Graduate School of Neural Information Processing, Tübingen
Philipp Berens, Alexander Ecker

Task 7: Developing the methods to fit receptive fields

Due date: Monday, June 13, 11:59 AM

Before you start

We will start again with toy data generated from an LNP model neuron of a simple cell to make sure everything works right. The model LNP neuron consists of one orientation selective linear filter, an exponential nonlinearity and a Poisson spike count generator. We look at it in discrete time with time bins of width Δt. The model is:

$$c_t \sim \text{Poisson}(r_t)$$
$$r_t = \exp\left(w^{\mathrm{T}} s_t\right)$$

Here, c_t is the spike count in time window t of length Δt, s_t is the stimulus and w is the receptive field of the neuron. The receptive field w is 15 × 15 pixels and, accordingly, the stimulus s_t is a 15 × 15 pixel image, drawn from an uncorrelated Gaussian distribution.
For computational ease, we reformat the stimulus and the receptive field in a 225 × 1 vector. The function `sampleLNP` can be used to draw data from this model. It returns a spike count vector `c` with samples from the model (dimensions: $1 \times \mathrm{T}/\Delta t$), a stimulus matrix `s` (dimensions: $225 \times T/\Delta t$) and the receptive field `w` (dimensions: 225 × 1).
Download the script called `task7.m`, which contains calls to all the functions you are supposed to implement and which sets the parameters for the different subtasks. Run its parts separately from the editor using the 'Evaluate cell' (Shift + Enter) feature.

Task

1. **Fit instantaneous receptive fields**. In the first part, the receptive field influences the spike count instantaneously just as in the above equations. We will use *Maximum Likelihood* to fit the receptive field. First calculate and implement the log-likelihood function of w and its gradient with respect to w (`logLikLnp`). Write down the log-likelihood of the model as

$$L(w) = \log \prod_t \frac{r_t^{c_t}}{c_t!} \exp(-r_t)$$

and simplify it as far as possible. Then compute the gradient $\frac{dL(w)}{dw}$ (which is vector-valued!) and plug it into an optimization function (e.g. `minimize` from Illias; flip the sign of the log likelihood). This should be done in the function `fitRf()`.
Figure 1: Visualize the true and the estimated receptive field on the same scale using imagesc().

2. **Fit receptive fields with time lag.** In the second part, there is also a temporal component to the receptive field, i.e. the spike rate of the neuron is not instantaneously influenced by the stimulus but by stimulus values in the past. To fit receptive fields, you can keep the formula for the maximum likelihood fit the same, but augment the stimulus space by adding stimulus values from the past as additional dimensions (see slides). Modify `fitRf()` to take care of this.
Figure 2: Estimate the receptive field with 10 time steps and visualize the true and the estimated receptive fields for each time step separately. Use the same color map for all panels.

3. **Decompose the receptive field into its temporal and spatial component.** The receptive field of the neuron used here consists of a spatial component (the Gabor filter) and a temporal component. Both are independent and the resulting spatio-temporal component is thus called separable. As discussed in the lecture, you can use singular-value decomposition to separate these two. Implement the function `sepSpaceTime()` to do so.
   Figure 3: (a) Plot the spectrum of singular values. How many SVs prominently stick out from the rest? (b) Plot the first temporal component and the first spatial component.
4. **Real data**: Download the dataset for this task from Ilias (`rf_data`). The file contains a stimulus matrix (`stim`) in the same format you used before and the spike counts in the corresponding time bins (`spk`). The bin width is 10 ms. Using the methods developed above, fit the receptive fields of the neuron for one time lag at a time (the ML fit is very sensitive to the number of parameters estimated and will not produce good results if you fit the full space-time receptive field for more than two time lags at once). V1 cells typically respond with a latency of 40–50 ms.
   Figure 4: Visualize the raw receptive fields for time lags between 20 and 90 ms.

Tips

- Matrix cookbook for computing derivatives: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf
- To convert a single stimulus frame or the receptive field to its 2D image, use `reshape(w,15,15)`.
- Reading material about singular value decomposition: http://en.wikipedia.org/wiki/Singular_value_decomposition http://xcorr.net/2011/09/20/using-the-svd-to-estimate-receptive-fields/
- Use `minimize.m` as in last week's exercises.