

Theoretical and Computational Neuroscience 2018

Problem Set 4: The Linear-Nonlinear Model

Due date: Thursday, March 1, 2018

1 Introduction

The linear-nonlinear model of a neuron is a way to predict the neural response to a stimulus in terms of an estimated firing rate for every time point in the stimulus sequence. Not every neuron behaves according to this simple model, but it can be a starting point for a more complete characterization.

The model makes the assumption that the neuron scans the input stimulus for the presence of a specific feature. The more closely the stimulus resembles this trigger feature, the greater the probability that the neuron will fire an action potential. What this feature *is* can be extracted by computing the spike-triggered average (see below).

The simple assumption that twice as much trigger feature will cause twice as much response is often too simple, as neurons exhibit nonlinear behavior, e.g. saturation at high stimulus intensities. The same is true for low trigger feature intensities as these are usually below the threshold for firing. A good description is a sigmoidal curve connecting trigger feature intensity with spiking rate.

So what is the linear part and what is the nonlinearity? The linear part of the model is the assumption that the neuron runs a linear filter (the spike-triggered average) over the data to find the intensity of the trigger feature. The filter output at each point in time is then transformed into a spiking rate through a nonlinear, often sigmoidal, function. In addition, we assume that the nonlinearity is *static*: it will not change over time. One reason that this assumption can be wrong is that the neuron may adapt to stimulus contrast. This happens, for example, in the retina.

2 The Spike-Triggered Average

2.1 Introduction

For certain stimuli, an unbiased estimate of the linear filter is the *spike-triggered average*. In this method, the neuron is probed with a white noise stimulus sequence. From time to time, the neuron will fire an action potential because, by chance, the noise resembled part of the “trigger feature” for the cell. Because a long sequence of white noise probes all aspects of a possible trigger feature with equal probability, we will be able to reconstruct this optimal stimulus piece by piece. When we build the average of all white noise sequences that precede a spike, unimportant features in the noise average out and important features add up, giving in theory the stimulus to which the cell’s response is tuned.

2.2 Adaptation of Retinal Ganglion Cells to Different Contrasts Levels

Read pages 19 - 23 in Dayan and Abbott to understand the details of the spike-triggered average analysis using white noise stimuli for neurons of the early visual system. Download the file `contrast_response.mat` from Blackboard containing the stimulus and the response of a retinal ganglion cell to different contrasts of a random full-field flicker stimulus. (The data is from experiments by Bart Borghuis and Kristy Simmons, Balasubramanian lab.) The data contains nine trials each 100 sec long at 2 kHz sampling rate. Each trial has a different contrast (standard deviation divided by mean) of random full-field flicker. The contrast steps are given in the variable called `contrasts`. The variable `stimulus` contains the random stimulus sequence for *one* trial. The stimulus at each time is given as the deviation of the monitor intensity from its mean

value. All nine trials use this identical stimulus sequence, scaled by a factor `contrasts(i)`. The variable `spikeTimes{i}` contains all spike times (in samples) for the i^{th} trial relative to the beginning of that trial.

Problem 1: (a) Display a plot of the spikes in the first 20 seconds of the trial with the highest contrast.

(b) For the trial with the highest contrast, plot the time-dependent firing rate for the first 20 seconds. To do that you can use the output of MATLAB's `hist` function. Use 100ms time bins. Be sure to scale the output so that the firing rate is measured in spikes/s.

(c) Now estimate the firing rate by adding up Gaussian functions centered at each spike time, each having a width of about $\sigma = 35\text{ms}$. This will give a smoother firing rate curve, which will be needed later. The normalization is important here: ensure that the values in each Gaussian sum to 2000 (the sampling rate).

(d) Plot the overall average firing rate as a function of contrast.

(e) Generate and plot the spike-triggered averages for all nine different contrast levels. Normalize each of them so that the sum of the *squared* values is one. You will have to find out how far back in the past from the spikes you have to go to catch the complete spike-triggered average.

(f) Describe how the spike-triggered average changes with contrast. What does that tell us about the behavior of this retinal ganglion cell?

3 The Static Nonlinearity

Before you start this question, you should make sure to complete the previous one, and pay attention to parts (c) and (e) in which the normalization of the spike triggered average is important. Read pages 48 - 51 of the textbook to understand the concept of static nonlinearities.

Problem 2: (a) For every trial, use the corresponding STA from problem 1 as a linear filter (as described in the lecture) to extract the trigger feature intensity as a function of time. Be sure to scale the stimulus by the appropriate contrast value, in order to match the actual stimulus that was displayed during the experiment.

(b) Compare your smooth spike rate curves from problem 1c with the trigger feature intensities from problem 2a. Tune the width of your Gaussian kernel until you get the best agreement between the two curves (in terms of how smeared out the bumps in both graphs look). A good way to do that is to temporally set all values of the trigger feature curve that are smaller than zero equal to zero so you get only positive bumps as in the firing rate plot. Also have a look at figure 2.26. This step is essential to get good results! It should be helpful to start with the highest intensity trial. Plot your result showing both the trigger feature intensity graph and the smoothed spiking rates for some of the contrasts. Be sure not to introduce an artificial offset between the two curves.

Now we want to generate the static nonlinearities for all nine trials.

(c) Plot a scatter plot of the filter outputs (x-values) against the firing rates (y-values) by using the two curves you just constructed. Using every 10^{th} time point in both plots should give you enough data points. This gives you something like figure 2.2 in Dayan and Abbott (for now without the fitted curve). Do this for all nine trials.

(d) Average the points in your scatter plot until it is possible to draw a nice graph of the nonlinearity for all nine trials. (You should aim for 20-40 points) What do you observe when you compare the nonlinearities for the different intensities?

(e) Apply your estimated nonlinearity to the filter output to generate the model's predicted firing rate. This can be done with the `interp1` command. `interp1(x,y,x0)` interpolates between a list of (x, y) points to give the function values at the points x0. (x,y) will be the points on the nonlinearity that you found in problem 2d, and x0 will be the full filter output. Plot your result together with the actual firing rate. Is the linear-nonlinear model a fair description of the response of the neuron? Check different contrast intensities.

4 Simulating V1 Simple Cells

The primary visual cortex, also called V1, is an early place (and functional unit) in the brain where the visual image coming from the retina is analyzed. One purpose of V1 is to build a "sparse" representation of the visual scene in terms of Gabor functions. Gabor functions will represent the image as a collection of moving edges which then are further analyzed in associated visual cortex areas (edge detection). You can find a typical sparse representation in terms of Gabor functions in figure 10.7 of our book. Building a sparse (which means efficient and compressed) representation of an image using Gabor functions is also used in image compression. The so-called simple cells in V1 are the key elements in building that representation. There are also cells in V1 doing "higher order" processing called complex cells. They are the first step in getting a translationally invariant representation of objects. You want to identify an object independently of its exact position in space! It is strongly recommended to read the introduction to V1 cells in Dayan and Abbott pages 58 - 76 for a better understanding of this homework set.

The simple cells in V1 build up their receptive field from a clever combination of inputs from retinal ganglion cells. How this is done is still not understood completely. Instead, we can measure the receptive field of simple cells directly via STA analysis without caring about how the cell is actually getting its receptive field. A good approximation of a receptive field of a V1 simple cell is the so-called Gabor function. A Gabor function is a cosine wave with a Gaussian envelope: a "wave packet." The functional description of a Gabor function is:

$$D_s(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \cos(kx - \phi) \quad (2.27)$$

where σ_x and σ_y determine the extent of the receptive field in the x and y directions. ϕ is the preferred spatial phase and k is the preferred spatial frequency.

At the same time the cell responds to temporal light variation. This can be described with the following functional form:

$$D_t(\tau) = \alpha \exp(-\alpha\tau) \left(\frac{(\alpha\tau)^5}{5!} - \frac{(\alpha\tau)^7}{7!} \right) \quad (2.29)$$

where α sets the timescale. At time $\tau = 0$ the neuron is most likely to fire an action potential. Bigger τ means going backwards in time. It is important to note that the function $D_t(\tau)$ is only different from zero for $\tau > 0$. This will become important in part 2. A full description of the spatiotemporal receptive field (which we could get with an STA analysis) can be given as a product of the time-dependent part $D_t(\tau)$ and the space-dependent part $D_s(x, y)$:

$$D(x, y, \tau) = D_s(x, y)D_t(\tau) \quad (2.a)$$

Because V1 simple cells are a kind of spatial frequency detector (because the Gabor function looks like a localized cosine wave), the cell can be probed with a time varying counterphase sinusoidal grating. The functional form of that grating is:

$$s(x, y, t) = A \cos(Kx \cos \Theta + Ky \sin \Theta - \Phi) \cos(\omega t) \quad (2.18)$$

K is the spatial frequency, ω describes the temporal frequency, Θ is the orientation of the grating, Φ is the spatial phase and A is the overall amplitude. We will set $A = 1$. The parameters of the counterphase sinusoidal grating are visualized in figure 2.8. The counterphase sinusoidal grating is not moving towards a direction. Instead it is oscillating with frequency $\omega/2\pi$. By changing the parameters of the grating we can understand what angles or spatial frequencies a specific V1 cell is tuned to. By doing that, one finds that there are all kinds of simple cells in V1, tuned to all possible angles, spatial frequencies, and temporal frequencies.

Because of the special structure 2.a of the receptive field, the spatial and the temporal parts of the receptive field response separate:

$$L(t) = L_s L_t(t) \quad (2.b)$$

The spatial linear response L_s is just the grating filtered with the spatial part of the receptive field $D_s(x, y)$:

$$L_s = \int dx dy D_s(x, y) A \cos(Kx \cos(\Theta) + Ky \sin(\Theta) - \Phi) \quad (2.31)$$

The temporal linear response function $L_t(t)$ is the temporal oscillation filtered with the temporal part of the receptive field $D_t(\tau)$:

$$L_t(t) = \int_0^\infty d\tau D_t(\tau) \cos(\omega(t - \tau)) \quad (2.32)$$

Problem 3:

First we want to construct the receptive field of a V1 cell according to equations 2.27, 2.29 and 2.a.

(a) Construct the spatial part of a vertically oriented V1 simple cell using function 2.27. Use the values from figure 2.15, as we will need them later. In particular, pick the special case $k = 1$ and $\sigma_x = \sigma_y = \sigma = 2$. Take the phase $\phi = 0$ as given in figure 2.15. Make plots of your spatial receptive field equivalent to figure 2.10 A. Instead of doing a contour plot, you can also make a color image (MATLAB function `image`) where different colors (or brightnesses) correspond to the bright or dark sensitive areas of the receptive field. The `image` function of MATLAB operates in a certain number range, so you probably have to scale your values to make the image visible.

(b) Construct the temporal part of the corresponding receptive field as given in formula 2.29. Use $\alpha = 1/(15ms)$. Convince yourself that you get a graph qualitatively equivalent to figure 2.14.

(c) Combine the spatial and the temporal parts of your receptive field (according to 2.a) to get a spatiotemporal field. Make a series of plots similar to figure 2.13 to see the temporal dynamics of your receptive field.

Now we want to construct the stimulus (a counterphase sinusoidal grating) according to equation 2.18.

(d) Construct the counterphase sinusoidal grating according to function 2.18. Have a look at your result by producing some pictures (using the `image` function of MATLAB), varying Θ or K . Keep $t = 0$ for the moment. You should get a picture similar to figure 2.8 A. Now convince yourself that the grating is oscillating by making a small "movie" (you can do this in MATLAB by having a loop where you change t and update your image in every iteration, use the `pause(sec)` function to slow the movie down.).

Now we want to test the linear response of our model V1 neuron to a counterphase sinusoidal grating stimulus.

(e) Test the selectivity (linear response functions) of the spatial part of the V1 receptive field (we can do that separately because the spatial and temporal part separate as we saw in equation 2.a. To test the selectivity of the receptive field we have to use it as a filter over the stimulus. This is exactly what equation 2.31 is saying. Reproduce the figures 2.15 A, B and C. Remember that we put $k = 1$ and $\sigma = 2$. To produce the graphs you have to numerically integrate equation 2.31. To get the orientation response (figure A) you have to set $K = k$ and $\Phi = 0$ and vary the parameter Θ as described in the figure. To get the spatial frequency response (figure B) you have to set $\Phi = 0$ and $\Theta = 0$ and vary K . To get the spacial phase response you have to set $K = k$ and $\Theta = 0$ and vary Φ . Be sure to use appropriate integration limits! You do not want to cut essential parts of the receptive field, nor do you want to integrate up to infinity!

(f) Test the frequency part of the receptive field selectivity. Because the temporal and the spatial part of the receptive field separate, we can use only the temporal part 2.29 as a filter. Equation 2.32 shows us how the temporal part of the stimulus is filtered by the temporal part of the receptive field. Take into account that our unit for $\omega/2\pi$ is kHz, because we measure times in *ms*. The range of frequencies depicted in figure 2.16 corresponds in our case to values $\omega = 0 \dots 2\pi \cdot 0.02$. The output $L_t(t)$ will now be oscillating in time. To get a plot like figure 2.16, you have to extract the amplitude of these oscillations. Be sure to use a long enough stimulus sequence (especially for low ω !) to get at least one full oscillation to measure its amplitude. Use $\alpha = 1/(15ms)$. Be sure to use appropriate integration boundaries as in the case of the spatial part!

5 Group Project

Please build a small group of 3-5 people with other students and select a group project of your choice from the additional handout (also available on Canvas) or according to your own interests. You can work on these group projects until the end of the semester. Groups smaller than 3 and larger than 5 are strongly discouraged.

Problem 4:

Write one paragraph of text introducing the topic of your choice and hand it in together with the other homework problems.

Turning in the homework: Please upload the following files in response to all Part 2 (numerical simulation) questions to the Canvas Assignment Homework 4:

1) Your legible and commented code. If you have multiple files (for example, for different problems), please indicate which can be run to reproduce the output for each problem in file name. Please be sure that your code runs! If it does not, it partial credit may not be awarded for incorrect outputs.

2) A word document containing your answers to each question, with your MATLAB output figures embedded where necessary. Please include a caption accompanying each figure describing the relevant content. Be sure to label your axes, and make sure all relevant dynamics are clearly visible in the figures you make. Legibility is key!