EMAT31530: Mathematical preliminaries

Laurence Aitchison

Here, I'm going to introduce all the mathematical preliminaries and notation that I'm going to use, with a view to their later use in AI. This was originally designed for CS students, rather than the more Math-y cohort in the course. So you've probably seen the early stuff, but not necessarily the later stuff. The emphasis might be quite different from stuff you've seen before (as I'm developing stuff necessary for AI). Everything here is examinable, so please do at least skim through. And in any case, you really have a go at the should do the exercises (especially the last ones): they're great preparation for later lectures.

1 Calculus

AI is all about calculus. PyTorch is — in a very literal sense — one big calculus engine.

1.1 Polynomials

The derivative of x^p, where p is some power,

$$\frac{dx^p}{dx} = px^{p-1}. \tag{1}$$

Perhaps the most important example (which we're going to encounter many times) is the derivative of a quadratic,

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

But the formula also applies for higher powers,

$$\frac{dx^5}{dx} = 5x^4. \tag{3}$$

And for negative powers,

$$\frac{dx^{-3}}{dx} = -3x^{-4}. \tag{4}$$

And for fractional powers,

$$\frac{d\sqrt{x}}{dx} = \frac{dx^{1/2}}{dx} = \frac{1}{2}x^{-1/2} = \frac{1}{2\sqrt{x}}. \tag{5}$$

And for powers of zero,

$$\frac{d1}{dx} = \frac{dx^0}{dx} = 0x^{-1} = 0. \tag{6}$$

(note that $x^0 = 1$, a constant, so the gradient has to be zero). And for powers of one,

$$\frac{dx}{dx} = \frac{dx^1}{dx} = 1x^0 = 1. \tag{7}$$

(note that $x^1 = x$, which has a slope of 1). We can apply the rule to each term in a polynomial,

$$\frac{d}{dx}[3x^4 + 2x^{-1/2} + x^{-2}] = 3\frac{dx^4}{dx} + 2\frac{dx^{-1/2}}{dx} + \frac{dx^{-2}}{dx} \tag{8}$$

Looking at each term separately,

$$\frac{dx^4}{dx} = 4x^3 \tag{9}$$

$$\frac{dx^{-1/2}}{dx} = -\tfrac{1}{2}x^{-1.5} \tag{10}$$

$$\frac{dx^{-2}}{dx} = -2x^{-3} \tag{11}$$

Putting everything back together,

$$\frac{d}{dx}[3x^4 + 2x^{-1/2} + x^{-2}] = 3(4x^3) + 2(-\tfrac{1}{2}x^{1.5}) - 2x^{-3} \tag{12}$$

$$= 12x^3 - x^{1.5} - 2x^{-3}. \tag{13}$$

1.2 Chain rule

To differentiate more complex expressions, we need the chain rule. For instance, we might have,

$$\frac{dy}{dx} \qquad \text{where} \qquad y(x) = (x+1)^3 \tag{14}$$

We could expand the brackets, but we don't want to because that would be alot of terms. Instead, we rewrite y in terms of u,

$$u = x + 1 \qquad\qquad y = u^3 \tag{15}$$

Then, we use the chain rule,

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}. \tag{16}$$

And each of these derivatives is much easier,

$$\frac{dy}{du} = \frac{du^3}{du} = 3u^2 \tag{17}$$

$$\frac{du}{dx} = \frac{dx+1}{dx} = \frac{dx}{dx} + \frac{d1}{dx} = 1 + 0 = 1 \tag{18}$$

Subsituting these derivatives into the chain rule, we get our answer,

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx} = 3u^2 \times 1 \tag{19}$$

And finally substituting $u = x + 1$,

$$\frac{dy}{dx} = 3(x+1)^2. \tag{20}$$

Despite its simplicity in this example, it turns out that the chain rule is the basis of backprop, and backprop is the basis of all modern AI, from ChatGPT to Stable Diffusion to just about everything else. This is because the chain rule allows you to "chain" together derivatives in a complex, multi-step pipeline. But that's for later.

1.3 Product rule

The final key rule of calculus is the product rule,

$$\frac{du(x)v(x)}{dx} = u(x)\frac{dv(x)}{dx} + v(x)\frac{du(x)}{dx}. \tag{21}$$

Again, this rule can be used to avoid an explosion of terms. For instance,

$$y = (x + x^2 + x^3)(2x + 3x^2 + x^3) \tag{22}$$

has 9 terms and is really painful to work with. Instead, if we set,

$$u(x) = x + x^2 + x^3 \tag{23}$$

$$v(x) = 2x + 3x^2 + x^3 \tag{24}$$

We get,

$$y = u(x)v(x). \tag{25}$$

Now, we can use the product rule!

$$\frac{dy}{dx} = \frac{du(x)v(x)}{dx} \tag{26}$$

$$= u(x)\frac{dv(x)}{dx} + v(x)\frac{du(x)}{dx} \tag{27}$$

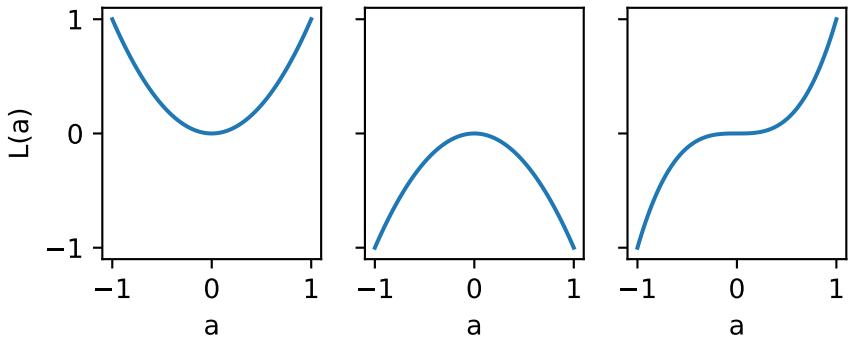
$$= (x + x^2 + x^3)\frac{d}{dx}[2x + 3x^2 + x^3] + (2x + 3x^2 + x^3)\frac{d}{dx}[x + x^2 + x^3] \tag{28}$$

$$= (x + x^2 + x^3)(2 + 6x + 3x^2) + (2x + 3x^2 + x^3)(1 + 2x + 3x^2) \tag{29}$$

1.4 Using calculus to do analytic optimization

At the top, we mentioned that we were going to use calculus to find e.g. the best fitting model (e.g. straight line) to some data. The starting point is to find a function that measures how good our model is at fitting the data. Typically, our model has some tunable parameters (e.g. the slope of a straight line, which we're going to call a). Then, we have a "loss" function $\mathcal{L}(a)$, which takes our tunable parameters as input, and tells us how well the corresponding model fits the data; small values indicate a good fit, and bad values indicate a bad fit. Typically, the loss function is defined in terms of a *distance*, e.g. the distance between our predictions and the data (we'll see this in more depth later).

For the moment, our question is how to find the best model, i.e. the value of a with the smallest loss, $\mathcal{L}(a)$. Well, this is an optimization problem, and it turns out we can use calculus to solve optimization problems.



Usually, in AI, we find the optimum by gradient descent: following the gradient down hill until you reach a minimum. But in simple settings, we can solve for the minimum analytically: the minimum of the loss is usually at a location where the gradient is zero (left). Of course, we have to be careful with this: the gradient can also be zero at a maximum (middle), or at neither a maximum or minmum (right). Though issues caused by this are very rare, and AI people generally ignore these possibilities.

For instance, the loss might be as simple as a quadratic,

$$\mathcal{L}(a) = a^2 + 5a - 3 \tag{30}$$

To minimize $\mathcal{L}(a)$, we find the place where the slope is zero,

$$0 = \frac{d\mathcal{L}(a)}{da} \tag{31}$$

$$0 = \frac{d}{da}[a^2 + 5a - 3] \tag{32}$$

$$0 = \frac{da^2}{da} + \frac{d5a}{da} - \frac{d3}{da} \tag{33}$$

$$0 = 2a + 5 \tag{34}$$

Then, we can solve for a,

$$2a = -5, \tag{35}$$

$$a = -\tfrac{5}{2}. \tag{36}$$

Now, this looks kind-of simplistic. Of course, the real calculation is more complicated, largely because it involves summing over datapoints (which we will see next). But finding the best model/straight line really does end up involving minimizing a quadratic!

2 Sums and products

Typically, we're going to be working N datapoints, where the ith datapoint is x_i. So the first datapoint is x_1, the second datapoint is x_2, and the last datapoint is x_N. We're often going to need to sum or multiply across many different datapoints, similarly to a for loop.

To sum over datapoints, we use the summation notation. For instance, the sum of i^2 for $i = 1$ to $i = 3$ can be written,

$$\sum_{i=1}^{3} i^2 = 1^2 + 2^2 + 3^2. \tag{37}$$

If we have datapoints, x_i, then we could sum over datapoints using,

$$\sum_{i=1}^{N} x_i = x_1 + x_2 + \cdots + x_N, \tag{38}$$

which indicates that we should sum x_i for $i = 1$ to $i = N$.

To take a product over datapoints, we use product (or capital pi) notation. This is just like the sum, except that we multiply each term, rather than adding

them. For instance, the product of i for $i = 1$ to $i = 3$ can be written,

$$\prod_{i=1}^{3} i = 1 \times 2 \times 3. \tag{39}$$

We could also take the product over datapoints, x_i, using,

$$\prod_{i=1}^{N} x_i = x_1 \times x_2 \times \cdots \times x_N, \tag{40}$$

which indicates that we should take the product of x_i for $i = 1$ to $i = N$.

I will generally try to be explicit about the upper and lower limits. But sometimes, especially in other people's material, you will see abbreviated notation, missing off the upper and lower limits. If the limits are left off, then there should be some "natural" values for them to take on. For instance, if we know that there are N datapoints ranging from x_1 to x_N, then "natural" lower limit is $i = 1$ and the "natural" upper limit is $i = N$.

$$\sum_i x_i = \sum_{i=1}^{N} x_i \qquad\qquad \prod_i x_i = \prod_{i=1}^{N} x_i \tag{41}$$

Of course this is context dependent. The natural limits here are $i = 1$ to $i = N$ only because this we knew we had N datapoints.

3 Logarithms

Here, log is *always* the natural logarithm, i.e.

$$x = e^{\log x} \qquad\qquad x = \log(e^x). \tag{42}$$

(and this is also true in numerical programming e.g. Python, PyTorch etc.) For our purposes, the logarithm is super-useful because it converts products into sums,

$$\log(b \times c) = \log b + \log c. \tag{43}$$

This extends to powers,

$$\log(x^y) = \log \underbrace{(x \times x \times \cdots \times x)}_{y \text{ times}} \tag{44}$$

$$= \underbrace{\log x + \log x + \cdots + \log x}_{y \text{ times}} \tag{45}$$

$$= y \log x. \tag{46}$$

But most importantly, it extends to products and summations over many elements,

$$\log(\prod_{i=1}^{N} x_i) = \sum_{i=1}^{N} \log x_i \tag{47}$$

Using logs to switch products to sums is going to be important for two reasons:

- its much easier to do e.g. calculus on sums than on products.
- If you have a big product, the result might lie outside the range of float32/float64. The minimum value of float32 is 1.2×10^{-38}. If we have $x_i = 0.1$ and $N = 38$, then $\prod_{i=1}^{N} x_i = 10^{-38}$, and we're only just inside the range of float32's.

4 Vectors, matrices and index notation

4.1 Notation/types

To be super-clear about everything's type, we always ensure scalars/vectors/matrices are distinguishable. In particular, we always write:

- Scalars as non-bold (e.g. a or A).
- Vectors as bold and lowercase (e.g. $\mathbf{a}$). (Single-underline when handwritten.)
- Matrices as bold and uppercase (e.g. $\mathbf{A}$). (Double-underline when handwritten.)

Unfortunately, vectors can be row *or* column vectors, depending on context. This ambiguity is PyTorch / Python's fault. But we can't really do anything about it. The components of the row-vector $\mathbf{r}$ and column-vector $\mathbf{c}$ can be written,

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \qquad \mathbf{r} = \begin{pmatrix} r_1 & r_2 & \dots & r_N \end{pmatrix} \tag{48}$$

Here, c_i or r_i is a single component of the vector. Note that elements of vectors/matrices, here c_i or r_i, are always scalars, so they are always non-bold.

We write matrices as bold upper-case letters, such as $\mathbf{A}$. A single component of $\mathbf{A}$ is a scalar, so it written non-bold, as A_{ij}, where i indexes the row and j indexes the column. We write the shape as $N \times M$, which means it has N rows and M columns.

- i indexes rows and we have N rows, so i runs from 1 to N.
- j indexes columns and we have M columns, so j runs from 1 to M.

Remember that we write N first in $N \times M$, and i is the first index in A_{ij}, so it is i that runs from 1 to N.

Dropping the colours, an $N \times M$ matrix $\mathbf{A}$, can be written,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM}. \end{pmatrix}, \tag{49}$$

Note how the first index maxes out in the last row at N and the second index maxes out in the last column at M.

4.2 Vectors as matrices with one row/column

Its going to be useful when we come to transposes and matrix products to think of a vector as a matrix with either one row or one column.

A column vector $\mathbf{c}$ with length N, can be written,

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix}, \tag{50}$$

and can be understood as an $N \times 1$ matrix.

A row vector, $\mathbf{r}$ with length M, can be written,

$$\mathbf{r} = \begin{pmatrix} r_1 & r_2 & \dots & r_M \end{pmatrix}. \tag{51}$$

and can be understood as an $1 \times M$ matrix.

4.3 Transposes

The transpose "mirrors" the matrix or vector along the diagonal. For instance, transpose converts the $N \times 1$ column vector $\mathbf{c}$ into a $1 \times N$ row vector, $\mathbf{c}^T$,

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \qquad \mathbf{c}^T = \begin{pmatrix} c_1 & c_2 & \dots & c_N \end{pmatrix} \tag{52}$$

Likewise, it converts the 2×3 matrix $\mathbf{B}$, into the 3×2 matrix $\mathbf{B}^T$,

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad \mathbf{B}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \tag{53}$$

4.4 Matrix-matrix product

The matrix product is just a very short notation for writing a sum and a product. Before delving into the exact definition, its worth thinking about the matrix sizes. For the product of $\mathbf{A}$ and $\mathbf{B}$,

$$\underbrace{\mathbf{C}}_{M\times P} = \underbrace{\mathbf{A}}_{M\times N}\underbrace{\mathbf{B}}_{N\times P}. \tag{54}$$

In particular:

- the two inner dimensions, N, must be the same size and they're going to disappear (these are the dimensions we're going to sum over).
- $\mathbf{C}$ and $\mathbf{A}$ have the same number of rows, M (first).
- $\mathbf{C}$ and $\mathbf{B}$ have the same number of columns, P (second).

Now lets delve in to the exact definition as a sum and product,

$$C_{ik} = \sum_{j=1}^{N} A_{ij}B_{jk}. \tag{55}$$

We've highlighted the indices in the same colors as above. You can see that the structure of the indices matches that of the sizes above. In particular:

- we sum over the inner two indices, j (i.e. the second index of $\mathbf{A}$ and the first index of $\mathbf{B}$.
- i is the first index of $\mathbf{A}$ and $\mathbf{C}$ (rows).
- k is the second index of $\mathbf{B}$ and $\mathbf{C}$ (columns).

4.4.1 Matrix multiplication by hand

To actually do the matrix multiplication, you can use the "two finger method". You move your left finger left-to-right along a row on the first matrix, and your right finger top-to-bottom down a column on the second matrix. You then multiply the first two elements you see, and add them to your running total. You choose the row and column based on the element you're trying to calculate. If you're trying to calculate the element in the second row and the third column, then you'd run your left finger along the second row and you'd run your right finger down the third column.

$$\begin{pmatrix} r_{11} & \\ & \end{pmatrix} = \begin{pmatrix} \rightarrow & \rightarrow \\ & \end{pmatrix}\begin{pmatrix} \downarrow & \\ \downarrow & \end{pmatrix} \tag{56}$$

$$\begin{pmatrix} & r_{12} \\ & \end{pmatrix} = \begin{pmatrix} \rightarrow & \rightarrow \\ & \end{pmatrix}\begin{pmatrix} & \downarrow \\ & \downarrow \end{pmatrix} \tag{57}$$

$$\begin{pmatrix} \\ r_{21} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \rightarrow \quad \rightarrow \\ \\ \end{pmatrix} \begin{pmatrix} \downarrow \\ \downarrow \\ \\ \end{pmatrix} \tag{58}$$

$$\begin{pmatrix} \\ \\ r_{22} \\ \end{pmatrix} = \begin{pmatrix} \\ \rightarrow \quad \rightarrow \\ \\ \end{pmatrix} \begin{pmatrix} \\ \downarrow \\ \downarrow \\ \end{pmatrix} \tag{59}$$

4.4.2 Matrix multiplication example

You can verify the first line by using the "two-finger method" described above, or by using the explicit formula for matrix multiplication.

$$\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 5 \times 1 + 6 \times 3 & 5 \times 2 + 6 \times 4 \\ 7 \times 1 + 8 \times 3 & 7 \times 2 + 8 \times 4 \end{pmatrix} \tag{60}$$

$$= \begin{pmatrix} 5 + 18 & 10 + 24 \\ 7 + 24 & 14 + 32 \end{pmatrix} \tag{61}$$

$$= \begin{pmatrix} 23 & 34 \\ 31 & 46 \end{pmatrix} \tag{62}$$

4.5 Matrix-vector product

A matrix-vector product is the same as the matrix-matrix product, but we take $P = 1$. That gives sizes,

$$\underbrace{\mathbf{v}}_{M \times 1} = \underbrace{\mathbf{A}}_{M \times N} \underbrace{\mathbf{c}}_{N \times 1} \tag{63}$$

where both $\mathbf{v}$ and $\mathbf{c}$ are column-vectors. The exact form for the summation is usually written a bit differently, omitting k because there's only one possible value for k,

$$v_i = \sum_{j=1}^{N} A_{ij} c_j. \tag{64}$$

4.6 Vector-matrix product

A vector-matrix product is the same as the matrix-matrix product, but we take $M = 1$. That gives sizes,

$$\underbrace{\mathbf{v}}_{1 \times P} = \underbrace{\mathbf{r}}_{1 \times N} \underbrace{\mathbf{B}}_{N \times P} \tag{65}$$

where both $\mathbf{v}$ and $\mathbf{r}$ are row-vectors. The exact form for the summation is usually written a bit differently, omitting i because there's only one possible value for i,

$$v_k = \sum_{j=1}^{N} r_j A_{jk} \tag{66}$$

4.7 Vector inner product

A vector inner product is the same as the matrix-matrix product, but we take $M = 1$ and $P = 1$, so the output is scalar. For a vector inner product, the first argument needs to be a row-vector, and the second argument needs to be a column-vector,

$$\underbrace{v}_{1\times 1} = \underbrace{\mathbf{r}}_{1\times N}\underbrace{\mathbf{c}}_{N\times 1} \tag{67}$$

$$\tag{68}$$

where remember $\mathbf{r}$ is a row-vector, $\mathbf{c}$ is a column-vector, and v is a scalar (or a 1×1 matrix). We write the summation for omitting i and k,

$$c = \sum_{j=1}^{N} r_j c_j. \tag{69}$$

Using transposes, we can also use the vector inner product to combine a column matrix with itself, or a row matrix with itself,

$$\underbrace{v}_{1\times 1} = \underbrace{\mathbf{c}^T}_{1\times N}\underbrace{\mathbf{c}}_{N\times 1} \qquad\qquad \underbrace{v}_{1\times 1} = \underbrace{\mathbf{r}}_{1\times M}\underbrace{\mathbf{r}^T}_{M\times 1} \tag{70}$$

4.8 Vector outer product

A vector-vector product is the same as the matrix-matrix product, but we take $N = 1$, so the output is a matrix. For a vector inner product, the first argument needs to be a column-vector, and the second argument needs to be a row-vector,

$$\underbrace{\mathbf{V}}_{M\times P} = \underbrace{\mathbf{c}}_{M\times 1}\underbrace{\mathbf{b}^T}_{1\times P} \tag{71}$$

where remember $\mathbf{c}$ is a column-vector, $\mathbf{r}$ is a row-vector, and $\mathbf{V}$ is a matrix. We can write the outer product in index notation as,

$$C_{i,k} = a_i b_k. \tag{72}$$

Using transposes, we can also use the vector outer product to combine a column matrix with itself, or a row matrix with itself,

$$\underbrace{\mathbf{V}}_{N\times N} = \underbrace{\mathbf{c}}_{N\times 1}\underbrace{\mathbf{c}^T}_{1\times N} \qquad\qquad \underbrace{\mathbf{V}}_{M\times M} = \underbrace{\mathbf{r}^T}_{M\times 1}\underbrace{\mathbf{r}}_{1\times M} \tag{73}$$

4.9 Identity matrix

The identity matrix has ones along the diagonal, and zero elsewhere,

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{74}$$

If we multiply the identity matrix by any vector / matrix, we just get back the same thing,

$$\mathbf{IC} = \mathbf{C} \tag{75}$$
$$\mathbf{CI} = \mathbf{C} \tag{76}$$
$$\mathbf{rI} = \mathbf{r} \tag{77}$$
$$\mathbf{Ic} = \mathbf{c} \tag{78}$$

4.10 Matrix inverse

The inverse, $\mathbf{A}^{-1}$, of a matrix, $\mathbf{A}$ is the matrix such that,

$$\mathbf{I} = \mathbf{A}^{-1}\mathbf{A} \tag{79}$$

In the 2×2 case,

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \mathbf{A}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{80}$$

We can prove that this holds by first substituting the value of $\mathbf{A}$ and $\mathbf{A}^{-1}$,

$$\mathbf{A}^{-1}\mathbf{A} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{81}$$

Then we use the expression for matrix multiplication,

$$\mathbf{A}^{-1}\mathbf{A} = \frac{1}{ad - bc}\begin{pmatrix} ad - bc & db - bd \\ -ca + ac & ad - bc \end{pmatrix} \tag{82}$$
$$\mathbf{A}^{-1}\mathbf{A} = \frac{1}{ad - bc}\begin{pmatrix} ad - bc & 0 \\ 0 & ad - bc \end{pmatrix} \tag{83}$$
$$\mathbf{A}^{-1}\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{I} \tag{84}$$

In the "real-world", we use the computer to compute our matrix inverses.

5 Kronecker delta

The Kronecker delta, δ_{ij} appears when we start differentiating sums / vectors /matrices. Its a bit like the indices of an identity matrix (except that we never write I_{ij},

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{85}$$

.

5.1 The Kronecker delta often appears when we're taking gradients of vectors

Consider a vector,

$$\mathbf{a}^T = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \tag{86}$$

One thing we might want to do (usually as part of a larger calculation) is compute the gradient of $\mathbf{a}$ wrt one of the components, a_1,

$$\frac{d\mathbf{a}^T}{da_2} = \begin{pmatrix} \frac{da_1}{da_2} & \frac{da_2}{da_2} & \frac{da_3}{da_2} \end{pmatrix} \tag{87}$$

The gradient is zero when the variables don't match, as e.g. changing a_2 doesn't cause a_1 to change at all,

$$\frac{d\mathbf{a}^T}{da_2} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \tag{88}$$

Now, we can do the same thing a bit more abstractly,

$$\mathbf{a}^T = \begin{pmatrix} a_1 & a_2 & \dots & a_N \end{pmatrix} \tag{89}$$

$$\frac{d\mathbf{a}^T}{da_j} = \begin{pmatrix} \frac{da_1}{da_j} & \frac{da_2}{da_j} & \dots & \frac{da_N}{da_j} \end{pmatrix} \tag{90}$$

To represent that these gradients are 1 only when the top index matches the bottom index, we use the Kronecker delta,

$$\frac{d\mathbf{a}^T}{da_j} = \begin{pmatrix} \delta_{1j} & \delta_{2j} & \dots & \delta_{Nj} \end{pmatrix} \tag{91}$$

And if we select out the ith element of the vector,

$$\frac{da_i}{da_j} = \delta_{ij}. \tag{92}$$

5.2 The Kronecker delta picks out an element of a sum

You should be happy to see a Kronecker delta turn up, because it typically makes things alot simpler! In particular, we often have Kronecker deltas in sums, and the Kronecker delta "picks out" one element of the sum.

$$\sum_{j=1}^{3} \delta_{2j} a_j = \delta_{21} x_1 + \delta_{22} x_2 + \delta_{23} x_3 \tag{93}$$

$$= 0x_1 + 1x_2 + 0x_3 \tag{94}$$

$$= x_2. \tag{95}$$

This notion of "picking out one element" is perhaps easier to see in the more general case,

$$\sum_j \delta_{ij} a_j = a_i, \tag{96}$$

which happens because δ_{ij} is zero for all j except when $j = i$.

6 Exercises

You can skip the unit exercises if they look straightforward to you. But please do try the last one!

Exercise 1. *Calculate:*

$$\frac{d}{dx}[4x^{2.5} + x^{1/2} - 6x^{-1/2}] \tag{97}$$

Exercise 2. *Use the chain rule to calculate:*

$$\frac{d}{dx}[(x+1)^3] \tag{98}$$

Exercise 3. *Calculate the same thing, without using the chain rule, by explicitly expanding the brackets and applying* $\frac{dx^p}{dx} = px^{p-1}$*,*

$$\frac{d}{dx}[(x+1)^3] \tag{99}$$

Check they give the same answer!

Exercise 4. *Find the minimum of,*

$$\mathcal{L}(a) = 4a^2 + 2a + 1. \tag{100}$$

Exercise 5. *Calculate,*

$$\sum_{i=0}^{5} i^2. \tag{101}$$

Exercise 6. *Calculate,*

$$\prod_{i=1}^{5} i. \tag{102}$$

Exercise 7. *Simplify,*

$$\sum_{i=1}^{N} b \tag{103}$$

where b *does not depend on* i*.*

Exercise 8. *Simplify,*

$$\prod_{i=1}^{N} b \tag{104}$$

where b *does not depend on* i*.*

Exercise 9. *Compute the matrix product,*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \tag{105}$$

Exercise 10. *Compute the matrix inverse,*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} \tag{106}$$

Exercise 11. *Use the matrix inverse to solve the following expression for x_1 and x_2,*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{107}$$

Exercise 12. *Find the value of*

$$\frac{dy_i}{dw_k} \tag{108}$$

in terms of the fixed $\mathbf{X}$*, where* $\mathbf{y}$ *is given by,*

$$\mathbf{y} = \mathbf{X}\mathbf{w} \tag{109}$$

7 Answers

Answer 1. *Calculate:*

$$\frac{d}{dx}[4x^{2.5} + x^{1/2} - 6x^{-1/2}] = 4\frac{dx^{2.5}}{dx} + \frac{dx^{1/2}}{dx} - 6\frac{dx^{-1/2}}{dx} \tag{110}$$

$$= 4(2.5x^{1.5}) + \tfrac{1}{2}x^{-1/2} - 6(-\tfrac{1}{2}x^{-1.5}) \tag{111}$$

$$= 10x^{1.5} + \tfrac{1}{2}x^{-1/2} + 3x^{-1.5}) \tag{112}$$

Answer 2. *Use the chain rule to calculate:*

$$\frac{d}{dx}[(x+1)^3] \tag{113}$$

set

$$u = (x+1) \tag{114}$$

$$y = (x+1)^3 = u^3 \tag{115}$$

Thus we can apply the chain rule,

$$\frac{d}{dx}[(x+1)^3] = \frac{dy}{dx} \tag{116}$$

$$= \frac{du}{dx}\frac{dy}{du} \tag{117}$$

$$= \frac{dx+1}{dx}\frac{du^3}{du} \tag{118}$$

$$= 1 \times 3u^2 \tag{119}$$

$$= 3(x+1)^2 \tag{120}$$

Answer 3. *Calculate the same thing, without using the chain rule, but explicitly expanding the brackets and applying* $\frac{dx^p}{dx} = px^{p-1}$*,*

$$\frac{d}{dx}[(x+1)^3] = \frac{d}{dx}[(x+1)(x+1)(x+1)] \tag{121}$$

$$= \frac{d}{dx}[(x^2+2x+1)(x+1)] \tag{122}$$

$$= \frac{d}{dx}[(x^3+2x^2+x) + (x^2+2x+1)] \tag{123}$$

$$= \frac{d}{dx}[(x^3+3x^2+3x+1)] \tag{124}$$

$$= 3x^2 + 6x + 3 \tag{125}$$

$$= 3(x^2+2x+1) \tag{126}$$

$$= 3(x+1)^2 \tag{127}$$

Answer 4. *Find the minimum of,*

$$\mathcal{L}(a) = 4a^2 + 2a + 1. \tag{128}$$

Solve for the value of a where the gradient is zero,

$$\begin{aligned} 0 &= \frac{d\mathcal{L}(a)}{da} && (129)\\ &= \frac{d}{da}[4a^2 + 2a + 1] && (130)\\ &= 4\frac{d}{da}[a^2] + 2\frac{da}{da} + \frac{d1}{da} && (131)\\ &= 4(2a) + 2 && (132)\\ &= 8a + 2. && (133) \end{aligned}$$

Now, we can solve for a,

$$\begin{aligned} 8a &= -2 && (134)\\ a &= -\tfrac{1}{4}. && (135) \end{aligned}$$

Answer 5. *Calculate,*

$$\begin{aligned} \sum_{i=0}^{5} i^2 &= 0^2 + 1^2 + 2^2 + 3^2 + 4^2 + 5^2 && (136)\\ &= 0 + 1 + 4 + 9 + 16 + 25 && (137)\\ &= 14 + 16 + 25 && (138)\\ &= 30 + 25 && (139)\\ &= 55 && (140) \end{aligned}$$

Answer 6. *Calculate,*

$$\begin{aligned} \prod_{i=1}^{5} i &= 1 \times 2 \times 3 \times 4 \times 5 && (141)\\ &= (2 \times 3) \times (4 \times 5) && (142)\\ &= 6 \times 20 && (143)\\ &= 120 && (144) \end{aligned}$$

Answer 7.

$$\sum_{i=1}^{N} b = \underbrace{b + b + \cdots + b}_{N \ times} = Nb \tag{145}$$

Answer 8.

$$\prod_{i=1}^{N} b = \underbrace{b \times b \times \cdots \times b}_{N \ times} = b^N \tag{146}$$

Answer 9.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix} \tag{147}$$

$$= \begin{pmatrix} 5 + 14 & 6 + 16 \\ 15 + 28 & 18 + 32 \end{pmatrix} \tag{148}$$

$$= \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix} \tag{149}$$

Answer 10. *Matrix inverse:*

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} = \frac{1}{1 \times 4 - 2 \times 3} \begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} \tag{150}$$

$$= \frac{1}{4 - 6} \begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} \tag{151}$$

$$= -\frac{1}{2} \begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} \tag{152}$$

$$= \frac{1}{2} \begin{pmatrix} -4 & 2 \\ 3 & -1 \end{pmatrix} \tag{153}$$

Answer 11.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{154}$$

Multiply on both sides by the inverse of the matrix,

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{155}$$

A matrix times matrix-inverse is the identity,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{156}$$

We can substitute for the value of the matrix inverse from the previous question,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -4 & 2 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{157}$$

Then compute the matrix-vector product,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} (-4) \times 5 + 2 \times 6 \\ 3 \times 5 + (-1) \times 6 \end{pmatrix} \tag{158}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -20 + 12 \\ 15 - 6 \end{pmatrix} \tag{159}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -8 \\ 9 \end{pmatrix}. \tag{160}$$

We can check this value for x_1 and x_2 is correct by substituting it back in,

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\frac{1}{2}\begin{pmatrix} -8 \\ 9 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \times (-8) + 2 \times 9 \\ 3 \times (-8) + 4 \times 9 \end{pmatrix} \tag{161}$$

$$= \frac{1}{2}\begin{pmatrix} -8 + 18 \\ -24 + 36 \end{pmatrix} \tag{162}$$

$$= \frac{1}{2}\begin{pmatrix} 10 \\ 12 \end{pmatrix} \tag{163}$$

$$= \begin{pmatrix} 5 \\ 6 \end{pmatrix} \tag{164}$$

So our result was correct!

Answer 12. *We can write out y_i index notation,*

$$y_i = \sum_{j=1}^{N} X_{ij} w_j \tag{165}$$

Then subsitute this expression for y_i into the gradient we're trying to compute,

$$\frac{dy_i}{dw_k} = \frac{d}{dw_k}[\sum_{j=1}^{N} X_{ij} w_j] \tag{166}$$

As X_{ij} is constant,

$$\frac{dy_i}{dw_k} = \sum_{j=1}^{N} X_{ij} \frac{dw_j}{dw_k} \tag{167}$$

The gradient is one when $j = k$, and zero otherwise, which matches the definition of the Kronecker delta,

$$\frac{dy_i}{dw_k} = \sum_{j=1}^{N} X_{ij} \delta_{jk}. \tag{168}$$

Remember that the Kronecker delta is 1 when $j = k$ and zero otherwise, so the Kronecker delta picks out the $j = k$ element of the loop,

$$\frac{dy_i}{dw_k} = X_{ik}. \tag{169}$$