COGS 118A, Winter 2020

Supervised Machine Learning Algorithms

Lecture 4: Estimation and regression

Midterm 1

Midterm I, 01/30/2020 (Thursday)

Time: 12:30-13:50PM

Location: Ledden Auditorium

You can bring one page "cheat sheet". No use of computers/smart-phones during the exam.

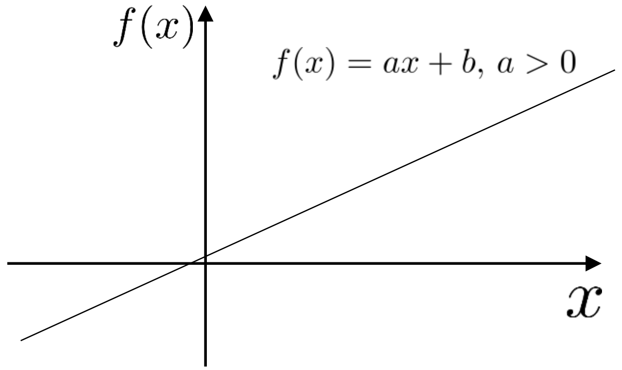
Bring your pen.

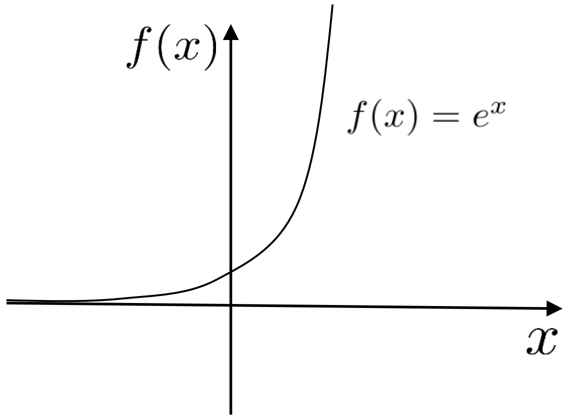
Bring your calculator.

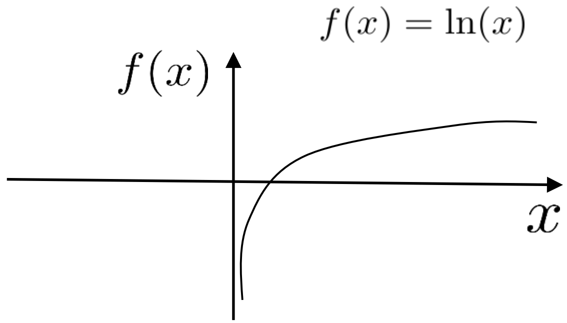
A study guide and practice questions will be provided.

No homework assignment for the next week.

Monotonic functions





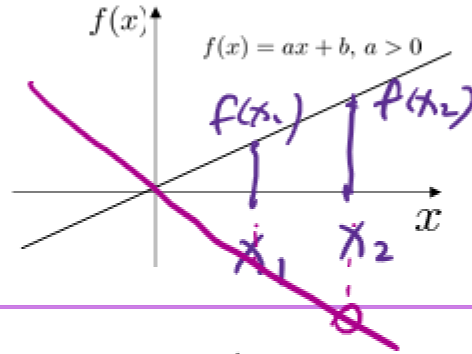


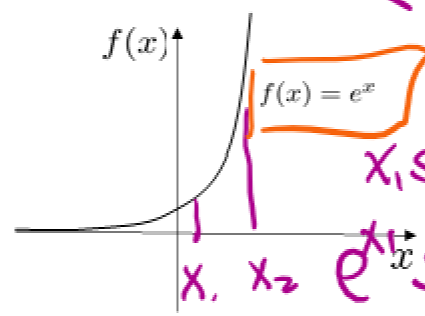
$x_1 \geq x_2$

$f(x_1) \geq f(x_2)$

$-f(x_1) \leq -f(x_2)$

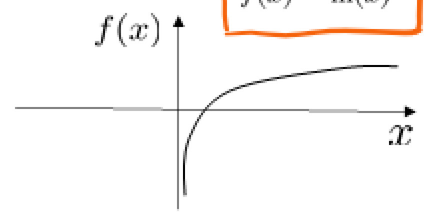
Monotonic functions





$x_1 \leq x_2$

$e^{x_1} \leq e^{x_2}$



$f(x)$ ↑ increasing

$x_1 \geq x_2$

$f(x_1) \geq f(x_2)$

$f(x)$ ↓ decreasing

$x_1 \geq x_2$

$f(x_1) \leq f(x_2)$

For a monotonically increasing function:

$f(x)$

Is $-f(x)$ monotonically decreasing?

A. Yes

B. No

C. It depends

For a monotonically increasing function:

$f(x)$

Is $-f(x)$ monotonically decreasing?

☆ A. Yes

B. No

C. It depends

For a monotonically increasing function:

$$f(x) \in R^+$$

Is $\ln f(x)$ monotonically increasing?

A. Yes

B. No

C. It depends

For a monotonically increasing function:

$$f(x) \in R^+$$

Is $\ln f(x)$ monotonically increasing?

☆ A. Yes

B. No

C. It depends

For two monotonically increasing functions:

$f(x)$ and $g(x)$

Is $f(x) + g(x)$ monotonically increasing?

A. Yes

B. No

C. It depends

For two monotonically increasing functions:

$f(x)$ and $g(x)$

Is $f(x) + g(x)$ monotonically increasing?

☆ A. Yes

B. No

C. It depends

For two monotonically increasing functions:

$f(x)$ and $g(x)$

Is $f(x) - g(x)$ monotonically increasing?

A. Yes

B. No

C. It depends

For two monotonically increasing functions:

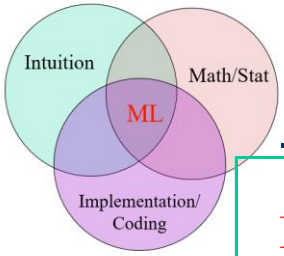
$f(x)$ and $g(x)$

Is $f(x) - g(x)$ monotonically increasing?

A. Yes

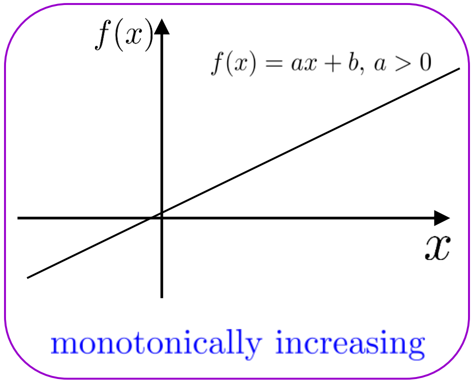
B. No

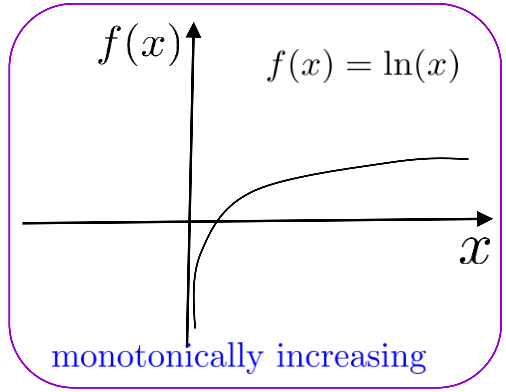
☆ C. It depends

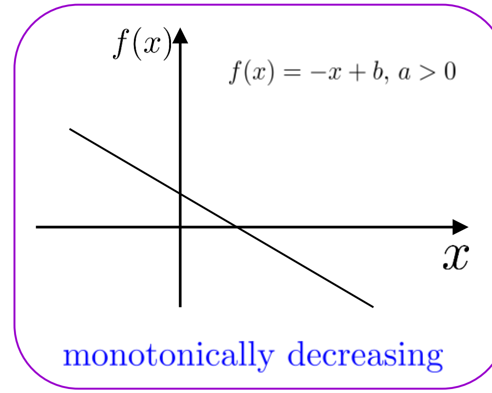


Recap: Monotonicity

Intuition: In machine learning, we use the monotonicity of functions to help significantly reduce the difficulty/complexity of an estimation/learning problem.







Math:

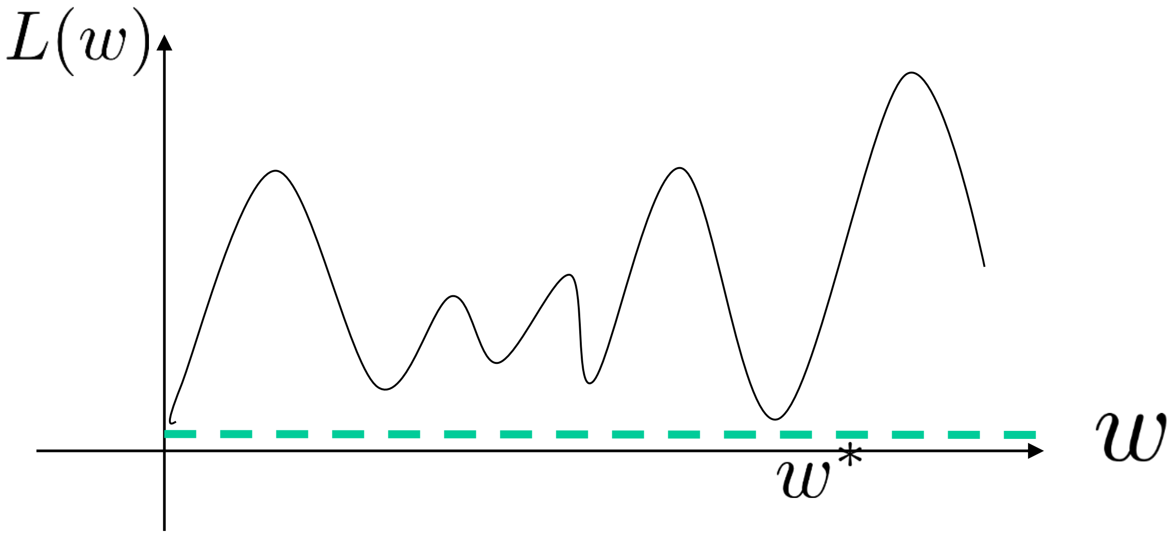
$$
\begin{aligned}
w^* &= \arg\max_w \prod_{i=1}^{n} p(y_i|x_i; w) \\
&= \arg\max_w \ln[\prod_{i=1}^{n} p(y_i|x_i; w)] \\
&= \arg\min_w -\sum_{i=1}^{n} \ln[p(y_i|x_i; w)]
\end{aligned}
$$

Optimization: argmin

$$w^* = \arg\min_w L(w)$$

The operator $\arg\min$ defines the optimal value (in the argument of function $L()$) w^* that minimizes $L(w)$

$\arg\min L(w)$ doesn't return the value of $L(w)$

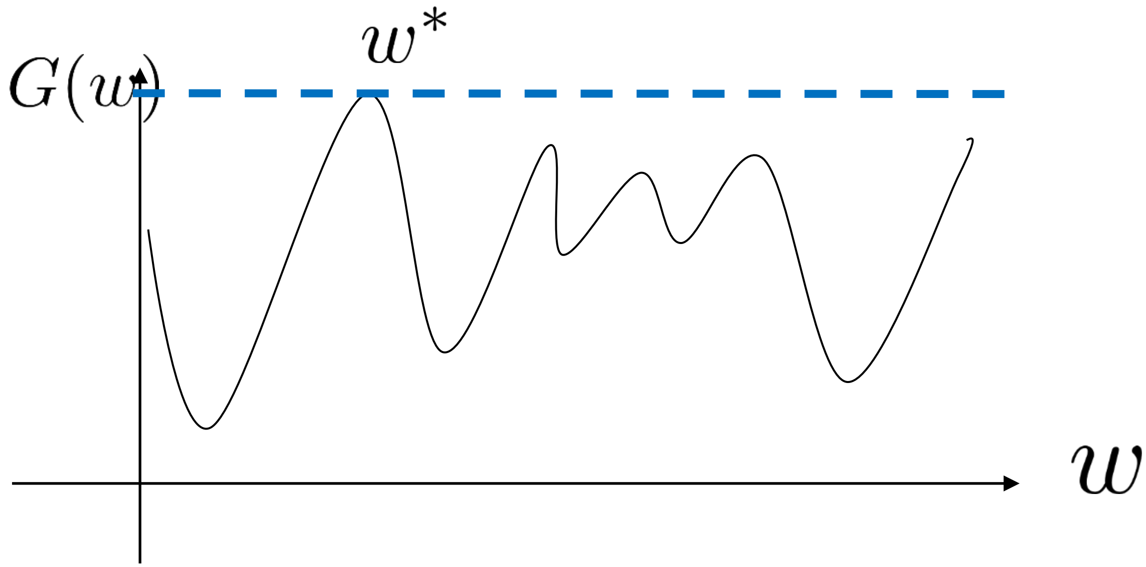


Optimization: argmax

$$w^* = \arg\max_w G(w)$$

The operator $\arg\max$ defines the optimal value (in the argument of function $G()$) w^* that maximizes $G(w)$

$\arg\max G(w)$ doesn't return the value of $G(w)$



argmin and argmax

argmin and argmax are two commonly used terms in machine learning and optimization.

The reason being we want to perform learning: a process in which the "best" model parameters are to be learned.

For example, who is the richest person in the world in 2019 (the answer concerns with the person not the amount of money this person has).

$$\text{person}^{richest} = \arg\max_{\text{person}} NetWorth(\text{person})$$

The answer is Jeff Bezos, not the net worth ($113 billion).

$$\text{person}^{richest} = \text{Jeff Bezos}$$

argmin and argmax

argmin and argmax are two commonly used terms in machine learning and optimization.

The reason being we want to perform learning: a process in which the "best" model parameters are to be learned.

For example, who is the richest person in the world in 2019 (the answer concerns with the person not the amount of money this person has).

$f(x) \nearrow$

$$\text{person}^{richest} = \arg\max_{\text{person}} NetWorth(\text{person})$$

$$\equiv \arg\max_{\text{persn}} \boxed{f(NetWorth(person))}$$

The answer is Jeff Bezos, not the net worth ($113 billion).

$$\text{person}^{richest} = \text{Jeff Bezos}$$

argmin and argmax

In addition:

$$w^* = \arg\min_w L(w)$$
$$= \arg\max_w -L(w)$$

$$\text{person}^{richest} = \arg\max_{\text{person}} NetWorth(\text{person})$$
$$= \arg\min_{\text{person}} -NetWorth(\text{person})$$

$$\text{person}^{richest} = \text{person}^{-poorest}$$

Optimization: argmin

$$w^* = \arg\min_w L(w)$$

If a function $g(v)$ is monotonic, e.g. $\forall v_1 > v_2$ it is always true that $g(v_1) > g(v_2)$, then:

$$w^* = \arg\min_w L(w) = \arg\min_w g(L(w))$$

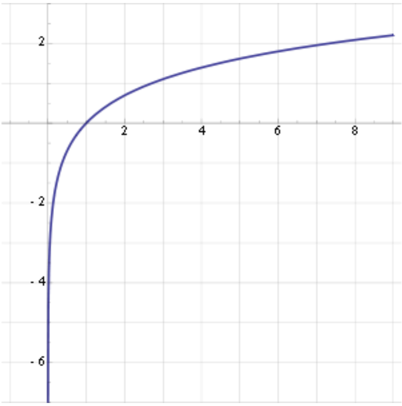
For example,

if $g(v) = 2 \times v + 10$

$$w^* = \arg\min_w L(w) = \arg\min_w 2 \times L(w) + 10$$

argmin and argmax

The function $\ln(v)$ is monotonically increasing, e.g. $\forall v_1 > v_2$ it is always true that $\ln(v_1) > \ln(v_2)$, then:



$$w^* = \arg\max_w G(w)$$

$$= \arg\max_w \ln(G(w))$$

$$= \arg\min_w -\ln(G(w))$$

$S_{tr} = \{ (x_1, y_1), \dots (x_n, y_n) \}$

$p(y_1|x_1), p(y_2|x_2) \dots p(y_n|x_n)$

Maximize $\prod_{i=1}^{n} p(y_i|x_i;w) = f(x)$

$\equiv$ Maximize $\ln\left(\prod_{i=1}^{n} p(y_i|x_i;w) \right)$

Maximize $f(x;w)$ when $f(x;w)$ is monotonical

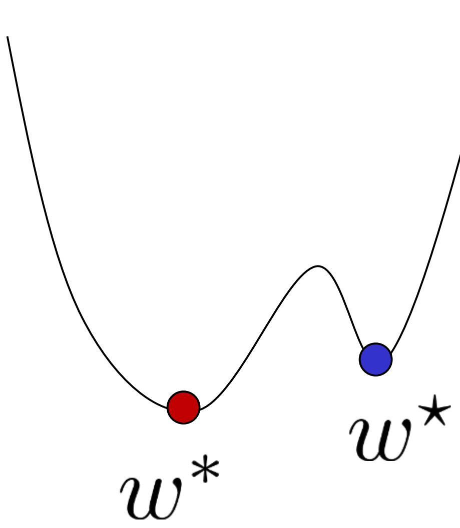
$\equiv$ maximize $\ln(f(x;w))$

Maximize $\sum_{i=1}^{n} \ln p(y_i|x_i;w)$

$$\boxed{\frac{\partial f_1(x) \cdot f_2(x)}{\partial x} = f_1(x) \cdot \frac{\partial f_2(x)}{\partial x} + f_2(x) \frac{\partial f_1(x)}{\partial x}}$$

$$\frac{\partial \sum_{i=1}^{n} \ln p(y_i|x_i;w)}{\partial w} = \sum_{i=1}^{n} \frac{\partial \ln p(y_i|x_i;w)}{\partial w}$$

Optimization



Things we ofen need to be able to do to solve optimization problem:

1. $\forall w$, check if $w \in \Omega$?
2. For $\forall w$, computing $L(w)$, $\nabla L(w)$, $\nabla^2 L(w)$.

Definition:

1. w^* is a globally optimal solution for $\theta^* \in \Omega$ and $L(w^*) \leq L(w) \forall w \in \Omega$
2. $w^\star$ is a locally optimal solution if there is a neighborhood $\mathcal{N}$ around w such that $w^\star \in \Omega$, $L(w^\star) \leq L(w)$, $\forall w \in \mathcal{N} \cap \Omega$.

Reasons to study optimization/estimation in machine learning

$$e_{testing} = e_{training} + generalization(f)$$

Ideally: minimize $e_{testing}$

For the moment: minimize $e_{training}$

Minimize $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i \neq f(\mathbf{x}_i; W))$

In general: $W^* = \arg\min_W \mathcal{L}(W)$, where $\mathcal{L}(W) = e_{training}$ defines a loss/objective function in machine learning.

Reasons to study optimization/estimation in machine learning

In general: $W^* = \arg\min_W \mathcal{L}(W)$, where $\mathcal{L}(W) = e_{training}$ defines a loss/objective function in machine learning.

Our goal in the learning process is to find the "optimal" W that minimizes the error (most of the time).

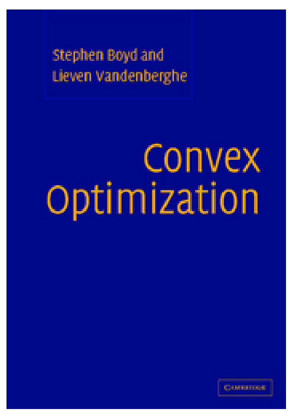
Sometimes, we have other constraints to satisfy, the model complexity (e.g. we cannot afford a full deep model on local smart phone devices).

Estimation and optimization

$$w^* = \arg min_w \quad L(w)$$

Learning and estimation with convex functions:

http://stanford.edu/~boyd/



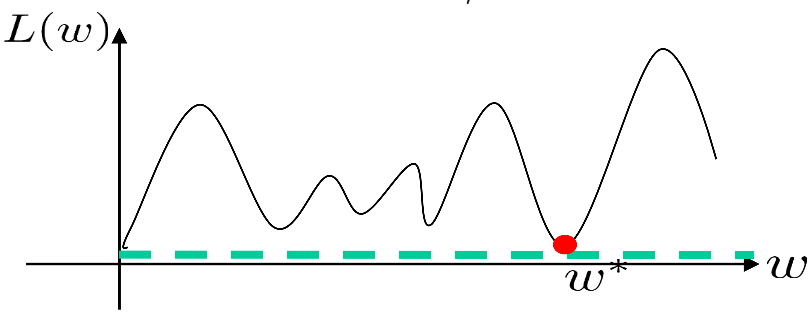
Convex Optimization
Stephen Boyd and Lieven Vandenberghe

Cambridge University Press

A new MOOC on convex optimization, CVX101, will run from 1/21/14 to 3/14/14.

Recap: Estimation

Intuition: Typically, optimization in machine learning refers the process in which an objective function is minimized/maximized. A major task in machine learning is to perform training to attain the optimal parameter that minimizes/maximizes the corresponding objective function. Note that the optimal model parameter is not the minimal/maximal value of the function itself.



Math:

$$w^* = \arg\min_w L(w)$$

Convexity

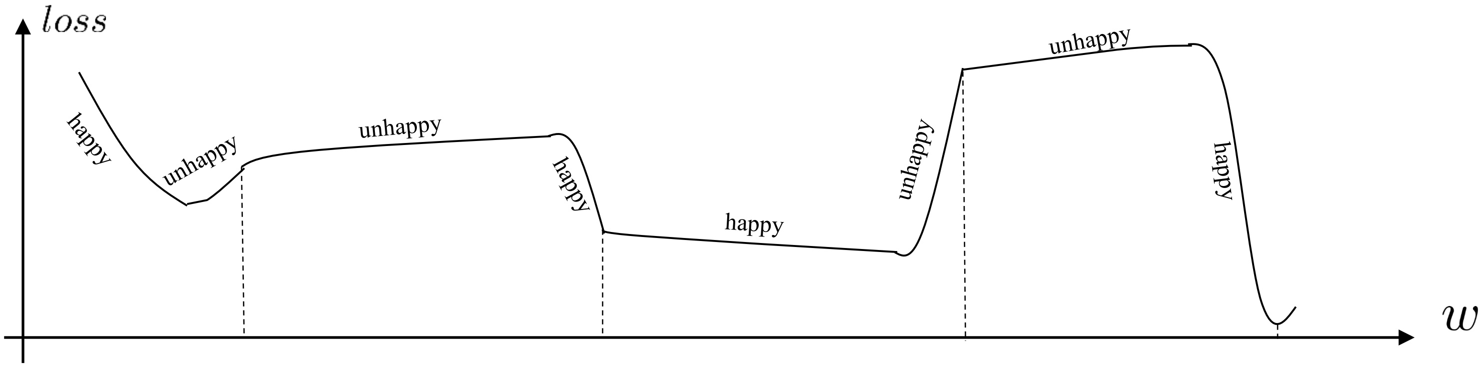
Why do we study the convexity of a function?

It gives us a good understanding about the shape of the function:

1. Does the optimal solution exist?
2. How to find the optimal solution.

A perspective of estimation

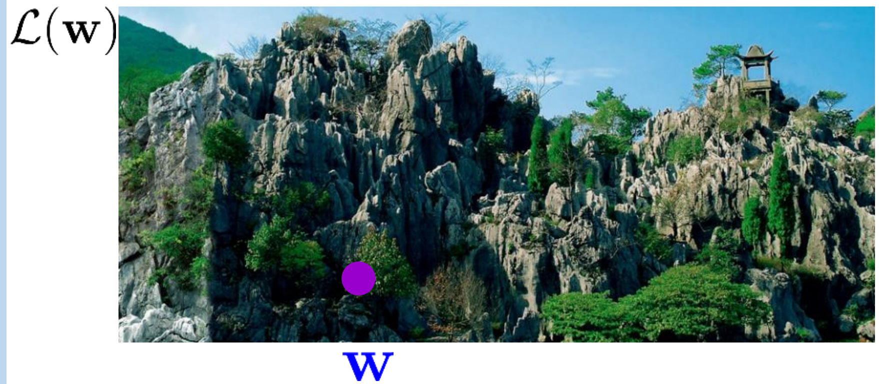
Mr. Sai
http://www.baike.com/wiki/



An analogy:
we want to find the lowest point in the figure.

http://menpiao.daiwoqu.com/

Essenes of estimation/optimization



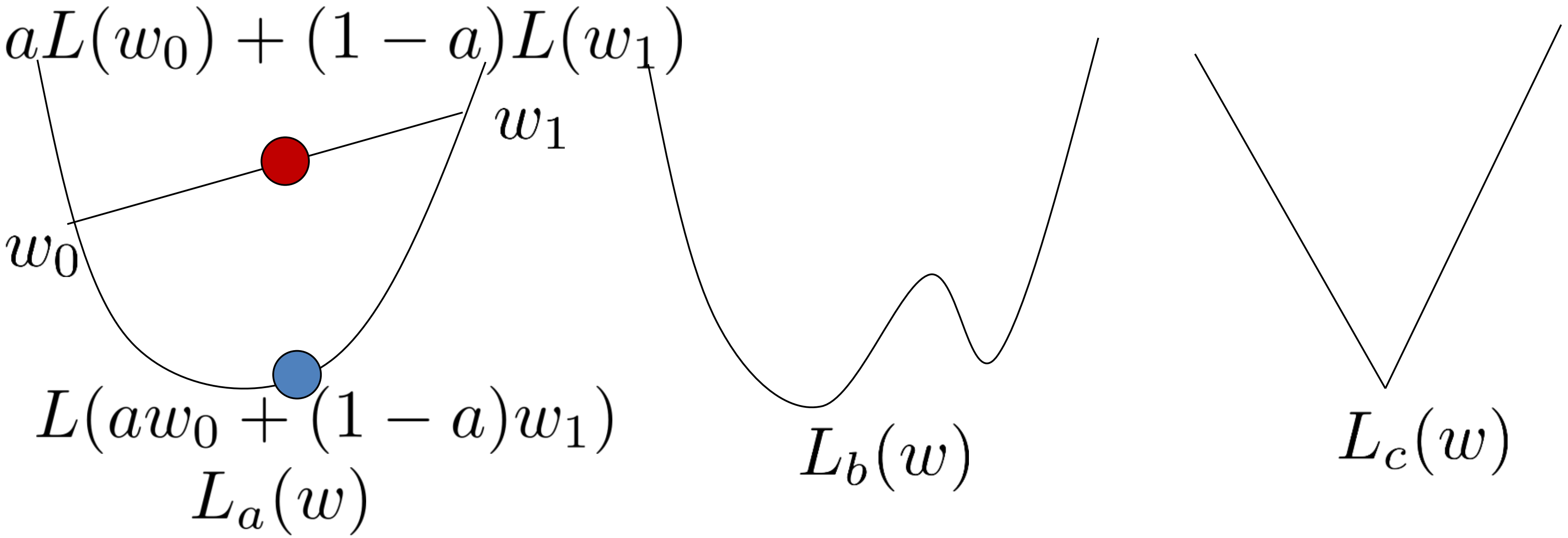
1. Given a **w** (model parameter), we can always evaluate the loss L(**w**).

2. However, we don't know a prior about the entire shape of the L(**w**) (no access to the entire map).

3. The estimation process finds, hopefully, the "optimal" **w** that produces the smallest L(**w**) (lowest point on the map).

Convex functions

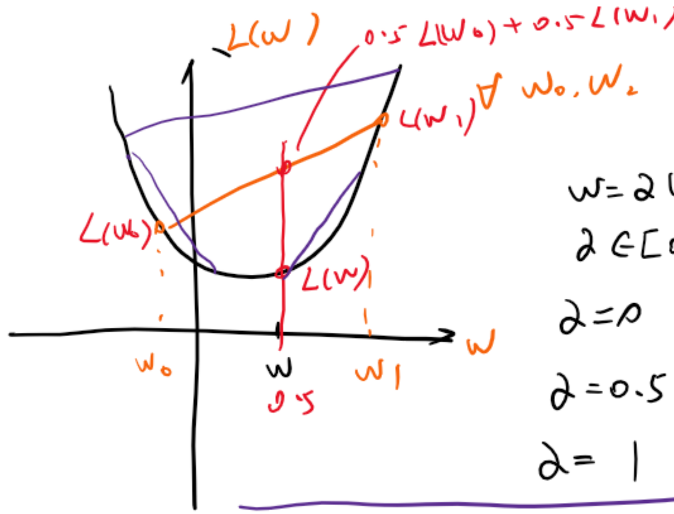
$$w^* = \arg min_w \quad L(w)$$



Definition:

$$\forall w_0, w_1, a \in [0, 1]$$

$$aL(w_0) + (1-a)L(w_1) \geq L(aw_0 + (1-a)w_1)$$



$w = \lambda w_0 + (1-\lambda) w_1$

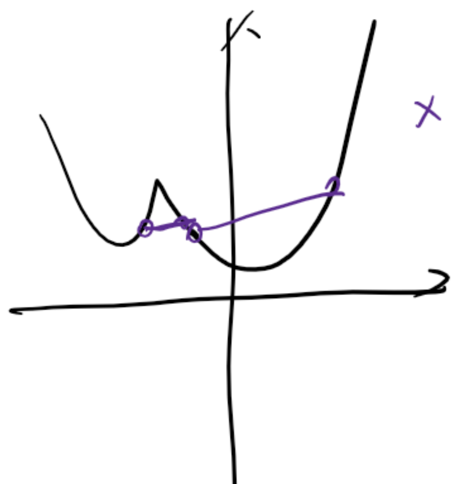
$\lambda \in [0,1]$

$\lambda = 0 \quad w = 0 w_0 + 1 w_1 = w_1$

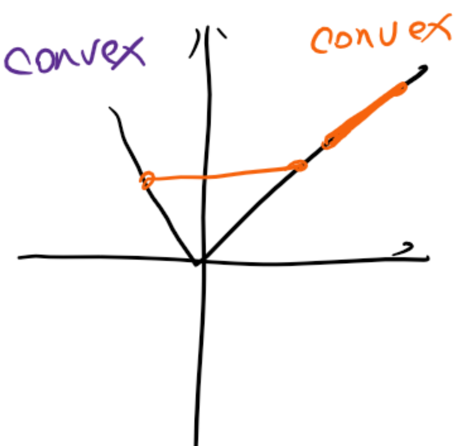
$\lambda = 0.5 \quad w = 0.5 w_0 + 0.5 w_1$

$\lambda = 1 \quad w = w_0$

$$\boxed{L(w) \le \lambda L(w_0) + (1-\lambda) L(w_1) \qquad 0.5 L(w_0) + 0.5 L(w_1)}$$

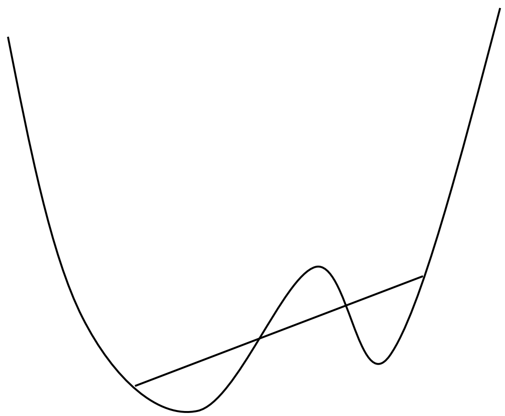


× not convex

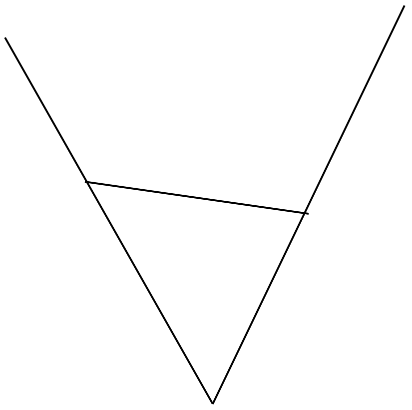


Convex functions

$$w^* = \arg min_w \quad L(w)$$



$L_b(w)$



$L_c(w)$

$$\forall w_0, w_1, a \in [0, 1]$$

$$aL(w_0) + (1 - a)L(w_1) \geq L(aw_0 + (1 - a)w_1)$$

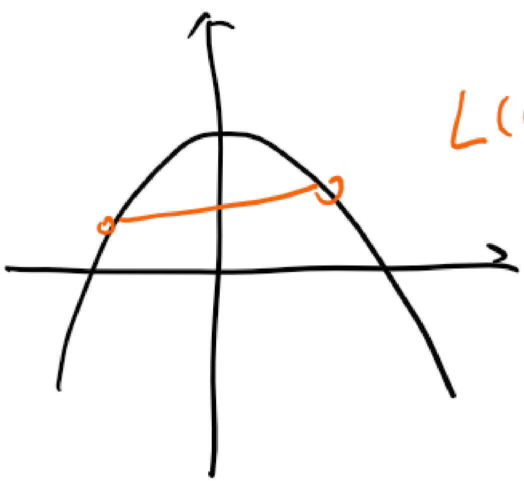
convex

$\forall\, w_0, w_1$

$$L(\lambda w_0 + (1-\lambda) w_1) \boxed{\leq} \lambda L(w_0) + (1-\lambda) L(w_1)$$

$$L(\lambda w_0 + (1-\lambda) w_1) \boxed{<} \lambda L(w_0) + (1-\lambda) L(w_1)$$

strictly convex

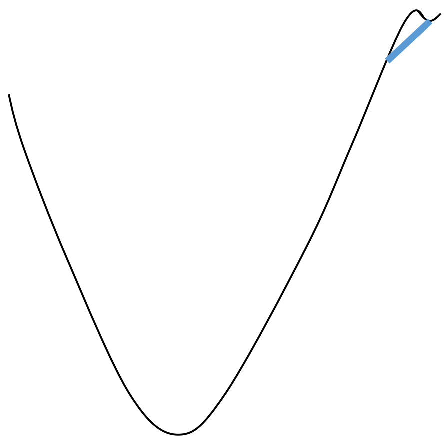


$$L(\lambda w_0 + (1-\lambda) w_1) \geq \lambda L(w_0) + (1-\lambda) L(w_1)$$

concave.

Convexity

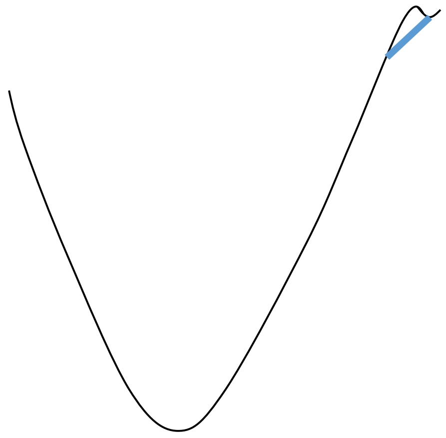
Is this a convex function?



A. Yes

B. No

C. It depends



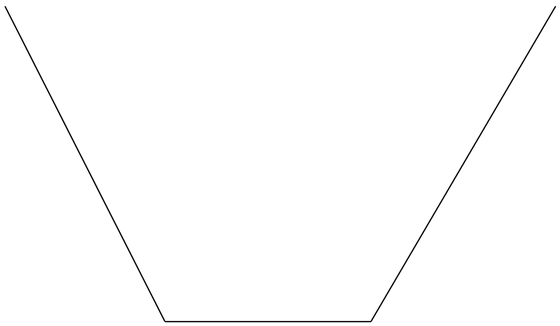
Convexity

Is this a convex function?

A. Yes

☆ B. No

C. It depends



Convexity

Is this a convex function?

A. Yes

B. No

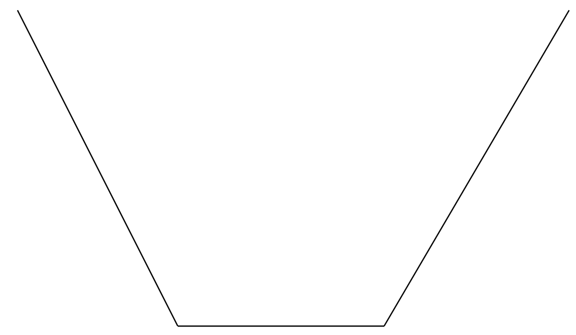
C. It depends

But not strictly convex

$$aL(w_0) + (1-a)L(w_1) > g(aw_0 + (1-a)w_1)$$

Convexity

Is this a convex function?



☆ A. Yes

B. No

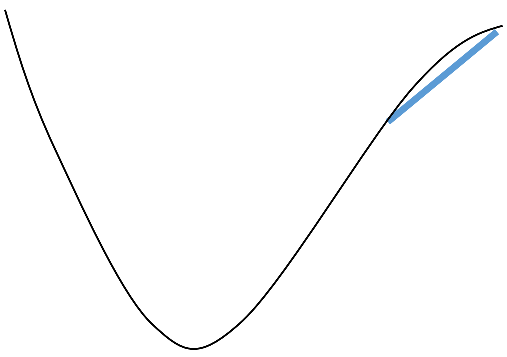
C. It depends

But not strictly convex

$$aL(w_0) + (1-a)L(w_1) > L(aw_0 + (1-a)w_1)$$

Convexity

Is this a convex function?



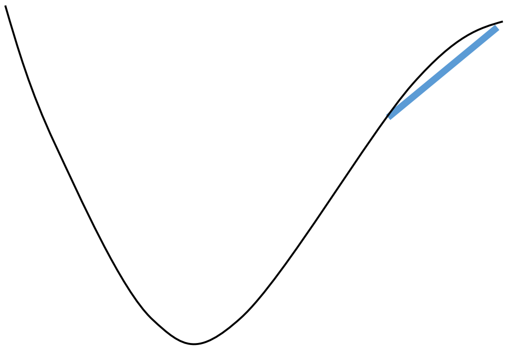
A. Yes

B. No

C. It depends

Convexity

Is this a convex function?



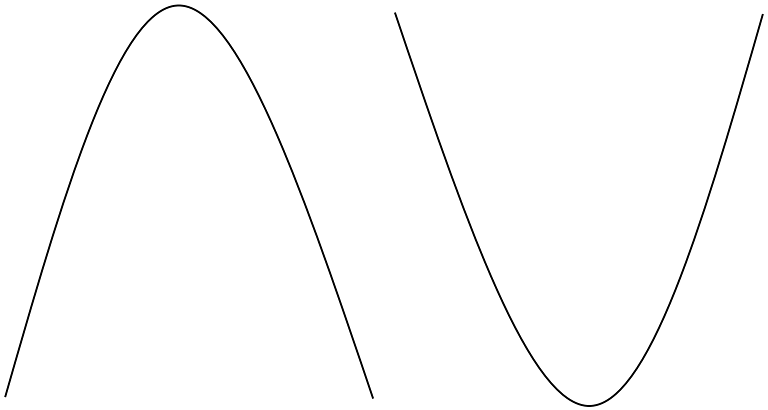
A. Yes

☆ B. No

C. It depends

Convexity

Is this a convex function?



A. Yes

B. No

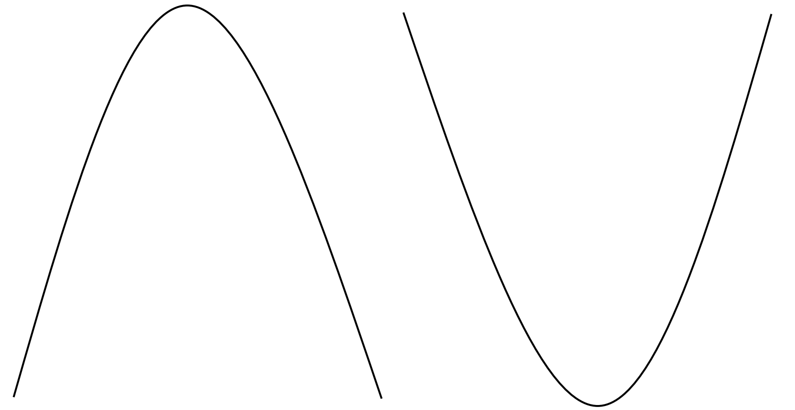
C. It depends

It is concave!

But for a concave function $L(w)$, $-L(w)$ is convex, and vice versa.

Convexity

Is this a convex function?



A. Yes

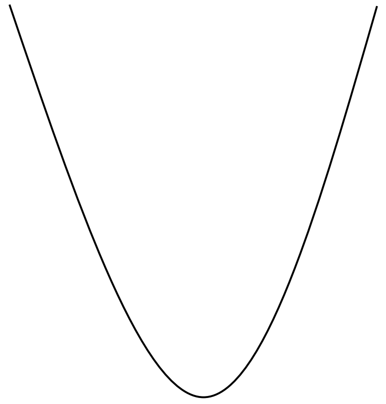
☆ B. No

C. It depends

It is concave!

But for a concave function $L(w)$, $-L(w)$ is convex, and vice versa.

Why do we study convex function

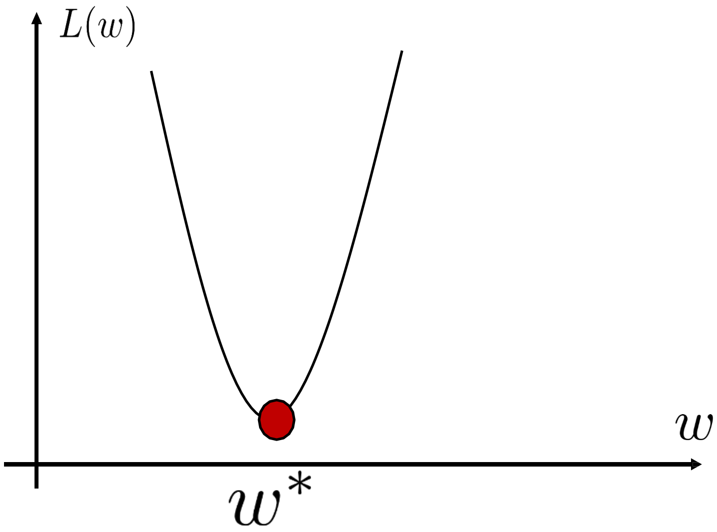


1. It has the globally optimal solution (to learn the best model).

2. Might have a closed form solution, if it is everywhere differentiable and has analytic form (learning accomplished in one-shot).

3. Gradient descent/ascent can be directly applied (iterative steps).

Convex function: differentiable



For a convex and differentiable function $L(w)$, it's global optimal is achieved at w^*,

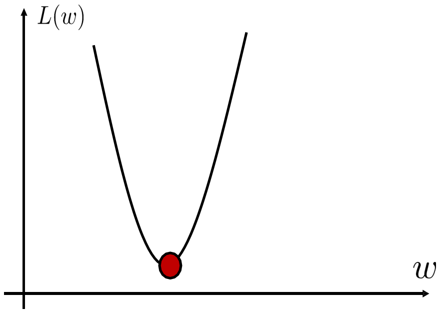
$$\text{when } \frac{\partial L(w)}{\partial w}\Big|_{w^*} = 0$$

To find w^*, we simply solve for the equation:

$$\frac{\partial L(w)}{\partial w} = 0$$

Why do we study/care about derivatives?

1. Find the optimal solution using an analytical (closed) form for a convex function that is everywhere differentiable.



$$\frac{\partial L(w)}{\partial w}\Big|_{w^*} = 0$$

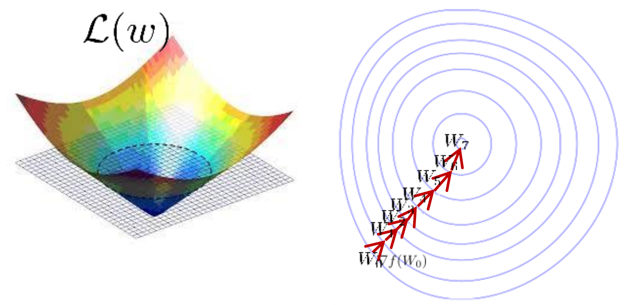
An analytical (closed) form refers to a direct solution as:

$w^* = q(X, Y)$ where X and Y consists of your training data with the corresponding ground-truth labels.

That is, you obtain your model by one-shot (no iterations needed).

Why do we study/care about derivatives?

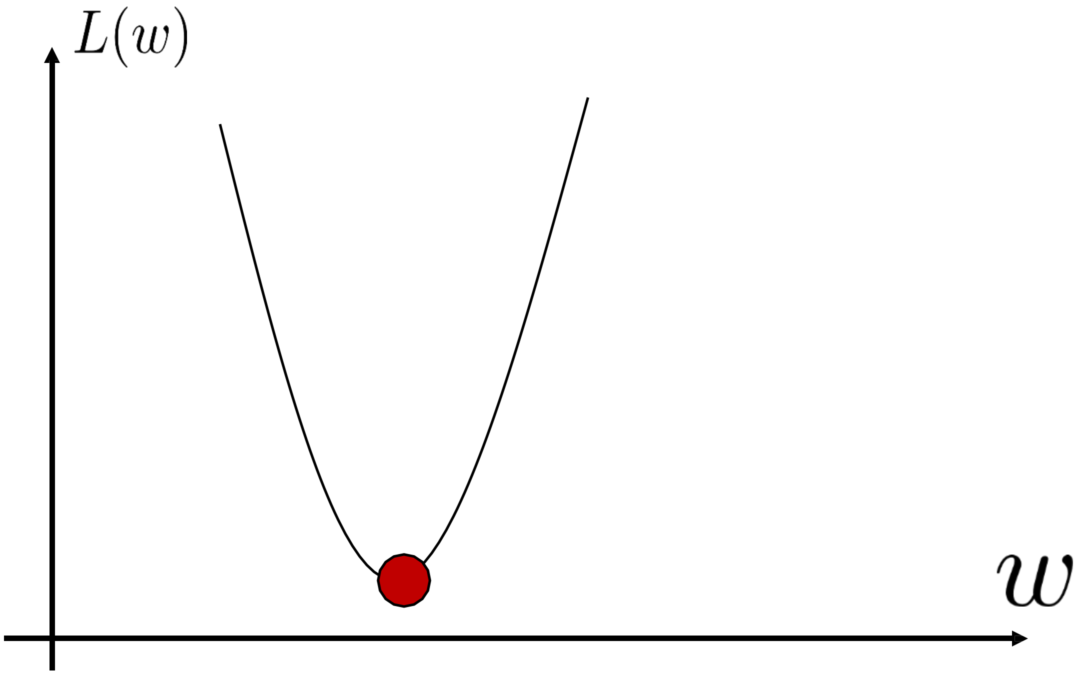
2. When your loss/objective function is NOT convex or/and NOT everywhere differentiable.



We still compute $\frac{\partial L(w)}{\partial w}$ to find the w^*
by an iteratively learning process (gradient decent).

An analytical (closed) form here no longer exists.

Convex function: differentiable



$$w^* = \arg min_\theta L(w)$$

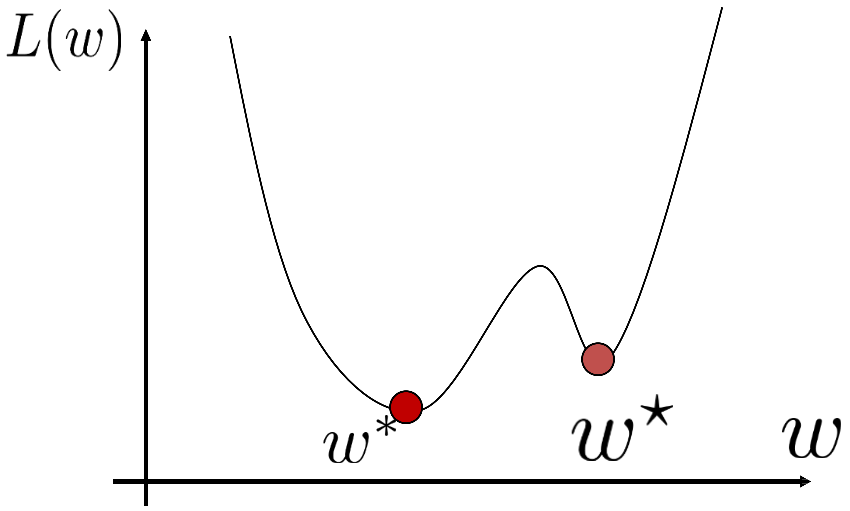
1. (Convex) Function
2. Set Derivative to 0
3. Solve for w

$$L(w) = (w-3)^2 + 4$$

$$\frac{dL(w)}{dw} = 2 \times (w-3) \qquad \frac{dL(w)}{dw} = 0$$

$$2 \times (w-3) = 0 \rightarrow w = 3$$

Convex function: differentiable



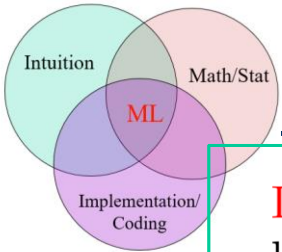
For a non-convex but differentiable function $L(w)$, it's optimal (either global or local) is achieved when,

$$\frac{\partial L(w)}{\partial w}\Big|_{w^*} = 0$$

$$\frac{\partial L(w)}{\partial w}\Big|_{w^\star} = 0$$

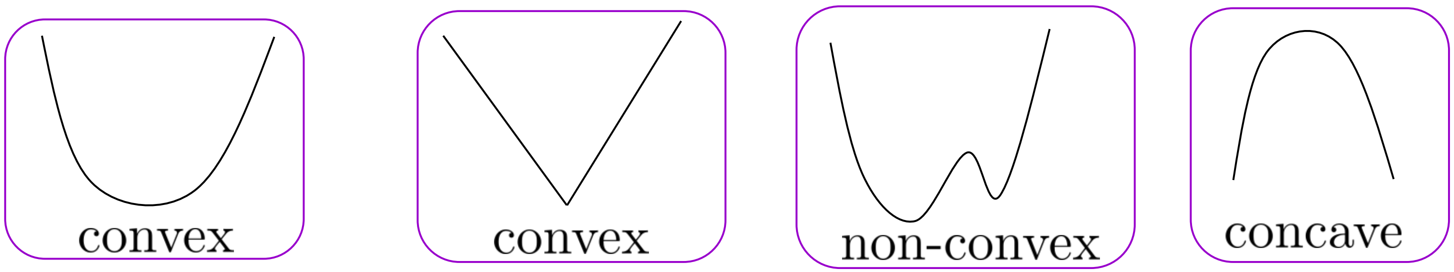
To find $w^\star$, we simply solve for the equation:

$$\frac{\partial L(w)}{\partial w} = 0$$



Recap: Convexity

Intuition: Understanding the convexity of the estimation functions allows us to better design the learning algorithms and allows us to judge the quality (global vs. local optimal) of the learned models.



Math:

$$\forall w_0, w_1, a \in [0,1]$$

$$aL(w_0) + (1-a)L(w_1) \geq L(aw_0 + (1-a)w_1)$$

or

Alternatively (for differentiable function):

$$L(w_1) \geq L(w_0) + < \nabla L(w_0), w_1 - w_0 >$$

Problem Definition and High-level Understanding

Regression: predicting blood presure

$$S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\} \qquad \mathbf{x}_i = (x_{i1}, ..., x_{im}) \qquad y \in \mathcal{R}$$

blood presure	age	male or female	weight (lb)	height (cm)
y_1=131	$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
y_2=150	$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
y_3=105	$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

$$Y = \begin{pmatrix} 131 \\ 150 \\ 105 \end{pmatrix} \qquad X = \begin{pmatrix} 22 & 1 & 0 & 160 & 180 \\ 51 & 1 & 0 & 190 & 165 \\ 43 & 0 & 1 & 120 & 165 \end{pmatrix}$$

$$W^* = \arg\min_\theta L(W)$$

$$Loss : L(W) = ||Y - XW||$$

Difference between training values Y and predicted values XW.

Problem overview

$$e_{testing} = e_{training} + generalization(f)$$

We will focus on training error for the moment:

$$S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$$

$$e_{training} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i \neq f(\mathbf{x}_i))$$

$$accuracy_{training} = 1 - e_{training}$$

$generalization(f)$: will be discussed later.

General approaches for optimization

- Exhaustive search
- Gradient descent
- Coordinate descent
- Newton’s method
- Line search
- Stochastic computing
- Stochastic sampling (Markov chain Monte Carlo)
- ….

Linear Regression and Least Square Estimation

An example

Birthweight based on the mother's Estriol

<u>Estriol</u> (mg/24h)	<u>Birthweight</u> (g/1000)
1	1
3	1.9
2	1.05
5	4.1
4	2.1

https://www.dailyclipart.net/

Linear Regression

The basic idea of linear regression is to learn a linear function:

$$
\begin{aligned}
f(\mathbf{x}; \mathbf{w}, b) &=< \mathbf{w}, \mathbf{x} > +b \\
&= \mathbf{w} \cdot \mathbf{x} + b \\
&= \mathbf{w}^T \mathbf{x} + b
\end{aligned}
$$

$\mathbf{x} = \mathbb{R}^m$ $\quad$ $\mathbf{w} = \mathbb{R}^m$ $\quad$ $b \in \mathbb{R}$

Further: $W = (\mathbf{w}, b)$ since b can be also viewed as a parameter in W when a constant 1 is appended to every $\mathbf{x}$.

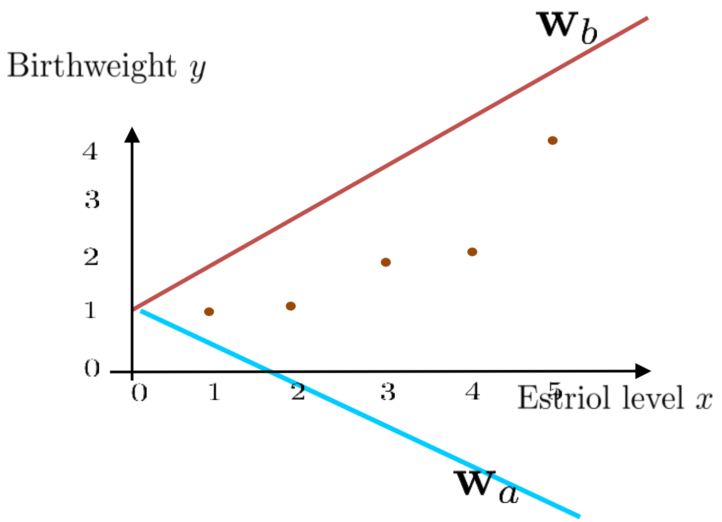
This is a linear function and our job is find the optimal **w** and **b** to best fit the prediction in learning.

Once learned, the linear regression function can be readily computed.

An example

Training data

Estriol (mg/24h)	Birthweight (g/1000)
1	1
3	1.9
2	1.05
5	4.1
4	2.1



$$S_{training} = \{(x_i, y_i), i = 1..n\} = \{(1,1), (3,1.9), (2,1.05), (5,4.1), (4,2.1)\}$$

W_a

If $\mathbf{w}_a = (w_0, w_1) = (1, -0.5)$

$$e_{training}(\mathbf{w}_a) = \frac{1}{5}\sum_{i=1}^{5}(y_i - (1 - 0.5x_i))^2 = 9.62$$

W_b

If $\mathbf{w}_b = (w_0, w_1) = (1, 0.5)$

$$e_{training}(\mathbf{w}_b) = \frac{1}{5}\sum_{i=1}^{5}(y_i - (1 + 0.5x_i))^2 = 0.54$$

$\mathbf{w}_b$ is better than $\mathbf{w}_a$ since $e_{training}(\mathbf{w}_b) < e_{training}(\mathbf{w}_a)$