Vector and Matrix Derivatives (Quick Reference)

1. Scalar Function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$

For $\mathbf{x} = [x_1, \ldots, x_n]^\top$,

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n.$$

Common results:

$$\begin{aligned}
\nabla_{\mathbf{x}}(a^\top \mathbf{x}) &= a, \\
\nabla_{\mathbf{x}}(\mathbf{x}^\top a) &= a, \\
\nabla_{\mathbf{x}}(\mathbf{x}^\top \mathbf{x}) &= 2\mathbf{x}, \\
\nabla_{\mathbf{x}}\left(\tfrac{1}{2}\mathbf{x}^\top \mathbf{x}\right) &= \mathbf{x}, \\
\nabla_{\mathbf{x}}(\mathbf{x}^\top A \mathbf{x}) &= (A + A^\top)\mathbf{x}, \\
\nabla_{\mathbf{x}}\left(\tfrac{1}{2}\mathbf{x}^\top A^\top A \mathbf{x}\right) &= A^\top A \mathbf{x}, \\
\nabla_{\mathbf{x}}\left(\tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2\right) &= \mathbf{A}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}).
\end{aligned}$$

—

2. Vector Function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$

Let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^\top$. Then the **Jacobian matrix** is

$$J_{\mathbf{f}}(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}^\top} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Common results:

$$\begin{aligned}
\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x}) &= \mathbf{A}^\top, \\
\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x} + \mathbf{b}) &= \mathbf{A}^\top, \\
\nabla_{\mathbf{x}}(\mathbf{x}^\top \mathbf{A}) &= \mathbf{A}, \\
\nabla_{\mathbf{x}}[(\mathbf{A}\mathbf{x} + \mathbf{b})^\top \mathbf{c}] &= \mathbf{A}^\top \mathbf{c}.
\end{aligned}$$

—

3. Matrix Calculus Results

For $X \in \mathbb{R}^{m \times n}$ and constant matrices A, B:

$$\begin{aligned}
\frac{\partial}{\partial X} \operatorname{tr}(A^\top X) &= A, \\
\frac{\partial}{\partial X} \operatorname{tr}(X^\top A X) &= AX + A^\top X, \\
\frac{\partial}{\partial X} \|AX - B\|_F^2 &= 2A^\top(AX - B), \\
\frac{\partial}{\partial X} \tfrac{1}{2}\|AX - B\|_F^2 &= A^\top(AX - B), \\
\frac{\partial}{\partial X} \operatorname{tr}(AXBX^\top) &= A^\top X(B + B^\top).
\end{aligned}$$

—

4. Chain Rule Identities

For composition $f(g(\mathbf{x}))$:

$$\nabla_{\mathbf{x}} f = J_g(\mathbf{x})^\top \nabla_g f.$$

For $\mathbf{f}(\mathbf{g}(\mathbf{x}))$:

$$J_{\mathbf{f} \circ \mathbf{g}}(\mathbf{x}) = J_{\mathbf{f}}(\mathbf{g}(\mathbf{x}))\, J_{\mathbf{g}}(\mathbf{x}).$$

—

5. Useful Vector Identities

$$\begin{aligned}
\nabla_{\mathbf{x}}(\mathbf{a}^\top \mathbf{x}) &= \mathbf{a}, \\
\nabla_{\mathbf{x}}\|\mathbf{x}\| &= \frac{\mathbf{x}}{\|\mathbf{x}\|}, \\
\nabla_{\mathbf{x}}\|\mathbf{A}\mathbf{x}\|^2 &= 2\mathbf{A}^\top \mathbf{A}\mathbf{x}, \\
\nabla_{\mathbf{x}}(\mathbf{x}^\top A^\top b) &= A^\top b, \\
\nabla_{\mathbf{x}}(b^\top A \mathbf{x}) &= A^\top b.
\end{aligned}$$

Activation Functions — Definitions and Derivatives

1. Sigmoid (Logistic)

Definition:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Derivative (scalar):

$$\sigma'(x) = \frac{d}{dx}\sigma(x) = \sigma(x)\big(1 - \sigma(x)\big).$$

Derivation:

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) = \sigma(x)\big(1-\sigma(x)\big).$$

2. Hyperbolic tangent (tanh)

Definition:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Derivative:

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x).$$

Derivation:

$$\tanh'(x) = \operatorname{sech}^2(x) = 1 - \tanh^2(x).$$

3. ReLU (Rectified Linear Unit)

Definition:

$$\mathrm{ReLU}(x) = \max(0, x).$$

Derivative (subgradient):

$$\mathrm{ReLU}'(x) = \begin{cases} 1, & x > 0, \\ 0, & x < 0, \\ \text{undefined (choose any } g \in [0,1]), & x = 0. \end{cases}$$

(Use the subgradient value 0 or 1 at $x = 0$ as convention in implementations.)

4. Leaky ReLU

Definition (slope $\alpha \in (0,1)$ small):

$$\mathrm{LReLU}(x) = \begin{cases} x, & x \geq 0, \\ \alpha x, & x < 0. \end{cases}$$

Derivative:

$$\mathrm{LReLU}'(x) = \begin{cases} 1, & x > 0, \\ \alpha, & x < 0, \\ \text{(choose } 1 \text{ or } \alpha) \text{ at } x = 0. \end{cases}$$

5. ELU (Exponential Linear Unit)

Definition (parameter $\alpha > 0$):

$$\mathrm{ELU}(x) = \begin{cases} x, & x \geq 0, \\ \alpha(e^x - 1), & x < 0. \end{cases}$$

Derivative:

$$\mathrm{ELU}'(x) = \begin{cases} 1, & x \geq 0, \\ \alpha e^x, & x < 0. \end{cases}$$

6. Softplus

Definition:

$$\mathrm{softplus}(x) = \log\left(1 + e^x\right).$$

Derivative:

$$\frac{d}{dx}\mathrm{softplus}(x) = \frac{e^x}{1+e^x} = \sigma(x).$$

(softplus is a smooth approximation to ReLU)

7. Softmax (vector $\mathbf{z} \in \mathbb{R}^K$ to probabilities $\mathbf{s} \in \Delta^{K-1}$)

Definition (component form):

$$s_i(\mathbf{z}) = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}, \qquad i = 1, \dots, K.$$

Jacobian (matrix of partial derivatives):

$$\frac{\partial s_i}{\partial z_j} = s_i(\delta_{ij} - s_j),$$

or in matrix form for Jacobian $J \in \mathbb{R}^{K \times K}$:

$$J = \mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top.$$

Derivation (sketch):

$$\partial_{z_j} s_i = \frac{\delta_{ij} e^{z_i} \sum_k e^{z_k} - e^{z_i} e^{z_j}}{(\sum_k e^{z_k})^2} = s_i(\delta_{ij} - s_j).$$

8. Softmax combined with Cross-Entropy (common simplification)

Let target one-hot vector $\mathbf{y} \in \{0,1\}^K$ and loss

$$L(\mathbf{z}) = -\sum_i y_i \log s_i(\mathbf{z}).$$

Gradient w.r.t logits $\mathbf{z}$ (single example):

$$\nabla_{\mathbf{z}} L = \mathbf{s} - \mathbf{y}.$$

(This is why softmax + categorical cross-entropy is numerically stable and simplifies backprop.)

9. Binary cross-entropy w.r.t. logit (sigmoid case)

For scalar logit z, sigmoid $p = \sigma(z)$, and label $y \in \{0,1\}$,

$$L(z) = -\big(y \log p + (1-y)\log(1-p)\big).$$

Gradient:

$$\frac{dL}{dz} = p - y = \sigma(z) - y.$$

Derivation: use $\frac{dp}{dz} = \sigma(1-\sigma)$ and chain rule; simplifies to $p - y$.

Frequently Used Gradients in Machine Learning

1. Mean Squared Error (MSE) Loss

Given data (x_i, y_i), prediction $\hat{y}_i = w^\top x_i$:

$$L = \frac{1}{2n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{2n}\|Xw - y\|^2.$$

Gradient:

$$\nabla_w L = \frac{1}{n} X^\top (Xw - y).$$

Derivation:

$$\frac{\partial}{\partial w}\frac{1}{2}\|Xw - y\|^2 = \frac{1}{2}\cdot 2X^\top(Xw - y) = X^\top(Xw - y).$$

—

2. Mean Absolute Error (MAE)

$$L = \frac{1}{n}\sum_i |y_i - w^\top x_i|.$$

Subgradient:

$$\nabla_w L = -\frac{1}{n}\sum_i \operatorname{sign}(y_i - w^\top x_i)x_i.$$

(Note: not differentiable at 0, use subgradient.)

—

3. Binary Cross-Entropy (BCE)

For binary classification with sigmoid output $p_i = \sigma(w^\top x_i)$:

$$L = -\frac{1}{n}\sum_i \left[y_i \log p_i + (1-y_i)\log(1-p_i)\right].$$

Gradient:

$$\nabla_w L = \frac{1}{n} X^\top (p - y),$$

where $p = \sigma(Xw)$. Derivation uses $\frac{dp}{dz} = p(1-p)$ and chain rule.

—

4. Softmax Cross-Entropy (Multiclass CE)

For logits $Z = XW \in \mathbb{R}^{n\times K}$, softmax outputs

$$s_{ik} = \frac{e^{z_{ik}}}{\sum_j e^{z_{ij}}}, \quad L = -\frac{1}{n}\sum_i \sum_k y_{ik}\log s_{ik}.$$

Gradient w.r.t. weights W:

$$\nabla_W L = \frac{1}{n} X^\top (S - Y),$$

where S and Y are $n \times K$ matrices of predicted and true probabilities.

—

5. Hinge Loss (SVM)

For binary labels $y_i \in \{-1, +1\}$:

$$L = \frac{1}{n}\sum_i \max(0, 1 - y_i w^\top x_i).$$

Subgradient:

$$\nabla_w L = -\frac{1}{n}\sum_i y_i x_i \mathbb{1}(y_i w^\top x_i < 1).$$

—

6. Negative Log-Likelihood (General Form)

For likelihood $p(y|x, \theta)$:

$$L(\theta) = -\sum_i \log p(y_i|x_i, \theta).$$

Gradient:

$$\nabla_\theta L = -\sum_i \frac{1}{p(y_i|x_i, \theta)} \frac{\partial p(y_i|x_i, \theta)}{\partial \theta}.$$

Example: Gaussian case below.

—

7. Gaussian Negative Log-Likelihood

For $y_i \sim \mathcal{N}(x_i^\top w, \sigma^2)$:

$$L(w) = \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{n}{2}\log(2\pi\sigma^2).$$

Gradient:

$$\nabla_w L = \frac{1}{\sigma^2} X^\top (Xw - y).$$

—

8. Logistic Loss (for binary classification)

$$L = \frac{1}{n}\sum_i \log\left(1 + e^{-y_i w^\top x_i}\right).$$

Gradient:

$$\nabla_w L = -\frac{1}{n}\sum_i \frac{y_i x_i}{1 + e^{y_i w^\top x_i}} = \frac{1}{n} X^\top (\sigma(-y \odot Xw) - \mathbf{1}) \odot (-y).$$

Simplifies to $\frac{1}{n} X^\top (p - y)$ with $p = \sigma(Xw)$ if $y \in \{0, 1\}$.

—

9. Generalized Cross-Entropy Relation

For discrete distribution targets y and predictions p:

$$H(y, p) = H(y) + D_{KL}(y\|p),$$

hence minimizing cross-entropy $\equiv$ minimizing KL divergence.

Linear Models and Their Gradients

1. Ordinary Least Squares (OLS)

Model: $y = Xw + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2 I)$.
Loss:

$$L(w) = \frac{1}{2}\|Xw - y\|^2.$$

Gradient:

$$\nabla_w L = X^\top (Xw - y).$$

Setting gradient to zero:

$$X^\top X w = X^\top y \quad \Rightarrow \quad \hat{w} = (X^\top X)^{-1} X^\top y.$$

—

2. Ridge Regression (L2 Regularization)

$$L(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

Gradient:

$$\nabla_w L = X^\top (Xw - y) + \lambda w.$$

Normal equation:

$$(X^\top X + \lambda I)w = X^\top y \quad \Rightarrow \quad \hat{w} = (X^\top X + \lambda I)^{-1} X^\top y.$$

—

3. Lasso Regression (L1 Regularization)

$$L(w) = \frac{1}{2}\|Xw - y\|^2 + \lambda \|w\|_1.$$

Subgradient:

$$\nabla_w L = X^\top (Xw - y) + \lambda\ \mathrm{sign}(w).$$

(No closed-form solution; solved via coordinate descent or soft thresholding.)

—

4. Logistic Regression

Hypothesis: $p(y = 1|x) = \sigma(w^\top x)$.
Loss (negative log-likelihood):

$$L(w) = -\sum_i [y_i \log \sigma(w^\top x_i) + (1 - y_i)\log(1 - \sigma(w^\top x_i))].$$

Gradient:

$$\nabla_w L = X^\top (\sigma(Xw) - y).$$

Hessian (for Newton's method):

$$H = X^\top D X, \quad D = \mathrm{diag}(p_i(1 - p_i)).$$

—

5. Linear Discriminant Analysis (LDA) — key formulas

Assuming Gaussian class-conditional densities:

$$p(x|y=k) = \mathcal{N}(\mu_k, \Sigma), \quad p(y=k) = \pi_k.$$

Decision boundary is linear:

$$\delta_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k.$$

Prediction: $\hat{y} = \arg\max_k \delta_k(x)$.

—

6. Regularized Logistic Regression

Add L2 term:

$$L(w) = -\sum_i [y_i \log p_i + (1-y_i)\log(1-p_i)] + \frac{\lambda}{2}\|w\|^2.$$

Gradient:

$$\nabla_w L = X^\top (p - y) + \lambda w.$$

—

7. Maximum Likelihood Connection

OLS and logistic regression can both be derived from MLE:

$$\text{OLS: } \epsilon_i \sim \mathcal{N}(0, \sigma^2) \Rightarrow \max_w p(y|X, w) \iff \min_w \|Xw - y\|^2.$$

$$\text{Logistic: } p(y|x, w) = \sigma(w^\top x)^y (1 - \sigma(w^\top x))^{1-y}.$$

—

8. Closed-form vs Gradient-based Solutions

- OLS, Ridge — closed form (normal equations).
- Lasso — subgradient/iterative methods.
- Logistic — no closed form, use GD, SGD, or Newton.

—

9. Gradient Descent Update (for any linear model)

$$w_{t+1} = w_t - \eta \nabla_w L(w_t),$$

e.g. for MSE:

$$w_{t+1} = w_t - \eta X^\top (X w_t - y).$$

Probabilistic Machine Learning Basics

1. Fundamental Idea

In probabilistic ML, we model the conditional distribution $p(y|x,\theta)$ with parameters θ. We estimate θ using:

$$\text{Maximum Likelihood (MLE): } \quad \hat{\theta}_{\mathsf{MLE}} = \arg\max_\theta \prod_{i=1}^n p(y_i|x_i, \theta) = \arg\max_\theta \sum_{i=1}^n \log p(y_i|x_i, \theta).$$

$$\text{Maximum A Posteriori (MAP): } \quad \hat{\theta}_{\mathsf{MAP}} = \arg\max_\theta p(\theta|D) = \arg\max_\theta \big[\log p(D|\theta) + \log p(\theta)\big].$$

—

2. Log-Likelihood and Negative Log-Likelihood

Given i.i.d. data $\{(x_i, y_i)\}$:

$$\ell(\theta) = \log p(D|\theta) = \sum_{i=1}^n \log p(y_i|x_i, \theta), \qquad L(\theta) = -\ell(\theta) = -\sum_i \log p(y_i|x_i, \theta).$$

Minimizing $L(\theta)$ = maximizing likelihood.

—

3. Gaussian Likelihood and Connection to Linear Regression

Assume $y_i \sim \mathcal{N}(x_i^\top w, \sigma^2)$. Then:

$$p(y|X, w, \sigma^2) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - x_i^\top w)^2}{2\sigma^2}\right].$$

Log-likelihood:

$$\ell(w) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|Xw - y\|^2.$$

Maximizing $\ell(w)$ w.r.t $w \Leftrightarrow$ minimizing MSE loss.

—

4. Bernoulli Likelihood and Logistic Regression

Assume $y_i \in \{0, 1\}$, with $p(y_i = 1|x_i, w) = \sigma(w^\top x_i)$:

$$p(y_i|x_i, w) = \sigma(w^\top x_i)^{y_i} (1 - \sigma(w^\top x_i))^{1-y_i}.$$

Log-likelihood:

$$\ell(w) = \sum_i \big[y_i \log \sigma(w^\top x_i) + (1-y_i)\log\big(1 - \sigma(w^\top x_i)\big)\big].$$

Negative log-likelihood:

$$L(w) = -\ell(w) = -\sum_i \big[y_i \log p_i + (1-y_i)\log(1-p_i)\big].$$

Gradient:

$$\nabla_w L = X^\top(\sigma(Xw) - y).$$

—

5. Bayesian Parameter Estimation

Using Bayes' theorem:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}.$$

MAP estimate maximizes posterior:

$$\hat{\theta}_{\text{MAP}} = \arg\max_\theta \big[\log p(D|\theta) + \log p(\theta)\big].$$

Example (Gaussian prior on w):

$$p(w) = \mathcal{N}(0, \tau^2 I) \Rightarrow \log p(w) \propto -\frac{1}{2\tau^2}\|w\|^2.$$

MAP cost:

$$L_{\text{MAP}}(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2, \quad \lambda = \frac{\sigma^2}{\tau^2}.$$

Hence, Ridge Regression = MAP under Gaussian prior.

—

6. Exponential Family of Distributions

A distribution is in the exponential family if:

$$p(x|\eta) = h(x)\exp\left(\eta^\top T(x) - A(\eta)\right),$$

where η = natural parameter, $T(x)$ = sufficient statistic, and $A(\eta)$ = log-partition function.

Examples:

$$\text{Bernoulli: } T(x) = x, \ \eta = \log\frac{p}{1-p}, \ A(\eta) = \log(1 + e^\eta).$$

$$\text{Gaussian: } T(x) = (x, x^2), \ \eta = (\mu/\sigma^2, -1/(2\sigma^2)).$$

—

7. KL Divergence and Cross-Entropy

Definition:

$$D_{KL}(p\|q) = \int p(x)\log\frac{p(x)}{q(x)}dx.$$

Relation to cross-entropy:

$$H(p, q) = H(p) + D_{KL}(p\|q), \quad H(p, q) = -\mathbb{E}_p[\log q(x)].$$

For classification:

$$L = -\sum_i y_i \log p_i \quad \Rightarrow \quad L = H(y) + D_{KL}(y\|p).$$

Minimizing cross-entropy ⇔ minimizing KL divergence.

—

8. Maximum Likelihood as Minimizing KL Divergence

MLE equivalently minimizes:

$$\theta^* = \arg\min_\theta D_{KL}(p_{\text{data}}(x) \,\|\, p_\theta(x)).$$

Proof (sketch):

$$D_{KL}(p_{\text{data}}\|p_\theta) = \mathbb{E}_{p_{\text{data}}}[\log p_{\text{data}}(x)] - \mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)].$$

Since first term constant w.r.t. θ,

$$\arg\min_\theta D_{KL} = \arg\max_\theta \mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)].$$

—

9. Conditional Likelihood and Discriminative Models

For discriminative modeling:

$$p(y|x, \theta) = \frac{p(x, y|\theta)}{p(x)}.$$

Only conditional term matters for training since $p(x)$ is independent of θ. Hence logistic regression and neural nets are discriminative (model $p(y|x)$), while Naïve Bayes is generative (model $p(x, y)$).

—

10. Common Probabilistic Identities

$$\mathbb{E}[X] = \int xp(x)\,dx, \quad \text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2.$$

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x).$$

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

$$\text{Law of total probability: } p(x) = \sum_y p(x|y)p(y).$$

—

11. Common Log-Likelihood Gradients

$$\text{Gaussian: } \nabla_\mu \log p(x) = \Sigma^{-1}(x-\mu),$$
$$\text{Exponential: } \nabla_\lambda \log p(x) = \frac{1}{\lambda} - x,$$
$$\text{Bernoulli: } \nabla_p \log p(x) = \frac{x}{p} - \frac{1-x}{1-p}.$$

—

12. Summary Table (at a glance)

Concept	Objective	Equivalent Form
MLE	$\max_\theta p(D\|\theta)$	$\min_\theta -\log p(D\|\theta)$
MAP	$\max_\theta p(D\|\theta)p(\theta)$	$\min_\theta[-\log p(D\|\theta) - \log p(\theta)]$
Ridge	Gaussian prior on w	L2 penalty
Lasso	Laplace prior on w	L1 penalty
Cross-Entropy	$\min D_{KL}(y\\|p_\theta)$	supervised classification loss