1 Lecture 1

1.1 Motivation

Suppose you want to quantitatively describe the resting potential of a specific neuron, neuron X, in *C. elegans*. To do this, you could:

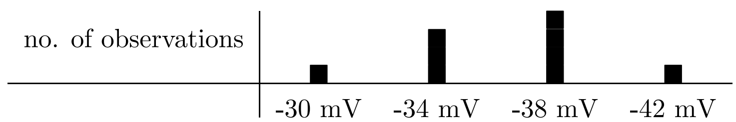
1. Gather 10,000 worms.
2. Measure the resting potential of neuron X in each worm. This produces a dataset of 10,000 continuous-valued measurements:

resting potential
-44.3 mV
-40.1 mV
-34.9 mV
⋮

3. Compute summary statistics from this dataset:

mean	-39.07 mV
standard deviation	3.76 mV

4. Compute more detailed statistics, such as a histogram of the data:



Great! Now suppose you want to describe the firing rate dynamics of a population of 5,000 motor neurons in the macaque brain during a motor movement. For every motor movement in every macaque, you record a dataset like:

	rate 1	rate 2	rate 3	⋯	rate 5,000
time = 0 s	3 Hz	4 Hz	12 Hz	⋯	25 Hz
time = 1 s	12 Hz	1 Hz	7 Hz	⋯	15 Hz
time = 2 s	10 Hz	20 Hz	2 Hz	⋯	1 Hz
⋮	⋮	⋮	⋮	⋮	⋮

How should you analyze this data?

This is a difficult problem! You could certainly apply the techniques you used to describe neuron X in *C. elegans* to describe each neuron in the macaque dataset individually. But, this would not give you any information about how the neurons vary together. In order to do that, we need mathematical techniques for *describing the relationships between high-dimensional measurements.*

Linear algebra provides a powerful framework for analyzing high-dimensional data.

In this course, we will use linear algebra to:

- Discover hidden structure in high-dimensional data.
- Quantitatively describe relationships between multiple high-dimensional datasets.
- Enable statistical analysis of these structures and relationships.

1.2 Vectors

Linear algebra lets us handle observations which consist of *many* numerical measurements. For example, when measuring motor neuron activity (above), a single observation would include 5,000 measurements. To reflect the fact that these measurements are all part of one observation, we place them in a mathematical object called a *vector*. For example, the first observation in the table above would be:

$$\vec{v} = \begin{pmatrix} 3 \\ 4 \\ 12 \\ \vdots \\ 25 \end{pmatrix} \text{Hz}$$

The arrow over $\vec{v}$ indicates that it is a vector. In this case it is 5,000-dimensional, since it contains 5,000 elements. Each element is just a number, and we often use subscript indices to refer to specific elements:

$$v_1 = 3\,\text{Hz}, \qquad v_2 = 4\,\text{Hz}, \qquad v_3 = 12\,\text{Hz}, \qquad \ldots, \qquad v_{5000} = 25\,\text{Hz}$$

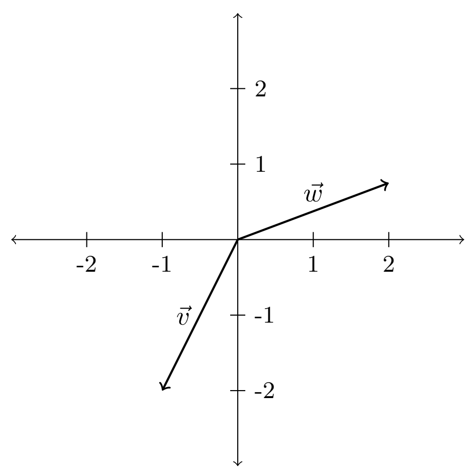
so that

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{5000} \end{pmatrix}$$

Vectors can have any number of dimensions. In two or three dimensions it is often helpful to imagine vectors as arrows from the origin to a specific point in space. So, in 2D, each vector is an arrow in the plane:

2D geometric picture

$$\vec{v} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \qquad \vec{w} = \begin{pmatrix} 2 \\ 0.75 \end{pmatrix}$$



In 3D, each vector is an arrow from the origin to a point in 3D space. (Use your imagination.) This 'geometric picture' of vectors is useful since it allows us to understand many operations on vectors in terms of our spatial intuitions. Often, operations on high-dimensional vectors (e.g., 5,000-D) can be better understood by considering a similar operation in 2D or 3D, and then imagining the geometric picture of the vectors involved.

1.3 Vector operations

The magic of linear algebra will come from studying the properties of $\vec{v}$ as an object in its own right, hiding all the specific elements v_j. To do this we need to define some algebraic operations on vectors. You could come up with many, many possible operations to define! Linear algebra uses one such collection of operations that turn out to be extremely useful.

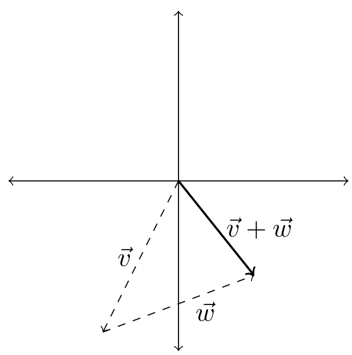
1.3.1 Vector addition

We can add two vectors with the same dimension n by adding each pair of elements:

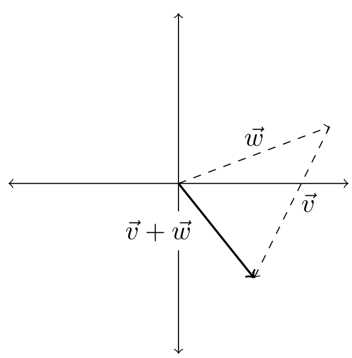
$$\vec{v}+\vec{w}=\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}+\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}=\begin{pmatrix} v_1+w_1 \\ v_2+w_2 \\ \vdots \\ v_n+w_n \end{pmatrix}$$

In the geometric picture, this corresponds to stacking the arrows of $\vec{v}$ and $\vec{w}$ tip-to-tail:

$$\vec{v}=\begin{pmatrix} -1 \\ -2 \end{pmatrix}, \qquad \vec{w}=\begin{pmatrix} 2 \\ 0.75 \end{pmatrix}, \qquad \vec{v}+\vec{w}=\begin{pmatrix} 1 \\ -1.25 \end{pmatrix}$$



The order of the stacking doesn't matter, reflecting that $\vec{v}+\vec{w}=\vec{w}+\vec{v}$:



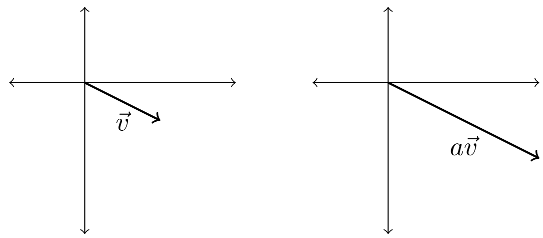
1.3.2 Scalar multiplication

We can scale a vector by a number a (for example, $a=0.5$) by multiplying each of its elements by a:

$$a\vec{v}=a\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}=\begin{pmatrix} av_1 \\ av_2 \\ \vdots \\ av_n \end{pmatrix}$$

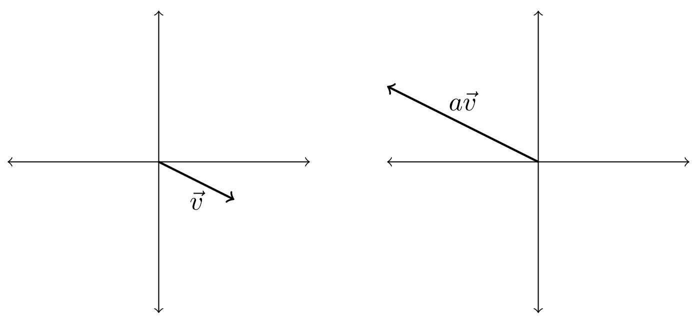
In the geometric picture, this corresponds to stretching a vector's arrow by a factor of a:

$$\vec{v} = \begin{pmatrix} 1 \\ -0.5 \end{pmatrix}, \qquad a = 2.0$$



If $a < 0$, then the direction of the arrow is flipped:

$$\vec{v} = \begin{pmatrix} 1 \\ -0.5 \end{pmatrix}, \qquad a = -2.0$$



As with normal algebra, we have that $a(\vec{v} + \vec{w}) = a\vec{v} + a\vec{w}$.

Exercise: with

$$\vec{v} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}, \qquad \vec{w} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

compute $\frac{1}{2}\vec{v} - \vec{w}$, and show each step geometrically.

1.3.3 Linear combinations

With scalar multiplication and vector addition in hand, we can talk about *linear combinations* of vectors. A linear combination of $\vec{v}$ and $\vec{w}$ is a vector $\vec{u}$ where

$$\vec{u} = a\vec{v} + b\vec{u}$$

for any scalar coefficients a and b. Linear combinations are important since they allow us to forget about coordinates, and instead write vectors as linear combinations of other vectors. For example, with

$$\vec{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \vec{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and

$$\vec{v} = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}, \qquad \vec{w} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

we can write

$$\vec{v} = \frac{1}{2}\vec{x}, \qquad \vec{w} = \vec{x} + \vec{y}$$

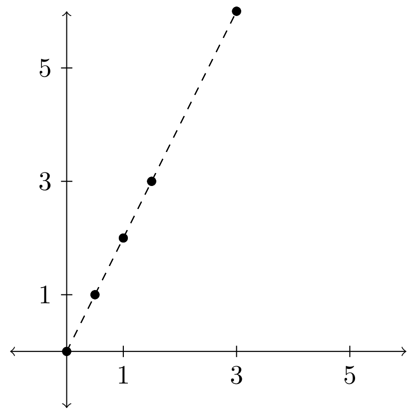
which is sometimes more convenient.

Exercise: express $\vec{x}$ and $\vec{y}$ as linear combinations of $\vec{v}$ and $\vec{w}$.

For example, consider the following dataset of 2D measurements:

	x	y
obs 1	0	0
obs 2	3	6
obs 3	1	2
obs 4	1.5	3
obs 5	0.5	1

If we plot this data it is easy to see that all measurements fall on a line:



so if we define the vector

$$\vec{z} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

then we can write

$$\begin{aligned} \text{obs 1} &= 0\,\vec{z} \\ \text{obs 2} &= 3\,\vec{z} \\ \text{obs 3} &= 1\,\vec{z} \\ \text{obs 4} &= 1.5\,\vec{z} \\ \text{obs 5} &= 0.5\,\vec{z} \end{aligned}$$

and 'forget' about the original measurements.

(This is dimensionality reduction! We will learn more about it later. What would happen if the measurements didn't fall exactly on the line?)

1.3.4 Dot products

It is useful to be able to quantify how 'aligned' two vectors are. This can be done with the *dot product* operation:

$$\vec{v} \cdot \vec{w} = v_1 w_1 + v_2 w_2 + v_3 w_3 + \cdots + v_n w_n = \sum_{j=1}^{n} v_j w_j$$

The dot product between a vector and itself, $\vec{v} \cdot \vec{v}$ is the 'squared norm' of $\vec{v}$, denoted

$$||\vec{v}||^2 = \vec{v} \cdot \vec{v}$$

where $||\vec{v}||$ is the *length* of the vector arrow in the geometric picture. It turns out that

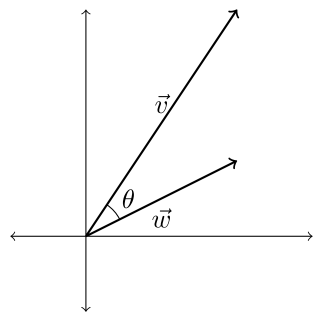
$$\vec{v} \cdot \vec{w} = ||\vec{v}|| \cdot ||\vec{w}|| \cdot \cos\theta$$

where θ is the angle (radians) between $\vec{v}$ and $\vec{w}$. This gives some intuition about the value of the dot product:

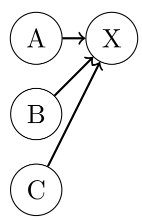
- $\vec{v} \cdot \vec{w}$ scales with the lengths of $\vec{v}$ and $\vec{w}$
- $\vec{v} \cdot \vec{w}$ is large (in magnitude) if $\vec{v}$ and $\vec{w}$ are nearly aligned, and small if they are nearly orthogonal
- $\vec{v} \cdot \vec{w}$ is negative if $\vec{v}$ and $\vec{w}$ point opposite directions

$$\vec{v} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \qquad \vec{w} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\vec{v} \cdot \vec{w} = 2 \cdot 2 + 3 \cdot 1 = 7$$



Dot products come up all the time. For example, they can be used to model a neuron's response to its inputs. Suppose we have a neuron X with 3 upstream synaptic partners A, B, C, and we are recording the firing rates of all of them:



We might obtain the following measurements of baseline-relative firing rates (units are 10 Hz):

A	B	C	X
1	0	0	1
1	1	2	5.5
2	2	1	5
-2	2	1	1

From this data, we can construct a linear model of X's firing rate given A, B, and C's. With

$$\vec{w} = \begin{pmatrix} 1 \\ 0.5 \\ 2 \end{pmatrix}$$

our model is

$$X = \vec{w} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

Exercise: show that our model is consistent with the observed data. What is the model's prediction of X's firing rate given $A = -1$, $B = 2$, and $C = -0.5$?

(This is linear regression! We will learn more about it later.)

Another way to write this is

$$X = \begin{pmatrix} 1 & 0.5 & 2 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

where

$$\vec{w}^{\mathrm{T}} = \begin{pmatrix} 1 & 0.5 & 2 \end{pmatrix}$$

is called a 'row' vector. I.e., $\vec{w}^{\mathrm{T}}\vec{v}$ is another way to write $\vec{w} \cdot \vec{v}$.

1.3.5 Matrices

Building on the previous example, suppose we were dealing with *two* downstream neurons, X and Y. We record from Y as well, filling out another column in our data table:

A	B	C	X	Y
1	0	0	1	0
1	1	2	5.5	-1
2	2	1	5	1
-2	2	1	1	1

Now, we use a dot product to model Y just as we did with X:

$$\text{(same as before)} \qquad X = \begin{pmatrix} 1 & 0.5 & 2 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

A more convenient way to write this model is to use *matrix multiplication*:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 1 & 0.5 & 2 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

where the object

$$M = \begin{pmatrix} 1 & 0.5 & 2 \\ 0 & 1 & -1 \end{pmatrix}$$

is a 2×3 matrix. For $\vec{v}$ a 3D vector, $M\vec{v}$ is a 2D vector.

> Exercise: what is the model prediction of the vector (X, Y) for $(A, B, C) = (5, 4, 3)$?

Matrices represent *linear transformations* on vectors: they are functions that map vectors to other vectors. Generally, a matrix M has an input dimension r and an output dimension c. Then M has r rows and c columns. The vector $\vec{w} = M\vec{v}$ is r-dimensional for c-dimensional $\vec{v}$. The matrix M has rc elements, indexed M_{jk} for j taking values from 1 to r and k taking values from 1 to c:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1c} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2c} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{r1} & A_{r2} & A_{r3} & \cdots & A_{rc} \end{pmatrix}$$

The entry A_{jk} describes the linear dependence of $(A\vec{v})_j$ on v_k. Accordingly, the matrix multiplication rule is that for $\vec{w} = A\vec{v}$,

$$w_j = \sum_{k=1}^{m} A_{jk} v_k$$

or, written out fully,

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1m} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2m} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nm} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} A_{11}v_1 + A_{12}v_2 + A_{13}v_3 + \cdots + A_{1m}v_m \\ A_{21}v_1 + A_{22}v_2 + A_{23}v_3 + \cdots + A_{2m}v_m \\ A_{31}v_1 + A_{32}v_2 + A_{33}v_3 + \cdots + A_{3m}v_m \\ \vdots \\ A_{n1}v_1 + A_{n2}v_2 + A_{n3}v_3 + \cdots + A_{nm}v_m \end{pmatrix}$$

One way to think about matrix-vector multiplication $M\vec{v}$ is as a stack of dot products: just compute the dot product between $\vec{v}$ and each row of M, and place all those dot products in another vector. This is how I like to remember the multiplication rule.

Exercise: evaluate

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ -3 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Matrices are useful because they are mathematically and computationally quite easy to deal with:

- For a given matrix, we can easily describe its properties.
- For a given set of desired properties, we can often easily find a matrix that satisfies them. (Or say that no such matrix exists.)

Here, we have just used them to directly model an input/output relation (the firing rates of X, Y given A, B, C). But matrices are generally very useful in data analysis as a tool for transforming data to make it easier to work with or to extract information from it. Many of the techniques we talk about later in the class will be formulated in terms of matrices and matrix operations.

1.3.6 Matrix composition

Linear transformations can be composed to create new linear transformations. Let A be an $n \times m$ matrix, and let B be an $m \times p$ matrix. Then for any p-dimensional vector $\vec{v}$, we can apply B and then A to get an n-dimensional vector $\vec{w}$:

$$\vec{w} = A(B\vec{v}) = (AB)\vec{v}$$

where AB is an $n \times p$ matrix. For example, take

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

which rotates vectors 90° counterclockwise. Then, take

$$B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

which rotates vectors 90° clockwise. Without doing any computations, we know that we must have

$$\vec{v} = AB\vec{v}$$

for *any* 2D vector $\vec{v}$. Therefore, we have

$$AB = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \equiv I$$

where I denotes the special *identity matrix*, which maps vectors onto themselves. Warning: generally $AB \neq BA$, i.e. matrix multiplication is not commutative! For non-square matrices the equation wouldn't even make sense; for square matrices, it is generally false.

Imagine developing an analysis pipeline in multiple stages. Your raw data comes as vectors of dimension 1000.

1. You first design a smoothing matrix S of size 1000×1000, which just smooths the data a little bit but does not change the dimension.
2. Then you design a 99×1000 B subsampling matrix, which reduces the dimensionality to 99.
3. Finally, you apply a 2×99 classification matrix C.

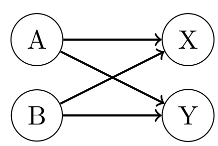
Your entire pipeline could be implemented by the matrix $P = CBA$, which has shape 2×1000:

$$\begin{aligned}&(2 \times 99)(99 \times 1000)(1000 \times 1000)\\ &\longrightarrow (2 \times 99)(99 \times 1000)\\ &\longrightarrow (2 \times 1000)\end{aligned}$$

1.3.7 Matrix inverses

We just saw a pair of matrices A and B where $AB = I$. These matrices are *inverses*. Many square matrices A have a unique inverse A^{-1} so that $AA^{-1} = A^{-1}A = I$. In intuitive terms this means that (usually) as long as a matrix preserves the dimension of a vector (i.e., the matrix is square), then applying it doesn't throw out any information since we can apply the inverse matrix to get out what we put in.

For example, consider the following neural circuit:



we might model it with a matrix equation like

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 0.5 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix}$$

but we could also *invert* this matrix to get the equivalent model:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} -1/7 & 4/7 \\ 4/7 & -2/7 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

1.4 Summary

The takeaways from this lesson are:

1. Linear algebra enables analyzing high-dimensional data.
2. A vector (typically) encodes one multi-dimensional observation.
3. Vectors can be added and scaled.
4. Matrices (linear transformations) can be used to represent relationships between vectors.
5. Composed matrix transformations can be written as a single matrix.
6. Many square matrices have inverses.