Mathematical Foundations for Computational Neuroscience

Li Kevin Wenliang, Yuxiu Shao

These notes provide a brief overview of key mathematical concepts essential for theoretical work and computational modelling. The target audience are people with neural/cognitive science background. Unlike other more comprehensive texts, such as Mitra & Bokil (2007) and Gabbiani & Cox (2017), these notes focus on giving a light taste of the most foundamental concepts rather than providing a rigorous and thorough course on how results are derived and connected. We provide examples and excercises that are easy to work with in order to form intuitions of the concepts introduced.

Contents

1 The mathematical language

Working in a multi-disciplinary field requires precision—we have to be very precise about what we express to avoid embarrassing confusion. Some terminologies and jargons may carry the same words but can mean different things across fields. For example, what do you think the following concepts mean in the field of broader computational neuroscience?

1. *activation*
2. *representation*
3. *inference*
4. *learning*

Depending on your background, these words may induce a specific definition or an uncertainty over multiple concepts. These confusions hinder scientific progress, because people can be expressing opinions about these words for hours and realising that they have been talking about completely different topics. Smaller but nuanced differences are much more difficult to spot, and can lead to frustrations and ineffective collaborations.

In any quantitative research field, mathematics provides a very efficient language for exchanging ideas. Here, efficiency means brevity, precision, rigour, and, most importantly, clarity. **Fundamentally, we as non-mathematicians do *not* learn maths in order to show off our hundreds of lines of proofs with techniques few readers can understand**; rather, we should all use maths as a basic tool to help us think, learn and communicate, just like any other language like English, Chinese, or musical notations.

1.1 Basic notations

Here are some very common notations.

1. Sets and set-builder notations (link): {variable | optionalpredicate}, e.g.
   - ϕ, the empty set;
   - $\{1, 3, 5, 7\}$;
   - $\{1, 2, \ldots, 100\}$;
   - $\{(a, b) \in \mathbb{R}^2 \mid (a + b = 5) \wedge (a > 3)\}$;
2. Set operations
   - product($\times$): $\{1, 2\} \times \{3, 4, 5\} = \{(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)\}$
   - exclusion($\backslash$): the function $f(x) = 1/x$ is defined on $\mathbb{R}\backslash\{0\}$
   - $\cup$ (union) and $\cap$ (intersection)
3. Defining a symbol $:=$ or $\triangleq$

$$r := 5, \quad v := [0, 1, 2], \quad f(x) := x^2.$$

"Definition" differs semantically from "equality", contrast the following

$$1 + 2 = 3, \quad \pi = 3.14\ldots, \quad a := 1 \Rightarrow a + 3 = 4, \quad f(x) := x^2 \Rightarrow f(3) = 9$$

1.2 Scalar and vector spaces

Every time we think about a number, we must first specify its space. This provides the most basic abstraction in order for us to leverage known results in mathematics to help us reason about the brain.

1.2.1 Space notations

We review some well-known variable spaces, their notations, and examples in neuroscience.

1. $\mathbb{Z}$, integers: index to each neuron in a population;
2. $\mathbb{Z}_+$, positive integers;
3. $\mathbb{N}$, natural numbers: spike counts within a time window;
4. $\mathbb{R}$, real numbers: membrane potentials, imagining recordings;
5. $[a, b]$ a closed real interval: the probability of some event is in $[0, 1]$;
6. $\mathbb{R}_+$, non-negative real numbers: firing rate, frequency.

It is a good habit to define variables by telling the reader which domain the variable belongs to using the "$\in$" notation. We can see how much information these very short symbols provide in the following example.

Example 1. Our brain has $m \in \mathbb{N}$ neurons, each of which has its own firing rate $r \in \mathbb{R}_+$. We associate each neuron with an index $i \in \mathbb{Z}$. Within a given time bin $[\tau_1, \tau_2]$ for $\tau_1 < \tau_2 \in \mathbb{R}$, the i'th neuron produces $n_i \in \mathbb{N}$ spikes. Other than spikes, fluorescence imaging also provides us with indirect measurements of neural activity. We usually use $(\Delta F/F) \in \mathbb{R}$ to represent how active a putative neuron is, where $F \in \mathbb{R}_+$ is a baseline fluorescence intensity.

Although the text above may have gone too far with specifying these commonplace variables, we can only gain more clarity using very efficient notations. This can be particularly helpful when communicating to an audience of diverse background who might not recognise what these numbers are ($\Delta F/F$ can be very unclear) but may provide valuable insights to your modelling effort.

So far, we have been talking about scalars. Since we usually deal with mulitple variables, vector representations can be quite convenient. Common vector spaces are

1. $\mathbb{R}^d$, space of d-dimensional real vectors.
2. $\mathbb{R}^{m\times n}$, space of m-by-n real matrices: connectivity of two neural populations, m samples of n-dimensional variables.
3. Δ^d, the probability simplex in d dimensions:

$$\Delta_d := \left\{ p \in [0,1]^d \;\middle|\; \sum_{i=1}^{d} p_i = 1 \right\},$$

   probabilities of a coin toss ($d = 2$) or a die throw ($d = 6$).

4. $\mathbb{1}_d$, space of d-dimnensional one-hot vector (no stable convention): discrete event of d categories.

$$\mathbb{1}_d := \left\{ v \in \{0,1\}^d \;\middle|\; \sum_{i=1}^{d} v_d = 1 \right\}$$

5. $\mathbb{S}_d$, d-dimensional hypersphere: $\mathbb{S}_1$ is the same as $[-\pi, +\pi]$ with circular boundaries, examples are dot motion orientations and head directions.

1.2.2 Vector norms and inner products

The distance between two scalars $a, b \in \mathbb{R}$ is obviously $|a - b|$. However, outside the scalar real domain, the notion of the distance between two elements is less straightforward. For real vectors, we can measure the distance in several ways:

Definition 1. Let $p \geq 1$, the p-norm of a real vector $v \in \mathbb{R}^d$ is defined as

$$\|v\|_p := \left(\sum_{i=d}^{d} |v|_i^p \right)^{1/p}.$$

In particular, we have $\|v\|_2$ as the Euclidean norm, and $\|v\|_\infty = \max_i |v_i|$ as the infinity norm. The distance between two vectors $u, v \in \mathbb{R}^d$ can be measured by $\|u - v\|_p$. $\|u - v\|_2$ is the Euclidean distance.

The norm satisfies intuitive and useful properties so that we can essentially treat them as distances. A larger value of p puts more emphasis on the elements with larger magnitudes.

Exercise 1. How can we define a distance for $\mathbb{S}^d$?

The inner product between two vectors $a, b \in \mathbb{R}^d$ is written as $a \cdot b = \sum_{i=1}^{d} a_i b_i$, or using matrix notation, $a^\intercal b$, where $\intercal$ denotes transpose. The equivalent of the inner product for matrices $A, B \in \mathbb{R}^{m \times n}$ is $\mathrm{Tr}[A^\intercal B]$, where Tr is the trace operator for a square matrix: $\mathrm{Tr}[A] = \sum_i a_{ii}$

Exercise 2. Show that $\mathrm{Tr}[A^\intercal B] = \mathrm{Tr}[B^\intercal A] = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij}$?

One can cyclicly permute the matrices inside the trace operator without affecting the result.

1.3 Functions

Informally, functions are mappings from a domain (input) space to a codomain (output) space such that each element in the domain maps to a single element in the codomain. We use the concept of functions almost everywhere in studying cognition, because cognitive capabilities are functions from some input (e.g. stimuli, contexts) to some output (e.g. behaviours, neural activities). What we usually call a model is in fact functional classes parametrised in ways that are helpful for us to understand the brain.

1.3.1 The "to" symbol $\to$

The key notation to describe what a function f does is $\to$, which specifies the domain and codomain. We give three examples.

Example 2. The vector norms in Definition 1 is a function $\mathbb{R}^d \to \mathbb{R}_+$.

Example 3. Suppose we study some deterministic spiking mechanism given the membrane potential of a neuron. This process can be treated as a function that maps a membrane potential v to the binary event of where the neuron spikes or not. Such a function can be expressed as

$$f : \mathbb{R} \to \mathbb{1}_2$$

where the one-hot vector indicates the event of spike or no spike.

Example 4. Consider the task expressed verbally as: *I want to study how humans recognise different objects in natural images.* We can describe the function of image object recognition as

$$f : \mathbb{R}^{w \times h \times c} \to \Delta_k$$

where $w, h, c \in \mathbb{N}$ are the width, height, and channel of an image. With this notation above, it becomes very explicit what the function is doing. We have abstracted images as a matrix, and object classes as one-hot vectors. Then, we can use all mathematical tools built for mapping between those spaces to construct a model of image recognition, such as logistic regression or deep neural networks.

1.3.2 The "mapsto" symbol $\mapsto$

When we do not need a handle to the function (the symbol "f") but just need to describe a mapping, we can use $\mapsto$. Unlike $\to$ which links a set to another set, $\mapsto$ links an *element* of the domain to an *element* of the codomain. As such, this is a much more explicit specification of a function without defining a function symbol.

Example 5. The Gaussian function $x \mapsto \exp(-x^2)$ is often used to define smoothing filters, probability distributions, and so on.

1.3.3 Functional norm and classes

Functions can be overly expressive, and one needs to consider a restricted set of functions for certain applications. Let us consider a common class of functions in mathematics, and then discuss its relevance to neuroscience.

Definition 2. Let $p \geq 1$, the class of p-integrable functions supported on $\mathbb{R}$ is defined as

$$L_p := \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; \int |f(x)|^p \mathrm{d}x < \infty \right\}$$

Typical examples of functions in L_p include the Gaussian function $x \mapsto \exp(-x^2)$ and $x \mapsto 1/x^2$. Non-examples are more abundant, including the hyperbolic function $x \mapsto 1/x$, the exponential function $x \mapsto \exp(x)$, polynomials and periodic functions. As we will see later, the probability density function of a continuous random variable on $\mathbb{R}$ is a strict subclass of L_1 satisfying non-negativity and normalization constraints.

Intuitively, L_p functions usually have vanishing tails as $x \to \pm\infty$. This class of function is commonly used for transient, finite-response filters. Examples include the kernel functions used in spike-time dependent plasticity (STDP) rules, and the haemodynamic response function in BOLD signals recorded in fMRI.

To study human image recognition in Example 4 in our, we also want our functions to be biologically plausible. One can loosely define such function class as

$$\mathscr{B} := \left\{ f : \mathbb{R}^{w \times h \times c} \to \Delta^k \;\middle|\; f \text{ is biologically plausible} \right\}.$$

Of course, what is deemed biologically plausible is at least partially up for interpretation. Various quantifiable criteria may be used to constrain the function space. Traditionally, reasonable constraints for a biologically plausible function may be smoothness or continuity (e.g. Lipschitz continuity and differentiable continuity).

Example 6. (Gaussian) Radial basis functions are a class of functions

$$\mathscr{R} := \left\{ f : \mathbb{R}^d \to \mathbb{R} \;\middle|\; f(x) = \sum_{i=1}^{m} w_i e^{-\frac{\|x-\mu_i\|_2^2}{2\sigma^2}},\ \forall x \in \mathbb{R}^d \right\},$$

where parameters are the $\{w\}_{i=1}^m \in \mathbb{R}^m$, the centers $\mu_i \in \mathbb{R}^d$ for all $i \in \{1, \ldots, m\}$, and width $\sigma >$. This function is infinitely-differentiable so is deemed very smooth. It can be used to approximate continuous functions, such as the density functions in examplar theory.

Example 7. The class of multinomial logistic functions is

$$\mathscr{L} := \left\{ f : \mathbb{R}^d \to \Delta_k \;\middle|\; f(x) = \begin{bmatrix} \frac{\exp(-w_1 \cdot x)}{\sum_{i=1}^d \exp(-w_i \cdot x)} \\ \ldots \\ \frac{\exp(-w_k \cdot x)}{\sum_{i=1}^d \exp(-w_i \cdot x)} \end{bmatrix},\ \forall x \in \mathbb{R}^d \right\},$$

where the parameters are weights $\{w_i \in \mathbb{R}^d\}_{i=1}^k$. This class of function is smooth and commonly used for classification problems.

Exercise 3. A subject sees the orientations of two Gabor patches and is asked to report whether the second one is more clockwise or counterclockwise. Write down a possible functional space that contains this behaviour.

Exercise 4. A subject is given two bandits, one on the left and one on the right. When a bandit is pressed, it produces a reward of value 1 or 0 with unknown probability. The subject makes 10 decisions in total according to a policy: it maps past decisions and outcomes into a distribution of choosing the left or the right bandit. Define the function space the policy.

1.4 Formulating constrained optimisation

By the time you tried to explain some data with a model, you most likely had used some sort of optimisation method to fit the model parameters. This is easily the most crucial and time-consuming part of a research project, and yet it only goes into the appendix of published work. Usually, you will face many challenges at this stage: the model does not converge, it finds a trivial solution that's meaningless, or it captures very little variance of the data.

Your life may be saved by consulting your peers who are more trained in mathematics, statistics or machine learning. The obstacle is that they speak a very different language—they most likely prefer a precise formulation of the problem you are dealing with. Here, we introduce the mathematical language for formulating optimisation problems. This builds on the notations we have introduced so far. You will see that fluency in this langauge makes it much easier to communicate and collaborate with more mathematically-minded people.

You should be aware that a typical optimisation problem requires the following ingredients to be specified:

1. A dataset $\mathbb{D}$;
2. A model/function class $\mathbb{M}$ with associated parameters space Θ;
3. An objective function $\mathcal{L} : \mathbb{D} \times \Theta \to \mathbb{R}$, potentially with penalty terms.
4. Constrains $\mathcal{C} : \mathbb{D} \times \Theta \to \{\text{True}, \text{False}\}$

With these elements, one can formulate the optimisation problem as

$$\min_{\theta \in \Theta} \mathcal{L}(\mathbb{D}, \theta) \text{ subject to } \mathcal{C}(\mathbb{D}, \theta)$$

The constraints can be equalities (e.g. normalisation), inequalities (e.g. positivity), or restriction to a subspace (e.g. within $\mathbb{S}_d$ Below, we illustrate with some examples. The set $\mathbb{M}$ and Θ do not need to be clearly written out but should still be clearly defined. Here are a few examples.

Example 8. We throw a coin with an unknown probability of landing on heads. We observe 2 heads and 3 tails. What is the probability of heads (p_h) that gives the maximum likelihood of the observed data?

1. $\mathbb{D} := \{0, 0, 1, 1, 1\}$;
2. $\mathbb{M} := \{\text{Bernoulli}(p_h) \mid p_h \in [0, 1]\}$, the set of all Bernoulli distributions with parameter p_h.
3. $\mathcal{L}(\mathbb{D}, p_h) := p_h^2(1 - p_h)^3$,
4. $p_h \in [0, 1]$.

Example 9. In ordinary least-squares regression of the form $\hat{y} = Ax + b$,

1. $\mathbb{D} := \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^l$, and $k, l \in \mathbb{Z}_+$ specified by data.
2. $\mathbb{M} := \{f : \mathbb{R}^k \to \mathbb{R}^l \mid f(x) = Ax + b\}$, where parameters $A \in \mathbb{R}^{l \times k}, b \in \mathbb{R}^l$.
3. $\mathcal{L}$ is given by
$$\mathcal{L}(\mathbb{D}, \theta) = \frac{1}{n} \sum_{i=1}^{n} \|Ax_i + b - y_i\|_2^2 + \lambda \left(\text{Tr}[A^\intercal A] + \|b\|_2^2 \right)$$
where $\lambda > 0$ is a regularisation strength parameter.
4. no constraints on A.

Example 10. In an autoencoder, we compress data x to a lower-dimensional representation z using an encoder E and decoder D.

- $\mathbb{D} := \{x_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^k$. $V := [x_1; \ldots; x_n]$
- $\mathbb{M} := \{E : \mathbb{R}^k \to \mathbb{R}^m, D : \mathbb{R}^m \to \mathbb{R}^k \mid h = E(x), \hat{x} = D(h),\ h \in \mathbb{R}_+^m\}$, where m is a parameter we specify, D and E may be matrices (linear autoencoder, $\theta = \{E \in \mathbb{R}^{m\times k}, D \in \mathbb{R}^{k\times m}\}$) or neural networks with parameters θ composed of weights and biases.
- $\mathcal{L}$ is given by
$$\mathcal{L}(\mathbb{D}, \theta) = \frac{1}{n}\sum_{i=1}^{n} \|D(E(x_i)) - x_i\|_2^2$$
- We may constrain $E(x) \in \mathbb{R}_+^m$ for all $x \in \mathbb{R}^k$.

Example 11. Given an autoencoder in Example 10, if $\{x_i\}_{i=1}^n$ is a smoothly varying time-series, we can add a penalty term to $\mathcal{L}$:

$$\lambda \sum_{i=1}^{n-1} \|E(x_i) - E(x_{i+1})\|_2^2,$$

which enforces smoothness of the encoder E

Exercise 5. You run the bandit experiment in Exercise 4 on your subject Alice and collect her behavioural data. Each session has 10 trials, and Alice reported her choices for these 10 trials. Now you want to build a model of her policy and optimise the model parameters on Alice's data. Define a reasonable model and formulate the optimisation problem.

1.5 Final remarks

The mathematical language we introduced should help you think more clearly when using and building mathematical models. Like all other languages, it needs regular practice. We should also learn to ask clarifying questions framed with mathematical language, and be brave to do so. This will prove to be a valuable skill for doing quantitative research.

A caveat of thinking too deeply in maths is the potential loss of intuition and inspirations. You should not rely heavily on maths to build much of their intuitions when developing a novel theoretical framework, because making things precise is also quite costly, regardless of whether your idea is sound or not. However, you should gradually sculpt out a mathematical description to your work, making your intuitions concrete and rigorous. This often reveals pitfalls and mistakes before spending too much time, energy and excitement on a wrong model. Iterating over these up-and-downs can lead to great success!

2 Linear algebra

Linear algebra studies linear equations and linear maps through vectors and matrices.

2.1 Matrix-vector product

The matrix-vector product is the most fundamental use of a matrix as a linear operator. We will use this operation to understand what matrices do.

Definition 3. A Linear map $f : V \to W$ from vector space V to vector space W satisfies the following conditions: for any vectors $u, v \in V$ and any scalar $c \in \mathbb{R}$,

- $f(u + v) = f(u) + f(v)$
- $f(cu) = cf(u)$

A matrix $A \in \mathbb{R}^{m \times n}$ is a linear map from input space $\mathbb{R}^m$ to output space $\mathbb{R}^n$ as defined by the matrix-vector product

Definition 4. The matrix-vector product between a matrix $A \in \mathbb{R}^{m \times n}$ a vector $b \in \mathbb{R}^n$ is defined as

$$Ab = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,n} \\ a_{21} & a_{22} & \dots & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,n} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_{1i} b_i \\ \sum_{i=1}^n a_{2i} b_i \\ \vdots \\ \sum_{i=1}^n a_{mi} b_i \end{bmatrix} \tag{1}$$

Exercise 6. Verify that the matrix-vector product defines a linear map.

This may be very obvious from your previous mathematical training. We now establish two views of the matrix-vector product.

2.1.1 Row view

One can see from (1) that if we define A by vectors $\{v_i\}_{i=1}^m$ with $v_i = [a_{i1}, \dots, a_{in}] \in \mathbb{R}^n$ such that

$$A = \begin{bmatrix} -v_1^\intercal - \\ -v_2^\intercal - \\ \vdots \\ -v_m^\intercal - \end{bmatrix}$$

then we have

$$Ab = \begin{bmatrix} v_1^\intercal b \\ v_2^\intercal b \\ \vdots \\ v_m^\intercal b \end{bmatrix}$$

Thus, we can view A as a stack of vectors in the input space $\mathbb{R}^n$ each of which acts on the input vector $b \in \mathbb{R}^n$

2.1.2 Column view

Another less obvious view is obtained by taking A as a bundle of column vectors $\{u_i\}_{i=1}^n$ with $v_i = [a_{1i}, \ldots, a_{mi}] \in \mathbb{R}^m$ such that

$$A = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \ldots & u_n \\ | & | & & | \end{bmatrix}$$

then we have

$$Ab = [b_1 u_1 + b_2 u_2 + \ldots + b_n u_n] = \sum_{i=1}^{n} b_n u_n \tag{2}$$

Thus, the matrix A defines a set of vectors in the output space $\mathbb{R}^m$ which are linearly weighted by the elements in the input vector b.

Which view makes more intuitive sense depends on the application.

Example 12. The spike-triggered average (STA) estimates the correlation between a spike and some inputs preceding the spike (e.g. stimulus, membrane potential, etc.). Given time bins $1:T$, suppose that we measure the inputs $\{x_t\}_{t=1}^T$ and the number of spikes within in each time bin $\{s_t\}_{t=1}^T$. To compute the STA over a given time interval of τ time steps. We then estimate the STA by the normalised matrix-vector product

$$\text{STA} = \frac{1}{\sum_{t=1}^T s_t} \begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_\tau & \ldots & x_T \\ 0 & x_1 & x_2 & \ldots & x_{\tau-1} & \ldots & x_{T-1} \\ 0 & 0 & x_1 & \ldots & x_{\tau-2} & \ldots & x_{T-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & x_1 & \ldots & x_{T-\tau+1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_T \end{bmatrix}$$

Exercise 7. Which view of the matrix-vector product is more relevant for the example above?

Example 13. We can write the prediction $\hat{y}$ in the regression problem in Example 9 using the matrix-vector product when the target variable y is in $\mathbb{R}$. Recall that the input $x \in \mathbb{R}^k$, and so the regression coefficients $A \in \mathbb{R}^{1\times k}$ and the bias $b \in \mathbb{R}$. After applying a transpose to the inputs and A, the predictions for all x_i, $i \in \{1, \ldots, n\}$ reads

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} -x_1^\intercal- & 1 \\ -x_2^\intercal- & 1 \\ \vdots & \vdots \\ -x_n^\intercal- & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1k} \\ b_1 \end{bmatrix}. \tag{3}$$

Exercise 8. Which view of the matrix-vector product is more relevant for the example above?

2.2 Basic matrix operations and special matrices

One can extend matrix-vector products to matrix-matrix products, simply by treating the second matrix as a horizontal stack of vectors.

Definition 5. For two $A \in \mathbb{R}^{m\times n}, B := [b_1, b_2, \ldots, b_k] \in \mathbb{R}^{n\times k}$, where $b_i \in \mathbb{R}^n$ for $i \in \{1, \ldots, n\}$, the matrix-matrix product is defined as

$$AB = [Ab_1, Ab_2, \ldots, Ab_k].$$

We list a few special matrices and common matrix operations.

- Identity matrix I: $AI = A$.
- Diagonal matrix D: $D_{ij} = 0$ for $i \neq j$.
- Symmetric matrix S: $S = S^\intercal$
- Square matrix: one that has the same number of rows as columns.
- Rectangular matrix: one that has distinct numbers of rows and columns.
- Matrix inverse: for a square matrix A, $A^{-1}A = I$ if A^{-1} is invertible.

These basic matrices help us define more properties of matrices. One thing to note that matrix products are associative: $(AB)C = A(BC)$ but not commutative in general: $AB \neq BA$

2.3 Span of vectors

The row and column views show that we may be able to understand the operation of a matrix in terms of vectors in its rows or columns.

Under the column view, the output of a matrix-vector product is a linear combination of its column vectors. It is important to understand the geometric property of the matrix. Let's do this step by step.

1. If there is only a single column $A = [u_1]$ and $u_1 \in \mathbb{R}^m$, then the output is a_1u_1, which is on the 1-D sub-space, or a straight line in $\mathbb{R}^m$ parallel to u_1.
2. If there are two columns, $A = [u_1, u_2]$, then the output is $a_1u_1 + a_2u_2$ which is a 2-D sub-space of $\mathbb{R}^m$, unless u_1 is parallel to u_2, in which case the output space is still a 1-D line.
3. If there are three columns, $A = [u_1, u_2, u_3]$, then the output is $a_1u_1 + a_2u_2 + a_3u_3$, which can be a 3-D, 2-D or 1-D subspace of $\mathbb{R}^m$.

More generally, the output space of a matrix A is the span of the column vectors.

Definition 6. The (linear) span of a set of vectors $\mathbb{U} = \{u_i\}_{i=1}^n$ is

$$\text{span}(\mathbb{U}) := \left\{ \sum_{i=1}^{n} a_i u_i \;\middle|\; a_i \in \mathbb{R} \;\forall\; i \in \{1, \ldots, n\} \right\}. \tag{4}$$

The dimensionality of span($\mathbb{U}$) is the number of *linearly independent* vectors in $\mathbb{U}$.

Definition 7. A set of vectors $\{u_i\}_{i=1}^n$ are said to be *linearly independent* if we cannot express any one vector by a linear combination of the other vectors. (e.g. $u_1 \neq \sum_{j=2}^n a_j u_j$ for all $a_j \in \mathbb{R}, j = \{2, \cdots, n\}$.) In other words, the equation $\sum_{i=1}^n a_i u_i = 0$ can only be satisfied by $a_1 = a_2 = \cdots = a_n = 0$.

We then have the following terminologies:

- The span of the column vectors of a matrix is called the *column space* (or range) of the matrix.
- The span of the row vectors of a matrix is called the *row space* of the matrix.
- The *rank* of matrix A, $\text{rank}(A)$, is the dimensionality of the column space.

These concepts are interconnected through the following result.

Theorem 1. *The dimensionality of the row space of a matrix equals the the dimensionality of the columns space of the matrix. They are both equal to the rank of the matrix.*

Proof. See this link □

For square matrices, we also have the following definition

Definition 8. A square matrix is said to be *full-rank* if its rank is equal to its size (number of columns). Otherwise, it is said to be *rank-deficient*.

Full-rank square matrices have many nice properties, one of them is that they are invertible, whereas rank-deficient matrices are not. Rank-deficient square matrices of size n span a subspace of $\mathbb{R}^n$. Denote its rank by r. then we say that there is a null-space of dimensionality $n - r$.

The rank is also defined for rectangular matrices, though it differs slightly from the definition for square matrix.

Definition 9. A rectangular square matrix of size $m \times n$ is said to be *full-rank* if $\text{rank}(A) = \min(m, n)$. Otherwise, it is said to be *rank-deficient*.

Example 14. A $m \times n$ rank-1 matrix L can be formed by a two vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ by

$$L = uv^\intercal$$

This matrix has a single independent row vector, because each i'th row is $u_i v^\intercal$ is a scaled version of the same row vector $v^\intercal$. Similarly, each column is a scaled version of the same column vector u.

2.4 Orthonormal basis

We have seen that one may characterise the operation of a matrix on a vector through its column vectors or its row vectors. Intuitively, we must be able to find a *minimal* set of vectors that define the column or row spaces, rather than using all of them which could be redundant due to linearly dependence. This is made possible by finding the orthonomal basis (bases for plural) of a given set of vectors.

Definition 10. A set of vectors $\{e_i\}$ are said to be orthonormal if

$$e_i \cdot e_j = \delta_{ij},$$

where δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise). As such, each vector in the orthonormal basis is a unit vector.

The space $\mathbb{R}^d$ can be spanned by d orthonormal vectors. These vectors form perpendicular coordinates, just like the Cartesian coordinates in 3D given by the orthonormal basis $[1, 0, 0]$, $[0, 1, 0]$ and $[0, 0, 1]$.

Remark 1. *Orthonormal matrices represent rotations and reflections.*

Exercise 9. Verify that the 2×2 rotation matrix and reflection matrices

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \qquad \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix}$$

are orthonomal for any $\theta \in \mathbb{R}$.

Further, as we show now, an orthonormal basis of a linear space forms a minimal set of unit vectors that can describe any vector in this space through linear combinations. Given d orthonormal basis $\{e_i\}_{i=1}^d$, where each e_i is d-dimensional, we can compute the coordinates of any vector $v \in \mathbb{R}^d$ on this basis by projection $v \cdot e_i$. Since e_i is a unit vector, this dot product gives the length (projection) of v along e_i. We can thus reconstruct the original vector by

$$v = \sum_{i=1}^{d} (v \cdot e_i) e_i. \tag{5}$$

This can be re-stated in matrix-vector product. Define $E := [e_1, e_2, \cdots, e_d]$, we have

$$v = EE^\intercal v = \begin{bmatrix} | & | & & | \\ e_1 & e_2 & \cdots & e_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} -e_1- \\ -e_2- \\ \vdots \\ -e_d- \end{bmatrix} v \tag{6}$$

Note that $E^\intercal v$ produces the length of the vector v along each unit vector of the orthonormal basis through the row view of matrix-vector product. Pre-multiplying the result by E gives the summation in (5) through the column view (2).

Since the result of (6) holds for any vector $v \in \mathbb{R}^d$, we can conclude that $EE^\intercal = I_d$, where I_d is the d-by-d identity matrix. In fact, this is one definition of orthonormal matrix.

Definition 11. A square matrix A is orthonormal if $AA^\intercal = I$

Exercise 10. Prove the following results for an orthonormal matrix E.

- $E^\intercal E = EE^\intercal = I$
- $E^{-1} = E^\intercal$

2.5 Singular value decomposition

Equipped with the tools for orthogonal matrices, we are ready to gain further understanding of the matrix-vector product by the following important matrix decomposition

Definition 12. The singular value decomposition (SVD) factorises a matrix A into two orthonormal matrices U and V and a diagonal matrix S so that

$$A = USV^\intercal$$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}}_{A,\, m \times n} = \underbrace{\begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_r \\ | & | & & | \end{bmatrix}}_{U,\ m \times r} \underbrace{\begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \end{bmatrix}}_{S, r \times r} \underbrace{\begin{bmatrix} -v_1- \\ -v_2- \\ \vdots \\ -v_r- \end{bmatrix}}_{V^\intercal, r \times n}$$

where $r = \min(m, n)$, and $s_i \geq 0$ for all i, $s_i \geq s_j$ for $i < j$

- $u_i \in \mathbb{R}^m$ is the i'th left singular vector.
- $v_i \in \mathbb{R}^n$ is the i'th right singular vector.
- $s_i \in \mathbb{R}_+$ (note the non-negativity) is the i'th singular value.
- the orthogonal U forms the basis of the output space.
- the orthogonal V forms the basis of the input space.
- the diagonal S scales the components along each coordinate of V.

The singular value decomposition always exists for any matrix. It says that any linear operation on a vector can be decomposed into three linear transformations: first a rotation/reflection by V, then a coordinate-wise scaling by S, and finally another rotation/reflection by U.

Alternatively, under the column and row views of matrix-vector product, since $Ab = USV^\intercal b$, we see that $V^\intercal b$ projects the input b onto the coordinates defined by V, and S scales these components, and the scaled components are used as coefficients to linearly combine the basis in U to product the output.

The size variable r is the maximum rank of the matrix A. When $m > n$, meaning that $r = n$, then the input basis V is square and forms the full basis of the input space $\mathbb{R}^n$ (it can reconstruct any vector in $\mathbb{R}^n$). The output space is in $\mathbb{R}^m$, but there are only $r < m$ orthonormal vectors, and thus U projects into a strict subspace of $\mathbb{R}^m$.

If we observe $s_j = 0$ for some j, then this says that the components of any input vector along the input coordinate u_j will be eliminated after multiplying by $s_j = 0$. This means that the original matrix A is low-rank. We have seen a low-rank matrix (rank-1) in Example 14. Its construction can be seen as from an SVD where there is only one singular value $s_1 = 1$.

Low-rankness of connectivity matrice is ubiquitous in the nervous systems. It is thought to produce robustness to noise or perturbation.

2.5.1 Null space

The (right) null-space of a matrix A (a.k.a kernel of A) is spanned by basis vectors orthogonal to the column space spanned by the right singular vectors.

Example 15. The matrix

$$A := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

is rank-2, because the third row is a linear combination of the first two. Its null-space has a basis vector $\mu := [0, 0, 1]^\intercal$. For any input vector $v \in \mathbb{R}^3$ applying this matrix A removes this component, so that $\mu \cdot (Av) = 0$.

In neuronal networks, if the perturbation is applied in the null-space of the connectivity matrix, then the activity is unaffected by this perturbation. For large-scale networks of n neurons, a rank r connectivity matrices with $r \leq n$ will be very robust, because the chance of the perturbation hitting the null-space is very high.

2.6 Eigendecomposition for square matrices

From here onwards, we focus on square matrices. Square matrices also have SVD, and U, S, and V matrices have the same shape. However, it is possible that $U \neq V$ and the output coordinates are mis-aligned with the input coordinates. This creates conceptual complications when we want to apply the matrix A (e.g. a connectivity matrix) a few times to an input vector (e.g. initial activations), because $A^2 = USV^\intercal USV^\intercal$, $V^\intercal U$ defines a third space in addition to U and V.

The Eigendecomposition offers a rescue. It defines a unified input and output coordinates for square matrices so that we can greatly simplify our understand of these matrices.

Definition 13. The eigendecomposition factorises a square matrix A into an orthogonal A and a diagonal Λ so that

$$A = Q\Lambda Q^\intercal$$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_{A,\, n \times n} = \underbrace{\begin{bmatrix} | & | & & | \\ q_1 & q_2 & \dots & q_n \\ | & | & & | \end{bmatrix}}_{Q,\ n \times n} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}}_{\Lambda, n \times n} \underbrace{\begin{bmatrix} -q_1- \\ -q_2- \\ \vdots \\ -q_n- \end{bmatrix}}_{Q^\intercal, n \times n}$$

where $\lambda_i \in \mathbb{C}$ is an eigenvalue of the matrix A, $q_i \in$ is the corresponding eigenvector so that

$$Aq_i = \lambda_i q_i$$

Exercise 11. Interpret the eigendecomposition as a sequence of linear transformations, using the column and row views of the matrix-vector product.

Repeated applications of the matrix A can then be seen as repeated scaling on the coordinates defined by the eigenvectors Q because

$$A^n = (Q\Lambda Q^\intercal)^n = Q\Lambda^n Q^\intercal.$$

Thanks to this property, eigendecomposition is frequently used in analysing dynamical systems. The eigenvalues are related to the stability and transient dynamics of a linear dynamical system of the form $s_t = As_{t-1}$ for a state variable s_t.

Exercise 12. For a matrix A with real eigenvalues, give the conditions on these eigenvalues such repeated multiplications of the matrix to an arbitrary vector remains finite.

The eigendecomposition and SVD of a square matrix A are tightly linked. One can show the following

- The left singular vectors of A are parallel (equal up to a sign difference) to the eigenvectors of $AA^\intercal$
- The right singular vectors of A are parallel to the eigenvectors of $A^\intercal A$
- The singular values of A are the square-root of the absolute (modulus) eigenvalues of $A^\intercal A$ or $AA^\intercal$.

2.6.1 Positive semi-definite matrices

Definition 14. Positive semi-definite matrices P are real symmetric matrices such that $v^\intercal Pv \geq 0$ for any vector v. Positive definite matrix P satisfies $v^\intercal Pv \geq 0$ for any nonzero vectors v

Positive (semi-)definite matrices are akin to positive (non-negative) numbers when restricted to 1×1 matrices, which are essentially real numbers.

Example 16. The following matrices are positive semi-definite.

- Diagonal matrix with non-negative entries.
- The rank-1 matrix in Example 14.
- A covariance matrix (see later).

Exercise 13. Prove that the above examples are indeed positive semi-definite.

3 Calculus

The most fundamental use of calculus deals with continuous change of functions defined over some continuous quantities, such as time or space. Since even a single neuron can be regarded as a function mapping from input current and to membrane potentials or spikes, calculus is essential for studying dynamics in neuronal circuits. Likewise, most perceptual capabilities can also be treated as a mapping from input stimuli to behaviours, and studying the sensitivity of behaviour to small changes in the stimulus also require calculus. In particular, **differential** calculus measures the rate of continuous change of a function over its domain, whereas **integral** calculus measures accumulation of small changes. In the following, we first review differentiation and Integration of real univariate functions $\mathbb{R} \to \mathbb{R}$, and then introduce useful rules of calculus for vector-valued functions.

3.1 Differentiation

The derivative (also known as the gradient) of a function $f : \mathbb{R} \to \mathbb{R}$ w.r.t its input $x \in \mathbb{R}$ is denoted by several ways

- most general:
$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} := \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}.$$
when this limit exists.

- prime notation, usually when the input symbol is not important:
$$f'(x) := \frac{\mathrm{d}f(x)}{\mathrm{d}x}.$$
we can then refer to the derivative as f', which is also a function.

- Newton's notation, usually for differentiating dynamical quantities (position, velocity, membrane potential etc.) w.r.t. time t,
$$\dot{v}(t) := \frac{\mathrm{d}v(t)}{\mathrm{d}t}.$$

Likewise, we can also define higher-order derivatives by repeated differentiation. For example, the secondary derivative can be denoted as:

$$f''(x) := \frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}f(x)}{\mathrm{d}x} =: \frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2}.$$

Example 17. For the function $f(x) = x^2$, we can derive its gradient by its definition:

$$f'(x) = \lim_{\delta \to 0} \frac{((x+\delta)^2 - x^2)}{\delta} = \lim_{\delta \to 0} \frac{2x\delta + \delta^2}{\delta} = \lim_{\delta \to 0} 2x + \delta = 2x$$

The computation of gradient for certain functions, like polynomials, exponentials, and sinusoids, can be derived and are available in tables.

The derivative captures how sensitive the function is to slight changes around an input x. Geometrically, the derivative of a function at a certain input is the *slope* of the function at that

input. For example, as you drive, the distance travelled changes continuously with time. A high velocity (big derivative) means that the distance travelled changes a lot for a small lapse of time.

The sign of the derivative of a function tells whether the function is increasing or decreasing as input increases. A positive (negative) gradient means the function is locally increasing (decreasing).

If the derivative is zero around an input x, then this function is at a *stationary point*—the slope of the signal is zero. There are three types of stationary points:

- a local maximum: the secondary derivative (if exists) is negative.
- a local minimum: the secondary derivative (if exists) is positive.
- a point of inflection: the secondary derivative (if exists) is zero.

It is crucial to realise that a zero gradient does not necessarily imply a local extremum (maximum of minimum), and one should always be careful about the type of stationary point.

Exercise 14. Find and classify all stationery points of the function

$$f(x) = x^2(1 - x^2)$$

3.1.1 Rules of differentiation

The *product rule* (or Leibniz rule, or Leibniz product rule) is useful for finding the derivatives of products of two or more functions. For two functions $u(x)$ and $v(x)$, the produce rule may be stated as

$$\frac{\mathrm{d}}{\mathrm{d}x}(u(x) \cdot v(x)) = \frac{\mathrm{d}}{\mathrm{d}x}u(x) \cdot v(x) + u(x) \cdot \frac{\mathrm{d}}{\mathrm{d}x}v(x). \tag{7}$$

Exercise 15. In neuroscience, the temporal response of excitatory/inhibitory postSynaptic potential (EPSP/IPSP) is described by a so-called α-function:

$$f(t) = t \cdot \exp\big(-\frac{t}{\tau}\big),\ t \in \mathbb{R}_+ \tag{8}$$

Can you use the product rule to calculate the derivative of this α-function?

The *chain rule* is a formula that expresses the derivative of the composition of two differentiable functions f and g in terms of the derivatives of f and g. More precisely, for the function $f(g(x))$, its derivative for every x can be expressed in Leibniz's notation as

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g)}{\mathrm{d}g} \cdot \frac{\mathrm{d}g(x)}{\mathrm{d}x} \tag{9}$$

Carrying the same reasoning further, given n functions $f_1,\ f_2, \ldots, f_n$ with the composite function $f_1 \circ f_2 \circ \ldots f_{n-1} \circ f_n$, if each function f_i is differentiable at its immediate input, then the composite function is also differentiable by the repeated application of the chain rule, where the derivative is expressed in Leibniz's notation as

$$\frac{\mathrm{d}f_1}{\mathrm{d}x} = \frac{\mathrm{d}f_1}{\mathrm{d}f_2}\frac{\mathrm{d}f_2}{\mathrm{d}f_3} \cdots \frac{\mathrm{d}f_n}{\mathrm{d}x} \tag{10}$$

Exercise 16. Calculate the derivative of this function:

$$f(x) = \frac{a - x}{1 - \exp(-bx)},\ x \in \mathbb{R} \tag{11}$$

In summary, we have seen that the derivative measures the rate of change of a function, and we can use the derivatives to identify stationary points.

3.1.2 Integration

The integral of a functions is related to the accumulation of the function value over its domain. A classical example is to compute the (signed) area of under the curve $A(x)$ of a given function $f(x)$. In neuroscience, the accumulation of input current into the soma of a neuron can also be modelled as an integral of the input current over time.

Consider a point in the input domain x and a small increment by δx to $x + \delta x$. We can approximate the small increase in the area $\delta F(x)$ around the point x under the function f by the trapezium rule:

$$\delta F(x) \approx \frac{1}{2}(f(x) + f(x) + f'(x)\delta(x))\delta x.$$

We can then divide both sides by δx and take the small δx limit to obtain

$$\frac{\mathrm{d}F(x)}{\mathrm{d}x} = f(x) \tag{12}$$

Thus, we see that the instantaneous change of the area under the function $f(x)$ at x is just $f(x)$. The function with uppercase symbol F is called the *indefinite integral* of f, defined as

$$F(x) := \int f(x)\mathrm{d}x. \tag{13}$$

There are in fact infinitely many F's that satisfy $F'(x) = f(x)$; since we can add any constant to F and its derivative is unchanged.

The total area under the function f over some interval $[a, b] \subset \mathbb{R}$ can be computed as the definite integral of f between a and b. The fundamental theorem of calculus, states that the definite integral is

$$\int_a^b f(x)\mathrm{d}x = F(b) - F(a)$$

which equals the signed area under the function f in $[a, b]$

Example 18. The function

$$g(x) = -x + x^2,$$

has the corresponding indefinite integral $G(x)$

$$G(x) = \int g(x)dx = -\frac{x^2}{2} + \frac{x^3}{3} + C,$$

where C is a constant of integration. This constant C exists because the derivative of a constant is 0, so we cannot know what the constant should be. In addition, the signed area under $g(x)$ over the interval $[1, 2]$ is the definite integral

$$\int_1^2 g(x)\mathrm{d} = G(2) - G(1) = 2/3 - (-1/6) = 5/6.$$

3.1.3 Definite integral as accumulation from an initial condition

Note that indefinite integral in (13) is related to the *change* of area, motivated from (12), not the area itself. To see this, we provide an alternative view of the definite integral based on initial conditions.

We write the definite integral of $f(x)$ as

$$F(x) = h(x) + C,$$

where we explicitly separate the part that depends on x and the constant of integration. In the example of Example 18, $h(x) = -x^2/2 + x^3/3$. Based on (12), we know that the change of area as a function of x is $F(x) + C$, but we do not know C. To find C, we use the fact that the area between the interval $[a, a]$ must be zero since it does not have any width. As such, we enforce that the change of area at a be zero, which gives $F(a) = h(a) + C = 0$, giving $C = -h(a)$. Now, the total change of area at b is then $F(b) = h(b) + C = h(b) - h(a)$, which is also equal to $F(b) - F(a)$.

This view of definite integral highlights the importance of *initial condition* when we interpret the integral of a function as the change of area under the curve, or accumulation of function values over an interval. The initial condition defines a baseline on which the accumulation occurs. In general, this initial condition does not have to be zero.

Example 19. Consider a water tank that has initial volume of water v_a at time $t = a$. The inflow of water (volume increase over unit time) from a pipe to this tank is time-dependent and given by $f(t)$. Then, the volume of water v_b at a later time $t = b > a$ can be expressed as

$$v_b = v_a + \int_a^b f(t)\mathrm{d}t = v_a + F(b) - F(a).$$

Alternatively, we can also integrate f first to obtain $F(t) = h(t) + C$, and then set $F(a) = v_a$ according to the initial condition at $t = a$. This implies that $C = v_a - h(a)$. Finally, we have

$$v_b = F(b) = h(b) + v_a - h(a) = v_a + F(b) - F(a),$$

which is the same as the previous expression.

3.2 Multivariate and matrix calculus

For functions with multiple variables, the *partial derivative* is the derivative of the function w.r.t a subset of the variable while holding other variables constant.

Example 20. Define $f(x, y) = x + y + xy$, the partial derivative $\frac{\partial}{\partial x} f(x, y) = 1 + y$

When for functions with multiple inputs and outputs expressed as vector, its derivatives have special names.

Definition 15. The *Jacobian* of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$\nabla_x f(x) := \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}$$

The Jacobian is used in first-order Taylor expansion of multivariate functions $f : \mathbb{R}^n \to \mathbb{R}^m$ around a point $x_0 \in \mathbb{R}^n$

$$f(x) \approx f(x_0) + \nabla_x f(x)|_{x=x_0}(x - x_0).$$

Combined with tools of eigendecomposition, the Jacobian is used in local stability analyses of dynamical systems, after linearising the dynamics around the fixed point.

When $f(x)$ is defined through matrices product, we can apply the tool of matrix calculus. The simplest form is the Jacobian of a matrix-vector product.

Example 21. For the function $f(x) = Ax$ where A is a matrix and x is a vector, we have

$$\frac{\mathrm{d}}{\mathrm{d}x}(Ax) = A$$

This is because

$$\frac{\partial f_i(x)}{\partial x_j} = \frac{\partial}{\partial x_j}\sum_k A_{ik}x_k = A_{ij}$$

As such, one can often work out the matrix derivatives using the definition of matrix multiplications. We give the following results, which can be verified by writing out the products as summations. For more matrix derivative results, see this Wikipedia page or these notes.

Remark 2. *For matrice A, B and vectors u, v, we have*

- $\frac{\mathrm{d}}{\mathrm{d}v}u^\intercal v = u^\intercal$
- $\frac{\mathrm{d}}{\mathrm{d}u}u^\intercal v = \frac{\mathrm{d}}{\mathrm{d}u}v^\intercal u = v^\intercal$
- $\frac{\mathrm{d}}{\mathrm{d}u}\|u\|_2^2 = \frac{\mathrm{d}}{\mathrm{d}u}u^\intercal u = 2u^\intercal$
- $\frac{\mathrm{d}}{\mathrm{d}A}\operatorname{Tr}[A^\intercal B] = B^\intercal$
- $\frac{\mathrm{d}}{\mathrm{d}A}\operatorname{Tr}[A^\intercal A] = 2A^\intercal$

Example 22. Here, we derive the solution to the least square regression with a scalar output (Example 13) using matrix derivatives. We arrange input data $x_i \in \mathbb{R}^m$ as columns of a matrix $X := [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$, output data $y_i \in \mathbb{R}$ as a vector $Y := [y_1, \ldots, y_n]^\intercal$ and define weights $w \in \mathbb{R}^m$.

$$\mathcal{L}(w) := \frac{1}{2}\sum_i (x_i^\intercal \cdot w - y_i)^2 = \frac{1}{2}\|X^\intercal w - Y\|_2^2$$

To work out the optimal w, we differentiate the loss w.r.t. w. Since the chain rule in matrix calculus is not straightforward, we expand the terms in the norm.

$$\begin{aligned}\frac{\partial}{\partial w}\mathcal{L}(w) &= \frac{1}{2}\frac{\partial}{\partial w}\left[w^\intercal XX^\intercal w - w^\intercal XY - Y^\intercal X^\intercal w + Y^\intercal Y\right] \\ &= w^\intercal XX^\intercal - Y^\intercal X^\intercal\end{aligned}$$

Setting this derivative to zero gives the optimal solution $w^* = (XX^\intercal)^{-1}(XY)$.

Exercise 17. Find the solution for a vector-output least-squares regression problem where $y_i \in \mathbb{R}^k$, using properties of the trace Tr operator and its jacobians.

3.3 Differential equations

Differential equations are fundamental mathematical tools for describing how something evolves over time. They can be used to simulate a significant portion of questions in neuroscience. The most famous of these in neuroscience is the Nobel Prize-winning Hodgkin-Huxley model, which describes neural spiking by modelling the dynamics of ion channel states of the squid giant axon. Here, we will start with simpler examples.

The general form of a first-order differential equation is:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(t, x(t)) \tag{14}$$

meaning "the change in a process x over time t is a function f of time t and itself x". This might initially seem like a paradox, as you are using a process x you want to know about to describe itself. But that is the beauty of mathematics—this equation can be solved analytically (giving a closed-form solution for $f(t)$) or simulated numerically given suitable initial conditions.

Example 23. We consider a linear population model. In this model the change of population size $\frac{d}{dt}p(t)$ is a function of population $p(t)$ with birth rate $\alpha = 0.3$, that is

$$\frac{\mathrm{d}}{\mathrm{d}t}p(t) = ap(t),\ t \in \mathbb{R}_+ \tag{15}$$

Exercise 18. How does the population size affect the rate of change of the population?

Next, we compute the exact temporal solution of this population equation, given an initial condition $p(t = 0) = P_0 \geq 0$. The trick to solving this equation is to move terms and integrate both sides as

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}p(t) &= \alpha p(t) \\
\frac{1}{p(t)}\frac{\mathrm{d}}{\mathrm{d}t}p(t) &= \alpha \\
\int_{t=0}^{t'} \frac{1}{p(t)}\frac{\mathrm{d}p(t)}{\mathrm{d}t}\mathrm{d}t &= \int_{t=0}^{t'} \alpha \mathrm{d}t \\
\int_{t=0}^{t'} \frac{1}{p(t)}\mathrm{d}p(t) &= \int_{t=0}^{t'} \alpha \mathrm{d}t \\
\log p(t') - \log p(t=0) &= \alpha t' \\
p(t') &= p(t=0)e^{\alpha t'}
\end{aligned}$$

Note that t' is a variable of a particular future time point. We can substitute in the initial condition $p(t = 0) = P_0$, and redefine t' as t, giving

$$p(t) = P_0 e^{\alpha t},\ t \in \mathbb{R}_+. \tag{16}$$

Once we know the exact solution of the differential equation, we can explore various behaviours in relation to various model parameters (regimes).

Exercise 19. How does the parameter of the population equation affect the outcome?

1. What happens when $\alpha < 0$?
2. What happens when $\alpha > 0$?
3. What happens when $\alpha = 0$?

The population differential equation is an over-simplification and has some very obvious limitations:

1. Population growth is not exponential as there are limited number of resources so the population will level out at some point.

2. It does not include any external factors on the populations like weather, and predators-preys.

These kinds of limitations can be addressed by extending the model.

One type of differential equation that is similar to the population equation is the Leaky Integrate and Fire model, which you will learn later.

3.3.1 The Leaky-Integrate-and-Fire model

The Leaky Integrate-and-Fire (LIF) Model is a linear differential equation that describes the membrane potential (V) of a single neuron. It was proposed by Louis Édouard Lapicque in 1907.

The sub-threshold membrane potential dynamics of a LIF neuron is described by

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - E_L) + R_m I \tag{17}$$

where τ_m is the time constant, V is the membrane potential, E_L is the resting potential, R_m is membrane resistance, and I is the external input current.

Autonomous differential equation In mathematics, an autonomous system or autonomous differential equation is a system of ordinary differential equations which do not depend on the independent variable. Here, we focus on the system without external inputs. The dynamics can be re-expressed as

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - E_L) \tag{18}$$

Exercise 20. Can you identify various behaviours of this differential equation in relation to various V regimes? (Hint: check the derivative of V in the cases $V > E_L$, $V < E_L$ and $V = E_L$.)

Exercise 21. Similar to what we did in Example 19, can you determine the exact time solution of the autonomous LIF model for $t > 0$, with the initial condition $V(0) = V_{reset}$?

The autonomous LIF model has the exact solution:

$$V(t) = E_L + (V_{reset} - E_L)e^{-\frac{t}{\tau_m}} \tag{19}$$

LIF model with external input. We will add back the input I and membrane resistance R_m giving the original equation:

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - E_L) + R_m I \tag{20}$$

The LIF with a constant input has an exact temporal solution:

$$V(t) = E_L + R_m I + (V_{reset} - E_L - R_m I)e^{\frac{-t}{\tau_m}} \tag{21}$$

Exercise 22. Can you compute the Inter-Spike-Interval (ISI) for a LIF model with constant external input? (Hint: from the initial condition $V(t = 0) = V_{reset}$, the neuron integrates external inputs and reaches the firing threshold V_{thresh}, at which point a spike is discharged. Assume that the membrane potential resets to V_{reset}.)

4 Probability

The observable world is inherently noisy and filled with randomness. The theory of probability and random variables are abstractions of hidden factors we do not (or find hard to) observe directly but are crucial for other things we care about. It provides us with tools to describe hidden factors or noise that result in the our observations, and help us describe and control risks when combined with utility functions.

4.1 Random variable

A random variable X realises events in some sample space Ω according to probabilities associated with those events. Logical operations, including the equality $X = x$, inclusion $X \in A \subseteq \Omega$, etc., can be used to define events.

It may be helpful to think of a random variable as an opaque bag of all possible realisations. Every time you want to check its value, you get a *realisation* or a *sample* from this random variable like taking a number from the bag. Keep in mind that although we do not know the probabilities of each event, for a given random variable, these probabilities are implicitly specified and fixed. This is unlike a draw of the random variable which is random. For example, although we do not know the outcome of a random coin flip, the probability of the coin landing on heads when flipped is regarded as a constant

We have the following axioms

1. $\mathbb{P}(X = x) \in [0, 1]$ for all $x \in \Omega$.
2. The whole sample space has probability 1, $\mathbb{P}(\Omega) = 1$.
3. For disjoint events, such as $x \in A$ and $x \in B$ where $A, B \subseteq \Omega$ and $A \cap B = \phi$ is the empty set, we have $\mathbb{P}(x \in (A \cup B)) = \mathbb{P}(x \in A) + \mathbb{P}(x \in B)$

It is useful to know that there are mainly two debated interpretations of probabilities:

- Bayesian: subjective belief of how likely certain event happens;
- Frequentiest: proportion of certain events happen out of an infinite number of draws/simulations of the random variable.

Sometimes we might want to consider the probability of events while restricting our attention to a subset of the sample space. For example, when might want to update our belief about certain events happening after observing other events that already happened. This can be expressed by conditional probabilities.

Definition 16. *Conditional probability* given a non-zero probability event $\mathbb{P}(E_2)$ is defined as

$$\mathbb{P}(E_1|E_2) = \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)} \tag{22}$$

or, for random variables X and Y where $\mathbb{P}(Y = y) \neq 0$

$$\mathbb{P}(X = x|Y = y) = \frac{\mathbb{P}(X = x \cap Y = y)}{\mathbb{P}(Y = y)} \tag{23}$$

A probability without any conditioning is called *marginal probability*.

Definition 17. Two events E_1 and E_2 are said to be independent of each other, denoted by

$$X \perp\!\!\!\perp Y$$

, if and only if

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1)\mathbb{P}(E_2).$$

For random variables X and Y, these two random variables are independent if and only if

$$\mathbb{P}(X = x \cap Y = y) = \mathbb{P}(X = x)\mathbb{P}(Y = y).$$

Exercise 23. Show that if random variables X and Y are independent, then

$$\mathbb{P}(X = x|Y = y) = \mathbb{P}(X = x),$$

assuming $\mathbb{P}(Y = y) > 0$

4.2 Distributions

Distributions are generalised notions of functions, and a probability distribution $p_X : \Omega \to R$ of a real random variable (X) measures the probability or likelihood of each realisation or a set of realisations. The domain of p_X (the sample space Ω) is also known as the support of p_X. When evaluated at a realisation $x \in \Omega$, we have $p_X(x) = \mathbb{P}(X = x)$. When the context is clear, with a slight abuse of notation, we may denote $p(X) = p_X$, and when evaluated on a realisation x, we may also write $p_X(x) = p(X = x)$. The distribution function provides almost everything we want to know about its random variable, and is immensely useful for modelling.

If a random variable X has distribution $p_X(x)$, then we say that X is distributed as $p_X(x)$, denoted by

$$X \sim p_X(x)$$

When the random variable consists of multiple dimensions, such as a random vector or a collection of different variables, we describe them using the *joint distribution*.

Definition 18. The joint distribution of two random variables X and Y is defined as

$$p_{X,Y}(x, y) := \mathbb{P}(X = x \cap Y = y),$$

sometimes also written as $p(X = x, Y = y)$

Likewise, we define the conditional distribution is defined as

Definition 19. The conditional distribution of X given Y is defined as

$$p_{X|Y}(x, y) := p_{X,Y}(x, y)/p_Y(y),$$

also written as $p_{X|Y}(x|y)$ or $p(X = x|Y = y)$.

Given a joint distribution, we can obtain the marginal distribution of some variables by *marginalising out* the other variables:

$$P_X(x) = \sum_y P_{X,Y}(x, y) \quad \text{or} \quad P_X(x) = \int P_{X,Y}(x, y)\mathrm{d}y$$

Random variables associated with certain abstracted events that have been studied by statisticians, such as Gaussian and Bernoulli random variables. They have nice properties that can help us model seemingly stochastic observations in neuroscience, such as neural spiking, decision time, and so on. One great advantage of using these established random variables is that most of them admit known or closed-form distributions, which is very convenient to manipulate with to derive theoretical predictions.

For a full list of commonly used random variables, see here. Throughout this tutorial, we use the following notation to represent a distribution function

$$\text{DistributionSymbol}(\text{variable}; \text{parameters})$$

to denote a known distribution. For example, a Gaussian variable with mean μ and variance σ^2 is $X \sim \mathcal{N}(x; \mu, \sigma^2)$, a categorical variable with probabilities $p \in \Delta_k$ is written $X \sim \text{Cat}(x; p)$

Definition 20. For one-dimensional real random variables, we can define the cumulative distribution function (CDF) as

$$F_X(x) = \int_{-\infty}^{x} p_X(x')\mathrm{d}x'$$

This is a well-defined integral regardless of whether X is continous of discrete, and is thus a very useful representation of the random variable.

4.2.1 Discrete random variables

Discrete RVs have realisations over a finite or countable set (e.g. $\mathbb{Z}$). Examples are Bernoulli, Categorical, Poisson, Binomial, etc. The distribution of a discrete random variable is also called the probability mass function (PMF) of the random variable. It maps each discrete realisations to a probability. Based on the axioms of probabilities, PMFs satisfy

- $p_X(x) \in [0, 1]$,
- $\sum_{x\in\Omega} p_X(x) = 1$
- For distributions supported on $\mathbb{Z}$, by the mutual exclusivity axiom, we additionally have $p_X(a \leq X \leq b) = \sum_{x=a}^{b} p_X(x)$.

4.2.2 Continuous random variables

Continuous random variables are those supported on uncountable sets. Typical examples are Gaussians, exponential and uniform distributions. Their distributions, unlike discrete distributions, do not directly map to probabilities, but *probability densities*, so they are also known as probability density functions. One obtains probabilities from probability densities by integrating over an interval, much like how we obtain mass of objects from their densities by integrating over volumes. The probability density is only non-negative and can be greater than 1, or even infinite.

Example 24. Define X as the continuous uniform random variable supported on $[0, w]$. Its density function is written

$$p_X(x) = \begin{cases} 1/w & x \in [0, w], \\ 0 & \text{otherwise.} \end{cases}$$

The probability $p_X(x \in [0, w/3])$ is calculated as

$$\int_0^{w/3} \frac{1}{w} \mathrm{d}x = \frac{1}{w}\frac{w}{3} = \frac{1}{3}$$

Continuous distributions have interesting properties summarized by the following Remarks. One can obtain the PDF from the CDF by differentiation.

Remark 3. *For continuous random variables X, the PDF p_X and CDF F_X are related by*

$$p_X(x) = \frac{\mathrm{d}}{\mathrm{d}x} F_X(x)$$

Example 25. The exponential random variable with rate λ has PDF is $p_X(x) = \lambda e^{-\lambda x}$ for $x \geq 0$. It's CDF is

$$F_X(x) = \int_0^x p_X(x) = 1 - e^{-\lambda x}.$$

Remark 3 can be easily verifed.

Remark 4. *Given a continuous random variable X in $\mathbb{R}^d$ and an invertible function $g : \mathbb{R}^n \to \mathbb{R}^n$, define the random variable $Y = g(X)$, then we have*

$$p_Y(y) = p_X(x) |\det(\nabla_y g^{-1}(y))|$$

One way to remember this result is to note that, if the random variables are scalars, then $x = g^{-1}(y)$ by using the intuition that $p(x)\mathrm{d}x = p(y)\mathrm{d}y$, leading to $p(y) = p(x)\frac{\mathrm{d}x}{\mathrm{d}y} = p(x)\nabla_y g^{-1}(y)$, The absolute determinant is needed to measure the rate of volumetric change of x in response to a change in y.

Example 26. If X is distributed as a continuous uniform $\mathcal{U}(0, 1)$, and define $Y = -\log(X/\lambda)$, then we can identify $g(x) = -\log(x/\lambda)$, and $g^{-1}(y) = e^{-\lambda y}$. Remark 4 implies that

$$p_Y(y) = p_X(x)| - \lambda e^{\lambda y}| = \lambda e^{-\lambda y}$$

It is clear that we can obtain an exponential distribution by transforming a uniform through $x \mapsto -\log(x/\lambda)$

Remark 5. *Given two independent continuous random variables $X \sim p_X$ and $Y \sim p_Y$, their sum $X := X + Y$ is distributed as*

$$P_Z(z) = \int p_X(x) p_Y(z - x)\mathrm{d}x,$$

the convolution of the two PMFs.

Exercise 24. Derive the the PDF of $Z := X + Y$ where $X \sim \mathcal{U}(0, 1)$ and $Y \sim \mathcal{U}(0, 2)$.

In the following example, we illustrate how to manipulate distributions using the Gaussian mixture model

Example 27. A *mixture distribution* is obtained by linearly weighting several distributions. Each constituent distribution is called a cluster or component. To obtain a sample from a mixture of $k \in \mathbb{N}_+$ components, one first draw a sample Z from a categorical distribution of K outcomes with probabilities specified by $\pi \in \Delta_k$

$$Z \sim \mathrm{Cat}(z; \pi) = \pi_z, \quad z \in \{1, \dots, k\}. \tag{24}$$

where $[z = i]$ is 1 if $z = 1$, 0 otherwise. Z is then an integer representing which one of the k components we are to draw the sample X. Given a realisation z of Z, the distribution from which we draw X is $P_{X|Z=z}$. In case this conditional is a Gaussian distributed as $\mathcal{N}(x; \mu_i, \sigma_i^2)$ for all $z \in \{1, \dots, k\}$, we have the Gaussian mixture model (GMM). The joint distribution of a GMM is written as

$$p_{X,Z}(x, z) = \pi_z \mathcal{N}(x; \mu_z, \sigma_z^2).$$

The marginal over X is

$$p_X(x) = p_{X,Z}(x, z) = \sum_{z=1}^{k} \pi_z \mathcal{N}(x; \mu_z, \sigma_z^2)$$

4.3 Statistics

The distribution provides rich information about the random variable, but often we do not care about too much details. Statistics provide a summary of the distribution and play important roles in modelling and interpretation of the results.

4.3.1 Expectations

Definition 21. The expectation of a function $f : \Omega \to \mathbb{R}$ under a distribution $p_X(x)$ with support Ω is

$$\mathbb{E}_{X \sim p_X}[f(X)] = \begin{cases} \int_\Omega f(x) p_X(x) \mathrm{d}x & \text{if } X \text{ is discrete;} \\ \sum_{x \in \Omega} f(x) p_X(x) & \text{if } X \text{ is continuous,} \end{cases}$$

assuming the integral and sum are finite. In particular, if f is the identity function, then $\mu_X := \mathbb{E}[X] := \mathbb{E}_{p_X}[x]$ is the mean or the expectation of p_X.

Note that the expectation of a random variable may be undefined. A well-known example is the Cauchy distribution.

Definition 22. The following (normalised) central moments are defined using expectations:

- The variance, $\sigma_X^2 := \mathrm{Var}_X := \mathbb{E}_{P_X}[(x - \mu_X)^2]$, the overall spread of the distribution.
- The covariance of two variables X and Y, $\sigma_{XY}^2 := \mathbb{E}_{P_{X,Y}}[(x - \mu_X)(y - \mu_X)]$, the overall spread of the distribution.
- The skewness, $\gamma_X := \mathbb{E}_{P_X}[(x - \mu_X)^3]/\sigma_X^{3/2}$, asymmetry of the distribution; positive skewness indicates long tail towards the right of the mean, and negative skewness towards the left.

- The kurtosis, $\kappa_X := \mathbb{E}_{P_X}[(x - \mu_X)^4]/\sigma_X^4$, how heavy the tails are. $(\kappa_X - 3)$ is someitmes known as excess kurtossis or non-Gaussianity, as Gaussian distribution has kurtosis equal to 3.

Exercise 25. Compute the statistics above for the exponential distribution $\text{Exp}(x; \lambda)$.

The statistics can also be defined for conditional distributions. For example, the conditional expectation of discrete X given Y is written as

$$\mathbb{E}[X|Y = y] = \mathbb{E}_{p_{X|Y=y}}[x] = \sum_x p_{X|Y}(x, y)x$$

Note that, while marginal expectation $\mathbb{E}[X]$ is a constant, the conditional expectation $\mathbb{E}[X|Y]$ is a function of Y. It evaluates to a constant when given a realisation of $Y = y$.

Remark 6. *The following rules concerning moments hold:*

- $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$,
- *Law of total expectation:* $\mathbb{E}_{p_Y}[y] = \mathbb{E}_{P_X}[\mathbb{E}_{P_{Y|X=x}}[y]]$.
- *If X and Y are independent, then*
  - $\sigma^2_{(X+Y)} = \sigma^2_X + \sigma^2_Y$
  - $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$

4.3.2 Quantiles

Definition 23. Let $\tau \in [0, 1]$. The τ-quantile for a scalar distribution is defined as the x' such that $F_X(x') = \tau$, or $x' := F_X^{-1}(\tau)$.

The median is the 0.5-quantiles. The τ-quantile splits the distributions by mass, with τ to its left and $1 - \tau$ to its right. It can provide a risk-sensitive measure. For example, consider a stock with random output whose 0.8'th quantile is \$50. Then, we know that \$50 is a high value event that is not realised with high probability. If we only sell the stock at \$50, the gain would exceed 80% of the stock value, but doing so risks of losing up to \$50 with probability 0.8. Quantile can be used to define value-at-risk, a measure of the risk of loss in decision-making.

4.4 The Bayes rule

The Bayes rule is the cornerstone for updating probabilistic beliefs of hidden variables given observations. It is the direct consequence of the two equivalent factorisations of the joint distribution. In terms of distributions, the Bayes rule states that

$$\underbrace{p_{Y|X}(y|x)}_{posterior} = \frac{\overbrace{p_{X|Y}(x|y)}^{\text{likelihood}}\overbrace{p_Y(y)}^{\text{prior}}}{\underbrace{p_X(x)}_{\text{evidence}}} \tag{25}$$

The best way to understand these four terms is through our the GMM example, Example 27. We redefine the prior in (25) using Y

$$p_Y(y) = \text{Cat}(y; \pi) = \pi_y, \ \ y \in \{1, \dots, , k\} \tag{26}$$

Suppose our goal is to predict which cluster generated a given observation x. The set of all possible y's can be regarded as a hypethesis space. The categorical distribution (24) is the prior distribution over the hypothesis space: it is our belief of which cluster generated any observation without observing x. The likelihood is the Gaussian distribution given the cluster identity y.

$$p_{X|Y}(x|y) = \mathcal{N}(x; \mu_y, \sigma_y^2) \tag{27}$$

Note that this is a likelihood rather than a probability distribution, because x here is fixed at the observed value, and the likelihood is in fact a function of y: it measures how likely each cluster could have generated x. As such, the likelihood term may not be normalised over the domain of y.

The evidence is the marignal likelihood of x, obtained by averaging the likelihood given all clusters over the prior weights. This ensures that the left-hand-side of (25) is normalised. The posterior represents the probabilistic belief of the cluster responsible for generating the observed x.

This process of computing the posterior through the Bayes rule is called Bayesian inference. Bayesian inference requires having a prior and likelihood to represent our belief about the data generating process. The prior and likelihood are collectively known as the *generative model*, because they can be used to simulate or generate new data.

In neuroscience, generative model is a form of the the internal world model hypothesised to be present in the brain. It could help the brain predict certain events by simulation, and can explain perceptual illusions from a computational level.

Exercise 26. Write down the posterior, using terms including *pi* and the parameters of the Gaussian components.

4.4.1 Approximate posterior

While Bayes rule is a powerful tool, computing the evidence may be intractable for a large hypothesis space. As such, simplifications or approximations may be essential to make Bayesian inference practical. One way is to compute the *maximum a posteriori* (MAP) estimate, rather than the full posterior

Definition 24. Given an observation x, the *maximum a posteriori* (MAP) estimate is defined as

$$\arg\max_y \ \{p_{X|Y}(x|y)p_Y(y)\}$$

The MAP does not depend on the denominator in (25), but provides only a point estimate of the variable y given x. This estimate loses all uncertainty present in the posterior. Nonetheless, in certain applications, the MAP is a reasonable approximation and has been used to explain percpetion and cognition.

4.5 Information theory

Information theory provides tools for measuring the amount of uncertainty in a distribution, and how much resource is required to transmit information to reduce uncertainty. It was first developed to advance electronic communication, but then finds numerous applications in compression and signal transimittion and of course in signal processing of the nervous system. The foundamental quantity is the entropy

4.5.1 Entropy

Definition 25. The entropy of a discrete distribution $p_X(x)$ is defined as

$$\mathbb{H}[p_X] = -\mathbb{E}_{p_X}[\log_2 p_X(x)] = -\sum_x p_X(x) \log_2 p_X(x)$$

in the unit of "bits".

The base of the logaritm is not crucial, but helps interpreting the entropy. If natural logarithm $\log_e$ is used, then the unit is "nats".

Exercise 27. Show that the entropy of two independent variables X and Y is the sum of the entropy of each of them.

Example 28.
1. A fair coin has an entropy of 1 bit. It means that the result of a sample requires exactly 1 binary number to specify.
2. If we have 10 independent coints, we need exactly 10 binary numbers.
3. For deterministic events, zero bits is requried to specify the outcome.

It turns out that the entropy provides the theoretical lower bound on the expected number of bits required to transmit a random sample. Note that the entropy is always non-negative and depends only on the probabilities rather than the variable itself.

For continuous variables, the entropy is usually infinite. But we use *differental entropy* to measure uncertainty

Definition 26. The differential entropy of a continuous distribution p_X is defined as

$$h(p_X) = -\mathbb{E}_{p_X}[\log_2 p_X(x)] = -\int p_X(x) \log_2 p_X(x) \mathrm{d}x$$

One can derive the differential entropy from the entroy by taking probabilities within infinitesimal intervls.

Exercise 28. Calculate the entropy of the exponential distribution $\mathrm{Exp}(x; \lambda)$

4.5.2 Mutual information

The mutual information is used to quantify how strongly coupled two random variables are. In other words, how much uncertainty in one random variable is resolved by known the value of the other random variable. For example, if we know that the firing rate of a particular neuron (e.g. in the visual cortex) codes some feature of sensory input (e.g. contrast), then we would expect a strong coupling between these two. In this case, we may want to quantify the coupling by mutual information.

Definition 27. The mutual information between two random variables X and Y is defined as

$$I(X,Y) := \mathbb{E}_{p_{X,Y}}\left[\log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\right]$$

Because of the ratio, the mutual information is unitless and is thus interpretable regardless of whether the variables are discrete or continuous.

Example 29. Let X be the outcome of a fair coin toss modelled as a Bernoulli distribution $\text{Bernoulli}(x; 0.5)$. Suppose that there is a copycat coin that always produces the same outcome as Y. Then the joint distribution is $p_{X,Y}(x,y) = 0.5$ when $x = y = 1$ or $x = y = 0$, and zero otherwise. It is easy to verify that the mutual information is 1.

Exercise 29. Show that the mutual information between two independent random variables is zero.

5 Acknowledgements

These notes were first prepared for the 11th Computational & Cognitive Neuroscience Summer School (CCNSS), held in Cold Spring Harbor Asia (CSHA) in July, 2023.

We thank Kun Tian for guiding us to relevant tutorials in the preparation of these notes. Heartfelt thanks go to the organisers of CCNSS (especially Prof Songting Li and Prof Guangyu Robert Yang) for extending the invitation to us, and to the dedicated staff at CSHA for their exceptional care and support throughout. Our appreciation goes to the 30 talented and enthusiastic students who provided us with the invaluable opportunity to revisit, reconsider, and impart the fundamental principles of computational neuroscience. The three weeks we spent together were truly remarkable and unforgettable, fostering strong connections and scientific exploration. We are looking forward to future meetings and collaborations. Special thanks to Xian Li for providing useful feedback on these notes.

LKW thanks its employer, Google DeepMind, for granting permission to participate in the summer school and fulfilling his long-cherished wish to be part of CCNSS.