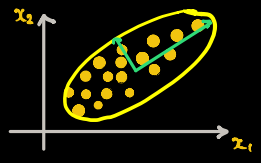
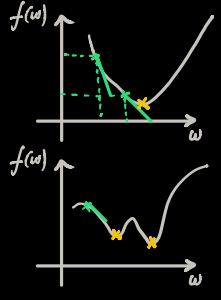
$\mathcal{M} \infty \mathcal{L}_4$

1. Recap on data covariance Σ
   - data shift
   - linear & nonlinear Bayes classifier

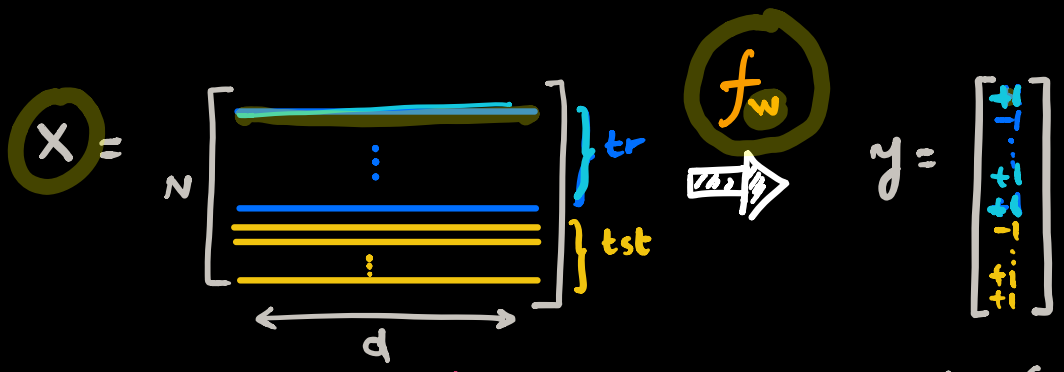


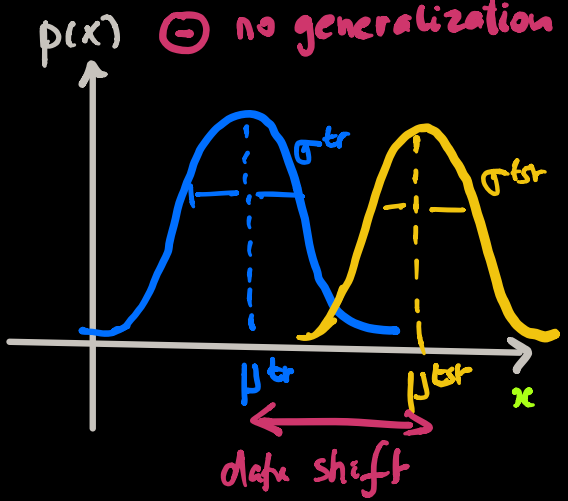
2. Fundamentals of numerical optimization $[\min_{w} \ell(w)]$

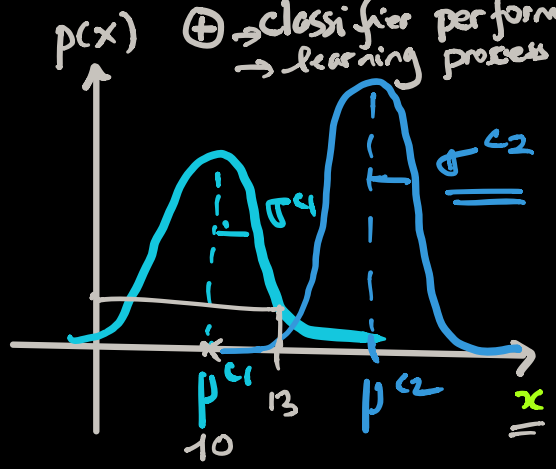


- Taxonomy of supervised learning
- Taylor approximation of a function
- Stationary points
- Convex and non-convex functions
- gradient descent

$\mathcal{M} \infty \mathcal{L}_3 \rightarrow ?$



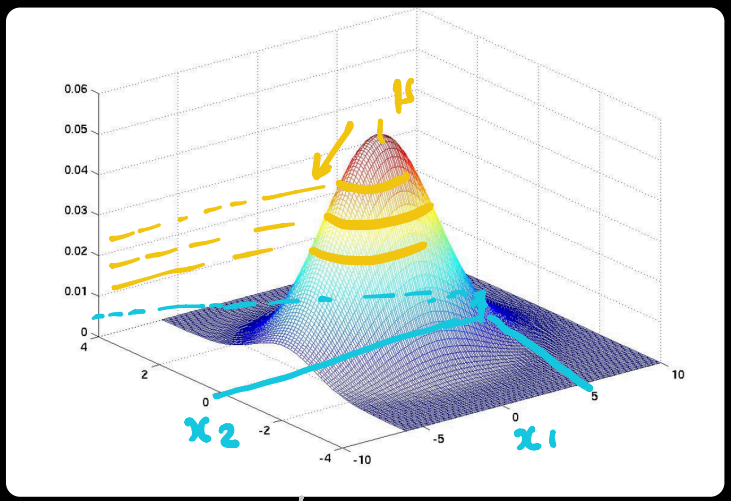


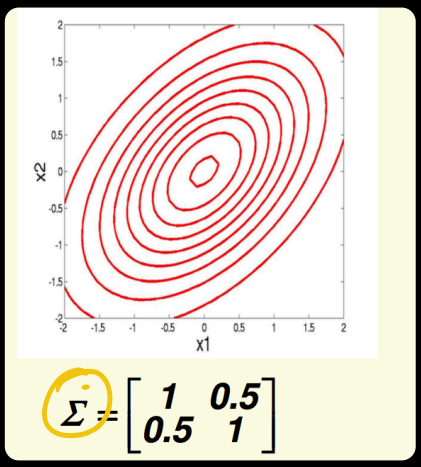


$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-1}{2} \frac{(x-\mu)^2}{\sigma}\right]$$

$p(x = 13)$ ← the probability of feature x taking value 13?

$x \in \mathbb{R}^d$, multivate distribution

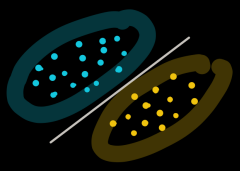




$$p(\vec{x}) = p(x_1, x_2, \dots, x_d) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{1/2}} \exp\left[\frac{-1}{2} (x-\mu)^t \Sigma^{-1} (x-\mu)\right]$$

$\vec{x} \in \mathbb{R}^d$

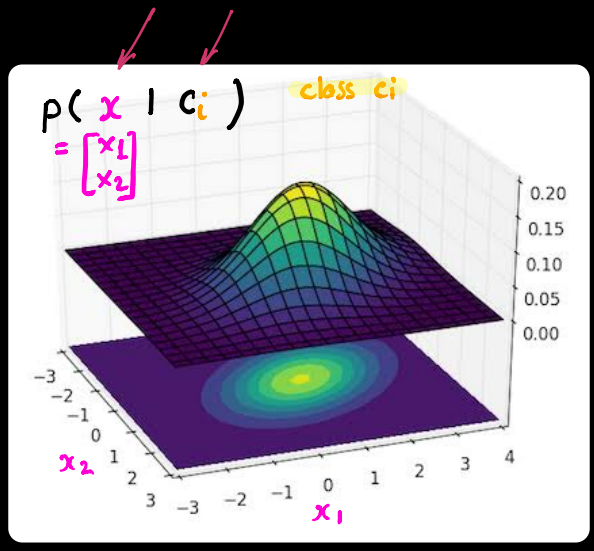
Bayes discriminant classifier



- new testing sample x ↓ what is the probability of x belonging to class c_i ?

$g_i(x) = P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)}$

- $g_i(x)$: class i
- $P(c_i|x)$: conditional probability, posterior ↓ we want to predict
- $P(x|c_i)$: likelihood of observing x given c_i class
- $P(c_i)$: prior "probability of belonging to class c_i"
- $P(x)$: marginal likelihood or probability of observing x

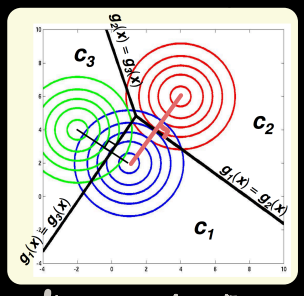


$\Sigma_i = \sigma^2 I$

(different means μ_i but equal variances)

$g_i(x) = w_i^t x + w_{io}$

(linear in x)



lines connecting the classes means are ⊥ to the decision boundaries

special case $\ln p(c_i) = \ln p(c_j)$

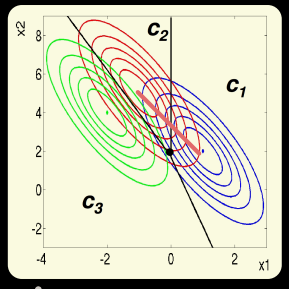
$g_i(x) = -\|x - \mu_i\|_2^2$

$\Sigma_i = \Sigma_j = \Sigma$

(arbitrary covariance but constant across classes)

$g_i(x) = w_i^t x + w_{io}$

(linear in x)



lines connecting the classes means are ⊥ to the decision boundaries

special case $\ln p(c_i) = \ln p(c_j)$

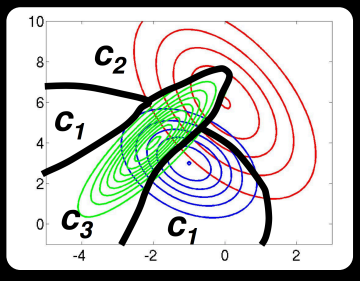
$g_i(x) = -\frac{1}{2}\|x - \mu_i\|^2_{\Sigma^{-1}}$

$(\Sigma_1, \ldots, \Sigma_k)$

(k covariance matrices, general case)

$g_i(x) = x^t W x + w_i^t x + w_{io}$

(quadratic in x)



decision boundaries between classes are non-linear.

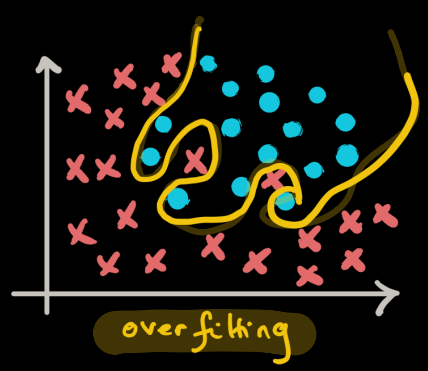
Mathematical notation	Definition
$\mathcal{D}$	dataset
n	number of samples in a dataset $\mathcal{D}$
d	number of features
$\mathbf{x} \in \mathbb{R}^{d \times 1}$	feature vector or data point (sample)
$\mathbf{x}^{sample}_{feature}$	–
$\mathbf{x}^i \in \mathbb{R}^{d \times 1}$	i^{th} sample in the population
$\mathbf{x}^i_j \in \mathbb{R}$	j^{th} feature of i^{th} sample in the population
$\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$	covariance matrix of data population $\{\mathbf{x}^i\}_{i=1}^n$
$\lvert\mathbf{A}\rvert \in \mathbb{R}$	determinant of matrix A
$p(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{-1}{2}\frac{\lvert\lvert x-\mu\rvert\rvert_2^2}{\sigma^2})$	probability density function of a variable $x \in \mathbb{R}$
$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\lvert\mathbf{\Sigma}\rvert^{1/2}} exp[\frac{-1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)]$	probability density function of a multidimensional variable $\mathbf{x} \in \mathbb{R}^{N \times 1}$
$\mu \in \mathbb{R}^{d \times 1}$	sample mean $\mu = \frac{1}{n}\sum_{i=1}^n \mathbf{x}^i$
$\lvert\lvert\mathbf{x}-\mu\rvert\rvert_{\mathbf{\Sigma}^{-1}} = (\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) \in \mathbb{R}$	Mahalanobis distance between $\mathbf{x}$ and μ
$\mathbf{I}_{d \times d} \in \mathbb{R}^{d \times d}$	identify matrix of size d ×d
$\lvert\lvert\mathbf{x}-\mu\rvert\rvert_{\mathbf{I}_{d\times d}} = (\mathbf{x}-\mu)^T(\mathbf{x}-\mu) \in \mathbb{R}$	Euclidean distance between $\mathbf{x}$ and μ also noted as L_2 norm $\lvert\lvert\cdot\rvert\rvert_2$
$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	discriminant Bayes function for class i when $\mathbf{\Sigma}_i = \sigma^2\mathbf{I}$ (i.e., different means μ_i and equal data variances across classes) (i.e., lines connecting means of different classes are perpendicular to decision boundaries) if $ln(p(c_i)) = ln(p(c_j))$, $g_i(\mathbf{x}) = -\lvert\lvert\mathbf{x}-\mu_i\rvert\rvert_2^2$
$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	discriminant Bayes function for class i when $\mathbf{\Sigma}_i = \mathbf{\Sigma}_j = \mathbf{\Sigma}$ (i.e., constant data covariance $\mathbf{\Sigma}$ across classes) (i.e., lines connecting means of different classes are not perpendicular to decision boundaries) if $ln(p(c_i)) = ln(p(c_j))$, $g_i(\mathbf{x}) = -\frac{1}{2}\lvert\lvert\mathbf{x}-\mu_i\rvert\rvert_{\mathbf{\Sigma}^{-1}}^2$
$g_i(\mathbf{x}) = \mathbf{x}^T\mathbf{W}\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	quadratic discriminant function (decision boundaries are nonlinear)

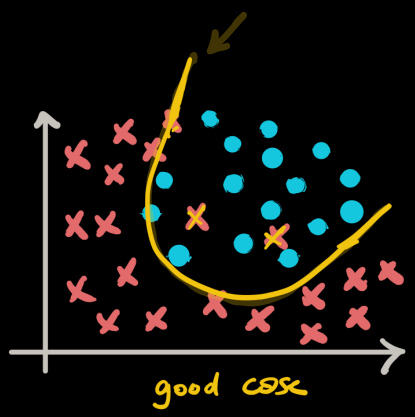
Taxonomy of supervised ML

$$\arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}^i, y^i) \in \mathcal{D}} E(f_{\mathbf{w}}(\mathbf{x}^i), y^i) + R(\dots)$$

- $\mathcal{L}(\mathbf{w})$: loss function
- $\mathbf{w}$: set of parameters to estimate
- E: error function
- $f_{\mathbf{w}}(\mathbf{x}^i)$: predicted output by the learner $f_{\mathbf{w}}$ for input $\mathbf{x}^i$
- y^i: ground truth observation
- $E(f_{\mathbf{w}}(\mathbf{x}^i), y^i)$: fidelity to data term
- $R(\dots)$: regularization term — controls the complexity of the model/learner + avoid overfitting

$$d\left(\hat{y}^i = \tilde{y}^i, y^i\right)$$

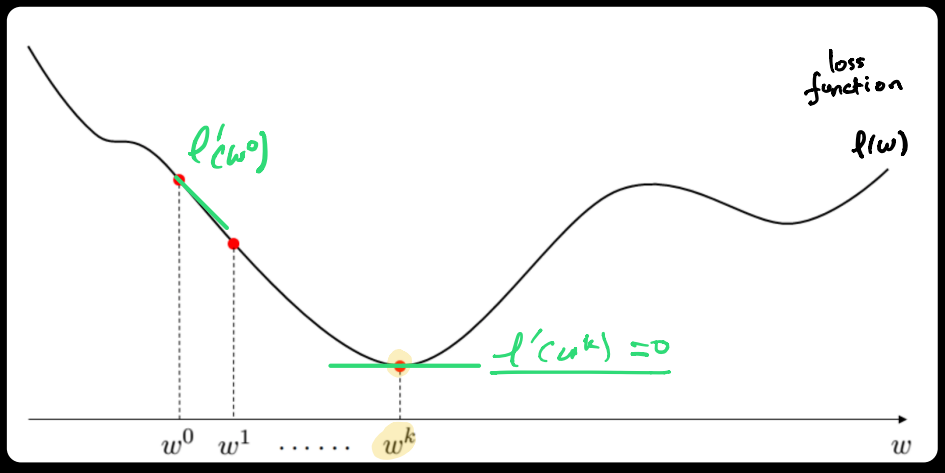




$\mathcal{I} \heartsuit M_x^i$ [2]

$\arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}^i, y^i) \in \mathcal{D}} E(f_{\mathbf{w}}(\mathbf{x}^i), y^i) + R(\dots)$	supervised learning energy cost (loss function)
f	the mapping function to learn from $\mathbf{x}^i \rightarrow y^i$
E	the error function between the predicted target by f and the ground truth observation y^i
R	regularization term to avoid overfitting and control model complexity
$\mathbf{w}$	optimization parameters (weight vector)
	set of parameters that minimize the loss functional $\mathcal{L}(\mathbf{w})$

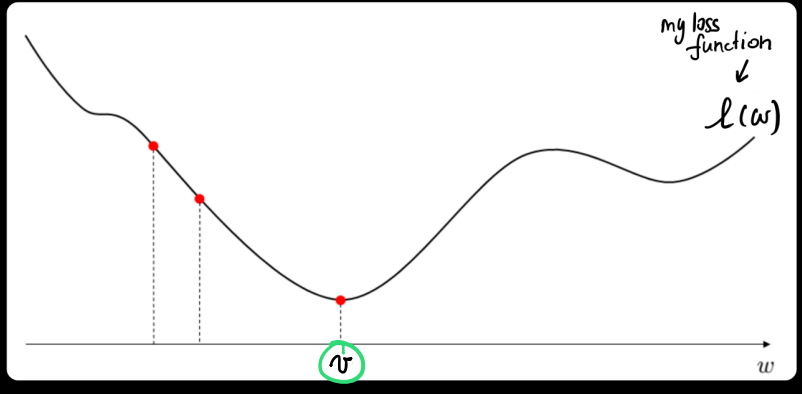
→ Goal: find the optimal parameters w for a loss function ℓ.



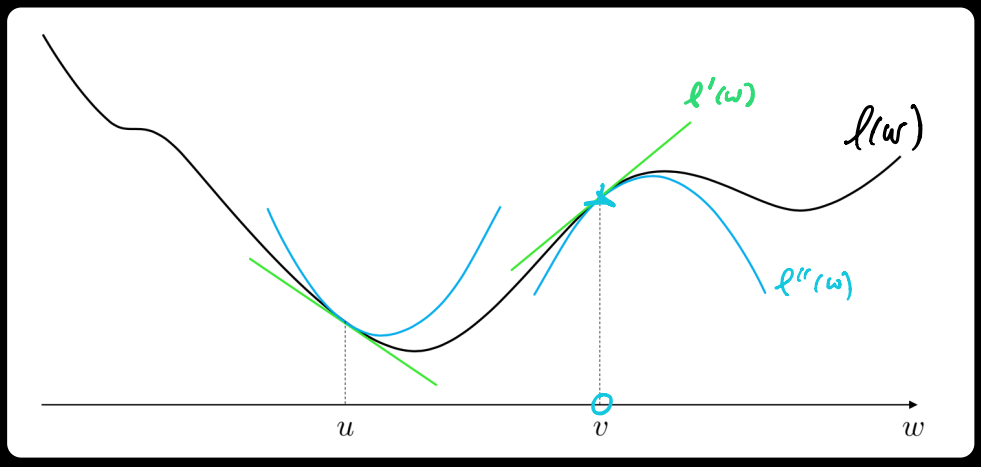
① what are we looking for ?

② How to find what we are looking for ?
(hint: " check the local geometry of $\ell(w)$ " ☺

How calculus can help us read the geometry of a function at different points ?



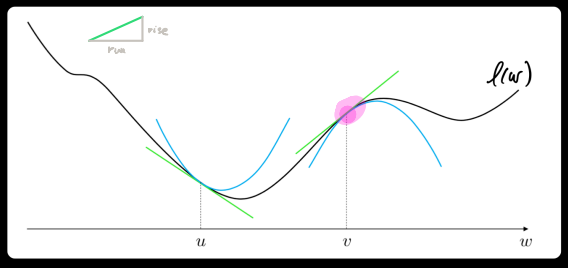
- 1-dimensional loss function $\ell(w)$, $w \in \mathbb{R}$, $\ell(w) \in \mathbb{R}$
- To capture the geometry of $\ell(w)$ at a point v, we can linearly or nonlinearly approximate it near this point.



(1D) univarite function ℓ in w, $w \in \mathbb{R}$

First-order Taylor approximation

"linear" approximation



- linear approximation of $\ell(w)$ at two points $\begin{cases} w = u \\ w = v \end{cases}$

$$h(w) = \ell(v) + \ell'(v)\,(w - v)$$

↑ linear in w

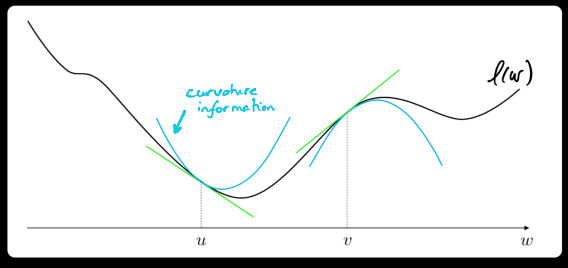
$$\begin{cases} h(v) = \ell(v) \\ h'(v) = \ell'(v) \end{cases}$$

→ this linear approximation holds well near v because the derivative contains the slope information.

Second-order Taylor approximation

"quadratic" approximation

non-linear



- A second-order Taylor approximation function $h(w)$ at points v located near w contains both first and second derivatives of the target function ℓ to approximate:

$$h(w) = \ell(v) + \ell'(v)\,(w - v) + \frac{1}{2}\,\ell''(v)\,(w - v)^2$$

$(w - v)^2 = w^2 - 2wv + v^2$

for $w = v$

$$\begin{aligned} h(v) &= \ell(v) \\ h'(v) &= \ell'(v) \\ h''(v) &= \ell''(v) \end{aligned}$$

"locally" $\simeq$ around $w = v$

$h(w)$ approximates $\ell(w)$

$$\frac{dh(w)}{dw} = h'(w) = \ell'(v) + \frac{2}{2}\,\ell''(v)\,w - \frac{2}{2}\,\ell''(v)\,v \Leftarrow$$

$$\Rightarrow h'(w = v) = \ell'(v) + 0 = \ell'(v)$$

(ND) multivariate loss function ℓ, $w \in \mathbb{R}^{N\times 1}$, $w = [w_1\ w_2\ \dots\ w_N]^T$

variable w, fixed v

$$h(w) = \ell(v) + \nabla\ell(v)^T (w - v)$$

$\nabla\ell(v) \in \mathbb{R}^N$

$$\nabla\ell(v) = \begin{bmatrix} \frac{\partial}{\partial w_1}\ell(v) \\ \vdots \\ \frac{\partial}{\partial w_N}\ell(v) \end{bmatrix}$$

$\nabla\ell(v)$: gradient of partial derivatives

$$\ell\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$w^T w$

$$= 2w_1 + w_2 + w_1^2 + w_2^2$$

$$\nabla\ell(w) = \begin{bmatrix} \frac{\partial\ell}{\partial w_1} \\ \frac{\partial\ell}{\partial w_2} \end{bmatrix} = \begin{bmatrix} 2 + 0 + 2w_1 + 0 \\ 0 + 1 + 0 + 2w_2 \end{bmatrix} = \begin{bmatrix} 2 + 2w_1 \\ 1 + 2w_2 \end{bmatrix}$$

$$\nabla\ell\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

$$h(w) = \ell(v) + \nabla\ell(v)^T (w - v) + \frac{1}{2}(w - v)^T \nabla^2\ell(v)(w - v)$$

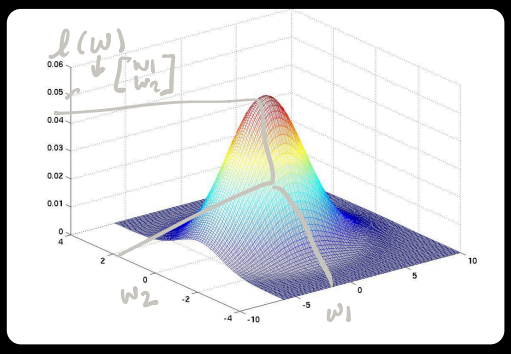
$$\nabla^2\ell(v) = \begin{bmatrix} \frac{\partial^2}{\partial w_1 \partial w_1}\ell(v) & \cdots & \frac{\partial^2}{\partial w_1 \partial w_N}\ell(v) \\ \frac{\partial^2}{\partial w_2 \partial w_1}\ell(v) & \cdots & \frac{\partial^2}{\partial w_2 \partial w_N}\ell(v) \\ \vdots & & \vdots \\ \frac{\partial^2}{\partial w_N \partial w_1}\ell(v) & \cdots & \frac{\partial^2}{\partial w_N \partial w_N}\ell(v) \end{bmatrix} \in \mathbb{R}^{N\times N}$$

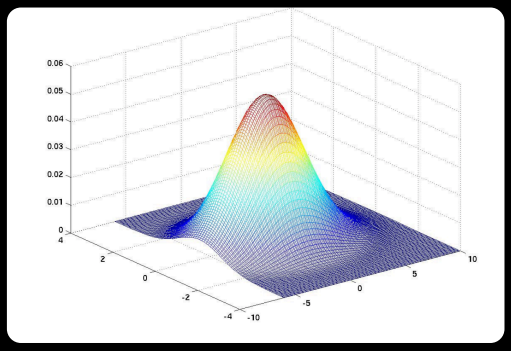
$i \to$ rows, $j \downarrow$ columns

$$H = \begin{bmatrix} & \vdots & \\ \cdots & \frac{\partial^2\ell}{\partial w_i \partial w_j} & \cdots \\ & \vdots & \end{bmatrix}$$

* special case $w \in \mathbb{R}$, $N = 1$

$\ell''(v) = ?$





in 2D : $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$ ← fixed point (evaluate at v)

$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ ← our variable

$$h(w) = \ell\left(\begin{bmatrix} 5 \\ 2 \end{bmatrix}\right) + \begin{bmatrix} \frac{\partial \ell(v)}{\partial w_1} & \frac{\partial \ell(v)}{\partial w_2} \end{bmatrix} \times \begin{bmatrix} w_1 - 5 \\ w_2 - 2 \end{bmatrix}$$

$h(w) \in \mathbb{R}$?

in 2D : $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$ ← fixed point (evaluate at v)

$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ ← our variable

$$h(w) = \ell\left(\begin{bmatrix} \quad \end{bmatrix}\right) + \begin{bmatrix} \quad \end{bmatrix} \times \begin{bmatrix} \quad \end{bmatrix}$$

$$+ \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} \frac{\partial^2 \ell(v)}{\partial w_1 \partial w_1} & \frac{\partial^2 \ell(v)}{\partial w_1 \partial w_2} \\ \frac{\partial^2 \ell(v)}{\partial w_2 \partial w_1} & \frac{\partial^2 \ell(v)}{\partial w_2 \partial w_2} \end{bmatrix} \times \begin{bmatrix} \quad \end{bmatrix}$$

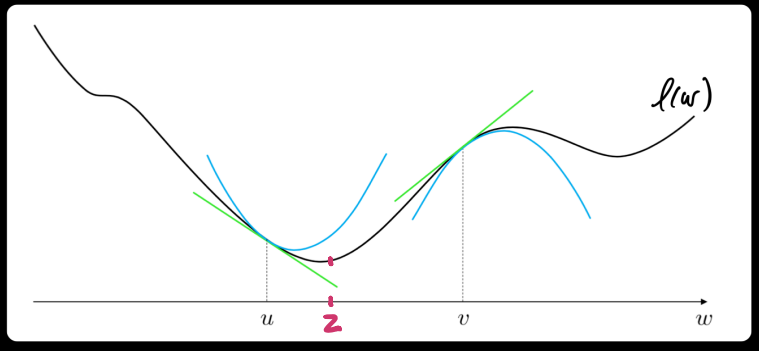
2 →

variable: w — fixed: v

$$h(w) = \underbrace{\ell(v)}_{\in \mathbb{R}} + \underbrace{\nabla \ell(v)^T}_{\uparrow} \underbrace{(w - v)}_{\in \mathbb{R}}$$

$$h(w) = \ell(v) + \nabla \ell(v)^T (w - v) + \frac{1}{2} (w - v)^T \nabla^2 \ell(v) (w - v)$$

First order condition of optimality

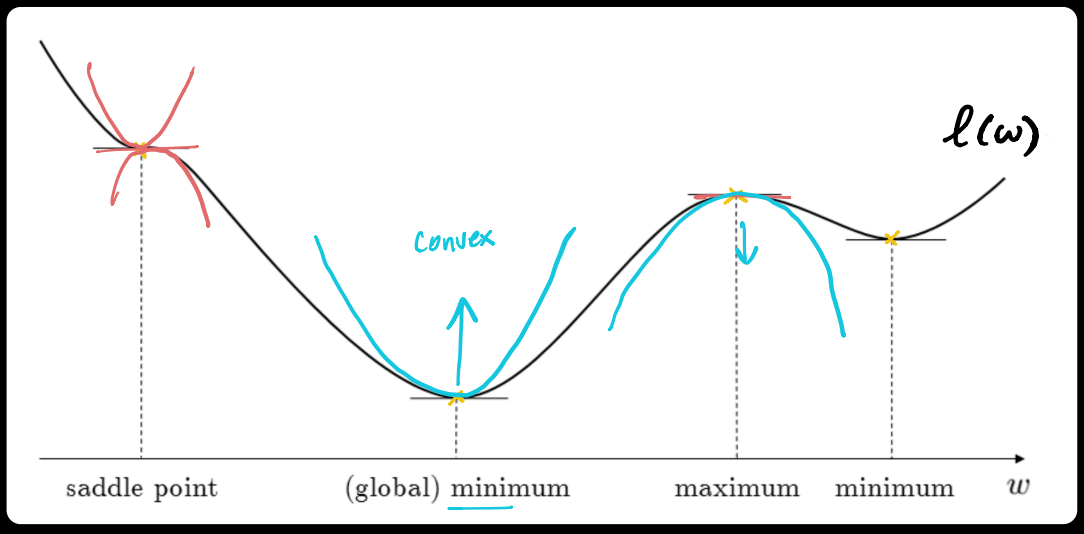


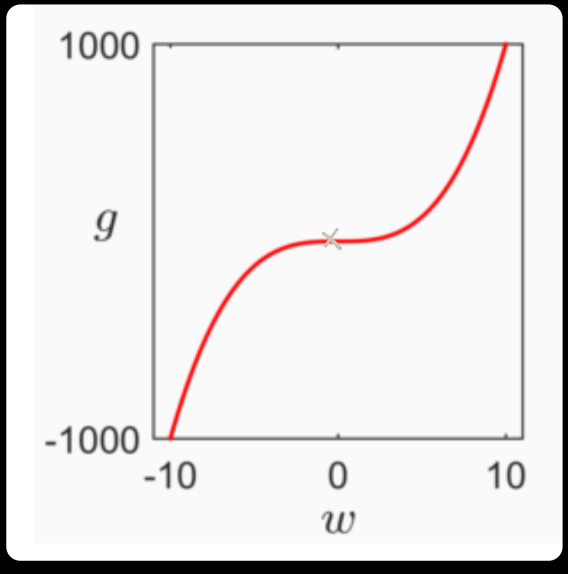
- Minimum values of $\ell(w)$ are located at 'valley floors' where the line (or hyperplane in high dimension) tangents to the function ℓ is flat (slope = 0).

$$\begin{cases} N = 1 \Rightarrow \ell'(z) = 0 \\ N > 1 \Rightarrow \nabla \ell(z) = 0_{N \times 1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \end{cases}$$

$$\nabla \ell(z) = \begin{bmatrix} \frac{\partial \ell}{\partial w_1}(z) \\ \frac{\partial \ell}{\partial w_2}(z) \\ \vdots \\ \frac{\partial \ell}{\partial w_M}(z) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\Rightarrow$ first order optimality





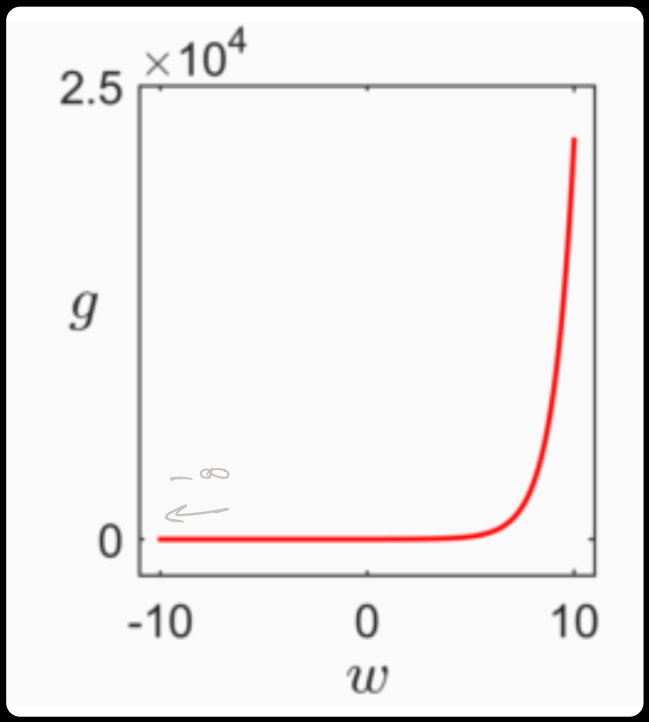
$g(w) = w^3$

$g'(w) = 3w^2$

$g'(w) = 0$

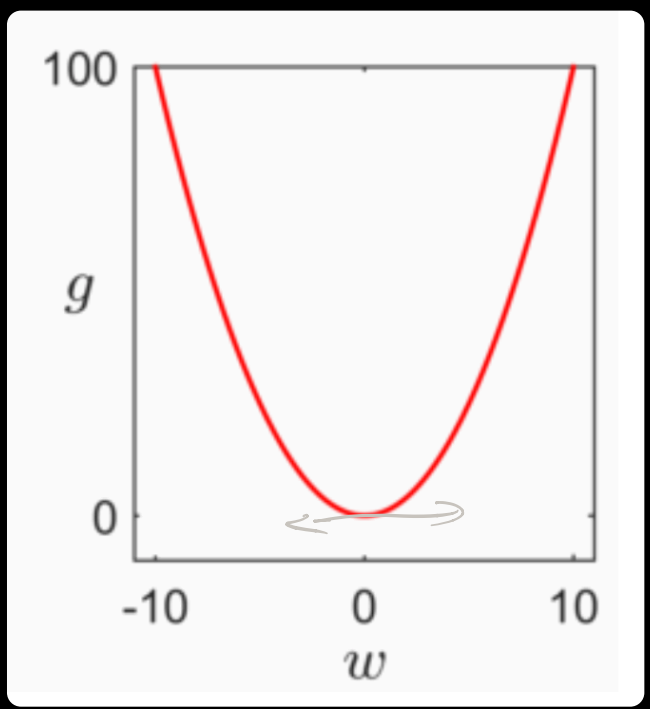
$w^2 = 0$

$\Rightarrow w = 0$



$$g(w) = e^w$$

$$\underline{g'(w) = e^w}$$



$$g(w) = w^2$$

$$g'(w) = 2w$$

$$g'(w) = 0$$

$$\underline{w = 0}$$

$$l(w) = \frac{1}{2} w^T Q w + r^T w + d \quad , \begin{cases} Q \in \mathbb{R}^{N \times N} \\ Q^T = Q \ \text{(Symmetric)} \\ r \in \mathbb{R}^{N \times 1} \\ d \in \mathbb{R} \end{cases}$$

$$\nabla l(w) = Qw + r$$

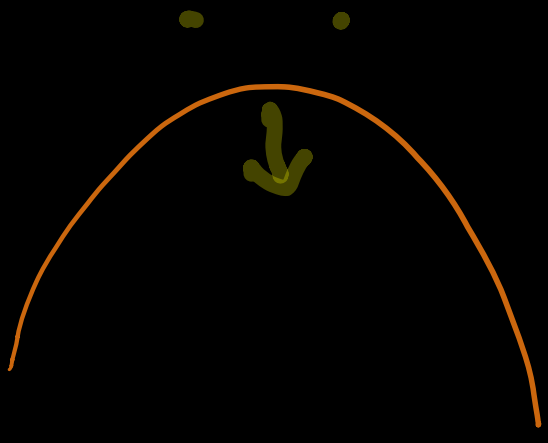
$$\Rightarrow \nabla l(w) = 0 \Rightarrow \boxed{Qw = -r}$$

rules $\begin{cases} \dfrac{\partial a^T x}{\partial x} = \dfrac{\partial x^T a}{\partial x} = a \\ \dfrac{\partial a^T X a}{\partial a} = 2Xa \end{cases}$ ↑ $[X = X^T]$ = symmetric

Second order condition of optimality

- Goal ⇒ find a global minimum.
- Ideal function is **convex** [faces upward ∪] ⇒ no maxima, no saddle points

How is the curvature of this function?



1D

$l''(v) > 0 \Leftrightarrow$ convex

$l''(v) < 0 \Leftrightarrow$ concave

$N > 1$ multivariate loss functions

$w \in \mathbb{R}^{N \times 1}$, $\ell(w)$ is convex if:

$\nabla^2 \ell(w)$ has positive eigenvalues.

Eigenvectors & eigenvalues of a matrix A

- u is an eigenvector of the matrix A if it satisfies:

$$Au = \lambda u$$

$$\Downarrow$$

$$(A - \lambda I)u = 0$$

λ: eigenvalue, u: eigenvector

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}, \quad u = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$Au = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 6+4 \\ 6+2 \end{bmatrix} = 4 \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Example (1st order of a quadratic loss)

$$\ell(w) = \frac{1}{2} w^T Q w + r^T w + d$$

w^*

$Q \in \mathbb{R}^{N \times N}$, $Q^T = Q$ (symmetric), $r \in \mathbb{R}^{N \times 1}$, $d \in \mathbb{R}$

$$\nabla \ell(w) = \frac{2}{2} Q w + r$$

$$\Rightarrow \nabla \ell(w) = 0 \Rightarrow Q w = -r$$

rules:

$$\frac{\partial a^T x}{\partial x} = \frac{\partial x^T a}{\partial x} = a$$

$$\frac{\partial a^T X a}{\partial a} = 2 X a \quad ([X = X^T] = \text{symmetric})$$

Example (2nd order of a quadratic loss)

$$\ell(w) = \frac{1}{2} w^T Q w + r^T w + d$$

$$\rightarrow \nabla \ell(w) = \frac{\partial \ell}{\partial w} = Q w + r$$

$$\rightarrow \nabla^2 \ell(w) = \frac{\partial (Q w + r)}{\partial w}$$

$$\Rightarrow \frac{\partial A x}{\partial x} = \frac{1}{2}(A^T + A)$$

matrix cookbook

$$= \frac{1}{2}(Q^T + Q)$$

$$\nabla^2 \ell(w) = \frac{1}{2}(Q + Q^T) \in \mathbb{R}^{N \times N}$$

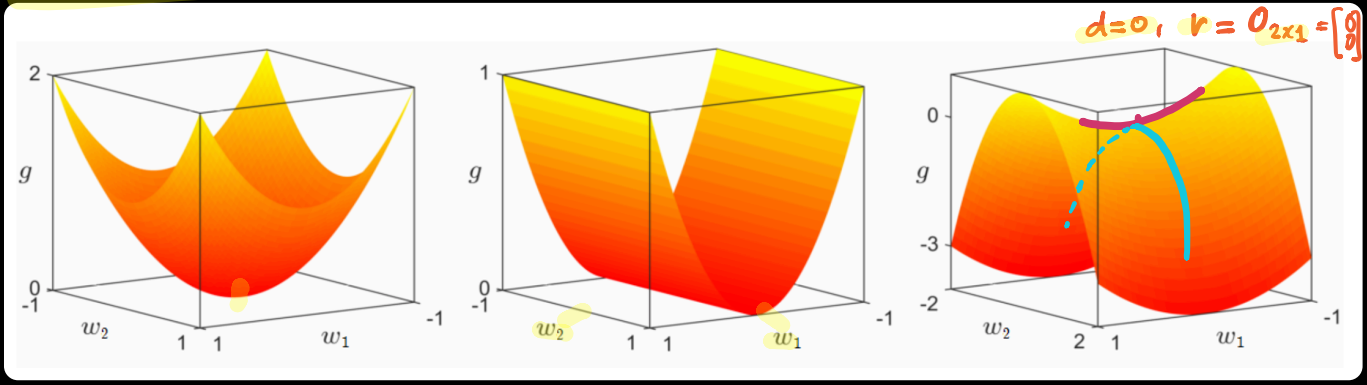
$\downarrow$ if Q is symmetric, $Q = Q^T$

$$\nabla^2 \ell(w) = Q$$

3 examples of convexity in high-dimensional space

$l(w) = w^T Q w$

$d=0, \quad r = 0_{2\times 1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$



$Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

$\nabla^2 l(w) = Q$

$\lambda_1 = \lambda_2 = 2$

$Q = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$

$\nabla^2 l(w) = Q$

$\begin{cases} \lambda_1 = 2 \\ \lambda_2 = 0 \end{cases}$

$Q = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$

$\nabla^2 l(w) = Q$

$\begin{cases} \lambda_1 = 2 \\ \lambda_2 = -2 \end{cases}$

I ♥ M_x^i [3]

Mathematical notation	Definition
$arg\ min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}^i, y^i) \in \mathcal{D}} E(f_{\mathbf{w}}(\mathbf{x}^i), y^i) + R(\dots)$	supervised learning energy cost (loss function)
f	the mapping function to learn from $\mathbf{x}^i \rightarrow y^i$
E	the error function between the predicted target by f and the ground truth observation y^i
R	regularization term to avoid overfitting and control model complexity
$\mathbf{w}$	optimization parameters (weight vector)
	set of parameters that minimize the loss function $\mathcal{L}(\mathbf{w})$
$h(w) = l(v) + l'(v)(w - v)$	first-order Taylor approximation of the loss function l at point w (in 1-dimensional space)
$l'(w)$	first derivative of function l evaluated at point w
$h(w) = l(v) + l'(v)(w - v) + \frac{1}{2} l''(v)(w - v)^2$	second-order Taylor approximation of the loss function l at point w (in 1-dimensional space)
$l''(w)$	second derivative of function l evaluated at point w
$h(\mathbf{w}) = l(\mathbf{v}) + \nabla l(\mathbf{w})(\mathbf{w} - \mathbf{v})$	first-order Taylor approximation of high-dimensional loss function l at vector point $\mathbf{w} \in \mathbb{R}^N$
$\mathbf{w} = [w_1\ w_2 \dots w_N] \in \mathbb{R}^N$	weight vector to learn
$\nabla l(\mathbf{w}) \in \mathbb{R}^{N \times 1}$	gradient vector of the multivariate loss function l at location $\mathbf{w}$
	note that $\nabla l(\mathbf{w})^T$ is $\in \mathbb{R}^{1 \times N}$ (row vector)
	$\nabla l(\mathbf{v}) = [\frac{\partial}{\partial w_1} l(\mathbf{v})\ \frac{\partial}{\partial w_2} l(\mathbf{v}) \dots \frac{\partial}{\partial w_N} l(\mathbf{v})]^T$
$h(\mathbf{w}) = l(\mathbf{v}) + \nabla l(\mathbf{w})(\mathbf{w} - \mathbf{v}) + \frac{1}{2}(\mathbf{w} - \mathbf{v})^T \nabla^2 l(\mathbf{w})(\mathbf{w} - \mathbf{v})$	second-order Taylor approximation of high-dimensional loss function l at vector point $\mathbf{w} \in \mathbb{R}^N$
$\nabla^2 l(\mathbf{w}) \in \mathbb{R}^{N \times N}$	Hessian symmetric matrix of second derivatives of l along all its dimensions (variables)
	l is many times differentiable at the vector valued input $\mathbf{w}$
$l'(w) = 0$	stationary point (min, max or saddle) for a 1-dimensional function
$\nabla l(\mathbf{w}) = \mathbf{0}_{N \times 1}$	stationary point (min, max or saddle) for an N-dimensional function
	(all elements of the gradient are zero)
$\mathbf{Q} = \mathbf{Q}^T$	$\mathbf{Q}$ is symmetric
$\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}$	matrix cookbook
$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{a}} = 2\mathbf{X}\mathbf{a}\ (\mathbf{X}^T = \mathbf{X})$	matrix cookbook
$l''(w) > 0$	convex function (facing upward)
$l''(w) < 0$	concave function (facing downward)
$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \frac{1}{2}(\mathbf{A}^T + \mathbf{A})$	matrix cookbook
$\nabla^2 l(\mathbf{w}) = \frac{1}{2}(\mathbf{Q}^T + \mathbf{Q}) \in \mathbb{R}^{N \times N}$	is the Hessian matrix of l equal to $l(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{Q} \mathbf{w} + \mathbf{r}^T \mathbf{w} + d$
	$d \in \mathbb{R}$ and $\mathbf{r} \in \mathbb{R}^{N \times 1}$
$\nabla^2 l(\mathbf{w}) = \mathbf{Q}\mathbf{w} + \mathbf{r} \in \mathbb{R}^{N \times 1}$	is the gradient vector of l equal to $l(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{Q} \mathbf{w} + \mathbf{r}^T \mathbf{w} + d$

Basic technique 1 for finding $\min_w \ell(w)$: gradient descent

$$w^* = \underset{w}{\text{minimize}}\ \ell(w)$$

minimize ℓ over all input values w

↓

solution = "optimal w"

↓

w^*

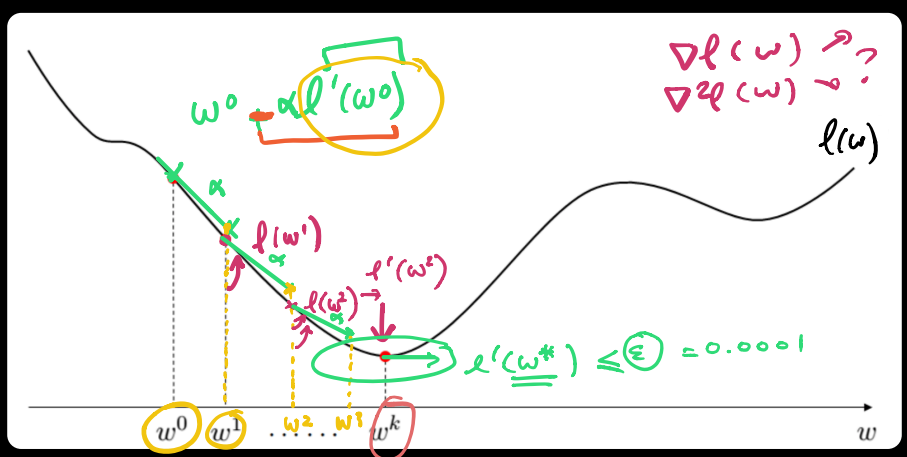
↓

Solve N equations

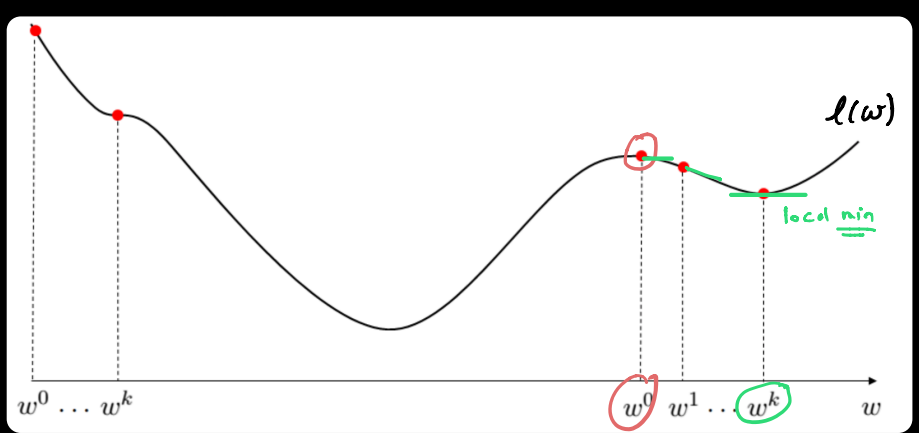
$$\nabla \ell(w) = 0_{N\times 1}$$

choice of w^0

non-convex

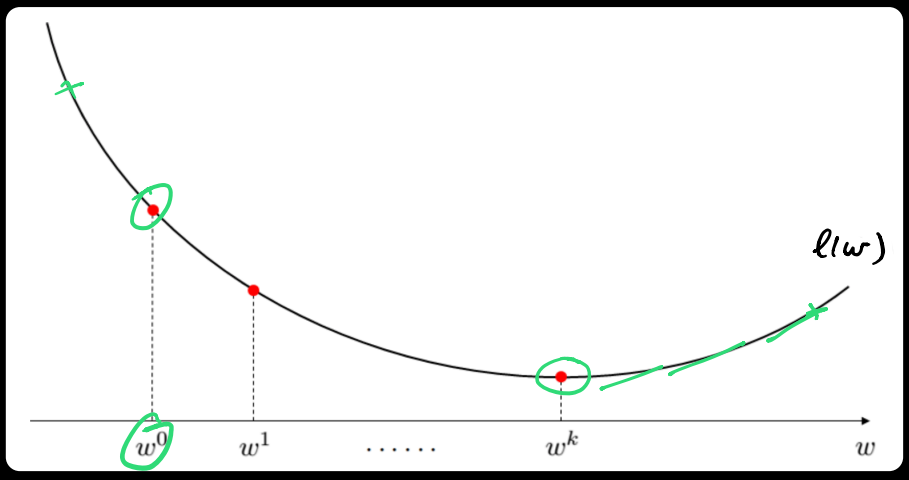


non-convex



influence of w^0 ??

convex or nonconvex ?



(1) Start the minimization process from some *initial point* $\mathbf{w}^0$.

(2) Take *iterative* steps denoted by $\mathbf{w}^1$, $\mathbf{w}^2$, ..., going "downhill" towards a stationary point of $\boldsymbol{\ell}$.

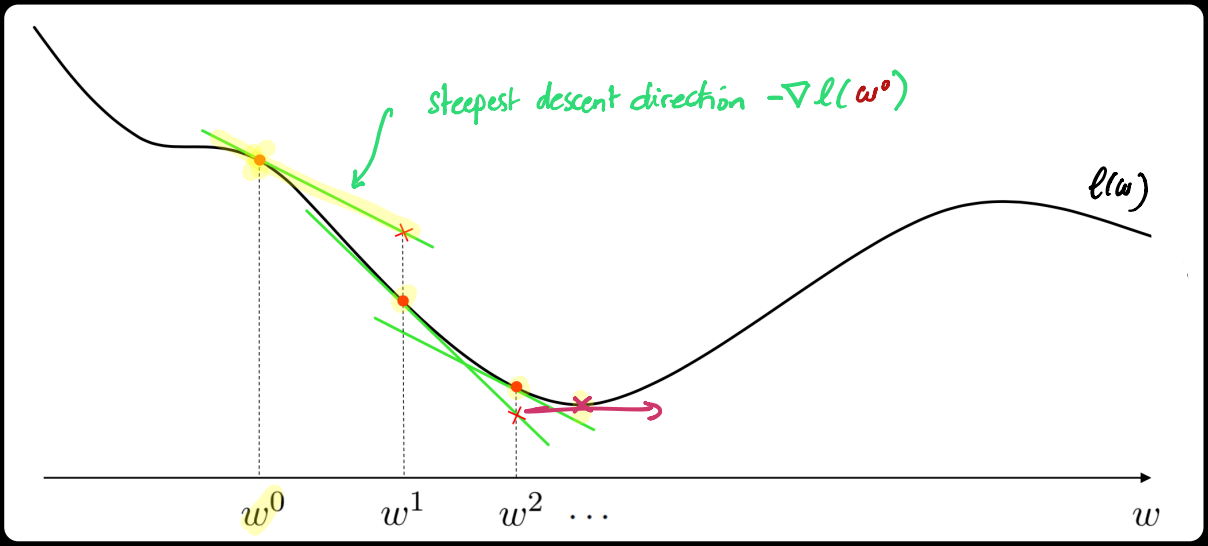
(3) Repeat step (2) until the sequence of points converges to a stationary point of $\boldsymbol{\ell}$.

STOPPING CONDITION

(1) When a pre-specified number of iterations are complete.

(2) When the gradient is small enough, i.e., $\left\|\nabla \boldsymbol{\ell}\left(\mathbf{w}^k\right)\right\|_2 < \epsilon$ for some small $\epsilon > 0$.

Gradient descent



① Travel in the downward direction of a linear approximation
② hop back onto the function
③ repeat in order to find a stationary point of ℓ.

$$* \quad h(w) = \ell(w^0) + \nabla \ell(w^0)^T (w - w_0)$$

→ take a first step of size α_1 downward:

$$\Rightarrow \boxed{w^1} = w^0 - \alpha_1 \nabla \ell(w^0)$$

α_1: learning rate / step length

$$w^2 = w^1 - \alpha_2 \nabla \ell(w^1)$$

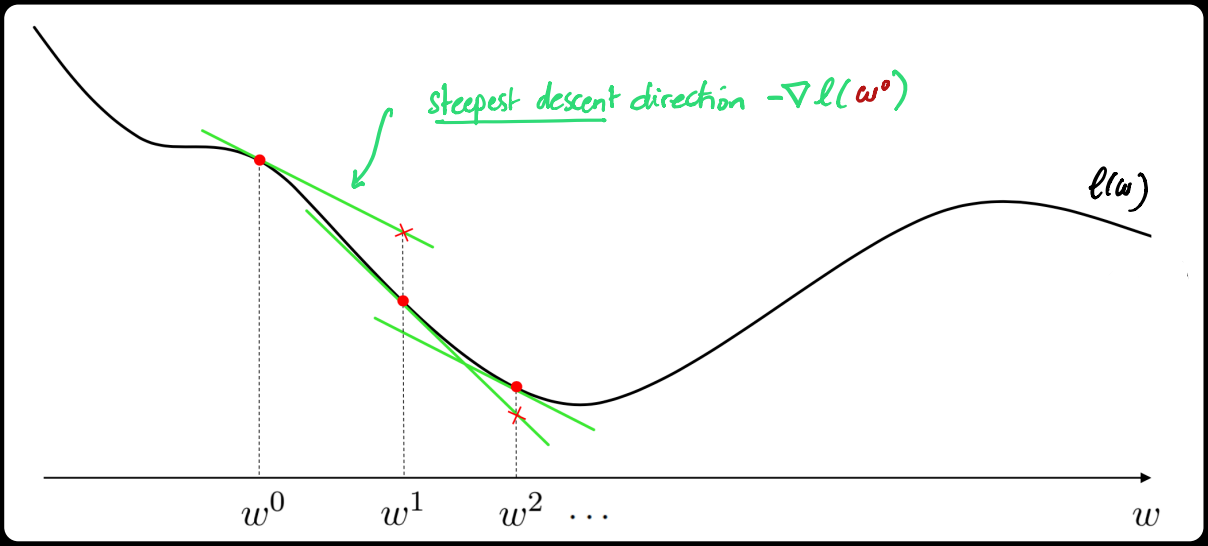
$$\Rightarrow \boxed{w^k = w^{k-1} - \alpha_k \nabla \ell(w^{k-1})}$$

w^*

k^{th} gradient descent

generally $\Rightarrow \boxed{\alpha_k = \alpha}$

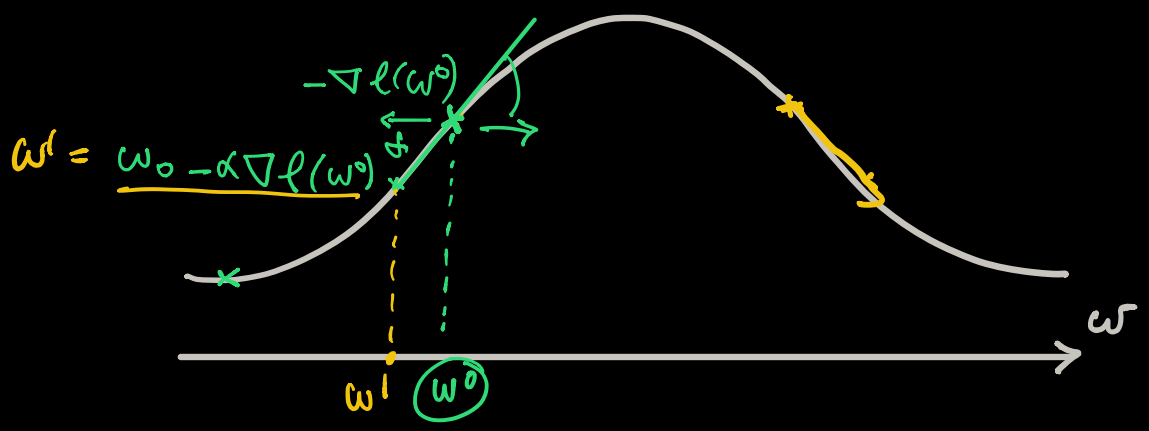
Gradient descent



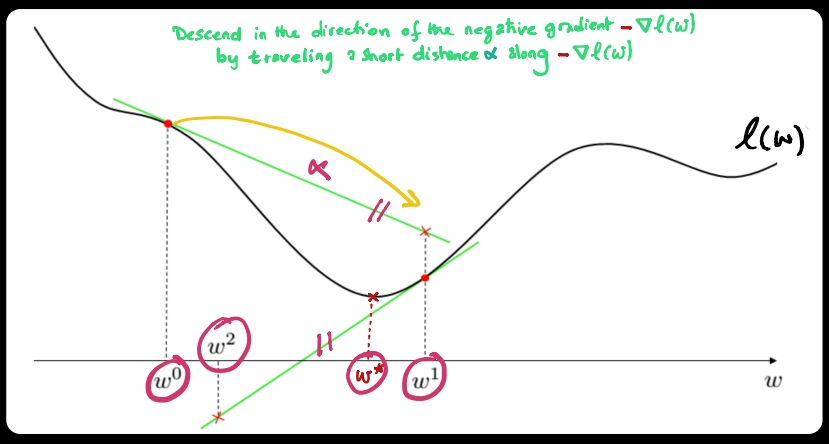
① Travel in the downward direction of a linear approximation
② hop back onto the function
③ repeat in order to find a stationary point of ℓ.

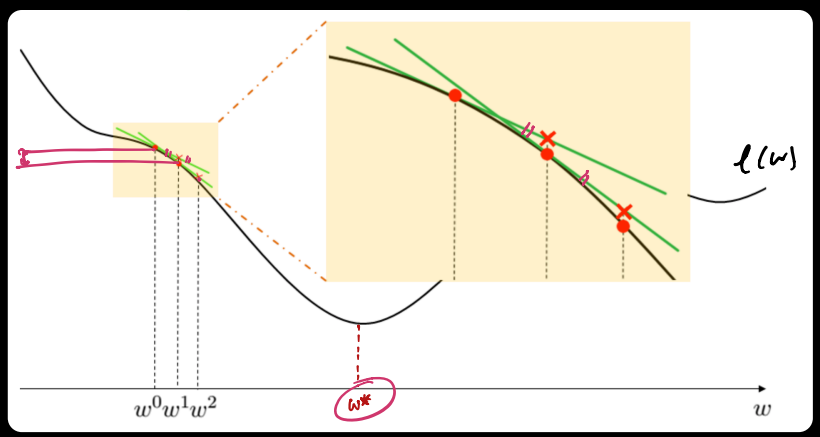
$$h(w) = \ell(w^0) + \nabla \ell(w^0)^T (w - w^0)$$

" How to find the direction with the steepest descent ? "



How to choose α ?





Algorithm 2.1 Gradient descent (with fixed step length)

Input: differentiable function ℓ, fixed step length α, and initial point $\mathbf{w}^0$

$k = 1$ $\quad \nabla\ell(w^k) \leq \varepsilon$

Repeat until stopping condition is met:

$$\mathbf{w}^k = \mathbf{w}^{k-1} - \alpha \nabla \ell(\mathbf{w}^{k-1})$$

$$k \leftarrow k + 1$$

[Matlab / Python demo]