Foundations of Machine Learning
African Masters in Machine Intelligence

Vector Calculus

Marc Deisenroth

Quantum Leap Africa
African Institute for Mathematical Sciences, Rwanda

Department of Computing
Imperial College London

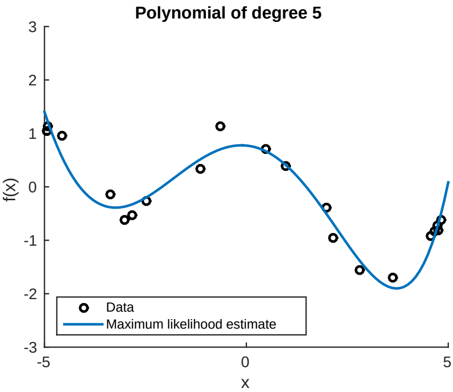
@mpd37
mdeisenroth@aimsammi.org

September 26, 2018

Reference

Deisenroth et al.: Mathematics for Machine Learning, Chapter 5
https://mml-book.com

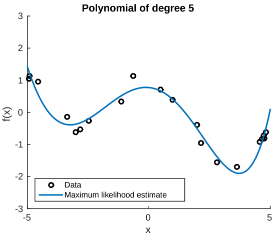
Curve Fitting (Regression) in Machine Learning (1)



- Setting: Given inputs $\boldsymbol{x}$, predict outputs/targets $\boldsymbol{y}$
- Model f that depends on parameters $\boldsymbol{\theta}$. Examples:
  - Linear model: $f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}, \quad \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^D$
  - Neural network: $f(\boldsymbol{x}, \boldsymbol{\theta}) = NN(\boldsymbol{x}, \boldsymbol{\theta})$

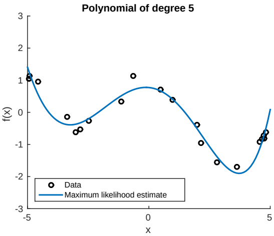
Curve Fitting (Regression) in Machine Learning (2)

- Training data, e.g., N pairs $(\boldsymbol{x}_i, y_i)$ of inputs $\boldsymbol{x}_i$ and observations y_i
- Training the model means finding parameters $\boldsymbol{\theta}^*$, such that $f(\boldsymbol{x}_i, \boldsymbol{\theta}^*) \approx y_i$



Curve Fitting (Regression) in Machine Learning (2)

- Training data, e.g., N pairs $(\boldsymbol{x}_i, y_i)$ of inputs $\boldsymbol{x}_i$ and observations y_i
- Training the model means finding parameters $\boldsymbol{\theta}^*$, such that $f(\boldsymbol{x}_i, \boldsymbol{\theta}^*) \approx y_i$



- Define a loss function, e.g., $\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}))^2$, which we want to optimize
- Typically: Optimization based on some form of gradient descent
  ▸▸ Differentiation required

Types of Differentiation

1. Scalar differentiation: $f : \mathbb{R} \to \mathbb{R}$

   $y \in \mathbb{R}$ w.r.t. $x \in \mathbb{R}$

2. Multivariate case: $f : \mathbb{R}^N \to \mathbb{R}$

   $y \in \mathbb{R}$ w.r.t. vector $\boldsymbol{x} \in \mathbb{R}^N$

3. Vector fields: $f : \mathbb{R}^N \to \mathbb{R}^M$

   vector $\boldsymbol{y} \in \mathbb{R}^M$ w.r.t. vector $\boldsymbol{x} \in \mathbb{R}^N$

4. General derivatives: $f : \mathbb{R}^{M \times N} \to \mathbb{R}^{P \times Q}$

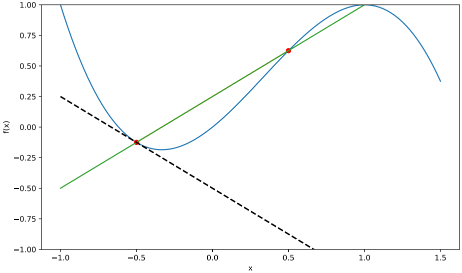
   matrix $\boldsymbol{y} \in \mathbb{R}^{P \times Q}$ w.r.t. matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$

Scalar Differentiation $f : \mathbb{R} \to \mathbb{R}$

- Derivative defined as the limit of the difference quotient

$$f'(x) = \frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

▶▶ Slope of the secant line through $f(x)$ and $f(x+h)$



Some Examples

$$
\begin{aligned}
f(x) &= x^n & f'(x) &= nx^{n-1} \\
f(x) &= \sin(x) & f'(x) &= \cos(x) \\
f(x) &= \tanh(x) & f'(x) &= 1 - \tanh^2(x) \\
f(x) &= \exp(x) & f'(x) &= \exp(x) \\
f(x) &= \log(x) & f'(x) &= \frac{1}{x}
\end{aligned}
$$

Rules

- Sum Rule

$$\big(f(x)+g(x)\big)' = f'(x) + g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

Rules

- Sum Rule

$$\big(f(x) + g(x)\big)' = f'(x) + g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

- Product Rule

$$\big(f(x)g(x)\big)' = f'(x)g(x) + f(x)g'(x) = \frac{df(x)}{dx}g(x) + f(x)\frac{dg(x)}{dx}$$

Rules

- Sum Rule

$$\big(f(x)+g(x)\big)' = f'(x) + g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

- Product Rule

$$\big(f(x)g(x)\big)' = f'(x)g(x) + f(x)g'(x) = \frac{df(x)}{dx}g(x) + f(x)\frac{dg(x)}{dx}$$

- Chain Rule

$$(g \circ f)'(x) = \big(g(f(x))\big)' = g'(f(x))f'(x) = \frac{dg(f(x))}{df}\frac{df(x)}{dx}$$

Rules

- Sum Rule

$$(f(x)+g(x))' = f'(x)+g'(x) = \frac{df(x)}{dx}+\frac{dg(x)}{dx}$$

- Product Rule

$$(f(x)g(x))' = f'(x)g(x)+f(x)g'(x) = \frac{df(x)}{dx}g(x)+f(x)\frac{dg(x)}{dx}$$

- Chain Rule

$$(g\circ f)'(x) = (g(f(x)))' = g'(f(x))f'(x) = \frac{dg(f(x))}{df}\frac{df(x)}{dx}$$

- Quotient Rule

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f(x)'g(x)-f(x)g(x)'}{(g(x))^2} = \frac{\frac{df}{dx}g(x)-f(x)\frac{dg}{dx}}{(g(x))^2}$$

Example: Scalar Chain Rule

$$(g \circ f)'(x) = \big(g(f(x))\big)' = g'(f(x))f'(x) = \frac{dg}{df}\frac{df}{dx}$$

Beginner

$$\begin{aligned} g(z) &= 6z + 3 \\ z = f(x) &= -2x + 5 \\ (g \circ f)'(x) &= \end{aligned}$$

Advanced

$$\begin{aligned} g(z) &= \tanh(z) \\ z = f(x) &= x^n \\ (g \circ f)'(x) &= \end{aligned}$$

Work it out with your neighbors

Example: Scalar Chain Rule

$$(g \circ f)'(x) = \big(g(f(x))\big)' = g'(f(x))f'(x) = \frac{dg}{df}\frac{df}{dx}$$

Beginner

$$\begin{aligned} g(z) &= 6z + 3 \\ z &= f(x) = -2x + 5 \\ (g \circ f)'(x) &= \underbrace{(6)}_{dg/df}\underbrace{(-2)}_{df/dx} \\ &= -12 \end{aligned}$$

Advanced

$$\begin{aligned} g(z) &= \tanh(z) \\ z &= f(x) = x^n \\ (g \circ f)'(x) &= \underbrace{\left(1 - \tanh^2(x^n)\right)}_{dg/df}\underbrace{nx^{n-1}}_{df/dx} \end{aligned}$$

Multivariate Differentiation $f : \mathbb{R}^N \to \mathbb{R}$

$$y = f(\boldsymbol{x}), \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N$$

- Partial derivative (change one coordinate at a time):

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_N) - f(\boldsymbol{x})}{h}$$

Multivariate Differentiation $f : \mathbb{R}^N \to \mathbb{R}$

$$y = f(\boldsymbol{x}), \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N$$

- Partial derivative (change one coordinate at a time):

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_N) - f(\boldsymbol{x})}{h}$$

- Jacobian vector (gradient) collects all partial derivatives:

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_N} \end{bmatrix} \in \mathbb{R}^{1 \times N}$$

Note: This is a row vector.

Example: Multivariate Differentiation

Beginner

$$f : \mathbb{R}^2 \to \mathbb{R}$$
$$f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 \in \mathbb{R}$$

Advanced

$$f : \mathbb{R}^2 \to \mathbb{R}$$
$$f(x_1, x_2) = (x_1 + 2x_2^3)^2 \in \mathbb{R}$$

Partial derivatives?

Work it out with your neighbors

Example: Multivariate Differentiation

Beginner

$$f : \mathbb{R}^2 \to \mathbb{R}$$

$$f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 \in \mathbb{R}$$

Advanced

$$f : \mathbb{R}^2 \to \mathbb{R}$$

$$f(x_1, x_2) = (x_1 + 2x_2^3)^2 \in \mathbb{R}$$

Partial derivatives

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 x_2 + x_2^3$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1^2 + 3x_1 x_2^2$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2(x_1 + 2x_2^3) \overbrace{(1)}^{\frac{\partial}{\partial x_1}(x_1 + 2x_2^3)}$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 2(x_1 + 2x_2^3) \underbrace{(6x_2^2)}_{\frac{\partial}{\partial x_2}(x_1 + 2x_2^3)}$$

Example: Multivariate Differentiation

Beginner

$$f : \mathbb{R}^2 \to \mathbb{R}$$

$$f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 \in \mathbb{R}$$

Advanced

$$f : \mathbb{R}^2 \to \mathbb{R}$$

$$f(x_1, x_2) = (x_1 + 2x_2^3)^2 \in \mathbb{R}$$

Partial derivatives

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 x_2 + x_2^3$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1^2 + 3x_1 x_2^2$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2(x_1 + 2x_2^3) \overbrace{(1)}^{\frac{\partial}{\partial x_1}(x_1+2x_2^3)}$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 2(x_1 + 2x_2^3) \underbrace{(6x_2^2)}_{\frac{\partial}{\partial x_2}(x_1+2x_2^3)}$$

Gradient $\quad \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(x_1,x_2)}{\partial x_1} & \frac{\partial f(x_1,x_2)}{\partial x_2} \end{bmatrix} \in \mathbb{R}^{1\times 2}$

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} 2x_1 x_2 + x_2^3 & x_1^2 + 3x_1 x_2^2 \end{bmatrix}$$

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} 2(x_1 + 2x_2^3) & 12(x_1 + 2x_2^3)x_2^2 \end{bmatrix}$$

Example: Multivariate Chain Rule

- Consider the function

$$L(\boldsymbol{e}) = \tfrac{1}{2}\|\boldsymbol{e}\|^2 = \tfrac{1}{2}\boldsymbol{e}^\top \boldsymbol{e}$$

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\,, \quad \boldsymbol{x} \in \mathbb{R}^N\,, \boldsymbol{A} \in \mathbb{R}^{M\times N}\,, \boldsymbol{e}, \boldsymbol{y} \in \mathbb{R}^M$$

- Compute the gradient $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}}$. What is the dimension/size of $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}}$?

Work it out with your neighbors

Example: Multivariate Chain Rule

- Consider the function

$$L(\boldsymbol{e}) = \tfrac{1}{2}\|\boldsymbol{e}\|^2 = \tfrac{1}{2}\boldsymbol{e}^\top \boldsymbol{e}$$

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{e}, \boldsymbol{y} \in \mathbb{R}^M$$

- Compute the gradient $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}}$. What is the dimension/size of $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}}$?

$$\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}} = \frac{\partial L}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}}$$

$$\frac{\partial L}{\partial \boldsymbol{e}} = \boldsymbol{e}^\top \in \mathbb{R}^{1\times M} \tag{1}$$

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}} = -\boldsymbol{A} \in \mathbb{R}^{M\times N} \tag{2}$$

$$\Longrightarrow \frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{x}} = \boldsymbol{e}^\top(-\boldsymbol{A}) = -(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})^\top \boldsymbol{A} \in \mathbb{R}^{1\times N}$$

Vector Field Differentiation $f : \mathbb{R}^N \to \mathbb{R}^M$

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_M(x_1, \ldots, x_N) \end{bmatrix}$$

Vector Field Differentiation $f : \mathbb{R}^N \to \mathbb{R}^M$

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M , \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_M(x_1, \ldots, x_N) \end{bmatrix}$$

- Jacobian matrix (collection of all partial derivatives)

$$\begin{bmatrix} \frac{dy_1}{d\boldsymbol{x}} \\ \vdots \\ \frac{dy_M}{d\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} \in \mathbb{R}^{M \times N}$$

Example: Vector Field Differentiation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

- Compute the gradient $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}$

Example: Vector Field Differentiation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

- Compute the gradient $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}$
  - Gradient:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{ij}x_j \qquad \Longrightarrow \frac{\partial f_i}{\partial x_j} = A_{ij}$$

Example: Vector Field Differentiation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

- Compute the gradient $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}$
  - Gradient:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{ij}x_j \qquad \Longrightarrow \quad \frac{\partial f_i}{\partial x_j} = A_{ij}$$

Example: Vector Field Differentiation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

- Compute the gradient $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}$
  - Gradient:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{ij}x_j \qquad \Longrightarrow \frac{\partial f_i}{\partial x_j} = A_{ij}$$

$$\Longrightarrow \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{M1} & \cdots & A_{MN} \end{bmatrix} = \boldsymbol{A} \in \mathbb{R}^{M\times N}$$

Dimensionality of the Gradient

In general: A function $\boldsymbol{f} : \mathbb{R}^{N} \to \mathbb{R}^{M}$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{M \times N}, \qquad \mathrm{d}\boldsymbol{f}[m,n] = \frac{\partial f_m}{\partial x_n}$$

Gradient dimension: # target dimensions $\times$ # input dimensions

Chain Rule

$$\frac{\partial}{\partial \boldsymbol{x}}(g \circ f)(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}\big(g(f(\boldsymbol{x}))\big) = \frac{\partial g(f)}{\partial f}\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

Example: Chain Rule

- Consider $f : \mathbb{R}^2 \to \mathbb{R}, \quad \boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$

$$f(\boldsymbol{x}) = f(x_1, x_2) = x_1^2 + 2x_2\,,$$
$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

Example: Chain Rule

- Consider $f : \mathbb{R}^2 \to \mathbb{R}, \quad \boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$

$$f(\boldsymbol{x}) = f(x_1, x_2) = x_1^2 + 2x_2\,,$$

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

- What are the dimensions of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$ and $\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$?

Work it out with your neighbors

Example: Chain Rule

- Consider $f : \mathbb{R}^2 \to \mathbb{R}, \quad \boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$

$$f(\boldsymbol{x}) = f(x_1, x_2) = x_1^2 + 2x_2\,,$$

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

- What are the dimensions of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$ and $\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$?

1×2 and 2×1

- Compute the gradient $\frac{\mathrm{d}f}{\mathrm{d}t}$ using the chain rule:

Example: Chain Rule

- Consider $f : \mathbb{R}^2 \to \mathbb{R}, \quad \boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$

$$f(\boldsymbol{x}) = f(x_1, x_2) = x_1^2 + 2x_2 \,,$$

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

- What are the dimensions of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}$ and $\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$?

1×2 and 2×1

- Compute the gradient $\frac{\mathrm{d}f}{\mathrm{d}t}$ using the chain rule:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial t} \end{bmatrix} = \begin{bmatrix} 2\sin t & 2 \end{bmatrix} \begin{bmatrix} \cos t \\ -\sin t \end{bmatrix}$$

$$= 2\sin t \cos t - 2\sin t = 2\sin t(\cos t - 1)$$

Derivatives with Respect to Matrices

- Recall: A function $f : \mathbb{R}^N \to \mathbb{R}^M$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}f}{\mathrm{d}x} \in \mathbb{R}^{M \times N}, \qquad \mathrm{d}f[m,n] = \frac{\partial f_m}{\partial x_n}$$

Gradient dimension: # target dimensions $\times$ # input dimensions

Derivatives with Respect to Matrices

- Recall: A function $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^M$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{M \times N}, \qquad \mathrm{d}\boldsymbol{f}[m,n] = \frac{\partial f_m}{\partial x_n}$$

Gradient dimension: # target dimensions × # input dimensions

- This generalizes to when the inputs (N) or targets (M) are **matrices**

Derivatives with Respect to Matrices

- Recall: A function $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^M$ has a gradient that is an $M \times N$-matrix with

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{M\times N}, \qquad \mathrm{d}\boldsymbol{f}[m,n] = \frac{\partial f_m}{\partial x_n}$$

  Gradient dimension: # target dimensions $\times$ # input dimensions

- This generalizes to when the inputs (N) or targets (M) are **matrices**
- Function $\boldsymbol{f} : \mathbb{R}^{M\times N} \to \mathbb{R}^{P\times Q}$, has a gradient that is a $(P \times Q) \times (M \times N)$ object (tensor)

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{X}} \in \mathbb{R}^{(P\times Q)\times(M\times N)}, \qquad \mathrm{d}\boldsymbol{f}[p,q,m,n] = \frac{\partial f_{pq}}{\partial X_{mn}}$$

Example 1: Derivatives with Respect to Matrices

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{x}\,, \quad \boldsymbol{f} \in \mathbb{R}^M, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} \in \mathbb{R}^{?}$$

Example 1: Derivatives with Respect to Matrices

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{x}\,, \quad \boldsymbol{f} \in \mathbb{R}^M, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{x} \in \mathbb{R}^N$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} \in \mathbb{R}^{\#\text{ target dim} \times \#\text{ input dim}} = M \times (M \times N)$$

$$\frac{d\boldsymbol{f}}{d\boldsymbol{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{A}} \\ \vdots \\ \frac{\partial f_M}{\partial \boldsymbol{A}} \end{bmatrix}, \quad \frac{\partial f_i}{\partial \boldsymbol{A}} \in \mathbb{R}^{1\times(M\times N)}$$

Example 2: Derivatives with Respect to Matrices

$$f_i = \sum_{j=1}^{N} A_{ij} x_j, \quad i = 1, \ldots, M$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_i(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & + A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{i1}x_1 + A_{i2}x_2 & \cdots & + A_{iN}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & + A_{MN}x_N \end{bmatrix}$$

$$\frac{\partial f_i}{\partial A_{iq}} = ? \qquad \frac{\partial f_i}{\partial A_{i,:}} = ? \qquad \frac{\partial f_i}{\partial A_{k \neq i,:}} = ? \qquad \frac{\partial f_i}{\partial \boldsymbol{A}} = ?$$

Example 2: Derivatives with Respect to Matrices

$$f_i = \sum_{j=1}^{N} A_{ij} x_j, \quad i = 1, \dots, M$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_i(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & +A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{i1}x_1 + A_{i2}x_2 & \cdots & +A_{iN}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & +A_{MN}x_N \end{bmatrix}$$

$$\frac{\partial f_i}{\partial A_{iq}} = \underbrace{x_q}_{\in \mathbb{R}} \quad \frac{\partial f_i}{\partial A_{i,:}} = ? \quad \frac{\partial f_i}{\partial A_{k \neq i,:}} = ? \quad \frac{\partial f_i}{\partial \boldsymbol{A}} = ?$$

Example 2: Derivatives with Respect to Matrices

$$f_i = \sum_{j=1}^{N} A_{ij} x_j, \quad i = 1, \dots, M$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_i(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & +A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{i1}x_1 + A_{i2}x_2 & \cdots & +A_{iN}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & +A_{MN}x_N \end{bmatrix}$$

$$\frac{\partial f_i}{\partial A_{iq}} = \underbrace{x_q}_{\in \mathbb{R}} \qquad \frac{\partial f_i}{\partial A_{i,:}} = \underbrace{\boldsymbol{x}^\top}_{\in \mathbb{R}^{1\times 1\times N}} \qquad \frac{\partial f_i}{\partial A_{k\neq i,:}} = ? \qquad \frac{\partial f_i}{\partial \boldsymbol{A}} = ?$$

Example 2: Derivatives with Respect to Matrices

$$f_i = \sum_{j=1}^{N} A_{ij} x_j, \quad i = 1, \ldots, M$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_i(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & +A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{i1}x_1 + A_{i2}x_2 & \cdots & +A_{iN}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & +A_{MN}x_N \end{bmatrix}$$

$$\frac{\partial f_i}{\partial A_{iq}} = \underbrace{x_q}_{\in \mathbb{R}} \qquad \frac{\partial f_i}{\partial A_{i,:}} = \underbrace{\boldsymbol{x}^\top}_{\in \mathbb{R}^{1\times1\times N}} \qquad \frac{\partial f_i}{\partial A_{k\neq i,:}} = \underbrace{\boldsymbol{0}^\top}_{\in \mathbb{R}^{1\times1\times N}} \qquad \frac{\partial f_i}{\partial \boldsymbol{A}} = ?$$

Example 2: Derivatives with Respect to Matrices

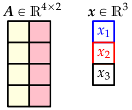
$$f_i = \sum_{j=1}^{N} A_{ij}x_j, \quad i = 1, \ldots, M$$

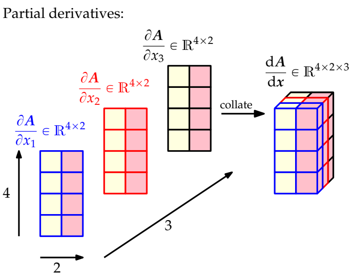
$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_i(\boldsymbol{x}) \\ \vdots \\ f_M(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + & \cdots & +A_{1N}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{i1}x_1 + A_{i2}x_2 & \cdots & +A_{iN}x_N \\ \vdots \quad\quad \vdots & \vdots & \vdots \\ A_{M1}x_1 + A_{M2}x_2 + & \cdots & +A_{MN}x_N \end{bmatrix}$$

$$\frac{\partial f_i}{\partial A_{iq}} = \underbrace{x_q}_{\in \mathbb{R}} \qquad \frac{\partial f_i}{\partial A_{i,:}} = \underbrace{\boldsymbol{x}^\top}_{\in \mathbb{R}^{1\times 1\times N}} \qquad \frac{\partial f_i}{\partial A_{k\neq i,:}} = \underbrace{\boldsymbol{0}^\top}_{\in \mathbb{R}^{1\times 1\times N}} \qquad \frac{\partial f_i}{\partial \boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{0}^\top \\ \vdots \\ \boldsymbol{x}^\top \\ \vdots \\ \boldsymbol{0}^\top \end{bmatrix}}_{\in \mathbb{R}^{1\times (M\times N)}}$$

Gradient Computation: Two Alternatives

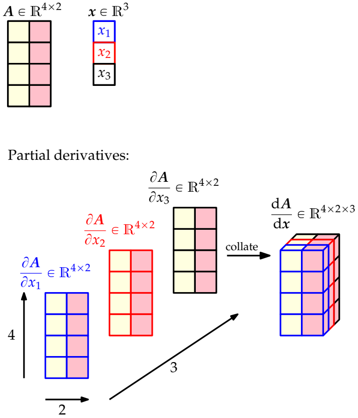
- Consider $\boldsymbol{f} : \mathbb{R}^3 \to \mathbb{R}^{4\times 2}$, $\quad \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A} \in \mathbb{R}^{4\times 2}$ where the entries A_{ij} depend on a vector $\boldsymbol{x} \in \mathbb{R}^3$
- We can compute $\frac{\mathrm{d}A(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{4\times 2\times 3}$ in two equivalent ways:

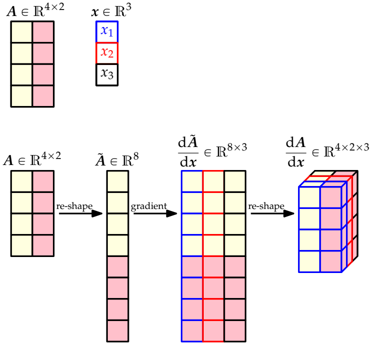




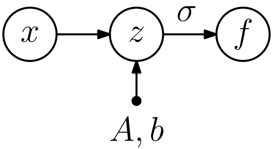
Gradient Computation: Two Alternatives

- Consider $\boldsymbol{f}: \mathbb{R}^3 \to \mathbb{R}^{4\times 2}$, $\quad \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A} \in \mathbb{R}^{4\times 2}$ where the entries A_{ij} depend on a vector $\boldsymbol{x} \in \mathbb{R}^3$
- We can compute $\frac{\mathrm{d}A(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{4\times 2\times 3}$ in two equivalent ways:





Gradients of a Single-Layer Neural Network



$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z}\in\mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} =$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} =$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M \times M} \in \mathbb{R}^{M \times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} =$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M \times M} \in \mathbb{R}^{M \times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M \times (M \times N)} \in \mathbb{R}^{M \times (M \times N)}$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z}\in\mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M\times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M\times M} \in \mathbb{R}^{M\times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M\times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M\times (M\times N)} \in \mathbb{R}^{M\times (M\times N)}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\mathrm{diag}(1 - \tanh^2(\boldsymbol{z}))}_{\in \mathbb{R}^{M\times M}}$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M \times M} \in \mathbb{R}^{M \times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M \times (M \times N)} \in \mathbb{R}^{M \times (M \times N)}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\mathrm{diag}(1 - \tanh^2(\boldsymbol{z}))}_{\in \mathbb{R}^{M \times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \underbrace{\boldsymbol{I}}_{\in \mathbb{R}^{M \times M}}$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z}\in\mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M\times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M\times M} \in \mathbb{R}^{M\times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M\times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M\times(M\times N)} \in \mathbb{R}^{M\times(M\times N)}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\operatorname{diag}(1 - \tanh^2(\boldsymbol{z}))}_{\in\mathbb{R}^{M\times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \underbrace{\boldsymbol{I}}_{\in\mathbb{R}^{M\times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top \end{bmatrix}}_{\in\mathbb{R}^{M\times(M\times N)}}$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M \times M} \in \mathbb{R}^{M \times M}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}}[i,j] = \sum_{l=1}^{M} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}[i,l] \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}[l,j]$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M \times (M \times N)} \in \mathbb{R}^{M \times (M \times N)}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\mathrm{diag}(1 - \tanh^2(\boldsymbol{z}))}_{\in \mathbb{R}^{M \times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \underbrace{\boldsymbol{I}}_{\in \mathbb{R}^{M \times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top \end{bmatrix}}_{\in \mathbb{R}^{M \times (M \times N)}}$$

Gradients of a Single-Layer Neural Network

$$\boldsymbol{f} = \tanh(\underbrace{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}}_{=:\boldsymbol{z} \in \mathbb{R}^M}) \in \mathbb{R}^M, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{b} \in \mathbb{R}^M$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}}_{M \times M} \in \mathbb{R}^{M \times M} \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{b}}[i, j] = \sum_{l=1}^{M} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}[i, l] \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}[l, j]$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}} = \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}}_{M \times M} \underbrace{\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}}_{M \times (M \times N)} \in \mathbb{R}^{M \times (M \times N)} \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{A}}[i, j, k] = \sum_{l=1}^{M} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}[i, l] \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}[l, j, k]$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\mathrm{diag}(1 - \tanh^2(\boldsymbol{z}))}_{\in \mathbb{R}^{M \times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \underbrace{\boldsymbol{I}}_{\in \mathbb{R}^{M \times M}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top \end{bmatrix}}_{\in \mathbb{R}^{M \times (M \times N)}}$$

Putting Things Together

- Inputs $\boldsymbol{x} \in \mathbb{R}^N$

Putting Things Together

- Inputs $\boldsymbol{x} \in \mathbb{R}^N$
- Observed outputs $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{z}) + \boldsymbol{\epsilon} \in \mathbb{R}^M, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$

Putting Things Together

- Inputs $\boldsymbol{x} \in \mathbb{R}^N$
- Observed outputs $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{z}) + \boldsymbol{\epsilon} \in \mathbb{R}^M, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$
- Train single-layer neural network with

$$f_{\boldsymbol{\theta}}(\boldsymbol{z}) = \tanh(\boldsymbol{z}) \in \mathbb{R}^M, \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^M, \quad \boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{b}\}$$

Putting Things Together

- Inputs $\boldsymbol{x} \in \mathbb{R}^N$
- Observed outputs $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{z}) + \boldsymbol{\epsilon} \in \mathbb{R}^M, \boldsymbol{\epsilon} \sim \mathcal{N}\big(\mathbf{0}, \boldsymbol{\Sigma}\big)$
- Train single-layer neural network with

$$f_{\boldsymbol{\theta}}(\boldsymbol{z}) = \tanh(\boldsymbol{z}) \in \mathbb{R}^M, \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^M, \quad \boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{b}\}$$

- Find $\boldsymbol{A}, \boldsymbol{b}$, such that the squared loss

$$L(\boldsymbol{\theta}) = \tfrac{1}{2}\|\boldsymbol{e}\|^2 \in \mathbb{R}, \quad \boldsymbol{e} = \boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}) \in \mathbb{R}^M$$

is minimized

Putting Things Together

Partial derivatives:

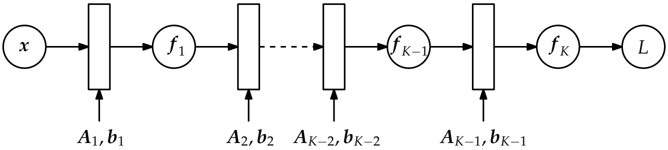
$$\frac{\partial L}{\partial \boldsymbol{A}} = \frac{\partial L}{\partial \boldsymbol{e}} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{f}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}$$

$$\frac{\partial L}{\partial \boldsymbol{b}} = \frac{\partial L}{\partial \boldsymbol{e}} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{f}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

$$\frac{\partial L}{\partial \boldsymbol{e}} = \underbrace{\boldsymbol{e}^\top}_{\in \mathbb{R}^{1\times M}} \qquad \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{f}} = \underbrace{-\boldsymbol{I}}_{\in \mathbb{R}^{M\times M}} \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \underbrace{\mathrm{diag}(1-\tanh^2(\boldsymbol{z}))}_{\in \mathbb{R}^{M\times M}}$$

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}} = \underbrace{\begin{bmatrix} \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top & \cdot & \boldsymbol{0}^\top \\ \cdot & & \cdot & & \cdot \\ \boldsymbol{0}^\top & \cdot & \boldsymbol{0}^\top & \cdot & \boldsymbol{x}^\top \end{bmatrix}}_{\in \mathbb{R}^{M\times(M\times N)}} \qquad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \underbrace{\boldsymbol{I}}_{\in \mathbb{R}^{M\times M}}$$

Gradients of a Multi-Layer Neural Network

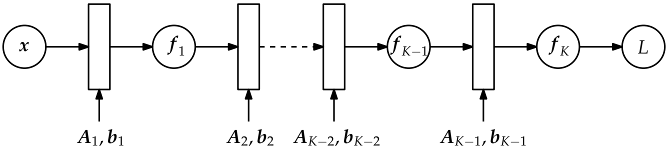


- Inputs $\boldsymbol{x}$, observed outputs $\boldsymbol{y}$
- Train multi-layer neural network with

$$\boldsymbol{f}_0 = \boldsymbol{x}$$

$$\boldsymbol{f}_i = \sigma_i(\boldsymbol{A}_{i-1}\boldsymbol{f}_{i-1} + \boldsymbol{b}_{i-1}), \quad i = 1, \ldots, K$$

Gradients of a Multi-Layer Neural Network



- Inputs $\boldsymbol{x}$, observed outputs $\boldsymbol{y}$
- Train multi-layer neural network with

$$\boldsymbol{f}_0 = \boldsymbol{x}$$

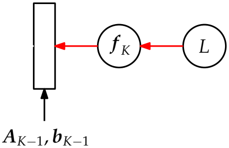
$$\boldsymbol{f}_i = \sigma_i(\boldsymbol{A}_{i-1}\boldsymbol{f}_{i-1} + \boldsymbol{b}_{i-1}), \quad i = 1, \ldots, K$$

- Find $\boldsymbol{A}_j, \boldsymbol{b}_j$ for $j = 0, \ldots, K-1$, such that the squared loss

$$L(\boldsymbol{\theta}) = \|\boldsymbol{y} - \boldsymbol{f}_{K,\boldsymbol{\theta}}(\boldsymbol{x})\|^2$$

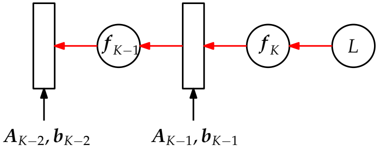
is minimized, where $\boldsymbol{\theta} = \{\boldsymbol{A}_j, \boldsymbol{b}_j\}, \quad j = 0, \ldots, K-1$

Gradients of a Multi-Layer Neural Network



$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}}$$

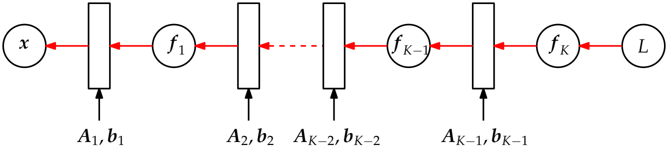
Gradients of a Multi-Layer Neural Network



$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \boxed{\frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}}$$

Gradients of a Multi-Layer Neural Network

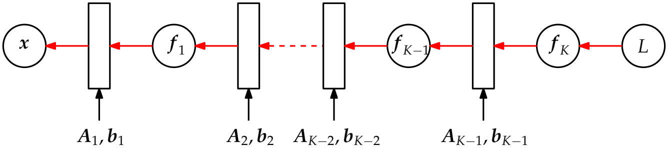


$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \boxed{\frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-3}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \boxed{\frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{f}_{K-2}} \frac{\partial \boldsymbol{f}_{K-2}}{\partial \boldsymbol{\theta}_{K-3}}}$$

Gradients of a Multi-Layer Neural Network



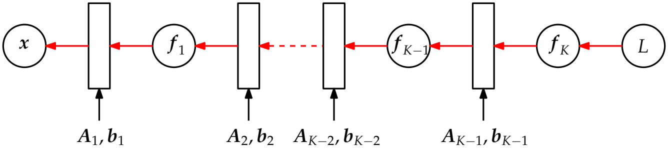
$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \boxed{\frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-3}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \boxed{\frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{f}_{K-2}} \frac{\partial \boldsymbol{f}_{K-2}}{\partial \boldsymbol{\theta}_{K-3}}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \cdots \boxed{\frac{\partial \boldsymbol{f}_{i+2}}{\partial \boldsymbol{f}_{i+1}} \frac{\partial \boldsymbol{f}_{i+1}}{\partial \boldsymbol{\theta}_i}}$$

Gradients of a Multi-Layer Neural Network



$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \boxed{\frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-3}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \boxed{\frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{f}_{K-2}} \frac{\partial \boldsymbol{f}_{K-2}}{\partial \boldsymbol{\theta}_{K-3}}}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \cdots \boxed{\frac{\partial \boldsymbol{f}_{i+2}}{\partial \boldsymbol{f}_{i+1}} \frac{\partial \boldsymbol{f}_{i+1}}{\partial \boldsymbol{\theta}_i}}$$

- Intermediate derivatives are stored during the forward pass

Example: Linear Regression with Neural Networks

- Linear regression with a neural network parametrized by $\boldsymbol{\theta}$, $f_{\boldsymbol{\theta}}$:

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_\epsilon^2\big)$$

Example: Linear Regression with Neural Networks

- Linear regression with a neural network parametrized by $\boldsymbol{\theta}$, $f_{\boldsymbol{\theta}}$:

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_\epsilon^2\big)$$

- Given inputs $\boldsymbol{x}_n$ and corresponding (noisy) observations y_n, $n = 1, \dots, N$, find parameters $\boldsymbol{\theta}^*$ that minimize the squared loss

$$L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \big(y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\big)^2 = \|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{X})\|^2$$

Training Neural Networks as Maximum Likelihood Estimation

- Training a neural network in the above way corresponds to maximum likelihood estimation:
  - If $\boldsymbol{y} = NN(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{I}\big)$ then the log-likelihood is

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\tfrac{1}{2}\|\boldsymbol{y} - NN(\boldsymbol{x}, \boldsymbol{\theta})\|^2$$

Training Neural Networks as Maximum Likelihood Estimation

- Training a neural network in the above way corresponds to maximum likelihood estimation:
  - If $\boldsymbol{y} = NN(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{I}\big)$ then the log-likelihood is

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\tfrac{1}{2}\|\boldsymbol{y} - NN(\boldsymbol{x}, \boldsymbol{\theta})\|^2$$

  - Find $\boldsymbol{\theta}^*$ by minimizing the negative log-likelihood:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} -\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \tfrac{1}{2}\|\boldsymbol{y} - NN(\boldsymbol{x}, \boldsymbol{\theta})\|^2 \\ &= \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \end{aligned}$$

Training Neural Networks as Maximum Likelihood Estimation

- Training a neural network in the above way corresponds to maximum likelihood estimation:
  - If $\boldsymbol{y} = NN(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ then the log-likelihood is

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\tfrac{1}{2}\|\boldsymbol{y} - NN(\boldsymbol{x}, \boldsymbol{\theta})\|^2$$

  - Find $\boldsymbol{\theta}^*$ by minimizing the negative log-likelihood:

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} -\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \tfrac{1}{2}\|\boldsymbol{y} - NN(\boldsymbol{x}, \boldsymbol{\theta})\|^2 \\ &= \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})\end{aligned}$$

- Maximum likelihood estimation can lead to overfitting (interpret noise as signal)

Example: Linear Regression (1)

- Linear regression with a polynomial of order M:

$$y = f(x, \boldsymbol{\theta}) + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma_\epsilon^2\big)$$

$$f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M = \sum_{i=0}^{M} \theta_i x^i$$

Example: Linear Regression (1)

- Linear regression with a polynomial of order M:

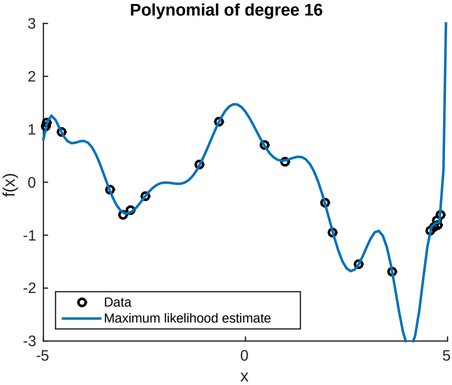
$$y = f(x, \boldsymbol{\theta}) + \epsilon\,, \quad \epsilon \sim \mathcal{N}\big(0, \sigma_\epsilon^2\big)$$

$$f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M = \sum_{i=0}^{M} \theta_i x^i$$

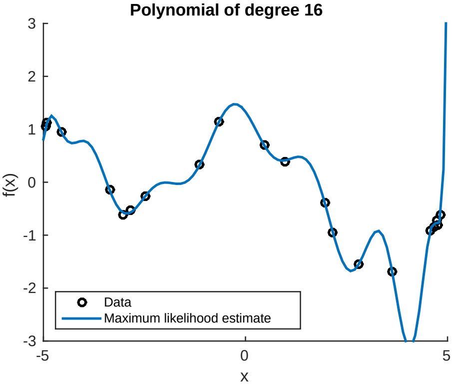
- Given inputs x_i and corresponding (noisy) observations y_i, $i = 1, \ldots, N$, find parameters $\boldsymbol{\theta} = [\theta_0, \ldots, \theta_M]^\top$, that minimize the squared loss (equivalently: maximize the likelihood)

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} (y_i - f(x_i, \boldsymbol{\theta}))^2$$

Example: Linear Regression (2)

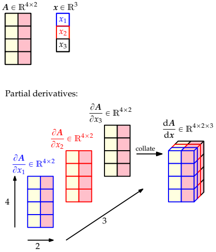


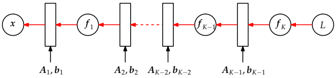
Example: Linear Regression (2)

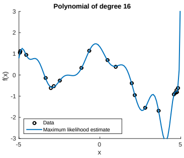


- Regularization, model selection etc. can address overfitting
- Alternative approach based on integration

Summary







- Vector-valued differentiation
- Chain rule
- Check the dimension of the gradients