Mathematics for Machine Learning

Lecture 5
(16.05.2024)

Vector Calculus

Mahammad Namazov

Table of contents

Univariate Functions

Quotient

Taylor Series

Rules

Difference Quotient

- The slope of the secant line can be computed as follows:

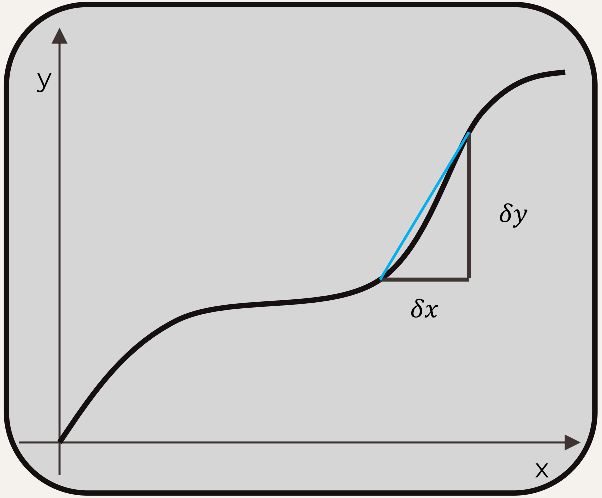
$$\frac{\delta y}{\delta x} \coloneqq \frac{f(x+\delta x)-f(x)}{\delta x}$$

- Derivative can also be computed using similar approach, when this difference converges to zero:

$$\frac{df}{dx} \coloneqq \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}, h>0$$

- Let's check the equation for the following polynomial:

$$f(x) = x^n$$



Taylor Series

- The Taylor polynomial of degree n of $f: \mathbb{R} \to \mathbb{R}$ at x_0 is defined as:

$$T_n(x) := \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

  - $f^{(k)}(x_0)$ is k^{th} derivative of f at x_0 (assuming it exists);
  - $\frac{f^{(k)}(x_0)}{k!}$ are coefficients of the polynomial;

- Taylor Series: For smooth function $f \in \mathcal{C}^{\infty}, f: \mathbb{R} \to \mathbb{R}$:

$$T_{\infty}(x) := \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

  - f is continuously differentiable infinitely many times;
  - When $x_0 = 0$, Taylor series becomes McLaurin series;

Differentiation Rules

- Product Rule:

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

- Quotient Rule:

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

- Sum Rule:

$$(f(x) + g(x))' = f'(x) + g'(x)$$

- Chain Rule:

$$\left(g\left(f(x)\right)\right)' = (g \circ f)'(x) = g'\left(f(x)\right)f'(x)$$

  - $(g \circ f)$ is function composition:

$$x \longmapsto f(x) \longmapsto g(f(x))$$

Partial Differentiation

Derivative

Basic Rules

Chain Rule

Partial Derivative

- For a function $f: \mathbb{R}^n \to \mathbb{R}$, where the value of function depends on n variables:
  - We can say that function value depends on n-dimensional vector $[x_1, x_2, \dots, x_n]^{\mathrm{T}} = \boldsymbol{x} \in \mathbb{R}^n$:
  - For x_j where $j \in [1, n]$:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_j} = \lim_{h \to 0} \frac{f\left(x_1, x_2, \dots, x_{j-1}, x_j + h, x_{j+1}, \dots, x_n\right) - f(\boldsymbol{x})}{h}$$

- Collecting all such elements in a row vector:

$$\nabla_{\boldsymbol{x}} f = \frac{df}{d\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1 \times n}$$

Hint: *When you compute partial derivative of f wrt x_j (i.e., $\frac{\partial f}{\partial x_j}$), other variables become constant for that step;*

Basic Rules

- Product Rule:

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

- Sum Rule:

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

- Chain Rule:

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

Chain Rule (more detailed)

- You are given a function $f: \mathbb{R}^2 \to \mathbb{R}$, which depends on vector $\boldsymbol{x} \in \mathbb{R}^2$;

- Now assume, each entry of this vector is a function of a single variable $t \in \mathbb{R}$:

$$f = f(\boldsymbol{x}(t))$$

- In other words, varying t we impact $\boldsymbol{x}$, which impacts the result of f. Thus, the main change happens in t-level:

$$\frac{df}{dt} = \frac{d}{dx} f\big(x(t)\big) \frac{dx}{dt}$$

  - What is missing in the equation above?

  - Another issue: What if t is also a vector?

Vector Valued Functions

Jacobian

Gradient

Least Squares

wrt. Matrices

Jacobian

- We analyzed the scenario, where a single function varies with respect to multiple variables;
- What if we have multiple functions (e.g., $\boldsymbol{f} \in \mathbb{R}^m$), which vary with respect to multiple variables (e.g., $\boldsymbol{x} \in \mathbb{R}^n$:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- We know that derivative of a single function wrt vector results in a row vector, which has as much columns as number of variables:

$$\nabla_{\boldsymbol{x}} f_k = \frac{d f_k}{d\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_k}{\partial x_1} & \dots & \frac{\partial f_k}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1\times n}$$

- Now if we have m such rows, we can simply write as follows:

$$\nabla_{\boldsymbol{x}} \boldsymbol{f} = [\nabla_{\boldsymbol{x}} f_1 \quad \dots \quad \nabla_{\boldsymbol{x}} f_{k-1} \quad \nabla_{\boldsymbol{x}} f_k \quad \nabla_{\boldsymbol{x}} f_{k+1} \quad \dots \quad \nabla_{\boldsymbol{x}} f_m]^T \in \mathbb{R}^{m\times n}$$

  - **Note: Transpose was used for space limitations ☺**

Gradient of VV functions

- Assume you are given:

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

- Steps to compute Jacobian:
  - Determine the dimension of $\frac{d\boldsymbol{f}}{dx}$:

$$\boldsymbol{f}: \mathbb{R}^N \to \mathbb{R}^M \Rightarrow \frac{d\boldsymbol{f}}{dx} \in \mathbb{R}^{M\times N}$$

  - Compute partial derivative of each function with respect to each variable:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{i,j}x_j \Rightarrow \frac{\partial f_i}{\partial x_j} = A_{i,j}$$

  - Thus, we have:

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

Gradient of Vectors wrt. Matrices

- Assume you are given (the same scenario):

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

- Let's determine the dimension and compute the partial derivative of each function wrt A:

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

- Using the similar approach:

$$f_i = \sum_{j=1}^{N} A_{ij}x_j, i = 1, \dots, M \Rightarrow \frac{\partial f_i}{\partial A_{iq}} = x_q$$

In Deep Networks

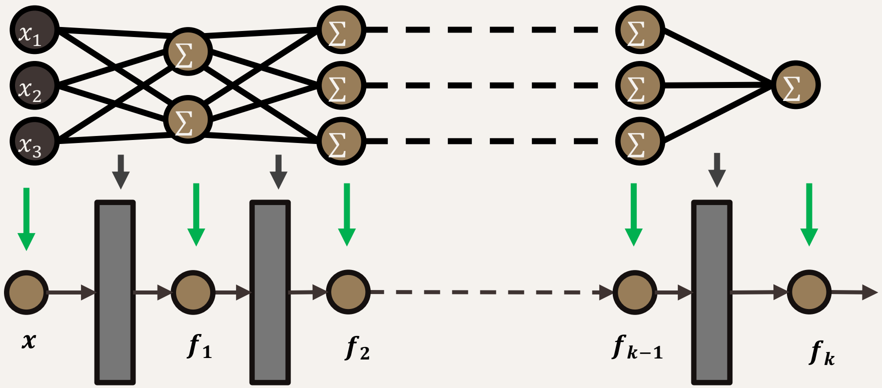
Backpropagation

Gradients in DN

Automatic Differentiation

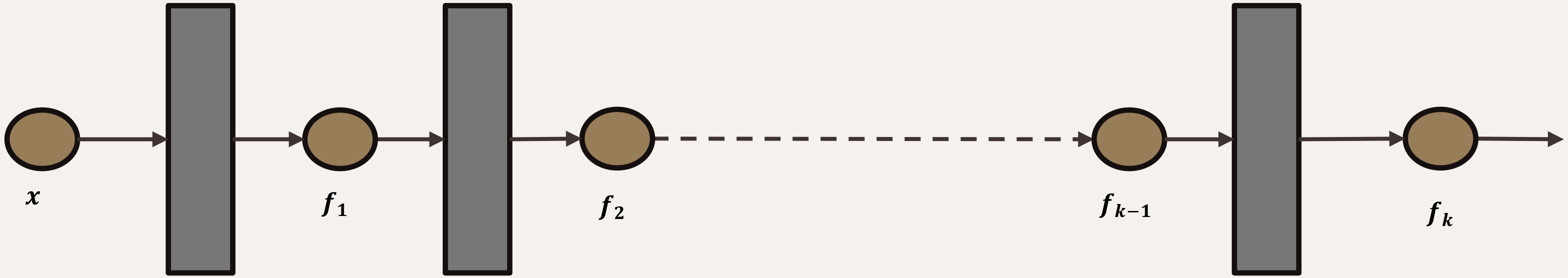
Backpropagation

- Usually Deep Networks are shown as below:



- $f_0 = x$
- $f_i(x_{i-1}) = \sigma(A_{i-1}f_{i-1} + b_{i-1})$
- $L(\boldsymbol{\theta}) = \|\boldsymbol{y} - f_k(\boldsymbol{\theta}, \boldsymbol{x})\|^2$

Backpropagation



$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial f_k} \frac{\partial f_k}{\partial f_{k-1}} \cdots \frac{\partial f_{i+2}}{\partial f_{i+1}} \frac{\partial f_{i+1}}{\partial f_i} \frac{\partial f_i}{\partial \theta_{i-1}}$$

$$\frac{\partial L}{\partial \theta_{k-2}} = \frac{\partial L}{\partial f_k} \frac{\partial f_k}{\partial f_{k-1}} \frac{\partial f_{k-1}}{\partial \theta_{k-2}}$$

$$\frac{\partial L}{\partial \theta_{k-1}} = \frac{\partial L}{\partial f_k} \frac{\partial f_k}{\partial \theta_{k-1}}$$

Automatic Differentiation

- Every complex operation is a combination of several operations
- Automatic Differentiation applies:
  - Elementary arithmetic operations: addition and multiplication
  - Elementary functions: sin, cos, exp, log
- Assume you have the following problem to solve:

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \sin(x^2 - 1)$$

- Steps would be:
  - Build a computation graphs with:
    - Inputs;
    - Functions;
    - Intermediate outputs;
  - Once you have the graph, go backward step by step;

Conclusion

Summary

Takeaways

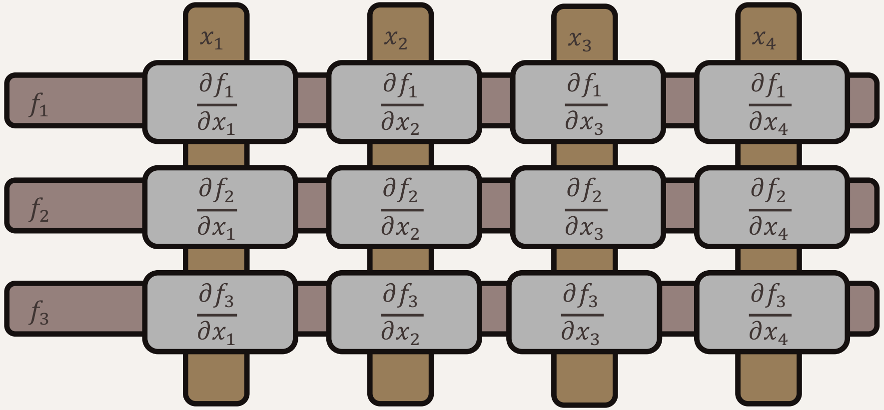
References

Summary

- Differentiation is significant for optimization purposes;
- It is useful to detect the change of the output with respect to some specific parameters;
- If you have a function of vector, then change of the function will be shown by a vector (i.e., gradient);
- If you have a vector of functions, each of which change with respect to vectors, the change of each function with respect to each variable will be shown by a Matrix (i.e., Jacobian);
- In the Deep Networks, we need to update model parameters based on the error;
- You have error, now you can compute ”what causes how to this error” by gradients;

Takeaways

- Using Jacobians in Backpropagation increases the speed of computation;
- Each row of Jacobian, represents each function’s partial derivative with respect to each variable;



References

- Further information to read:
  - Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.
  - Chapter 5, all sections (Sections 7, 8, 9 are optional)

The End

Thanks for your attention and patience!

Mahammad Namazov