Mathematics for Machine Learning 1

Floor Eijkelboom*, Tin Hadži Veljković*
{eijkelboomfloor, tin.hadzi}@gmail.com

Contents

[*] These authors contributed equally to this work.

1 Introduction

Hi there! Welcome to the Machine Learning 1 (ML1) course here at MSc Artificial Intelligence from the University of Amsterdam. These notes are provided to help you familiarize yourself with the 'prior knowledge' needed to follow the ML1 course. In practice, however, you will probably not know all the material here, and please do not feel discouraged by that. We advise you to read through these notes and see which things you already knew and which things you need to brush up on.

The document is divided into three parts: linear algebra, multivariate calculus, and a general introduction to machine learning. The first section aims to refresh your knowledge about vectors, matrices, linear transformations, determinants, bases, orthonormal projections, eigen decompositions, and similar topics. The second section is focused on calculus in higher dimensions, essentially generalizing the derivative from standard (real) functions to real functions between higher-dimensional Euclidean spaces. The last section will zoom into the machine learning problem, the actual reason you are probably reading these notes, to begin with. It is a concise sketch of the general problem you will be facing for the next weeks.

If you see any errors in this document, please write us and we will make sure to address them as soon as possible. Moreover, feel free to share this document with whoever might profit from it. Good luck with your studies!

Floor & Tin

2 Linear Algebra

2.1 Basics

Linear algebra serves as a core of most machine learning algorithms that you will encounter throughout the course, as the majority of objects are represented as vectors and matrices (matrices are called arrays/tensors in `NumPy`/`PyTorch`). For this reason, we will systematically revise all the essential concepts such as vectors, matrices, linear operators, and determinants. To intuitively explain certain concepts, we will use jargon which will be denoted in *italic* font.

2.2 Vector spaces

In order to introduce vector spaces, which is a space where vectors *live*, we will first try to motivate its formal definition which will follow later.

Informal definition

Firstly, let's denote a vector by a bold letter $\mathbf{v}$. The easiest way to visualize a vector is to associate it with something familiar. For example, imagine you live on a flat Earth and you're on a hike and you wish to send your friends your location. You could, for example, represent your location as a 3D vector:

$$\mathbf{v} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

In this notation, x_1 and y_1 are your initial longitude/latitude offset from the bottom of the mountain (the amount you moved west/east and south/north), while z_1 might represent your altitude. You continue your hike, change your longitude/latitude by x_2 and y_2, and climb up by z_2 to reach the peak. Then, your new coordinates $\mathbf{v}$ are:

$$\mathbf{v}' = \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \end{bmatrix}$$

In other words, your new coordinates are simply a sum of the two offsets. Notice that the sum of two independent offsets produced a new location $\mathbf{v}'$ which also represents a valid location.

Now, imagine you're going on the same hike, but this time the mountain grew in size by a factor of λ, and you wish to come to the same peak as last time. Intuitively, we can deduce that the you will have to move further by a factor of λ in each direction, so the total offset $\mathbf{w}$ will be given by:

$$\mathbf{w} = \begin{bmatrix} \lambda x_1 + \lambda x_2 \\ \lambda y_1 + \lambda y_2 \\ \lambda z_1 + \lambda z_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \end{bmatrix} = \lambda \mathbf{v}'$$

This tells us that even if we multiply our offsets by a number λ, we can still represent a valid location.

This was a very specific example to aid the visualization of certain properties that define a vector space, which we will soon define. If we think of a vector as an abstract object which doesn't correspond to anything visualizable, then the above-mentioned properties can be thought as the following. First, we want the sum of two vectors to also be vector from the same space. Second, if we scale a given vector, we wish that the scaled version is also a part of the same vector space.

Formal definition

We shall now introduce a formal definition of a vector space.

Definition 2.1. A vector space over a field $\mathbb{F}$ is a set V with two binary operations:

1. Vector addition assigns to any two vectors $\mathbf{v}$ and $\mathbf{w}$ in V a third vector in V which is denoted by $\mathbf{v} + \mathbf{w}$.
2. Scalar multiplication assigns to any scalar λ in $\mathbb{F}$ and any vector $\mathbf{v}$ in V a new vector in V, which is denoted by $\lambda\mathbf{v}$.

Vector spaces also have to satisfy 8 axioms, which can be found here (most of them are trivial and intuitive).

In the definition above, a field $\mathbb{F}$ is simply a structure from which we take scalars that we multiply our vectors by. In most cases, the field will simply be real numbers $\mathbb{R}$.

If we come back to the hiking example, we were dealing with the vectors from $\mathbb{R}^3$, as we had 3 entries of the vector, and each entry was a real number (coordinates are real numbers).

Summary

Vectors are objects that live in a vector space. It is important to note that a vector space is a space defined by only two operations with objects: how to add objects and how to scale them. If we know how to do that, we call that space a vector space. In further sections, we will explore other ways to utilize and transform vectors besides the addition of vectors and multiplication by a scalar.

2.3 Basis

Similar to the previous section, we will first informally motivate the definition of a basis, and only then formalize it.

Informal definition

The basis of a vector space provides an organized way to represent any vector in that space. As a simple example, let's think about possible colors produced by a pixel on the screen you are reading this on. Every pixel consists of 3 lighting elements: red, green, and blue, and every other color can be reproduced by varying the intensities of each of these colors. Since the lighting elements are independent, we can represent an arbitrary color $\mathbf{c}$ as follows:

$$\mathbf{c} = \begin{bmatrix} r_i \\ g_i \\ b_i \end{bmatrix}$$

where $r_i/g_i/b_i$ denote the intensities of the red/green/blue light. By tuning these three numbers, we can represent any color reproducible by our monitor. Now, let's rewrite this more suggestively:

$$\mathbf{c} = \begin{bmatrix} r_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ g_i \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_i \end{bmatrix} = r_i \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + g_i \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + b_i \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

We can now also give unique names to the column vectors and write the previous expression as follows:

$$\mathbf{c} = r_i \cdot \mathbf{r} + g_i \cdot \mathbf{g} + b_i \cdot \mathbf{b},$$

where:

$$\mathbf{r} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The color vector **c** has been written as a weighted sum of other vectors, and this is called a linear combination. Since we can uniquely represent any vector (color) using these three vectors, we say that vectors **r**, **g** and **b** form a basis. A basis can be thought of as a set of independent vectors whose linear combination can uniquely represent any vector. The basis of this form, where the n-th basis vector has 1 as the n-th element and 0 otherwise, is called a canonical basis and is the most simple form of basis.

It is worth investigating why we impose the condition that the basis vectors need to be independent, and what independence means. For simplicity, imagine that a purple color **p** can be expressed as a linear combination of red and blue:

$$\mathbf{p} = \frac{1}{\sqrt{2}}\mathbf{r} + \frac{1}{\sqrt{2}}\mathbf{b} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Now, let's imagine that we add the color purple to our basis, so our basis now consists of $\{\mathbf{r}, \mathbf{g}, \mathbf{b}, \mathbf{p}\}$ (you have 4 lights in your pixel now). Your friend told you about an imaginary color durple **d**, and they told you that they use the $\{\mathbf{r}, \mathbf{g}, \mathbf{b}\}$ basis for their pixels. They represent the color durple as follows:

$$\mathbf{d} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

In order to reproduce this color, you start turning the 4 knobs of color intensities (one for each different color in your pixel). First, you do not use your purple color, and you just stick with red and blue. You find that the following combination reproduces durple:

$$\mathbf{d} = -\frac{1}{\sqrt{2}}\mathbf{r} + \frac{1}{\sqrt{2}}\mathbf{b} = -\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

However, you start turning the purple knob, tune red and blue a bit, and you realize that also the following combination produces durple:

$$\mathbf{d} = -2 \cdot \mathbf{r} + \mathbf{p} = -\frac{1}{\sqrt{2}} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

This tells us that after adding the purple color to our basis, our representation of the durple color was no longer unique, i.e. there were multiple ways to produce it. This stems from the fact that we have added purple to our basis, which was not independent since we were able to write it as a linear combination of already existing colors (red and blue). An equally valid choice of basis would have been to remove the color red from our basis set, and simply have $\{\mathbf{g}, \mathbf{b}, \mathbf{p}\}$ as our basis.

An intuitive way to think of a basis is as a set of vectors, of which none can be written as a linear combination of the rest. Formally it can be shown that the number of basis vectors has to equal the dimension of the vector space. For example, in the example above, the number of basis vectors was 3, as we had 3-dimensional vectors. If we were to add any more vectors to our basis, we would necessarily add vectors that are no longer independent of each other, and thus we wouldn't have a systematic and unique way to represent arbitrary vectors. If we were to remove any vectors (for example, have 2 vectors in our basis), we wouldn't be able to express an arbitrary vector as a linear combination of the basis vectors, as we would be missing *building blocks*.

Formal definition

In Linear Algebra, the basis of a vector space V is formally defined as follows.

Definition 2.2. A basis B of a vector space V over a field $\mathbb{F}$ is a linearly independent subset of V that *spans* V. This subset, therefore, has to satisfy the following conditions:

1. **Linear independence**: For every finite subset $\{\mathbf{v_1}, \ldots, \mathbf{v}_m\}$ of B, neither of the m elements can be represented as a linear combination of the rest.
2. **Spanning property**: For every vector $\mathbf{v}$ in V, one can choose scalars $\lambda_1, \ldots, \lambda_n$ from the field $\mathbb{F}$ and $\mathbf{v_1}, \ldots, \mathbf{v_n}$ such that $\mathbf{v} = \lambda_1 \mathbf{v_1} + \ldots + \lambda_n \mathbf{v_n}$.

The first condition states what we discussed above; if we wish to have a basis, we mustn't be able to represent any of the basis elements by a linear combination of other basis elements. This is required if we wish to uniquely represent every vector using basis vectors.
The second condition tells us that we must be able to represent **any** vector from the vector space using a linear combination of the basis vectors. These two conditions combined lead to the fact that for a basis of a n-dimensional vector space we must have exactly n linearly independent basis vectors.

2.4 Dot product

So far, we have only seen two operations we can do with vectors: addition of vectors and multiplication of vectors by a scalar. The two operations combined allowed us to form a definition of a linear combination and basis.

A vanilla vector space does not have any other operations that involve two vectors. However, a vector space can be equipped with an inner product to form an inner product space.[1] A dot product [2] between vectors $\mathbf{v} = [a_1, \ldots, a_n]$ T[3] and $\mathbf{w} = [b_1, \ldots, b_n]$ T is defined as follows:

$$\mathbf{v} \cdot \mathbf{w} = a_1 b_1 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i$$

To see the benefit and the interpretation of the dot product, let's take a closer look at a case when we calculate a dot product of a vector with itself:

$$\mathbf{v} \cdot \mathbf{v} = \sum_{i=1}^{n} a_i a_i = \sum_{i=1}^{n} a_i^2$$

[1] An interested reader can find more information here.

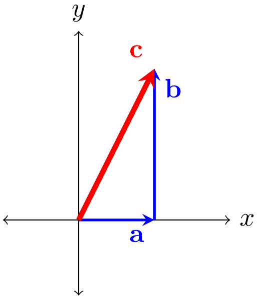
[2] Inner product and a dot product are often used interchangeably, although there are subtle differences, refer here for a brief discussion.

[3] Letter T stands for the transpose operation, more information can be found here.

What we can see from this is that this corresponds to the squared norm/magnitude of the vector $\mathbf{v}$. The usual notation for the norm of a vector is $\|\cdot\|$, so we can write:

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} a_i^2} \implies \|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

As a simple example, let's imagine that we have a 2D vector $\mathbf{c} = \mathbf{a} + \mathbf{b}$, where $\mathbf{a} = [a, 0]^{\mathrm{T}}$ and $\mathbf{a} = [0, b]^{\mathrm{T}}$, as shown in the figure below:



If we calculate the dot product of the vector $\mathbf{c}$ with itself, we get:

$$\|\mathbf{c}\|^2 = a^2 + b^2,$$

which is exactly the Pythagorean theorem in 2D.

Besides being useful for calculating norms of vectors, dot product can be used as a measure of similarity. If we imagine two n-dimensional vectors $\mathbf{v}$ and $\mathbf{w}$, the angle θ between them can be calculated using the following formula:

$$\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\| \, \|\mathbf{w}\| \, \cos\theta$$

We can divide both sides by the norms of both vectors to get the expression for the cosine of the angle between the vectors:

$$\cos\theta = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \, \|\mathbf{w}\|}$$

When the cosine of the angle between two vectors is equal to 1, the vectors are perfectly aligned (interpreted as being as similar as possible), and when it is equal to 0, the vectors are perpendicular (interpreted as being as different as possible). This can be interpreted as a measure of similarity (often called the *cosine similarity*), which is often used in many areas, such as Natural Language Processing (more information with some examples can be found here).

To sum up, we have introduced a new operation we can use to manipulate vectors, the dot product. It is a useful tool because it allows us to easily calculate the norms of vectors, and also the cosine similarity between them.

2.5 Linear Operators

Mappings

In linear algebra, besides the operations that involve two vectors (vector addition, dot product), there are functions (mappings) that take as an input a vector, and output a vector. Let's denote this mapping as f. Formally, any mapping of this sort can be written as:

$$f : V \to W$$

This is a standard mathematical notation which means the following: a function f takes as an input a vector from the vector space V and outputs a vector from a vector space W. Now, you might wonder why there are two vector spaces involved, and this will become more clear after a few examples.

Let's consider the following two mappings f and g:

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \\ y \\ x+y \end{bmatrix}, \quad g\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \\ y \\ xy \end{bmatrix}$$

This is a generalization of functions that we are used to; here our inputs are vectors, and so are the outputs. If x and y are real numbers, then formally we can write this mapping as $f : \mathbb{R}^2 \to \mathbb{R}^3$, since the input to our mapping is a 2D vector, and the output is a 3D vector (therefore they *live* in different vector spaces). You will work more with these types of functions in the Multivariate Calculus section.

Now, which mappings can be called *linear* mappings? The conditions are intuitive, and quite similar to the ones of the vector spaces, so we shall provide now a formal definition.

Definition 2.3. Let V and W be vector spaces over the same field $\mathbb{F}$. A function $f : V \to W$ is said to be a *linear map* if for any two vectors $\mathbf{v}, \mathbf{w}$ from V and any scalar λ from $\mathbb{F}$, the following two conditions are satisfied:

1. **Additivity**: $f(\mathbf{v} + \mathbf{w}) = f(\mathbf{v}) + f(\mathbf{w})$
2. **Homogeneity**: $f(\lambda\mathbf{v}) = \lambda f(\mathbf{v})$

The first condition states that the transformation of the sum of vectors has to be equal to the sum of transformations of every vector individually.
The second condition simply states that it shouldn't matter whether we first multiply the vector $\mathbf{v}$ by a scalar λ and then transform it, or we first transform the vector $\mathbf{v}$ and then multiply it by λ.

Now, are the mappings f and g above linear or not? The way to check it is by testing whether they satisfy the additivity and homogeneity conditions, which we leave as an exercise[4].

Matrix-vector multiplication

If you've encountered linear algebra before, then you probably associate linear mappings/transformations/operators with matrices. Let's first discuss how and why matrix-vector multiplication works, and then we will connect it to the concept of linear mappings discussed in the previous subsection.

To start, let's imagine we have a very simple canonical basis $B = \{\mathbf{b_1}, \mathbf{b_2}\}$ in $\mathbb{R}^2$, where the basis vectors are:

$$\mathbf{b_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{b_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

[4] You should find out that f is indeed linear, while g isn't.

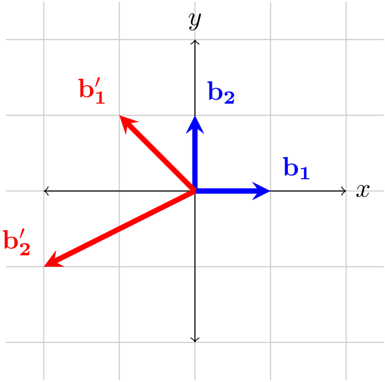
The matrix representation of a linear transformation is *defined* to have the following form: n-th column of the matrix corresponds to a vector to which the n-th canonical basis vector transforms. For example, let's observe the following matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} -1 & -2 \\ 1 & -1 \end{bmatrix},$$

This means that the matrix A will transform the vectors $\mathbf{b_1}$ and $\mathbf{b_2}$ into $\mathbf{b_1'}$ and $\mathbf{b_2'}$ in the following way:

$$\mathbf{b_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \mathbf{b_1'} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbf{b_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \mathbf{b_2'} = \begin{bmatrix} -2 \\ -1 \end{bmatrix},$$

which is visualized in the figure below.



Now that we know how a matrix transformation transforms our basis vectors, let's see how this applies to an arbitrary vector. Let's consider a general matrix $\mathbf{A}$ and a vector $\mathbf{v}$ which have the following form:

$$\mathbf{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

The vector produced by the matrix-vector multiplication shall be denoted as $\mathbf{w}$. Let's try to calculate it using the rules of vector spaces and linear operators that we have learned so far:

$$\begin{aligned}
\mathbf{w} &= \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\
&\overset{1}{=} \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \left(\begin{bmatrix} v_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ v_2 \end{bmatrix} \right) \\
&\overset{2}{=} \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \left(v_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\
&\overset{3}{=} \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \left(v_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \left(v_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\
&\overset{4}{=} v_1 \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_2 \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&\overset{5}{=} v_1 \begin{bmatrix} \mathrm{A}_{11} \\ \mathrm{A}_{21} \end{bmatrix} + v_2 \begin{bmatrix} \mathrm{A}_{12} \\ \mathrm{A}_{22} \end{bmatrix} \\
&\overset{6}{=} \begin{bmatrix} \mathrm{A}_{11} v_1 + \mathrm{A}_{12} v_2 \\ \mathrm{A}_{21} v_1 + \mathrm{A}_{22} v_2 \end{bmatrix}
\end{aligned}$$

It is important to discuss all the properties used in the derivation above, as they serve as a backbone to all calculations in linear algebra in general:

1. We have decomposed the vector **v** into its separate components.
2. We have pulled out the scalar from each vector in order to easily recognize the basis vectors $\mathbf{b_1}$ and $\mathbf{b_2}$.
3. Since we are dealing with a linear operator, we use the *additivity* property defined above.
4. Again, as we are dealing with a linear operator, we use *homogeneity* property defined above.
5. We use the definition of what matrix columns represent, i.e. we transform the canonical basis vectors accordingly.
6. We simply sum up the two remaining vectors.

Using known rules we have derived the elements of the transformed vector. This result is general, and if we have a matrix-vector multiplication of the type $\mathbf{w} = \mathbf{A}\mathbf{v}$, then the i-th element of the output vector **w** is given by:

$$\boxed{w_i = \sum_k \mathrm{A}_{ik} v_k} \tag{1}$$

Note that ik-th element of the matrix **A** is simply the entry of the matrix at the i-th row and k-th column. Using this formula, we can find every element of the output vector **w**.

In the example above, we have assumed that the matrix $\mathbf{A}$ is a square matrix, which resulted in vectors $\mathbf{v}$ and $\mathbf{w}$ having the same dimension. Let's take a look at another matrix, $\mathbf{A'}$. We will define $\mathbf{A'}$ as:

$$\mathbf{A'} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Now, let's try to interpret the meaning of this matrix. We have stated that the columns of the matrix correspond to the vectors to which our basis vector will transform. So, this means the following:

$$\mathbf{b_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \to \mathbf{b_1'} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{b_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \to \mathbf{b_2'} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

i.e. we have a transformation from $\mathbb{R}^2 \to \mathbb{R}^3$. To calculate how this matrix would transform an arbitrary vector, we would use the procedure same as above, and would again retrieve equation 1. As a simple exercise, let's calculate the output vector $\mathbf{w} = \mathbf{A'v}$, where vector $\mathbf{v} = [x, y]^{\mathrm{T}}$ using relation 1

$$\begin{aligned} w_1 &= \sum_{k=1}^{2} \mathrm{A}'_{1k} v_k = \mathrm{A}'_{11} v_1 + \mathrm{A}'_{12} v_2 = x + 0 = x \\ w_2 &= \sum_{k=1}^{2} \mathrm{A}'_{2k} v_k = \mathrm{A}'_{21} v_1 + \mathrm{A}'_{22} v_2 = 0 + y = y \\ w_3 &= \sum_{k=1}^{2} \mathrm{A}'_{3k} v_k = A'_{31} v_1 + A'_{32} v_2 = x + y \end{aligned}$$

Therefore, the output vector $\mathbf{w}$ is equal to:

$$\mathbf{w} = \begin{bmatrix} x \\ y \\ x + y \end{bmatrix}$$

This is exactly the mapping f defined in 2.5![5]

Let's summarize our current findings regarding matrix-vector multiplication:

- We have a general formula 1 for calculating how a matrix transforms a vector.
- The matrix-vector multiplication may or may not change the dimensionality of the input vector.
- If we have a $n \times k$ matrix (n rows, k columns), then the input vector has to be k-dimensional, while the output will be n-dimensional.
- All linear transformations (in finite dimensions) can be written in the matrix form.

Matrix-matrix multiplication

In the previous subsection, we have discussed how matrices (linear operators) transform vectors, and how to calculate elements of the transformed vectors. Matrix-matrix multiplication can be

[5] This is actually a very general result, all linear mappings (in finite dimensional vector spaces) can be written as matrix multiplication. More info can be found here.

thought of as chaining two transformations one after another, and for this reason, we can calculate the resulting matrix elements by analyzing how the two transformations act on the basis vectors. For simplicity, let's assume that we have two 2×2 matrices $\mathbf{A}$ and $\mathbf{B}$ of the following form:

$$\mathbf{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} \\ \mathrm{B}_{21} & \mathrm{B}_{22} \end{bmatrix}$$

Now, we wish to calculate elements of the resulting matrix $\mathbf{C} = \mathbf{AB}$. As we stated before, columns of the matrix represent to what the canonical basis vectors transform to. Therefore, for example, the first column of the matrix $\mathbf{C}$ will be given by the vector to which the vector $\mathbf{b}_1 = [1, 0]^{\mathrm{T}}$ will transform to. Let's calculate this by first acting with the matrix $\mathbf{B}$ and then with matrix $\mathbf{A}$ on the vector $\mathbf{b}_1$:

$$\begin{aligned} \mathbf{C}\,\mathbf{b}_1 &= (\mathbf{AB})\,\mathbf{b}_1 \\ &= \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} \\ \mathrm{B}_{21} & \mathrm{B}_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \begin{bmatrix} \mathrm{B}_{11} \\ \mathrm{B}_{21} \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{A}_{11}\mathrm{B}_{11} + \mathrm{A}_{12}\mathrm{B}_{21} \\ \mathrm{A}_{21}\mathrm{B}_{11} + \mathrm{A}_{22}\mathrm{B}_{21} \end{bmatrix}, \end{aligned}$$

where we have used identities and properties described in the matrix-vector multiplication section. Now, if we write the matrix $\mathbf{C}$ in the following form:

$$\mathbf{C} = \begin{bmatrix} \mathrm{C}_{11} & \mathrm{C}_{12} \\ \mathrm{C}_{21} & \mathrm{C}_{22} \end{bmatrix},$$

we can recognize that the elements of the first column are given by:

$$\begin{aligned} \mathrm{C}_{11} &= \mathrm{A}_{11}\mathrm{B}_{11} + \mathrm{A}_{12}\mathrm{B}_{21} \\ \mathrm{C}_{21} &= \mathrm{A}_{21}\mathrm{B}_{11} + \mathrm{A}_{22}\mathrm{B}_{21} \end{aligned}$$

Similarly, we could calculate the elements of the second column of the matrix $\mathbf{C}$ by observing how the two transformations transform the vector $\mathbf{b}_2 = [0, 1]^{\mathrm{T}}$.

In general, if we have a matrix-matrix multiplication of the type $\mathbf{C} = \mathbf{AB}$, the the ij-th element of the matrix $\mathbf{C}$ is given by:

$$\boxed{\mathrm{C}_{ij} = \sum_k \mathrm{A}_{ik}\mathrm{B}_{kj}} \tag{2}$$

It is important to note that we used a simple example where both matrices have the same dimensions. A more general case would be if the matrix $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times m}$. Then, the matrix $\mathbf{B}$ would take as the input a m-dimensional vector and transform it to a k-dimensional vector. Afterward, the matrix $\mathbf{A}$ would take as the input the transformed k-dimensional vector, and output a n-dimensional vector. So, the total transformation $\mathbf{C}$ would be a $n \times m$ matrix, i.e. $\mathbf{C} \in \mathbb{R}^{n \times m}$. Note that the elements of the matrix $\mathbf{C}$ would still be calculated using formula 2.

Next, let's take a look at two special types of matrices:

- **Identity matrix** - Identity matrix is often denoted by $\mathbf{I}$ or $\mathbb{I}$, and it represents a matrix that leaves a vector unchanged, i.e. $\mathbf{Iv} = \mathbf{v}$. Such matrix has elements 1 on the diagonal, and 0 otherwise. For example, a 3×3 identity matrix has the following form:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Inverse matrix** - An inverse of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}^{-1}$, and is defined by the following equation:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

  Intuitively, we can think of the inverse matrix $\mathbf{A}^{-1}$ as a matrix that counteracts the operation done by the matrix $\mathbf{A}$. Therefore, if we chain the two transformations together, it should be the same as if we did nothing (i.e. the total transformation is equal to the identity matrix $\mathbf{I}$). A matrix that has an inverse is called an invertible matrix, and only square matrices are invertible. More information can be found here.

Let's briefly summarize important information regarding matrix-matrix multiplication:

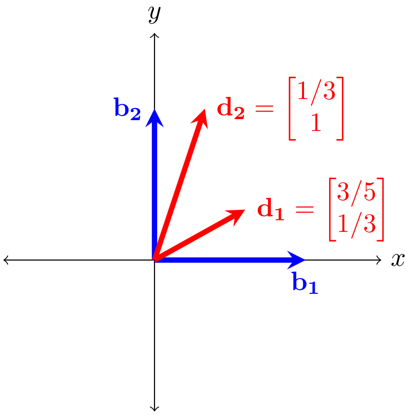
- Using formula 2 we can find elements of a matrix that is the result of matrix multiplication.
- Multiplying a $n \times k$ matrix with a $k \times m$ matrix will result in a $n \times m$ matrix.
- In general, matrix multiplication is not commutative, i.e. $\mathbf{AB} \neq \mathbf{BA}$.
- An identity matrix $\mathbf{I}$ leaves the vector unchanged.
- Some square matrices $\mathbf{A}$ have an inverse, which is denoted by $\mathbf{A}^{-1}$.

2.6 Change of Basis

In the previous section, the elements of the matrix were determined by how they transform the basis vectors. Let's take a closer look at two different basis in $\mathbb{R}^2$: a canonical basis $\{\mathbf{b_1}, \mathbf{b_2}\}$ and an arbitrary non-canonical basis $\{\mathbf{d_1}, \mathbf{d_2}\}$ whose elements can be expressed in the canonical basis as:

$$\mathbf{d_1} = \begin{bmatrix} 3/5 \\ 1/3 \end{bmatrix}, \quad \mathbf{d_2} = \begin{bmatrix} 1/3 \\ 1 \end{bmatrix} \tag{3}$$

The two bases are visualized in the figure below.


y
$\mathbf{b_2}$
$\mathbf{d_2} = \begin{bmatrix} 1/3 \\ 1 \end{bmatrix}$
$\mathbf{d_1} = \begin{bmatrix} 3/5 \\ 1/3 \end{bmatrix}$
x
$\mathbf{b_1}$

We can think of a basis as a language we use to explicitly write vectors and operators as matrices. However, the way an arbitrary operator $\mathbf{A}$ transforms a vector $\mathbf{v}$ shouldn't depend on the basis we use. Therefore, we must adjust the entries of the matrix depending on which basis we use, because as described before, rows of the matrix correspond to the vectors to which the basis vectors transform to. So let's try to motivate intuitively how we can transform a matrix $\mathbf{A}$ that is written in the canonical basis $\{\mathbf{b_1}, \mathbf{b_2}\}$ into a matrix $\mathbf{A}'$ which describes the same operation, but in the new basis $\{\mathbf{d_1}, \mathbf{d_2}\}$. The procedure is as follows:

1. We take a vector from the written using vectors from the new basis and *translate*[6] it into a *language* of the old basis using a transformation $\mathbf{S}$.
2. We act on this *translated* vector with the operator $\mathbf{A}$ expressed in the canonical basis.
3. We convert the transformed vector back to the *language* of the new basis using the inverse transformation $\mathbf{S}^{-1}$.

So, in total, we can express the change of basis of a matrix as:

$$\boxed{\mathbf{A}' = \mathbf{S}^{-1}\mathbf{A}\mathbf{S}} \tag{4}$$

Next question is, what does the transformation $\mathbf{S}$ between *languages* correspond to? Well, if we speak the *language* of the new basis, then we would express the vectors of the new basis as $\mathbf{d_1} = [1, 0]^{\mathrm{T}}$ and $\mathbf{d_2} = [0, 1]^{\mathrm{T}}$. However, if we want to express these new vectors in the old canonical basis, then we would write them in the form of equation 3. Therefore, the transformation $\mathbf{S}$ for this example is equal to:

$$\mathbf{S} = \begin{bmatrix} 3/5 & 1/3 \\ 1/3 & 1 \end{bmatrix}$$

The inverse transformation can be found and is equal to:

$$\mathbf{S}^{-1} = \begin{bmatrix} 45/22 & -15/22 \\ -15/22 & 27/22 \end{bmatrix},$$

which is not the nicest expression, but we can transform **any** operator $\mathbf{A}$ written in the canonical basis $\{\mathbf{b_1}, \mathbf{b_2}\}$ into a matrix $\mathbf{A}'$ written in the $\{\mathbf{d_1}, \mathbf{d_2}\}$ basis.

We can check whether the transformation $\mathbf{S}^{-1}$ makes sense by for example applying it on the vector $\mathbf{d_1}$ written in the canonical basis:

$$\mathbf{S}^{-1}\,\mathbf{d_1} = \begin{bmatrix} 45/22 & -15/22 \\ -15/22 & 27/22 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

which is exactly the expected result, because if we speak the *language* of the $\{\mathbf{d_1}, \mathbf{d_2}\}$ basis, we would write the vector $\mathbf{d_1}$ as $\mathbf{d_1} = [1, 0]^{\mathrm{T}}$.

[6] In this context, translation is meant in the context of the language, not as a spatial translation.

3 Multivariate Calculus

3.1 What are derivatives and why should we care?

Before deep diving into derivatives, it is reasonable to ask ourselves what we mean when we talk about the derivative of some function with respect to some variable. You may know that the derivative describes the **rate of change** of the function. With ‘rate of change’ we refer to how quickly the function value increases at some point x when we increase the value of x. A running metaphor we will use is the following. We can imagine a variable y which is formed through applying function f to x, i.e. $y = f(x)$. In this case, we call x an **input** and call y an **output**. We are often interested in studying how **sensitive** our outputs are to a change in the inputs, or how much our inputs **influence** our outputs, as we will get more into it soon. This sensitivity is exactly what is captured by the derivative, e.g. if the derivative of the output with respect to the input is large in some point, we know that output is ‘sensitive’ to a small change increase around that point. Now, you can picture this as a machine spitting out outputs y controlled with many knobs, where each knob corresponds to a variable x. The derivative tells us how sensitive the value our function spits out is to any turn of the knobs. Note that standard functions $f : \mathbb{R} \to \mathbb{R}$ are machines with one knob and spit out one value, but general functions $f : \mathbb{R}^m \to \mathbb{R}^n$ are machines with m knobs and spit out n different values. As we will look at later in this section, we have $m \times n$ derivatives in the latter case, for we can look at the sensitivity of each output to any of the knobs.

More important, perhaps, is the question of why we care about derivatives at all. In the context of machine learning, we are often very interested in a function that describes how well our model performs given our parameters. What we mean with ‘doing well’ is reflected in section 4, but for now, we presume that we have some measure of ‘doing well’. It is common to instead of maximizing performance, minimize the error we make, which are equivalent views on the same thing. Let us, for the sake of simplicity, say that our model parameter is given by x and our error rate is given by $f(x) = x^2 + 4x - 2$:

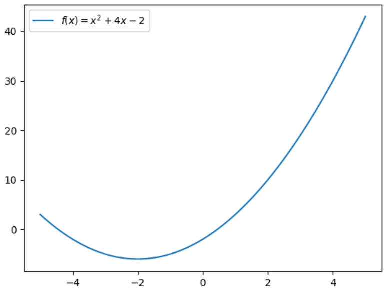


Figure 1: Example loss function. Horizontal axis describes the model parameter value x, the vertical axis describes the corresponding error $f(x)$.

If this function describes our error given our model parameters, we would be very interested in finding the point where this error rate is minimum, which is exactly why we want to use the derivative. We notice that in our minimum (which soon enough will turn out to be given by $x = -2$), the rate of change of our function is 0. Please take your time to verify this, because this point is crucial.

As you might remember from a previous calculus course, the derivative of the function $f(x)$ is given by $f'(x) = 2x + 4$:

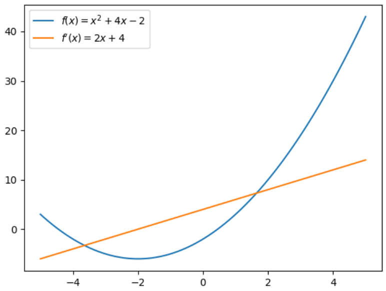


Figure 2: Example loss function plus its derivative. Horizontal axis describes the model parameter value x, the vertical axis describes the corresponding error $f(x)$ and derivative function $f'(x)$.

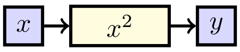
Here we see that for all points less than -2, indeed the derivative is negative (i.e. the function decreases) and for all points greater than $x = -2$, the derivative is positive (i.e. the function increases). It is exactly the minimum point $x = -2$ where be function does from decreasing to increasing. If we want to find this point $x = -2$ algebraically, we simply solve $2x + 4 = 0$, which of course turns out to be for $x = -2$.

The following sections will deep dive into how you can find these derivatives. We will first review the univariate case such as the function we just covered. We will then steadily work our way up to higher-dimensional derivatives with the aim of you being able to differentiate any ML/DL type of function.

We do now want to spend too much time on basic differentiation techniques and rather give you a general approach to differentiation from which things such as the sum rule, product rule, chain rule, et cetera, will follow directly. If you need a refresher on the basic derivative rules, we included them in Appendix A. Let us now dive into the actual derivatives!

3.2 Univariate derivatives

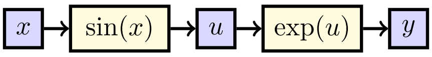
Let us start nice and easy with our basic functions over the reals, i.e. functions $f : \mathbb{R} \to \mathbb{R}$. Though this initially may look superfluous, we will introduce a visual way of representing these functions. This new approach will make it easier to consider multivariate functions and is commonplace in machine learning. Consider the function f such that $f : x \mapsto x^2$, i.e. the functions that squares its input. Again, our output is given by $y = f(x) = x^2$. In our example, we can visualize this function as follows:



The blue squares represent **values** and the yellow rectangles represent ways to **determine** a value. The most important insight you should take away is that the sensitivity of y to x is given by the sum of influences of all the paths from x to y. In this case, there is only one path, that is through the function x^2. Using basic differentiation techniques, we hence observe that:

$$\frac{dy}{dx} = \frac{dx^2}{dx} = 2x.$$

A slightly more *spicy* example if the function $f : \mathbb{R} \to \mathbb{R}$ such that $f : x \mapsto \exp(\sin(x))$.[7] If we make a diagram of this function as above, we can represent it as follows:



Please note that we had to introduce a new variable $u := \sin(x)$ that represents the intermediate value found after applying the sine function to x. When finding the derivative of y with respect to x, we again count all the paths from x to y. Again, there is only one path, now going through our intermediate value u. In this case, the effect of x on y is equal to the effect of x on u times the effect of u on y, i.e.

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}.$$

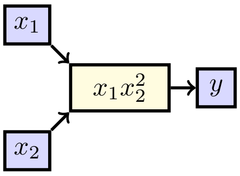
You may have encountered this separation of derivatives before as the **chain rule**. These derivatives are quite simple, giving us

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx} = \exp(u) \cdot \cos(x) = \exp(\sin(x))\cos(x),$$

where we substituted $u = \sin(x)$ in the last step. So, we **sum** all the paths from x to y, and we **multiply** the intermediate effects, e.g. if x influences u which influences y, the influence of x on y is the influence of x on u times the influence of u on y.

3.3 Multivariate derivatives

Let's go one step further, and consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ such that $f : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto x_1 x_2^2$. We can again draw this function:



In this case, we can consider two derivatives: we can look at the effect of x_1 on y and the effect of x_2 on y. When we can consider multiple derivatives for different variables, we do not write $\frac{dy}{dx_1}$ but rather $\frac{\partial y}{\partial x_1}$, to avoid confusion. We call such a derivative a **partial derivative**. Considering our earlier metaphor, a derivative in a real function is just the effect of turning a knob of a machine with one knob, whereas a partial derivative is an effect of turning one of the multiple knobs and keeping the other still. Luckily for us, we can still apply our same tricks and count the paths from a variable to y. In this case, we have that there is only one path from x_1 to y, and only one path from x_2 to y, giving us:

$$\frac{\partial y}{\partial x_1} = \frac{\partial x_1 x_2^2}{\partial x_1} = x_2^2,$$

[7] If you are not familiar with the $\exp(x)$ function, it is just another way to write e^x.

and

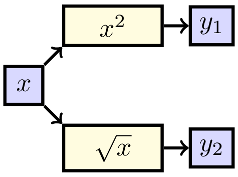
$$\frac{\partial y}{\partial x_2} = \frac{\partial x_1 x_2^2}{\partial x_2} = 2x_1x_2.$$

Please note that since we only consider the influence of one variable at the time, all the other variables are **constant** when taking derivatives. What we sometimes do, is write the 'full' derivative $\frac{df}{d\mathbf{x}}$ as the following vector:

$$\frac{dy}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} x_2^2 & 2x_1x_2 \end{bmatrix}.$$

We call this full derivative a **gradient** in the case we have functions $f : \mathbb{R}^n \to \mathbb{R}$, denoted as $\frac{dy}{d\mathbf{x}} = \nabla y(\mathbf{x}) = \text{grad } y(\mathbf{x})$. However, in the the general case of functions $f : \mathbb{R}^n \to \mathbb{R}^m$ we call the resulting matrix a **Jacobian**, denoted as $\frac{d\mathbf{y}}{d\mathbf{x}} = \mathbf{J}_\mathbf{y}(\mathbf{x})$. The Jacobian is just the matrix which has on its i row all the partial derivatives of y_i with respect to x_j, i.e. $\mathbf{J}_{ij} = \frac{\partial y_i}{\partial x_j}$. Hence, since we only have one output here, we have that the Jacobian has only one row.[8]

We can also have a function $\mathbf{f} : \mathbb{R} \to \mathbb{R}^2$ which maps $\mathbf{f} : x \mapsto \begin{bmatrix} x^2 \\ \sqrt{x} \end{bmatrix}$. In this case, we have that $\mathbf{y} = \mathbf{f}(x)$ where $\mathbf{y}$ is a vector (and hence is written in bold font), and thus we can consider $y_1 = x^2$ and $y_2 = \sqrt{x}$. Drawing this, we find:



When again looking at the paths, we see that

$$\frac{dy_1}{dx} = \frac{dx^2}{dx} = 2x,$$

and

$$\frac{dy_2}{dx} = \frac{d\sqrt{x}}{dx} = \frac{1}{2\sqrt{x}}.$$

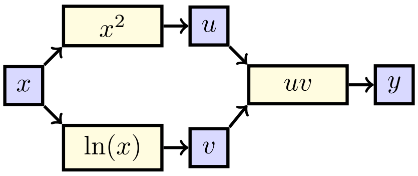
Here we can also group the different derivatives into one matrix:

$$\frac{d\mathbf{y}}{dx} = \begin{bmatrix} 2x \\ \frac{1}{2\sqrt{x}} \end{bmatrix}.$$

Please note that if we have a function $f : \mathbb{R}^n \to \mathbb{R}^m$ our Jacobian will be of the shape $m \times n$.

Now we are finally ready to consider a function with multiple streams of influence. Consider the $y = g(\mathbf{h}(x))$, where $\mathbf{h}(x) = (x^2, \ln(x))$ and $g(u, v) = uv$. That is, y is found by first calculating intermediate values $u = x^2$ and $v = \ln(x)$ and then finding $y = uv$. If we draw these functions, we see the following:

[8] For pedagogical reasons, we will call all such higher-order derivatives of y Jacobians and denote them with $\frac{d\mathbf{y}}{d\mathbf{x}}$, but in practice, most people will just use the word 'gradient' here anyway.



It is now very clear that the effect of x of y is twofold: both through u and v. As mentioned earlier, we need to consider all streams of influence. Specifically, we **sum** the different paths/effects, i.e.:

$$\frac{dy}{dx} = \frac{\partial y}{\partial u}\frac{du}{dx} + \frac{\partial y}{\partial v}\frac{dv}{dx}.$$

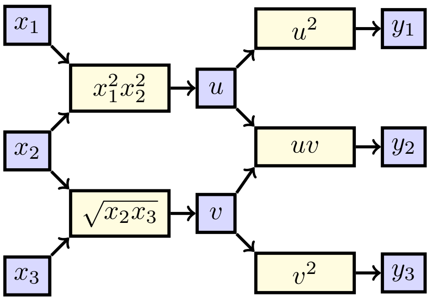
Plugging everything in, we find

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx} + \frac{dy}{dv}\frac{dv}{dx} = v \cdot 2x + u \cdot \frac{1}{x} = \ln(x) \cdot 2x + x^2 \cdot \frac{1}{x} = 2x(\ln(x) + \frac{1}{2}).$$

You may recognize this as the product rule, now you know where that comes from!

Finishing4 up, we go over one big example. Suppose $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ such that $f(x_1, x_2, x_3) = \mathbf{h}(\mathbf{g}(x_1, x_2, x_3))$, where $\mathbf{g}(x_1, x_2, x_3) = (x_1^2 x_2^2, \sqrt{x_2 x_3})$ and $\mathbf{h}(u, v) = (u^2, uv, v^2)$. Try it for yourself! Find $\frac{\partial y_2}{\partial x_2}$. Hint: draw out what happens.

When visualizing this function, we get the following:



When counting the paths from x_2 to y_2, we find two paths: one through u and one through v. We hence find

$$\frac{\partial y_2}{\partial x_2} = \frac{\partial y_2}{\partial u}\frac{\partial u}{\partial x_2} + \frac{\partial y_2}{\partial v}\frac{\partial v}{\partial x_2}.$$

Plugging our derivatives, we find

$$\frac{\partial y_2}{\partial x_2} = v \cdot 2x_1^2 x_2 + u \cdot \frac{x_3}{2\sqrt{x_2 x_3}} = 2x_1^2 x_2 \sqrt{x_2 x_3} + \frac{x_1^2 x_2^2 x_3}{2\sqrt{x_2 x_3}}.$$

Sweet! We now know how to find derivatives in multivariate functions. As you have seen, this approach is quite a time intensive, and sometimes (especially in deep learning) it is not necessary to write out everything by hand like this. This will be the topic of the rest of this section.

3.4 Jacobians

One of the most iconic functions in deep learning is the 'linear layer', which takes some input $\mathbf{x} \in \mathbb{R}^m$ and takes n linear combinations (with different factors) of the inputs. This linear layer can be considered a function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ such that $y_i = w_{i1}x_1 + \cdots w_{im}x_m = \sum_{j=1}^m, w_{ij}x_j$, where we still write $\mathbf{y} = \mathbf{f}(\mathbf{x})$. We call the $\{w_{ik}\}_{k=1}^m$ the **weights** of the function. Notice that for the entire function $\mathbf{f}$ we have n of such sets of weights, i.e. in total $n \times m$ weights. We can write this functions more compactly as

$$\mathbf{y} = \mathbf{W}\mathbf{x},$$

where

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m}.$$

When we now imagine all the streams of influence found between the $\mathbf{y}$ and $\mathbf{x}$, we realize that each element y_i is dependent on each variable x_j. If you do not see this immediately, please draw out the respective diagram.

A consequence of this is that we have a lot of derivatives, namely for each of the n outputs y_i we have m different derivatives (for the m inputs). To make our lives a whole lot easier, we simply determine the derivative of the ith element for the jth variable and see if what we end up with generalizes. We hence wanna find $\frac{\partial y_i}{\partial x_j} = \frac{d}{dx_j}(\sum_{k=1}^m w_{ik}x_k)$. We know that

$$\frac{d}{dx_j}(\sum_{k=1}^m w_{ik}x_k) = \sum_{k=1}^m \frac{d}{dx_j} w_{ik}x_k.$$

Let us know zoom in into one of the terms of the summation, i.e. we only consider $\frac{d}{dx_j} w_{ik}x_k$. If we have that $x_k \neq x_j$, we will always have that $\frac{d}{dx_j} w_{ik}x_k = 0$, because the entire term does not depend on x_j. When $x_j = x_k$, however, we see that the derivative is given by w_{ik}. We can express this 'if-else' statement quite easily mathematically using something called the **Kronecker delta**. The Kronecker delta over two variables i and j is equal to 1 if i is equal to j, and equal to 0 other, or:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Sometimes this is written with so-called **Iverson brackets** as $[i = j]$. These brackets do the same thing, i.e. $[\mathsf{S}] = 1$ if S is true, else $[\mathsf{S}] = 0$ for any statement S. The most important property (for us) of this Kronecker delta is that

$$\sum_j \delta_{ij}x_j = x_i,$$

i.e. when summing over elements x_j, we can filter out x_i by introducing δ_{ij}. Please verify this carefully, for this will be our main workhorse throughout this section.

If we go back to our example, we see that hence our derivative is given by $\frac{d}{dx_j} w_{ik}x_k = \delta_{jk}w_{ik}$ for any combination of x_j and x_k. That is, the derivative is equal to 0 if x_j and x_k are different, and equal to w_{ik} when x_j and x_k are the same. Plugging this back in, we find

$$\sum_{k=1}^m \frac{d}{dx_j} w_{ik}x_k = \sum_{k=1}^m \delta_{jk}w_{ik}.$$

This we know how to evaluate using our workhorse, and hence we see that

$$\frac{df_i}{dx_j} = \sum_{k=1}^{m} \delta_{jk} w_{ik} = w_{ij}.$$

Neat! We just found a general approach to taking the derivative of the linear layer and concluded that the effect of the jth variable on the ith output is given by the weight w_{ij}. We can write out the entire Jacobian (where the element in ith row, jth column is the derivative of y_i with respect to x_j) again:

$$\frac{d\mathbf{y}}{d\mathbf{x}} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m}.$$

But wait! This matrix we recognize from earlier, namely as our matrix $\mathbf{W}$. This allows us to write

$$\frac{d\mathbf{y}}{d\mathbf{x}} = \mathbf{W}.$$

We call this approach if finding a single entry of the derivative and then generalizing the 'index method'.

Please note that not only did we just derive the derivative of the linear layer, but we found that $\frac{d}{d\mathbf{x}}\mathbf{W}\mathbf{x} = \mathbf{W}$ for arbitrary matrices and vectors, e.g. we also know now that

$$\frac{d}{d\mathbf{v}}(\mathbf{AB} + \mathbf{C})\mathbf{v} = \mathbf{AB} + \mathbf{C},$$

by simply remembering that $\mathbf{AB} + \mathbf{C} = \mathbf{W}'$ for some matrix $\mathbf{W}'$.

Another very common derivative you will encounter is $\frac{d}{d\mathbf{x}}\mathbf{a}^{\mathbf{T}}\mathbf{x}$. In this case, we have that $\mathbf{a}^{\mathbf{T}}\mathbf{w}$ is simply a scalar, and hence our Jacobian will be of the shape $(1 \times m)$ if $\mathbf{x}$ is m-dimensional. Let us again use the index method, and aim to find

$$\frac{d}{dx_j}\mathbf{a}^{\mathbf{T}}\mathbf{x} = \frac{d}{dx_j}\sum_{k=1}^{m} a_k x_k = \sum_{k=1}^{m} \frac{d}{dx_j} a_k x_k.$$

As before, we see that the derivative is equal to a_k when $k = j$, and equal to zero otherwise, and thus

$$\sum_{k=1}^{m} \frac{d}{dx_j} a_k x_k = \delta_{jk} a_k = a_j,$$

where we find a_j by applying our workhorse again. This means that the jth element of our derivative is given by a_j, or the entire derivative is given by $\mathbf{a}$.

But... $\mathbf{a}$ is a column vector, where our Jacobian should be a row vector as we argued earlier. Sadly, this problem cannot quite be overcome, and we just need to always check of our answer should be transposed or not. In this case, we see that our Jacobian matches $\mathbf{a}^{\mathbf{T}}$. This is slightly annoying, but luckily our answer is always either correct or needs to be only transposed, and checking it will become second nature soon enough! Let this inconvenience not distract us from the fact that we did just find our new identity though, that is:

$$\frac{d}{d\mathbf{x}}\mathbf{a^T x} = \mathbf{a^T}.$$

Now it is your turn, please try and verify that $\frac{d}{d\mathbf{x}}\mathbf{y^T sAx} = \mathbf{y^T A}$, where $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{n\times m}$, and $\mathbf{y} \in \mathbb{R}^m$. Please do this 1) using index notation, and 2) using our identity friends we have already found.

We know that $\mathbf{y^T Ax}$ is a scalar (why?), and hence the Jacobian will be again of the form $(1 \times m)$ if $\mathbf{x}$ is a m-dimensional vector. Using index notation, we aim to find $\frac{d}{dx_i}\mathbf{y^T Ax}$. We observe that

$$\frac{d}{dx_i}\mathbf{y^T Ax} = \frac{d}{dx_i}\sum_{k=1}^{n}\sum_{j=1}^{m} y_k A_{kj} x_j = \sum_{k=1}^{n}\sum_{j=1}^{m}\frac{d}{dx_i} y_k A_{kj} x_j.$$

Again, since $y_k A_{kj}$ are just scalars, we know that the derivative is simply found by

$$\frac{d}{dx_i} y_k A_{kj} x_j = \delta_{ij} y_k A_{kj}.$$

This gives us the following derivative:

$$\sum_{k=1}^{n}\sum_{j=1}^{m}\frac{d}{dx_i} y_k A_{kj} x_j = \sum_{k=1}^{n}\sum_{j=1}^{m}\delta_{ij} y_k A_{kj} = \sum_{k=1}^{n} y_k A_{ki}.$$

But this term we recognize as $[\mathbf{y^T A}]_i$. This means that our full derivative is simply given by $\mathbf{y^T A}$, which aligns with our desired shape so we are done.

So... That's quite a lot of work. And actually, we could have done way less work using our previous identities. Observe that $\mathbf{y^T A}$ is just a row vector, i.e. it can be written as $\mathbf{v^T} = \mathbf{y^T A}$ for some vector $\mathbf{v}$. Thus, we can write $\mathbf{y^T Ax} = \mathbf{v^T x}$. But this we know how to differentiate with our tricks, that is $\frac{d}{d\mathbf{x}}\mathbf{v^T x} = \mathbf{v^T}$, and hence we know that $\frac{d}{d\mathbf{x}}\mathbf{y^T Ax} = \mathbf{y^T A}$.

This should cover the basics of vector calculus! During the first week of the course, we will spend some more time on time on this and you will receive an excellent document written by two other TAs. If you understand these basics, you are well on your way to doing machine learning soon enough!

4 Statistical Learning

In this section, we will do a quick overview of to goal of machine learning in general. Though machine learning is quite a broad topic in general, exploring one specific type of machine learning will cast some light on the topic in general. What we will look at is **supervised learning**, in which we have example input-output pairs of some process, and we want to learn the **relationship** between the inputs and outputs. A classic example is having many examples of images of dogs and cats plus a label for each image saying either 'cat' or 'dog'. We would then want to find a model that takes in the image (input) and predicts either 'cat' or 'dog' (output). In general in machine learning, we assume that we do not ourselves already know what this model looks like exactly. If we would, then why bother going for this statistical approach rather than just implementing the function directly?

Let us imagine we aim to predict your **grade** based on only **number of hours studied**. This of course is slightly silly to do, which is exactly why it is a good example to consider! Our goal is to find a function y that takes in hours studied x and predicts your grade based on the data: the examples (x, t) of numbers of hours studied and the grade that person obtained. We denote $\mathcal{D} = \{(x_n, t_n)\}$ as the entire dataset, and we will assume it contains N points. Ideally, our function y satisfies that

$$y(x_n) = t_n \qquad \text{for all } (x_n, t_n) \in \mathcal{D}.$$

However, this will **never** be possible. Suppose you and your friend both study together for the same number of hours, that does not imply that you will necessarily obtain the same grade (which is why this example is silly). Thus, for the same x_n, we can have a whole range of different t_n that are 'correct'. However, our function y can (by definition) only return one value $y(x_n)$ to estimate t_n, so what do we do now...

This is where we call out our (soon-to-be) best friend statistics. Instead of considering our points as just fixed values, we consider them as **random variables** (or RVs). We will not bother writing this out formally right now, but in essence, a random variable X is just an object that can take on specific values (e.g. x_1, x_2, x_3, and x_4) and assigns a probability describing how 'probable' each assignment is.[9] We write $p(X = x_1)$ as the probability that X takes on the value x_1.

If we have more than one RV, let's say X and Y, which can take on values $\{x_i\}$ and $\{y_j\}$ respectively, we can make a distribution describing how likely X and Y are joint. We call this the **joint distribution**, denoted as $p(X, Y)$, which can take on values $p(X = x_i, Y = y_j)$ for all combinations i and j.

With this joint distribution, we can define our (for now) last type of distribution. Suppose X is someone's age and Y someone's height, the distribution $p(X, Y)$ describes how likely people of certain ages are to have certain heights. However, what if we want to just know how the heights are distributed for people that are 22 years old? This information is actually in the $p(X, Y)$, but we need to filter out all the points for which we have that $X \neq 22$. We denote this filtered distribution, or **conditional distribution**, as $P(Y \mid X)$ in general, or as $P(Y \mid X = 22)$ for a specific age.

Back to our example. Let X denote our input variables and T denote the corresponding data outputs. We are specifically interested in finding the distribution $p(T \mid X)$ which for any datapoint X assigns a distribution of targets T. To decide what this distribution should look like, we zoom out and consider the data generation process for a second.

[9] Actually, what 'probabilities' are is quite a divisive thing. The two main interpretations are the frequentist approach and the Bayesian approach. Though you probably learned mostly frequentist approaches, it is worth deep diving into Bayesian approaches for they lend themselves better to machine learning often.

We know that for any value $X = x$, we can have many different valid predictions that are valid, and we wanna quantify the validity of each one using a distribution. First, how do we assess how 'close' our predictions are to the actual values? E.g., if the height of someone was $1.70m$ and I predicted $1.50m$, is that twice as bad as predicting $1.60m$? Or is it four times as bad? Actually, for this, we will in general use the **mean squared error** (or MSE). As the name suggests, your 'wrongness' is given by the average $(y(x_n) - t_n)^2$ over all data points, i.e.

$$\mathsf{MSE} = \frac{1}{N}\sum_{n=1}^{N}(y(x_n) - t_n)^2.$$

What is super convenient about using MSE, is that **a lot** of mathematics will simplify. Please note, though, that this is just a choice (though one often made), and choosing different error functions can have drastic effects on what your model will consider 'good' predictions.

The main reason why MSE is used so often is that under a few mild assumptions, the function y that is 'ideal' (i.e. the function that minimizes the MSE) is simply the function that for some point x returns the weighted average of the target $T \mid X = x$, which we write as the expected value:

$$f(x) = \mathbb{E}[T \mid X = x].$$

That is to say, right as we might expect it, if we have 100 people that are 22 years old, of which there are 50 of height $1.70m$, 30 of height $1.60m$, and 20 of height $1.80m$, and we predict $0.5 \cdot 1.70m + 0.3 \cdot 1.6m + 0.2 \cdot 1.8m = 1.69m$ (and do the same for all other ages), this function will exactly minimize the mean squared error. That's pretty neat, right?

This is where ML1 starts! We will be studying ways of predicting target distributions for our different inputs, i.e. finding $p(T \mid X)$. Ideally, we would always use $y(x) = \mathbb{E}[T \mid X = x]$, but in practice, we will not have enough data for each input point x to predict the average value of T for those x. This is where you come in, using your linear algebra and multivariate calculus tools to find a good solution to this problem!

A Derivative rules

Here we provide the basic derivative rules. We separated them into (1) derivatives of specific functions, and (2) properties of the derivatives of combined functions.

Standard derivatives

- $f(x) = c \implies f'(x) = 0$,
- $f(x) = x^n \implies f'(x) = nx^{n-1}$,
- $f(x) = a^x \implies f'(x) = a^x \log a$, and hence $f(x) = e^x \implies f'(x) = e^x$,
- $f(x) = \log_b x \implies f'(x) = \frac{1}{\log(b) \cdot x}$, and hence $f(x) = \log x \implies f'(x) = \frac{1}{x}$,
- $f(x) = \sin x \implies f'(x) = \cos x$,
- $f(x) = \cos x \implies f'(x) = -\sin x$.

Moreover, it is useful to remember special cases of the second rule, e.g. $f(x) = x \implies f'(x) = 0$, $f(x) = ax \implies f'(x) = a$, and $f(x) = \sqrt{x} \implies f'(x) = \frac{1}{2\sqrt{x}}$.

Derivative rules

- $(c \cdot f)'(x) = c \cdot f'(x)$,
- $(f + g)'(x) = f'(x) + g'(x)$,
- $(f \cdot g)'(x) = f'(x) \cdot g(x) + f(x) \cdot g'(x)$,
- $(\frac{f}{g})'(x) = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$,
- $(f \circ g)'(x) = (f' \circ g)(x) \cdot g'(x)$.