Neural Encoding: Simple Model

Notes by Callyn V

1 Optimal Adaptation Strategies for Seeing in a Dynamic World

1.1 Constructing Response Models

We consider the probability of a response given a stimulus:

$$P(\text{response}|\text{stimulus}) \to r(t) \text{ given a stimulus } s(t). \tag{1}$$

This equation represents the probability distribution of a neural response given an input stimulus, describing how neurons encode information.

1.2 Basic Coding Model: Linear Response

$$r(t) = fs(t) \tag{2}$$

This is a simple linear relationship where the response is directly proportional to the stimulus, with f acting as a scaling factor.

Example Calculation: If $s(t) = 2$ and $f = 3$, then:

$$r(t) = 3 \times 2 = 6. \tag{3}$$

1.3 Basic Coding Model: Temporal Filtering

$$r(t) = \sum_{k=0}^{n} s(t-k) f_k \tag{4}$$

This discrete summation represents how past stimulus values contribute to the current response through a set of filter weights f_k.

Example Calculation: If $s(t) = \{1, 2, 3\}$ and $f_k = \{0.2, 0.5, 0.3\}$, then:

$$r(t) = (1 \times 0.2) + (2 \times 0.5) + (3 \times 0.3) = 0.2 + 1.0 + 0.9 = 2.1. \tag{5}$$

1.4 Basic Coding Model: Spatial Filtering

$$r(x, y) = \sum_{x'=-n}^{n} \sum_{y'=-n}^{n} s(x - x', y - y') f(x', y') \tag{6}$$

Example Calculation: Consider a 3×3 spatial filter with:

$$s(x,y) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad f(x',y') = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{7}$$

Applying convolution at the center (x,y):

$$r(x,y) = (1\times0)+(2\times1)+(3\times0)+(4\times1)+(5\times-4)+(6\times1)+(7\times0)+(8\times1)+(9\times0) = 2+4-20+6+8 = 0. \tag{8}$$

1.5 Basic Coding Model: Spatiotemporal Filtering

$$r(x,y,t) = \int\int\int f(x',y',t')s(x-x',y-y',t-t')dx'dy'dt' \tag{9}$$

This extends the previous models to include both spatial and temporal filtering, combining information over space and time..this one is tricky (still learning) lol :p

1.6 Linear/Nonlinear Models

A basic linear-nonlinear model:

$$r(t) = g\left(\int s(t-\tau)f(\tau)d\tau\right) \tag{10}$$

This model applies a nonlinear transformation $g(\cdot)$ to the linear filter output to better match neural response characteristics.

Example Calculation: If $g(x) = x^2$ and the filtered signal is 3, then:

$$r(t) = g(3) = 3^2 = 9. \tag{11}$$

1.7 Principal Component Analysis for Feature Selection

PCA can be used for spike sorting and feature extraction:

- Eigenfaces: ATT Labs, Cambridge (useful for face recognition)
- Spike Sorting: Koepsell et al., 2009 (used in neurophysiology to distinguish neurons)
- Retina Feature Selection: Fairhall et al., 2007 (applied in visual processing)