CNS Assignment: Propagation of Orientation Tuning in the Visual System

November 2026

Submission Details

The deadline for this assignment is 20th November at 12pm (standard late penalties apply). The assignment should be submitted via the course Learn page in pdf format, using the naming convention B0123456.pdf (i.e., your exam number).

You should include the code you used to produce any figures in the assignment as an appendix (the code will not be marked, but may be checked where results are wrong and/or for plagiarism/AI detection purposes, do not include any code in your main report).

Your submission should be no longer than **10 pages** (excluding references, front matter, and appended code). To ensure fair and consistent marking, any submission that extends beyond the 10 page limit will receive an automatic zero in the "Marks for Style of Presentation" rubric.

Plagiarism, AI Policy and Academic Misconduct

You may discuss this assignment with others, but direct copying of code, text, or mathematical results is not allowed. Any text that is directly copied from other sources should be placed in quotation marks with the source cited at the end of the sentence. Text which is written in your own words but based on material from other sources also requires a citation.

The AI policy for this assignment is AI-Assisted Planning:

"Academic integrity is an underlying principle of research and academic practice. All submitted work is expected to be your own. AI tools (e.g., ELM) should not be used to generate content for this assessment. However, you are allowed to use these tools to identify ideas, key themes, and plan your assessment. If you use AI software, you must acknowledge its use in your submission."

Please remember the good scholarly practice requirements of the University regarding work for credit. You can find guidance at the School page https://informatics.ed.ac.uk/taught-students/all-students/your-studies/academic-misconduct. This also has links to the relevant University pages.

Marks for Style of Presentation (10 marks)

You should present your findings in the style of a scientific report. By this we mean that the document should include text broken into appropriate sections (e.g., by question number and topic) with equations (where relevant), figures (with axes clearly labeled, font size similar to the font in

the main text, figure captions to explain their content, and numbering/lettering of figures and subpanels in order to refer to them from the main text), and references (if relevant).

You should consider how to best communicate your findings in a clear and concise way to the reader. This includes designing figures appropriately, for example by making use of subplots and plotting multiple things on the same graph to save space or for aid of visual comparison, setting the axis limits appropriately, using figure legends and colour schemes to delineate different things plotted on the same graph, etc. You are encouraged to convey your findings clearly and succinctly - longer submissions are not necessarily better.

Any concepts, terminology, etc. you introduce that were not covered in the lecture material require an explanation and a reference.

You should report details of your analyses as clearly as possible, including for example: the parameters used, the number of random seeds, etc.

To take these factors into account, 10 marks have been allocated for the quality of presentation with regards to: 1) the overall layout and style (3 marks), 2) the quality of figures (3 marks), and the scientific clarity and precision of written answers including appropriate referencing of external material where relevant (4 marks).

Introduction and Background

In this assignment, we consider information transmission in a simplified model of the early visual system. We consider the response of visual cortex (V1) to a stimulus orientation $\theta \in [-\pi, \pi)$, which we model using the feedforward cascade (Figure 1):

$$\theta \to \mathbf{z} \to \mathbf{x}. \tag{1}$$

We consider $\mathbf{x} \in \mathbb{R}^N$ to be the responses of a set of N orientation-tuned neurons in V1, which we model using a feedforward network with Poisson variability described below. $\mathbf{z} \in \mathbb{R}^2$ is an input layer containing a simplified encoding of the circular stimulus variable θ, and may be interpreted as representing the cumulative processing and noise in all layers before V1.

We model these two layers as follows:

$$\mathbf{z} = \boldsymbol{\mu}(\theta) + \mathbf{n}, \qquad \boldsymbol{\mu}(\theta) = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \qquad \mathbf{n} \sim \mathcal{N}(0, \sigma_z^2 I). \tag{2}$$

$$\mathbf{x} = \text{Poisson}(\boldsymbol{\lambda}), \qquad \boldsymbol{\lambda} = \max(W_{\text{in}}\mathbf{z} + b, 0). \tag{3}$$

This amounts to treating the input as a stimulus-tuned mean $\boldsymbol{\mu}(\theta)$ plus isotropic Gaussian noise $\mathbf{n}$ (with variance σ_z^2), which is then transformed through feedforward weights $W_{\text{in}} \in \mathbb{R}^{N\times 2}$, bias b (applied uniformly to all units) and an element-wise ReLU nonlinearity to obtain the firing rate $\boldsymbol{\lambda}$. Spike counts $\mathbf{x}$ are then drawn from a Poisson distribution.

Here, we are interested in how stimulus information and decoder performance propagate through this feedforward system, and how this depends on the network connectivity. This is useful, for example, in understanding how sensory information propagates between brain areas and how this relates to the connectivity observed in the brain. Throughout, we will assume random connectivity[1] matrices with elements drawn iid from a normal distribution

$$W_{\text{in},ij} \sim \mathcal{N}(0, \sigma_W^2). \tag{4}$$

[1] It is also interesting to consider how information may propagate through non-random connectivity.

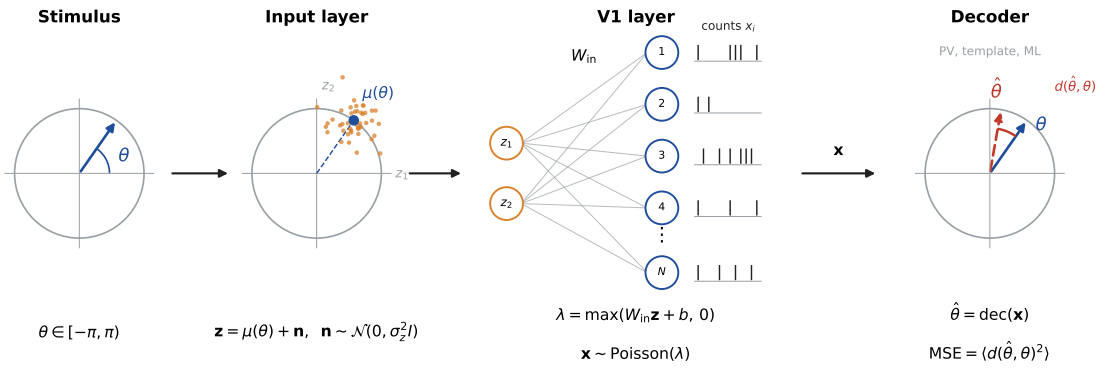

Figure 1: Model schematic. A circular stimulus θ is encoded as a noisy two-dimensional input $\mathbf{z} = \mu(\theta) + \mathbf{n}$, which drives N V1 units through random weights W_{in}, a bias b and a ReLU. Spike counts $\mathbf{x}$ are Poisson distributed. A decoder then estimates the stimulus $\hat{\theta}$ from the spike counts $\mathbf{x}$.

This leaves the key parameters N, σ_z, σ_W, b, the effect of which are explored in detail in the questions below. Note that the connectivity W_{in} is fixed for a given (randomly sampled) network, but the two noise processes $\mathbf{n}$ and $\mathbf{x}|\mathbf{n}$ vary over *trials* (i.e., repeats of a stimulus θ for a given network).

Question 1 (15 marks)

We first consider how the stimulus orientation θ is encoded in the visual layer $\mathbf{x}$.

1. Generate and plot examples of tuning curves $f_i(\theta)$ (i.e., the mean response of a neuron x_i vs orientation θ in a fixed, randomly sampled, network) with $N = 25$, $\sigma_W = 5$, $b = 15$ and $\sigma_z = 0$. Note that these tuning curves do not require numerical estimation via averaging of responses over trials in this setting, you should observe a simpler way. For a given network, you may overlay the tuning curves of all neurons in the same plot. (5 marks)

2. Investigate how these vary when you change b from 15 to 0, showing representative examples. (2 marks)

3. The set of all tuning curves can be stacked to form a $N_{\text{stim}} \times N$ matrix $F = [f_1(\theta), ..., f_N(\theta)]$ (where N_{stim} is the number of values of θ sampled). Plot the top three principal components[2] of this matrix as a 3d line or scatter plot (i.e. the set of N_{stim} points in the 3d space with axes PC1 vs PC2 vs PC3) for representative example networks drawn with $b = 15$ and $b = 0$. (4 marks)

4. Comment on what you observe: how is the stimulus orientation encoded in single neuron responses and in the overall population responses, and how does this depend on the bias term b? Why do you think this is happening? (4 marks)

[2]This can be done using the PCA function in scikit-learn in Python. If you choose to do this manually (e.g., using the singular value decomposition), make sure to centre the data first by mean-subtracting each neuron's average tuning curve.

Question 2 (30 marks)

We next consider the performance of stimulus decoders applied to each layer, and how this depends on connectivity. Consider the following decoders:

- A population vector decoder, with $\hat{\theta}_{PV} = \text{atan2}\left(\sum_i x_i \sin(\theta_i), \sum_i x_i \cos(\theta_i)\right)$ where $\theta_i = \text{argmax}_\theta(f_i(\theta))$ is the preferred stimulus of neuron i.
- A template matching decoder. Construct the template $F(\theta) = [f_1(\theta), ..., f_N(\theta)]$ where $f_i(\theta)$ is the tuning curve of the ith neuron. Decode the stimulus as $\hat{\theta}_{\text{TM}} = \text{argmin}_\theta \sum_i (F_i(\theta) - x_i)^2$.
- A maximum likelihood decoder based on an encoding model with Poisson variability in independent neurons, i.e. $\hat{\theta}_{ML} = \text{argmax}_\theta \left[\sum_i x_i \log f_i(\theta) - f_i(\theta) - \log x_i!\right]$. If $f_i(\theta) = 0$ and $x_i = 0$, evaluate as $x_i \log f_i(\theta) = 0$.

The following questions ask you to understand the properties of these decoders in the context of the present model, and to implement them and test their performance numerically.

1. Consider the Population Vector decoder applied to the responses of x_i. Explain why the properties of the present model (in particular, the tuning curves), could make this decoder biased or inaccurate without modifications. Explain why applying the PV decoder to the normalised responses $\tilde{x}_i = x_i / \max(f_i(\theta))$ resolves some of these problems. Throughout the rest of this assignment, assume the PV decoder is applied to the normalised responses $\tilde{x}_i$. (4 marks)

2. What benefits might the template matching decoder have over the Population vector decoder for the tuning curves considered in this assignment? Under what parameter settings of the model are the assumptions of the maximum likelihood decoder met? What properties are violated when these conditions are not met and why? (4 marks)

3. Define the mean squared decoding error as $\text{MSE} = \frac{1}{K} \sum_{k=1}^{K} d(\hat{\theta}_k, \theta_k)^2$ where $\theta_k, \hat{\theta}_k$ are the true and estimated stimuli (with $k = 1 : K$ samples[3]) and $d(\hat{\theta}_k, \theta_k) = \min(|\theta - \hat{\theta}|, 2\pi - |\theta - \hat{\theta}|)$ is a circular distance. Investigate how the decoding error of each above decoder varies with N, σ_W and b for $\sigma_z = 0$ (you can set $N = 25$, $b = 5$, $\sigma_W = 5$ as baseline values and independently vary each parameter while keeping the others fixed). Based on your findings, how would you choose these parameters to optimise decoding with each decoder? (12 marks)

4. How does decoding error scale with N for non-zero σ_z? You may fix $b = 5, \sigma_W = 5$ and choose several settings of σ_z. Give a principled explanation for the different scaling of decoder error for $\sigma_z > 0$ vs $\sigma_z = 0$. Note you will need to either numerically estimate the tuning curves or perform a more challenging analytical derivation to find them in this case. (10 marks)

[3]Note that you will need to average over networks (by sampling different settings of W_{in}), trials (by sampling different values of $\mathbf{n}$ and $\mathbf{x}$) and stimuli θ. Check that you have chosen a large enough number of each to get reliable results.

Question 3 (25 marks)

In the above questions you investigated the behaviour of the model numerically. Indeed, analytical expressions for the decoder performance in this model may be hard to obtain (or even impossible). However, the model becomes analytically tractable if we make two modifications:

- First, we replace the Poisson noise with Gaussian noise $p(\mathbf{x}|\mathbf{z}) = N(\boldsymbol{\lambda}, \sigma_x^2 I)$.
- Second, we assume that all neurons remain above threshold[4], so that $\lambda = W_{\text{in}}\mathbf{z} + b$.

We now investigate this simplified model analytically to obtain insight into the coding properties of this system:

1. Compute in closed form[5] the response distribution $p(\mathbf{x}|\theta)$. Using this, write down an expression[6] for the linear Fisher information available about θ in (i) the response $\mathbf{x}$ and (ii) the input layer $\mathbf{z}$. (7 marks)
2. Compare the expressions derived for the Fisher information in $\mathbf{z}$ and $\mathbf{x}$. Which is generally larger, and why? Can you see any settings in which the two expressions become equal and how do you interpret these cases? (5 marks)
3. Now focus on the case of random connectivity as above. Plot the expression for linear fisher information in $\mathbf{x}$ vs N for both $\sigma_z = 0$ and $\sigma_z = 0.1$ (fixing $\sigma_W = 5, b = 5$, $\sigma_x = 2$ and averaging over randomly sampled networks), and compare against the corresponding expression for $\mathbf{z}$. How do you interpret this scaling of Fisher information with N in each case? How does the Fisher information in $\mathbf{x}$ depend on σ_W and b, and what is the intuition for this? What are the parameter settings (or limiting cases) under which the Fisher information in $\mathbf{z}$ and $\mathbf{x}$ agree and how do you interpret these cases? (8 marks)
4. Why might the Fisher information in the full model with thresholding and Poisson variability behave differently to this simplified linear-Gaussian model? How does the Fisher information in the simplified model align with the decoder performances observed in the full model in Q2, and how do you explain any discrepancies? (5 marks)

Question 4 (20 marks)

Consider the following extension to the linear-Gaussian model.

- Model neurons as having tuning curves $\mathbf{f}(\theta) = \langle \max(W_{\text{in}}\mathbf{z} + b, 0) \rangle$ (where angled brackets denote average over $\mathbf{z}$ for fixed θ).
- Model the response as $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{f}(\theta) + GW_{\text{in}}\mathbf{n}, \Sigma_{\mathbf{x}})$, with covariance $\Sigma_{\mathbf{x}} = \text{diag}(\mathbf{f}(\theta))$ (i.e., $(\Sigma_{\mathbf{x}})_{ij} = 0$ for $i \neq j$ and $(\Sigma_{\mathbf{x}})_{ii} = f_i(\theta)$) and $G(\theta) = \text{diag}(g_1(\theta), ..., g_N(\theta))$ where $g_i(\theta)$ is the probability that neuron i is above threshold, i.e. that $(W_{\text{in}}\mathbf{z} + b)_i > 0$ for stimulus θ.

[4] In simulations, it may occur that some units go below threshold. We will not attempt to correct this, we simply replace the ReLU with an identity map throughout.

[5] It is not necessary to solve Gaussian integrals to do this. It is straightforward to find using the fact that Gaussians are closed under linear transformations.

[6] Your expression can be a function of W (i.e., valid for any choice of W). It is not necessary to express it in terms of σ_W or solve for random weight matrices.

1. (a) Explain why this model is a closer approximation to the full ReLU-Poisson model than the model of Question 3. Which aspects of the full model does it still miss? (4 marks)

   (b) Show that $\mathbf{x} \,|\, \theta$ is Gaussian, give its mean and covariance, and write down the linear Fisher information about θ in $\mathbf{x}$. [Hint: the most challenging step is to evaluate the tuning curve derivatives, which can be solved as $\mathbf{f}'(\theta) = GW_{\text{in}}\boldsymbol{\mu}'(\theta)$] (6 marks)

2. Plot the linear Fisher information[7] against N, b and σ_W for $\sigma_z = 0$ and $\sigma_z = 0.1$ (baseline values $N = 25$, $b = 5$, $\sigma_W = 5$, varying each parameter independently as in Question 2.3), and compare with the Fisher information in $\mathbf{z}$. Compare your results with those of Questions 2 and 3: does this model offer any new insight into the reason decoder performance scales as it does with N, b or σ_W in the full ReLU-Poisson model that was not apparent in the linear-Gaussian approximation of Question 3? (10 marks)

[7] Note that, for $\sigma_z = 0$, you may encounter cases in which neurons have both mean $f_i(\theta) = 0$ and variance $(\Sigma_{\mathbf{x}})_{ii} = 0$, which causes a 0/0 error. If you encounter such cases, you can discard these neurons before computing linear Fisher information.