Differentiating Vector- and Matrix-Valued Functions

Szymon Rusinkiewicz
COS 302, Fall 2020
Generalizing Functions…

Functions of scalars, vectors, matrices … *returning* scalars, vectors, matrices

- Function of a scalar, returning a scalar: $\mathbb{R} \rightarrow \mathbb{R}$
  - Example: $f(x) = ax + b$

- Function of a scalar, returning a vector: $\mathbb{R} \rightarrow \mathbb{R}^n$
  - Example: $f(x) = xv$

- Function of a vector, returning a scalar: $\mathbb{R}^n \rightarrow \mathbb{R}$
  - Example: $f(x) = v^\top x$

- Function of a vector, returning a vector: $\mathbb{R}^n \rightarrow \mathbb{R}^n$
  - Example: $f(x) = Mx$

- Function of a vector, returning a matrix: $\mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$
  - Example: $F(x) = xx^\top$

- Many other possibilities: function of a matrix, etc.
Generalizing Functions... and Taking Their Derivatives

- Function of a scalar, returning a scalar: $\mathbb{R} \rightarrow \mathbb{R}$
  - Example: $f(x) = ax + b$  \rightarrow  Ordinary derivative $\frac{df}{dx} : \mathbb{R} \rightarrow \mathbb{R}$

- Function of a scalar, returning a vector: $\mathbb{R} \rightarrow \mathbb{R}^n$
  - Example: $f(x) = xv$  \rightarrow  Vector-valued derivative $\frac{df}{dx} : \mathbb{R} \rightarrow \mathbb{R}^n$

- Function of a vector, returning a scalar: $\mathbb{R}^n \rightarrow \mathbb{R}$
  - Example: $f(x) = v^T x$  \rightarrow  Gradient $\nabla f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{1 \times n}$

- Function of a vector, returning a vector: $\mathbb{R}^n \rightarrow \mathbb{R}^n$
  - Example: $f(x) = Mx$  \rightarrow  Jacobian $\nabla f(x)$ or $J(f(x)) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$

- Function of a vector, returning a matrix: $\mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$
  - Example: $F(x) = xx^T$  \rightarrow  Generalized Jacobian $\nabla F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n \times n}$
Generalizing Functions... and Taking Their Derivatives

In general, if $f$ is a function

$$f : \mathbb{R}^{(input\ shape)} \rightarrow \mathbb{R}^{(output\ shape)}$$

then its generalized derivative will be a function

$$\nabla f : \mathbb{R}^{(input\ shape)} \rightarrow \mathbb{R}^{(output\ shape)\times(input\ shape)}$$

where the extra dimensions on output correspond to taking partial derivatives with respect to all the input dimensions.
So what if we end up with e.g. an $n \times n \times n$ object?

- **Tensors** are multidimensional generalizations of scalars, vectors, matrices.
- For our purposes, represented as multidimensional arrays of numbers.
- The number of indices in the shape can be called the *order*, *degree*, (confusingly) *dimension*, or (even more confusingly) *rank* of the tensor.
  - Example: Take a function of a vector, returning a matrix, and differentiate it. The resulting $n \times n \times n$ beastie is a degree-3 or 3rd-order tensor.
• NumPy arrays can represent tensors
  - \( A = \text{np.zeros}(\ (5, \ 6, \ 7) \) \) \hspace{1cm} \( \rightarrow \hspace{0.5cm} A.\text{shape} == (5, \ 6, \ 7) \)

• Transpose can take a permutation of dimensions
  - \( B = \text{np.transpose}(A, \ (2, \ 0, \ 1)) \) \hspace{1cm} \( \rightarrow \hspace{0.5cm} B.\text{shape} == (7, \ 5, \ 6) \)

• Careful if using \text{np.matmul} or \text{np.dot} for tensor multiplication — \text{np.tensordot} lets you explicitly specify axes to sum over
  - \( C = \text{np.tensordot}(A, B, \ (2, \ 0)) \) \hspace{1cm} \( \rightarrow \hspace{0.5cm} C.\text{shape} == (5, \ 6, \ 5, \ 6) \)
  - \( D = \text{np.tensordot}(A, B, \ ([2,1], [0,2])) \) \hspace{1cm} \( \rightarrow \hspace{0.5cm} D.\text{shape} == (5, \ 5) \)
Tensors and Differentiation

\[ \nabla f : \mathbb{R}^{(\text{input shape})} \rightarrow \mathbb{R}^{(\text{output shape}) \times (\text{input shape})} \]

- If you take more advanced math, you’ll learn that the “dimensions” of tensors behave in two different ways: **covariant** and **contravariant**.
- We won’t go into that here, except to note that the dimensions arising from differentiation always behave “transposed”.
  - For example, the gradient of a scalar function of a vector is a **row** vector.
  - Intimate connection to **directional derivatives**: multiplying a gradient by a **direction** $d$ (an object of the input shape) gives you derivative of the output in that direction:

\[
D_d f(x) \text{ or } \nabla_d f(x) = \nabla f(x) \cdot d
\]

where the last dimension(s) of $\nabla f$ are dotted against $d$. 
Before we get into specific examples of these generalized derivatives, let’s review which rules from single-variable calculus still work:

- Derivative of a *constant* of any shape is 0
- Derivative of the variable with respect to which we’re differentiating is 1, or the *identity* of the appropriate shape
- Derivative of a *sum* is the sum of derivatives
- Derivative of a *scalar multiple* is the constant times the derivative
- *Chain rule* works, but order might matter: \( \frac{d}{dx} f(g(x)) = f'(g(x)) g'(x) \)
- *Product rule* requires care about dimensions and transposes (stay tuned!)
Function of a Scalar, Returning a Vector

Simple...

\[ f(x) = \begin{bmatrix} 3x + 42 \\ \sin x \end{bmatrix} \]

\[ \frac{df}{dx} = \begin{bmatrix} 3 \\ \cos x \end{bmatrix} \]

Can also consider functions written as scalar/vector products:

\[ f(x) = x\mathbf{v} \]

\[ \frac{df}{dx} = \begin{bmatrix} \frac{d}{dx}(x\mathbf{v}_1) \\ \frac{d}{dx}(x\mathbf{v}_2) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \end{bmatrix} = \mathbf{v} \]
Function of a Vector, Returning a Scalar

This is the ordinary gradient, which is a row vector of partial derivatives:

\[
\begin{align*}
    f \left( \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) &= x_1^3 + x_1 x_2 + 42 x_3 \\
    \nabla f &= \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \end{bmatrix} \\
    &= \begin{bmatrix} 3x_1^2 + x_2 & x_1 & 42 \end{bmatrix}
\end{align*}
\]
Directional Derivative

\[ \nabla f = \begin{bmatrix} 3x_1^2 + x_2 & x_1 & 42 \end{bmatrix} \]

How does \( f \) change with an infinitesimal step in direction \( d = \begin{bmatrix} 0.6 \\ 0 \\ 0.8 \end{bmatrix} \)?

\[ \nabla_d f = \begin{bmatrix} 3x_1^2 + x_2 & x_1 & 42 \end{bmatrix} \begin{bmatrix} 0.6 \\ 0 \\ 0.8 \end{bmatrix} = 1.8x_1^2 + 0.6x_2 + 33.6 \]

This is a scalar— the same shape as the output of \( f \).
Directional Derivative

- What if we had done this in the previous case, where we had a function of a *scalar*, returning a *vector*?

  \[
  f(x) = x\mathbf{v} \\
  \frac{df}{dx} = \mathbf{v}
  \]

- Multiplying by a (scalar) infinitesimal step in \(x\) in “direction” 1, we get just \(\mathbf{v}\).
- This is a (column) vector — the same shape as the output of \(f\).
Let’s try a dot product!

$$f(x) = \mathbf{v} \cdot \mathbf{x} = \sum_i v_i x_i$$

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} v_1 & v_2 & \cdots \end{bmatrix}$$

$$= \mathbf{v}^T$$

But $\mathbf{v} \cdot \mathbf{x} = \mathbf{v}^T \mathbf{x} = \mathbf{x}^T \mathbf{v}$, so we have the following:

$$\nabla (\mathbf{v}^T \mathbf{x}) = \mathbf{v}^T \quad \text{and} \quad \nabla (\mathbf{x}^T \mathbf{v}) = \mathbf{v}^T$$
Next interesting case:

\[ f(x) = x \cdot x = x^T x = \|x\|^2 = \sum_i x_i^2 \]

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots \\ 2x_1 & 2x_2 & \cdots \end{bmatrix} = 2x^T \]

Note the analogy to \( \frac{d}{dx} x^2 = 2x \), but we need the transpose to get the output shape right.
Function of a Vector, Returning a Scalar

Even more interesting:

\[ f(x) = \|x\| = \sqrt{x^T x} = \sqrt{\sum x_i^2} \]

Applying the chain rule:

\[ \nabla f = \frac{1}{2} (x^T x)^{-\frac{1}{2}} \nabla (x^T x) \]
\[ = \frac{2x^T}{2 \sqrt{x^T x}} \]
\[ = \frac{x^T}{\|x\|} \]
Directional Derivative

\[ \nabla \| x \| = \frac{x^\top}{\| x \|} \]

As \( x \) changes by an infinitesimal step in direction \( d \),

\[ \nabla_d \| x \| = \frac{x}{\| x \|} \cdot d \]

Intuitive: if \( d \) is in the direction of \( x \), change in \( \| x \| \) is 1 times the step size, etc.
Function of a Vector, Returning a Vector

Let’s move on to a Jacobian:

\[ f(x) = Mx \]

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \ldots \end{bmatrix} \]

The only terms in \( Mx \) involving \( x_i \) come from the \( i^{th} \) column of \( M \), so:

\[ \frac{\partial f}{\partial x_1} = \begin{bmatrix} M_{11} \\ M_{21} \\ \vdots \end{bmatrix}, \quad \frac{\partial f}{\partial x_2} = \begin{bmatrix} M_{12} \\ M_{22} \\ \vdots \end{bmatrix}, \text{ etc.} \]
Function of a Vector, Returning a Vector

• Stitching everything together,

\[ \nabla(Mx) = \begin{bmatrix}
\begin{pmatrix} M_{11} \\ M_{21} \end{pmatrix} & \begin{pmatrix} M_{12} \\ M_{22} \end{pmatrix} & \cdots \\
\vdots & \vdots & \ddots 
\end{bmatrix} = M \]

• Special case: \( \nabla(x) = \nabla(Ix) = I \)

• This reinforces our intuition that differentiating any constant thing times \( x \) gives just that constant, whether it’s a scalar, vector, matrix, etc.
Function of a Vector, Returning a Matrix

\[ F(x) = x\mathbf{v}^\top = \begin{bmatrix} x_1 v_1 & x_1 v_2 & \cdots \\ x_2 v_1 & x_2 v_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \]

Just apply the rules, and watch the tensor appear!

\[ \nabla F(x) = \begin{bmatrix} \frac{\partial F}{\partial x_1} & \frac{\partial F}{\partial x_2} & \cdots \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots \\ v_1 & v_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \cdots \]
Function of a Vector, Returning a Matrix

\[ \nabla F(x) = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 1 & \cdots \end{bmatrix} v^T \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} v^T \cdots \]

At this point, it might be tempting to factor out the \( v^T \). But be careful!

This object is an \( n \times n \times n \) tensor, and if you factor it into an \( n \times 1 \times n \) tensor and a \( 1 \times n \) vector, you have to remember which dimensions should be multiplied!
Generalizing Product Rule

- As we just saw, tensor multiplication can get confusing. This complicates cleanly stating a generalized product rule.
- But, let’s derive a rule for vector-vector (dot) products:

\[ \nabla (\mathbf{v} \cdot \mathbf{w}) = \nabla \left( \sum_i v_i w_i \right) \]

where \( \mathbf{v} \) and \( \mathbf{w} \) are both potentially functions of \( \mathbf{x} \).
- Writing out the partial derivatives,

\[ \nabla (\mathbf{v} \cdot \mathbf{w}) = \begin{bmatrix} \frac{\partial (\sum v_i w_i)}{\partial x_1} & \frac{\partial (\sum v_i w_i)}{\partial x_2} & \ldots \end{bmatrix} \]
Generalizing Product Rule

• Because $v_i$ and $w_i$ are just scalars, the product rule works normally:

$$\frac{\partial}{\partial x_1} \left( \sum v_i w_i \right) = \sum_i \left( v_i \frac{\partial w_i}{\partial x_1} + \frac{\partial v_i}{\partial x_1} w_i \right)$$

• Applying the distributive rule, we get

$$\nabla (v \cdot w) = v \cdot (\nabla w) + (\nabla v) \cdot w$$

• Or, in matrix notation,

$$\nabla (v^T w) = v^T \nabla w + w^T \nabla v$$
Let’s apply our newly-derived knowledge!

\[
f(x) = x^\top M x \\
\nabla f = x^\top \nabla (M x) + (M x)^\top \nabla x \\
= x^\top M + x^\top M^\top I \\
= x^\top (M + M^\top)
\]

Note the similarity to \( \frac{d}{dx}(ax^2) = 2ax \).
We’ve mentioned before that our methods for solving overdetermined linear systems of the form $Ax = b$ minimize a least-squares residual:

$$\arg \min_x \|Ax - b\|^2$$

Let’s apply the methods we’ve learned to find the $x$ that minimizes this, by taking the derivative (gradient) and setting it equal to 0.
The Grand Finale: Least Squares

• Applying the chain rule:

$$\nabla \|Ax - b\|^2 = 2\|Ax - b\| \nabla \|Ax - b\|$$

• And again (order matters!):

$$= 2\|Ax - b\| \frac{(Ax - b)^\top}{\|Ax - b\|} \nabla (Ax - b)$$

• And computing a final Jacobian:

$$= 2(Ax - b)^\top A$$
The Grand Finale: Least Squares

To check, let's derive this a different way:

\[ \|Ax - b\|^2 = (Ax - b)^T(Ax - b) \]
\[ = x^T A^T Ax - x^T A^T b - b^T A x + b^T b \]
\[ \nabla \|Ax - b\|^2 = x^T (A^T A + (A^T A)^T) - b^T A - b^T A + 0 \]
\[ = 2x^T A^T A - 2b^T A \]
\[ = 2(x^T A^T - b^T)A \]
\[ = 2(Ax - b)^T A \]
The Grand Finale: Least Squares

• Setting the gradient equal to a row vector of zeros:

\[ 2(Ax - b)^\top A = 0^\top \]

• Transposing and dividing by 2:

\[ A^\top (Ax - b) = 0 \]

• And finally, rearranging:

\[ A^\top Ax = A^\top b \]