Notes on Artificial Intelligence

Richard Kelley

September 29, 2015

1 Linear Algebra

1.1 Vectors: Basics

We'll use the notation $x \in \mathbb{R}^n$ to indicate that x is a vector with n real-valued components. I'm going to assume that you know the basics of vectors from a previous course - maybe Calculus or Linear Algebra. It's probably worth emphasizing that vectors are, unless otherwise noted, *column vectors*. The only thing that makes this tricky is that writing vectors as column vectors takes more space:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{1}$$

so we will often write our vectors inline and expect the reader to know what we're talking about: $x = [x_1, \ldots, x_n]$. In practice this is almost never an issue.

Arithmetic does what you'd expect:

$$x + y = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix}, \tag{2}$$

as does multiplication by a scalar:

$$cx = c \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} cx_1 \\ \vdots \\ cx_n \end{bmatrix}. \tag{3}$$

1.2 Inner Products & Norms

The "inner product" between two vectors $x, y \in \mathbb{R}^n$ is defined in the usual way:

$$x \cdot y = x^T y = \sum_{1 \le i \le n} x_i y_i \tag{4}$$

Later on, it will be useful to remember that this can be written:

$$x^T y = [x_1 \cdots x_n] \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{5}$$

We can define the "size" of a vector in terms of several different norms:

1. The *1-norm*: For $x \in \mathbb{R}^n$,
$$\|x\|_1 = \left(\sum_{1 \le i \le n} |x_i| \right)$$

2. The *2-norm*: For $x \in \mathbb{R}^n$,
$$\|x\|_2 = \left(\sum_{1 \le i \le n} |x_i|^2 \right)^{1/2}$$

3. The *p-norm*: For $x \in \mathbb{R}^n$ and $p \in \mathbb{N}$, $p \ge 1$,
$$\|x\|_p = \left(\sum_{1 \le i \le n} |x_i|^p \right)^{1/p}$$

4. The *∞-norm*: For $x \in \mathbb{R}^n$,
$$\|x\|_\infty = \max_i(|x_1|, |x_2|, \dots, |x_n|)$$

(For the curious: yes, you do get the ∞-norm from the p-norm if you let $p \to \infty$.)

We will care quite a bit about the 1-norm and the 2-norm when we talk about something called *regularization* in machine learning. For the most part we'll focus on the 2-norm, and if you ever see the norm symbol without a subscript, assume that we're talking about the 2-norm unless otherwise noted.

With the inner product and the norm defined, we can define the *angle* between two vectors in $\mathbb{R}^n$: suppose $x, y \in \mathbb{R}^n$. Then the angle between x and y is defined to be the number θ in the range $0 \le \theta \le \pi$ satisfying the equation
$$\cos\theta = \frac{x \cdot y}{\|x\|\|y\|}.$$
This definition works for any n you would care about, and for $n = 2$ does exactly what you expect from geometry.

1.3 Matrices

Let's look at matrices. The notation we'll use is exactly what you might expect from previous courses. We'll typically use uppercase letters like A to denote a matrix. We'll say that a matrix with m rows and n columns has dimensions $m \times n$, and we'll denote the components of a matrix in the usual way: the element of the matrix in the i-th row and j-th column is denoted by a_{ij}. If a matrix A has dimensions $m \times n$, I will sometimes indicate this type with the notation $A \in \mathbb{R}^{m \times n}$. If we have to write out the matrix, we'll probably do something like this:
$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}. \tag{6}$$

We will sometimes want to refer to the i, j-th entry of a matrix expression. We'll do this by wrapping the expression in parentheses and writing the indices as subscripts. So the i, j-th entry of a matrix A can be referred to as $(A)_{ij} = a_{ij}$. This comes in handy when the expression gets more complicated than a single matrix. For instance, we can define "matrix addition" and "scalar multiplication" very easily as
$$(A + B)_{ij} = a_{ij} + b_{ij} \tag{7}$$
and
$$(cA)_{ij} = ca_{ij}. \tag{8}$$

We will occasionally want to refer to a particular row or column of a matrix A. We will do this with notation that is similar to what you would see in Numpy: The i-th row of A will be denoted by $A_{i,:}$ and the j-th column of A will be denoted by $A_{:,j}$.

After the basic arithetic operations, the next thing we'll want to remember is *matrix-vector multiplication.* Given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $x \in \mathbb{R}^n$, we define the product Ax as follows:

$$y = Ax = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{1\le j\le n} a_{1j}x_j \\ \vdots \\ \sum_{1\le j\le n} a_{mj}x_j \end{bmatrix}, \tag{9}$$

So that the typical element of the vector y is

$$y_i = \sum_{1\le j\le n} a_{ij}x_j.$$

An important thing to note is that the operation of matrix-vector multiplication is only defined when the dimensions of A and x match up: if $x \in \mathbb{R}^n$, then the number of columns of A *must* be n. The number of rows can be any positive integer.

While the product of a matrix and a vector is *essential* for making any sense at all out of machine learning, it will turn out that efficient implementation of machine learning will hinge on the related operation of *matrix-matrix* multiplication. If $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{p\times n}$, then the product AB is an element of $\mathbb{R}^{m\times n}$ whose components are defined by the following equation:

$$(C)_{ij} = (AB)_{ij} = \sum_{1\le k\le p} a_{ik}b_{kj}. \tag{10}$$

You can either memorize this sum, or you can just remember that the i, j-th entry of the product AB is the inner product of the i-th row of A with the j-th column of B:

$$(C)_{ij} = A_{i,:} \cdot B_{:,j} \tag{11}$$

I've been using the operation implicitly now for a while, but for the sake of completeness you should remember that the *transpose* of a matrix A is denoted by A^T. The i, j-th entry of A^T is the j, i-th entry of A. We'll say that a matrix is *symmetric* if $A^T = A$.

We will once or twice need to use the idea of a *quadratic form.* Given a vector $x \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{n\times n}$, the expression x^TAx is called a quadradic form. The reason for that name is clear if you write everything out in terms of components:

$$x^TAx = \sum_{1\le i\le n}\sum_{1\le j\le n} a_{ij}x_ix_j. \tag{12}$$

2 Calculus

There are a lot of ways to look at Calculus, but the particular viewpoint we'll adopt is that Calculus is all about approximating "complicated things" by "simple things." For us, the word "complicated" will (usually) mean *nonlinear* and the word "simple" will (always) mean *linear.* We'll start with derivatives.

2.1 Derivatives

Suppose we have a function f that takes a real number as its input and yields a real number as its output. For example, we may have $f(x) = 2 + x$. To concisely describe the "type" of f, we'll use the notation

$f\colon \mathbb{R} \to \mathbb{R}$. That is: the name of the function, then a colon, then the type of function inputs, then an arrow, then the type of the function's outputs. This is a common notation in math, and we'll use it extensively.

Given a function $f\colon \mathbb{R} \to \mathbb{R}$, we'll use the definition of *derivative* that you should have seen in Calculus. We'll denote the derivative of f at a point x as $f'(x)$, and that will be given by

$$f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}. \tag{13}$$

The intuition is that ϵ is a very tiny number. When f is differentiable, it turns out that by rearranging the definition of derivative we can obtain

$$f(x+\epsilon) \approx f(x) + \epsilon \cdot f'(x), \tag{14}$$

which is just a way that the linear function of ϵ defined by $f(x)+\epsilon \cdot f'(x)$ is a *very* good approximation to f if we're close enough to x. (In fact, more advanced Calculus courses typically *define* the idea of differentiability using a definition like this.)

Remember: differentiability is all about approximating something complicated (a possibly nonlinear function) with something simple (a linear function). The thing that makes differentiability "special" is that there is no limit to how good we can make this approximation at a point where a function is differentiable.

2.2 Partial Derivatives

In the real world, we're never so lucky as to work with functions of just one variable. It's much more common to deal with a lot of variables, and in AI it's currently routine to work with functions of several million variables. Some of these functions will be very complicated, and so to work with them we will want to build very good approximations. To do this, we need some generalization of derivatives that works for a function of n variables, no matter how large n is.

In Multivariable Calculus, you saw the *partial derivative*, which is one of the key components necessary to make sense of complicated functions. Suppose $f\colon \mathbb{R}^n \to \mathbb{R}$, and suppose that we're looking at a point $x = [x_1, \ldots, x_n]$. Since there are n inputs to f, there are n *partial derivatives.* Each such partial derivative tells you, roughly, how much f changes when you vary *exactly* one of the inputs to the function. We'll use a few different schemes to denote the partial derivative of f with respect to one of its variables x_i: $\frac{\partial f}{\partial x_i}$, $\partial_{x_i} f(x)$, or even just $\partial_i f(x)$. Each of these notations is convenient in one or more distinct settings, but each is also entirely equivalent to the others. So, with this notation, we can define the partial derivative $\partial f_i(x)$ as

$$\frac{\partial f}{\partial x_i} = \lim_{\epsilon \to 0} \frac{f(x_1, \ldots, x_i + \epsilon, \ldots, x_n) - f(x_1, \ldots, x_n)}{\epsilon}. \tag{15}$$

Remember, the key idea is that we change *one variable at a time.*

As an example, consider the function $f(x, y) = 2x + 3y$. Convince yourself that $\partial_x f(x, y) = 2$ and $\partial_y f(x, y) = 3$.

2.3 Gradients

The above description of partial derivatives should give you a rough idea of what partial derivatives *are*, but seems pretty removed from the idea of "approximating nonlinear functions by linear ones." It turns out that partial derivatives are essential to the idea of linear approximation. Suppose that $x, h \in \mathbb{R}^n$, and suppose that $f\colon \mathbb{R}^n \to \mathbb{R}$. What would it mean to say that there's a linear function that is a very good approximation to f at x? It would mean that there's some *vector* $c \in \mathbb{R}^n$ for which the following is true:

$$f(x+h) \approx f(x) + h \cdot c. \tag{16}$$

The trick, of course, is to find just the right c. I'll spare you the suspense: the right c is called the *gradient*, and it's defined using partial derivatives:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}, \tag{17}$$

so that the "really good" linear approximation we want to keep in mind is

$$f(x+h) \approx f(x) + h \cdot \nabla f(x). \tag{18}$$

2.4 Directional Derivatives

This part is *very* important for machine learning. The partial derivatives tell us what happens when we change one variable at a time. A very natural thing to do is change multiple variables at once to see how our function changes. The idea of a *directional derivative* captures this idea of a changing a function in an arbitrary direction to see how much the function changes.

Suppose that $x \in \mathbb{R}^n$, and suppose that u is a *unit vector* in $\mathbb{R}^n$. That means that $u \in \mathbb{R}^n$ and $\|u\| = 1$. The fact that u is a unit vector is important. Two unit vectors have (by definition) the same magnitude, so they can only be different if they point in different directions. So unit vectors are a very natural way to talk about the idea of direction in $\mathbb{R}^n$. We'll define the directional derivative of f at the point x in the direction of u by

$$D_u f(x) = \lim_{h \to 0} \frac{f(x+hu) - f(x)}{h}. \tag{19}$$

Notice that $h \in \mathbb{R}$. Unpacking what this definition means would take us really far from where we want to go, so instead of memorizing the definition I would say it's better to remember the following theorem:

$$D_u f(x) = \nabla f(x) \cdot u, \tag{20}$$

Which says that you can find the derivative of f at x in the direction of u by first calculating $\nabla f(x)$ and then taking the inner product of ∇f with u.

Importantly, this implies that the direction θ in which f increases the most at x is the direction that is parallel to ∇f: for any u we have $D_u f(x) = \nabla f(x) \cdot u = \|\nabla f(x)\| \|u\| \cos\theta = \|\nabla f(x)\| \cos\theta$. Since $\|\nabla f(x)\| \geq 0$ and $\theta \in [0, \pi]$, you should be able to convince yourself that $D_u f(x)$ is largest for $\theta = 0$. Also important is that the direction of greatest *decrease* is when $\theta = \pi$, so that u points in the opposite direction of $\nabla f(x)$. This paragraph will be crucial for our understanding of learning. Make sure you're clear on what's happening here.

2.5 Optimal Values of Functions

In single-variable Calculus, you should remember that if $x^\star$ is a *local optimum* (which means either a maximum or a minimum), then $f'(x^\star) = 0$. Pretty much the same thing is true for a function of n variables: if $f \colon \mathbb{R}^n \to R$ and $x^\star \in \mathbb{R}^n$ is a local optimum for f, then

$$\nabla f(x^\star) = 0, \tag{21}$$

so that each of the partial derivatives of f is zero at $x^\star$.

We won't spend too much time worrying if the optimum we've found is a local maximum or a local minimum. There are tests involving higher-order derivatives, but that's more complexity than we need to worry about at this level.

2.6 Derivatives With Respect to Vectors

Up to this point we've talked about the derivative of a scalar function with respect to a scalar variable ($f\colon \mathbb{R} \to \mathbb{R}$) and we've talked about derivatives of scalar functions with respect to vector arguments ($f\colon \mathbb{R}^n \to \mathbb{R}$). We have seen that the *gradient* of f will give us all the derivatives that we need for a function from $\mathbb{R}^n \to \mathbb{R}$. If f is such a function and $y = f(x)$ for some $x \in \mathbb{R}^n$, then another notation for the gradient is

$$\nabla f(x) = \frac{\partial y}{\partial x} = [\partial_{x_1} y, \ldots, \partial_{x_n} y] \tag{22}$$

So far nothing new here, except a slightly different notation. The twist comes when we make the next generalization: we want to define the "derivative" of a function of type $f\colon \mathbb{R}^n \to \mathbb{R}^m$. This is a function that produces m outputs, each of which can depend on any of n inputs. As an example, you might consider the function $f\colon \mathbb{R}^4 \to \mathbb{R}^2$ defined by the equation

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1^2 + 2x_2 + 5x_3x_4 \\ x_2 + x_3 + x_4 \end{bmatrix}. \tag{23}$$

The trick for us here is to define what we mean by the derivative of f. If you think about it, you should be able to convince yourself that since each of the m outputs can depend on any of the n inputs, we need an object that tracks each of the mn relationships. Suppose $f\colon \mathbb{R}^n \to \mathbb{R}^m$, and suppose that for $x \in \mathbb{R}^n$ we have $y = f(x)$. Then

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{24}$$

or, more concisely,

$$\left(\frac{\partial y}{\partial x}\right)_{ij} = \frac{\partial y_i}{\partial x_j}. \tag{25}$$

Note: In some parts of the world, there's another convention, which is that the derivative $\frac{\partial y}{\partial x}$ is the transpose of what I've written above. You'll have to figure out from context which layout convention is used. We'll stick to the above convention for this course.

2.6.1 Examples

Here are some example calculations for you to see how things work in practice.

1. **Derivative of an inner product.** First, suppose that $y = c^T x$, where c is a constant vector. What is $\frac{\partial y}{\partial x}$? Since we are taking the derivative of a scalar with respect to a vector, we know that $\frac{\partial y}{\partial x}$ has to be a vector. If we look at $y = c^T x$ at the level of scalars, we know that

   $$y = \sum_{1 \le i \le n} c_i x_i,$$

   and if we take the derivative of this expression with respect to x_i, then we see that

   $$\frac{\partial y}{\partial x_i} = c_i,$$

   leading us to conclude that if $y = c^T x$ then

   $$\frac{\partial y}{\partial x} = c. \tag{26}$$

2. **Derivative of a matrix-vector product.** Next let's try something a bit more complicated. Suppose that $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$. Suppose we define $y = Ax$. Then what is $\frac{\partial y}{\partial x}$? We know from the definition of A that $y \in \mathbb{R}^m$. Moreover, we know that the typical element of y has this form:

$$y_i = \sum_{1 \le j \le n} a_{ij} x_j.$$

   Since we are calculating the derivative of a vector quantity with respect to a vector quantity, we know that $\frac{\partial y}{\partial x}$ must be a matrix, and that the typical element of that matrix is $\frac{\partial y_i}{\partial x_j}$. From the expression for y_i above, you should be able to convince yourself that

$$\frac{\partial y_i}{\partial x_j} = a_{ij}, \tag{27}$$

   so that $\frac{\partial y}{\partial x} = A$.

3. **Derivative of a Quadratic Form.** Call one this an "exercise left for the reader." Suppose that $y = x^T Ax$, where $x \in \mathbb{R}^n$. Show that $\frac{\partial y}{\partial x} = x^T(A + A^T)$.

2.7 The Chain Rule

By now you should be pretty comfortable with the idea of calculating the derivative $\frac{\partial y}{\partial x}$ whenever y depends directly on x through some function f. However, we'll find it *very* useful to be able to calculate the derivative $\frac{\partial L}{\partial x}$ when the quantity L depends on x only *indirectly*, through some other object y. We'll see that the way to perform this calculation is to invoke the *Chain Rule*, which we review for one variable before walking through the multivariable case.

Suppose that $z = f(g(x))$. We'll want to know $\frac{dz}{dx}$. From your single-variable Calculus class, you may remember that we can define $y = g(x)$ and write the desired derivative as:

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}. \tag{28}$$

We can generalize this to the multivariable case fairly easily. Suppose that $f \colon \mathbb{R}^n \to \mathbb{R}$, that $y = f(x_1, \ldots, x_n)$, and that each $x_i = g_i(s,t)$. What is $\frac{\partial y}{\partial s}$? This situation is much more complicated, but if we trace all the ways that s can affect y, then it's not hard to figure out the derivative. We can start by drawing out the dependency diagram for this situation:

TODO: create attractive diagram.

By examining the diagram above, we can see all the ways that a change in s can lead to a change in y. Graphically, we can trace each change by finding all of the paths from y to s. Each such path with travel from y to one of the x_i, and then from that x_i to s. This observation leads to a simple algorithm for writing down the derivative of y with respect to s. That derivative is a sum of terms. Each term corresponds to a path from y to s. To find the term in the sum that corresponds to the path from y to x_i to s, we write down the product $\frac{\partial y}{\partial x_i}\frac{\partial x_i}{\partial s}$. So in our current case we can write down the derivative as

$$\begin{aligned}\frac{\partial y}{\partial s} &= \frac{\partial y}{\partial x_1}\frac{\partial x_1}{\partial s} + \cdots + \frac{\partial y}{\partial x_n}\frac{\partial x_n}{\partial s} &\quad (29)\\ &= \sum_{1 \le i \le n} \frac{\partial y}{\partial x_i}\frac{\partial x_i}{\partial s}. &\quad (30)\end{aligned}$$

By following all paths from the output to the desired input, writing down a product of partial derivatives as we go and adding up all the products at the end, we can find the derivative of our output with respect to our input regardless of the complexity of the intervening chain of variables.

2.7.1 The Chain Rule for Linear Functions

Let's look at $y = Ax$ again. Suppose that $y \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$. Suppose further that $L \in \mathbb{R}$, and that we know the derivative of L with respect to y:

$$\frac{\partial L}{\partial y}.$$

There are two derivatives that we will want to calculate, corresponding to the two parts of the linear function:

$$\frac{\partial L}{\partial x} \text{ and } \frac{\partial L}{\partial A}.$$

We'll use the chain rule as described above to find these derivatives.

TODO: create attractive diagram.

1. **The derivative of L with respect to x.** By following the dependencies from L to x_i above, we can see that

$$\frac{\partial L}{\partial x_i} = \sum_{1 \le k \le m} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial x_i}. \tag{31}$$

We know (by assumption) the value of $\frac{\partial L}{\partial y_k}$ for each possible k, so to be able to calculate $\frac{\partial L}{\partial x_i}$ we just need to figure out $\frac{\partial y_k}{\partial x_i}$. We know by definition that

$$y_k = \sum_{1 \le j \le n} a_{kj} x_j,$$

so we have

$$\begin{aligned} \frac{\partial y_k}{\partial x_i} &= \frac{\partial}{\partial x_i} \sum_{1 \le j \le n} a_{kj} x_j && (32) \\ &= \sum_{1 \le j \le n} \frac{\partial}{\partial x_i} a_{kj} x_j && (33) \\ &= a_{ki}, && (34) \end{aligned}$$

from which we conclude that

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= \sum_{1 \le k \le m} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial x_i} && (35) \\ &= \sum_{1 \le k \le m} a_{ki} \frac{\partial L}{\partial y_k}. && (36) \end{aligned}$$

You should be able to convince yourself that in matrix-vector notation this is the same as

$$\frac{\partial L}{\partial x} = A^T \frac{\partial L}{\partial y}, \tag{37}$$

which is the solution to our first problem, in a form that is amenable to (relatively) efficient implementation on a computer.

2. **The derivative of L with respect to A.** We'll find the derivative of L with respect to an element a_{ij}. Start by drawing out the dependencies again, focusing on the relationship between L and a_{ij} this time. From the diagram, you should be able to see that

$$\frac{\partial L}{\partial a_{ij}} = \sum_{1 \le k \le m} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial a_{ij}}. \tag{38}$$

Moreover, we know (again, by assumption) $\frac{\partial L}{\partial y_k}$. So all we have to do is figure out $\frac{\partial y_k}{\partial a_{ij}}$. To do this, let's write out a typical y_k:

$$y_k = \sum_{1 \leq \ell \leq n} a_{k\ell} x_\ell.$$

The first thing to observe that if $i \neq k$, then $\frac{\partial y_k}{\partial a_{ij}}$ must be 0. So we only have to worry about the case where $i = k$. In this case, the partial derivative is just x_j. So we can write down the derivative

$$\frac{\partial L}{\partial a_{ij}} = \sum_{1 \leq k \leq m} \frac{\partial L}{\partial y_k} \frac{\partial y_k}{\partial a_{ij}} \tag{39}$$

$$= \frac{\partial L}{\partial y_i} x_j, \tag{40}$$

where the second equality follows from the fact that when $i \neq k$ then $\frac{\partial y_k}{\partial a_{ij}} = 0$, so the sum actually has only one term. We have our answer, but we can simplify it a bit more by noticing that it is an outer product, leading to the following matrix equation:

$$\frac{\partial L}{\partial A} = \frac{\partial L}{\partial y} x^T. \tag{41}$$

2.7.2 Summary For Matrix-Vector Product

So to summarize, we have the following results for a linear function $y = Ax$:

1. Given an input x, we can calculate the output y via the equation $y = Ax$.
2. Given $\frac{\partial L}{\partial y}$, we can calculate $\frac{\partial L}{\partial x}$ via the equation

$$\frac{\partial L}{\partial x} = A^T \frac{\partial L}{\partial y}.$$

3. Given $\frac{\partial L}{\partial y}$, we can calculate $\frac{\partial L}{\partial A}$ via the equation

$$\frac{\partial L}{\partial A} = \frac{\partial L}{\partial y} x^T.$$

With these three equations, you are now in a position to implement a large percentage of the machine learning algorithms currently in use.