Essential Calculus and Optimization for ML (1)

CS771: Introduction to Machine Learning

Piyush Rai

Calculus and Optimization for ML

- Regularized Linear Regression (a.k.a. Ridge Regression)

$$\boldsymbol{w}_{ridge} = \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} (y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 + \lambda \boldsymbol{w}^\top \boldsymbol{w} = (\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda I_D)^{-1} (\sum_{n=1}^{N} y_n \boldsymbol{x}_n)$$

Problem more compactly written as $\arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2$

Solution more compactly as $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_D)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$

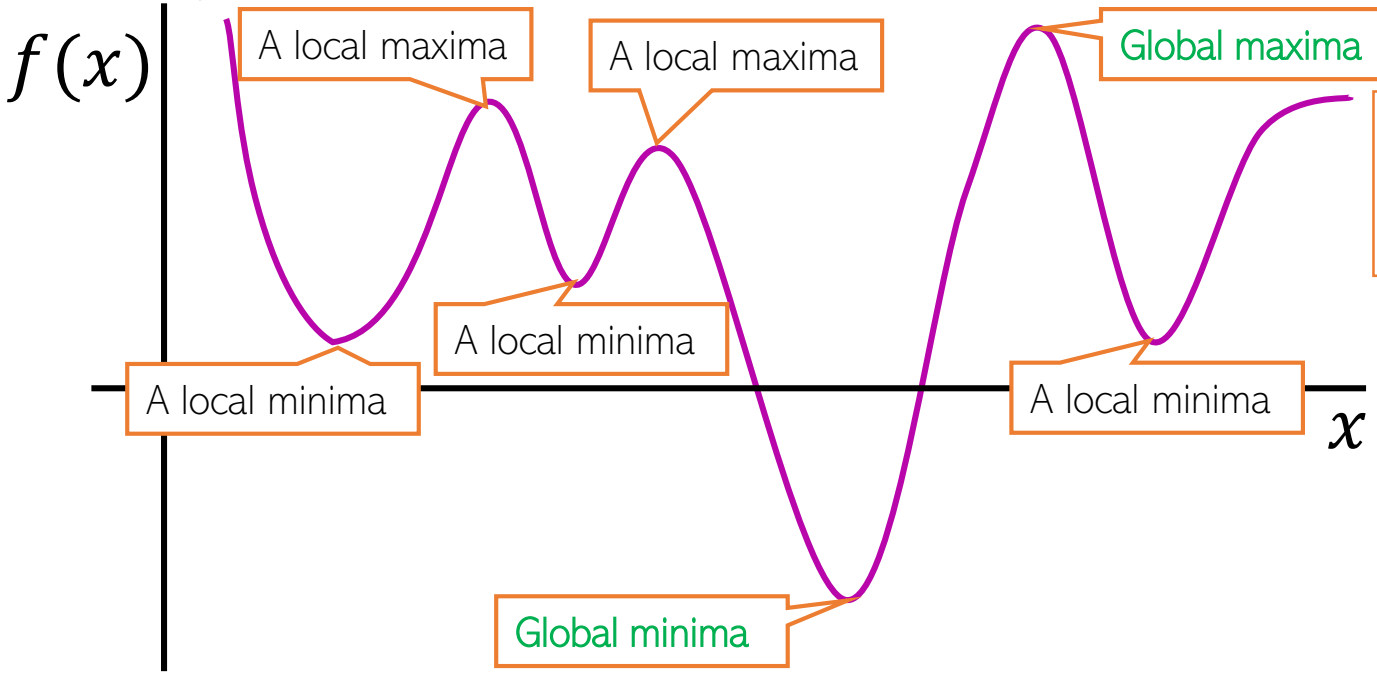
- Getting closed-form soln required simple calculus, but is expensive to compute
  - Especially when D is very large (since we need to invert a $D \times D$ matrix)
- How to solve this and other (possibly more difficult) optimization problems arising in ML efficiently?
- What's the basic calculus and optimization knowledge we need for ML?

Functions and their optima

The objective function of the ML problem we are solving (e.g., squared loss for regression)

Assume unconstrained for now, i.e., just a real-valued number/vector

- Many ML problems require us to optimize a function f of some variable(s) x
- For simplicity, assume f is a scalar-valued function of a scalar x ($f: \mathbb{R} \rightarrow \mathbb{R}$)



Usually interested in global optima but often want to find local optima, too

For deep learning models, often the local optima are what we can find (and they usually suffice) – more later

Will see what these are later

- Any function has one/more optima (maxima, minima), and maybe saddle points
- Finding the optima or saddles requires derivatives/gradients of the function

Derivatives

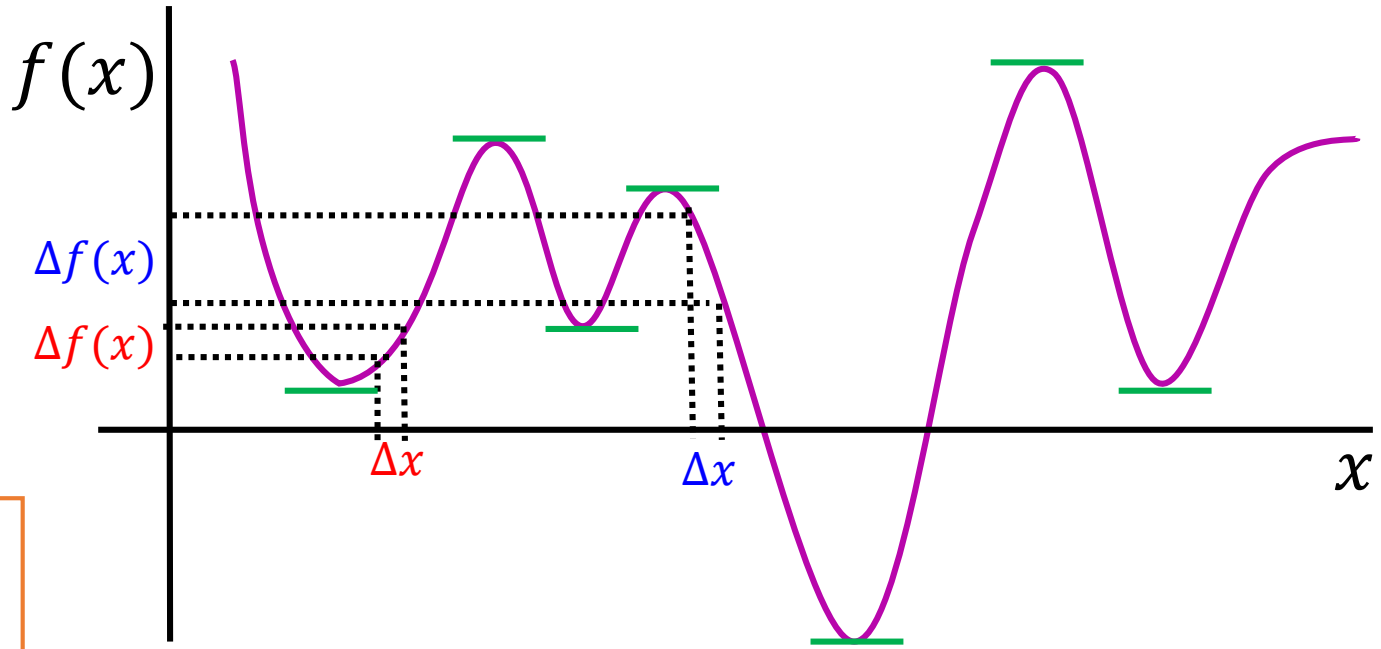
Will sometimes use $f'(x)$ to denote the derivative

- Magnitude of derivative at a point is the rate of change of the func at that point

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{\Delta f(x)}{\Delta x}$$

Sign is also important: Positive derivative means f is increasing at x if we increase the value of x by a very small amount; negative derivative means it is decreasing

Understanding how f changes its value as we change x is helpful to understand optimization (minimization/maximization) algorithms

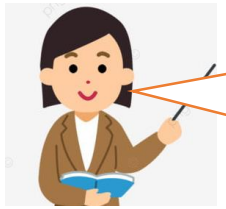


- Derivative becomes zero at stationary points (optima or saddle points)
  - The function becomes "flat" ($\Delta f(x) = 0$ if we change x by a very little at such points)
  - These are the points where the function has its maxima/minima (unless they are saddles)

Rules of Derivatives

Some basic rules of taking derivatives

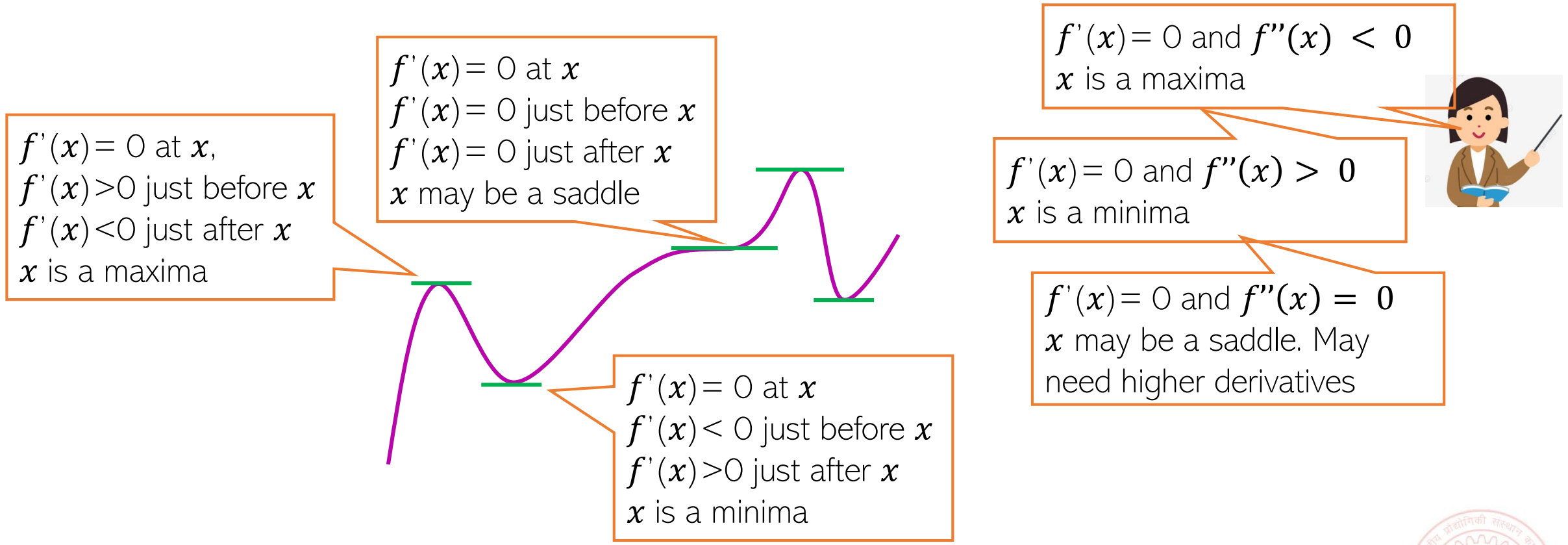
- Sum Rule: $\big(f(x) + g(x)\big)' = f'(x) + g'(x)$
- Scaling Rule: $\big(a \cdot f(x)\big)' = a \cdot f'(x)$ if a is not a function of x
- Product Rule: $\big(f(x) \cdot g(x)\big)' = f'(x) \cdot g(x) + g'(x) \cdot f(x)$
- Quotient Rule: $\big(f(x)/g(x)\big)' = \big(f'(x) \cdot g(x) - g'(x)f(x)\big)/\big(g(x)\big)^2$
- Chain Rule: $\Big(f\big(g(x)\big)\Big)' \stackrel{\text{def}}{=} (f \circ g)'(x) = f'\big(g(x)\big) \cdot g'(x)$



We already used some of these (sum, scaling and chain) when calculating the derivative for the linear regression model

Derivatives

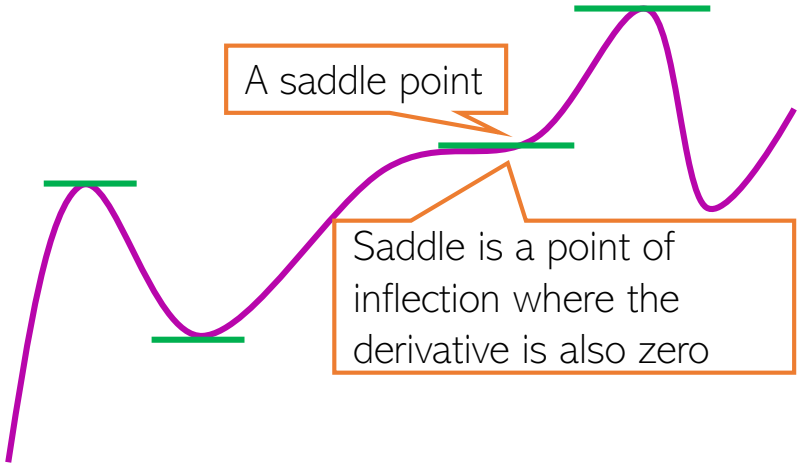
- How the derivative itself changes tells us about the function's optima

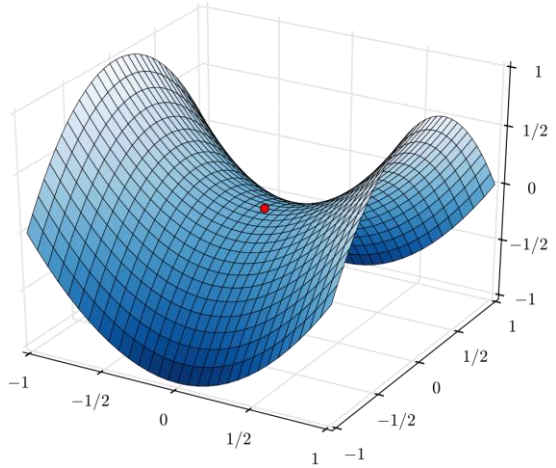


- The second derivative $f''(x)$ can provide this information

Saddle Points

- Points where derivative is zero but are neither minima nor maxima





- Saddle points are very common for loss functions of deep learning models
  - Need to be handled carefully during optimization

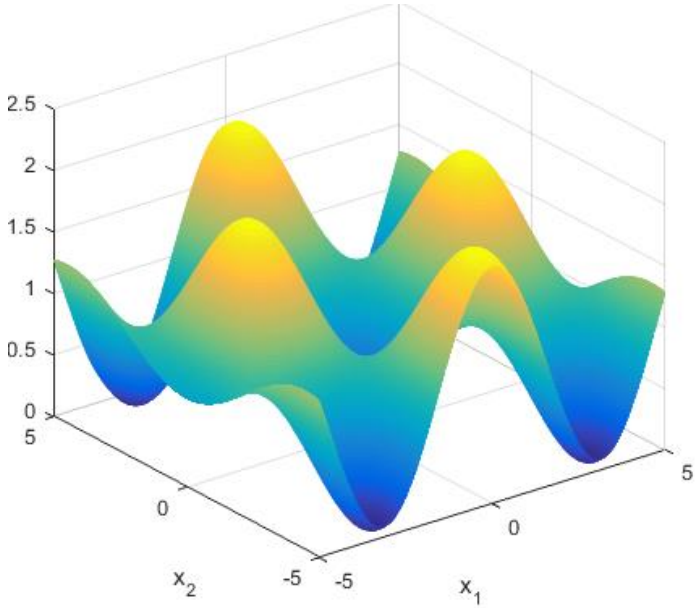
- Second or higher derivative may help identify if a stationary point is a saddle

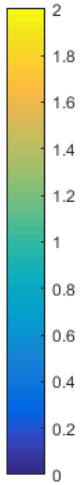
Multivariate Functions

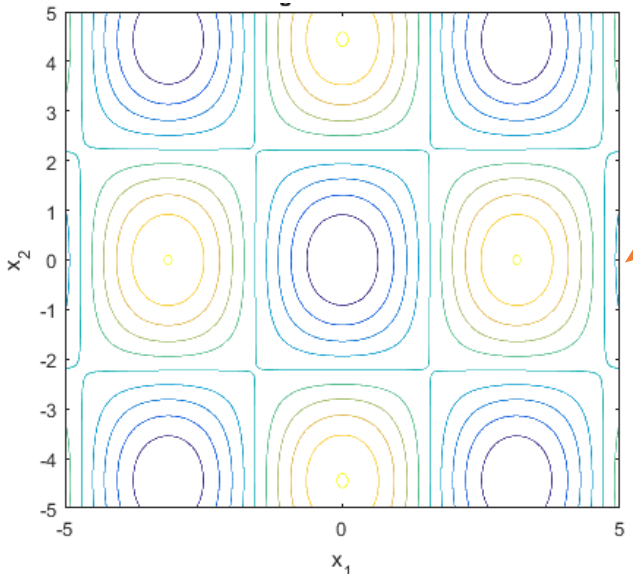
- Most functions that we see in ML are multivariate function
- Example: Loss fn $\boldsymbol{L(w)}$ in lin-reg was a multivar function of $\boldsymbol{D}$-dim vector $\boldsymbol{w}$

$$\boldsymbol{L(w)}: \mathbb{R}^D \rightarrow \mathbb{R}$$

- Here is an illustration of a function of 2 variables (4 maxima and 5 minima)







Two-dim contour plot of the function (i.e., what it looks like from the above)

Plot courtesy: http://benchmarkfcns.xyz/benchmarkfcns/griewankfcn.html

Derivatives of Multivariate Functions

- Can define derivative for a multivariate functions as well via the gradient
- Gradient of a function $f(\boldsymbol{x}): \mathbb{R}^D \rightarrow \mathbb{R}$ is a $D \times 1$ vector of partial derivatives

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_D}\right)$$

Each element in this gradient vector tells us how much f will change if we move a little along the corresponding (akin to one-dim case)

- Optima and saddle points defined similar to one-dim case
  - Required properties that we saw for one-dim case must be satisfied along all the directions
- The second derivative in this case is known as the Hessian

The Hessian

- For a multivar scalar valued function $f(\boldsymbol{x})$: $\mathbb{R}^D \rightarrow \mathbb{R}$, Hessian is a $D \times D$ matrix

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_D} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_D x_1} & \frac{\partial^2 f}{\partial x_D x_2} & \cdots & \frac{\partial^2 f}{\partial x_D^2} \end{bmatrix}$$

Gives information about the curvature of the function at point $\boldsymbol{x}$

Note: If the function itself is vector valued, e.g., $f(\boldsymbol{x})$: $\mathbb{R}^D \rightarrow \mathbb{R}^K$ then we will have K such $D \times D$ Hessian matrices, one for each output dimension of f

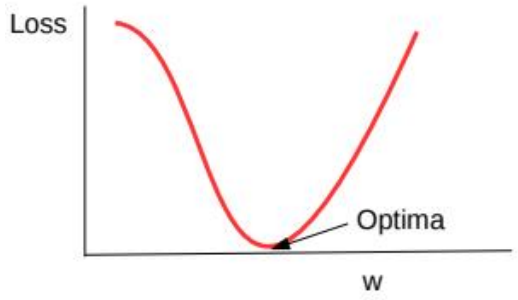
A square, symmetric $D \times D$ matrix M is PSD if $\boldsymbol{x}^\top M\boldsymbol{x} \geq \boldsymbol{0}\ \forall\ \boldsymbol{x} \in \mathbb{R}^D$
Will be NSD if $\boldsymbol{x}^\top M\boldsymbol{x} \leq \boldsymbol{0}\ \forall\ \boldsymbol{x} \in \mathbb{R}^D$

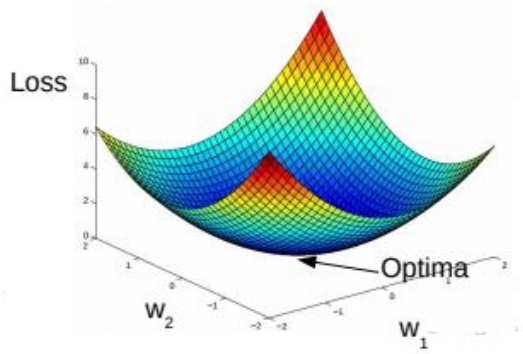
PSD if all eigenvalues are non-negative

- The Hessian matrix can be used to assess the optima/saddle points
  - $\nabla f(\boldsymbol{x}) = 0$ and $\nabla^2 f(\boldsymbol{x})$ is a positive semi-definite (PSD) matrix then $\boldsymbol{x}$ is a minima
  - $\nabla f(\boldsymbol{x}) = 0$, and $\nabla^2 f(\boldsymbol{x})$ is a negative semi-definite (NSD) matrix then $\boldsymbol{x}$ is a maxima

Convex and Non-Convex Functions

- A function being optimized can be either convex or non-convex
- Here are a couple of examples of convex functions

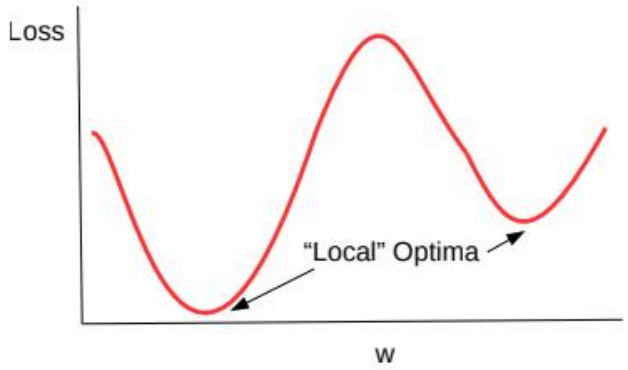


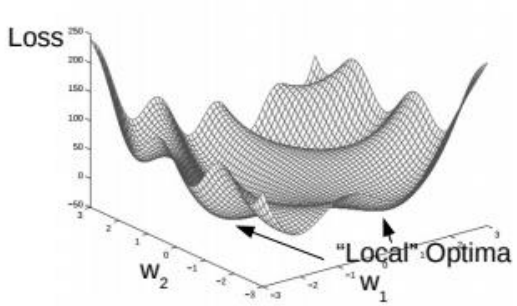


Convex functions are bowl-shaped. They have a unique optima (minima)

Negative of a convex function is called a concave function, which also has a unique optima (maxima)

- Here are a couple of examples of non-convex functions





Non-convex functions have multiple minima. Usually harder to optimize as compared to convex functions

Loss functions of most deep learning models are non-convex

Convex Sets

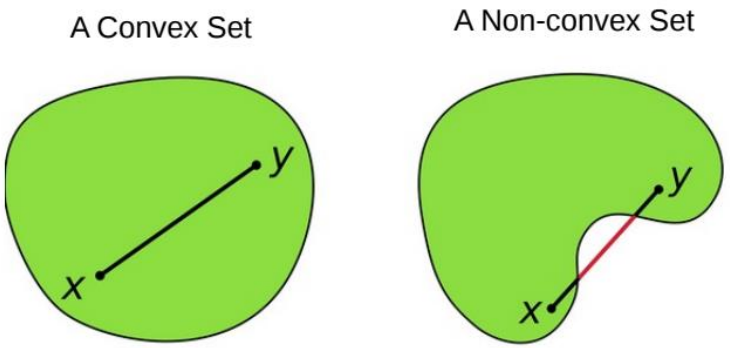
- A set S of points is a convex set, if for any two points $x, y \in S$, and $0 \leq \alpha \leq 1$

z is also called a "convex combination" of two points

$$z = \alpha x + (1 - \alpha) y \in S$$

Can also define convex combination of N points $x_1, x_2, \ldots, x_N$ as $z = \sum_{i=1}^{N} \alpha_i x_i$

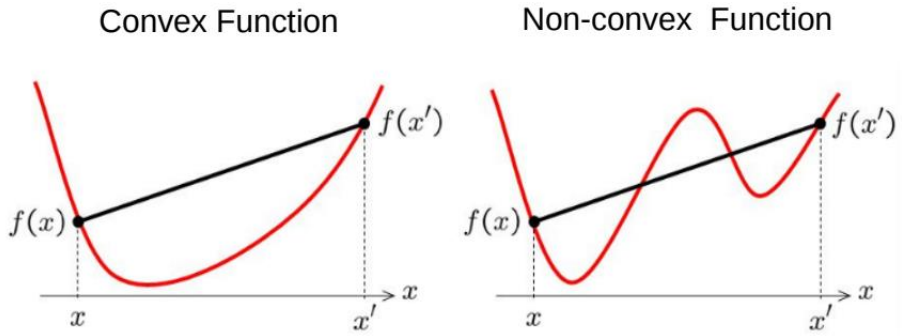
- Above means that all points on the line-segment between x and y lie within S



- The domain of a convex function needs to be a convex set

Convex Functions

- Informally, $f(x)$ is convex if all of its chords lie above the function everywhere



Note: "Chord lies above function" more formally means

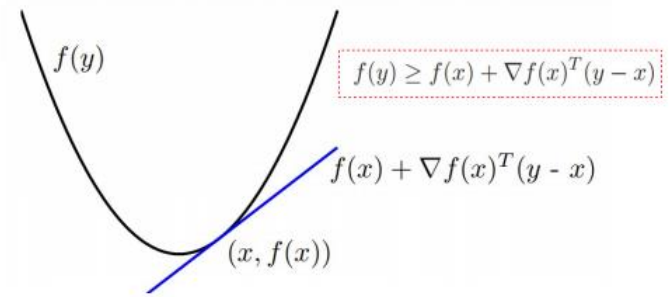
If f is convex then given

$$\alpha_1, \ldots, \alpha_n \text{ s.t } \sum_{i=1}^{n} \alpha_i = 1$$

$$f\left(\sum_{i=1}^{n} \alpha_i x_i\right) \leq \sum_{i=1}^{n} \alpha_i f(x_i)$$

Jensen's Inequality

- Formally, (assuming differentiable function), some tests for convexity:
  - First-order convexity (graph of f must be above all the tangents)



Exercise: Show that ridge regression objective is convex

  - Second derivative a.k.a. Hessian (if exists) must be positive semi-definite

Some Basic Rules for Convex Functions

- Some basic rules to check if $f(x)$ is convex or not
  - All linear and affine functions (e.g., $ax + b$) are convex
  - $\exp(ax)$ is convex for $x \in \mathbb{R}$, for any $a \in \mathbb{R}$
  - $\log(x)$ is concave (not convex) for $x > 0$
  - x^a is convex for $x > 0$, for any $a \geq 1$ and $a < 0$, concave for $0 \leq a \leq 1$
  - $|x|^a$ is convex for $x \in \mathbb{R}$, for any $a \geq 1$
  - All norms in $\mathbb{R}^D$ are convex
  - Non-negative weighted sum of convex functions is also a convex function
  - Affine transformation preserves convexity: if $f(x)$ is convex then $f(x) = f(ax + b)$ is also convex
  - Some rules to check whether composition $f(x) = h(g(x))$ of two functions h and g is convex

f is convex if h is convex and nondecreasing, and g is convex,
f is convex if h is convex and nonincreasing, and g is concave,
f is concave if h is concave and nondecreasing, and g is concave,
f is concave if h is concave and nonincreasing, and g is convex.

Coming up next

- Gradients when the function is non-differentiable
- Solving optimization problems
- Iterative optimization algorithms, such as gradient descent and its variants