

# Matrix Derivatives. Automatic Differentiation

## Seminar

Optimization for ML. Faculty of Computer Science. HSE University

# Theory recap. Differential

- Differential  $df(x)[\cdot] : U \rightarrow V$  in point  $x \in U$  for  $f(\cdot) : U \rightarrow V$ :

$$f(x+h) - f(x) = \underbrace{df(x)[h]}_{\text{differential}} + o(\|h\|)$$

- Canonical form of the differential:

$U \rightarrow V$	$\mathbb{R}$	$\mathbb{R}^n$	$\mathbb{R}^{n \times m}$
$\mathbb{R}$	$f'(x)dx$	$\nabla f(x)dx$	$\nabla f(x)dx$
$\mathbb{R}^n$	$\nabla f(x)^T dx$	$J(x)dx$	—
$\mathbb{R}^{n \times m}$	$tr(\nabla f(X)^T dX)$	—	—

# Theory recap. Differentiation Rules

- Useful differentiation rules and standard derivatives:

Differentiation Rules	Standard Derivatives
$dA = 0$	$d(\langle A, X \rangle) = \langle A, dX \rangle$
$d(\alpha X) = \alpha(dX)$	$d(\langle Ax, x \rangle) = \langle (A + A^T)x, dx \rangle$
$d(AXB) = A(dX)B$	$d(\text{Det}(X)) = \text{Det}(X)\langle X^{-T}, dX \rangle$
$d(X + Y) = dX + dY$	$d(X^{-1}) = -X^{-1}(dX)X^{-1}$
$d(X^T) = (dX)^T$	
$d(XY) = (dX)Y + X(dY)$	
$d(\langle X, Y \rangle) = \langle dX, Y \rangle + \langle X, dY \rangle$	
$d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$	

# Matrix Calculus. Problem 1

## Example

Find  $\nabla f(x)$ , if  $f(x) = \frac{1}{2}x^T Ax + b^T x + c$ .

## Matrix Calculus. Problem 2

### Example

Find  $\nabla f(X)$ , if  $f(X) = \text{tr}(AX^{-1}B)$

- $h(x) = f(g(x)) \Rightarrow dh(x_0)[dx] = df(g(x_0))[dg(x_0)[dx]]$

## Matrix Calculus. Problem 3

### Example

Find the gradient  $\nabla f(x)$  and hessian  $\nabla^2 f(x)$ , if  $f(x) = \frac{1}{3}\|x\|_2^3$

- $d^2 f(x)[h_1, h_2] = d \left( \underbrace{df(x)[h_1]}_{\text{fixed when take outer } d(\cdot)} \right) [h_2]$
- Canonic form for  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ :  $d^2 f(x)[h_1, h_2] = h_1^T \underbrace{\nabla^2 f(x)}_{\text{hessian}} h_2$

# Automatic Differentiation. Forward mode



Figure 1: Illustration of forward chain rule to calculate the derivative of the function  $v_i$  with respect to  $w_k$ .

- Uses the forward chain rule
- Has complexity  $d \times \mathcal{O}(T)$  operations

## Automatic Differentiation. Reverse mode



Figure 2: Illustration of reverse chain rule to calculate the derivative of the function  $L$  with respect to the node  $v_i$ .

- Uses the backward chain rule
- Stores the information from the forward pass
- Has complexity  $\mathcal{O}(T)$  operations



# Automatic Differentiation. Problem 1

## Example

Which of the AD modes would you choose (forward/ reverse) for the following computational graph of primitive arithmetic operations?



Figure 3: Which mode would you choose for calculating gradients there?

## Automatic Differentiation. Problem 2



Figure 4:  $x$  could be found as a solution of linear system

Suppose, we have an invertible matrix  $A$  and a vector  $b$ , the vector  $x$  is the solution of the linear system  $Ax = b$ , namely one can write down an analytical solution  $x = A^{-1}b$ .

Find the derivatives  $\frac{\partial L}{\partial A}$ ,  $\frac{\partial L}{\partial b}$ .

## Automatic Differentiation. Problem 3



Figure 5: Computation graph for singular regularizer

Suppose, we have the rectangular matrix  $W \in \mathbb{R}^{m \times n}$ , which has a singular value decomposition:

$$W = U\Sigma V^T, \quad U^T U = I, \quad V^T V = I, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min(m,n)})$$

The regularizer  $R(W) = \text{tr}(\Sigma)$  in any loss function encourages low rank solutions. Find the derivative  $\frac{\partial R}{\partial W}$ .

# Computation experiment with JAX

- JAX docs: <https://jax.readthedocs.io/en/latest/notebooks/quickstart.html>