

# Computational Neuroscience

## Contents

<b>1</b>	<b>Course Introduction and Basic Neurobiology</b>	<b>2</b>
1.1	Descriptive, Mechanistic and Interpretive Models . . . . .	2
1.2	Neurobiology Fundamentals . . . . .	2
1.2.1	Action Potential . . . . .	3
1.2.2	Synapses . . . . .	3
1.2.3	Brain Areas, Networking and Functions . . . . .	4
<b>2</b>	<b>Neural Encoding Models</b>	<b>5</b>
2.1	The Neural Code . . . . .	5
2.2	Basic Encoding Models . . . . .	5
2.2.1	Feature Selection and Output Generation . . . . .	6
2.2.2	Introducing Variability . . . . .	7
<b>3</b>	<b>Neural Decoding Models</b>	<b>8</b>
3.1	Single Neuron Decoding and Signal Detection Theory . . . . .	8
3.2	Population Coding and Maximum Likelihood Estimation . . . . .	9
3.2.1	Maximum Likelihood Estimation (MAP) . . . . .	9
3.2.2	Maximum A Posteriori Estimation (MAP) . . . . .	10
3.2.3	Bayesian Estimation . . . . .	10
<b>4</b>	<b>Information Theory</b>	<b>11</b>
4.1	Entropy and Information . . . . .	11
4.1.1	Fundamentals . . . . .	11
4.1.2	Mutual Information and Kullback-Leiber Divergence . . . . .	11
4.2	Computing Information for Spike Trains . . . . .	12
4.3	Coding Principles . . . . .	12
<b>5</b>	<b>Summary I</b>	<b>14</b>
<b>6</b>	<b>Computing in Carbon</b>	<b>14</b>

## Week 1 Course Introduction and Basic Neurobiology

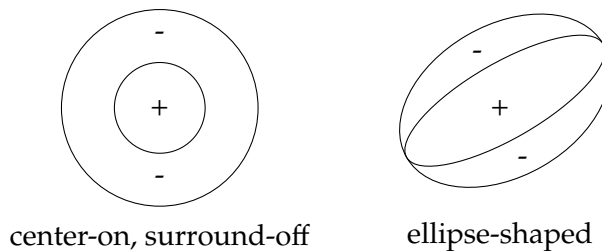
### § 1.1 Descriptive, Mechanistic and Interpretive Models

1. Descriptive models (**What**): describe ways that a neuron respond to external stimuli; encoding (describe responses), decoding (extract information)
2. Mechanistic models (**How**): simulating single/a network of neurons
3. Interpretive (Normative) models (**Why**): explaining the behaviour of neurons and neural networks

**Receptive fields:** Specific region or sensory space in which a stimulus can incur an electrical response in a sensory neuron.

#### 1. Descriptive

- How visual information is received: visual information → retina (ganglion cells) → lateral geniculate nucleus (LGN) → primary visual cortex (v1)



- The receptive fields are modeled by **reverse correlation**.
2. Mechanistic: How are receptive fields constructed? (from LGN to primary visual cortex)
    - Hubel & Wiesel: v1 RFs are constructed from converging LGN inputs
    - **Controversial**: The hypothesis does not take into account other recurrent inputs
  3. Descriptive: Why do we need receptive fields?
    - **Efficient Coding Hypothesis**: the brain's goal is to represent images as faithfully and efficiently as possible using receptive fields. Let  $RF_i$  denote the  $i$ th receptive field:

$$\hat{I} = \sum_{i=1}^N RF_i \omega_i$$
$$\min \frac{1}{N} \sum_{i=1}^N (\hat{I} - I)^2$$

Our brain adjusts the weights regarding each receptive field  $\omega_i$  to minimize mean square error.

- Efficient coding algorithms: sparse coding, ICA, predictive coding; starting with random  $RF_i$  and run until convergence.

### § 1.2 Neurobiology Fundamentals

#### The Neuron Doctrine

- Neuron is the fundamental structural and functional unit of the brain.
- Neurons are discrete cells and not continuous with other cells.
- Information flows from the dendrites to the axon via the cell body.

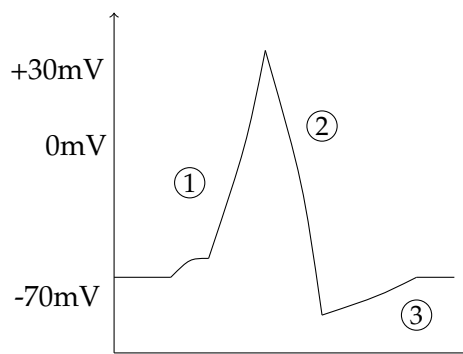
### §1.2.1 Action Potential

EPSP refers to excitatory postsynaptic potential. When  $\sum \text{EPSP} \geq \text{threshold}$ , action potential, or output spike, is generated.

1. Identities of cell membrane

2. Neuron Signaling

- spikes from presynaptic neurons → chemically gated channels open (at synapses) → changes in local membrane potential
- → voltage-gated channels → reaches action potential  
depolarization (voltage increases), hyperpolarization (voltage decreases)



**Stage 1** Chemically-gated channels activated; voltage-gated channels activated, sodium channels open with  $\text{Na}^+$  ions rushing into the neuron.

**Stage 2** Sodium channels inactivated and potassium channels open ( $\text{K}^+$  ions rushing out)

**Stage 3** Potassium channels close, sodium-potassium pumps open using ATP

3. Myelination of axons: action potential hops from non-myelinated regions

Active wiring: enable lossless signal propagation (saltatory conduction)

### §1.2.2 Synapses

There are two types of junctions between neurons:

- *Electrical synapses*: Gap junctions enable direct connection between neurons which lead to fast and rapid **synchronization**.
- *Chemical synapses*: Neurotransmitter molecules are received by receptors, or the chemically-gated channels. This junction enables optionally connected ionic channels and more **modifiable communication**.

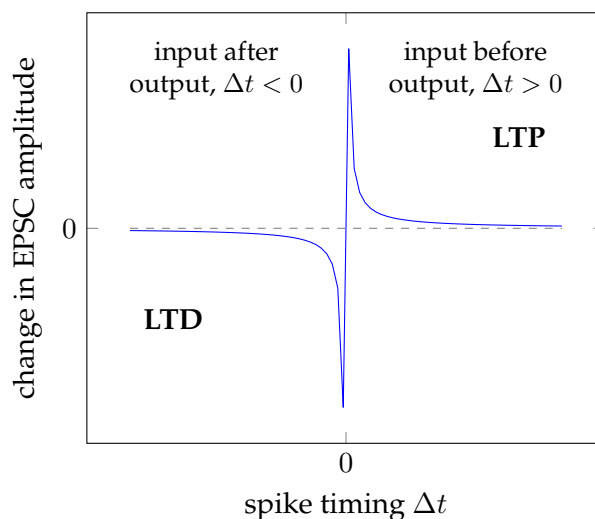
The table below shows what occurs to postsynaptic membranes in excitatory or inhibitory synapses.

Membrane/ Synapse Type	Presynaptic Membrane	Postsynaptic Membrane
Excitatory Synapse (1)	→	potential +
Inhibitory Synapse		potential -

In (1), excitatory postsynaptic potential is generated (what we known as EPSP).

**The Synapses Doctrine**: basis for memory and learning

- **Hebbian plasticity:** If neuron *A* repeatedly fires neuron *B*, the synapse from *A* to *B* is strengthened, which indicates a faster response.
- *Long Term Potentiation (LTP)* and *Long term Depression (LTD)* refers to the experimentally observed change in EPSP size or the same input over time. LTP and LTD strongly depends on spike sequence and timing: “Those who fire together, wire together”



### §1.2.3 Brain Areas, Networking and Functions

The nervous system incorporates *peripheral system* and *central nervous system*.

- Peripheral nervous system (PNS)
  1. **Somatic:** nerves connecting to voluntary skeletal muscles and sensory receptors. Afferent nerve fibres transmit signals from periphery to CNS; Efferent nerve fibres work the opposite way.
  2. **Autonomic:** nerves that connect to the heart, blood vessels, smooth muscles and glands.
- Central nervous system (CNS) = spinal cord + brain

**Comparisons between biological brain and digital computer** Due to the parallel computation and adaptive connectivity of biological brains, they are better at ill-posed problems such as speech and vision. For digital computers, they excel in math and symbol processing with sequential information processing with fixed connectivity of CPUs.

## Week 2 Neural Encoding Models

### § 2.1 The Neural Code

- **Neural Coding** performs the characterization of the hypothetical relationship between stimulus and neuronal responses, and the relationship among the electrical activities of neurons in the ensemble. The subject of investigation can be described as  $\mathbb{P}(\text{response} | \text{stimulus})$ , or, how does a stimulus cause a pattern of responses?
- **Raster Plot** is a tool for recording neuronal responses (spikes) with a given stimulus. The figure below shows a sample raster plot with the behavior of different neurons.



- When we observe action potentials, or, spikes in a particular situation, the neurons are responding to external stimulus by **encoding features**.
- **Tuning curves:** The y-axis of these curves are usually average firing rate, and the x-axis represents the stimulus parameter. It can be the orientation angle of the stimulus, the intensity of lighting on our subject of investigation, movement directions or anything.
- The neural codes demonstrate increasing levels of complexity when the stimulus parameter becomes more closely aligned with realistic scenarios. For example, fundamental geometric shapes can be perceived by simple receptive fields, while portrait and sceneries might involve more complicated stimulus representations. From morphological features to semantic implications, firing behavior becomes increasingly abstract along with the rise in complexity of the receptive fields.
- Neural Code Conclusion
  1. Neural coding translates the neuronal responses into specific forms for further investigation.
  2. Creating raster plots is an easy way to record and represent action potentials.
  3. Neural codes become increasingly abstract when the complexity of receptive field rises. For instance, semantic expressions usually have a more intricate nature.

### § 2.2 Basic Encoding Models

1. **Subject of investigation:**  $\mathbb{P}(\text{response} | \text{stimulus})$
2. Simplest linear mapping

$$r(t) = \phi s(t)$$

3. Temporal filters

$$r(t) = \sum_{k=0}^n s_{t-k} f_k$$
$$r(t) = \int_{-\infty}^t s(t-\tau) f(\tau)$$

#### 4. Spatial filters

$$r(x, y) = \sum_{x_0=-n}^n \sum_{y_0=-n}^n s_{x-x_0, y-y_0} f_{x_0, y_0}$$
$$r(x, y) = \iint s(x - x_0, y - y_0) f(x - x_0, y - y_0) dx_0 dy_0$$

- **Gaussian differences** can be used to simulate retina cells by mimicking excitatory centers and inhibitory retinal ganglion
- Especially useful for detecting local changes, or borders in spacial presentations

#### 5. Spatiotemporal filters

$$r_{x,y}(t) = \iiint f(x, y, \tau) dx dy d\tau$$

#### 6. Problems associated with linear filtering

- An infinite increase in the dot product of  $s \cdot t$
- Inability to deal with negative inputs and outputs
- These problems have introduced an output generation function (just like the activation function in a neural network, though this comes subsequently)

### §2.2.1 Feature Selection

1. **Rationale for feature selection** Mapping the stimulus to a feature with the the same number of dimensions as itself would require tremendous computing power. Consequently, more re presentative and more efficient features are needed.
2. **Spike-triggered Average** The core idea of STA is to average the stimuli that precede the firing of a neuron over many trials. This average stimulus can then give insights into the features of the stimulus that are most effective in driving the neuron's response.
  - Using white noise for stimuli is a very common choice since the noise levels are distributed without bias or predisposition.
  - Then, the idea of **reverse correlation** is presented, where we work backwards from the observed responses to infer properties that trigger a spike.
  - The spike-triggered average for a time lag  $\tau$  within the window  $\delta t$  is given by

$$STA(\tau) = \frac{1}{N} \sum_{i=1}^N s(t_i - \tau)$$

where  $N$  is the total number of spikes observed,  $t_i$  are the times at which spikes occurred, and  $\tau$  is the specific time lag before each spike at which we are looking at the stimulus.

3. **Principal Component Analysis** can identify multiple orthogonal dimensions which captures a broader range of significant features instead of only one primary feature in the STA.
4. **Output generation function** maps the encoded values within a certain range.

### §2.2.2 Introducing Variability

1. **Problems associated with using gaussian distribution** In real life, neurons aren't living in a world of white noise and the statistics of this stimulus used to sample a model affect the predicted results. By choosing white noise than some more natural stimulus cannot model variability since not matter how you filter it, it's always Gaussian with no special structure or directional bias in the stimulus itself.
2. **Kullback leiber divergence** is used to determine whether spikes are related or unrelated to the stimulus. The extent to which the filter is discriminative is modeled by

$$D_{KL}(P(s), Q(s)) = \int \log_2 \frac{P(s)}{Q(s)} ds$$

where the  $D_{KL}$  between spike conditional and prior distributions is maximized.

3. Given an interval  $T$  and sub-intervals  $\Delta t$ , a binomial formula gives the probability of finding  $n$  spikes in a certain interval given a total of  $k$  spikes in  $T$ :

$$P\{n \text{ spikes during } \Delta t\} = \binom{n}{k} p^n (1-p)^{k-n}$$

where  $p$  is the probability of a spike occurrence. When  $k \rightarrow \infty$  while keeping  $r = kT$  constant, the results turns into a poisson spiking process

$$P\{n \text{ spikes during } \Delta t\} = e^{-r\Delta t} \frac{(r\Delta t)^n}{n!}$$

## Week 3 Neural Decoding Models

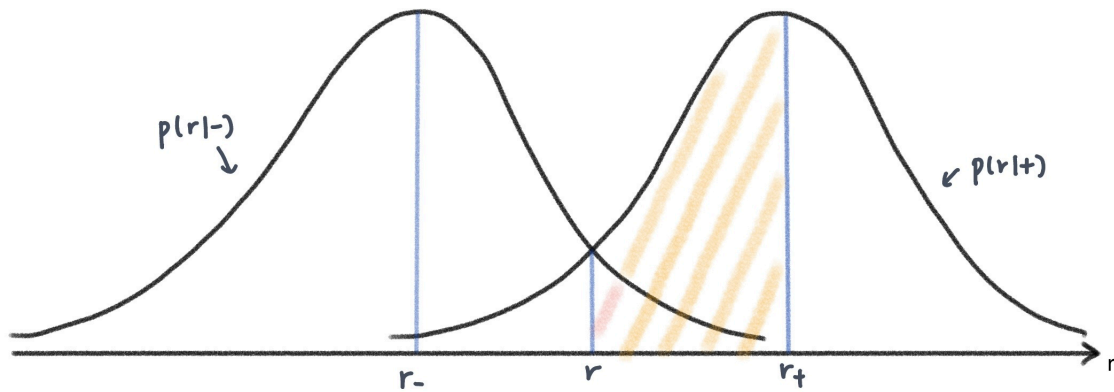
**Subject of investigation:**  $p(s|r)$ , the probability of stimulus given responses, to approximate an input-output function.

### §3.1 Single Neuron Decoding and Signal Detection Theory

We first consider a single-neuron scenario where

1. It is to decide the overall motion in groups of random moving dots with different coherence levels.
2. The decisions are binary, defined as plus (+) and minus (−).

A simple decoding procedure in this context is to determine the firing rate  $r$  during a trial and compare it to a threshold number  $z$ . If  $r \geq z$  we report (+), and if  $r < z$  we report (−).



- Define  $\alpha(z) = p(r \geq z|-)$ ,  $\beta(z) = p(r \geq z|+)$
- $p(r \geq z|-) + p(r < z|-) = 1$ ,  $p(r \geq z|+) + p(r < z|+) = 1$
- To make  $z$  the most effective threshold in discriminating the + or − stimulus, we find

$$\arg \max_z \frac{\beta(z) + 1 - \alpha(z)}{2}$$

which is an alternative expression for the sum of probabilities in correct scenarios,  $\frac{p(r \geq z|+) + p(r < z|-)}{2}$ . In the diagram shown above, it's trivial that  $z$  should take the value of  $r$  on the intersection of two probability density curves.

- The accuracy of our designation is hence:

$$p_c = p(+ )p(r \geq z|+) + p(-)[1 - p(r \geq z|-)]$$

**Nonlinear Separation of Signal and Noise** Prior probabilities are very important in the decision-making process. For example, biologically, some neuron responses are very rare at certain light levels, resulting in the resize of probability density functions.

### Constructing with Penalty and Cost

- $\text{Loss}_- = L_-p(+|r)$ ,  $\text{Loss}_+ = L_+p(-|r)$ , note that  $L_+$  denotes choosing − with a factual answer +
- Likelihood test when  $\text{Loss}_+ < \text{Loss}_-$  and answering +:

$$\frac{p(r|+)}{p(r|-)} > \frac{L_+p(-)}{L_-P(+)}$$



## §3.2 Population Coding and Maximum Likelihood Estimation

**Population Coding** Examples with cricket cercal cells and M1. Includes specific modeling about realistic observations, using cosine tuning curves, directional comparisons, and value normalization.

The examples in population coding includes modeling of specific scenarios, and the following sections will discuss three other approaches for decoding an arbitrary continuous stimulus. They are Maximum Likelihood Estimation (MLE), Maximum A Posteriori Estimation (MAP), and Bayesian Estimation. The lectures on Coursera for this section appear somewhat disorganized; therefore, the content has been re-structured while retaining all the essential details. Note that the subject investigation shifts from  $p(s|r)$  to  $p(s|\mathbf{r})$ .

### §3.2.1 Maximum Likelihood Estimation (MAP)

Consider an arbitrary tuning curve for neuron  $a$  which investigates the average firing rate:

$$f_a(s) = r_{\max} \exp\left(-\frac{1}{2}\left[\frac{s - s_a}{\sigma_a}\right]^2\right)$$

Note that instead of a probability density function,  $f_a$  is mapping stimulus parameter to the firing rate of neuron  $a$ , and we are not sure of the most possible  $s$  that results in the observed spikes. The first assumption in this model is **the sufficient coverage** of neurons, where

$$\lim_{N \rightarrow \infty} \sum_{a=1}^N f_a(s) = \text{Const}$$

which leads to constant sum of firing rate among all neuronal responses irrespective of the stimulus. According to the Bayes rule,

$$p(s|r) = \frac{p(r|s)p(s)}{p(r)}$$

where  $p(s|r)$  is a posteriori,  $p(r|s)$  is likelihood,  $p(s)$  is prior, and  $p(r)$  is evidence. **Maximum Likelihood Estimation** (MLE) focuses on  $p(r|s)$ , and decodes the stimulus by finding the  $s^*$  that maximizes the likelihood of our choice, or the most probable results that leads to our observations.

Poisson firing is assumed, as we investigate the probability of the number of spike occurrences in time interval  $T$  for neuron  $a$  given  $s$ :

$$P_T(r_a|s) = \exp(-f_a(s)T) \frac{(f_a(s)T)^{r_aT}}{(r_aT)!}$$

where  $\phi$  is tuned to  $f_a(s)T$ , the expected value of spikes and  $k$  is tuned to  $r_aT$ , the number of observed spikes in  $T$ . Since we have  $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$ ,

$$p(\mathbf{r}|s) = \prod_{a=1}^N p(r_a|s)$$

assuming that each neurons fires independently without inner correlation. To linearize the function, we take the logarithm of  $p(\mathbf{r}|s)$ :

$$\ln p(\mathbf{r}|s) = \sum_{a=1}^N (r_aT \ln(f_a(s)T) - f_a(s)T - \ln(r_aT!))$$

We take partial derivative of  $\ln p(r|s)$  and find its zero in search for the global maximum,

$$\frac{\partial \ln p(r|s)}{\partial s} = r_aT \sum_{a=1}^N \frac{f'_a(s)}{f_a(s)}$$

where the other terms are reduced due to the sufficient coverage assumption and constants. The solution,  $s^*$ , of this partial derivative equivalent to zero is stated as:

$$s^* = \frac{\sum_a r_a s_a / \sigma_a^2}{\sum_a r_a / \sigma_a^2}$$

If the standard deviations are all considered the same,  $s$  will then be:

$$s^* = \frac{\sum_a r_a s_a}{\sum_a r_a}$$

which is similar to the idea of centroid, or center of mass in physics.

### §3.2.2 Maximum A Posteriori Estimation (MAP)

In MAP, instead of likelihood, prior distribution  $p(s)$  is taken into consideration and the subject of investigation turns into  $p(r|s)p(s)$ . The logarithm of this form, again, can be formulated as

$$\ln p(s|\mathbf{r}) = \ln p(\mathbf{r}|s) + \ln p(s) - \ln p(\mathbf{r})$$

Maximizing this determines the MAP estimate,

$$T \sum_{a=1}^N \frac{r_a f'_a(s)}{f_a(s)} + \frac{p'(s)}{p(s)} = 0$$

Let  $s_{\text{prior}}$  be the mean and  $\sigma_{\text{prior}}$  be the variance, using the gaussian array of tuning curves, we derive

$$s_{MAP} = \frac{T \sum r_a s_a / \sigma_a^2 + s_{\text{prior}} / \sigma_{\text{prior}}^2}{T \sum r_a / \sigma_a^2 + 1 / \sigma_{\text{prior}}^2}$$

When data size increases, the ratio  $p(s)/p(r)$  become more and more independent of  $s$  and the value of  $s_{MAP}$  will be approaching  $s^*$ .

### §3.2.3 Bayesian Estimation

Bayesian inference is based on the minimization of a particular loss function  $L(s, s_{\text{bayes}})$  that quantifies the "cost" of reporting the value of  $s_{\text{bayes}}$  when the correct answer is  $s$ . The loss function is mapped to the a posteriori distribution  $p(s|\mathbf{r})$ . Hence, we minimize

$$\int L(s, s_{\text{bayes}}) p(s|\mathbf{r}) ds$$

The loss function can be an easy mean square error:

$$L(s, s_{\text{bayes}}) = (s - s_{\text{bayes}})^2$$

To solve for  $\arg \min_{s_{\text{bayes}}} \int L(s, s_{\text{bayes}}) p(s|\mathbf{r}) ds$ , we take the partial derivative of this function and  $s_{\text{bayes}}$  will be the zero of this partial derivative. The solution is

$$s_{\text{bayes}} = \int s p(s|\mathbf{r}) ds$$

which is exactly the spike-triggered average (described in previous sections weighted) by the a posteriori distribution.

## Week 4 Information Theory

Week 4 discusses the applications of information theory in computational neural science, specifically evaluating encoding and decoding models, or explaining code principles.

### §4.1 Entropy and Information

#### §4.1.1 Fundamentals

According to information theory, given an event  $E$ , the information  $I$  derived from this event will be

$$I(E) = -\log_2 P(E)$$

where  $P(E)$  is the probability of event  $E$ . The amount of information with regard to an event can also be interpreted as the surprisal brought by knowing this event. As the probability of  $E$  decreases, it carries more information and gives us more surprisal. The definition of **entropy**  $H$  of random variable  $X$  is given by

$$H(X) = \sum_i p(x_i) \log_2(x_i)$$

which is the expectation, or average value of the information we can obtain.

#### §4.1.2 Mutual Information and Kullback-Leiber Divergence

Recall that in neural encoding, we predict the probability of response given a stimulus. In this mapping, we aim to maximize the ability to generate stimulus-driven variations in the output. How to achieve this? Mutual information may work as a possible solution.

In probability theory, mutual information is a quantity that measures a relationship between two random variables that are sampled simultaneously. Consider two random variable  $X$  and  $Y$ , the mutual information  $I(X, Y)$  is given by

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

where  $H(\cdot)$  defines the entropies of the respective random variables. Mutual information quantifies the information received of a random variable while observing another random variable, or, how independent  $X$  and  $S$  are. The larger mutual information is, the higher variability in our mapping.

Prior to establishing the principles of mutual information in neural encoding, we begin by defining the context. Say, we have two stimulus  $s_+$  and  $s_-$ , where a constant error probability  $q$  is assigned to model the chance of responses:

$$\begin{aligned} p(r_+|s_+) &= 1 - q, & p(r_-|s_+) &= q \\ p(r_-|s_-) &= 1 - q, & p(r_+|s_-) &= q \end{aligned}$$

Hence, the mutual information of  $S$  and  $R$  is given by

$$I(S; R) = \sum_r p(r) \log_2(r) - \sum_s [-\sum_r p(r|s) \log_2 p(r|s)]$$

which models the variability of response due to noise and quantifies how independent  $R$  and  $S$  are.

Notably, this formulation aligns with what we've already discussed in week 2, the **Kullback-Leiber Divergence** for evaluating stimulus-response mappings. The KL Divergence version of mutual information

measures the distance between the joint distribution  $P(R, S)$  and assumed independent joint distribution  $P(R)P(S)$ . This definition is exactly equivalent to the already established formulations:

$$\begin{aligned}
I(S; R) &= D_{KL}[P(R, S), P(R)P(S)] \\
&= \int p(s, r) \log_2 \frac{p(r, s)}{p(r)p(s)} ds dr \\
&= \int p(s, r) \log_2 \frac{p(r|s)}{p(r)} ds dr \\
&= \int p(s, r) [\log_2 p(r|s) - \log_2 p(r)] ds dr \\
&= \int p(s) p(r|s) \log_2 p(r|s) ds dr - \int p(s, r) \log_2 p(r) dr \\
&= H(R) - H(R|S)
\end{aligned}$$

It is intuitive that when there is a great distance between joint distribution and assumed independent joint distribution, there is a maximum amount of variability in the response modeling based on observing a stimulus.

## §4.2 Computing Information for Spike Trains

In a spike sequence, information can be modeled by words. Consider binary words with letter size  $\Delta t$ , length  $T$  constituted by spikes denoted by 0 or 1, the mutual information is then calculated by the following steps:

1. Compute  $p(w_i)$ , calculate entropy of the word  $H(w) = -\sum_i p(w_i) \log_2 p(w_i)$
2. Take a stimulus sequence and repeat many times
3. Obtain a set of words  $P(w|s(t))$
4. Calculate noise entropy,  $H_{\text{noise}} = \langle H(P(w|s_i)) \rangle_i$ , averaged over  $i$

## §4.3 Coding Principles

This section will set out to introduce three essential interpretive questions for coding:

1. What are some of the challenges posed by natural stimuli?
2. What is the responsibility of neural systems?
3. What are the principles that shape the way neural codes work?

The responses to these questions are the keys to understanding the rationale behind computational models on neural activities.

Natural stimulus has a huge dynamic range, varying over many orders of magnitude, such as changes in light intensity from dawn to midnight. As a result, these stimulus also have structure at many scales. To figure out a general standard to take effect in these scenarios, **the Efficient Coding Principle** states that neurons' spikes should be highly informative about an ensemble of stimuli, minimizing the spikes needed to transfer maximum information.

Laughlin (1981) proposed a simple coding procedure that enhances a neuron's information capacity. According to empirical evidence, a good encoder should match the outputs to its distribution of inputs for maximum entropy output.

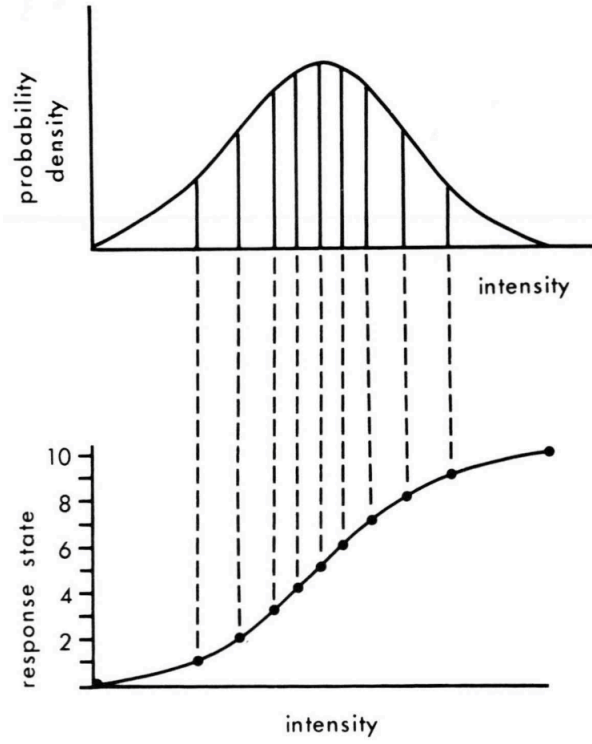


Figure 1: The coding strategy for maximizing a neuron's capacity [1].

Theoretically, the response function should look like the cumulative integral of stimulus distribution as we encode equal shares of distribution. In this case, total entropy is maximised because there is maximum amount of surprisal arising from equal chances of occurrence in each response state. Where there is an external change and stimulus distribution varies, neurons will be very sensitive and alter its input-output function, or, its slope at half-max. This property is known as **adaptive representation of information**. The adaptation process can also be modeled by Kullback-Leiber divergence where we find the most informative mapping based on calculating mutual information.

Now our opinion argues the neurons should be independently as possible, since

$$H(R_1, R_2) \leq H(R_1) + H(R_2)$$

joint distribution always indicates less information than independent distribution. However, correlation may also bring about some advantages:

1. A population of dependent neurons may work together to correct errors and achieve robust coding.
2. Dependent neurons may help with discrimination as collective responses are likely to override small deviations.

**Redundancy reduction** stresses the benefits of correlation and points out it is desirable to use fewer spikes as possible by organizing encoding properties. In reconstruction tasks, there is a cost term which tends to limit the number of coefficients needed:

$$I(\vec{x}) = \sum_i a_i \phi_i(\vec{x}) + \epsilon(\vec{x})$$

$$E = \sum_{\hat{\vec{x}}} (I(\vec{x}) - \sum_i a_i \phi_i(\vec{x}))^2 + \lambda \sum_i C(a_i)$$

where  $C$  is the cost term,  $C(a_i) = |a_i|$ ,  $\lambda$  is the weight of the strength of constraint,  $a$  represents the coefficients,  $\vec{x}$  is the positions on the image,  $\epsilon$  denotes the noise,  $I$  is the reconstructed scene,  $\hat{I}$  is the original scene, and  $E$  is the cost function.

**Week 5** Summary I

**Week 6** Computing in Carbon

## References

- [1] Laughlin, S. B. (1981). A simple coding procedure enhances a neuron's information capacity. *Zeitschrift Für Naturforschung. C, a Journal of Biosciences*, 36(9–10), 910–912. <https://doi.org/10.1515/znc-1981-9-1040>