

Week 1: Preliminaries

<http://mlvu.github.io>

February 9, 2021

This week, the homework will be a review of subjects you should be familiar with already. Machine learning is built mostly on three types of math: *Linear Algebra*, *Calculus* and *Probability Theory*. Luckily, we only use a very limited number of concepts from each, so even if you don't know these subjects or don't know them well, it shouldn't take too much work to get up to speed.

Here we'll review some linear algebra and some calculus, saving the probability for later. We'll also review three concepts that you should become very familiar with to follow machine learning theory: sums, expectations and logarithms. Even if you know these in principle, you should take the time to get comfortable with them.

This homework should give you an indication of whether you have a sufficient grasp of the preliminaries. If you struggle with anything, please follow the links provided to brush up, before continuing with next week's homework.

1 Sums, expectations and logarithms

You've probably encountered all three of these before. However, in the lectures, we will often see long derivations which require you to be very familiar with these in order to follow all the steps. If you find yourself baffled by some of the math in the lecture, you can probably get a lot closer by just practising your sums, expectations and logarithms a little.

Sums Sigma (Σ) notation is simply a concise way of writing down long sums. Assume we have a list of numbers a_1, a_2, a_3, a_4, a_5 . We can write:

$$a_1 + a_2 + a_3 + a_4 + a_5$$

but we can also write

$$\sum_{i=1}^5 a_i \quad \text{or} \quad \sum_{i=1}^5 a_i$$

The bit below the sigma tells you which symbol is used as an index (i) to iterate over the elements to sum over, and what the first value is (1). The bit above the sigma tells you what the last value of the index is. ¹

Since this is a very dense notation, it's often simplified by leaving out the start and end values, assuming that they can be figured out from context:

$$\sum_i a_i$$

Sums can be nested as well (using a sum within a sum). This is not a special notation, it simply follows from the definition. You can 'unpack' these by turning the sigmas into a regular sum one by one:

$$\begin{aligned} & \sum_{i=1}^2 \sum_{j=1}^3 a_i b_j \\ &= \sum_{j=1}^3 a_1 b_j + \sum_{j=1}^3 a_2 b_j \\ &= (a_1 b_1 + a_1 b_2 + a_1 b_3) + (a_2 b_1 + a_2 b_2 + a_2 b_3) \end{aligned}$$

For nested sums, the indices are often combined into a single sigma:

$$\sum_{i,j} a_i b_j = \sum_i \sum_j a_i b_j$$

question 1: Let $x_1 = 5, x_2 = 1, x_3 = 4, x_4 = 1, x_5 = 3$. Compute the following values:

1. $\sum_i x_i$
2. $\sum_i 2 \cdot x_i$
3. $2 \cdot \sum_i x_i$

¹If you're an experienced programmer, it may be helpful to think of this as a for loop: something like `for(int i = 1; i < 6; i++) ...`

4. $\sum_{i=1}^3 x_i$
5. $\sum_{i=1}^{x_5} x_i$
6. $\sum_{i=1}^3 i \cdot x_i$

question 2: Which of the following are always true? If you struggle, try rewriting as a normal sum, and using what you already know about sums. The variable c is a constant.

1. $\sum_i c y_i = c \sum_i y_i$
2. $\sum_i (c + y_i) = c + \sum_i y_i$
3. $\sum_{i=1}^n (c + y_i) = n c + \sum_{i=1}^n y_i$
4. $\sum_i (x_i + y_i) = (\sum_i x_i) + (\sum_i y_i)$
5. $\sum_i \sum_j a_i b_j = \sum_j \sum_i a_i b_j$
6. $\sum_i a_i b_i = (\sum_i a_i) (\sum_i b_i)$

It can be a little ambiguous where the sum operator ends. For instance, in $\sum_i x_i + c$, we don't know whether the c should be added once, or for every term in the sum. There's no official standard, and authors should make sure to write sums in an unambiguous way. For instance $\sum_i c + x_i$ or $c + \sum_i x_i$ or simply $\sum_i (c + x_i)$.

The capital pi Π works in exactly the same way as the Σ operator, but describes a *product*:

$$\prod_{i=1}^4 i = 24$$

More practice? Try these links:

- <https://www.khanacademy.org/math/algebra2/sequences-and-series/alg2-sigma-notation/e/evaluating-basic-sigma-notation>
- <https://www.mathsisfun.com/algebra/sigma-notation.html>
- <https://www.youtube.com/watch?v=TjMLzklmn2c>

Expectations An expectation is nothing more than a sum, with each term weighted by a probability. Let's say you are playing a game where you roll a die, and you receive the number of eyes rolled in euros. Your expected reward is:

$$\frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = \frac{21}{6} = 3.5$$

In general, if you have a random variable V (an experiment with a numeric outcome) which has possible outcomes $1, 2, \dots$ with associated values v_1, v_2, \dots and associated probabilities p_1, p_2, \dots , the **expected value** of the experiment is written as

$$\mathbb{E}(V) = \sum_i p_i v_i \quad (1)$$

The brackets can be omitted at the author's discretion. If multiple probability distributions are applicable, the choice of distribution may be indicated in the subscript $\mathbb{E}_p V$.

We can also extend the experiment by applying a function to the outcome, and computing the expected value over the result. Say that V describes a betting game with winnings v_i , but you will have to pay a 20% tax over whatever you win. Then, the expectation for the amount of money you actually get to keep is

$$\mathbb{E}(0.8 \cdot V) = \sum_i p_i \cdot 0.8 \cdot v_i$$

question 3: Because the expectation is basically a sum, most of the things that hold for sums, also hold for expectations. Show that the following hold by applying what you know about sums. If you get stuck, try expanding the expectation into a sum, as in (1).

1. $\mathbb{E}(c \cdot V) = c \cdot \mathbb{E}(V)$ (homogeneity)
2. Let V and W have different numeric values, v_i and w_i , for outcomes with the same probability p_i . Then $\mathbb{E}(V) + \mathbb{E}(W) = \mathbb{E}(V + W)$ (additivity)
3. $\mathbb{E}(V) + \mathbb{E}(\sin(V)) = \mathbb{E}(V + \sin(V))$
4. $\mathbb{E}((V - \mathbb{E}V)^2) = \mathbb{E}(V^2) - (\mathbb{E}V)^2$

If your outcomes form a *continuum*, like real-valued numbers (e.g. if you sample from a normal distribution), the expectation is defined with an integral instead of a sum ($E_V = \int_x p(x)v(x)dx$). Since the integral has most of the same properties as the sum, you can use both types of expectation in the same way.²

More practice? Try these links:

- <https://www.khanacademy.org/math/probability/probability-geometry/expected-value-geo/a/expected-value-basic>

Logarithms The *logarithm* is nothing more than the inverse of the exponential. If $y = b^x$ then $x = \log_b y$.

Since $100 = 10 \cdot 10 = 10^2$, $1000 = 10 \cdot 10 \cdot 10^3$ and so on, $\log_{10}x$ gives us an indication of the *order of magnitude* of x . That is, how many digits do we need to write down x . To be precise, $1 + \log_{10}(x)$ gives us the number of digits in x , with the value after the decimal point telling us how close we are to needing more digits. For instance $\log_{10}(314) \approx 2.5$ and $\log_{10}(31400) \approx 4.5$.

In other words, if two numbers have the same order or magnitude, the difference between their base-10 logarithms will be at most 1. This also works for numbers smaller than one, except that the logarithm becomes negative: $\log_{10}(0.1) = -1$, $\log_{10}(0.01) = -2$, $\log_{10}(0.001) = -3$ and so on.

The base-10 logarithm is between 1 and 2 for all numbers between 10 and 100, between 2 and 3 for all numbers between 100 and 1000, between 3 and 4 for all numbers between 1000 and 10 000 and so on. Every increase of 1 in the range captures 10 times as much of the domain.

For bases other than 10, it's not quite so neat, but the same principle holds: **all numbers with a given order of magnitude are squeezed into a constant-sized interval.**

There are two main reasons why it's such a useful function in machine learning:

- The standard numerical representations used in computers have *limited precision*: they cannot accurately represent differences smaller than about 10^{-16} . Instead of storing the actual number, *we can store the logarithm of the number*. This is useful, for instance, in representing probabilities. Since our probabilities are usually guesses anyway, we don't care as much about the difference between 0.0004 and

²In fact, we use this trick in the course to hide some of the complexity: by talking about expectations without unpacking the notations, we can discuss many aspects of distributions on continuous spaces, without ever using integrals.

0.00035 as we do about the difference between 10^{-21} and 10^{-24} . If we store the logarithm of the probability, the second difference remains clear, even if the actual probability is far too small to store.

- Taking the logarithm **turns a product into a sum**:

$$\log(\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{c}) = \log \mathbf{a} + \log \mathbf{b} + \log \mathbf{c}$$

$$\log \prod_i x_i = \sum_i \log x_i$$

Sums are much easier to analyze and manipulate. This can be a tremendous help in working out derivatives and in other symbolic manipulation.

question 4: Which of the following are true for all \mathbf{a}, \mathbf{b} ? If true show why, from the following facts:

- If $x = c^y$ then $y = \log_c x$ (and vice versa).
- $c^x \cdot c^y = c^{x+y}$ (check that you have an intuition for why this is true).

If not true, give a counterexample.

1. $\log_b(b^a) = a$
2. $b^{\log_b a} = a$
3. $\log_c a + \log_c b = \log_c(ab)$. Hint: let $a' = \log_c a$ and $b' = \log_c b$ and apply one of the rules given.
4. $\log(a^b) = b \log(a)$
5. $\log(a + b) = \log(b) \cdot \log(a)$
6. $\log(a/b) = \log(a) - \log(b)$

More practice? Try these links:

- <http://www.oxfordmathcenter.com/drupal7/node/532>
- <https://www.khanacademy.org/math/algebra2/exponential-and-logarithmic-functions>
- <https://betterexplained.com/articles/demystifying-the-natural-logarithm-ln>

2 Linear Algebra

A **vector** is a list of numbers, of a fixed length. The length (how many numbers there are in the vector) is its *dimension*. Here is a vector of dimension 4:

$$\mathbf{x} = \begin{pmatrix} 3 \\ 3.1 \\ -2.5 \\ 0.1 \end{pmatrix}$$

In our course materials, the variable name for a vector is a **bold** lowercase letter. To indicate the individual elements of the vector, we use the non-bold version of the letter, subscripted with an index: $x_3 = -2.5$. The notation $\mathbf{x} \in \mathbb{R}^4$ means that \mathbf{x} is a vector consisting of four, real-valued numbers.

Vectors are often used to denote points or directions in space. For instance, for a vector in \mathbb{R}^2 :

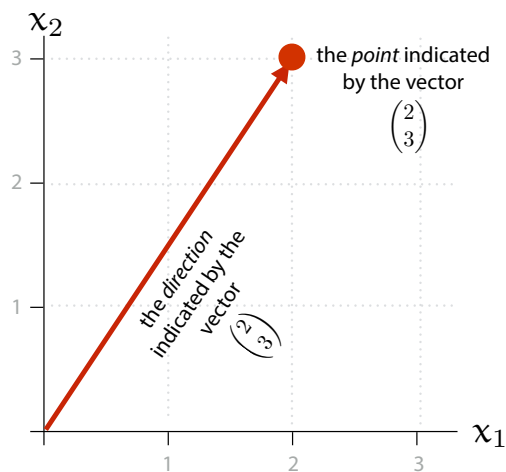


Figure 1: Vectors as points and vectors as directions.

If we think of the vector \mathbf{x} as an arrow (as shown in Figure 1), the length of that arrow, denoted $\|\mathbf{x}\|$, is ³

$$\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}.$$

³This follows from the Pythagorean theorem. For a 2D example, see Figure 1. Let $l = \|\mathbf{x}\|$. By Pythagoras $l^2 = x_1^2 + x_2^2$, so $l = \sqrt{x_1^2 + x_2^2}$.

A **matrix** is also a bunch of numbers, but arranged in a *grid*. Here is an example of a 3×4 matrix:

$$\mathbf{W} = \begin{pmatrix} 3 & 5 & 1.1 & 0 \\ 0 & 3.1 & 3 & 1 \\ 1 & 9 & -2.5 & 0 \end{pmatrix}$$

In our course materials, the variable name for a matrix is a **bold uppercase** letter. To indicate the individual elements of the matrix, we use the non-bold version of the letter with two indices: $W_{31} = 1$. Note that the row-index is first, followed by the column index. The notation $\mathbf{W} \in \mathbb{R}^{3 \times 4}$ means that \mathbf{W} is a 3-by-4 matrix with real-valued elements.

Note that both the columns and the rows of a matrix form vectors. In general, it is good to keep track of whether a vector is a row-vector (oriented horizontally) or a column-vector (oriented vertically).

element-wise operations In general, when we talk about applying an operation like $\sin(\cdot)$ to a vector or matrix, we mean applying the operation element-wise. That is in $\mathbf{U} = \sin(\mathbf{W})$, \mathbf{U} is the matrix where each element is computed by applying the \sin operation to the corresponding element in \mathbf{W} : $U_{ij} = \sin(W_{ij})$. This also holds for summing, multiplying or dividing by a scalar. $\mathbf{W} + 3$ just means that you add 3 to every element in \mathbf{W} . Likewise for multiplication:

$$\begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix} \times 4 = \begin{pmatrix} 4 & 0 & 8 & 4 \\ 0 & 4 & 4 & 8 \\ 8 & 4 & 8 & 0 \end{pmatrix}$$

For binary operations over two matrices, like $\mathbf{W} + \mathbf{V} = \mathbf{U}$, we require that both matrices have the same size. The result is just the matrix you get by lining up \mathbf{W} and \mathbf{V} and summing the elements of each: $U_{ij} = W_{ij} + V_{ij}$. Similarly for multiplication:

$$\begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

transpose The transpose of a matrix, written as \mathbf{W}^T , turns the columns into rows, and the rows into columns:

$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 1 \end{pmatrix}$$

You can think of this as flipping the matrix around the **primary diagonal** (from top-left to bottom-right). Note that the *first* row becomes the *first* column; the axes are swapped, but the order of the elements along the axes is not changed (we don't flip the matrix horizontally or vertically).

The transpose of a column vector is the corresponding row vector (and vice versa). This is often used when we want to specify a column vector without wasting too much whitespace: $\mathbf{x} = (3, 3.1, 3.1, -2.5, 0.1)^T$.

dot product The dot product is a simple operation on two vectors that we will encounter *a lot*. The dot product of vectors \mathbf{x} and \mathbf{y} (of equal dimension) is defined as:

$$\mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i .$$

That is, we multiply the two vectors element-wise and sum the result. We will usually write the dot product as $\mathbf{x}^T \mathbf{y}$, to avoid ambiguity with element-wise multiplication. This notation follows from the operation of matrix-multiplication explained below.

The following is a very useful identity for the dot product:

$$\mathbf{x}^T \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos \alpha$$

Where α is the angle between the two vectors (when we interpret both as a direction, as shown in Figure 1).

matrix multiplication When we talk about matrix multiplication, we don't usually mean the element-wise variant, but a kind of generalization of the dot product.

Here is a good visual illustration of matrix multiplication.⁴

⁴Source: This file was derived from: Matrix multiplication diagram.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=15175268>, by user Bilou

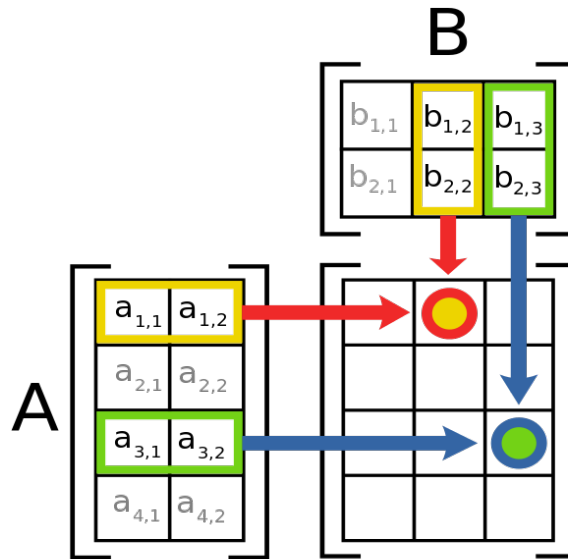


Figure 2: Matrix multiplication

If $\mathbf{AB} = \mathbf{C}$, then we compute element C_{ij} by taking the dot product of the i -th row of \mathbf{A} and the j -th column of \mathbf{B} .

Note that the first matrix must have the same number of *columns*, as the second has *rows*. The other dimensions can be chosen freely.

One specific application of matrix multiplication is matrix-by-vector multiplication (where we see a column vector as just an n -by-1 matrix). The multiplication $\mathbf{y} = \mathbf{W}\mathbf{x}$ transforms vector \mathbf{x} into vector \mathbf{y} . If \mathbf{W} is a square matrix, this operation is called a *map*.

The notation $\mathbf{x}^T \mathbf{y}$ for the dot product should now be clear: we are matrix multiplying \mathbf{x} (transposed to a row vector) by \mathbf{y} , resulting in a 1-by-1 matrix.

Both the dot product and matrix multiplication have many different intuitive explanations. We don't have room to develop these intuitions here (follow the links below if you're curious), but throughout the course, you'll see different ways of using these concepts.

question 5: Explain in words what the following notations represent:

1. $f: \mathbb{R}^3 \rightarrow \mathbb{R}^2$
2. $\mathbf{y} = \mathbf{W}\mathbf{x}$
3. $z = \mathbf{y}^T \mathbf{x}$

4. $\mathbf{W} \in \mathbb{R}^{5 \times 4}$

In future, if you see unfamiliar notation, see if you can find the symbols in this page: https://en.wikipedia.org/wiki/List_of_mathematical_symbols. Even if that doesn't explain it fully, it may provide you with some keywords to google.

question 6: Which of the following operations are possible, and which aren't?

1. Multiplying a 5×4 matrix by another 5×4 matrix.
2. Element-wise multiplying a 5×4 matrix by another 5×4 matrix.
3. Multiplying a 5×4 matrix by a 4×5 matrix.
4. Multiplying a 5×4 matrix by a 4×1 matrix.
5. Element-wise multiplying a 5×4 matrix by a 4×5 matrix.
6. Multiplying any matrix by its transpose.
7. Element-wise multiplying any matrix by its transpose.
8. Element-wise multiplying a square matrix by its transpose.

question 7: Which of the following is true?

1. Let $\mathbf{U} = \mathbf{V}\mathbf{W}$ for matrices \mathbf{U} , \mathbf{V} , \mathbf{W} . u_{ij} is the dot product of the i -th column of \mathbf{V} and the j -th column of \mathbf{W} .
2. Let $\mathbf{U} = \mathbf{V}\mathbf{W}$ for matrices \mathbf{U} , \mathbf{V} , \mathbf{W} . u_{ij} is the dot product of the i -th row of \mathbf{V} and the j -th column of \mathbf{W} .
3. Matrix multiplication is *commutative*.
4. Matrix multiplication is *distributive*.
5. Matrix multiplication is *associative*.
6. There exist matrices \mathbf{U} and \mathbf{V} such that $\mathbf{UV} = \mathbf{VU}$.

question 8:

1. The linear function $f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$ can be written as the dot product of two vectors. What are the vectors?
2. The linear function $f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} + \delta$ can be also be written as the dot product of two vectors. What are the vectors?
3. The *quadratic* function $f(\mathbf{a}, \mathbf{b}) = \alpha\mathbf{a}^2 + \beta\mathbf{a}\mathbf{b} + \gamma\mathbf{b}\mathbf{a} + \delta\mathbf{b}^2$ can be written as $f(\mathbf{x}) = \mathbf{x}^T \mathbf{W} \mathbf{x}$ (also known as a bilinear product). What should \mathbf{x} and \mathbf{W} be?

If you found these exercises completely impossible, you should brush up on your Linear Algebra a little. You don't need much, the basics of vectors, matrices and matrix multiplication should do the trick. The following articles may help. If not, hunt around for one that explains things at your level:

- <https://betterexplained.com/articles/linear-algebra-guide/>
- <https://www.khanacademy.org/math/linear-algebra>
- <http://samples.jbpub.com/9781556229114/chapter7.pdf>

If you find anything that helps you, let us know on the discussion board.

3 Calculus

To denote the derivative of $f(\mathbf{x})$ we will use the notation $\frac{df(\mathbf{x})}{d\mathbf{x}}$ (with ∂ taking the place of d for partial derivatives). You may be more comfortable with the notation $f'(\mathbf{x})$ for the derivative. I sympathize, but when it comes to machine learning, the former notation makes things *much* simpler down the line, so I recommend getting used to it.

We will assume that you know the basics of calculus (specifically, taking 1D derivatives), so we won't offer an explanation here. Use the exercises below to check whether your understanding is sufficient. If not, practice using the links below.

question 9: Let $f(\mathbf{x}) = 3\mathbf{x}^2 + 5\mathbf{x} + 1$ with \mathbf{x} a scalar.

1. What is the derivative of $f(\mathbf{x})$?
2. For which \mathbf{x} is $f(\mathbf{x})$ at its minimum?
3. Let $h(\mathbf{x}) = g(f(\mathbf{x}))$, with f defined as above. Let $\frac{dg(\mathbf{x})}{d\mathbf{x}} = \sin(\mathbf{x})/\mathbf{x}$. Without knowing what $g(\mathbf{x})$ is (or working it out), can we find the derivative of $h(\mathbf{x})$?

question 10: Let $\mathbf{x} \in \mathbb{R}^2$ and let $f(\mathbf{x}) = 3x_1^2 + 4x_1x_2 - x_2^2$

1. What is the *partial* derivative of $f(\mathbf{x})$ with respect to x_1 ?
2. What is the partial derivative of $f(\mathbf{x})$ with respect to x_2 ?
3. What is the *gradient* of $f(\mathbf{x})$?
4. The gradient is a function derived from f , just like the derivative is a function. What are the domain and range of the gradient of $f(\mathbf{x})$?

As before, if you found these exercises tricky, even after reading the answers, you should probably brush up a little on your calculus. Here are some links you may find helpful.

- <https://betterexplained.com/articles/vector-calculus-understanding-the-gradient/>
- <https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/partial-derivative-and-gradient-articles/a/introduction-to-partial-derivatives>
- <http://tutorial.math.lamar.edu/Classes/CalcIII/PartialDerivatives.aspx>