Exam: Introduction to computational neuroscience of vision

Matthew Chalk, Ulisse Ferrari, Olivier Marre

Introduction

You are given data consisting of recorded neural responses to a flashing black and white checkerboard stimulus. In the following exercises will use computational models to understand how this neuron responds to visual stimuli, and investigate its information processing capacities.

You have been given a Matlab file called 'exam.m'. Certain lines have been commented out. In each stage of this exam you will have to alter the code as instructed, to complete the assignments. You will be asked to answer questions about your results.

Description of data

The dataset consists of the visual stimulus (X) and recorded spike counts (r).

- The stimulus (called X in the code), is an $n_x \times n_x \times N$ array, consisting of N black and white images. The first figure shows a presented stimulus on one trial.
- The response (called r in the code), is an $1 \times N$ vector, consisting of recorded spike counts to each image. The second figure shows a histogram of recorded spike counts.

1 Reverse correlation (10 points)

We first assume the following linear model of neural responses:

$$r \approx \boldsymbol{w} \cdot (\boldsymbol{x} - \langle \boldsymbol{x} \rangle) + \langle r \rangle \tag{1}$$

where $\boldsymbol{w}$ is a vector of linear weights, $\boldsymbol{x}$ is a stimulus vector and r is the recorded spike count. $\langle r \rangle$ and $\langle \boldsymbol{x} \rangle$ are the trial averaged mean spike count and stimulus vector, respectively. We can simplify this expression by defining $\tilde{r} = r - \langle r \rangle$, and $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \langle \boldsymbol{x} \rangle$, giving:

$$\tilde{r} \approx \boldsymbol{w} \cdot \tilde{\boldsymbol{x}} \tag{2}$$

We estimate $\boldsymbol{w}$ by minimising the mean squared error between observed and recorded responses: $\left\langle (\tilde{r} - \boldsymbol{w} \cdot \tilde{\boldsymbol{x}})^2 \right\rangle$. In our case, as different pixels of the stimulus are independent and have equal variance, $\sigma^2 = \left\langle x_i^2 \right\rangle$, we arrive at the solution:

$$\boldsymbol{w} = \frac{1}{\sigma^2} \langle \tilde{r} \tilde{\boldsymbol{x}} \rangle \tag{3}$$

- **Exercise 1a**: Complete the code to compute $\tilde{r}$ and $\tilde{\boldsymbol{x}}$. **(1 point)**
- **Exercise 1b**: Check in the code that the stimulus is 'white' (i.e. different pixels have zero covariance) **(1 point)**

- **Exercise 1c**: Complete the code to estimate $\boldsymbol{w}$. Plot a 2-d visualisation of weights, $\boldsymbol{w}$ **(2 points)**

  What does this tell us about how the neuron responds to visual stimuli? **(2 points)**

  Which neural area might this neuron be from (justify your response)? **(2 points)**

- **Exercise 1d**: Complete the code to plot $\boldsymbol{w} \cdot \tilde{\boldsymbol{x}}$ versus the observed spike count.

  Comment and relate to our modelling assumptions (see Equation 2) **(2 points)**

2 Linear-nonlinear-Poisson (LNP) model (10 points)

We next consider a 'linear-nonlinear-Poisson' (LNP) model of the neuron's responses. Here, we assume that the neuron's mean firing rate is obtained as follows:

$$f\left(\tilde{\boldsymbol{x}}\right) = \exp\left(\boldsymbol{w} \cdot \tilde{\boldsymbol{x}} + b\right) \tag{4}$$

Spike counts are assumed to be sampled from the following Poisson distribution.

$$p\left(r|\tilde{\boldsymbol{x}}\right) = \frac{1}{r!} f\left(\tilde{\boldsymbol{x}}\right)^r e^{-f\left(\tilde{\boldsymbol{x}}\right)} \tag{5}$$

We learn the parameters, by maximising the average log-likelihood:

$$L = \left\langle \log p\left(r|\tilde{\boldsymbol{x}}\right)\right\rangle$$

We do this using gradient ascent, with updates:

$$\begin{aligned} \boldsymbol{w} &\leftarrow \boldsymbol{w} + \eta \nabla_{\boldsymbol{w}} L \\ \boldsymbol{b} &\leftarrow \boldsymbol{b} + \eta \nabla_b L \end{aligned}$$

where η is a constant learning rate.

- **Exercise 2a**: Complete the code to compute $f\left(\tilde{\boldsymbol{x}}\right)$. **(1 point)**
- **Exercise 2b**: Complete the code to perform updates of $\boldsymbol{w}$ and b. **(1 point)**
- **Exercise 2c**: Complete the code to obtain a visualisation of the learned filter, $\boldsymbol{w}$. **(1 point)**

  How does this compare to the filter obtained in the previous section? **(1 point)**

- **Exercise 2d**: Plot the predicted mean firing rate versus the observed spike count. **(1 point)**

  Comment and compare to what you obtained using reverse correlation. **(1 point)**

  What are the advantages of the LNP model versus reverse correlation? **(2 points)**

  What are the limitations of the LNP model for modelling neural responses? **(2 points)**

3 Measuring mutual information (15 points + 4 bonus points)

We can use mutual information to quantify how much the neuron encodes about the presented stimulus. The formula for mutual information is as follows:

$$I(R;X) = H(R) - H(R|X)$$

where $H(R)$ and $H(R|X)$ are the response and noise entropy, defined as:

$$\begin{aligned} H(R) &= -\sum_r p(r)\log p(r) \\ H(R|X) &= -\left\langle \sum_r p(r|x)\log p(r|x) \right\rangle \end{aligned}$$

where the average is taken over stimulus presentations.

We will use our LNP model to estimate how much information the neuron encodes. We estimate $p(r)$ by generating samples of r from the model, and then use a histogram to estimate the empirical distribution of r. We estimate $p(r|x)$ using Equation 5.

- **Exercise 3a**: Sample spike counts from model (use 'poissrnd'), and complete the code to estimate $p(r)$. **(2 points)**
- **Exercise 3b**: Complete the code to estimate $H(R)$ and $H(R|X)$, and compute the mutual information. **(3 points)**

  Do you obtain the same result for the mutual information each time you run the code? Comment & list some advantages/disadvantages of using mutual information. **(2 points)**

How does the mutual information depend on the response properties of the neuron? To investigate this, let us consider a model of neural responses with mean firing rate:

$$f(\tilde{\boldsymbol{x}}) = \frac{A}{1+\exp(-\boldsymbol{w}\cdot\tilde{\boldsymbol{x}} - b)}$$

We ask how does the encoded information varies as we alter the constant, b?

- **Exercise 3c**: Plot how the non-linearity varies with b, and describe what you see. **(2 points)**
- **Exercise 3d**: Complete the code to estimate $p(r)$ and $H(R)$ and $H(R|X)$ with each value of b. (you can reuse your answer from exercise 3a-b) **(2 points)**
- **Exercise 3e**: Plot how the mutual information varies with b. **(1 point)**

  Comment and explain what you see **(3 points)**
- **Bonus question**: Try seeing how the information depends on other aspects of the non-linearity. For example, what happens if you vary the magnitude of the filter, $\boldsymbol{w}$? Plot your results and comment on what you see **(4 bonus points)**