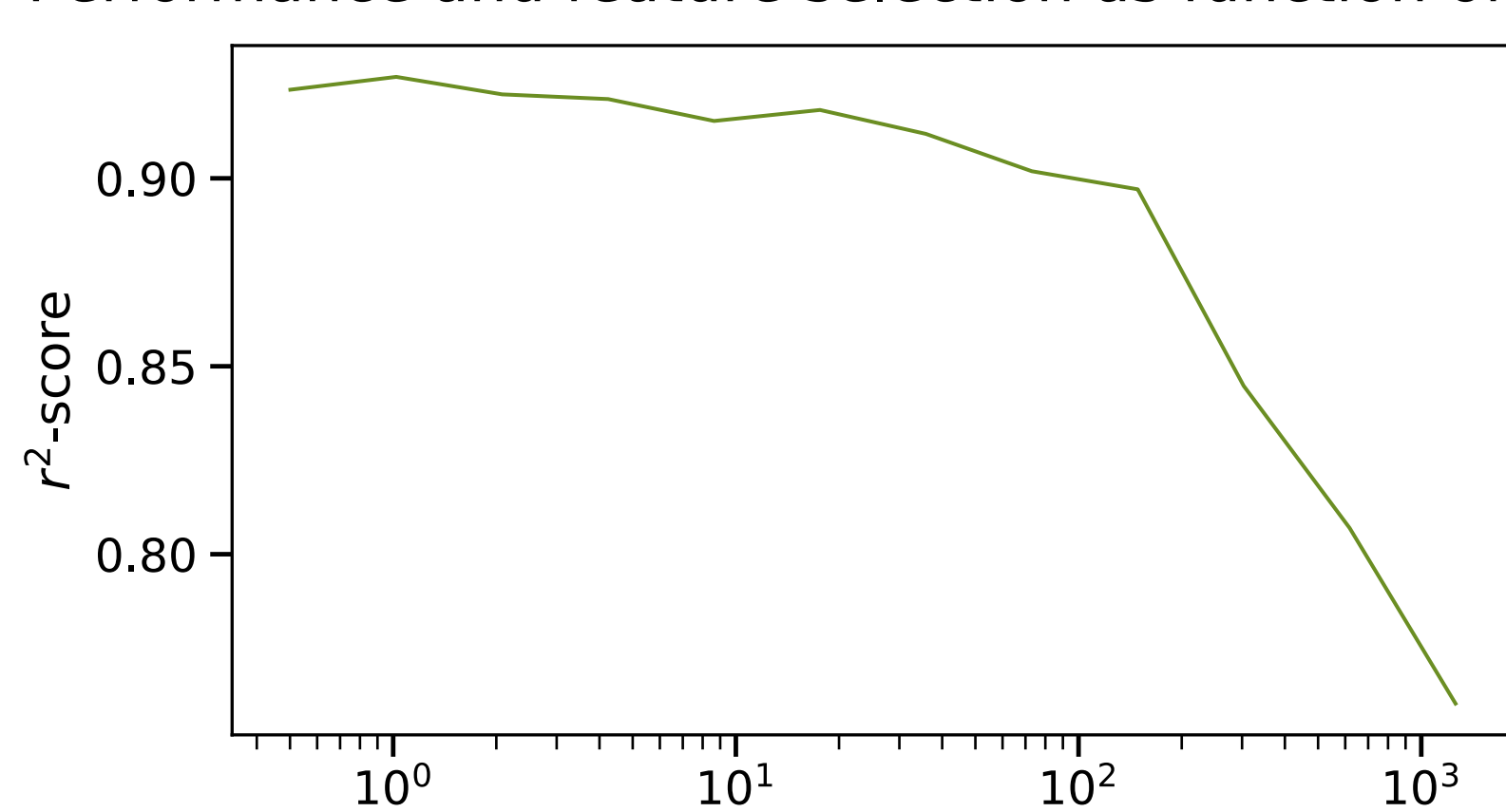
Another example: Classification task for clustered data (3 classes, generated in sklearn). First 10 features are informative, the others are noise.

Performance and feature selection as function of α
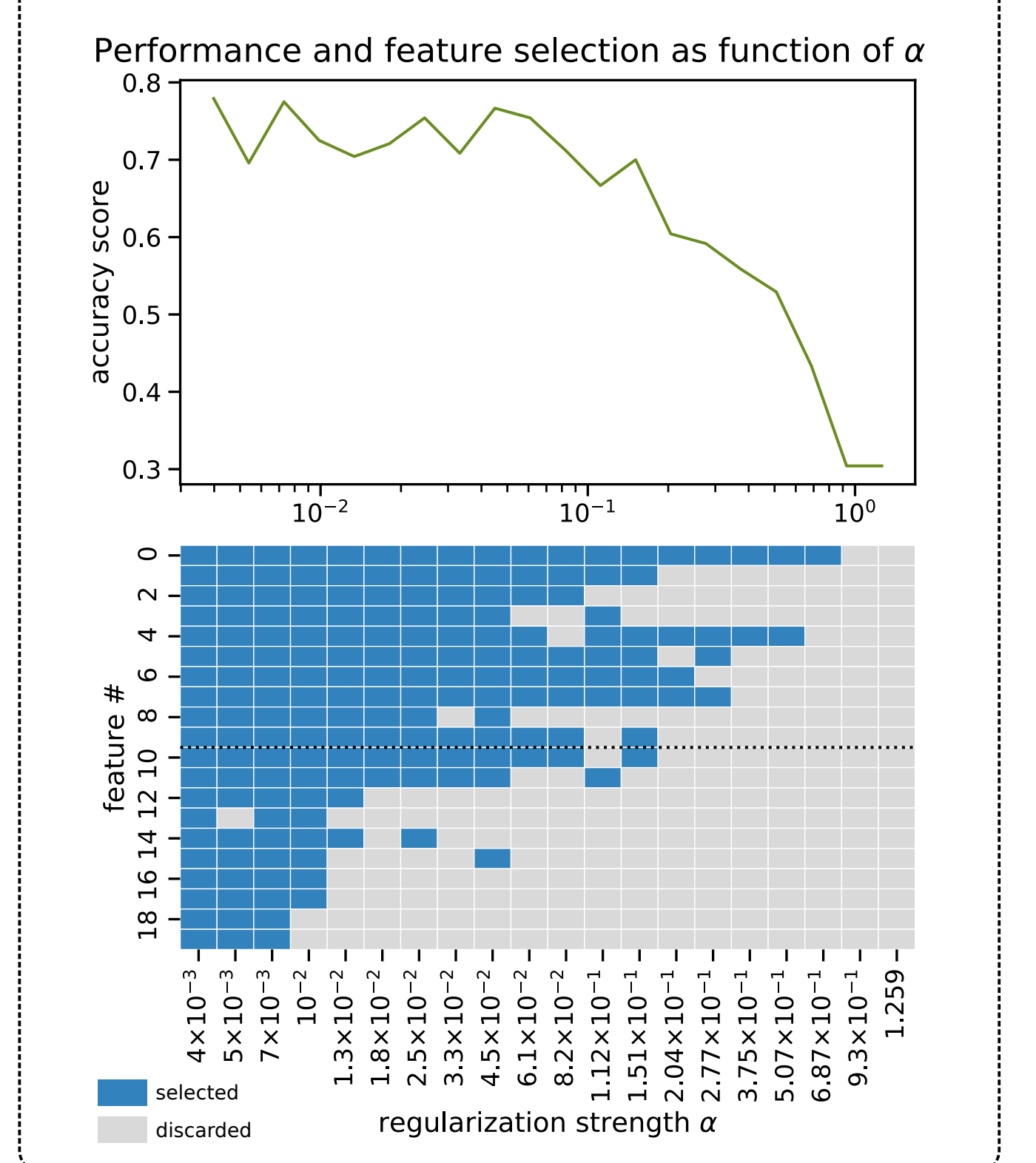
SUMMARY

The selection algorithm seems to work well (for rather structured data). A similar approach has been followed in [1] and refs therein.
Importantly, neurons are directly selected in an all-or-nothing manner and not ranked (in contrast to most algorithms). Thereby, specific, optimal combinations of features are chosen nonlinearly, and the selection is not based on features considered independently. The selection likely also helps generalization.
However, this is work in progress and further evaluation is necessary. We need to address how variable the feature selection is across trials, and how the learning rate and β influence outcomes.

[1] N Vecoven. Master thesis : Feature selection with deep neural networks