When working with neural networks, we often take derivatives of various objects (scalars, vectors, matrices, tensors) with respect to various other objects. In this note we will walk through the various flavors of derivatives that we commonly work with in the context of deep learning.

Essentially, these notes attempt to answer the question: what exactly does a derivative \( \frac{\partial y}{\partial x} \) mean when \( y \) and \( x \) are tensors with arbitrary shapes? How can we generalize the chain rule to compute derivatives of functions which are compositions of other tensor-valued functions?

1 Scalar Case

You are probably familiar with the concept of a derivative in the scalar case: given a function \( f : \mathbb{R} \rightarrow \mathbb{R} \), the derivative of \( f \) at a point \( x \in \mathbb{R} \) is defined as:

\[
f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}
\]

Derivatives are a way to measure change. In the scalar case, the derivative of the function \( f \) at the point \( x \) tells us how much the function \( f \) changes as the input \( x \) changes by a small amount \( \varepsilon \): 

\[
f(x + \varepsilon) \approx f(x) + \varepsilon f'(x)
\]

For ease of notation we will commonly assign a name to the output of \( f \), say \( y = f(x) \), and write \( \frac{\partial y}{\partial x} \) for the derivative of \( y \) with respect to \( x \). This notation emphasizes that \( \frac{\partial y}{\partial x} \) is the rate of change between the variables \( x \) and \( y \); concretely if \( x \) were to change by \( \varepsilon \) then \( y \) will change by approximately \( \varepsilon \frac{\partial y}{\partial x} \). We can write this relationship as

\[
x \to x + \Delta x \implies y \to y + \frac{\partial y}{\partial x} \Delta x
\]

You should read this as saying “changing \( x \) to \( x + \Delta x \) implies that \( y \) will change to approximately \( y + \Delta x \frac{\partial y}{\partial x} \). This notation is nonstandard, but I like it since it emphasizes the relationship between changes in \( x \) and changes in \( y \).
The chain rule tells us how to compute the derivative of the composition of functions. In the scalar case suppose that \( f, g : \mathbb{R} \to \mathbb{R} \) and \( y = f(x), z = g(y) \); then we can also write \( z = (g \circ f)(x) \), or draw the following computational graph:

\[
x \xrightarrow{f} y \xrightarrow{g} z
\]

The (scalar) chain rule tells us that

\[
\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}
\]

This equation makes intuitive sense. The derivatives \( \frac{\partial z}{\partial y} \) and \( \frac{\partial y}{\partial x} \) give:

\[
\begin{align*}
  x \to x + \Delta x & \implies y \to y + \frac{\partial y}{\partial x} \Delta x \\
  y \to y + \Delta y & \implies z \to z + \frac{\partial z}{\partial y} \Delta y
\end{align*}
\]

Combining these two rules lets us compute the effect of \( x \) on \( z \): if \( x \) changes by \( \Delta x \) then \( y \) will change by \( \frac{\partial y}{\partial x} \Delta x \), so we have \( \Delta y = \frac{\partial y}{\partial x} \Delta x \). If \( y \) changes by \( \Delta y \) then \( z \) will change by \( \frac{\partial z}{\partial y} \Delta y = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x} \Delta x \) which is exactly what the chain rule tells us.

2 Gradient: Vector in, scalar out

This same intuition carries over into the vector case. Now suppose that \( f : \mathbb{R}^N \to \mathbb{R} \) takes a vector as input and produces a scalar. The derivative of \( f \) at the point \( x \in \mathbb{R}^N \) is now called the gradient, and it is defined as:

\[
\nabla_x f(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{\|h\|}
\]

Now the gradient \( \nabla_x f(x) \in \mathbb{R}^N \) is a vector, with the same intuition as the scalar case. If we set \( y = f(x) \) then we have the relationship

\[
\begin{align*}
  x \to x + \Delta x & \implies y \to y + \frac{\partial y}{\partial x} \cdot \Delta x
\end{align*}
\]

The formula changes a bit from the scalar case to account for the fact that \( x, \Delta x, \) and \( \frac{\partial y}{\partial x} \) are now vectors in \( \mathbb{R}^N \) while \( y \) is a scalar. In particular when multiplying \( \frac{\partial y}{\partial x} \) by \( \Delta x \) we use the dot product, which combines two vectors to give a scalar.

One nice outcome of this formula is that it gives meaning to the individual elements of the gradient \( \frac{\partial y}{\partial x} \). Suppose that \( \Delta x \) is the \( i \)th basis vector, so that the \( i \)th coordinate of \( \varepsilon \) is 1 and all other coordinates of \( \varepsilon \) are 0. Then the dot product \( \frac{\partial y}{\partial x} \cdot \Delta x \) is simply the \( i \)th coordinate of \( \frac{\partial y}{\partial x} \); thus the \( i \)th coordinate of \( \frac{\partial y}{\partial x} \) tells us the approximate amount by which \( y \) will change if we move \( x \) along the \( i \)th coordinate axis.
This means that we can also view the gradient $\frac{\partial y}{\partial x}$ as a vector of partial derivatives:

$$\frac{\partial y}{\partial x} = \left( \frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_N} \right)$$

where $x_i$ is the $i$th coordinate of the vector $x$, which is a scalar, so each partial derivative $\frac{\partial y}{\partial x_i}$ is also a scalar.

### 3 Jacobian: Vector in, Vector out

Now suppose that $f : \mathbb{R}^N \rightarrow \mathbb{R}^M$ takes a vector as input and produces a vector as output. Then the derivative of $f$ at a point $x$, also called the Jacobian, is the $M \times N$ matrix of partial derivatives. If we again set $y = f(x)$ then we can write:

$$\frac{\partial y}{\partial x} = \begin{pmatrix} \frac{\partial y}{\partial x_1} & \cdots & \frac{\partial y}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_M}{\partial x_1} & \cdots & \frac{\partial y_M}{\partial x_N} \end{pmatrix}$$

The Jacobian tells us the relationship between each element of $x$ and each element of $y$: the $(i, j)$-th element of $\frac{\partial y}{\partial x}$ is equal to $\frac{\partial y_i}{\partial x_j}$, so it tells us the amount by which $y_i$ will change if $x_j$ is changed by a small amount.

Just as in the previous cases, the Jacobian tells us the relationship between changes in the input and changes in the output:

$$x \rightarrow x + \Delta x \implies y \rightarrow y + \frac{\partial y}{\partial x} \Delta x$$

Here $\frac{\partial y}{\partial x}$ is a $M \times N$ matrix and $\Delta x$ is an $N$-dimensional vector, so the product $\frac{\partial y}{\partial x} \Delta x$ is a matrix-vector multiplication resulting in an $M$-dimensional vector.

The chain rule can be extended to the vector case using Jacobian matrices. Suppose that $f : \mathbb{R}^N \rightarrow \mathbb{R}^M$ and $g : \mathbb{R}^M \rightarrow \mathbb{R}^K$. Let $x \in \mathbb{R}^N$, $y \in \mathbb{R}^M$, and $z \in \mathbb{R}^K$ with $y = f(x)$ and $z = g(y)$, so we have the same computational graph as the scalar case:

$$x \xrightarrow{f} y \xrightarrow{g} z$$

The chain rule also has the same form as the scalar case:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$$

However now each of these terms is a matrix: $\frac{\partial z}{\partial y}$ is a $K \times M$ matrix, $\frac{\partial y}{\partial x}$ is a $M \times N$ matrix, and $\frac{\partial z}{\partial x}$ is a $K \times N$ matrix; the multiplication of $\frac{\partial z}{\partial y}$ and $\frac{\partial y}{\partial x}$ is matrix multiplication.
4 Generalized Jacobian: Tensor in, Tensor out

Just as a vector is a one-dimensional list of numbers and a matrix is a two-dimensional grid of numbers, a tensor is a \( D \)-dimensional grid of numbers\(^1\).

Many operations in deep learning accept tensors as inputs and produce tensors as outputs. For example an image is usually represented as a three-dimensional grid of numbers, where the three dimensions correspond to the height, width, and color channels (red, green, blue) of the image. We must therefore develop a derivative that is compatible with functions operating on general tensors.

Suppose now that \( f : \mathbb{R}^{N_1 \times \cdots \times N_{D_x}} \times \mathbb{R}^{M_1 \times \cdots \times M_{D_y}} \). Then the input to \( f \) is a \( D_x \)-dimensional tensor of shape \( N_1 \times \cdots \times N_{D_x} \), and the output of \( f \) is a \( D_y \)-dimensional tensor of shape \( M_1 \times \cdots \times M_{D_y} \). If \( y = f(x) \) then the derivative \( \frac{\partial y}{\partial x} \) is a \textit{generalized Jacobian}, which is an object with shape

\[
(M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x})
\]

Note that we have separated the dimensions of \( \frac{\partial y}{\partial x} \) into two groups: the first group matches the dimensions of \( y \) and the second group matches the dimensions of \( x \). With this grouping, we can think of the generalized Jacobian as generalization of a matrix, where each “row” has the same shape as \( y \) and each “column” has the same shape as \( x \).

Now if we let \( i \in \mathbb{Z}^{D_y} \) and \( j \in \mathbb{Z}^{D_x} \) be vectors of integer indices, then we can write

\[
\left( \frac{\partial y}{\partial x} \right)_{i,j} = \frac{\partial y_i}{\partial x_j}
\]

In this equation note that \( y_i \) and \( x_j \) are scalars, so the derivative \( \frac{\partial y_i}{\partial x_j} \) is also a scalar. Using this notation we see that like the standard Jacobian, the generalized Jacobian tells us the relative rates of change between all elements of \( x \) and all elements of \( y \).

The generalized Jacobian gives the same relationship between inputs and outputs as before:

\[
x \rightarrow x + \Delta x \implies y \rightarrow y + \frac{\partial y}{\partial x} \Delta x
\]

The difference is that now \( \Delta x \) is a tensor of shape \( N_1 \times \cdots \times N_{D_x} \) and \( \frac{\partial y}{\partial x} \) is a generalized matrix of shape \( (M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x}) \). The product \( \frac{\partial y}{\partial x} \Delta x \) is therefore a \textit{generalized matrix-vector multiply}, which results in a tensor of shape \( M_1 \times \cdots \times M_{D_y} \).

\(^1\)The word \textit{tensor} is used in different ways in different fields; you may have seen the term before in physics or abstract algebra. The machine learning definition of a tensor as a \( D \)-dimensional grid of numbers is closely related to the definitions of tensors in these other fields.
The generalized matrix-vector multiply follows the same algebraic rules as a traditional matrix-vector multiply:

\[
\left( \frac{\partial y}{\partial x} \Delta x \right)_j = \sum_i \left( \frac{\partial y}{\partial x} \right)_{i,j} (\Delta x)_i = \left( \frac{\partial y}{\partial x} \right)_{j,:} \cdot \Delta x
\]

The only difference is that the indices \(i\) and \(j\) are not scalars; instead they are vectors of indices. In the equation above the term \(\left( \frac{\partial y}{\partial x} \right)_{j,:}\) is the \(j\)th “row” of the generalized matrix \(\frac{\partial y}{\partial x}\), which is a tensor with the same shape as \(x\). We have also used the convention that the dot product between two tensors of the same shape is an elementwise product followed by a sum, identical to the dot product between vectors.

The chain rule also looks the same in the case of tensor-valued functions. Suppose that \(y = f(x)\) and \(z = g(y)\), where \(x\) and \(y\) have the same shapes as above and \(z\) has shape \(K_1 \times \cdots \times K_{D_z}\). Now the chain rule looks the same as before:

\[
\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}
\]

The difference is that now \(\frac{\partial z}{\partial y}\) is a generalized matrix of shape \((K_1 \times \cdots \times K_{D_z}) \times (M_1 \times \cdots \times M_{D_y})\), and \(\frac{\partial y}{\partial x}\) is a generalized matrix of shape \((M_1 \times \cdots \times M_{D_y}) \times (N_1 \times \cdots \times N_{D_x})\); the product \(\frac{\partial z}{\partial y} \frac{\partial y}{\partial x}\) is a generalized matrix-matrix multiply, resulting in an object of shape \((K_1 \times \cdots \times K_{D_z}) \times (N_1 \times \cdots \times N_{D_x})\). Like the generalized matrix-vector multiply defined above, the generalized matrix-matrix multiply follows the same algebraic rules as the traditional matrix-matrix multiply:

\[
\left( \frac{\partial z}{\partial x} \right)_{i,j} = \sum_k \left( \frac{\partial z}{\partial y} \right)_{k,i} \left( \frac{\partial y}{\partial x} \right)_{j,i} = \left( \frac{\partial z}{\partial y} \right)_{i,:k} \cdot \left( \frac{\partial y}{\partial x} \right)_{j,:}
\]

In this equation the indices \(i, j, k\) are vectors of indices, and the terms \(\left( \frac{\partial z}{\partial y} \right)_{i,:k}\) and \(\left( \frac{\partial y}{\partial x} \right)_{j,:}\) are the \(k\)th “column” of \(\frac{\partial z}{\partial y}\) and the \(j\)th “row” of \(\frac{\partial y}{\partial x}\) respectively.