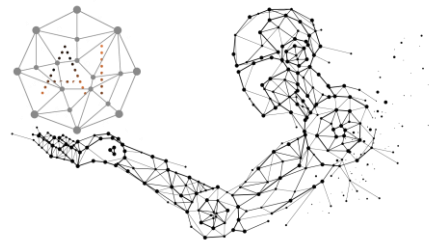
Applied Machine Learning

Chapter 5- Optimization

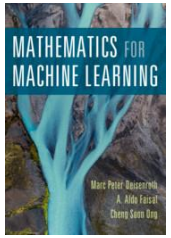
دانشگاه تربیت مدرس
Tarbiat Modares University

Hossein Homaei
Department of Electrical & Computer Engineering

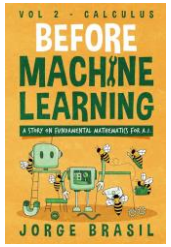


Some resources

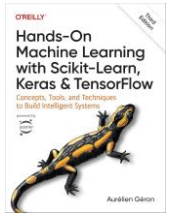
- Books



M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.



J. Brasil, *Before Machine Learning*, vol. 2, Calculus. 2023.



A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. O'Reilly Media, 2023.

… Some resources

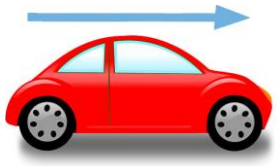
- Online
  - Calculus for Machine Learning and Data Science
    - Instructor: Luis Serrano
    - DeepLearning.AI
    - **Some parts of this lecture's content is sourced from this course.**

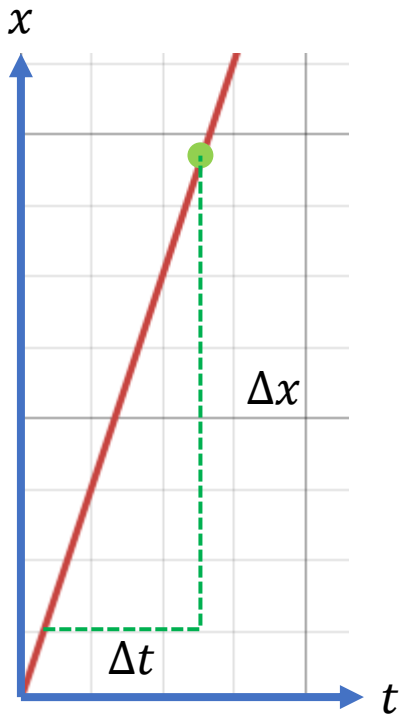
  - Differential Equations for Engineers
    - Instructor: Jeffrey R. Chasnov
    - The Hong Kong University of Science and Technology

A Fast Review of Derivatives

Introduction

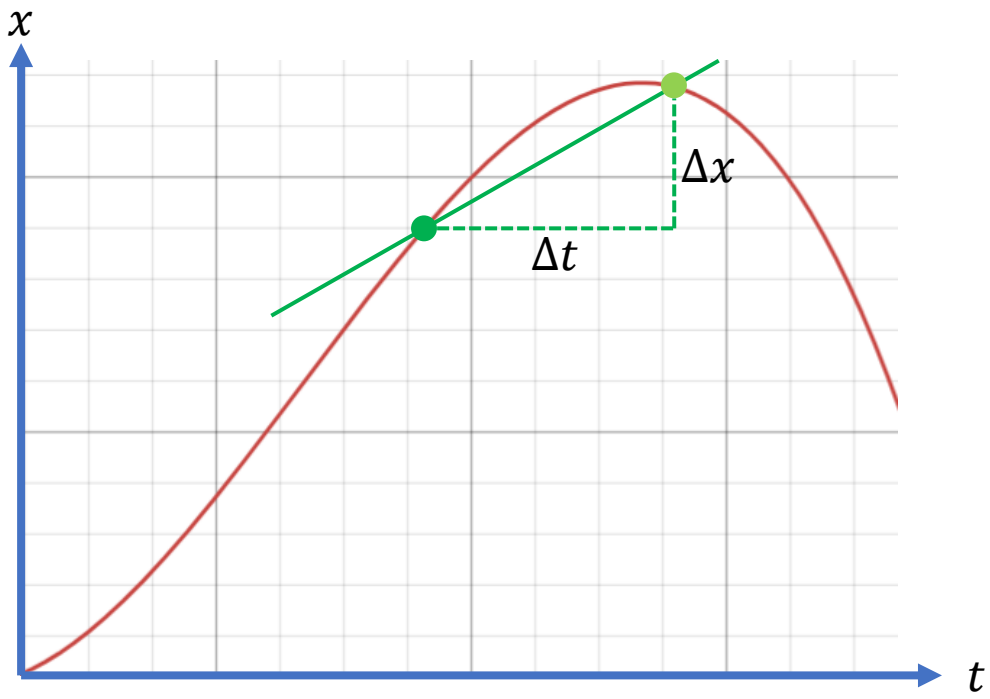
$$velocity = \frac{change\ in\ distance}{change\ in\ time} = \frac{\Delta x}{\Delta t} = slope$$





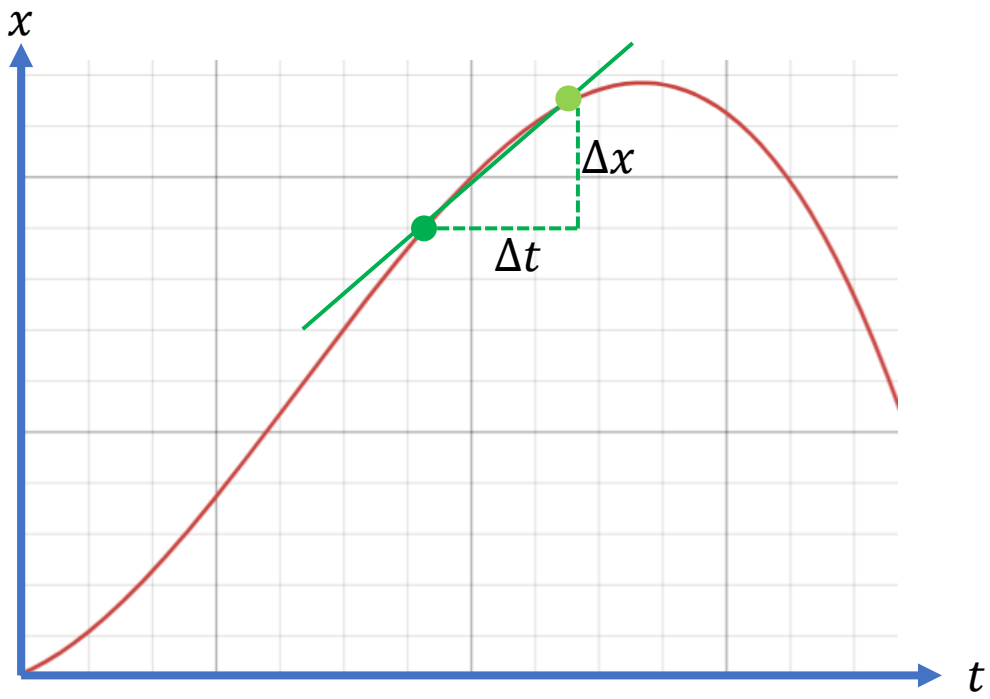
… Introduction

$$Slope = \frac{change\ in\ distance}{change\ in\ time} = \frac{\Delta x}{\Delta t}$$



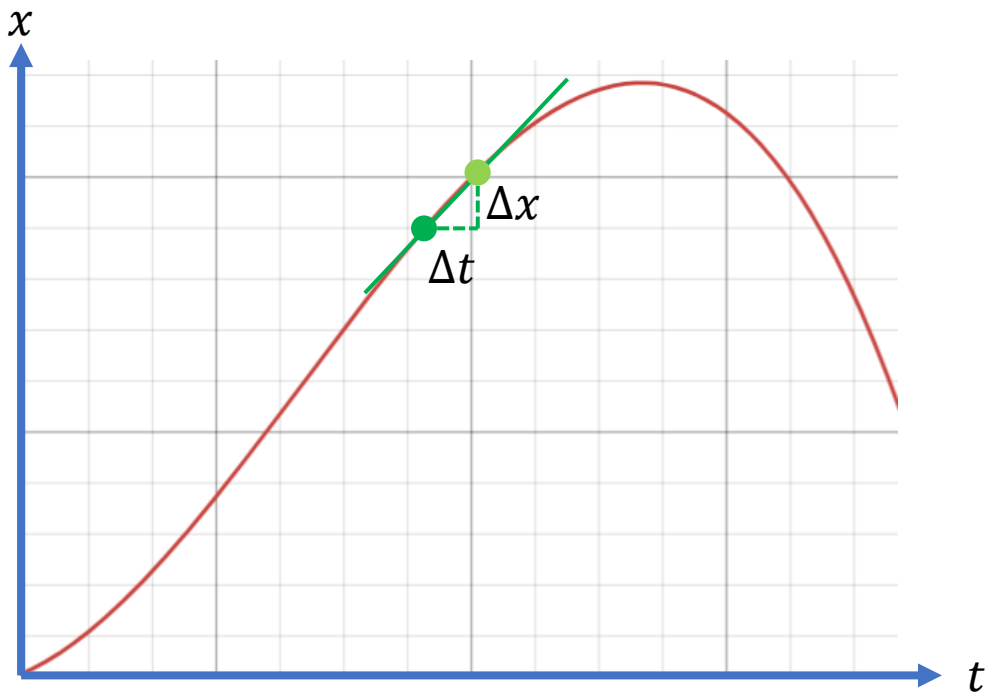
… Introduction

$$Slope = \frac{change\ in\ distance}{change\ in\ time} = \frac{\Delta x}{\Delta t}$$



… Introduction

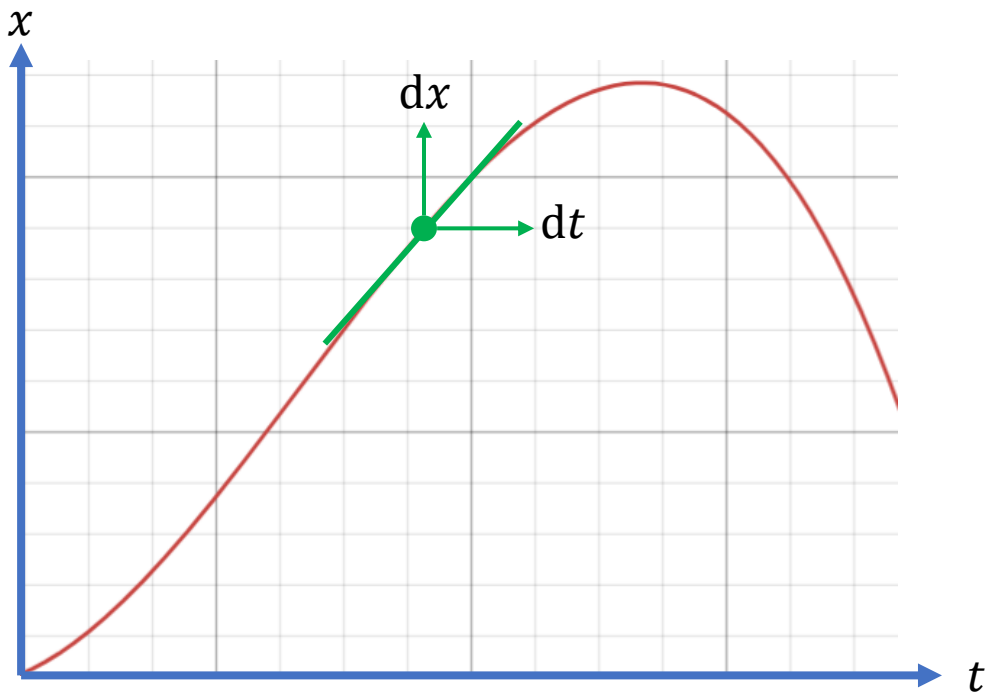
$$Slope = \frac{change\ in\ distance}{change\ in\ time} = \frac{\Delta x}{\Delta t}$$



… Introduction

$$Slope = \frac{change\ in\ distance}{change\ in\ time} = \frac{\Delta x}{\Delta t}$$

$$Slope\ at\ a\ point = \frac{\mathrm{d}x}{\mathrm{d}t}$$

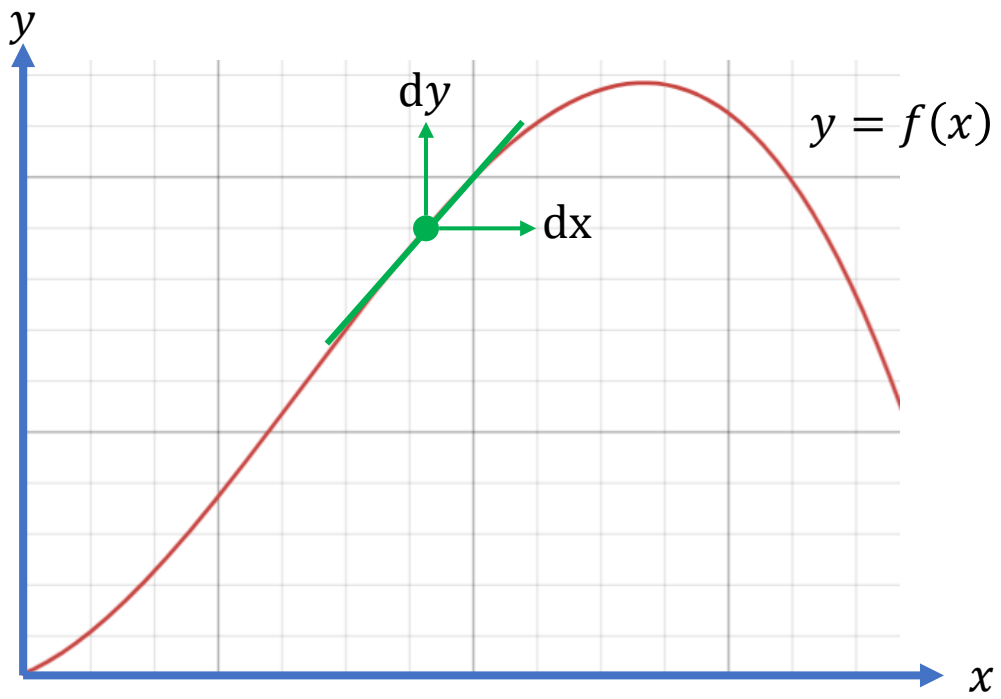


… Introduction

Derivative of f

Leibniz's notation: $\frac{dy}{dx} = \frac{d}{dx} f(x)$

Lagrange's notation: $f'(x)$



Common Derivatives

- Constants
  - $f(x) = c \Rightarrow f'(\mathrm{x}) = 0$
- Lines
  - $f(x) = ax + b \Rightarrow f'(x) = a$
- Quadratic Functions
  - $f(x) = x^2 \Rightarrow f'(x) = 2x$
- Quadratic Functions
  - $f(x) = \frac{1}{x} = x^{-1} \Rightarrow f'(x) = -x^{-2}$
- Power functions in general
  - $f(x) = x^n \Rightarrow f'(x) = nx^{n-1}$

… Common Derivatives

- Trigonometric functions
  - $f(x) = \sin(x) \Rightarrow f'(x) = \cos(x)$
  - $f(x) = \cos(x) \Rightarrow f'(x) = -\sin(x)$
- Exponential
  - $f(x) = e^x \Rightarrow f'(x) = e^x$
  - $f(x) = a^x \Rightarrow f'(x) = a^x \ln a$
- Logarithm
  - $f(x) = \ln x \Rightarrow f'(x) = \frac{1}{x} \quad , x > 0$
  - $f(x) = \log_a x \Rightarrow f'(x) = \frac{1}{x \ln a}$

From now on, whenever we say log, we mean ln!

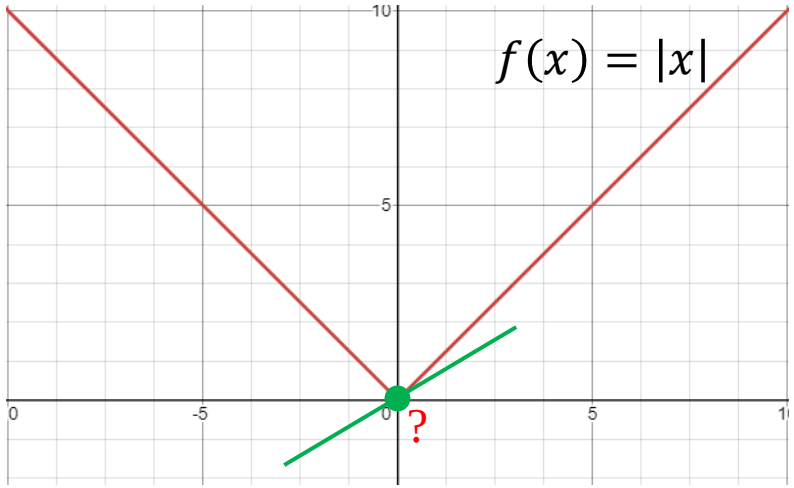
Some Differentiation rules

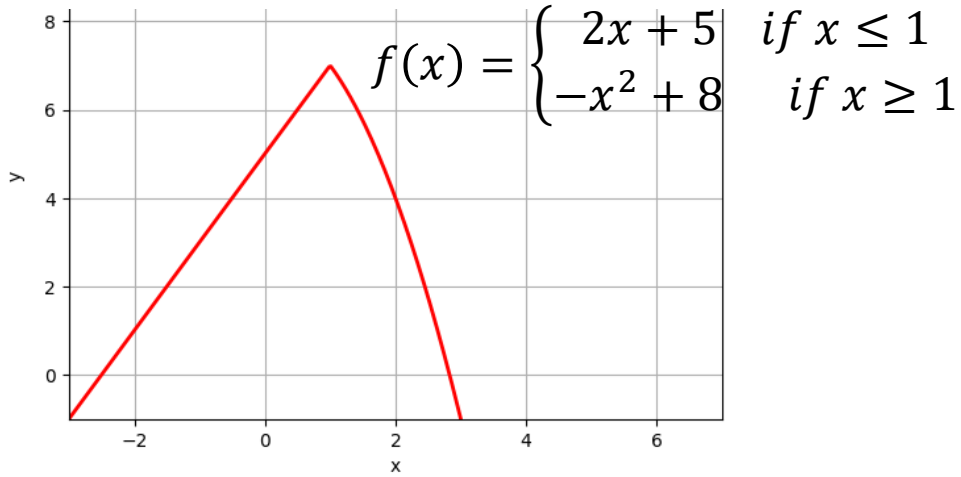
- Multiplication by scalar
  - If $f = cg$ (for some functions f and g, and a constant c), then $f'(x) = cg'(x)$
- Sum rule
  - If $f = g + \mathrm{h}$, then $f' = g' + h'$
- Product rule
  - If $f = g\mathrm{h}$, then $f' = g'h + gh'$
- Chain rule
  - If $\mathrm{l} = f\left(g\left(\mathrm{h}(t)\right)\right)$, then $\frac{df}{dt} = \frac{df}{dg}.\frac{dg}{dh}.\frac{dh}{dt}$
  - Or, $l' = f'\left(g\left(h(t)\right)\right).g'\left(h(t)\right).h'(t)$
  - Example: $f(x) = (3x+1)^5 \rightarrow f'(x) = 5(3x+1)^4.(3) = 15(3x+1)^4$

Exercise: prove that the derivative of $\frac{f(x)}{g(x)}$ is $\frac{g(x)f'(x) - f(x)g'(x)}{\left(g(x)\right)^2}$

Non Differentiable Function

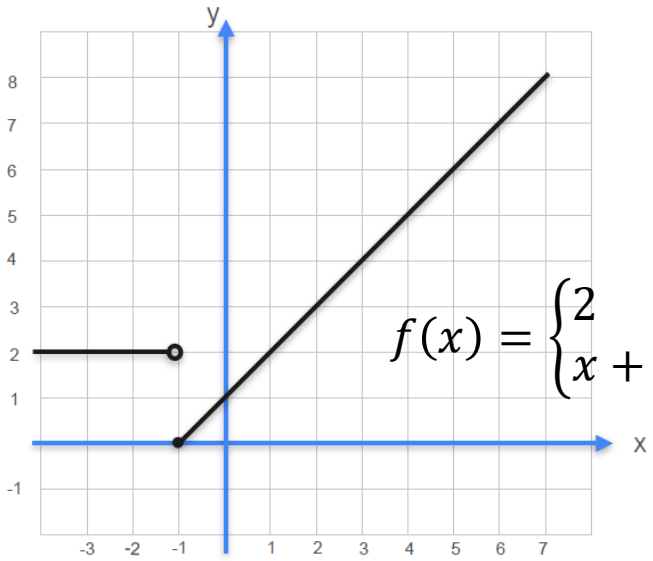
- Functions which have a corner or a cusp





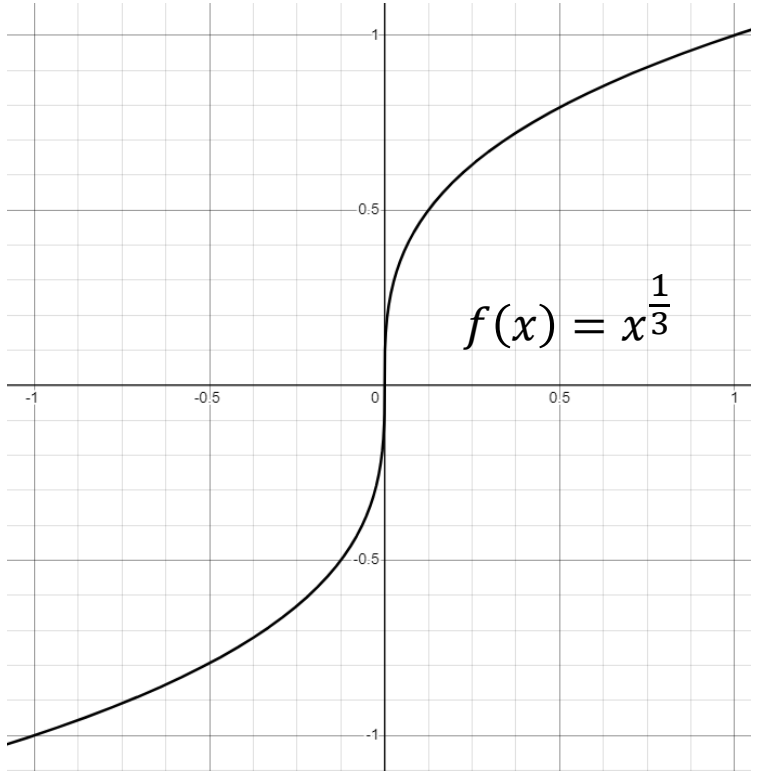
… Non Differentiable Function

- Jump Discontinuity



$$f(x) = \begin{cases} 2 & if\ x \leq -1 \\ x + 1 & if\ x \geq -1 \end{cases}$$

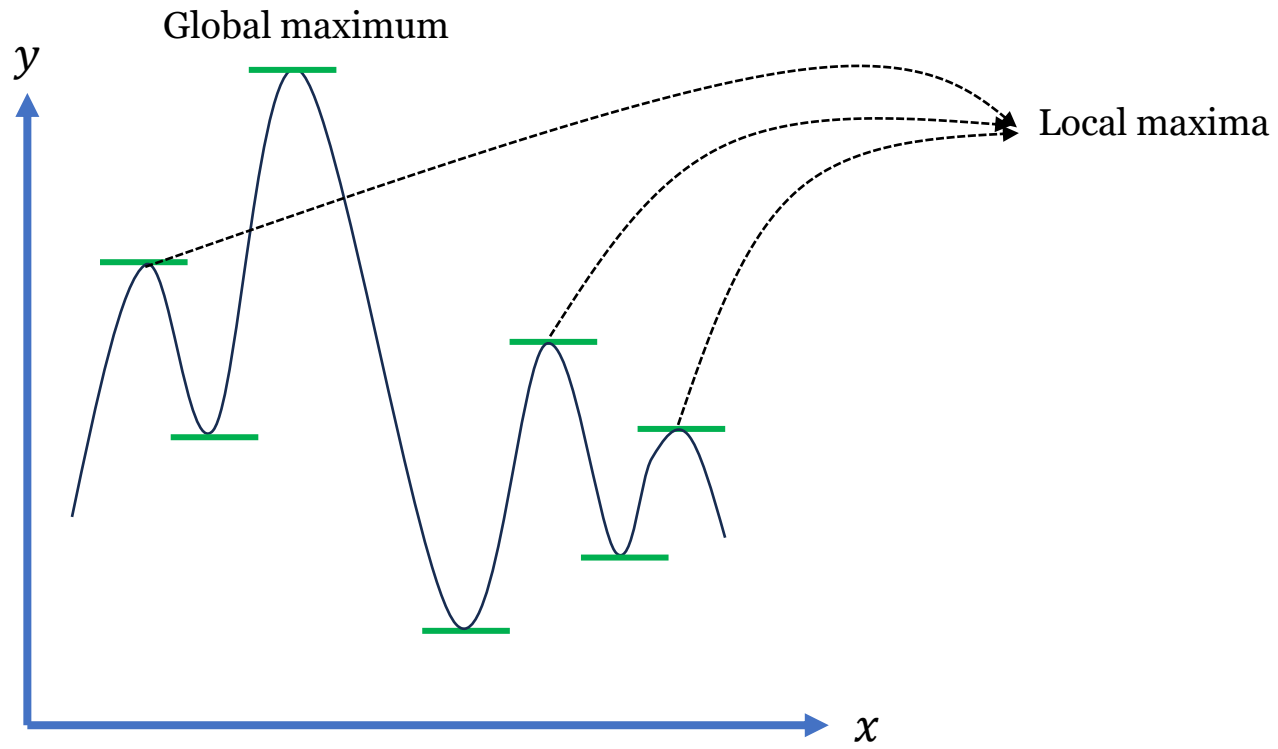
- Vertical tangents



$$f(x) = x^{\frac{1}{3}}$$

Minimum and Maximum

- At the minima and maxima of a function, the derivative is zero.



Partial Derivatives

$$f(x,y) = x^2 + y^2$$

Treat y as a constant

$$f_x = \frac{\partial f}{\partial x} = 2x$$

The partial derivative of f with respect to x

Treat x as a constant

$$f_y = \frac{\partial f}{\partial y} = 2y$$

The partial derivative of f with respect to y

… Partial Derivatives

- Example
  - $f(x,y) = 3x^2y^3$
    - $\frac{\partial f}{\partial x} = 6xy^3$
    - $\frac{\partial f}{\partial y} = 9x^2y^2$
  - $f(y_1,y_2,y_3) = 9\frac{y_1y_2y_3}{y_1+y_2+y_3}$
    - $\frac{\partial f}{\partial y_3} =?$
    - $\frac{\partial f}{\partial y_3} = 9\frac{(y_1+y_2)y_1y_2}{(y_1+y_2+y_3)^2}$
  - $f(x_1,x_2,x_3,x_4) = 3\frac{\cos(x_1x_4)\sin(x_2^5)}{e^{x_2+(1+x_2^2)/(x_1x_2x_4)}} + 5x_1x_3x_4$
    - $\frac{\partial f}{\partial x_3} =?$
    - $\frac{\partial f}{\partial x_3} = 5x_1x_4$

Gradient

- For a function that has n variables, the gradient of f (nabla f) is a vector of n entries, each one corresponding to the partial derivative of the function with respect to each one of the n variables.

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Example

$$f(x,y) = x^2 + y^2 \rightarrow \nabla f(x,y) = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Second Derivative

- The derivative of the derivative!

  Leibniz's notation: $\frac{d^2 f(x)}{dx^2} = \frac{d}{dx}\left(\frac{df(x)}{dx}\right)$

  Lagrange's notation: $f''(x)$

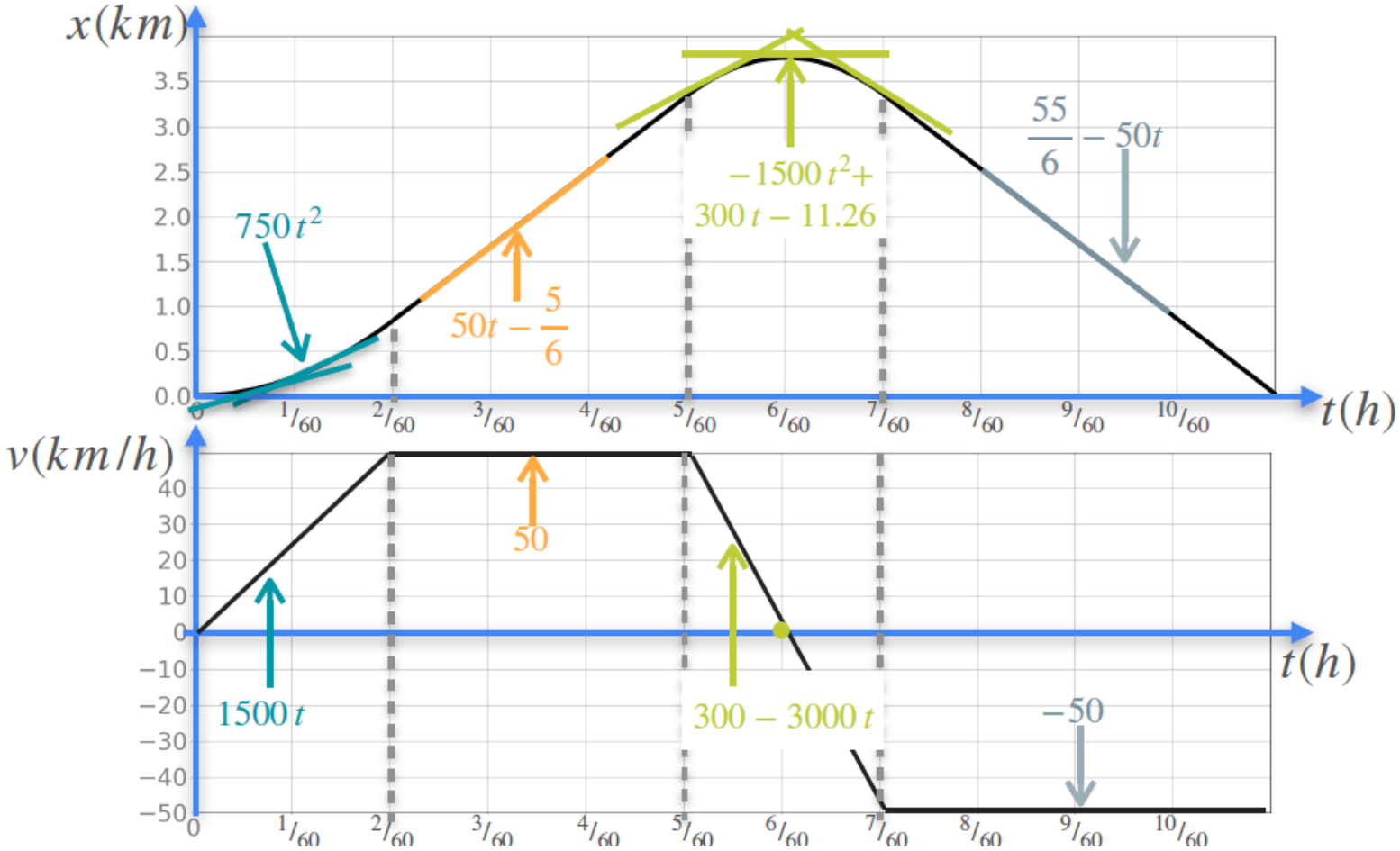
- Example: velocity and acceleration

$$v = \frac{dx}{dt}$$

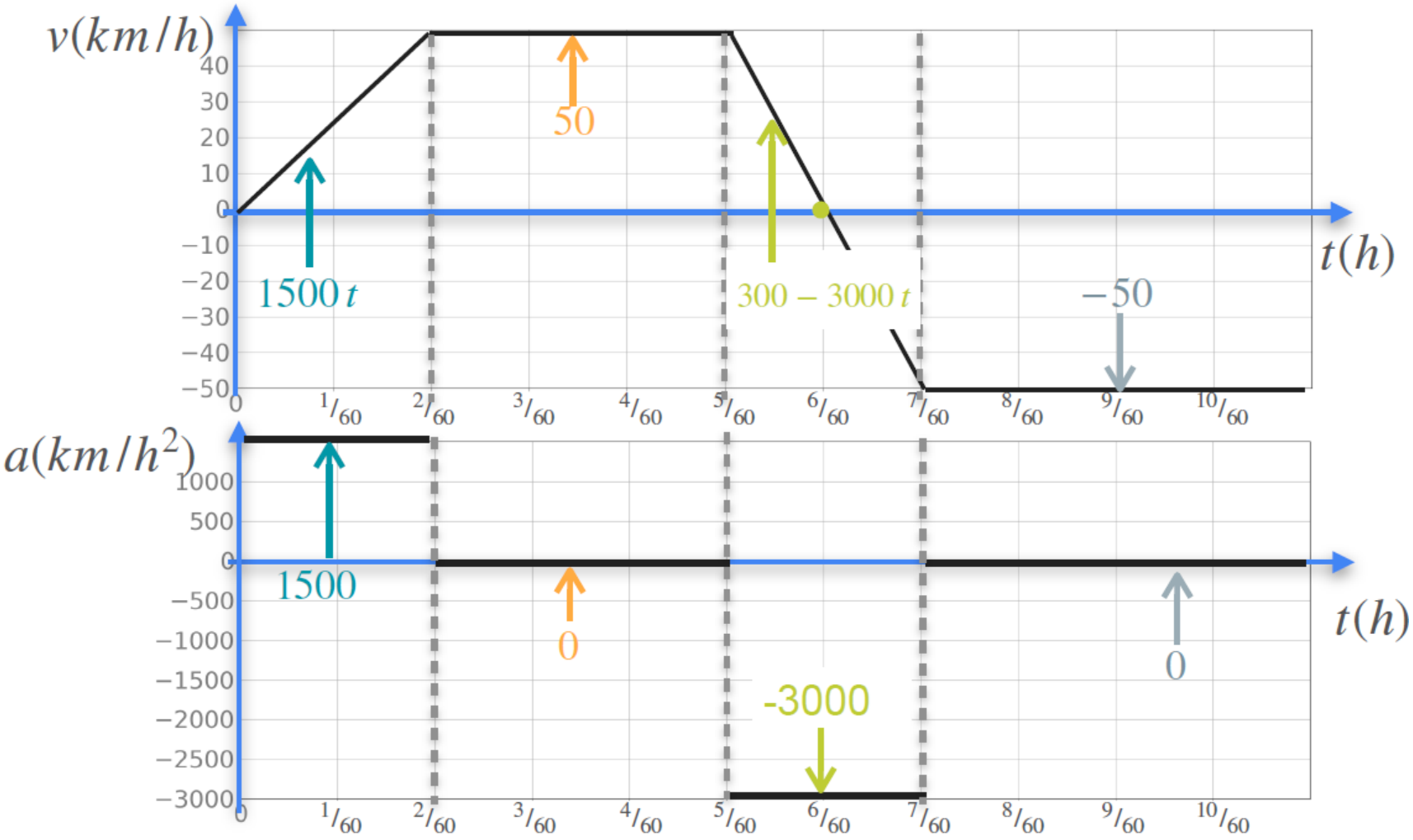
$$a = \frac{dv}{dt} = \frac{d^2 x}{dt^2}$$

  - Positive/Negative acceleration means the velocity is increasing/decreasing with respect to time. Zero means constant velocity.

… Second Derivative

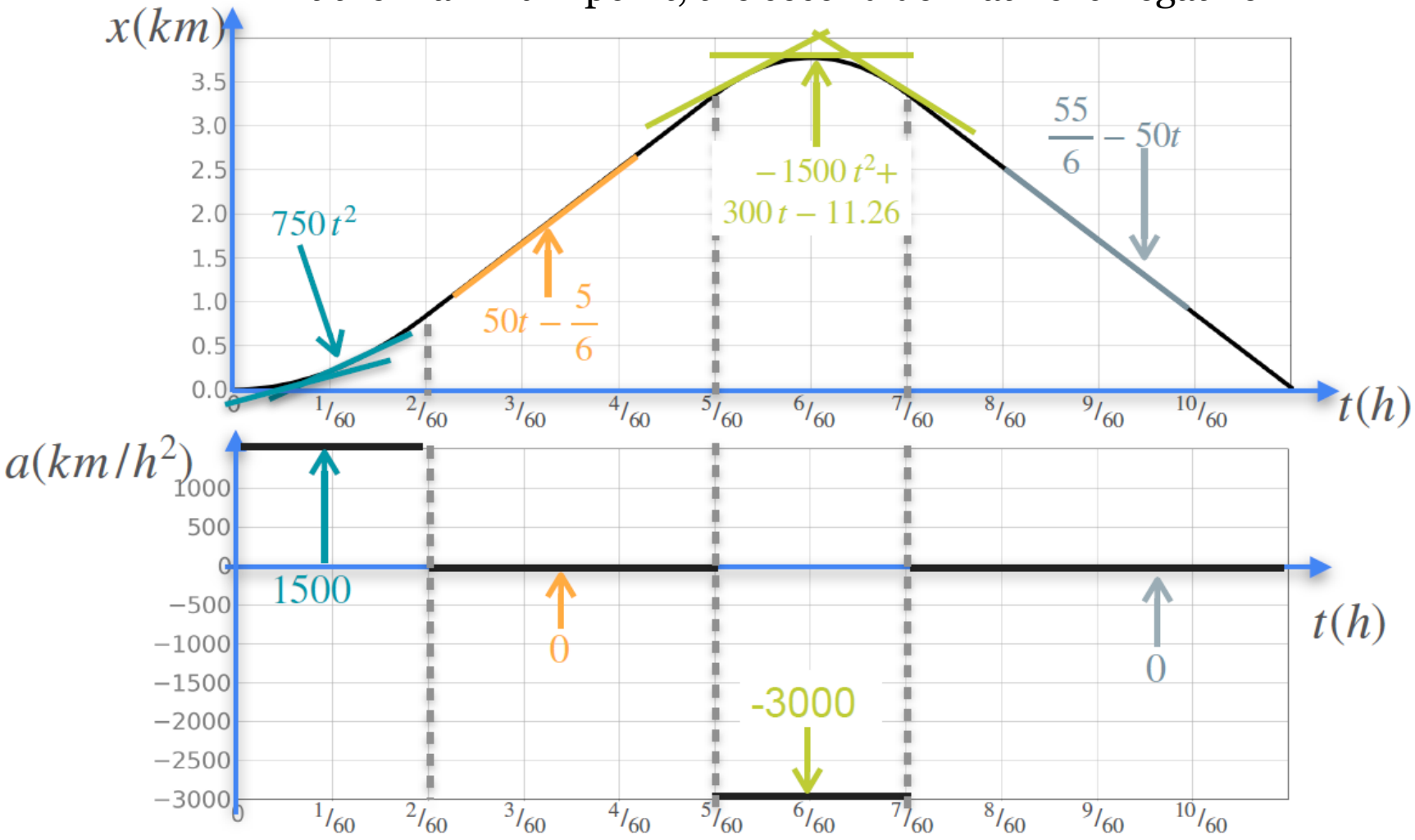


… Second Derivative



… Second Derivative

At the maximum point, the second derivative is negative



Hessian Matrix

	1 variable	2 variables
Function	$f(x)$	$f(x,y)$
1st derivative	$f'(x)$ ☞ Rate of change of $f(x)$	$\frac{\partial}{\partial x}f(x,y)$ ☞ Rate of change with respect to x $\frac{\partial}{\partial y}f(x,y)$ ☞ Rate of change with respect to y $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$
2nd derivative	$f''(x)$ ☞ Rate of change of the rate of change of $f(x)$	$\frac{\partial^2 f(x,y)}{\partial x^2}$ ☞ Rate of change of $f'_x(x,y)$ with respect to x $\frac{\partial^2 f(x,y)}{\partial y \partial x}$ ☞ Rate of change of $f'_x(x,y)$ with respect to y $\frac{\partial^2 f(x,y)}{\partial x \partial y}$ ☞ Rate of change of $f'_y(x,y)$ with respect to x $\frac{\partial^2 f(x,y)}{\partial y^2}$ ☞ Rate of change of $f'_y(x,y)$ with respect to y Hessian matrix = $H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{bmatrix}$

…Hessian Matrix

- For 3 variables x_1, x_2, x_3

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix}$$

…Hessian Matrix

- For n variables $x_1, x_2, \dots, x_n$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Square & Log Loss

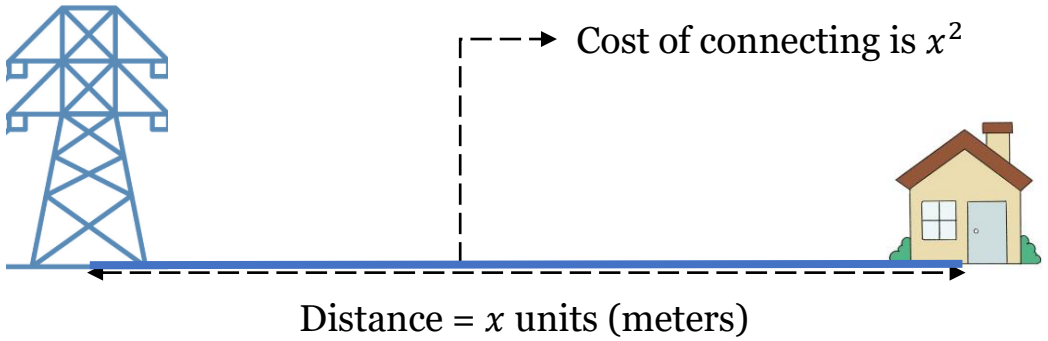
Described by Examples

Example 1- Powerline Problem

- Problem:
  - Building a house in a place that minimizes the cost of connecting it to several power lines (e.g. three lines) .
  - We should connect the house to all lines for better power supply.

… Example 1- Powerline Problem

- Assume that the cost is related to the squared distance from the power line



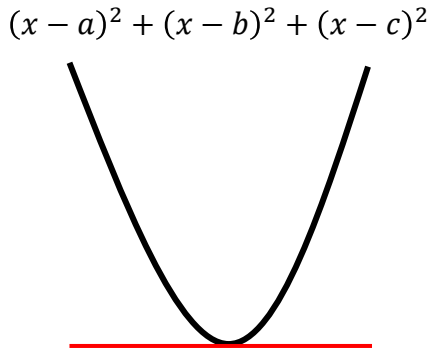
- Importance:
  - This problem is similar to the cost minimization problem in linear regression and some neural networks.

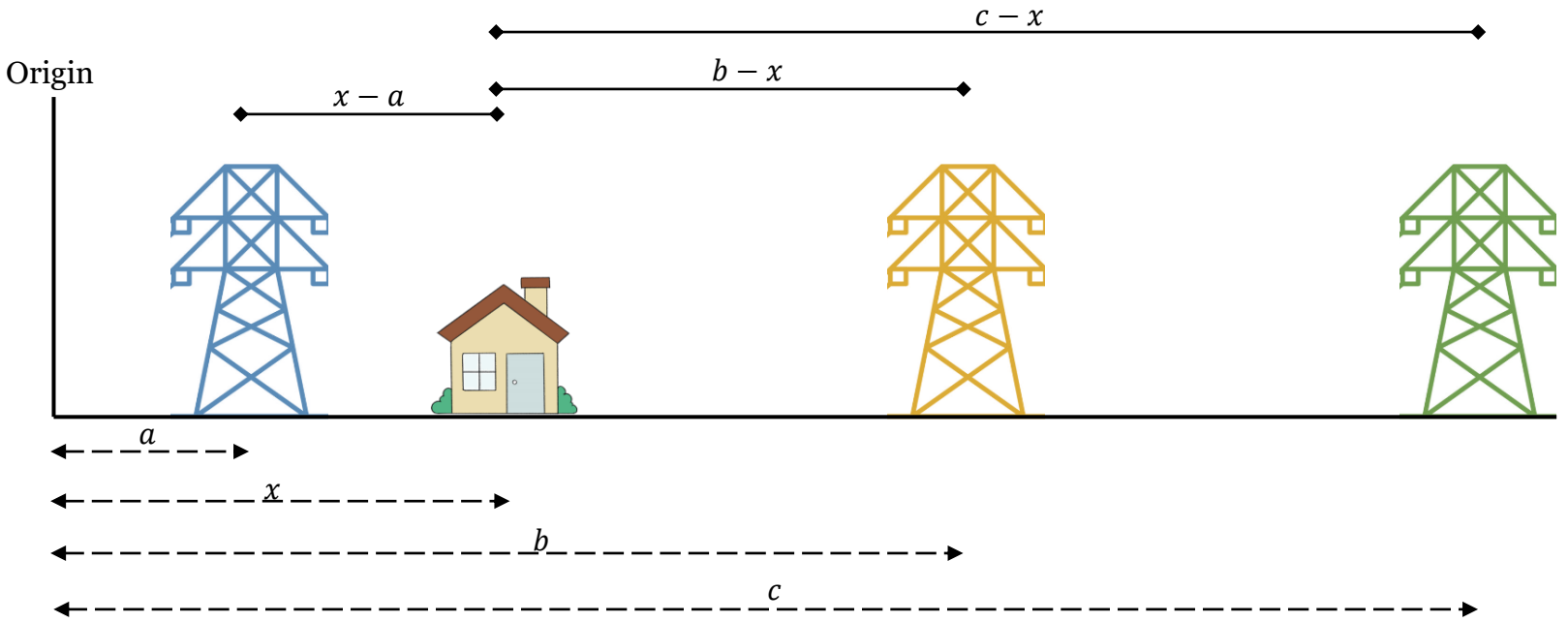
… Example 1- Powerline Problem

$$f = cost\ function = (x-a)^2 + (x-b)^2 + (x-c)^2$$

$$f' = 2(x-a) + 2(x-b) + 2(x-c) = 0$$

$$3x - a - b - c = 0 \rightarrow x = \frac{a+b+c}{3}$$





… Example 1- Powerline Problem

- Generalized problem
  - The square loss
  - Minimize $(x - a_1)^2 + (x - a_2)^2 + \cdots + (x - a_n)^2$

- Solution
  - $x = \frac{a_1 + a_2 + \cdots + a_n}{n}$

Example 2- Coin Toss

- Problem
  - Toss a coin 10 times
  - If the results are 7 head followed by three tail, we will win money!
  - Otherwise, don't win any money

  - We should pick a coin among lots of coin
    - We can pick a biased coin
      - E.g. head probability = 0.3 and tail probability = 0.7
    - Which coin would be the best choice?

… Example 2- Coin Toss

- Assume that we choose the coin that the probability of landing in heads and tails are p and $1-p$ respectively.
- Find the optimal value for p that maximize the probability of winning

$$f(p) = chance\ of\ winning = p^7(1-p)^3$$

$$f'(p) = 7p^6(1-p)^3 - 3p^7(1-p)^2$$

$$f'(p) = p^6(1-p)^2[7(1-p) - 3p]$$

$$f'(p) = p^6(1-p)^2(7-10p) = 0$$

$$p = \begin{cases} 0 & \text{✗} \\ 1 & \text{✗} \\ 0.7 & \end{cases}$$

… Example 2- Coin Toss

- Is there any easier way to solve the problem?
  - Take the logarithm of $f(p)$
    - If $f(p)$ is maximum, then so it is the $\log\big(f(p)\big)$
      - Maximizing $\log\big(f(p)\big)$ is the same thing as maximizing $f(p)$
    - Why logarithm?
      - The logarithm of a product is the sum of the logarithms
        - Derivative of product is hard, derivative of sum is easy.
      - Product of tiny things ($0 \leq p \leq 1$) is tiny.
        - It is difficult for computers to handle these tiny numbers.

$$G(p) = \log\big(f(p)\big) = \log(p^7(1-p)^3) = \log(p^7) + \log\big((1-p)^3\big) = 7\log(p) + 3\log(1-p)$$

$$G'(p) = 7\frac{1}{p} + 3\frac{1}{1-p}(-1) = 0$$

$$\frac{7(1-p) - 3p}{p(1-p)} = 0 \rightarrow p = 0.7$$

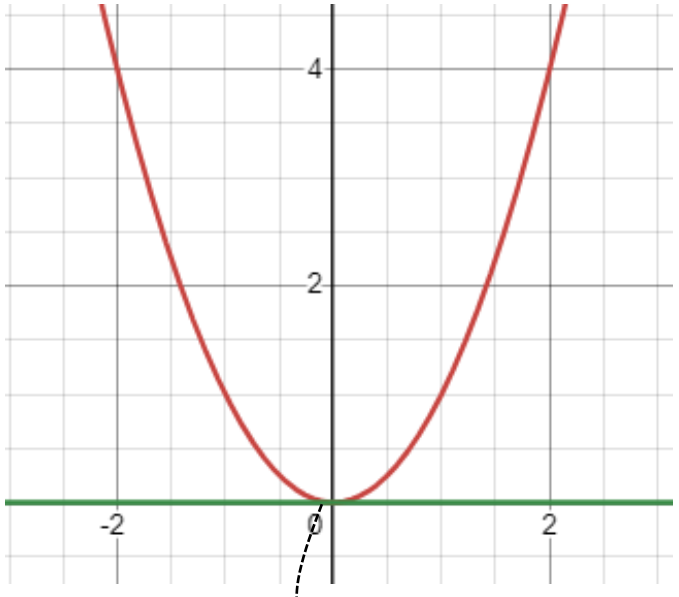
… Example 2- Coin Toss

- Generalized problem
  - The log loss
  - Minimize $-\log\big(f(p)\big)$
    - Instead of maximizing $\log\big(f(p)\big)$

- Importance
  - Useful in lot of classification problems

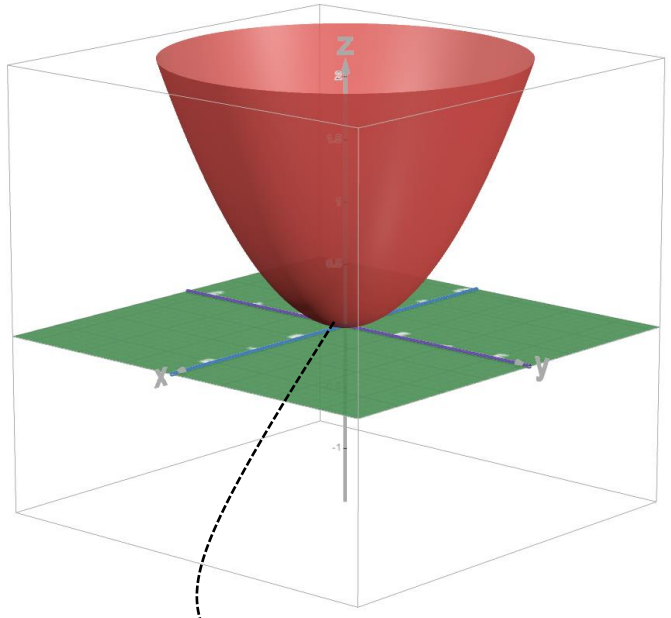
Gradient Descent

Gradient & Maxima/Minima



Minimum: slope = 0

$f'(x) = 0$



Minimum: the tangent plane is parallel to the floor

both slopes = 0

$$\frac{\partial f}{\partial x} = 0, \frac{\partial f}{\partial y} = 0$$

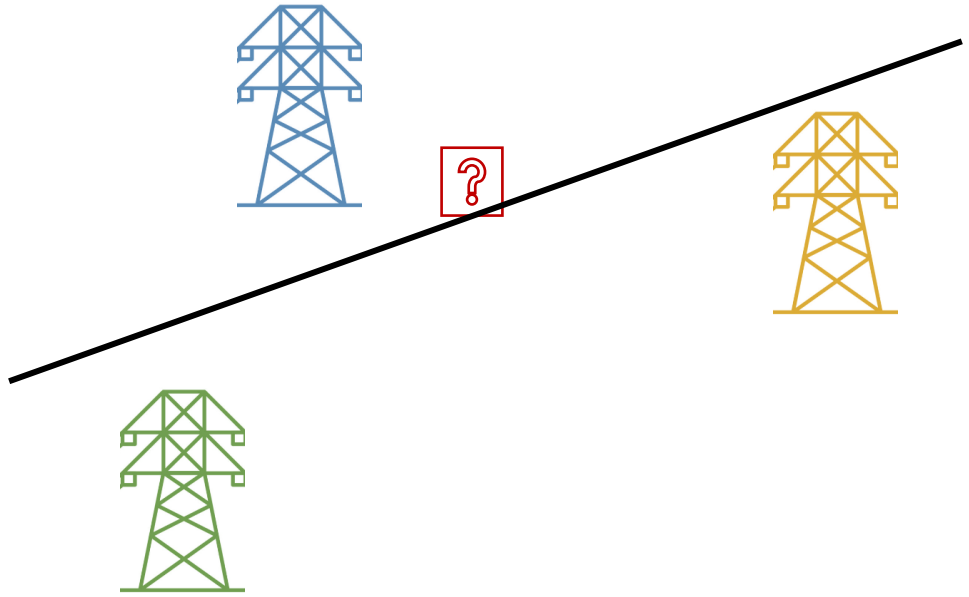
… Gradient & Maxima/Minima

- Example
  - $f(x,y) = 85 - \frac{1}{90}x^2(x-6)y^2(y-6)$
  - $\frac{\partial f}{\partial x} = -\frac{1}{90}x(3x-12)y^2(y-6) = 0 \rightarrow x = 0, x = 4, y = 0, y = 6$
  - $\frac{\partial f}{\partial y} = -\frac{1}{90}x^2(x-6)y(3y-12) = 0 \rightarrow \mathrm{x} = 0, \mathrm{x} = 6, \mathrm{y} = 0, \mathrm{y} = 4$

    - Based on other limitations of the problem we can choose the appropriate combinations.
      - Example of other constraints: $0 \leq x, y \leq 5$

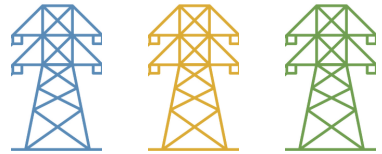
Example- Powerline Problem

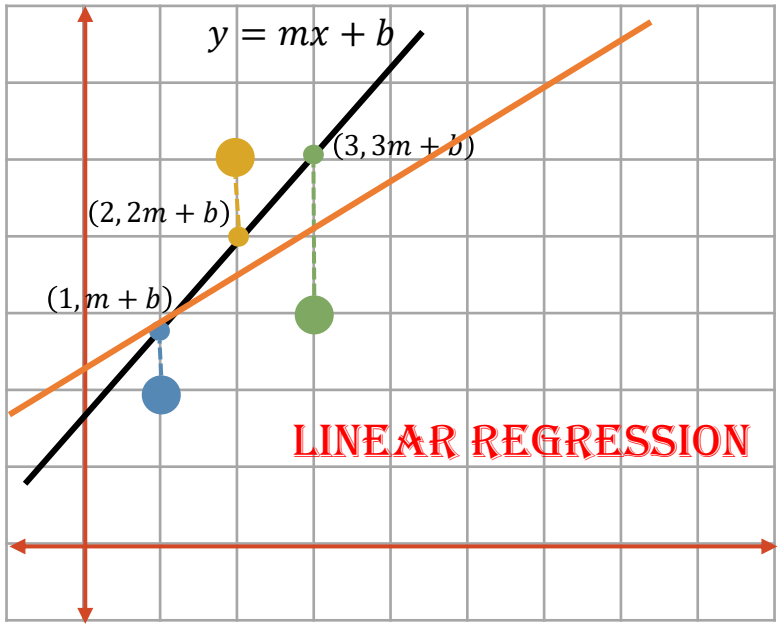
- Problem:
  - Find the optimal place for a fiber line connection that goes in a straight line in such a way that you reduce the total cost of connecting to the three power lines.



… Example- Powerline Problem

- Assumptions
  - The powerlines are located at (1,2), (2,5), and (3,3) coordinates.
  - The wire needs to be parallel to the y-axis .
  - The cost for connection is the square of the length of the wires.





Minimize the cost function

$$E(m,b) = (m + b - 2)^2 + (2m + b - 5)^2 + (3m + b - 3)^2$$
$$= 14m^2 + 3b^2 + 38 + 12mb - 42m - 20b$$

Solve the following system of linear equations:

$$\begin{cases} \frac{\partial E}{\partial m} = 28m + 12b - 42 = 0 \\ \frac{\partial E}{\partial b} = 6b + 12m - 20 \end{cases} \Rightarrow m = \frac{1}{2}, b = \frac{7}{3}$$

The Minimum cost is $E\left(\frac{1}{2}, \frac{7}{3}\right) = 4.167$

… Example- Powerline Problem

- Problem!
  - Solving the system of equations for two unknowns is pretty hard.
  - What happens when you have many variables?
    - It can be expensive!
  - What happens when you want to find the min/max of a complex function?
    - It may be impossible!

  - Is there a faster method to find the minimum of a function?
    - Yes, Gradient Descent

Gradient Descent

- Problem description
  - Find the minimum of $f(x) = e^x - \log(x)$
    - Hard to solve analytically

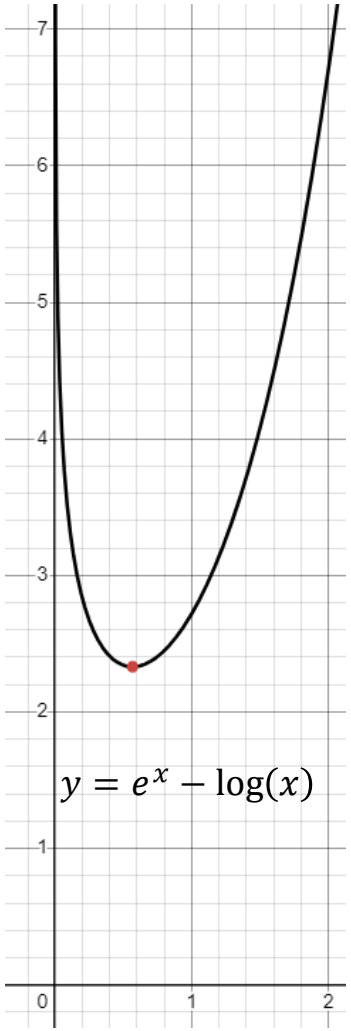
$$f'(x) = e^x - \frac{1}{x} = 0$$

$$\Rightarrow e^x = \frac{1}{x}$$

$$\Rightarrow ?$$

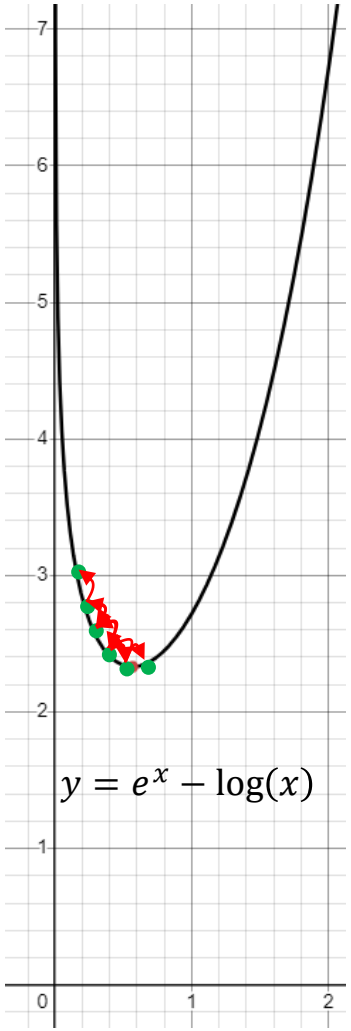
$$x = 0.5671\ldots$$

Omega constant (Ω)



… Gradient Descent

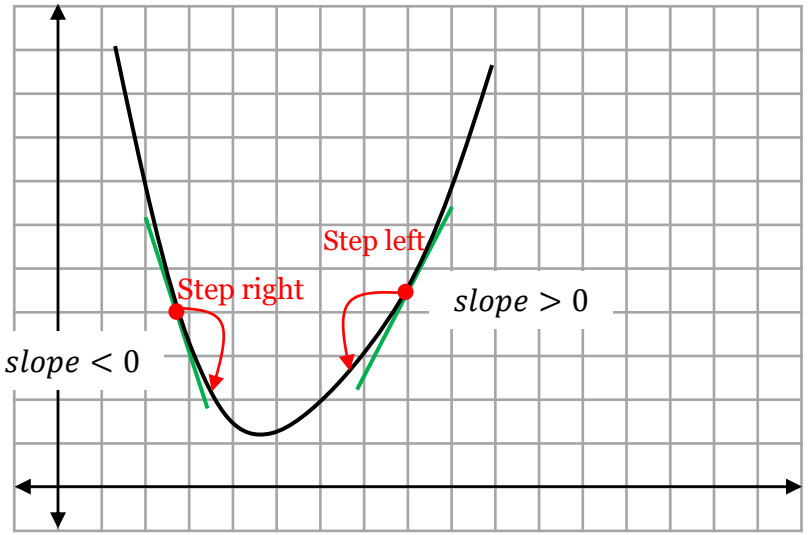
- Solving method
  - Pick some random value
  - Move to the left and right a little bit
  - Choose the better point
  - Repeat until no better point found
  - Result: the point that is close enough to the minimum



… Gradient Descent

- Be smarter!
  - New point = old point − slope

$$x_1 = x_0 - f'(x_0)$$



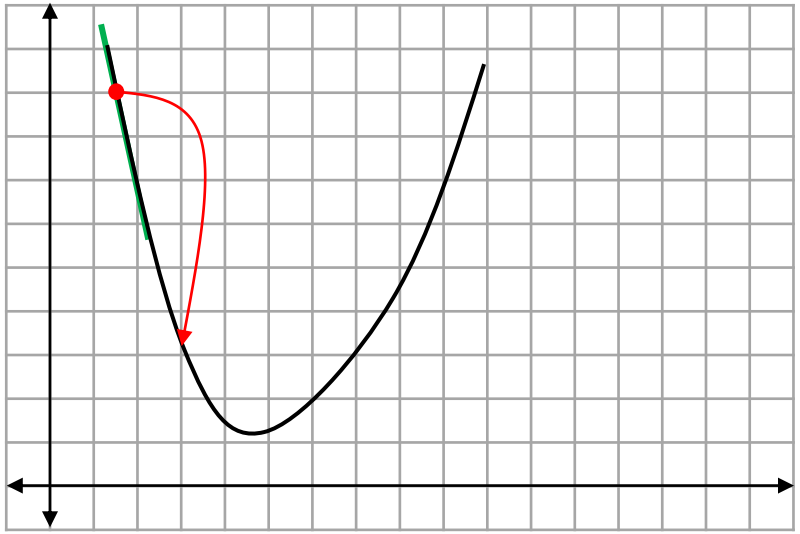
… Gradient Descent

- … Be smarter!
  - What happens if you are at a very steep part of the curve?
    - The derivative at the point is very large
    - Make a long jump

$$x_1 = x_0 - f'(x_0)$$

$$x_1 = x_0 - \alpha f'(x_0)$$

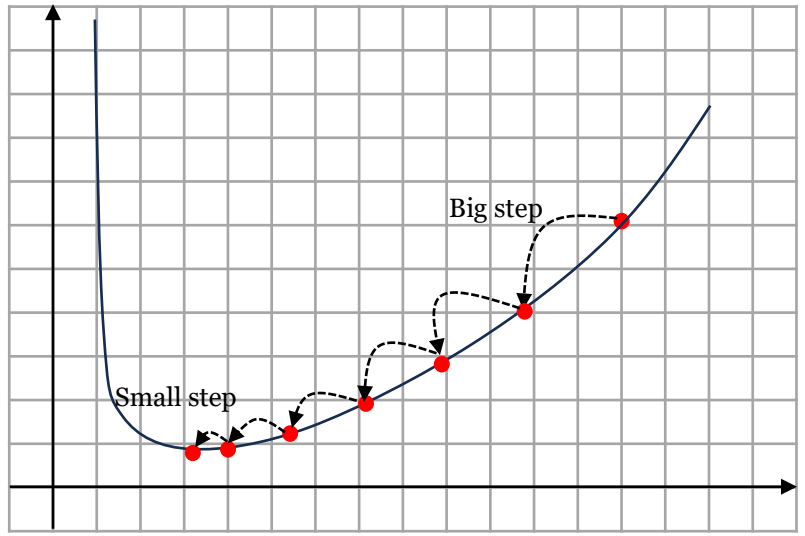
Learning rate



Example: $x_1 = x_0 - 0.01 f'(x_0)$

… Gradient Descent

- Gradient Descent (GD) Algorithm
  - Input: Function $f(x)$
  - Output: Minimum of $f(x)$
  - Step-1:
    - Define a learning rate α
    - Choose a starting point x_0
  - Step-2: Update $x_k = x_{k-1} - \alpha f'(x_{k-1})$
  - Step-3: Repeat step-2 until you are close enough to the true minimum
    - When your steps don't really change that much.
    - The algorithm converges to a minimum

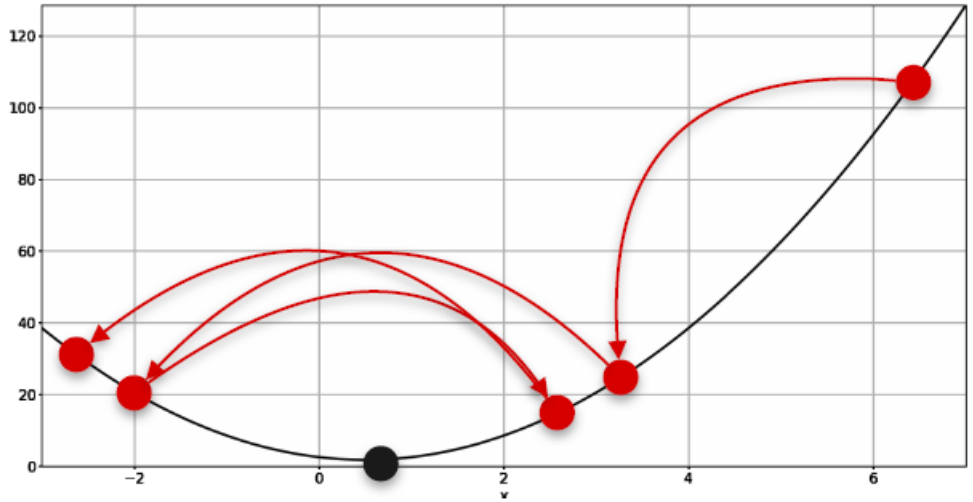


We do not need to solve the equation $f'(x) = 0$
We only need to know $f'(x)$

GD Challenges

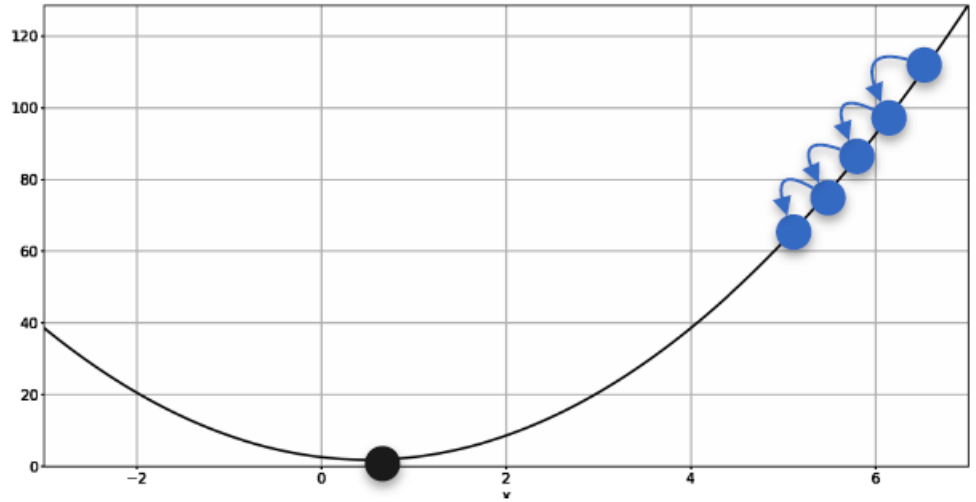
- Finding a good learning rate is influential
- There is no definite method to give the best learning rate.

Too large



Miss the minimum

Too small

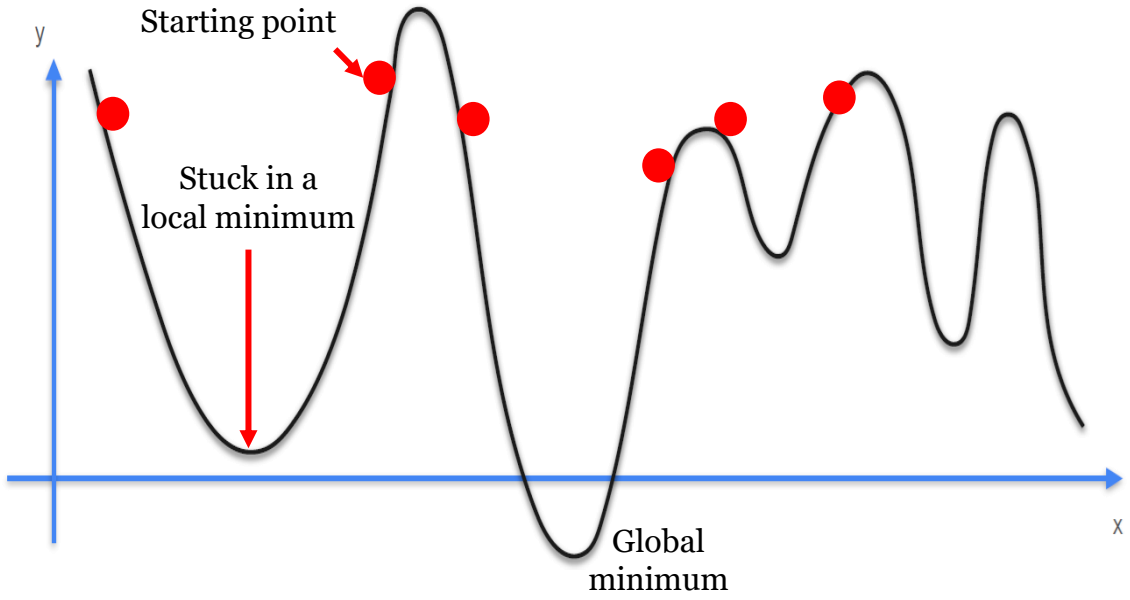


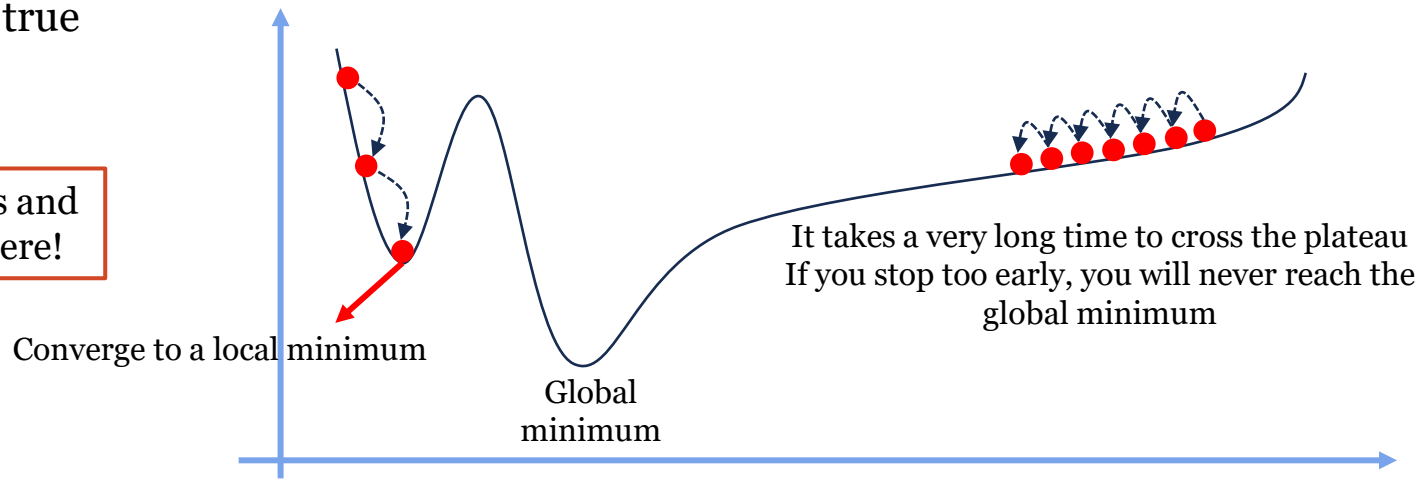
Take forever or never reach the minimum

… GD Challenges

- Problem: Finding the global minimum
  - Stuck in a local minimum
  - Different convergence time
- How to solve?
  - There is no secure way
  - But, there is a method to get a better result
    - Run the GD algorithm many times with different starting points.
      - Increase the chance of finding the true minimum

There exist more techniques to address the flat regions and avoid poor initial points; But, we do not cover them here!





GD for Multiple Variables

- Algorithm for two variables (x, y)
  - Input: Function $f(x, y)$
  - Output: Minimum of $f(x, y)$
  - Steps:
    - Step-1:
      - Define a learning rate α
      - Choose a starting point (x_0, y_0)
    - Step-2: Update $\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} - \alpha \nabla f(x_{k-1}, y_{k-1})$
    - Step-3: Repeat step-2 until you are close enough to the true minimum (x_{min}, y_{min})
      - When your steps don't really change that much.
- More variables are the same

… GD for Multiple Variables

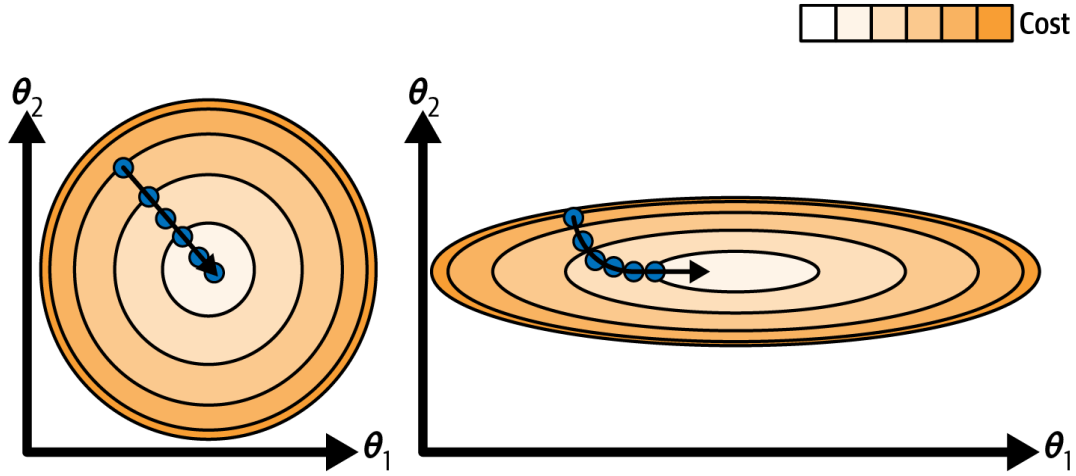
- Example- Solving the powerline problem by GD
  - Minimize the cost function

$$E(m,b) = 14m^2 + 3b^2 + 38 + 12mb - 42m - 20b$$

- Solution
  - Start with (m_0, b_0)
  - Iterate $(m_{k+1}, b_{k+1}) = (m_k, b_k) - \alpha \nabla E(m_k, b_k)$
    - Where $\nabla E = \begin{bmatrix} 28m + 12b - 42 \\ 6b + 12m - 20 \end{bmatrix}$

… GD for Multiple Variables

- Problem: feature scaling impact
  - Features have the same scale
    - The GD algorithm goes straight toward the minimum
  - Features have very different scales
    - The GD algorithm goes in a direction almost orthogonal to the direction of the global minimum, and it ends with a long march down the flat valley.
    - Take a long time
- Solution
  - Ensure that all features have a similar scale.



Reference: A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd ed. O'Reilly Media, 2023.

GD for Linear Regression

- Model
  - $f_{w,b}(x) = wx + b$
- Cost function (squared error)
  - $J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)^2$
- (Batch) Gradient Descent Algorithm
  - Repeat until convergence

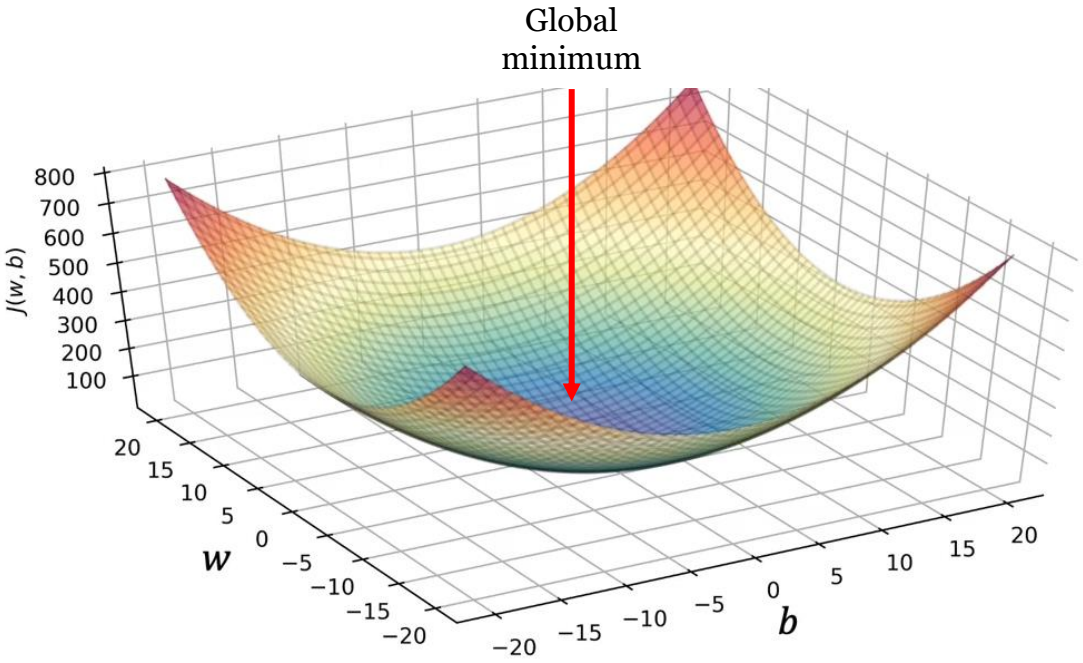
Each step of GD uses all training examples
makes it very slow when the training set is large

$wx^{(i)} + b$

$$\begin{cases} w = w - \alpha\frac{\partial}{\partial w}J(w,b) = w - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)\left(x^{(i)}\right) \\ b = b - \alpha\frac{\partial}{\partial b}J(w,b) = b - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right) \end{cases}$$

… GD for Linear Regression

- Good news!
  - The squared error cost function has a bowl-shaped curve
  - It has a single global minimum
    - We will not trap in local minimums



Slides of Machine Learning Specialization, presented on Coursera, by Andrew Ng, DeepLearning.AI.

Types of GD

- Min square error cost function
  - $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$
    - h_θ is he hypothesis function (the predicted value)
    - $y^{(i)}$ is the actual value
    - θ represents the parameters

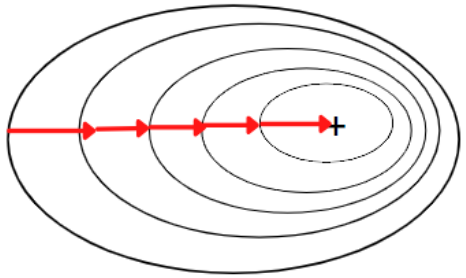
… Types of GD

Batch GD

Explained ✓

Update rule for each parameter θ_j

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

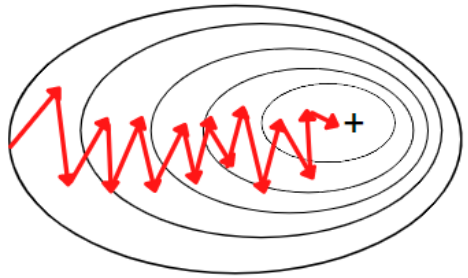


Stochastic GD

Picks a random instance in the training set at every step and computes the gradients based only on that single instance.

Update rule for each parameter θ_j

$$\theta_j = \theta_j - \alpha \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

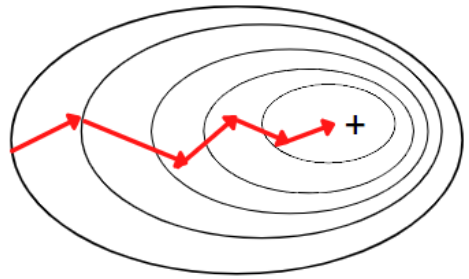


Mini-batch GD

Computes the gradients on small random sets of instances

Update rule for each parameter θ_j

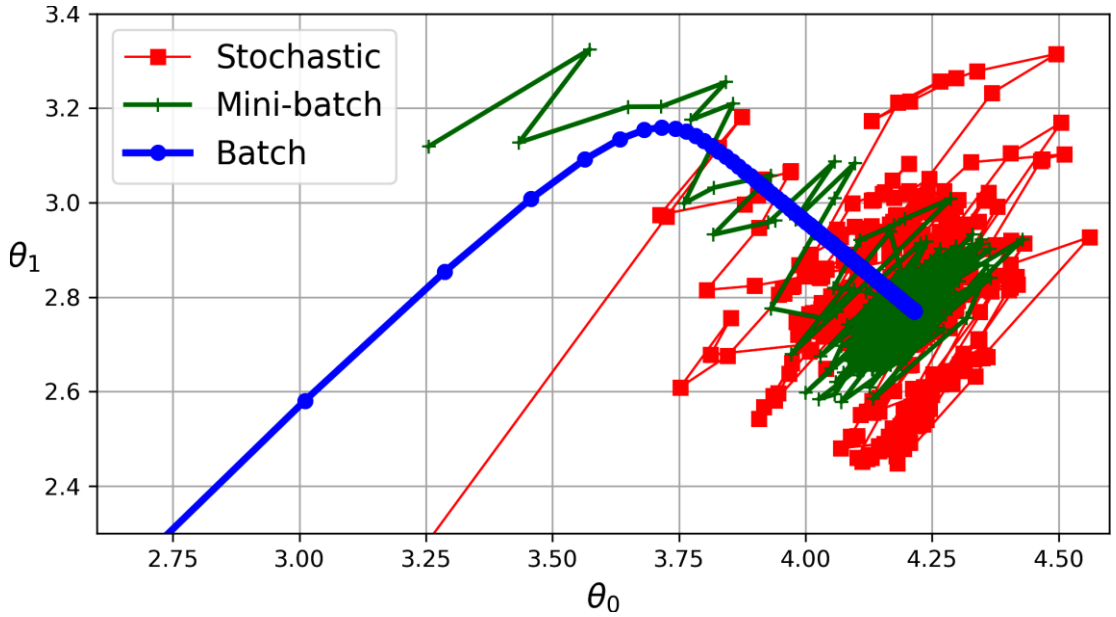
$$\theta_j = \theta_j - \alpha \frac{1}{k} \sum_{i=1}^{k} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$



… Types of GD

- Stochastic GD
  - Advantages
    - Speed: Less computation → Faster
    - Memory efficiency: Only one instance needs to be in memory at each iteration → Makes it possible to train on huge training sets
    - Jump out of local minima → Better chance of finding the global minimum in irregular functions
  - Disadvantage
    - Less Accurate: Noisy updates → Over time it will end up very close to the minimum, but never settling down
    - Slow Convergence: updates the parameters for each training example one at a time → requires more iterations to converge
  - In comparison to mini-batch GD
    - Farther from the minimum
    - More chance to escape from local minima

… Types of GD

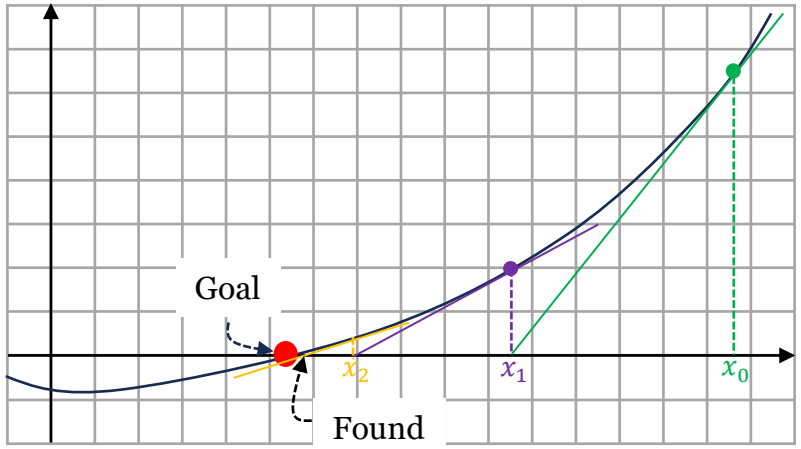


Reference: A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd ed. O'Reilly Media, 2023.

Newton's Method

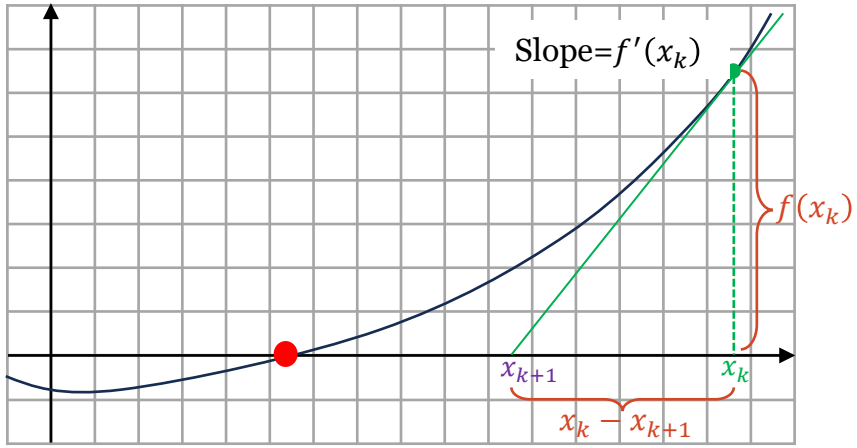
Overview

- An alternative method to gradient descent
- Globally, used to estimate the zeros of a function
  - We can adapt it to be used for optimization



Iterative steps

$$f'(x_k) = \frac{f(x_k)}{x_k - x_{k+1}} \Rightarrow \boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}}$$

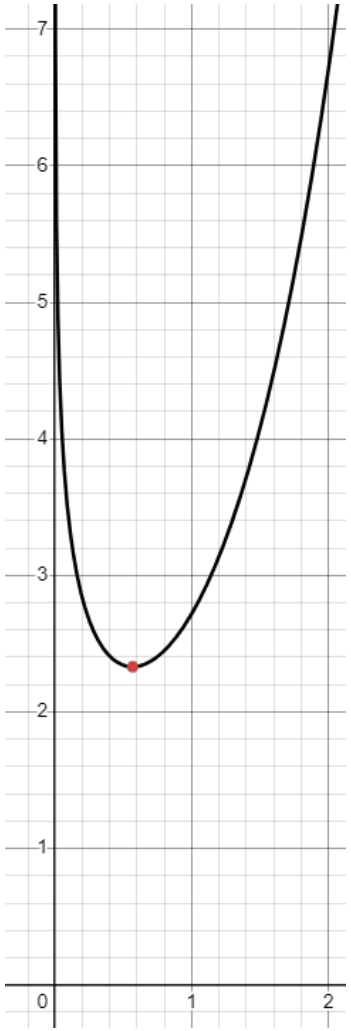


Newton's Method for Optimization

- Goal: Finding the minimum of $f(x)$
  - At the minimum point $f'(x)$ equals to zero
- How?
  - Find zeros of $f'(x)$
  - Use the Newton's method
    - Note that we should calculate $\left(f'(x)\right)'$
- Algorithm
  - Start with some x_0
  - Update: $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$
  - Repeat step 2 until you find the candidate for minimum

Example

- Problem: Find the minimum for $f(x) = e^x - \log(x)$
- Solution:
  - Find the zero of this function $f'(x) = e^x - \frac{1}{x}$
  - $f''(x) = e^x + \frac{1}{x^2}$
  - Random initial point: $x_0 = 0.05$
  - Iterate using $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$
    - $x_1 = 0.05 - \frac{\left(e^{0.05} - \frac{1}{0.05}\right)}{\left(e^{0.05} + \frac{1}{0.05^2}\right)} = 0.097$
    - $x_2 = 0.097 - \frac{\left(e^{0.097} - \frac{1}{0.097}\right)}{\left(e^{0.97} + \frac{1}{0.097^2}\right)} = 0.183$
    - $x_3 = 0.320 \rightarrow x_4 = 0.477 \rightarrow x_5 = 0.558 \rightarrow x_6 = 0.567$



Newton's Method to Optimize Functions of Many Variables

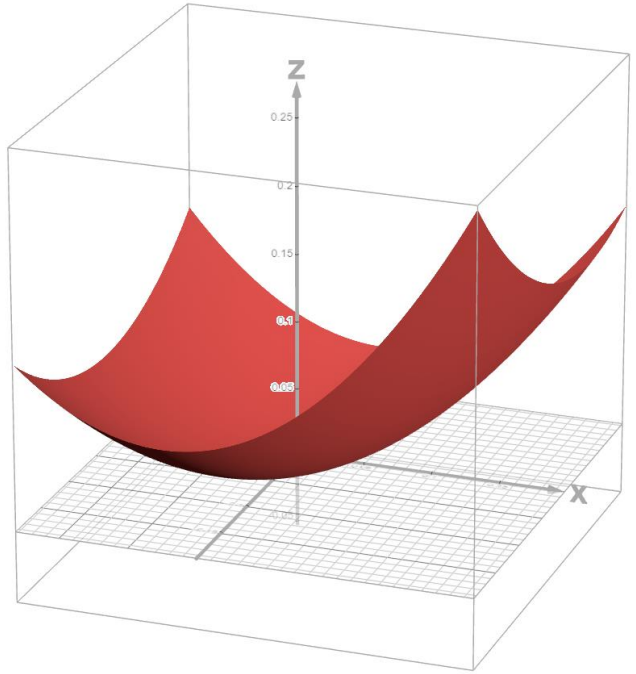
- For one variable
  - $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \Rightarrow x_{k+1} = x_k - f''(x_k)^{-1} f'(x_k)$
- For two variables
  - $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - H^{-1}(x_k, y_k) \nabla f(x_k, y_k)$
- For n variables
  - $\begin{bmatrix} x_{1_{(k+1)}} \\ x_{2_{(k+1)}} \\ \vdots \\ x_{n_{(k+1)}} \end{bmatrix} = \begin{bmatrix} x_{1_{(k)}} \\ x_{2_{(k)}} \\ \vdots \\ x_{n_{(k)}} \end{bmatrix} - H^{-1}\left(x_{1_{(k)}}, x_{2_{(k)}}, \dots, x_{n_{(k)}}\right) \nabla f\left(x_{1_{(k)}}, x_{2_{(k)}}, \dots, x_{n_{(k)}}\right)$

Example

$$f(x,y) = x^4 + 0.8y^4 + 4x^2 + 2y^2 - xy - 0.2x^2y$$

$$\nabla f(x,y) = \begin{bmatrix} 4x^3 + 8x - y - 0.4xy \\ 3.2y^3 + 4y - x - 0.2x^2 \end{bmatrix}$$

$$H(x,y) = \begin{bmatrix} 12x^2 + 8 - 0.4y & -1 - 0.4x \\ -1 - 0.4x & 9.6y^2 + 4 \end{bmatrix}$$



… Example

- Start at a random point $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$
- $\nabla f(4,4) = \begin{bmatrix} 277.6 \\ 213.6 \end{bmatrix}$, $H(4,4) = \begin{bmatrix} 198.4 & -2.6 \\ -2.6 & 157.6 \end{bmatrix}$
- $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} - \begin{bmatrix} 198.4 & -2.6 \\ -2.6 & 157.6 \end{bmatrix}^{-1} \begin{bmatrix} 277.6 \\ 213.6 \end{bmatrix} = \begin{bmatrix} 2.58 \\ 2.61 \end{bmatrix}$
- $\nabla f(2.58, 2.61) = \begin{bmatrix} 84.25 \\ 63.4 \end{bmatrix}$, $H(2.58, 2.61) = \begin{bmatrix} 86.83 & -2.03 \\ -2.03 & 69.39 \end{bmatrix}$
- $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2.58 \\ 2.61 \end{bmatrix} - \begin{bmatrix} 86.83 & -2.03 \\ -2.03 & 69.39 \end{bmatrix}^{-1} \begin{bmatrix} 84.25 \\ 63.4 \end{bmatrix} = \begin{bmatrix} 1.59 \\ 1.67 \end{bmatrix}$
- …
- $\begin{bmatrix} x_8 \\ y_8 \end{bmatrix} = \begin{bmatrix} 4.15 \times 10^{-17} \\ -2.05 \times 10^{-17} \end{bmatrix}$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - H^{-1}(x_k, y_k)\nabla f(x_k, y_k)$$

$$\nabla f(x,y) = \begin{bmatrix} 4x^3 + 8x - y - 0.4xy \\ 3.2y^3 + 4y - x - 0.2x^2 \end{bmatrix}$$

$$H(x,y) = \begin{bmatrix} 12x^2 + 8 - 0.4y & -1 - 0.4x \\ -1 - 0.4x & 9.6y^2 + 4 \end{bmatrix}$$