To be completed individually or in groups of two people. **Please be sure matriculation numbers are clearly included at the top of your submission.** Submissions can be handwritten or in LaTeX formatting, but hard-to-read handwritten submissions will not be graded.
Please submit via Ilias. Submissions should be a single PDF document (note that Jupyter notebooks can and should also be downloaded as PDFs, and not submitted as .ipynb files).
Each question will be graded "pass" (full points) or "fail" (no points). We award 0.5 bonus points for the exam for each theory and practical question solved. You must complete 50% of all exercises to enter the final exam.

1. **EXAMple Question** Recall the simple coin flip experiment where a coin is thrown N times. Given that the experiment results in H heads and $T = N - H$ tails (summarized as data D), our goal is to obtain the posterior over the underlying parameter q, $p(q|D)$. The data likelihood for this experiment is given by

$$p(D|q) = \binom{N}{H} q^H (1-q)^T, \tag{1}$$

and there exists a conjugate Beta prior (B is the Beta function),

$$p(q) = Beta(q; \alpha, \beta) = \frac{q^{\alpha-1}(1-q)^{\beta-1}}{B(\alpha, \beta)} \tag{2}$$

so that the analytical posterior is given by

$$p(q|D) = Beta(q; \alpha + H, \beta + T) = \frac{q^{\alpha+H-1}(1-q)^{\beta+T-1}}{B(\alpha + H, \beta + T)} = \frac{q^{\alpha+H-1}(1-q)^{\beta+T-1}}{\int x^{\alpha+H-1}(1-x)^{\beta+T-1}dx}. \tag{3}$$

Calculate the Laplace approximation for $p(q|D)$.

2. **Theory Question** For the rest of this week's exercise and the programming exercise we take an excursion to computational neuroscience. This exercise might appear longer than usual, but it mostly contains introductory text about the model—the actual amount of calculations you have to do should be similar to previous weeks. You will use a Poisson Generalized Linear Model (GLM) to analyze neuronal activity. First you consider a basic version of the model and derive an analytical maximum likelihood estimate (a-b), to then extend the model to a GLM and perform inference with a Laplace approximation to the otherwise intractable posterior (c-d).

Neurons process information by emitting (firing) so-called action potentials (AP). APs are pronounced all-or-none events that are generated in response to a certain input and that are often modelled as single binary events, also called spikes. The activity of a neuron over a certain time period can then be described by a spike-train, a long vector of zeros and ones, where each entry corresponds to a time bin of size Δt and whether the neuron produced a spike (1) or not (0).

Using these assumptions one can model the activity of a neuron using a Poisson process that generates a number of spikes y in time bins of size Δt (spike counts) with a certain rate r:

$$p(y|r) = \frac{(r\Delta t)^y \exp(-r\Delta t)}{y!}, \tag{4}$$

where the rate can be interpreted as the firing rate of the neuron.

When analyzing neuronal data, one usually studies the activity of a neuron over multiple trials (independent repetitions of an experiment) i, $i = 1, \ldots, L$, and over a range of timebins Δt, $t = 1, \ldots, T$. One further assumes that the spike counts in each timebin are independent.

(a) Using this setup, write down the likelihood of observing spike counts $y_{1:T,1:L}$ over T time bins and L trials given a rate Poisson rate r, $p(y_{1:T,1:L}|r)$ and estimate the firing rate underlying the observed by obtaining the likelihood estimate (MLE) of r.

(b) The MLE for the firing rate makes sense intuitively, why?

To actually process information neurons need to generate spikes in response to some input. We now extend the model to account for this, by adding a dependence of the firing rate r_t at time t on a stimulus x_t, for each timebin t. Even more, we assume that the neuron's activity not only depends on the stimulus value in the current timebin, but on the stimulus history, i.e., on all stimulus values up to K timebins in the past, $x_{(t-K):t}$. We further assume that the neuron puts a weight on each stimulus value, depending on how far in the past it occurred, i.e., we assume a weight vector $\boldsymbol{\omega}$ applied to the stimulus vector $x_{(t-K):t}$. Finally, we add an offset variable b, and an exponential non-linearity around the firing rate estimate, to ensure it is always positive.

Overall, this results in the following model of the firing rate:

$$z_t = \sum_{k=1}^{K} \omega_k x_{t-k} + b \tag{5}$$

$$r_t = \exp(z_t) \tag{6}$$

$$y_t = \text{Poisson}(r_t). \tag{7}$$

You might recognize the canonical form of a GLM, i.e., we have constructed a Poisson GLM to model the firing rate of a neuron. The fact that we formulated the model as a GLM will come in handy below.

With this model we can study the relationship between a given stimulus and the response of the neuron. For example, if we were to estimate the weights $\boldsymbol{\omega}$ with which the neuron weighs the stimulus history, this would inform us about the stimulus preference of this neuron (its receptive field). To estimate $\boldsymbol{\omega}$, we might want to obtain a maximimum likelihood estimate (MLE). However, in contrast to exercise (a) there is no closed form MLE for this formulation of the model [1]. And here the GLM property comes in—because the likelihood is in the exponential family, and the exponential non-linearity is monotonically increasing, the likelihood is guaranteed to be concave and has a single maximum. Thus, we can use standard numerical optimization tools to obtain it (you will do so in the programming exericise).

As a next step, we might want to incorporate prior knowledge about the filter weights $\boldsymbol{\omega}$, i.e., introduce a prior over $\boldsymbol{\omega}$ and obtain the maximum a posteriori (MAP) estimate instead of the MLE. Again, the GLM comes in handy, because if we choose a concave prior then the posterior is concave too and we can again use numerical optimization to get the MAP (see programming exercise). We change to vector notation for $\boldsymbol{\omega}$ and $\boldsymbol{x}$.

[1] The rate now depends on the time as well so that the underlying process is an inhomogeneous Poisson process for which there is no general closed form solution for the MLE

The Poisson likelihood is given by

$$p(y_{1:T}|\boldsymbol{\omega}, \mathbf{x}) = \prod_{t=1}^{T} \frac{(r_t \Delta t)^{y_t} \exp^{-r_t \Delta t}}{y_t!} \tag{8}$$

$$r_t = \exp(\boldsymbol{\omega} * \mathbf{x}_{t-K\,:\,t}), \tag{9}$$

We define a multivariate standard normal prior on $\boldsymbol{\omega}$:

$$p(\boldsymbol{\omega}) = \mathcal{N}\big(\mathbf{0}, \mathbf{I}\big), \tag{10}$$

The corresponding log-posterior is given by

$$\log p(\boldsymbol{\omega}|y_{1:T}, \mathbf{x}_{1:T}) = \log(p(y_{1:T}|\boldsymbol{\omega}, \mathbf{x}_{1:T}) + \log p(\boldsymbol{\omega}) + C \tag{11}$$

$$= \sum_{t=1}^{T} \Big(y_t \log r_t \Delta t - r_t \Delta t - \log(y_t!) \Big) + \log p(\boldsymbol{\omega}) + C \tag{12}$$

(c) Like the maximum likelihood estimate, the MAP does not have a closed form solution. However, we can make a Laplace approximation to it, how? Write down the steps needed to get the Laplace approximation.

(d) Calculate the Hessian of $\log p(\boldsymbol{\omega}|y_{1:T}, \mathbf{x}_{1:T})$ with respect to $\boldsymbol{\omega}$ and show that it is negative semidefinite (Hint: to take the gradient of the rate r_t it may be easier to write it in terms of the Hankel matrix - see programming exercise).

Use these results to complete the Laplace approximation in the programming exercise.

3. **Practical Question** See `Exercise_05.ipynb`.